

Oleg Sergiyenko
Wendy Flores-Fuentes
Paolo Mercorelli *Editors*

Machine Vision and Navigation



Springer

Machine Vision and Navigation

Oleg Sergiyenko • Wendy Flores-Fuentes
Paolo Mercorelli
Editors

Machine Vision and Navigation

 Springer

Editors

Oleg Sergiyenko
Universidad Autónoma de Baja California
Mexicali
Baja California, Mexico

Wendy Flores-Fuentes
Universidad Autónoma de Baja California
Mexicali
Baja California, Mexico

Paolo Mercorelli
Leuphana University of Lueneburg
Lueneburg, Low Saxony
Niedersachsen, Germany

ISBN 978-3-030-22586-5 ISBN 978-3-030-22587-2 (eBook)
<https://doi.org/10.1007/978-3-030-22587-2>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Machine vision to provide spatial coordinate's measurement has developed in a wide range of technologies for multiple fields of applications such as robot navigation, medical scanning, and structural health monitoring, to mention some. Machine vision methods also have applications in search, classification, industrial process robotics (monitoring tools capable of visualizing various phenomena that occur during industrial process), rescue, vigilance, mapping, dangerous objects/subjects detection, and other areas where machine control based on vision plays an important role. The computer vision has guided the machine vision to the tendency of duplicating the abilities of human vision by electronically perceiving and understanding an image for high-dimensional data and optimizing the data storage requirement and the time processing due to the complexity of algorithms to extract important patterns and trends to understand what data says.

Autonomous mobile robots are every day more common; they can be commercially available dotted with machine vision capabilities for diverse tasks and applications, like surveillance, 3D model reconstruction, localization and mapping based on stereo vision, cleaning, medical assistance, and assist handicapped and elderly people. All these robots missions require to the ability to work interactively in human environments and with online learning. Mobile robots with machine vision can be set up to detect, track, and avoid obstacles for optimal navigation. They can also estimate its pose and construct a 3D structure of a scene. Their vision can be based on stand-alone sensors or cameras based on sensors, filters, lens, and electronic and mechanical focus. Everyday cameras are more considered in research projects because they are affordable, inexpensive, robust, and compact. They capture a large amount of data reflecting both the photometric and geometric properties of the observed scene; however, they require considerable computing power and have a number of limitations related to the used sensors and their whole optical path design.

In this sense, *Machine Vision and Navigation* is important to modern science and industrial practical implementation. Hence, it is necessary to create new algorithms and systems to improve their performance. Although machine vision, control systems, and navigation applications for research and industrial areas are the primary interest in exploration, contaminated areas after natural and man-made

disasters on our planet, as well as of unknown terrains on other planets, are also important, and the conjunctive use of these technologies and automatic systems is desirable.

The chapters in this book relate to contributions in machine vision applications. Each book chapter shows the state of the art in stand-alone sensors, cameras, methods, and 3D and 2D technologies in machine vision as well as the novel strategies in navigation performance.

These contributions are focused on optoelectronic sensors, 3D and 2D machine vision technologies, robot navigation, control schemes, motion controllers, intelligent algorithms, and vision systems, particularly on applications of unmanned aerial vehicle, autonomous and mobile robots, industrial inspection applications, and structural health monitoring. Recent advanced research in measurement and others areas where 3D and 2D machine vision and machine control play an important role as well as significant surveys and reviews about machine vision applications.

This book covers both theories and application of machine vision and navigation topics. In our opinion, this book should be attractive for potential consumers/citers because in our vision it is a well-balanced source of novel technologies in the area of machine vision and navigation with an explicit overview of recently existing systems, giving the comparative analysis of its features, advantages, and disadvantages. The topics are of interest to readers from a diverse audience in different areas of specialty as electrical, electronics, and computer engineering, technologists, and nonspecialist readers. The book is intended to be used as a text and reference work on advanced topics in machine vision and navigation. It is dedicated to academics, researchers, advanced-level students, and technology developers who will find this text useful in furthering their research exposure to pertinent topics in *Machine Vision and Navigation* and assisting in their future own research efforts in this field.

An Overview of Machine Vision and Navigation

The combination of machine vision and navigation is most promising nowadays, in recent years, we are finally seeing the full-scale release of daily-use devices, such as robot cleaners, robot assistants for the elderly, and so on. From previous experience, this is the best marker of the upcoming boom of demand of novel competitive technologies in the field. Covering all the topics of study in this book would be impossible; however, the most significant have been selected. The book contains 26 chapters which have been classified into five parts: (1) Image and Signal Sensors; (2) Detection, Tracking, and Stereoscopic Vision Systems; (3) Pose Estimation, Avoidance of Objects, Control, and Data Exchange for Navigation; (4) Aerial Imagery Processing; and (5) Machine Vision for Scientific, Industrial, and Civil Applications. These are briefly described in the following.

Chapter 1 is dedicated to image sensors and signal sensors used in current machine vision systems. Some of them suffer from low dynamic range and poor color constancy and are brittle and unmalleable, limiting their use in applications for which

there will be considerable demand in the future. Most approaches aiming to resolve these inadequacies focus on developing improvements in the lighting, and software (processing algorithms) or hardware surrounding the photosensor such as filters is presented. Also discussed are other strategies that involve changing the architecture of the image sensor and the photo-sensing material; both have experienced recent success. Although they are yet to break fully into the market, image sensors developed from alternative solution-processed materials such as organic semiconductors and organohalide perovskites have immense potential to address the above issues and to “disrupt” machine vision technology.

Chapter 2 proposes a novelty passive vision sensor with a 360° horizontal field of view for mobile robots. With the implementation of this sensor, the robots can be provided with the ability of rapid detection of objects with a peripheral and central vision. The development of this sensor has been inspired by the peripheral/foveal typical vision in cooperation with the visual perception of vertebrates. It is based on the exploit of a catadioptric camera, while a rotating perspective camera makes it possible to measure distances, focusing attention on an already detected object, with a simple calibration methodology of the hybrid field-of-view vision systems. The sensor has been set up as a stand-alone and real-time sensor. It is a self-contained unit hosted in a single-board embedded computer with parallel processing capabilities that can be installed on any mobile robot, even those that have very limited computing power.

Chapter 3 focuses on the color and depth sensing technologies and analyzes how they play an important role in localization and navigation in unstructured environments. It discusses the important role of scanning technologies in the development of trusted autonomous systems for robotic and machine vision with an outlook for areas that need further research and development. A review of sensor technologies for specific environments is included, with special focus on the selection of a particular scanning technology to deal with constrained (indoor) or unconstrained (outdoor) environments. Fundamentals, advantages, and limitations of color and depth (RGB-D) technologies such as stereo vision, time of flight, and structured light and shape from shadow are discussed in detail. Strategies to deal with lighting, color constancy, occlusions, scattering, haze, and multiple reflections are evaluated in detail. It also introduces the latest developments in this area by discussing the potential of emerging technologies, such as dynamic vision and focus-induced photoluminescence.

Chapter 4 is a work developed for the construction of mixed image processor (IP) and neural networks (NNs) and image intensity transformation and the fundamentals of continuous logic cell (CLC) design based on current mirrors (CM) with functions of preliminary analog processing. The intention of the authors is to create video sensors and processors for parallel (simultaneous by pixel) image processing with advanced functionality and multichannel picture outputs to work in particular in hardware with high-performance architectures of neural networks, convolutional neural structures, parallel matrix-matrix multipliers, and special-processor systems. The theoretical foundations are analyzed. The mathematical apparatus of the matrix and continuous logic, their basic operations, and their functional completeness are

described. The evaluation of their advantages and prospects for application in the design of biologically inspired devices and systems for processing and analysis of array signals are presented. It is demonstrated that some functions of continuous logic, including operations of normalized equivalence of vector and matrix signals, and the operation of a limited difference in continuous logic are a powerful basis for designing improved smart micro-cells for analog transformations and analog-digital encodings.

Chapter 5 proposes the use of a robotic total station assisted with cameras for detection and tracking of targets that are not signalized by reflectors. It introduces the principles of standard total stations, defining them as “modern geodetic multi-sensor systems measuring horizontal and vertical angles as well as distances using time-of-flight methods, thus delivering 3D-coordinates for static as well as moving objects.” However, it focuses on the equipment of these systems with cameras and the application of photogrammetric techniques for the development of robotic image-assisted total stations for static and kinematic objects. Some examples of applications are described and a quality control study result is presented.

Chapter 6 offers a clear presentation of the methods and mathematical models for coordinate estimation using radar technologies and problems related to the recognition of object characteristics (landmarks) for mobile autonomous robots. Basically, it is devoted to the actual problem of navigating mobile autonomous robots on unknown terrains in the absence of GPS. Such a problem is considered solved if the robot is capable to detect a landmark and estimate own coordinates relative to the landmark. A reliable method for solving the problem is the simultaneous use of several measuring systems operating on different physical principles. In classical radar, the reliable detection of the echo signals from immovable landmark, which differ little from the echo signals that are reflected from the surrounding area, is impossible. Comparison of such signals is carried out in the chapter for various terrains at different lengths of electromagnetic waves. It is found that the only difference between them is the possible amplitude jump of signal, reflected from the landmark. This jump occurs during the movement of the robot or scanning the space by the robot antenna. The probability of detecting such a jump, the accuracy of the amplitude estimation, and the speed of the device operation are analyzed in the chapter based on the developed system of stochastic differential equations.

Chapter 7 overviews different machine vision systems in agricultural applications. Several different applications are presented but a machine vision system which estimates fruit yield, an example of an orchard management application, is discussed at length. From the farmer’s perspective, an early yield prediction serves as an early revenue estimate. From this prediction, resources, such as employees and storage space, can more efficiently be allocated, and future seasons can be better planned. The yield estimate is accomplished using a camera with a color filter that isolates the blossoms on a tree when the tree is in its full blossom. The blossoms in the resulting image can be counted and the yield estimated. An estimate during the blossom period, as compared to when the fruit has begun to mature, provides a crop yield prediction several months in advance. Discussed as well, in this chapter, is a machine vision system which navigates a robot through orchard rows. This system

can be used in conjunction with the yield estimation system, but it has additional applications such as incorporating a water or pesticide system, which can treat the trees as it passes by. To be effective, this type of system must consider the operating scene as it can limit or constrain the system effectiveness. Such systems tend to be unique to the operating environment.

Chapter 8 presents a deep review of stereoscopic vision systems (SVS), and their description, classification (geometric configuration, quantity of cameras, and other characteristics related with mathematical and computer processing), advantages, disadvantages, and applications in the current state of the art are stated. It is also noted that geometries of the SVS's shown in this chapter are ideal and are not considered factors that could affect the accuracy of measurements. The aim of the chapter is to provide information for everyone who wants to implement an SVS and needs an introduction to several available options to use the most convenient according to a specific application.

Chapter 9 is focused on the development of machine vision for robots, robot pose estimation, and 3D restructure of scenes through a set of matched correspondences and features extracted from multiple images. It provides modern and advanced strategies for image filtering and image feature extraction. It concentrates on stereo vision noise source that results in a 3D reconstruction of a scene. Strategies for image filtering and feature extraction are described based on techniques, such as Kalman Filter (KF), extended Kalman filter (EKF), and unscented Kalman filter (UKF). These filters are presented in order to increase the efficiency of visual simultaneous localization and mapping (VSLAM) algorithm to increase its efficiency. Practical examples in the field of robotics vision research are described, like pose tracking using UKF and stereo vision and localization approach-based 2D-landmarks map.

Chapter 10 is dedicated to the development of mathematical fundamentals for pose estimation. Pose estimation requires optimally estimating translation and rotation. The chapter is focused on rotation, since it involves nonlinear analysis. It is demonstrated how the computation can be done systematically if it is exploited the fact that the set of rotations forms a group of transformations, called the "special orthogonal group." A linear space spanned by infinitesimal rotations called the "Lie algebra" is defined. A computational procedure for minimizing the optimization function of a rotation based on Lie algebra formulation is described and applied to three computer vision problems: (1) Given two sets of 3D points, it is optimally estimated the translation and rotation between them in the presence of inhomogeneous anisotropic noise. (2) Given corresponding points between two images, it is optimally computed the fundamental matrix. (3) It is described the procedure of bundle adjustment for computing, from images of multiple points in the scene taken by multiple cameras, the 3D locations of all the points, and the postures of all the cameras as well as their internal parameters.

Chapter 11 shows a methodology for the accurate generation and tracking of closed trajectories over arbitrary, large surfaces of unknown geometry, using a robot whose control is based on the use of a non-calibrated vision system. The proposed technique referred to as camera-space manipulation is combined with a geodesic-mapping approach, with the purpose of generating and tracking a trajectory stored as

a CAD model, over an arbitrarily curved surface, along with a user-defined position and orientation. A measure used to diminish the distortion caused by the mapping procedure and a technique for achieving closure of a given closed path, when this is tracked over large, non-developable surfaces, are presented herein. The performance of the proposed methodology was evaluated using an industrial robot with a large workspace whose geometry is not known in advance, combined with structured lighting used to reduce the complexity of the image analysis process.

Chapter 12 deals with generic image-based visual servoing control structure with onboard camera, based on the passivity theory and application. It gives a mathematical approach and a detailed literature research including also contributions of recent publications. The authors prove the convergence to zero of the control error and its robustness in the context of L_2 -gain performance. A unified passivity-based visual servoing control structure considering a vision system mounted on the robot is presented. This controller is suitable to be applied for robotic arms, mobile robots, as well as mobile manipulators. The proposed control law makes the robot able to perform a moving target tracking in its workspace. Taking advantage of the passivity properties of the control system and considering exact knowledge of the target's velocity, the asymptotic convergence to zero of the control errors is proved. Later, it is carried a robustness analysis out based on L_2 -gain performance, hence proving that control errors are ultimately bounded even when there exist bounded errors in the estimation of the target velocity. Both numerical simulation and experimental results illustrate the performance of the algorithm in a robotic manipulator, in a mobile robot, and also in a mobile manipulator.

Chapter 13 is about data exchange and task of navigation for robotic group. Robotic group collaboration in a densely cluttered terrain is one of the main problems in mobile robotics control. The chapter describes the basic set of tasks solved in model of robotic group behavior during the distributed search of an object (goal) with the parallel mapping. Navigation scheme uses the benefits of authors' original technical vision system (TVS) based on dynamic triangulation principles. According to the TVS output data were implemented fuzzy logic rules of resolution stabilization for improving the data exchange. The dynamic communication network model was modified and implemented the propagation of information with a feedback method for data exchange inside the robotic group. For forming the continuous and energy-saving trajectory, the authors are proposing to use two-step post-processing method of path planning with polygon approximation. Combination of our collective TVS scans fusion and modified dynamic data exchange network forming method with dovetailing of the known path planning methods can improve the robotic motion planning and navigation in unknown cluttered terrain.

Chapter 14 proposes a hierarchical navigation system combining the benefits of perception space local planning and allocentric global planning. Perception space permits computationally efficient 3D collision checking, enabling safe navigation in environments that do not meet the conditions assumed by traditional navigation systems based on planar laser scans. Contributions include approaches for scoring and collision checking trajectories in perception space. Benchmarking results show the advantages of perception space collision checking over popular alternatives

in the context of real-time local planning. Simulated experiments with multiple robotic platforms in several environments demonstrate the importance of 3D collision checking and the utility of a mixed representation hierarchical navigation system.

Chapter 15 corresponds to a deep overview regarding autonomous mobile vehicles for wheeled ground applications. The different autonomy levels of vehicles are approached. The main concepts from path planning, going through the basic components that an autonomous vehicle must have, all the way to the perception it has of its environment, including the identification of obstacles, signs, and routes, are presented. The most commonly used hardware for the development of these vehicles is discussed. In the last part of this chapter, a case study, “Intelligent Transportation Scheme for Autonomous Vehicles in Smart Campus,” is incorporated in order to help illustrate the goal of the chapter. Finally, an insight is included on how the innovation on business models can and will change the future of vehicles.

Chapter 16 is devoted to the approach of passive combined correlation-extreme systems implementing the survey-comparative method for recognition and analysis of images obtained from the machine vision system of a flying robot, which is able to significantly improve the correct localization of the objects in the image frame. A basic model for the radiometric channel operation of the correlation-extreme navigation systems is proposed. The factors that lead to distortions of the decisive function formed by the combined correlation-extreme navigation system of flying robots in a developed infrastructure are allocated. A solution of the problem of autonomous low-flying flying robots navigation in a developed infrastructure using the radiometric channel extreme correlation navigation systems (CENS), when the size of the solid angle of associated object is much larger than the size of the partial antenna directivity diagram (ADD), is proposed.

Chapter 17 is focused in the description of an analytic image stabilization approach where pixel information from the focal plane of the camera is stabilized and georegistered in a global reference frame. The aerial video is stabilized to maintain a fixed relative displacement between the moving platform and the scene. The development of the algorithm that is able to stabilize aerial images using its available weak/noisy GPS and IMU measurements, based on the use of analytically defined homographies between images and minimizing the cost function on a 2D equation space, is presented. The algorithm is applied in the Defense Advanced Research Projects Agency (DARPA) video and image retrieval and analysis tool (VIRAT) data set and wide area motion images (WAMI).

Chapter 18 describes a visual servo controller designed for an unmanned aerial vehicle dedicated to tracking vegetable paths. In the inspection and data collection of large areas as crop fields, where an aerial vehicle should follow an object’s line accurately, autonomous flight is a desirable feature with unmanned aerial vehicles. To attain this objective, three visual servo controllers are proposed; one of them is position based and the other two are image based using inverse Jacobian and concepts of passivity, respectively. All controllers are developed based on the kinematic model of the vehicle, and a dynamic compensation is designed to be added in cascade with the kinematic one. The performance of the control systems is compared through simulation results. The main contribution is the development of the image-

based controller using passivity properties of the system, the stability and robustness analysis, and the comparative performance with other controllers when used for an unmanned aerial vehicle following vegetal lines. These comparative results are valuable to choose the appropriate driver for a specific application.

Chapter 19 is the result of a deep study and research of multimedia compression advances, focusing on the use of the integer discrete cosine transform, the wavelet transform, and fovea centralis. Data compression is concerned with minimization of the number of information-carrying units used to represent a given data set. Such smaller representation can be achieved by applying coding algorithms. Coding algorithms can be either lossless algorithms that reconstruct the original data set perfectly or lossy algorithms that reconstruct a close representation of the original data set. Both methods can be used together to achieve higher compression ratios. Lossless compression methods can either exploit statistical structure of the data or compress the data by building a dictionary that uses fewer symbols for each string that appears on the data set. Lossy compression on the other hand uses a mathematical transform that projects the current data set onto the frequency domain. The coefficients obtained from the transform are quantized and stored. The quantized coefficients require less space to be stored.

Chapter 20 shows a method to solve the stairway localization and recognition problem for both indoor and outdoor cases by using a convolutional neural network technique. This work has been motivated because for blind and visually impaired persons, this assistive technology application has an important impact in their daily life. The proposed algorithm should be able to solve the problem of stair classification for indoor and outdoor scenes. The proposed idea describes the strategy for introducing an affordable method that can recognize stairways without taking into account the environments. Firstly, this method uses stair features to classify images by using convolutional neural networks. Secondly, stairway candidate is extracted by using the Gabor filter, a linear filter. Thirdly, the set of lines belonging to the ground plane are removed by using the behavioral distance measurement between two consecutive frames. Finally, it is extracted from this step the tread depth and the riser height of the stairways.

Chapter 21 gives a deep review of new- and advanced-phase triangulation methods for 3D-shape measurements in scientific and industrial applications. The mathematical methods for phase triangulation are presented, which allow the measurement of 3D data under the conditions of arbitrary light-scattering properties of the scanning surface, varying measurement setting external illumination and limited depth of field of optical elements of the source and receiver of optical radiation. The book chapter provides a deep mathematical approach about the proposed steady-state method for decoding phase images and presents a method for nonlinearity compensation of the source-receiver path of optical radiation in 3D measurements. The application of the proposed methods provides higher metrological characteristics of measuring systems and expands the functionality and the range of application of optical-electronic systems for geometric control in the production environment.

Chapter 22 presents a thermal image processing method to monitor a moving melt pool of a blown powder deposition process using infrared thermography. Thereby, the moving melt pool is created on a substrate material by the use of a laser and a motorized work table, where the material is (stainless steel 316) deposited in a layer-by-layer sequence on the substrate material. The steel is placed in powder form on the substrate and brought to its melting point by means of a 1 kW fiber laser with a wavelength of 1064 nm. By controlling a fixed melting pot size in closed-loop configuration, a consistent material deposition and layer thickness of the deposited material are ensured. For the feedback of the closed-loop control, an energy management system and a height control system are used to track the total spectral radiance of the melt pool and to track the top of the deposited material. The chapter gives a good and practical overview of the blown powder deposition process using infrared thermography and names the used technologies to implement the melting and tracking process. It uses Planck's law to define the spectral radiance of the melt pool for the energy management system. It also presents infrared thermographs to detect different temperature regions of the melt pool.

Chapter 23 describes the importances of image processing of measurement signals that are contaminated with noise for accurate fault detection and isolation in machines. This chapter presents processing filters to detect step changes in noisy diagnostic signals of a gas turbine, which the authors use as indicator for an onset of a single fault of these signals. By using the process filters, the noise of the gas turbine diagnostic signals is reduced and then examined for a step change. Various linear and nonlinear process filters are described and compared, where the weighted recursive median (WRM) filter is highlighted for good noise reduction. Also, the ant colony optimization (ACO) method is used to calculate the integer weights of the weighted recursive median filter.

Chapter 24 proposes a new method to control and automatize the position of three-axis piezoelectric nano-manipulators that handle a GSG nanoprobe to ensure the precise positioning of the probe on the substrate under test. The method is based on a measurement setup that consists of a vector network analyzer (VNA) connected through coaxial cables to miniaturized homemade coplanar waveguide (CPW) probes (one signal contact and two ground contacts), which are themselves mounted on three-axis piezoelectric nano-manipulators SmarAct™. The device under test (DUT) is positioned on a sample holder equipped also with nano-positioners and a rotation system with μ -degree resolution. The visualization is carried out by a scanning electron microscope (SEM) instead of conventional optics commonly found in usual on-wafer probe stations. This study addresses the challenge related to the control of nano-manipulators in order to ensure precisely the contact between the probe tips and the DUT to be characterized.

Chapter 25 shows the design of an industrial inspection system for plastic parts. The development of user-friendly design and training tool for convolutional neural networks (CNNs) and support vector machines (SVMs) as an application development environment based on MATLAB is presented. As the first test trial, an application of deep CNN (DCNN) for anomaly detection is developed and trained using a large number of images to distinguish undesirable small defects such as crack,

burr, protrusion, chipping, spot, and fracture phenomena that occur in the production process of resin molded articles. Then, as the second test trial, a SVM incorporated with the AlexNet and another SVM incorporated with our original sssNet are, respectively, designed and trained to classify sample images into accepting as OK or rejecting as NG categories with high categorization rate. In the case of these SVMs, the training can be conducted by using only images of OK category. The AlexNet and the sssNet are different types of DCNNs, whose compressed feature vectors have 4096 and 32 elements, respectively. The two lengths of compressed feature vectors are used as the inputs for the two types of SVMs, respectively. The usability and operability of the developed design and training tool for DCNNs and SVMs are demonstrated and evaluated through training and classification experiments.

Chapter 26 is dedicated to a structural health monitoring application. It describes that due to the increase of frequency and weight of commercial ship trips in waterways, bridges are more vulnerable than ever to ship-bridge collision accidents. It explains that there are plenty of reports of such cases all over the world, leading to millions of economic losses. For ancient bridges, irreparable damage might come in the sense of cultural value except for economic losses. The development of computer vision-based technology provides an active defense method to prevent the damage in advance. This chapter presents a computer vision-based method for ship-bridge collision assessment and warning for an ancient arch bridge over the Beijing-Hangzhou Grand Canal in Hangzhou, China. The structural characteristic and current status of the arch bridge were analyzed. The traffic volume and parameters of passing ships including velocity and weight were investigated. Water area in both sides of the bridge was divided into three different security districts corresponding to different warning levels. Image processing techniques were exploited to identify the types of ships for tracking. The potential of ship-bridge collision was assessed, and warning was generated according to the security evaluation.

Mexicali, Mexico
Mexicali, Mexico
Lueneburg, Germany

Oleg Sergiyenko
Wendy Flores-Fuentes
Paolo Mercorelli

Acknowledgment

It is a pleasure to offer our acknowledgment to all the *authors* that have contributed with a chapter in this book *Machine Vision and Navigation*. Every author has given their best effort in writing his most recent research findings. We feel satisfied with the academic product generated with an international vision. One hundred researchers around the world have collaborated in this project, representing the participation of 19 countries: Argentina, Australia, Canada, China, Ecuador, Egypt, France, Germany, India, Japan, Mexico, Panamá, Poland, Russia, South Korea, Spain, Syria, Ukraine, and the USA. A whole list of the authors with the affiliations and biographies is provided in the “Contributors” and “About the Authors” sections of this book.

We are delighted and thankful to have received the support of researchers in the areas of machine vision, navigation, robotics, control, and artificial intelligences. We extend our acknowledgment to all the *reviewers* who have done a great job reading and suggesting improvements for each chapter: Alexander Gurko, Daniel Hernández-Balbuena, Danilo Cáceres-Hernández, Fabián N. Murrieta-Rico, Julio C. Rodríguez-Quíñonez, Lars Lindner, Moisés Rivas-López, Moisés J. Castro-Toscano, Vera Tyrsa, and Wilmar Hernández Perdomo.

Acknowledgments go also for the editorial board and the officials at Springer International AG for their invaluable efforts, great support, and valuable advice for this project toward the successful publication of this book. We also want to thank our institutions Universidad Autónoma de Baja California and Leuphana University of Lueneburg that provided us with a location and time where to develop this project.

Baja California, Mexico
Lueneburg, Germany
Baja California, Mexico

Oleg Sergiyenko
Paolo Mercorelli
Wendy Flores-Fuentes

Contents

Part I Image and Signal Sensors

- 1 Image and Signal Sensors for Computing and Machine Vision: Developments to Meet Future Needs** 3
Ross D. Jansen-van Vuuren, Ali Shahnewaz, and Ajay K. Pandey
- 2 Bio-Inspired, Real-Time Passive Vision for Mobile Robots** 33
Piotr Skrzypczyński, Marta Rostkowska, and Marek Wąsik
- 3 Color and Depth Sensing Sensor Technologies for Robotics and Machine Vision** 59
Ali Shahnewaz and Ajay K. Pandey
- 4 Design and Simulation of Array Cells of Mixed Sensor Processors for Intensity Transformation and Analog-Digital Coding in Machine Vision** 87
Vladimir G. Krasilenko, Alexander A. Lazarev, and Diana V. Nikitovich

Part II Detection, Tracking and Stereoscopic Vision Systems

- 5 Image-Based Target Detection and Tracking Using Image-Assisted Robotic Total Stations** 133
Volker Schwieger, Gabriel Kerekes, and Otto Lerke
- 6 The Methods of Radar Detection of Landmarks by Mobile Autonomous Robots** 171
Oleksandr Poliarus and Yevhen Poliakov
- 7 Machine Vision System for Orchard Management** 197
Duke M. Bulanon, Tyler Hestand, Connor Nogales, Brice Allen, and Jason Colwell

8 Stereoscopic Vision Systems in Machine Vision, Models, and Applications 241
 Luis Roberto Ramírez-Hernández, Julio Cesar Rodríguez-Quiñonez, Moisés J. Castro-Toscano, Daniel Hernández-Balbuena, Wendy Flores-Fuentes, Moisés Rivas-López, Lars Lindner, Danilo Cáceres-Hernández, Marina Kolendovska, and Fabián N. Murrieta-Rico

9 UKF-Based Image Filtering and 3D Reconstruction 267
 Abdulkader Joukhadar, Dalia Kass Hanna, and Etezaz Abo Al-Izam

Part III Pose Estimation, Avoidance of Objects, Control and Data Exchange for Navigation

10 Lie Algebra Method for Pose Optimization Computation 293
 Kenichi Kanatani

11 Optimal Generation of Closed Trajectories over Large, Arbitrary Surfaces Based on Non-calibrated Vision 321
 Emilio J. Gonzalez-Galvan, Ambrocio Loredó-Flores, Isela Bonilla-Gutierrez, Marco O. Mendoza-Gutierrez, Cesar Chavez-Olivares, Luis A. Raygoza, and Sergio Rolando Cruz-Ramírez

12 Unified Passivity-Based Visual Control for Moving Object Tracking 347
 Flavio Roberti, Juan Marcos Toibero, Jorge A. Sarapura, Víctor Andaluz, Ricardo Carelli, and José María Sebastián

13 Data Exchange and Task of Navigation for Robotic Group 389
 Mikhail Ivanov, Oleg Sergiyenko, Vera Tyrsa, Lars Lindner, Miguel Reyes-García, Julio Cesar Rodríguez-Quiñonez, Wendy Flores-Fuentes, Jesús Elías Miranda-Vega, Moisés Rivas-López, and Daniel Hernández-Balbuena

14 Real-Time Egocentric Navigation Using 3D Sensing 431
 Justin S. Smith, Shiyu Feng, Fanzhe Lyu, and Patricio A. Vela

15 Autonomous Mobile Vehicle System Overview for Wheeled Ground Applications 485
 Luis Carlos Básaca-Preciado, Néstor Aarón Orozco-García, Oscar A. Rosete-Beas, Miguel A. Ponce-Camacho, Kevin B. Ruiz-López, Verónica A. Rojas-Mendizabal, Cristobal Capiz-Gómez, Julio Francisco Hurtado-Campa, and Juan Manuel Terrazas-Gaynor

Part IV Aerial Imagery Processing

16 Methods for Ensuring the Accuracy of Radiometric and Optoelectronic Navigation Systems of Flying Robots in a Developed Infrastructure 537
 Oleksandr Sotnikov, Vladimir G. Kartashov, Oleksandr Tymochko, Oleg Sergiyenko, Vera Tyrsa, Paolo Mercorelli, and Wendy Flores-Fuentes

17 Stabilization of Airborne Video Using Sensor Exterior Orientation with Analytical Homography Modeling 579
 Hadi Aliakbarpour, Kannappan Palaniappan, and Guna Seetharaman

18 Visual Servo Controllers for an UAV Tracking Vegetal Paths 597
 Jorge A. Sarapura, Flavio Roberti, Juan Marcos Toibero, José María Sebastián, and Ricardo Carelli

Part V Machine Vision for Scientific, Industrial and Civil Applications

19 Advances in Image and Video Compression Using Wavelet Transforms and Fovea Centralis 629
 Juan C. Galan-Hernandez, Vicente Alarcon-Aquino, Oleg Starostenko, Juan Manuel Ramirez-Cortes, and Pilar Gomez-Gil

20 Stairway Detection Based on Single Camera by Motion Stereo for the Blind and Visually Impaired 657
 Javier E. Sanchez-Galan, Kang-Hyun Jo, and Danilo Cáceres-Hernández

21 Advanced Phase Triangulation Methods for 3D Shape Measurements in Scientific and Industrial Applications 675
 Sergey Vladimirovich Dvoynishnikov, Ivan Konstantinovich Kabardin, and Vladimir Genrievich Meledin

22 Detection and Tracking of Melt Pool in Blown Powder Deposition Through Image Processing of Infrared Camera Data 711
 Sreekar Karnati and Frank F. Liou

23 Image Processing Filters for Machine Fault Detection and Isolation 733
 Ranjan Ganguli

24 Control and Automation for Miniaturized Microwave GSG Nanoprobing 751
 Alaa Taleb, Denis Pomorski, Christophe Boyaval, Steve Arscott, Gilles Dambrine, and Kamel Haddadi

25 Development of Design and Training Application for Deep Convolutional Neural Networks and Support Vector Machines 769
Fusaomi Nagata, Kenta Tokuno, Akimasa Otsuka, Hiroaki Ochi, Takeshi Ikeda, Keigo Watanabe, and Maki K. Habib

26 Computer Vision-Based Monitoring of Ship Navigation for Bridge Collision Risk Assessment 787
Xiao-Wei Ye, Tao Jin, and Peng-Peng Ang

About the Authors 809

Further Reading 839

Index 841

Contributors

Vicente Alarcón-Aquino Department of Computing, Electronics and Mechatronics, Universidad de las Americas Puebla, Cholula, Puebla, Mexico

Hadi Aliakbarpour Computational Imaging and VisAnalysis (CIVA) Lab, EECS, University of Missouri, Columbia, MO, USA

Etezaz Abo Al-Izam Department of Computer Engineering, Faculty of Electrical and Electronic Engineering, University of Aleppo, Aleppo, Syria

Brice Allen Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA

Víctor Andaluz Universidad de las Fuerzas Armadas, Sangolquí, Ecuador

Peng-Peng Ang Department of Civil Engineering, Zhejiang University, Hangzhou, China

Steve Arscott Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520-IEMN, Lille, France

Luis Carlos Básaca-Preciado CETYS Universidad, Mexicali, Mexico

Isela Bonilla-Gutierrez Autonomous University of San Luis Potosi (UASLP), San Luis Potosí, SLP, Mexico

Christophe Boyaval Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520-IEMN, Lille, France

Duke M. Bulanon Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA

Danilo Cáceres-Hernández Grupo de Sistemas Inteligentes, Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá, Panama, Republic of Panama

Cristobal Capiz-Gómez CETYS Universidad, Mexicali, Mexico

Ricardo Carelli Instituto de Automática, UNSJ-CONICET, San Juan, Argentina

Moisés J. Castro-Toscano Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Cesar Chávez-Olivares Campus Sur, UAA, Aguascalientes, AGS, Mexico

Jason Colwell Department of Mathematics and Computer Science, Northwest Nazarene University, Nampa, ID, USA

Sergio Rolando Cruz-Ramírez SLP Campus, ITESM, San Luis Potosi, SLP, Mexico

Gilles Dambrine Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520-IEMN, Lille, France

Sergey Vladimirovich Dvoynishnikov Kutateladze Institute of Thermophysics SB RAS, Novosibirsk, Russia

Shiyu Feng Georgia Tech, School of Mechanical Engineering, Atlanta, GA, USA

Wendy Flores-Fuentes Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Juan C. Galan-Hernandez Department of Computing, Electronics and Mechatronics, Universidad de las Americas Puebla, Cholula, Puebla, Mexico

Ranjan Ganguli Department of Aerospace Engineering, Indian Institute of Science (IISc), Bangalore, Karnataka, India

Pilar Gomez-Gil Department of Electronics and Computer Science, Instituto Nacional de Astrofísica, Tonantzintla, Puebla, Mexico

Emilio J. Gonzalez-Galvan Autonomous University of San Luis Potosi (UASLP), San Luis Potosí, SLP, Mexico

Maki K. Habib The American University in Cairo, Cairo, Egypt

Kamel Haddadi Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520-IEMN, Lille, France
Univ. Lille, IUT A-Département GEII, Lille, France

Dalia Kass Hanna Department of Mechatronics Engineering, Faculty of Electrical and Electronic Engineering, University of Aleppo, Aleppo, Syria

Daniel Hernández-Balbuena Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Tyler Hestand Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA

Julio Francisco Hurtado-Campa CETYS Universidad, Mexicali, Mexico

Takeshi Ikeda Sanyo-Onoda City University, Sanyo-onoda, Japan

Mikhail Ivanov Universidad Autónoma de Baja California (UABC), Instituto de Ingeniería, Mexicali, BC, Mexico

Ross D. Jansen-van Vuuren Department of Chemistry, Queen's University, Kingston, ON, Canada

Tao Jin Department of Civil Engineering, Zhejiang University, Hangzhou, China

Kang-Hyun Jo Intelligent Systems Laboratory, Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea

Abdulkader Joukhadar Department of Mechatronics Engineering, Faculty of Electrical and Electronic Engineering, University of Aleppo, Aleppo, Syria

Ivan Konstantinovich Kabardin Kutateladze Institute of Thermophysics SB RAS, Novosibirsk, Russia

Kenichi Kanatani Professor Emeritus, Okayama University, Okayama, Japan

Sreekar Karnati Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA

Vladimir G. Kartashov Kharkiv National University of Radioelectronics, Kharkiv, Ukraine

Gabriel Kerekes Institute of Engineering Geodesy, University of Stuttgart, Stuttgart, Germany

Marina Kolendovska National University of Radioelectronics, Kharkiv, Ukraine

Vladimir G. Krasilenko Vinnytsia National Technical University, Vinnytsia, Ukraine

Alexander A. Lazarev Vinnytsia National Technical University, Vinnytsia, Ukraine

Otto Lerke Institute of Engineering Geodesy, University of Stuttgart, Stuttgart, Germany

Lars Lindner Instituto de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Frank F. Liou Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA

Ambrocio Loredó-Flores Autonomous University of San Luis Potosí (UASLP), San Luis Potosí, SLP, Mexico

Fanzhe Lyu Georgia Tech, School of Electrical and Computer Engineering, Atlanta, GA, USA

Vladimir Genrievich Meledin Kutateladze Institute of Thermophysics SB RAS, Novosibirsk, Russia

Marco O. Mendoza-Gutiérrez Autonomous University of San Luis Potosi (UASLP), San Luis Potosí, SLP, Mexico

Paolo Mercorelli Leuphana University of Lueneburg, Lueneburg, Germany

Jesús Elías Miranda-Vega Universidad Autonoma de Baja California (UABC), Instituto de Ingeniería, Mexicali, BC, Mexico

Fabián N. Murrieta-Rico Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Fusaomi Nagata Sanyo-Onoda City University, Sanyo-onoda, Japan

Diana V. Nikitovich Vinnytsia National Technical University, Vinnytsia, Ukraine

Connor Nogales Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA

Hiroaki Ochi Sanyo-Onoda City University, Sanyo-onoda, Japan

Néstor Aarón Orozco-García CETYS Universidad, Mexicali, Mexico

Akimasa Otsuka Sanyo-Onoda City University, Sanyo-onoda, Japan

Kannappan Palaniappan Computational Imaging and VisAnalysis (CIVA) Lab, EECS, University of Missouri, Columbia, MO, USA

Ajay K. Pandey School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

Yevhen Poliakov Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

Oleksandr Poliarus Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

Denis Pomorski Univ. Lille, CNRS, Centrale Lille, UMR 9189–CRIStAL–Centre de Recherche en Informatique, Signal et Automatique de Lille, Lille, France
Univ. Lille, IUT A–Département GEII, Lille, France

Miguel A. Ponce-Camacho CETYS Universidad, Mexicali, Mexico

Juan Manuel Ramirez-Cortes Department of Electronics and Computer Science, Instituto Nacional de Astrofisica, Tonantzintla, Puebla, Mexico

Luis Roberto Ramírez-Hernández Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Luis A. Raygoza Centro de Ciencias Basicas, UAA, Aguascalientes, AGS, Mexico

Miguel Reyes-García Universidad Autonoma de Baja California (UABC), Instituto de Ingeniería, Mexicali, BC, Mexico

Moisés Rivas-López Instituto de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Flavio Roberti Instituto de Automática, UNSJ-CONICET, San Juan, Argentina

Julio Cesar Rodríguez-Quiñonez Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico

Verónica A. Rojas-Mendizabal CETYS Universidad, Mexicali, Mexico

Oscar A. Rosete-Beas CETYS Universidad, Mexicali, Mexico

Marta Rostkowska Institute of Control, Robotics, and Information Engineering, Poznań University of Technology, Poznań, Poland

Kevin B. Ruiz-López CETYS Universidad, Mexicali, Mexico

Javier E. Sanchez-Galan Grupo de Investigación en Biotecnología, Bioinformática y Biología de Sistemas, Centro de Producción e Investigaciones Agroindustriales (CEPIA), Universidad Tecnológica de Panamá, Panama, Republic of Panama
Institute of Advanced Scientific Research and High Technology, Panama, Republic of Panama

Jorge A. Sarapura Instituto de Automática, UNSJ-CONICET, San Juan, Argentina

Volker Schwieger Institute of Engineering Geodesy, University of Stuttgart, Stuttgart, Germany

José María Sebastián Centro de Automática y Robótica, Universidad Politécnica de Madrid, Madrid, Spain

Guna Seetharaman Advanced Computing Concepts, U.S. Naval Research Laboratory, Washington, DC, USA

Oleg Sergiyenko Universidad Autónoma de Baja California (UABC), Instituto de Ingeniería, Mexicali, BC, Mexico

Ali Shahnewaz School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

Piotr Skrzypczyński Institute of Control, Robotics, and Information Engineering, Poznań University of Technology, Poznań, Poland

Justin S. Smith Georgia Tech, School of Electrical and Computer Engineering, Atlanta, GA, USA

Oleksandr Sotnikov Scientific Center of Air Forces, Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Oleg Starostenko Department of Computing, Electronics and Mechatronics, Universidad de las Americas Puebla, Cholula, Puebla, Mexico

Alaa Taleb Univ. Lille, CNRS, Centrale Lille, UMR 9189-CRISTAL-Centre de Recherche en Informatique, Signal et Automatique de Lille, Lille, France

Juan Manuel Terrazas-Gaynor CETYS Universidad, Mexicali, Mexico

Juan Marcos Toibero Instituto de Automática, UNSJ-CONICET, San Juan, Argentina

Kenta Tokuno Sanyo-Onoda City University, Sanyo-onoda, Japan

Oleksandr Tymochko Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Vera Tyrsa Universidad Autónoma de Baja California (UABC), Facultad de Ingeniería, Mexicali, BC, Mexico

Patricio A. Vela Georgia Tech, School of Electrical and Computer Engineering, Atlanta, GA, USA

Marek Wasik Institute of Control, Robotics, and Information Engineering, Poznań University of Technology, Poznań, Poland

Keigo Watanabe Okayama University, Okayama, Japan

Xiao-Wei Ye Department of Civil Engineering, Zhejiang University, Hangzhou, China

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
4WDDMR	Four-wheeled differential drive mobile robot
AAM	Auto-associative memory
AB	Aborted
ABAC	Adaptive binary arithmetic coding
ABC	Analog-digital basic cell
ACO	Ant colony optimization
ACS	Automated control systems (Meaning in Chap. 16)
ACS	Ant colony system (Meaning in Chap. 23)
AD*	Anytime D*
ADC	Analog to digital converter
ADD	Antenna directivity diagram
AFV-SPECK	Adaptive fovea centralis set partitioned embedded block codec
AlexNet	A well-known convolutional neural network designed by Alex Krizhevsky
AM	Associative memory (Meaning in Chap. 4)
AM	Additive manufacturing (Meaning in Chap. 22)
AMCW	Amplitude modulated continuous wave
AO	Absolute orientation
AP	Antenna pattern
APD	Avalanche photo diode
API	Application programming interface
AS/RS	Automated storage and retrieval system
ASCII	American standard code for information interchange
ASIC	Application-specific integrated circuit
AVC	Advance video coding
AWFV-Codec	Adaptive wavelet/fovea centralis-based codec
BA	Bundle adjustment
BC	Basic cell (Meaning in Chap. 4)

BC	Bumper collision (Meaning in Chap. 14)
BDS	BeiDou navigation satellite system
Bel	Degree of belief
BGR	Blue green red color space
BIA	Binary image algebra
BLS	Bottommost line segments
BM	Block matching
BP	Back propagation algorithm
BPD	Blown powder deposition
bpp	Bits per pixel
CAD	Computer-aided design
CAF	Correlation analysis field
Caffe	Convolutional architecture for fast feature embedding
CAS	Computer-assisted surgery
CCC	Coefficient of cross correlation
CCCA	Current-controlled current amplifiers on current mirror multipliers
CCD	Charge-coupled device
CDF9/7	Cohen-Daubechies-Feauveau wavelet
CDNE	Complementary double NE
CENS	Channel extreme correlation navigation systems
CENS – I	CENS in which information is currently removed at a point
CENS – II	CENS in which information is currently removed from a line
CENS– III	CENS in which information is currently removed from an area (frame)
CFA	Color filter array
CI	Current image
CIF	Common intermediate format
CIS	Cmos image sensors
CL	Continuous logic
CLC	Continuous logic cell
CLEM	Continuous logical equivalence model
CLF	Continuous logic function
CM	Current mirror
CML	Concurrent mapping and localization
CMM	Current multiplier mirror
CMOS	Complementary metal-oxide semiconductor
CMYK	Cyan, magenta, yellow, black color space
CNC	Computer numerical control
CNN	Convolutional neural network
CNN-CRF	Convolutional neural network-conditional random field
Cov	Covariance
CPR	Cycles per revolution
CPU	Central processing unit
CPW	Coplanar waveguide
CQDs	Colloidal quantum dots

CS	Control systems
CSM	Camera-space manipulation
C-Space	Configuration space
CUDA	Compute unified device architecture
CVM	Curvature velocity method
CWT	Continuous wavelet transform
D_*	Specific detectivity (Meaning in Chap. 1)
D_*	Dynamic a_* (Meaning in Chap. 14)
D/A	Donor-acceptor
DAC	Digital-to-analog converter
dB	Decibel
DBSCAN	Density-based spatial clustering of applications with noise
DC	Digital-analog cell
DCF	Decision function
DCT	Discrete cosine transform
DF	Decision function
DGPS	Differential global positioning system
DL	Deep learning
DLP	Dwa local planner
DOEP	Digital optoelectronic processor
DOF	Degree of freedom
DoG	Difference of gradient
DSSC	Dye-sensitized solar cells
DUT	Device under test
DWA	Dynamic window approach
DWT	Discrete wavelet transform
EB	Elastic bands
ECS	Environment cooling system
EDM	Electronic distance measurement
EGT	Exhaust gas temperature
EKF	Extended Kalman filter
EM	Equivalence model
EMR	Electromagnetic radiation
EMW	Electromagnetic waves
EO	Exterior orientation
EQ_CL	Equivalent continuous-logical
EQE	External quantum efficiency
ExG-ExR	Excess green minus excess red vegetation index
FCL	Flexible collision library
FDI	Fault detection and isolation
FET	Field-effect transistor
FIP	Focus-induced photoluminescence
FIR	Finite impulse response
FIT	Frame interline transfer
FMCW	Frequency-modulated continuous wave

FNS	Fundamental numerical scheme
FO	False object
FoV	Field of vision
FPGA	Field programmable gate array
FPS	Frames per second
FR	Flying robots
FT	Frame transfer
FVHT	Fovea centralis hierarchical trees
FW	Fixed window
FWHM	Full width at half maximum
FWT	Fast wavelet transform
FW-UAV	Fixed wings unmanned aerial vehicle
G	Gray
GaN nanowires	Gallium nitride nanowires
GIF	Graphics interchange format
GNSS	Global navigation satellite system
GPGPU	General purpose graphics processing unit
GPS	Global positioning system
GPU	Graphics processing unit
GRV	Gaussian random variable
GSD	Ground sampling distance
GSG	Ground signal ground
GVD	Generalized voronoi diagram
HAM	Hetero-associative memory
HD	High definition
HEVC	High efficiency video coding
HF	High frequency
HH	Horizontal histogram
HIL	Hardware-in-the-loop
HPC	High pressure compressor
HPT	High pressure turbine
HSV	Hue-saturation-value (color model)
HVS	Human visual system
Hz	Horizontal
IATS	Image-assisted total station
ICP	Integrated color pixel
ID	Identification
iDCT	Integer discrete cosine transform
IDE	Integrated development environment
IEEE	Institute of electrical and electronics engineers
IF	Informational field
IIR	Infinite impulse response
iLWT	Inverse LWT
IMU	Inertial measurement units
INS	Inertial navigation system

IO	Interior orientation
IoT	Internet of things
IP	Image processor
IPT	Image process technology
IR	Infrared
IRNSS	Indian regional navigation satellite system
iSPECK	Inverse SPECK
IT	Information technologies
IT SB RAS	Institute of Thermophysics Siberian branch of Russian academy of Science
JBIG	Joint bi-level image group
J_d	Dark current
JPEG	Joint photographic experts group
JPEG2000	Joint photographic experts group 2000
J_{ph}	Photocurrent
JSR	Just solidified region
k -d tree	k -dimensional tree
KF	Standard Kalman filter
K_I	Integral gain
K -nn	K -nearest neighbors
K_p	Proportional gain
ILSVRC2012	Large-scale visual recognition challenge 2012
LCL	Lossless compression limit
LCM	Lane curvature method
LDR	Linear dynamic range
LED	Light-emitting diode
LIDAR	Light detection and ranging
LIP	List of insignificant pixels
LIS	List of insignificant sets
LoG	Laplacian of Gaussian
LPC	Low-pressure compressor
LPF	Low-pass filter
LPT	High-pressure compressor
LSP	List of significant pixels
LWT	Lifting wavelet transform
MAAM	Multi-port AAM
MAD	Mean of Absolute Differences
MAE	Mean absolute error
MAP	Maximize the Posterior estimation
MAR	Mobile autonomous robots
MATLAB	A high performance computing environment provided by Math-Works
MAV	Micro aerial vehicle
MDPG	Maximum distance of plane ground
MEMS	Microelectromechanical systems

MEO	Medium earth orbit
MHAM	Multi-port hetero-associative memory
MIMO	Multi input and multi output
MIPI	Mobile industry processor interface
MIS	Minimally invasive surgery
MIT	Massachusetts Institute of Technology
MLA	Array of microlenses
MLE	Maximum likelihood estimator
MPEG	Moving picture experts group
MPixel	Mega pixel
MRS	Multi-robot systems
MSCA	M-estimator Sample Consensus
MSE	Mean squared error
MUTCD	Manual on uniform traffic control devices
MVS	Machine vision systems
N1	High rotor speed
N2	Low rotor speed
NCC	Normalized cross correlation
ND	Nearness diagram
NE	Neural element
NEP	Noise equivalent power
Neq	Normalized equivalence
Neqs	Neuron-equivalentors
NEU	North-east-up
NIR	Near infrared
NN	Neural network
NnEq	Normalized nonequivalence
NS	Navigation system
NSEqF	Normalized spatial equivalence function
PA	Candidate area
PE	Number of areas
PL	Number of line
PP	Number of pixels
OB	Object of binding
OC	Opposite cathetus
OD	Obstacle detection
OE-VLSI	Optoelectronic very large-scale integration
OFET	Organic field-effect transistor
OHP	Organohalide perovskite
OLED	Organic light-emitting diode
OPD	Organic photodiode
OPT	Organic phototransistor
OR	Object of binding
ORB	Oriented FAST and Rotated BRIEF
OSC	Organic semiconductor

PC	Personal computer
PCL	Point cloud library
PCX	Personal computer exchange
pdf	Probability density function
PI	Proportional-plus-integral
PID controller	Proportional-integral-derivative controller
PiPS	Planning in perception space
pixel	Picture element
PL	Positioning laser
PM	Propagation medium
PNG	Portable network graphics
ppi	Pixels per inch
PPR	Pulses per revolution
PRM	Probabilistic road map
PSNR	Peak signal-to-noise ratio
PT	Phototransistor
QR	Code-quick response matrix code
RADAR	Radio detection and ranging
RANSAC	Random sample consensus
RAR	Roshal archive file format
RCS	Radar cross section
ReLU	A rectified linear unit function
RF	Radio frequency
RFBR	Russian fund of basic research
RGB	Red, green, blue
RGB-D	Red, green, blue depth
RI	Reference image
RLE	Run length encoding
RM	Radiometric (Meaning in Chap. 16)
RM	Recursive median (Meaning in Chap. 23)
RMI	Radiometric imaging
RMS	Root mean square
RMSE	Root mean square error
RO	Relative orientation
ROI	Region of interest
ROIC	Read-out integrated circuitry
ROS	Robot operating system
ROV	Remotely operated vehicle
RRT	Rapidly exploring random tree
RTS	Robotic total station
RW-UAV	Rotary wings unmanned aerial vehicle
S/R	Storage and retrieval
SA	Scanning aperture
SAD	Sum of absolute differences
SAE	Society of automotive engineers

SD	Standard deviation
SD_NEF	Spatially dependent normalized equivalence function
SDPN	Sensors of different physical nature
SE(2)	Special Euclidean group for two-dimensional space
SE(3)	Special Euclidean group for three-dimensional space
SEM	Scanning electron microscope
SfM	Structure from motion
SGBM	Semi-global block matching
SHD	Sample and hold device
SHM	Structural health monitoring
SI	Source image
SI EM AM	Spatially invariant equivalence model associative memory
SIFT	Scale invariant feature transform
SLAM	Simultaneous localization and mapping
SLECNS	Self-learning equivalent-convolutional neural structure
SM	Simple median
SMC_ADC	Multi-channel sensory analog-to-digital converter
SMO	Sequential minimal optimization algorithm
SoC	System on a chip
SPAD	Single-photon avalanche diodes
SPECK	Set partitioned embedded block codec
SPIHT	Set partitioning in hierarchical tree
sRGB	Standard Red, Green, Blue color space
SS	Sighting surface
SSD	Sum of squared differences
SSIM	Structural similarity index
STP	Spanning tree protocol
SURF	Speeded-up robust feature
SVD	Singular value decomposition
SVM	Support vector machine
SWNT	Single walled nanotube
TC	Transfer characteristics
TCC	Turbine cooling casing
TCP/IP	Transmission control protocol/Internet protocol
TEB	Timed elastic bands
TensorFlow	An open source software library which can be used for the development of machine learning software such as neural networks
TFD	Transverse field detector
TFLOPS	Tera floating point operations per second
TLS	Total least squares algorithm
TM	Trademark
TNM	Technical navigation means
TO	Time-out
ToF	Time of flight
TPCA	Time-pulse-coded architecture

TSR	Traffic sign recognition
TVS	Technical vision system
UABC	Universidad Autonoma de Baja California
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
UKF	Unscented Kalman filter
UL	Unit load
ULE	Universal (multifunctional) logical element
USB	Universal serial bus
V	Vertical
vAOV	Vertical angle of view
Var	Variance
VFH	Vector field histogram
VH	Vertical histogram
VIRAT	DARPA video and image retrieval and analysis tool
VMO	Vector or matrix organization
VNA	Vector network analyzer
VO	Visual odometry
VPH	Vector polar histogram
VSLAM	Visual simultaneous localization and mapping
WAAS	Wide area aerial surveillance
WAMI	Wide area motion imagery
WAPS	Wide-area persistent surveillance
WebP	Webp
WF	Fuel flow
WFOV	Wide field of view
WRM	Weighted recursive median
WT	Wavelet transform
$Y' C_B C_R$	Luma Chrominance color space
ZIP	.ZIP file format

Part I
Image and Signal Sensors

Chapter 1

Image and Signal Sensors for Computing and Machine Vision: Developments to Meet Future Needs



Ross D. Jansen-van Vuuren, Ali Shahnewaz, and Ajay K. Pandey

Acronyms

ADC	Analogue-to-digital convertor
ASIC	Application-specific integrated circuit
CCD	Charge-coupled device
CFA	Colour filter array
CIS	CMOS image sensors
CMOS	Complementary metal-oxide semiconductor
CQDs	Colloidal quantum dots
D/A	Donor–acceptor
D^*	Specific detectivity
EQE	External quantum efficiency
FET	Field-effect transistor
FIT	Frame interline transfer
FT	Frame transfer
FWHM	Full width at half maximum
ICP	Integrated colour pixel
IoTs	Internet of things
IR	Infrared
J_d	Dark current
J_{ph}	Photocurrent

R. D. Jansen-van Vuuren (✉)
Department of Chemistry, Queen’s University, Kingston, ON, Canada
e-mail: rdjv@queensu.ca

A. Shahnewaz · A. K. Pandey
School of Electrical Engineering and Computer Science, Queensland University of Technology,
Brisbane, QLD, Australia
e-mail: shahnewaz.ali@hdr.qut.edu.au; a2.pandey@qut.edu.au

LDR	Linear dynamic range
MVS	Machine vision systems
NEP	Noise-equivalent power
OFET	Organic field-effect transistor
OHP	Organohalide perovskite
OLED	Organic light-emitting diode
OPD	Organic photodiode
OPT	Organic phototransistor
OSC	Organic semiconductor
PT	Phototransistor
RGB	Red green blue (referring to a colour filter system)
ROIC	Read-out integrated circuitry
TFD	Transverse field detector
ToF	Time of flight

1.1 Introduction

1.1.1 *Image Sensing in Machine Vision Systems*

Digital cameras offer many advantages over conventional photo technologies, including the elimination of film processing, the ease of editing and affordability. Evidence for their increasing popularity worldwide can be seen in the resultant consumer success. The market for image sensors has experienced major growth over recent years with the value predicted to reach USD 23.97 billion by 2023 [1, 2]. This increase is largely due to digital still and video cameras, but also includes the expansion of digital imaging to cellular phones, laptop and personal computers (e.g. Internet-based video conferencing), security and surveillance, and the automotive, medical and entertainment industries. Digital cameras are also used extensively for image capture in machine vision systems (MVS), which rely upon object recognition and image analysis/indexing to extract data which is then used to control a process or activity. The applications of MVS are broad and range from automated industrial applications such as inspection and quality evaluation of products [3–5] to robotic guidance and control [6], autonomous vehicles [7–9], precision viticulture [10], picking and sorting fruit and vegetables [11] and colorimetric sorting systems [12].

Conventional image sensors are generally considered to be sufficient for consumer digital photography but are limited when meeting the level of imaging required for MVS applications which demand accurate and rapid colour image capture [6, 13], often in scenes of uncontrolled lighting with a large dynamic light range [14]. Furthermore, given the conditions under which imaging in rapidly advanced applications (e.g. self-driving cars, military applications, robotics) occurs, research is underway to find ways to develop photodetection systems which have the requisite size, lightness, compatibility with flexible and miniaturized substrates

and durability, preferably with a reduced cost. In order to try and meet these requirements, modifications to the sensing systems can be made including the use of different photodetector materials and/or image processing technologies, changes to the design and arrangement of the colour separation systems, altering the image sensor architectures or the individual pixel sensor arrangements (typically passive or active) or integration of ‘smart functions’ onto the chips of image sensors.

This chapter seeks to review the limitations of current MVS and the research being carried out to address these. The focus is largely upon applications which depend on colour image capture for object recognition and image indexing. MVS depending on colour recognition need to satisfy a vastly more complex requirement since ‘color images include not only the brightness but also the color information, such as hue and saturation’ [15, 16]. From the literature, the default approaches to improving colour recognition in MVS involve either modifying the image processing algorithms (these could include colour segmentation techniques) [15, 17] or exerting more control over the environmental conditions under which colour sensing takes place [3, 7, 18]. A key objective of this chapter is to flesh out two proposed alternative approaches, namely changes that can be made to the *architecture* of the image sensor and the *photosensor material* within the image sensor.

1.1.2 Image Capture by Digital Cameras

Firstly, consider the general set of operations carried out by a camera in capturing an image using an image sensor. The basic operations carried out by all digital cameras, regardless of their specific function and application, are essentially the same and consist of five separate steps [19]. These include: (1) photon collection, usually via a lens, which entails focusing the light before transmitting it through the optical system; (2) separation of the incoming photons by energy/wavelength (colour discrimination)—typically carried out using colour filter systems, for example Bayer colour filter array [20]; (3) (a) formation of photocurrent and (b) readout of the resultant signal (performed by the image sensor); (4) interpretation and processing of the data—now in digital form—in order to reproduce the colour image and (5) colour management and image compression processes as carried out by the microprocessor prior to data storage and export.

The image sensor plays the vital role of capturing the image, and the means by which this fundamentally occurs can be summarized in four steps [21]. (1) The absorption of photons by the photoactive material which constitutes the pixels, generating electron–hole pairs. (2) The electrons and holes are driven by means of an external electric field towards opposite electrodes, where they are extracted and give rise to the signal charge, which is collected and accumulated at each pixel. (3) The accumulated charge is then read out from each pixel in the two-dimensional array. Various means by which this occurs result in a range of different architectures giving rise to the range of image sensors on the current market, such as charge-coupled device (CCD) sensors, complementary metal-oxide semiconductor

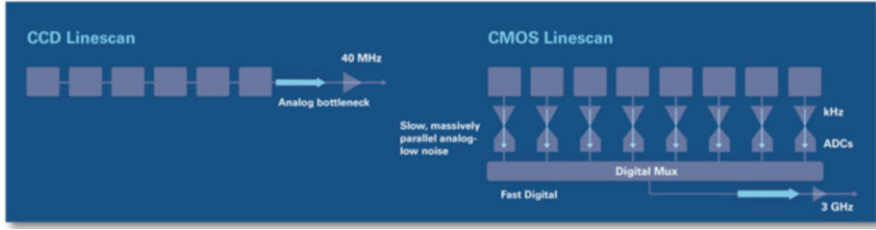


Fig. 1.1 General operation mechanism of a CCD versus that of a CMOS image sensor (image courtesy of [23])

(CMOS) sensors, MOS $x - y$ using addressed devices and frame transfer (FT) devices. (4) Finally, the charges are detected, which occurs in a manner that is essentially independent of the type of sensor. Although the different image sensor architectures commence at the same point, namely the transduction of photons to electrons, they differ in *how the charge is collected*. The most common image sensors have CCDs and CMOS architectures (Fig. 1.1). In general, CCDs, developed in the late 1960s, work by transporting the charge (generated through light absorption) across the chip to one corner of the photodiode array, before an analogue-to-digital convertor (ADC) transforms this into a digital signal, whereas in CMOS image sensors (CIS) (developed in the 1990s), photogenerated charge is collected at each pixel, before being amplified and transferred using traditional wiring (Fig. 1.1) [22].

Thus, CIS offer the following advantages over CCDs: ‘ease of system integration, low power consumption, and freedom of device architecture’ [22]. In addition, considering that machine vision systems require high speed with low noise during imaging, CMOS image sensors ‘can be designed to have much lower noise than high speed CCDs’, as shown in Fig. 1.1 [23]. There are multiple ways in which CIS can be configured—the two main approaches differ in the position of the light-receiving photodiodes: in front-illuminated CIS, the incoming light needs to pass through the colour filters and metal wiring before reaching the photodiode, whereas in back-illuminated CIS, the light reaches the photodiode more efficiently [22, 24]. Initially, due to their simple pixel layout, CCDs enabled more pixels per unit area and, therefore, higher pixel count and resolution. This was a major reason for their contribution towards the development and rise in popularity of digital still cameras [22, 25]. However, when CMOS image sensors made an appearance on the market, they rapidly gained popularity since the incorporation of ‘in-pixel transistors’ supported rapid image capture with low power consumption [22, 26]. The history of the development of CCDs and CMOS image sensors has been covered in detail elsewhere [27–29].

Although a major difference between the structure and operation of the various sensors lies in the method used to read out the signal produced by the photodetectors, it is the *type of photodetector material* and *means of colour recognition* within the camera that ultimately defines the quality of the image, since this has the largest influence over the spectral sensitivity and resolution of the sensor. There are several

approaches to capture colour images using conventional (broadband) photosensing materials. These can generally be classified into two major groupings which include (1) sensors which make use of an auxiliary structure that does not constitute the active layer of the pixel, such as a colour filter on top of the sensor cells, and (2) those in which the colour separation system is integrated within the imaging array (see Fig. 1.2). (1)(a) Although there are several arrangements of filters whose selection depends upon the application, a common system employs the Bayer filter [20], which consists of a mosaic of red (R), green (G) and blue (B) filters such that there are twice as many Gs as there are R and B to simulate the human visual system. (1)(b) The second method involves taking three sequential exposures, each with a different optical filter (RGB) mounted in a colour wheel [30], before combining the three separate images to form the final picture (Fig. 1.2b). (1)(c) The third approach involves the use of a beam-splitter, classically a trichroic prism assembly (Fig. 1.2c), which separates the light into its R, G and B components before these are focused onto three discrete image sensors ('3-CCD' or '3-CMOS'). Although considered somewhat superior in image quality and resolution, 3-CCD cameras are generally more expensive than single-sensors and the potential for miniaturization of cameras is somewhat limited [30]. (1)(d) An emerging technology called the integrated colour pixel (ICP) involves replacement of the colour filter array (CFA) with an array of metal strips in a specific pattern which enables colour separation during image capture. The patterned metal layers are placed within each pixel such that they control the transmission of light to the photodetector within the pixel [31].

In the second group, there are two approaches that can be taken. The first of these involves the direct absorption of red, green and blue lights at each location by stacking the colour pixels in a three-layer arrangement [32], as shown in Fig. 1.2. (2)(e). For example, this system is applied within the Foveon X3 direct image sensor [33, 34] and is similar in many respects to the layers of chemical emulsion comprising colour film. Foveon X3 image sensors have three layers of pixels, and each layer is embedded in silicon, taking advantage of the fact that red, green and blue lights penetrate silicon to different depths, therefore enabling an image sensor that captures full colour at every point in the captured image. Although stacked image sensors are able to increase the fill factor of the sensor surface area, as separate receptors are no longer required for each colour, the spectral sensitivity in these image sensing devices and the resultant colour reproducibility is insufficient to meet the demands of modern applications and cross-talk between layers presents a major device challenge [35, 36]. Finally, (2)(f) shows a method relying on the application of an electric field across the device, enabling the generation of carriers at varying but specific depths, subsequent to their collection [37, 38].

1.1.3 Performance Metrics of Image Sensor Photodiodes

Having covered the basic structure of modern image sensors and the various systems by which colour separation can be carried out, the figures of merit of photodetectors

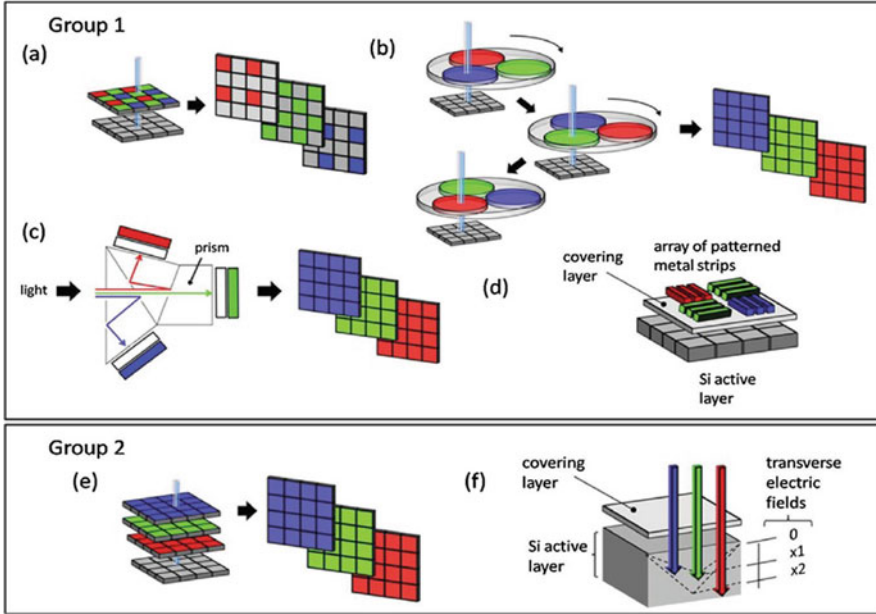


Fig. 1.2 The two main approaches for colour separation within broadband inorganic semiconductor photosensors. Group 1: (a) Bayer filter mosaic; (b) sequential triple exposure with R, G and B filters; (c) prism separation system and three sensor arrays (3MOS or 3CCD); (d) the Integrated Colour Pixel (ICP). Group 2—image sensors who achieve colour separation through an internal mechanism: (e) Foveon X3 image sensor; (f) Transverse Field Detector (TFD). Used with permission from Jansen-van Vuuren RD, Armin A, Pandey AK, Burn PL and Meredith PM (2016) Organic Photodiodes: The Future of Full Color Detection and Image Sensing. *Advanced Materials*, 28, 4766–4802. Copyright (2018) American Chemical Society. Taken from [39], Figure 2

(i.e. performance metrics) now need to be defined (Table 1.1), as these will be referred to in the text that follows. These metrics apply in general to the photodiodes within the image sensor, regardless of the material from which they are fabricated. Inorganic semiconductors are traditionally used, but these have limitations surrounding their use in MVS.

1.2 Limitations of Current Inorganic-Based Imaging Systems

Current image sensors in MVS are based on traditional silicon- or germanium-based technologies where silicon, Si (or germanium, Ge), is the material used as the photosensing material within the image sensor. III–V compounds such as InSb, GaN, AlN and InN are also used when a different bandgap is required. Si is the most commonly used semiconductor in optoelectronic devices due to its prevalence and the well-established technology enabling its integration within devices on large scales

Table 1.1 Performance metrics for photodetectors

Metric	Unit	Basic definition	Influenced by
Gain (G)	No unit	Number of photogenerated carriers circulating within the system divided by the number of incident photons	Active layer optical constants of material; charge transport; device optics
Photocurrent (J_{ph})	Amps (A)	Current flowing in a device as a result of illumination	Absorption coefficient of material; mobility, trapping and doping; electrode work functions
Dark current (J_d)	Amps (A) or nA/cm ²	Current flowing in a device without illumination	Mobility, trapping and doping; electrode work functions
Responsivity (R)	Watt/Amp (W A ⁻¹)	Ratio of photocurrent to the incident optical power (gives some idea of how efficiently the detector responds to an optical signal)	Absorption coefficient of material; mobility, trapping and doping; electrode work functions
External quantum efficiency (EQE)	%	Number of photogenerated carriers collected divided by the number of incident photons	Active layer optical constants of material; charge transport; device optics
Linear dynamic range (LDR)	Decibels (dB)	Range of incident light intensity to which the photodetector responds linearly	J_d , carrier mobilities, bimolecular recombination rate, device thickness
Noise-equivalent power (NEP)	Watt/Hertz (W Hz ^{-1/2})	Minimum light power detected—essentially a measure of the device sensitivity	Morphology; noise current, responsivity
Specific detectivity (D^*)	Jones (J)	The lower limit of light intensity detection (corresponds to NEP normalized to the device area and the electrical bandwidth of the noise measurement). D^* can be calculated as follows: $D^* = \frac{\sqrt{A}}{NEP}$, where A = the device area	Morphology; noise current, responsivity and device area
Spectral selectivity	Nanometres (nm)	Full width at half maximum (FWHM) of the spectral response	Properties of the photoactive chromophore (optical gap engineering); light matter interaction within the cavity (device)
Response speed/bandwidth (BW)	Hertz (Hz)	Speed of response, also known as the 'operation frequency' or 'operation bandwidth' of the signals detected	Charge carrier mobilities; charge trapping defects; range of light intensity and modulation frequency of the photo-response; thickness of material layer
Flexibility (mechanical strain)	Angle of tolerance (°)	Angle through which device can be bent without reducing J_{ph} reproducibly	Thickness, morphology

[40, 41]. Photodetectors used in current colour image-sensing applications are made from hydrogenated amorphous (a -Si:H) [42] or crystalline (c-Si) silicon [28]. This is typically deposited on top of the application-specific integrated circuit (ASIC), which is then responsible for the readout and processing of the photo signals. Photodiodes convert light into electrical signals by optical absorption resulting in the formation of electron–hole pairs which subsequently form separated charge carriers across a p - n junction. The charge separation occurs rapidly and without the need for an additional driving force, resulting in large charge mobilities (greater than $10^2 \text{ cm}^2\text{Vs}^{-1}$) [43] and nanosecond transient times, which has resulted in very high internal quantum efficiencies and sensitivities [44]. However, the significant problems with silicon (both amorphous and crystalline) with respect to photodetection for MVS are outlined as follows.

1.2.1 Weak Light Absorption

Si absorbs light relatively weakly over the visible spectrum [45], particularly in the blue region (400–460 nm) [32, 46]. While GaN detectors demonstrate superior UV light detection to Si, their practical use is still limited by cost and the need for complex architectures to achieve high detectivities [47]. In extremely low lighting, conventional PDs require low temperatures to reduce the J_d [48].

1.2.2 Low Dynamic Range

Firstly, image sensors fabricated with silicon photosensors are unable to cope with a high dynamic range (DR) of lighting. This can be experienced when trying to capture an image of a scene consisting of a very bright component as well as an object in complete shadow, resulting in the formation of images saturated either by bright white or dark black. This can be understood by considering the range in which the photocurrent generated by the photodiode is linearly dependent on the incident light power, with a tolerance of $\pm 1\%$. This range is called the linear dynamic range (LDR). Outside of the LDR, the device saturates completely at any incident light power level and the photodiode is said to be non-linear. The LDR depends on the wavelengths of light absorbed, the inherent properties of the photodiode (carrier mobility, device thickness and the noise current generated), the reverse bias applied and the resistance of the circuit in which the photodiode current is generated and collected. Silicon photodiodes have DRs of 100–120 dB [49], which corresponds to a J_{ph}/J_d of ~ 6 orders of magnitude. Although this is discussed in more detail in subsequent sections, the highest value reported for organic photodetectors is 160 dB [50] and for metal halide perovskite photodetectors is 230 dB [51].

1.2.3 Incompatibility with Complicated Processing and Fabrication on Flexible, Miniaturized Devices

For many MVS applications, incompatibility of traditional inorganic semiconductors with read-out integrated circuitry (ROIC) presents a major obstacle to realizing compactness (whilst maintaining high detectivity and sensitivity) of light sensing devices [52]. Furthermore, since conventional inorganic semiconductors absorb a wide range of wavelengths, they require the use of colour filter arrays (and wavelength cut-off filters for colour sensing applications), thus complicating the design of such devices. Photodetectors consisting of c-Si, Si/Ge heterojunctions or III–V semiconductor alloys (e.g. InGaAs) are usually fabricated on rigid substrates, which precludes their applications in novel device concepts such as stretchable devices and bendable cameras [53]. The ability to conform to the various shapes of surfaces could simplify optical systems and enable the integration of photodiodes into miniaturized devices and ground robots [54].

1.2.4 Inability to Cope with Illuminant Variation

In general, MVS face a number of challenges when used in outdoor environments due to unpredictable and uncontrollable changes in the illumination [55–57]. The core reason for the inability of systems to cope under variable illuminance can be attributed to deficiencies in the photodetector portion of the image sensors. Silicon photodetectors are panchromatic and therefore unable to discriminate between photons having different wavelengths, relying on colour filters or depth-dependent absorption to form colour images [58], which results in images whose colour characteristics deviate from reality [7, 59, 60]. The need for colour filters additionally complicates the architecture and fabrication of imaging devices [61]. Humans possess an in-built capability called ‘colour constancy’ that enables the true perception of the colour of an object, regardless of illuminant, over a reasonable range of wavelengths [7]. Much research has focused on the development of image processing software able to compensate for this limitation by attempting to estimate the true colours of objects from the captured light through the use of algorithms. Indeed, there are other physical approaches to coping with illuminant variance, for example applying a digital filter to the output of the photosensor [62]; however, this text focuses on the use of a different image sensor material to achieve illuminant invariant image production.

1.2.5 Low Bandgap

Silicon also tends to have a smaller bandgap than required for visible detection, and therefore, the photodetectors require infrared (IR) filters in order to avoid unwanted IR sensitivity which contributes to excess noise [43, 63]. Group III–V compounds have different bandgaps, and therefore, wafers fabricated from these compounds offer variable options; however, these face most of the same issues already highlighted—lack of flexibility, complicated fabrication onto miniature or curved devices, etc.

1.2.6 Crosstalk

Despite the high mobilities and carrier lifetimes of silicon, this can also be regarded as disadvantageous as it causes crosstalk and distortion of the optical signals between neighbouring pixels, placing a high demand on the pixellation procedures, which are already delicately balanced between resolution and sensitivity [43, 64]. Pixel crosstalk can be attributed to leakage of photocurrent and/or the deflection and scattering of photons by adjacent pixels; both the effects (electrical and optical) contribute to reduction of the resolution of the colour and resolution of the final image [39].

1.3 Overcoming Limitations of Conventional Imaging Systems Using Alternative Photosensing Materials

The development of alternative semiconductor materials to fulfil some of the shortcomings presented by traditional inorganic semiconductors in photodetection forms a research field in its own right. The materials highlighted in the remainder of the chapter include semiconductors that can be processed under low-temperature conditions through ‘wet chemistry’ techniques, which include 3D printing, spray coating, spin coating, inkjet printing and doctor blading [52]. These approaches open up the possibility for large-area deposition and compatibility with substrates of different shapes and sizes and with flexible surfaces [53]. Furthermore, many of these alternative semiconductors have demonstrated similar and even superior performance metrics compared with their inorganic counterparts. The two major classes of materials that have been researched include (1) organic semiconductors (OSCs), and, more recently, (2) organohalide perovskites (OHPs). Although colloidal quantum dots (CQDs) have also been given research attention, these are yet to make meaningful gains in the production of image sensors. Therefore, CQDs will not be explored in this chapter. Organic semiconductors have been studied in greatest detail and so will be dealt with followed by OHPs.

1.3.1 Organic Photodetectors in Image Sensing

Organic semiconductors have already replaced inorganic materials in a range of applications available on the market, for example photovoltaic cells (NanoFlex Power Corporation, Infinitypv.com), light-emitting diodes LEDs (Sony OLED TVs, Panasonic OLED TVs, LG OLED TVs) and thin-film transistors (NeuDrive). The main reasons include the prospect of cheaper processing methods which involve solution-deposition or inkjet printing, the fact that they can be lightweight, thin and flexible, and the existence of a wide selection of organic materials which allows for tuning of the physical and optoelectronic properties. As a result, OPDs are in fact a ‘disruptive technology’ for MVS and large-area digital imagers as they enable ‘lightweight, flexible, mechanically robust, and even conformable imagers’ [65].

The first organic photodetector (OPD) was demonstrated in 1981 using dyes [66], before Yu and colleagues demonstrated a bulk heterojunction OPD with a sensitivity greater than that of UV-enhanced commercial Si-photodiodes in 1994 [67]. OPDs have since been developed with figures of merit that are comparable or even superior to traditional inorganic photodiodes [39, 49, 52, 68–70].

Compared with the three-dimensional networks of covalent bonds found in inorganic semiconductor structures such as silicon wafers, active films of organic semiconductors possess covalent intramolecular bonds but weak intermolecular van der Waals forces. This difference in the bonding systems results in the localization of the electronic wave function to individual molecules (instead of extending over the entire structure), which affects the separation of the electron–hole pairs in organic semiconductors, and their electronic bandwidth [71]. Optical excitation of organic semiconductors results in the formation of bound electron–hole pairs (called ‘excitons’) which can only be separated efficiently at a heterojunction of two materials with differing electron affinities. The energy difference between the electron affinities needs to be around 0.4–0.5 eV [72] to overcome the exciton-binding energy [73]. The separated holes and electrons then travel through the electron donor (D) and electron acceptor (A) materials, respectively, where they are extracted to the electrodes. For the process to work, the excitons need to diffuse to the D/A interface (the distance travelled by excitons is referred to as the exciton diffusion length and is typically 5–10 nm) [74]. During this process, there is a possibility of radiative or non-radiative recombination of the electron–hole pairs. Hence, exciton diffusion and separation must proceed more rapidly than the recombination processes. These steps, illustrated in Fig. 1.3c for the compounds shown in Fig. 1.3a, are fundamental to the conversion of light into electrical energy within excitonic (organic semiconductor-based) photodiodes.

Figure 1.3b illustrates the origin of J_d within photodetectors. Organic semiconductors may therefore play a part in reducing inter-pixel crosstalk since the excitonic movement from one pixel to the next is typically low and easily controlled.

Modifications to the chemical structures of the D or A compounds influence their packing in a film (morphology) and the electronic and optical energy levels, which in turn can lead to altered behaviour at the BHJ interfaces and different

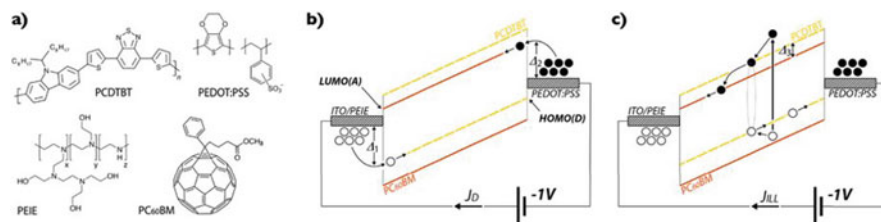


Fig. 1.3 (a) Chemical structures of PCDTBT (electron donor, D), PEDOT:PSS (top electrode), PEIE (ITO modifier) and PC60BM (electron acceptor, A); (b) Working principle of the photodetector in dark illustrating the origin of dark current and (c) under illumination showing the photovoltaic effect. Filled circles are electrons, empty circles represent holes (taken from [85], Fig. 2, used through a Creative Commons Attribution 4.0 International License)

light sensitivities [75]. Thus, OPDs can be tuned, depending on the application requirements, enabling OPDs to overcome the low bandgap problem that is prevalent in inorganic semiconductors.

OPDs can be either ‘broadband’ or ‘narrowband’, depending on whether the semiconductor material absorbs light over a broad spectrum of wavelengths or a more narrow spectrum (typically absorbing one colour from the spectrum). Broadband OPDs can be incorporated into colour sensing systems in the same way as inorganic photosensors—using filters or stacking (or any of the other approaches shown in Fig. 1.2) [76, 77], with the same two major consequences previously described, namely (1) complicated device fabrication and (2) low colour accuracy under varying illuminant conditions. Deckman et al. (2018) report how a combination of a broadband OPD and broadband filters ‘can successfully detect and reconstruct colors in the RGB system, with an average accuracy of 98.5%’ [78]. Conversely, narrowband absorbing organic semiconductors enable the construction of filter-free photodetectors [79–81]. The use of four narrowband absorbers (each having an FWHM <100 nm) was found to be sufficient for achieving colour constancy for applications involving object recognition in MVS [82]. Other approaches to achieving narrowband and filter-free absorption involving the manipulation of the internal quantum efficiency of thick (μm) OPDs have been developed [50, 83, 84]. Thus, it is in this manner that OPDs are able to overcome a major limitation faced by traditional inorganic semiconductors—unable to cope with scenes of uncontrolled and variable illumination (e.g. in outdoor environments).

Although the concept of using organic materials as photodetectors is still relatively new, significant progress has been made, with organic semiconductors having superior photodetectivities, for example 1.03×10^{14} J at 735 nm under a positive +1 bias [86], higher linear dynamic responses over a wide spectral range, for example 160 dB for a broadband OPD [86] and 160 dB for a narrowband OPD [50], and similar J_{ds} (dark currents), for example 1.2×10^{-10} A/cm² [87], when compared with conventional inorganic photodiodes [88–92]. Thus, OPDs are able to overcome the weak light absorption and low dynamic range posed by traditional photodetectors.

Furthermore, the electro-optical properties of organic materials can be fine-tuned through simple modifications made to the chemical structure [93].

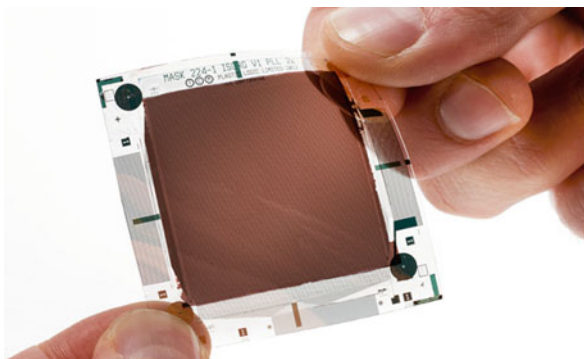
Image sensors have been fabricated with organic semiconductors as the photoactive layer [81, 94–98], demonstrating their applicability and feasibility in imaging and colour sensing. Samsung has reported the fabrication of image sensors with colour-selective OPDs [81, 99–101]. Panasonic has reportedly developed organic photosensing technologies, incorporating OPDs into an AK-SHB 810 model camera [102]. ISORG (based in Grenoble, France) has pioneered large-area OPDs and image sensors, collaborating with Plastic Logic in 2013 to co-develop the first OPD image sensor on plastic (Fig. 1.4) [103]. ISORG recently announced a substantial sum to be invested in developing value-added applications, ‘primarily in personal electronic devices such as smartphones, wearables, tablets and laptops, biometrics for homeland security and medical imaging’ [104].

Finally, OPDs have also been shown to demonstrate superior temperature stabilities compared with c-Si photodiodes [68]. This is a significant factor in the design of MVS for the applications in which the camera equipment is exposed to variable environmental conditions.

1.3.1.1 OPDs Beyond Photodetection

Current machine vision systems exploit CMOS technology for imaging, taking measurements, locating, identifying, inspecting or navigation. Emerging applications make use of CMOS image sensors in vision-based aerial imaging and navigation. In these latter applications, the camera technology has to be lightweight and low power-consuming to ensure economic viability and to be able to last over long flight durations. This would need development of new materials beyond discrete devices to full-fledged imaging systems. The previous section demonstrates that there is a strong potential for organic semiconductors in further simplifying the 2D layout of current camera technology. In this section, we present an outlook for organic and other family of advanced materials for their potential application

Fig. 1.4 Organic image sensor on a flexible organic thin-film transistor backplane (image used with permission, taken from [103])



in a number of new areas with a particular focus on how the research on this emerging technology should be aimed at integrating into advanced 3D imaging systems. The intrinsic advantages of OPD technology has not been realised, and features associated with spectral selectivity, tenability of multi-colour detection at relatively small form factors, mechanical flexibility and fabrication advantages are all poised to add intelligence at the pixel level. The ability to customise the sensor response without the need of complex fabrication protocols means OPD technology is suitable for combining shared intelligence at the hardware and software levels. These are some of the important attributes in achieving the next generation of smart, intelligent, light-weight and low-power demanding imaging systems for robotics and IoTs.

The cost of electronics has significantly reduced by integration of the emitter and receiver systems in integrated circuits. The ability to print light emitting diodes and organic photodetectors side by side would further allow denser integration of light signals and their detection. The recent emergence of bi- and multifunctional performance of organic optoelectronic devices offers great promise in simplifying the fabrication and integration of emitter and receiver functions by reducing the need of complex interconnects that reduce the effective form factors [105–109].

Conventional vision technology projects 3D world information into a 2D plane with no depth information. There is a growing demand for robust imaging technologies that can extend 2D imaging to 3D view of the scene in real time [110, 111]. For example, a robot with 3D vision can do more than detect the orientation of an object—it can actually *recognize* the object. This allows for intelligent, real-time decision-making and can be used to add intelligence to robot to learn quickly and be aware of the environment it is placed in [112, 113]. Microsoft has filed a patent application for a single handheld device that can detect material properties such as reflectivity, true colour and other properties of surfaces in a real-world environment [114]. The device exploits known relationships between lighting conditions, surface normals, true colour and image intensity.

Depth information improves system reliability and efficiency, for example an autonomous vehicle needs to perceive the objects present in the 3D scene from its sensors in order to plan its motion safely. It is important to highlight that the current state of the art imaging technologies still lack the ability to deal with a number of factors such as objects that have low textures or objects that are soft and deformable. Lighting conditions play an important role in the ability to image a scene, and the ability to design detectors that are selective to only a part of spectrum has great potential in reducing the artefacts introduced by ambient lighting. In this regard, spectrum or colour-selective OPDs have huge potential in improving image capture with constrained (indoor) or unconstrained (outdoor) environments. Therefore, OPDs have desirable attributes that can meet the application-specific requirements for diverse imaging environments including autonomous systems, mining, medical, social, aerial and marine robotics.

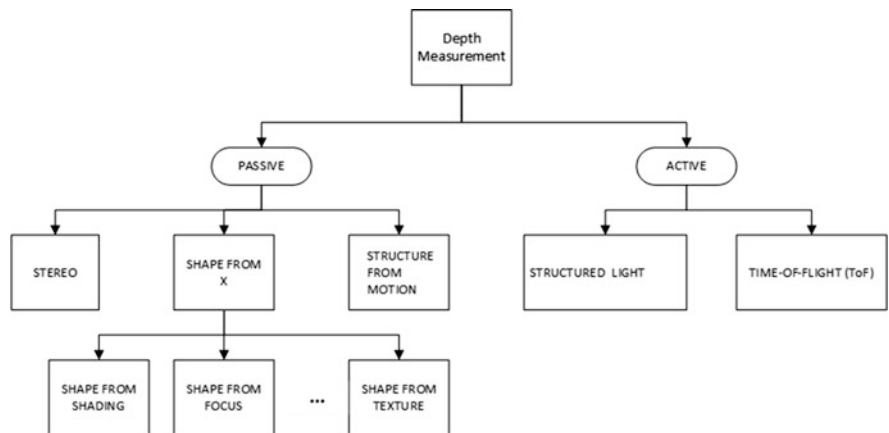


Fig. 1.5 Classification of depth measurement technology

Current depth sensor technologies can be classified into two main classes, as illustrated in Fig. 1.5. Passive estimation technology relies on machine learning algorithms and mathematical approaches which are used to extrapolate depth information from 2D image or images. The other class is active depth estimation technology, which relies on sensor technology or depth sensing devices to estimate distance. One aim of this section is to provide a path for OPDs and associated technologies to design devices that can measure or estimate depth using either a passive or an active approach.

Both passive and active depth measurement technology can provide depth perceptions of a scene. Active measurement technology is principally limited to image array size, therefore produces low-resolution images. Passive imaging technology uses natural or ambient illumination to capture scene. Most of the passive image sensors are based on charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS).

OPDs for Depth Measurement Using Stereo Vision

Human vision is the most sophisticated and powerful vision solution to observe the environment and extract location information. Akin to the human visual system, robotic stereo vision forms a reliable depth perception technique for successful navigation of robots in unknown and unstructured environments [115]. The stereo vision technique requires two cameras to observe a scene from different locations and in turn produces different image locations of the objects. The disparity and baseline of the system are used for distance estimation and three-dimensional (3D) reconstruction of the scene. The simplest way to gain depth using OPDs is to fabricate

a set of OPD array that are separated by a known distance called the ‘baseline’. There are no stereo cameras based on OPDs yet, but these can be readily fabricated to infer depth from 2D images. Computer vision algorithms are used to reconstruct depth from single or multiple images. Single-view 3D reconstruction methodology uses only one image. On the other side, multi-view 3D construction considers two or more images to reconstruct depth information. When two images are used, the system is known as a binocular stereo vision system, and probably it is the most widely focused research area of computer vision.

Stereo matching is the core technique of the stereo vision. Stereo matching is the process that matches each pixel from reference image to target and perceives the depth of each pixel. An intensive comparison takes place to find the corresponding pixel on the target image. Pre-configuration and pre-processing always take place before the actual stereo matching. In the stereo vision system, the reference and the target camera capture same scene point at the same time with a slightly different viewpoint. Stereo vision algorithms are based on this hypothesis. Therefore, the term synchronization is always used to convey the sense that the image acquisition system captures the same scene point at the same time with no time lag. When the object is in motion, this precondition plays a pivotal role to reduce reconstruction noises.

Calibration is the process that reduces image distortions. Stereo rectification is a transformation process that aligned two images into the same plane, so that same horizontal line becomes parallel to both the camera centres. Depth is calculated by finding the disparity in a pair of images. Disparity refers to the distance between two corresponding points in the left and right images of a stereo pair. It is inversely proportional to the depth and vice versa. In a stereo vision system, the relationship between depth and disparity can be expressed by the following equation:

$$d = b * f/z \quad (1.1)$$

where b is the baseline, f is the focal length, z stands for depth and disparity is expressed by the letter d . The basic idea of the disparity calculation is to match each pixel from the left image to the right image. In some circumstances, it may happen that some of the parts of a scene may not be visible through one or two cameras. This part of a scene is sometimes referred as a missing part. When this match process ends up, the difference of the pixel position in right image with respect to left image is known as disparity. Depth is estimated from the disparity by using the geometric principle of triangulation.

OPDs for Active 3D Imaging

In addition to an imaging array of photodetectors, the active 3D imaging system consists of a light source known as a projector. The aim of the projector is to emit signals. Received reflected signals are analysed to construct the 3D structure of the surrounding environments. Most commonly emitted signals are from a laser light source, an ultrasound signal or near infrared light. Many terms are used to describe

3D active imaging technology such as rangefinder, range imaging and 3D scanner. Several methods are used to measure the distance, but probably the most practiced principles are time of flight, triangulation and phase shift. This section provides a brief introduction of these three principles that OLED and OPD technology can use in inferring depth information from a scene. Dense depth map with less ambiguity and minimum depth error are the most reported advantages of active 3D imaging technology. However, the resolution of the depth map is limited. Miniaturized, high-resolution and low-power active depth sensors have a potential demand in various fields like medical and aerial robotics.

Among other systems, time-of-flight (ToF) systems measure the distance from the scanner to surface points through the measure of the time employed by the radiation to reach the object and come back to the scanner. This technique is very similar to the mobility measurement in organic semiconductors but albeit used here for imaging the real-world objects using a set of OPD arrays.

In this section, we focus on the time-of-flight (ToF) principle for OPDs. The basic idea of the active sensing technology is to emit photons as signal. When a compatible OLED projector emits the signal, then the clocking system inside the OPD-based imaging system can be set to start counting. This approach is known as direct ToF. If the object exists within the range of the imaging system, then it reflects a potential amount of signal to the camera. When the OPD receiver receives this signal, it then computes round trip time and from the basic principle of the light or electromagnetic source, the distance of the object from the camera can be estimated, using the following relationship:

$$d = \delta T * \text{light} \quad (1.2)$$

Within a defined range, ToF provides high-quality depth maps. The precise clock is the challenging part of this approach, and OPD systems would be limited by intrinsic mobility of organic materials used in the fabrication of such detectors. For example, when an object is placed very near to camera, for example in millimetre distance, it is challenging to design a clock that can measure a time gap in nanoseconds. OLEDs and OPDs based on high-mobility polymers with very high sensitivities should therefore be developed to meet the needs of active depth measurements. However, to keep aside the high-precise clock, the transmitter or projector could use a modulated signal. This approach is known as indirect time of flight. The transmitter could contain a signal emitter array of OLEDs to generate a modulated signal. Different kind of modulations is used such as sine, square, etc. The received signal is compared to the original signal. Different signal characteristics such as signal phase can be used to probe distance and resulting phase difference could be used to measure time and distance. It is a continuous process and more suitable for OPD and organic optoelectronics. The depth sensing technologies described here is equally applicable to organohalide perovskite or similar materials.

1.3.2 Metal Halide Perovskite (MHP)/Organohalide Perovskite (OHP) Photodetectors

MHPs and OHPs are compounds with a crystalline structure of the form ABX_3 , where A and B represent cations of different sizes and X is an anion, typically a halide ion. In perovskites used to fabricate optoelectronic devices, A can represent an organic cation (e.g. methylammonium, CH_3NH_3^+) in the case of OHPs or an inorganic cation (e.g. Cs^+) in the case of MHPs, B is an inorganic cation (usually Pb^{2+} or Sn^{2+}), and X is a halide ion (I^- , Br^- or Cl^-). B and X together form an octahedron: $[\text{BX}_6]^{4-}$. A common example is methylammonium lead iodide, $\text{CH}_3\text{NH}_3\text{PbI}_3$: each unit cell of this compound consists of a central methylammonium (CH_3NH_3^+) in coordination with 12 anions of PbI_6 (occupying each corner), as shown in Fig. 1.6i [116]. Ion ‘A’ (CH_3NH_3^+) needs to be able to fit into the space between the eight octahedron, each connected to one another via ‘corner-sharing’ [117]. ‘A’ has a permanent electric dipole and is able to orient itself within the perovskite structure.

This ability to orient (and reorient) itself contributes to the high dielectric properties of perovskite materials, conferring upon the perovskites high mobilities and large diffusion lengths [118–120]. The good solution-processability and relatively low cost of perovskites, combined with their electric properties, give materials that are comparable to traditional crystalline Si and group III–V semiconductors [121]. Furthermore, solution-processable perovskites have absorption coefficients of $\approx 10^5 \text{ cm}^{-1}$ in the UV-visible section of the spectrum [122] and can therefore be

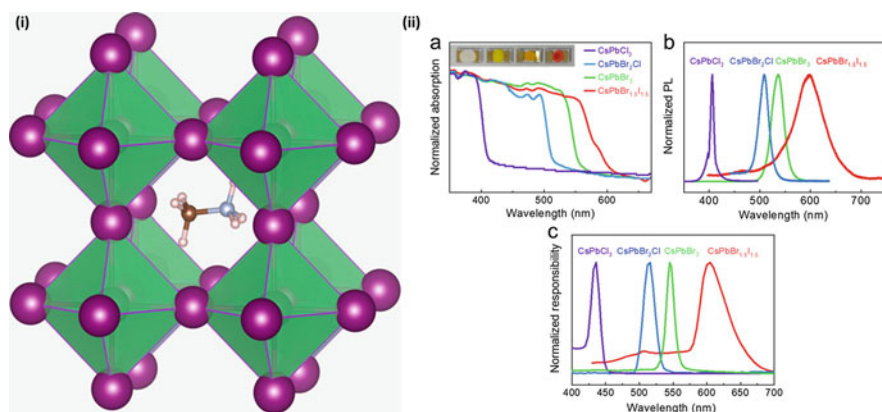


Fig. 1.6 (i) Methylammonium cation (CH_3NH_3^+) occupies the central ‘A’ site surrounded by 12 nearest-neighbour iodide ions in corner-sharing $[\text{PbI}_6]^{4-}$ octahedron (taken from [116], Fig 1; used through a Creative Commons Attribution 4.0 International License). (ii) Bandgap tuneability based on halide composition of the MHP: CsPbX_3 demonstrated by (a) the tuneable absorption of the MHP within thin-film devices (inset: a photograph of the devices); (b) photoluminescence spectra of CsPbX_3 films and (c) normalized responsivity of CsPbX_3 photodetectors. Adapted with permission from Xue J, Zhu Z, Xu X, Wang S, Xu L, Zou Y, Song J, Zeng H and Chen Q (2018) Narrowband Perovskite Photodetector-Based Image Array for Potential Application in Artificial Vision. *Nano Letters*, 18(12):7628–7634. Copyright (2018) American Chemical Society

fabricated as thin films, with rapid response times [123]. Perovskites are also capable of high specific detectivities, and their bandgaps can be tuned based on the halide ratio (Fig. 1.6ii) [61], making them strong candidates for filter-free narrowband photodetectors capable of detection of light of specific wavelengths [123–127].

The first OHP photodiodes were only realized in 2015 [51, 128, 129]. In a similar fashion to OPDs, OHPs have been designed to be either broadband or narrowband absorbing, with the same implications regarding the use of colour filters as has been discussed for inorganic semiconductors and OPDs (see Fig. 1.2). Since then, intensive research has produced photodiodes with figures of merit comparable and superior to those of OPDs and inorganic photosensors. For example, a broadband OHP developed by Dou et al. (2014) demonstrated a J_d of 10^{-10} A/cm² (at 0 V) and a detectivity of 10^{14} Jones [128]. Lin et al. demonstrated narrowband, filter-free OHPs (absorbing light with wavelengths from 610 to 690 nm) with an LDR = 120 dB, $J_d = 5 \times 10^{-8}$ A/cm² (at -5 V) and a detectivity of 1.9×10^{11} Jones at 650 nm and under a -0.5 bias [130]. Hu et al. fabricated a flexible OHP using a ‘vapour-solution’ process with a very low J_d ($\sim 3 \times 10^{-5}$ A cm⁻² at 1 V), an on/off ratio of 100 at 1 V, D^* greater than 10^{11} J and a linear response over 4 orders of magnitude incident power (at 680 nm and a bias of 1 V) [131].

OHPs have demonstrated the ability to switch between broadband and narrowband photodetection by changing between bottom and top illumination [118]. Furthermore, and in line with the scope of this article, OHPs have been integrated within image sensors and demonstrated reasonable performances [132–135]. For example, Wu and colleagues [133] fabricated a 10×10 flexible $\text{CH}_3\text{NH}_3\text{PbI}_{3-x}\text{Cl}_x$ -based OHP array on a polyethylene terephthalate (PET) substrate as an image sensor demonstration. This flexible OHP image sensor demonstrated the following: (1) an on/off current ratio of 1.2×10^3 under illumination (38.3 mW/cm²); (2) a detectivity (D^*) of up to 9.4×10^{11} Jones at a light intensity of 0.033 mW cm⁻¹ (corresponding to a responsivity = 2.17 AW⁻¹) and (3) a stable electrical performance and no visible physical change under repeated bending (from 0° to 150°), with only a slight decrease observed for the light current (due to an increase in the resistance of the electrodes with bending).

Although OHPs are still relatively undeveloped (it has only been 4 years since the first OHP was conceived), large strides have been made, as shown by the performance metrics that have been achieved. In the same way that OPDs are able to overcome the limitations faced by inorganic photodetectors, OHPs offer tuneability and the opportunity to be fabricated on flexible substrates. More importantly, there is still much to be discovered in the area of organic photodetection using these materials.

1.4 Phototransistors

Until now, we have discussed two-terminal photodiode devices. A second type of architecture consists of three terminals—this is a phototransistor. The extra terminal enables the device to sense the level of light and modify the current flowing between

the emitter and the photosensor (and photon collector), based on the level of light received. Thus, phototransistors (PTs) or field-effect transistors (FETs) combine the photosensing function of a diode with a high gain, due to the electric field effects of transistors, making them more sensitive, capable of providing rapid output, and able to produce a higher current than PDs. PTs are therefore used widely in applications such as encoders, smart cards, active matrix displays and photodetection for artificial vision [136–138].

The organic PT (OPT) as a device platform is a natural extension to the more widely established organic field effect transistor (OFET) devices. First reported by Tsumura et al. [139] and subsequently developed by Horowitz et al. [140], OFETs are now used across different device platforms, from developing the basic understanding of electronic properties of organic semiconductors to chemical and bio-electronic sensors [141]. Figure 1.7a shows the typical layout of an OPT [105]. Figure 1.7b shows typical optical absorption spectra of the materials when fabricated as thin-film devices, demonstrating, in this case, how the absorption profiles evolve with a change in the proportion of donor and acceptor in the semiconductor blend.

In the operation of OPTs/OFETs, the saturation drain current ($I_{d,sat}$), which refers to the maximum current carried by the drain of the OPT when the gate source = 0, is given by the Horowitz equation [141]:

$$I_{d,sat} = \frac{W}{2L} C_i \mu_{FE} (V_g - V_t)^2, \quad (1.3)$$

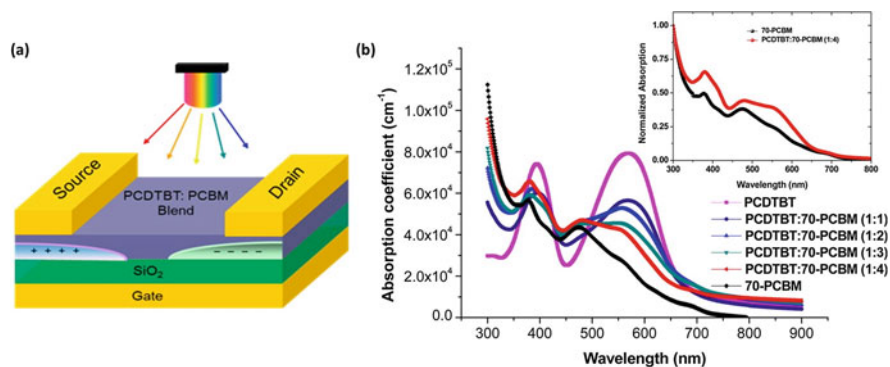


Fig. 1.7 (a) Device architecture of a typical OPT, in this case, with the light-absorbing layer consisting of poly[*N*-9'-heptadecanyl-2,7-carbazole-alt-5,5'-(4',7'-di-2-thienyl-2',1',3'-benzothiadiazole)] (PCDTBT) blended with [6,6]-phenyl C₆₁ butyric acid methyl ester (PCBM). (b) Absorption spectra of the neat polymer PCDTBT, 70-PCBM and PCDTBT/70-PCBM blends (i.e. thin films) in 1:1–1:4 ratios (by weight) on glass substrates (inset shows the absorption profile of the 1:4 blend compared with that of a neat 70-PCBM thin film on glass). Adapted with permission from Pandey AK, Aljada M, Pivrikas A, Velusamy M, Burn PL, Meredith P and Namdas EB (2014) Dynamics of Charge Generation and Transport in Polymer-Fullerene Blends Elucidated Using a PhotoFET Architecture. ACS Photonics, 1(2):114–120, ref. [105]. Copyright (2014) American Chemical Society

where W is the width of the channel, L the length of the channel, C_i the capacitance per unit area of the gate dielectric, μ_{FE} the field-effect mobility, V_g the gate voltage and V_t the so-called threshold voltage.

The photocurrent generated in the presence of light is calculated by taking the difference of I_d under illumination and I_d in the dark. The responsivity of OPTs can be estimated by taking the photocurrent density (J_{ph}) in either the p - or n -channel mode of OPTs, using the following equation [105]:

$$J_{ph} = \frac{1}{s} \int_{\lambda=300 \text{ nm}}^{\lambda=700 \text{ nm}} \varphi(\lambda) \cdot \text{EQE}(\lambda) d\lambda \quad (1.4)$$

where s is the active surface area of the photoFET channel, φ is the photon-flux from light source, EQE is the external quantum efficiency of either the p - or the n -channel operation of the OPT and λ is the absorption onset and cut off wavelengths of the photosensing material (or composition).

Evidence of significant photosensing in an OPT platform was first reported by Narayan and Kumar in 2001 [142]. Then, a major boost in the widespread adoption of OPTs came from the demonstration of ambipolar operation of solution-processed OFETs by Meijer et al. [143]. Since then, research interest in combining the photosensing ability of OFETs with their operation has grown significantly, and it has emerged as a new class of organic optoelectronic device in its own right. Like OPDs, OPTs usually require a D-A network for efficient photosensing, and these different components can be fabricated by standard spin coating, inkjet printing or vacuum sublimation processes (described in previous sections). Lombardo and Dodabalapur evaluated the non-geminate recombination rate in P3HT:PCBM photovoltaic blends using an ambipolar OFET geometry [144]. The optical gap and transport properties of the main absorber usually defines the photosensing efficiency of OPTs. Pandey et al. demonstrated that some of the most efficient compositions of polymer:fullerene blends benefit from the light responsive and good charge transporting ability of fullerenes by operating OPTs in p - and n -channel modes [105]. In OPTs, the surface states at the organic–dielectric interface play an important role with this interface property, determining the performance of OFETs and therefore the efficiency of photosensing in OPTs [105, 145]. Furthermore, the high photoconductive gain coupled with ‘sublinear responsivity to irradiance’ of OPTs enables a wider LDR than for photodiode-based image sensors [146, 147], which would be of clear benefit to machine and robotic vision systems.

Variation to the photosensing layer by the use of organic–inorganic hybrid layer is seen as yet another extension of OPTs; for example this could include OHP PTs or hybrid organic–halide perovskite PTs (hybrid OHP PTs) [148, 149].

Baeg et al. provide a concise overview of OPTs [150]. A more comprehensive review on the diversity of solution-processed materials for photosensing applications in OPDs as well as OPTs is presented by Pelayo et al. [52] and Gasparini et al. [98]. An impressive photosensing performance under UV light exposure with high gain was demonstrated using vacuum-sublimed thin films of small molecules (copper phthalocyanine and para-sexiphenyl) in OPTs by Qian et al. [151]. Li et al. reported

high photoresponsivity (R) values of 320 A/W over a broad range of lighting spectrum for $\text{CH}_3\text{NH}_3\text{PbI}_3$ -based OPH PTs [148].

OPTs have been integrated within image sensors and have demonstrated themselves fully capable of overcoming some of the limitations of conventional inorganic-based image sensors. For example, Pierre et al. developed a solution-processed OPT on a flexible substrate able to achieve a dynamic range of 103 dB for a video capture (30 frames/s) [146]. Milvich et al. designed and tested the performance of an array of 16 OPTs based on dinaphtho[2,3-b:2',3'-f]thieno[3,2-b]thiophene covering an area of $2 \times 4 \text{ cm}^2$ on a flexible PEN substrate [152].

The scope for OPTs in image sensors and MVS is expected to grow, and it will be interesting to see photosensing and switching functions further refined towards real-world applications.

1.5 Conclusions and Outlook

Current MVS have been recognized as having severe limitations when it comes to the demands of modern-day applications involving machine vision and robotics. These include weak light absorption over the visible range, low dynamic range, existence of crosstalk, an inability to cope with illuminant variation and incompatibility with complicated processing and fabrication on flexible, miniaturized devices. Such limitations could be overcome using alternative photoactive materials fabricated on the ROIC of the image sensor.

We have presented an outlook for further development of OPD systems for digital imaging, colour constancy and depth measurements. The soft, conformal and up-scaling of OPDs allow unparalleled possibilities of designed imaging systems that are not only low power-consuming and light-weight but highly intelligent in selective sensing over a range of applications.

While OHPs are still in the early stages of understanding and development, much has been accomplished already, and their potential for transforming the landscape of machine vision and artificial vision in robotic systems will surely dawn in the near future.

Knowledge translation [153, 154] is key to realizing the potential of both OPDs and OHPs in commercial machine and robotic vision systems. Furthermore, a more cross-disciplinary approach needs to be implemented to harness the potential of OPDs and OHPs in MVS; at the moment, the field lacks chemists and material scientists with a strong knowledge of image sensing, machine vision and future market trends. Likewise, camera experts are largely ignorant of the advances made in developing alternative semiconductor materials able to replace Si or InGaAs. The authors hope that this chapter helps to bridge this gap and initiate conversations between chemists, physicists, material scientists and mechatronic engineers.

References

1. ResearchandMarkets.com. (2018). *Global image sensors market analysis, growth, trends & forecast 2018-2023*. Retrieved February 21, 2019, from <https://www.businesswire.com/news/home/20180530005711/en/Global-Image-Sensors-Market-Analysis-Growth-Trends>
2. IC insights. (2018). *CMOS image sensor sales stay on record-breaking pace*. Retrieved February 21, 2019, from <http://www.icinsights.com/data/articles/documents/1065.pdf>
3. Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., & Blasco, J. (2011). Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food and Bioprocess Technology*, 4(4), 487–504.
4. Patel, K., Kar, A., Jha, S., & Khan, M. (2011). Machine vision system: A tool for quality inspection of food and agricultural products. *Journal of Food Science Technology*, 49(2), 1–19.
5. Kaur, H., Sawhney, B. K., & Jawandha, S. K. (2018). Evaluation of plum fruit maturity by image processing techniques. *Journal of Food Science and Technology*, 55(8), 3008–3015.
6. Sridharan, M., & Stone, P. (2009). Color learning and illumination invariance on mobile robots: A survey. *Robotics and Autonomous Systems*, 57, 629–644.
7. Marchant, J. A., Tillett, N. D., & Onyango, C. M. (2004). Dealing with color changes caused by natural illumination in outdoor machine vision. *Cybernetics and Systems: An International Journal*, 35(1), 19–33.
8. Buluswar, S. D., & Draper, B. A. (1998). Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 11, 245–256.
9. Maddern, W., Stewart, A. D., McManus, C., Upcroft, B., Churchill, W., & Newman, P. (2014). Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation*.
10. Fernández, R., Montes, H., Salinas, C., Sarria, J., & Armada, M. (2013). Combination of RGB and multispectral imagery for discrimination of cabernet sauvignon grapevine elements. *Sensors (Basel, Switzerland)*, 13(6), 7838–7859.
11. Bloss, R. (2013). Robots use machine vision and other smart sensors to aid innovative picking, packing and palletizing. *Industrial Robot: An International Journal*, 40(6), 525–529.
12. Oestreich, J. M., Tolley, W. K., & Rice, D. A. (1995). The development of a color sensor system to measure mineral compositions. *Minerals Engineering*, 8(1/2), 31–39.
13. Schmittmann, O., & Lammers, P. S. (2017). A true-color sensor and suitable evaluation algorithm for plant recognition. *Sensors*, 17(8), 1823.
14. Yamada, K., Nakano, T., & Yamamoto, S. (1998). A vision sensor having an expanded dynamic range for autonomous vehicles. *IEEE Transactions on Vehicular Technology*, 47(1), 332–341.
15. Xiong, N. N., Yang, S., Kangye, Y., Changhoon, L., & Chunxue, W. (2018). Color sensors and their applications based on real-time color image segmentation for cyber physical systems. *EURASIP Journal on Image and Video Processing*, 2018, 23.
16. Pathare, P. B., Opara, U. L., & Al-Said, F. A. (2013). Colour measurement and analysis in fresh and processed foods: A review. *Food and Bioprocess Technology*, 6(1), 36–60.
17. Ratnasingam, S., & Collins, S. (2008). An algorithm to determine the chromaticity under non-uniform illuminant. In *ICISP 2008: Image and signal processing* (pp. 244–253).
18. Logvinenko, A. D., Funt, B., Mirzaei, H., & Tokunaga, R. (2015). Rethinking colour constancy. *PLoS One*, 10(9), e0135029.
19. Lukac, R., Plataniotis, K. N., & Hatzinakos, D. (2005). Color image zooming on the Bayer pattern. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(11), 1475–1492.
20. Bayer, B. E. (1975). *Color imaging array*. U.S. Patent No. 3,971,065.
21. Nakamura, J. (2006). Basics of image sensors. In *Image sensors and signal processing for digital still cameras* (pp. 55–61). Boca Raton, FL: Taylor & Francis.

22. Suzuki, T. (2010). Challenges of image-sensor development. In *2010 IEEE International Solid-State Circuits Conference—(ISSCC)* (pp. 28–30).
23. Teledyne Dalsa, CCD vs CMOS. Retrieved March 30, 2019, from <https://www.teledynedalsa.com/en/learn/knowledge-center/ccd-vs-cmos/>
24. Lahav, A., Fenigstein, A., & Strum, A. (2014). Backside illuminated (BSI) complementary metal-oxide-semiconductor (CMOS) image sensors. In *High performance silicon imaging fundamentals and applications of CMOS and CCD sensors* (pp. 98–123).
25. Nomoto, T., Oike, Y., & Wakabayashi, H. (2016). Accelerating the sensing world through imaging evolution. In *2016 Symposium on VLSI Circuits Digest of Technical Papers* (pp. 1–4).
26. Fossum, E. R. (1997). CMOS image sensors: Electronic camera-on-a-chip. In *IEEE Proceedings of International Electron Devices Meeting*.
27. Fossum, E. R., & Hondongwa, D. B. (2014). A review of the pinned photodiode for CCD and CMOS image sensors. *IEEE Journal of the Electron Devices Society*, 2(3), 33–43.
28. Bigas, M., Cabruja, E., Forest, J., & Salvi, J. (2006). Review of CMOS image sensors. *Microelectronics Journal*, 37(5), 433–451.
29. Lesser, M. (2014). Charge coupled device (CCD) image sensors. In *High performance silicon imaging: Fundamentals and applications of CMOS and CCD sensors* (pp. 78–97).
30. Hamilton, G., Brown, N., Oseroff, V., Huey, B., Seagraves, R., Sudar, D., Kumler, J., Albertson, D., & Pinkel, D. (2006). A large field CCD system for quantitative imaging of microarrays. *Nucleic Acids Research*, 34(8), e58, 1–e58,14.
31. Catrysse, P., & Wandell, B. A. (2003). Integrated color pixels in 0.18- μm complementary metal oxide semiconductor technology. *Journal of the Optical Society of America A*, 20(12), 2293–2306.
32. Knipp, D., Herzog, P. G., & Stiebig, H. (2002). Stacked amorphous silicon color sensors. *IEEE Transactions on Electron Devices*, 49(1), 170–176.
33. Hubel, P. M. (2005). Foveon technology and the changing landscape of digital cameras. In *13th IS&T Color Imaging Conf., Scottsdale, AZ, USA* (pp. 314–317).
34. Lyon, R., & Hubel, P. M. (2002). Eyeing the camera: Into the next century. In *IS&T/SID 10th Color Imaging Conference Proceedings Scottsdale, AZ* (p. 349).
35. Blockstein, L., & Yadid-Pecht, O. (2010). Crosstalk quantification, analysis, and trends in CMOS image sensors. *Applied Optics*, 49(24), 4483–4488.
36. Anzagira, L., & Fossum, E. R. (2015). Color filter array patterns for small-pixel image sensors with substantial cross talk. *Journal of the Optical Society of America A*, 32(1), 28–34.
37. Langfelder, G., Longoni, A., & Zaraga, F. (2009). Further developments on a novel color sensitive CMOS detector. In *Proceedings Volume 7356, Optical Sensors 2009; 73562A*.
38. Longoni, A., Zaraga, F., Langfelder, G., & Bombelli, L. (2008). The transverse field detector (TFD): A novel color-sensitive CMOS device. *IEEE Electron Device Letters*, 29(12), 1306–1308.
39. Jansen-van Vuuren, R. D., Armin, A., Pandey, A. K., Burn, P. L., & Meredith, P. M. (2016). Organic photodiodes: The future of full color detection and image sensing. *Advanced Materials*, 28, 4766–4802.
40. Moloney, A. M., Wall, L., Mathewson, A., Healy, G., & Jackson, J. C. (2006). Novel black silicon PIN photodiodes. In *Proceedings Volume 6119, Semiconductor Photodetectors III; 61190B*.
41. Tut, T., & Dan, Y. (2014). Silicon photodetectors integrated with vertical silicon nitride waveguides as image sensor pixels: Fabrication and characterization. *Journal of Vacuum Science & Technology B*, 32, 031201.
42. Theil, J. A., Snyder, R., Hula, D., Lindahl, K., Haddad, H., & Roland, J. (2002). a-Si:H photodiode technology for advanced CMOS active pixel sensor imagers. *Journal of Non-Crystalline Solids*, 299–302, 1234–1239.
43. Konstantatos, G., & Sargent, E. H. (2010). Nanostructured materials for photon detection. *Nature Nanotechnology*, 5, 391–400.

44. Konstantatos, G., & Sargent, E. H. (2009). Solution-processed quantum dot photodetectors. *Proceedings of the IEEE*, 97(10), 1666–1683.
45. Goetzberger, A., & Hebling, C. (2000). Photovoltaic materials, past, present, future. *Solar Energy Materials and Solar Cells*, 62(1–2), 1–19.
46. Lule, T., Benthien, S., Keller, H., Mutze, F., Rieve, P., Seibel, K., Sommer, M., & Bohm, M. (2000). Sensitivity of CMOS based imagers and scaling perspectives. *IEEE Transactions on Electron Devices*, 47(11), 2110–2122.
47. Liu, L., Yang, C., Patanè, A., Yu, Z., Yan, F., Wang, K., Lu, H., Liab, J., & Zhao, L. (2017). High-detectivity ultraviolet photodetectors based on laterally mesoporous GaN. *Nanoscale*, 9, 8142–8148.
48. Haddadi, A., Dehzangi, A., Adhikary, S., Chevallier, R., & Razeghi, M. (2017). Background-limited long wavelength infrared InAs/InAs_{1-x}Sbx type-II superlattice-based photodetectors operating at 110 K. *Applied Physics Letters Materials*, 5, 035502.
49. Gong, X., Tong, M., Xia, Y., Cai, W., Moon, J. S., Cao, Y., Yu, G., Shieh, C.-L., Nilsson, B., & Heeger, A. J. (2009). High-detectivity polymer photodetectors with spectral response from 300 nm to 1450 nm. *Science*, 325(5948), 1665–1667.
50. Armin, A., Jansen-van Vuuren, R. D., Kopidakis, N., Burn, P. L., & Meredith, P. (2015). Narrowband light detection via internal quantum efficiency manipulation of organic photodiodes. *Nature Communications*, 6, 6343.
51. Lin, Q., Armin, A., Lyons, D. M., Burn, P. L., & Meredith, P. (2015). Low noise, IR-blind organohalide perovskite photodiodes for visible light detection and imaging. *Advanced Materials*, 27(12), 2060–2064.
52. Pelayo de García de Arquer, F., Armin, A., Meredith, P., & Sargent, E. H. (2017). Solution-processed semiconductors for next-generation photodetectors. *Nature Reviews Materials*, 2(16100), 1–16.
53. Xie, C., & Yan, F. (2017). Flexible photodetectors based on novel functional materials. *Small*, 13(43), 1701822(1 of 36).
54. Ng, T. N., Wong, W. S., Lujan, R. A., & Street, R. A. (2009). Characterization of charge collection in photodiodes under mechanical strain: Comparison between organic bulk heterojunction and amorphous silicon. *Advanced Materials*, 21(18), 1855–1859.
55. Vasavi, V., Shaik, A. F., & Sunkara, P. C. K. (2018). Moving object classification under illumination changes using binary descriptors. In M. Rivas-Lopez, O. Sergiyenko, W. Flores-Fuentes, & J. C. Rodríguez-Quirón (Eds.), *Optoelectronics in machine vision-based theories and applications* (pp. 188–189). Hershey, PA: IGI Global.
56. Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Computers and Electrical Engineering*, 38(5), 1186–1195.
57. Son, J., Kim, S., & Sohn, K. (2015). A multi-vision sensor-based fast localization system with image matching for challenging outdoor environments. *Expert Systems with Applications*, 42(22), 8830–8839.
58. Antognazza, M. R., Musitelli, D., Perissinotto, S., & Lanzani, S. (2010). Spectrally selected photodiodes for colorimetric application. *Organic Electronics*, 11(3), 357–362.
59. Xiong, J., Liu, Z., Lin, R., Bu, R., He, Z., Yang, Z., & Liang, C. (2018). Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (CCD) vision sensor with artificial illumination. *Sensors*, 18(969), 1–17.
60. Ratnasingam, S., & McGinnity, T. M. (2012). Chromaticity space for illuminant invariant recognition. *IEEE Transactions on Image Processing*, 21(8), 3612–3623.
61. Xue, J., Zhu, Z., Xu, X., Wang, S., Xu, L., Zou, Y., Song, J., Zeng, H., & Chen, Q. (2018). Narrowband perovskite photodetector-based image array for potential application in artificial vision. *Nano Letters*, 18(12), 7628–7634.
62. Flores-Fuentes, W., Miranda-Vega, J. E., Rivas-López, M., Sergiyenko, O., Rodríguez-Quirón, J. C., & Lindner, L. (2018). Comparison between different types of sensors used in the real operational environment based on optical scanning system. *Sensors*, 18(1684), 1–15.

63. Guo, W., Rage, U. K., & Ninomiya, S. (2013). Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model. *Computers and Electronics in Agriculture*, *96*, 58–66.
64. Estribeau, M., & Magnan, P. (2005). CMOS pixels crosstalk mapping and its influence on measurements accuracy for space applications. In *Proceedings of SPIE, Volume 5978, Sensors, Systems, and Next-Generation Satellites IX*; 597813.
65. Natali, D., & Caironi, M. (2016). Organic photodetectors. In *Photodetectors, materials, devices and applications* (p. 233).
66. Kudo, K., & Moriizumi, T. (1981). Spectrum-controllable color sensors using organic dyes. *Applied Physics Letters*, *39*, 609–611.
67. Yu, G., Pakbaz, K., & Heeger, A. J. (1994). Semiconducting polymer diodes: Large size, low cost photodetectors with excellent visible-ultraviolet sensitivity. *Applied Physics Letters*, *64*, 3422–3424.
68. Biele, M., Benavides, C. M., Hürdler, J., Tedde, S. F., Brabec, C. J., & Schmidt, O. (2019). Spray-coated organic photodetectors and image sensors with silicon-like performance. *Advanced Materials Technologies*, *4*(1), 1800158:1–6.
69. Yang, D., & Ma, D. (2019). Development of organic semiconductor photodetectors: From mechanism to applications. *Advanced Optical Materials*, *7*(1), 1800522:1–23.
70. Cai, S. (2019). Materials and designs for wearable photodetectors. *Advanced Materials, Early View*, *31*(18), 1808138.
71. Natali, D., & Sampietro, M. (2003). Detectors based on organic materials: Status and perspectives. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *512*(1–2), 419–426.
72. Arkhipov, V. I., & Bässler, H. (2003). Exciton dissociation and charge photogeneration in pristine and doped conjugated polymers. *Physica Status Solidi A*, *201*(6), 1152–1187.
73. Gregg, B. A. (2003). Excitonic solar cells. *Journal of Physical Chemistry B*, *107*(20), 4688–4698.
74. Thompson, B. C., & Fréchet, J. M. J. (2007). Polymer–fullerene composite solar cells. *Angewandte Chemie International Edition*, *47*(1), 58–77.
75. Rauch, T., Henseler, D., Schilinsky, P., Waldauf, C., Hauch, J., Brabec, C. J. (2004). Performance of bulk-heterojunction organic photodetectors. In *4th IEEE Conference on Nanotechnology*.
76. Yu, G., Wang, J., McElvain, J., & Heeger, A. J. (1999). Large-area, full-color image sensors made with semiconducting polymers. *Advanced Materials*, *10*(17), 1431–1434.
77. Seo, H., Aihara, S., Watabe, T., Ohtake, H., Sakai, T., Kubota, M., Egami, N., Hiramatsu, T., Matsuda, T., & Furuta, M. (2011). A 128×96 pixel stack-type color image sensor: Stack of individual blue-, green-, and red-sensitive organic photoconductive films integrated with a ZnO thin film transistor readout circuit. *Japanese Journal of Applied Physics*, *50*, 024103.
78. Deckman, I., Lechêne, P. B., Pierre, A., & Arias, A. C. (2018). All-printed full-color pixel organic photodiode array with a single active layer. *Organic Electronics*, *56*, 139–145.
79. Jansen-van Vuuren, R. D., Pivrikas, A., Pandey, A. K., & Burn, P. L. (2013). Colour selective organic photodetectors utilizing ketocyanine-cored dendrimers. *Journal of Materials Chemistry C*, *1*, 3532–3543.
80. Jansen-van Vuuren, R. D., Johnstone, K. D., Ratnasingam, S., Barcena, H., Deakin, P. C., Pandey, A. K., Burn, P. L., Collins, S., & Samuel, I. D. W. (2010). Determining the absorption tolerance of single chromophore photodiodes for machine vision. *Applied Physics Letters*, *96*, 253303.
81. Han, M. G., Park, K.-B., Bulliard, X., Lee, G. H., Yun, S., Leem, D.-S., Heo, C.-J., Yagi, T., Sakurai, R., Ro, T., Lim, S.-J., Sul, S., Na, K., Ahn, J., Jin, Y. W., & Lee, S. (2016). Narrow-band organic photodiodes for high-resolution imaging. *Applied Materials and Interfaces*, *8*(39), 26143–26151.
82. Ratnasingam, S., & Collins, S. (2010). Study of the photodetector characteristics of a camera for color constancy in natural scenes. *Journal of the Optical Society of America A*, *27*(2), 286–294.

83. Yoon, S., Koh, C. W., Woo, H. Y., & Chung, D. S. (2018). Systematic optical design of constituting layers to realize high-performance red-selective thin-film organic photodiodes. *Advanced Optical Materials*, 6(4), 1701085.
84. Yazmaciyan, A., Meredith, P., & Armin, A. (2019). Cavity enhanced organic photodiodes with charge collection narrowing. *Advanced Optical Materials, Early View*, 7(8), 1801543.
85. Kielar, M., Dhez, O., Pecastaings, G., Curutchet, A., & Hirsh, L. (2016). Long-term stable organic photodetectors with ultra low dark currents for high detectivity applications. *Scientific Reports*, 6(39201), 1–11.
86. Nie, R., Deng, X., Feng, L., Hu, G., Wang, Y., Yu, G., & Xu, J. (2017). Highly sensitive and broadband organic photodetectors with fast speed gain and large linear dynamic range at low forward bias. *Small*, 13(24), 1603260.
87. Hu, L., Han, J., Qiao, W., Zhou, X., Wang, C., Ma, D., Li, Y., & Wang, Z. H. (2018). Side-chain engineering in naphthalenediimide-based n-type polymers for high-performance all-polymer photodetectors. *Polymer Chemistry*, 9(3), 327–334.
88. *Si image sensor S1336-18BK, Hamamatsu Photonics KK*. Retrieved February 28, 2019, from <https://www.datasheets360.com/part/detail/s1336-18bk/1668072914334887650/>
89. *InGaAs image sensor G10899-03K, Hamamatsu*. Retrieved February 28, 2019, from <https://www.hamamatsu.com/us/en/product/type/G10899-03K/index.html>
90. *Si (S6428-01) blue light absorbing image sensor, Hamamatsu*. Retrieved February 28, 2019, from <https://www.hamamatsu.com/us/en/product/type/S6428-01/index.html>
91. *Si (S6429-01) green light absorbing image sensor, Hamamatsu*. Retrieved February 28, 2019, from <https://www.hamamatsu.com/us/en/product/type/S6429-01/index.html>
92. *Si (S6430-01) red light absorbing image sensor, Hamamatsu*. Retrieved February 28, 2019, from <https://www.hamamatsu.com/us/en/product/type/S6430-01/index.html>
93. Jansen-van Vuuren, R. D., Deakin, P. C., Olsen, S., & Burn, P. L. (2014). Tuning the optoelectronic properties of cyanine and ketocyanine dyes by incorporation of 9,9-di-n-propylfluorenylindolenine. *Dyes and Pigments*, 101, 1–8.
94. Jahnel, M., Tomshcke, M., Fehse, K., Vogel, U., An, J. D., Park, H., & Im, C. (2015). Integration of near infrared and visible organic photodiodes on a complementary metal-oxide-semiconductor compatible backplane. *Thin Solid Films*, 592(Part A), 94–98.
95. Zalar, P., Matsuhisa, N., Suzuki, T., Enomoto, S., Koizumi, M., Yokota, T., Sekino, M., & Someya, T. (2018). A monolithically processed rectifying pixel for high-resolution organic imagers. *Advanced Electronic Materials*, 4(6), 1700601.
96. Swathi, K., & Narayan, K. S. (2016). Image pixel device using integrated organic electronic components. *Applied Physics Letters*, 109, 193302.
97. Baierl, D., Pancheri, L., Schmidt, M., Stoppa, D., Betta, G.-F. D., Scarpa, G., & Lugli, P. (2012). A hybrid CMOS-imager with a solution-processable polymer as photoactive layer. *Nature Communications*, 3, 1175.
98. Gasparini, N., Gregori, A., Salvador, M., Biele, M., Wadsworth, A., Tedde, S., Baran, D., McCulloch, I., & Brabec, C. J. (2018). Visible and near-infrared imaging with nonfullerene-based photodetectors. *Advanced Materials Technologies*, 3(7), 1800104.
99. Lim, S.-J., Leem, D.-S., Park, K.-B., Kim, K.-S., Sul, S., Na, K., Lee, G. H., Heo, C.-J., Lee, K.-H., Bulliard, X., Satoh, R.-I., Yagi, T., Ro, T., Im, D., Jung, J., Lee, M., Lee, T.-Y., Han, M. G., Jin, W. Y., & Lee, S. (2015). Organic-on-silicon complementary metal-oxide-semiconductor colour image sensors. *Scientific Reports*, 5, 7708.
100. Lee, K.-H., Leem, D.-S., Castrucci, J. S., Park, K.-B., Bulliard, X., Kim, K.-S., Jin, Y. W., Lee, S., Bender, T. P., & Park, S. Y. (2013). Green-sensitive organic photodetectors with high sensitivity and spectral selectivity using subphthalocyanine derivatives. *ACS Applied Materials and Interfaces*, 5(24), 13089–13095.
101. Leem, D.-S., Lim, S.-J., Bulliard, X., Lee, G. H., Lee, K.-H., Yun, S., Yagi, T., Satoh, R.-I., Park, K.-B., Choi, Y. S., Jin, Y. W., Lee, S. (2016). Recent developments in green light sensitive organic photodetectors for hybrid CMOS image sensor applications (conference presentation). In *SPIE Proceedings, 9944, Organic Sensors and Bioelectronics IX; 99440B*.

102. Panasonic Corporation. (2018). *Panasonic develops industry's first 8K high-resolution, high performance, global shutter technology using organic-photoconductive-film cmos image sensor*. Press Release. Retrieved February 28, 2019, from <https://news.panasonic.com/global/press/data/2018/02/en180214-2/en180214-2.pdf>
103. Editor. (2019). *The OSA direct newsletter. ISORG and plastic logic co-develop the world's first organic image sensor on plastic*. Retrieved February 28, 2019, from <http://www.osadirect.com/news/article/980/isorg-and-plastic-logic-co-develop-the-worlds-first-organic-image-sensor-on-plastic/>
104. Andrew Lloyd & Associates. (2018). *Isorg raises €24M to finance the ramp up of large-scale commercialization*. Retrieved February 28, 2019, from <http://ala.com/isorg-raises-e24m-to-finance-the-ramp-up-of-large-scale-commercialization/>
105. Pandey, A. K., Aljada, M., Pivrikas, A., Velusamy, M., Burn, P. L., Meredith, P., & Namdas, E. B. (2014). Dynamics of charge generation and transport in polymer-fullerene blends elucidated using a PhotoFET architecture. *ACS Photonics*, *1*(2), 114–120.
106. Ullah, M., Yambem, S. D., Moore, E. G., Namdas, E. B., & Pandey, A. K. (2015). Singlet fission and triplet exciton dynamics in rubrene/fullerene heterojunctions: Implications for electroluminescence. *Advanced Electronic Materials*, *1*(12), 1500229.
107. Pandey, A. K. (2015). Highly efficient spin-conversion effect leading to energy up-converted electroluminescence in singlet fission photovoltaics. *Scientific Reports*, *5*, 7787.
108. Lee, W., Kobayashi, S., Nagase, M., Jimbo, Y., Saito, I., Inoue, Y., & Yambe, T. (2018). Nonthrombogenic, stretchable, active multielectrode array for electroanatomical mapping. *Science*, *4*(10), eaau2426.
109. Rogers, J., Malliaras, G., & Someya, T. (2018). Biomedical devices go wild. *Science Advances*, *4*(9), eaav1889.
110. Bock, R. D. (2018). Low-cost 3D security camera. In *Proceedings Volume 10643, Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything; 106430E*.
111. Semeniutaa, O., Dransfeld, S., Martinsena, K., & Falkmanc, P. (2018). Towards increased intelligence and automatic improvement in industrial vision systems. In *11th CIRP Conference on Intelligent Computation in Manufacturing Engineering—CIRP ICME'17* (pp. 256–261).
112. Smith, L. N., Zhang, W., Hansen, M. F., Hales, I. J., & Smith, M. L. (2018). Innovative 3D and 2D machine vision methods for analysis of plants and crops in the field. *Computers in Industry*, *97*, 122–131.
113. Haouchine, N., Kuang, W., Cotin, S., & Yip, M. (2018). Vision-based force feedback estimation for robot-assisted surgery using instrument-constrained biomechanical three-dimensional maps. *IEEE Robotics and Automation Letters*, *3*(3), 2160–2165.
114. Hilliges, O., Weiss, H. M., Izadi, S., Kim, D., & Rother, C. C. E. (2018). *Using photometric stereo for 3D environment modeling*. US Patent Application US9857470B2.
115. Murray, D., & Little, J. J. (2000). Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, *8*(161), 161–171.
116. Eames, C., Frost, J. M., Barnes, P. R. F., O'Regan, B. C., Walsh, A., & Islam, M. S. (2015). Ionic transport in hybrid lead iodide perovskite solar cells. *Nature Communications*, *6*, 7497.
117. Saparov, B., & Mitzi, D. B. (2016). Organic–inorganic perovskites: Structural versatility for functional materials design. *Chemical Reviews*, *116*(7), 4558–4596.
118. Saidaminov, M. I., Haque, M. A., Savoie, M., Abdelhady, A. L., Cho, N., Dursun, I., Buttner, U., Alarousu, E., Wu, T., & Bakr, O. M. (2016). Perovskite photodetectors operating in both narrowband and broadband regimes. *Advanced Materials*, *28*(37), 8144–8149.
119. Xiao, Z. (2016). Thin-film semiconductor perspective of organometal trihalide perovskite materials for high-efficiency solar cells. *Materials Science and Engineering: R: Reports*, *101*, 1–38.
120. Strainks, S. D., Eperon, G. E., Grancini, G., Menelaou, C., Alcocer, M. J. P., Leijtens, T., Herz, L. M., Petrozza, A., & Snaith, H. J. (2013). Electron-hole diffusion lengths exceeding 1 micrometer in an organometal trihalide perovskite absorber. *Science*, *342*(6156), 341–344.

121. Brenner, T. M., Egger, D. A., Kronik, L., Hodes, G., & Cahen, D. (2016). Hybrid organic—Inorganic perovskites: Low-cost semiconductors with intriguing charge-transport properties. *Nature Reviews Materials*, *1*, 15007.
122. Burschka, J., Pellet, N., Moon, S.-J., Humphry-Baker, R., Gao, P., Nazeeruddin, M. K., & Grätzel, M. (2013). Sequential deposition as a route to high-performance perovskite-sensitized solar cells. *Nature*, *499*, 316–319.
123. Ahmadi, M., Wu, T., & Hu, B. (2017). A review on organic–inorganic halide perovskite photodetectors: Device engineering and fundamental physics. *Advanced Materials*, *29*(41), 1605242.
124. Zhou, J., & Huang, J. (2017). Photodetectors based on organic–inorganic hybrid lead halide perovskites. *Advanced Materials*, *5*(1), 1700256.
125. Wang, H., & Kim, D. H. (2017). Perovskite-based photodetectors: Materials and devices. *Chemistry Society Reviews*, *46*, 5204–5236.
126. Wang, X., Li, M., Zhang, B., Wang, H., Zhao, Y., & Wang, B. (2018). Recent progress in organometal halide perovskite photodetectors. *Organic Electronics*, *52*, 172–183.
127. Tian, W., Zhou, H., & Li, L. (2017). Hybrid organic–inorganic perovskite photodetectors. *Small*, *13*(41), 1702107.
128. Dou, L., Yang, Y. M., You, J., Hong, Z., Chang, W.-H., Li, G., & Yang, Y. (2014). Solution-processed hybrid perovskite photodetectors with high detectivity. *Nature Communications*, *5*, 5404.
129. Sutherland, B. R., Johnston, A. K., Ip, A. H., Xu, J., Adinolfi, V., Kanjanaboos, P., & Sargent, E. H. (2015). Sensitive, fast, and stable perovskite photodetectors exploiting interface engineering. *ACS Photonics*, *2*(8), 1117–1123.
130. Lin, Q., Armin, A., Burn, P. L., & Meredith, P. (2015). Filterless narrowband visible photodetectors. *Nature Photonics*, *9*, 687–694.
131. Hu, W., Huang, W., Yang, S., Wang, X., Jiang, Z., Zhu, X., Zhou, H., Liu, H., Zhang, Q., Zhuang, X., Yang, J., Kim, D. H., & Pan, A. (2017). High-performance flexible photodetectors based on high-quality perovskite thin films by a vapor–solution method. *Advanced Materials*, *29*(43), 1703256.
132. Lee, W., Lee, J., Yun, H., Kim, J., Park, J., Choi, C., Kim, D. C., Seo, H., Lee, H., Yu, J. W., Lee, W. B., & Kim, D.-H. (2017). Perovskite thin films: High-resolution spin-on-patterning of perovskite thin films for a multiplexed image sensor array. *Advanced Materials*, *29*(40), 1702902.
133. Wu, W., Wang, X., Han, X., Yang, Z., Gao, G., Zhang, Y., Hu, J., Tan, Y., Pan, A., & Pan, C. (2018). Flexible photodetector arrays based on patterned $\text{CH}_3\text{NH}_3\text{PbI}_{3-x}\text{Cl}_x$ perovskite film for real-time photosensing and imaging. *Advanced Materials*, *31*(3), 1805913.
134. Lyashenko, D., Perez, A., & Zakhidov, A. (2016). High-resolution patterning of organohalide lead perovskite pixels for photodetectors using orthogonal photolithography. *Physica Status Solid A*, *214*(1), 1600302.
135. Gu, L., Tavakoli, M. M., Zhang, D., Zhang, Q., Waleed, A., Xiao, Y., Tsui, K.-H., Lin, Y., Liao, L., Wang, J., & Fan, Z. (2016). 3D arrays of 1024-pixel image sensors based on lead halide perovskite nanowires. *Advanced Materials*, *28*(44), 9713–9721.
136. Kim, M. S., Lee, G. J., Kim, H. M., & Song, Y. M. (2017). Parametric optimization of lateral NIPIN phototransistors for flexible image sensors. *Sensors*, *17*(8), 1774:1–13.
137. Tomioka, K., Miyake, K., Misawa, K., Toyoda, K., Ishizaki, T., & Kimura, M. (2018). Photosensing circuit using thin-film transistors for retinal prosthesis. *Japanese Journal of Applied Physics*, *57*, 1002B1.
138. Kimura, M., Miura, Y., Ogura, T., Ohno, S., Hachida, T., Nishizaki, Y., & Shima, T. (2010). Device characterization of p/i/n thin-film phototransistor for photosensor applications. *IEEE Electron Device Letters*, *31*(9), 984–986.
139. Tsumura, A., Koezuka, H., & Ando, T. (1986). Macromolecular electronic device: Field-effect transistor with a polythiophene thin film. *Applied Physics Letters*, *49*, 1210.
140. Horowitz, G., Fichou, D., Peng, X., Xu, Z., & Garnier, F. (1989). A field-effect transistor based on conjugated alpha-sexithienyl. *Solid State Communications*, *72*(4), 381–384.

141. Horowitz, G. (2004). Organic thin film transistors: From theory to real devices. *Journal of Materials Research*, 19(7), 1946–1962.
142. Narayan, K. S., & Kumar, N. (2001). Light responsive polymer field-effect transistor. *Applied Physics Letters*, 79, 1891.
143. Meijer, E. J., Leeuw, D. M. D., Setayesh, S., Van Veenendaal, E., Huisman, B.-H., Blom, P. W. M., Hummelen, J. C., Scherf, U., & Klapwijk, T. M. (2003). Solution-processed ambipolar organic field-effect transistors and inverters. *Nature Materials*, 2, 678–682.
144. Ooi, Z.-E., Danielson, E., Liang, K., Lombardo, C., & Dodabalapur, A. (2014). Evaluating charge carrier mobility balance in organic bulk heterojunctions using lateral device structures. *Journal of Physical Chemistry C*, 18(32), 18299–18306.
145. Unni, K. N. N., Dabos-Seignon, S., Pandey, A. K., & Nunzi, J.-M. (2008). Influence of the polymer dielectric characteristics on the performance of pentacene organic field-effect transistors. *Solid-State Electronics*, 52(2), 179–181.
146. Pierre, A., Gaikwad, A., & Arias, A. C. (2017). Charge-integrating organic heterojunction phototransistors for wide-dynamic-range image sensors. *Nature Photonics*, 11, 193–199.
147. Hwang, I., Kim, J., Lee, M., Lee, M.-W., Kim, H.-J., Kwon, H.-I., Hwang, D. K., Kim, M., Yoon, H., Kim, Y.-H., & Park, S. K. (2017). Wide-spectral/dynamic-range skin-compatible phototransistors enabled by floated heterojunction structures with surface functionalized SWCNTs and amorphous oxide semiconductors. *Nanoscale*, 9, 16711–16721.
148. Li, F., Ma, C., Wang, H., Hu, W., Yu, W., Sheikh, A. D., & Wu, T. (2015). Ambipolar solution-processed hybrid perovskite phototransistors. *Nature Communications*, 6, 8238.
149. Cao, M., Zhang, Y., Yu, Y., & Yao, J. (2018). Improved perovskite phototransistor prepared using multi-step annealing method. In *Proceedings Volume 10529, Organic Photonic Materials and Devices XX; 105290I*.
150. Baeg, K.-J., Binda, M., Natali, D., Caironi, M., & Noh, Y.-Y. (2013). Organic light detectors: Photodiodes and phototransistors. *Advanced Materials*, 25(31), 4267–4295.
151. Qian, C., Qian, C., Sun, J., Kong, L.-A., Gou, G., Zhu, M., Yuan, Y., Huang, H., Gao, Y., & Yang, J. (2017). Organic phototransistors: High-performance organic heterojunction phototransistors based on highly ordered copper phthalocyanine/para-sexiphenyl thin films. *Advanced Functional Materials*, 27(6), 1604933.
152. Milvich, J., Zaki, T., Aghamohammadi, M., Rödel, R., Kraft, U., Klauk, H., & Burghartz, J. N. (2015). Flexible low-voltage organic phototransistors based on air-stable dinaphtho[2,3-b:2',3'-f]thieno[3,2-b]thiophene (DNFTT). *Organic Electronics*, 20, 63–68.
153. Hayter, C. S., Rasmussen, E., & Rooksby, J. H. (2018). Beyond formal university technology transfer: Innovative pathways for knowledge exchange. *The Journal of Technology Transfer*, 1–8.
154. Jessop, P. G., & Reyes, L. M. (2018). GreenCentre Canada: An experimental model for green chemistry commercialization. *Physical Sciences Reviews*, 3(6), 20170189.

Chapter 2

Bio-Inspired, Real-Time Passive Vision for Mobile Robots



Piotr Skrzypczyński, Marta Rostkowska, and Marek Wąsik

Acronyms

CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
FPS	Frames Per Second
GPGPU	General Purpose Graphics Processing Unit
HSV	Hue-Saturation-Value (color model)
MIPI	Mobile Industry Processor Interface
ORB	Oriented FAST and Rotated BRIEF
QR-code	Quick Response matrix code
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
TFLOPS	Tera Floating Point Operations Per Second
USB	Universal Serial Bus
VFH	Vector Field Histogram

2.1 Introduction

We are witnessing how robots proliferate to everyday life, and the number of commercially available mobile robots increases gradually. Mobile robots perform tasks like surveillance, cleaning or they assist handicapped people. However, to have

P. Skrzypczyński (✉) · M. Rostkowska · M. Wąsik
Institute of Control, Robotics, and Information Engineering, Poznań University of Technology,
Poznań, Poland
e-mail: piotr.skrzypczynski@put.poznan.pl; marta.a.rostkowska@doctorate.put.poznan.pl;
marek.s.wasik@doctorate.put.poznan.pl

a satisfying level of autonomy these robots need to reliably perceive objects and events in their environment. At the same time robots have to be affordable and easy to maintain. Hence, the design of sensors adequate for the navigation-related tasks becomes important.

Nowadays cameras are considered the most compact and affordable exteroceptive sensors in robotics. Passive vision captures a large amount of data, reflecting both the photometric and geometric properties of the observed scene, but requires considerable computing power, and has a number of limitations related to the used sensors. A monocular camera has a limited field of view, and gives only an angle to the observed feature/landmark, but no range information. Cameras on a stereo rig can measure depth in unknown scenes, but their field of view is also limited.

However, natural evolution has developed visual perception systems that perfectly fit to the needs and environment niches of particular species of animals. Some of them are incredible, like the visual sense of flying insects [37]. These insects have a wide-field view and complex eyes, which allow them to navigate efficiently. Similarly, some mobile robots use omnidirectional cameras, which perceive whole surroundings from a single view [33]. Such cameras ensure that the robot gathers necessary knowledge about the environment in reasonable time. Regrettably, it is not easy to calculate robot's or objects' position employing only data from omnidirectional camera. Visual perception that has developed in more complex animals consists of peripheral and foveal vision. The brain of an animal can provide a correct interpretation of the environment employing cues from both systems. In general, it is possible, because peripheral vision cues are pursued by the eye fixation. However, accurate perception of distances requires foveal analysis, involving central vision. Eventually, two or more views of the scene are required to produce 3D location of unknown objects, which in animals is possible owing to binocular vision.

Following the most efficient biological vision examples, we decided to combine omnidirectional and peripheral/foveal vision mechanisms in our construction. In this way, we delivered a system which combines advantages of both camera types: 360° field of view and accurate environment's data (robot's and objects' position). We created a vision sensor having a hybrid field of view through combination of a camera looking upward into a curved mirror, and a typical perspective camera mounted on top of this mirror (Fig. 2.1). This sensor was presented for the first time in [21], while the obstacle detection algorithm was developed separately, using only the omnidirectional camera [36]. In this chapter we present in a unified way the peripheral vision part, and the algorithms for distance measurement and obstacle detection, that are related to foveal vision. Moreover, the new version of the sensor is presented, which has the perspective camera mounted on a servo. Owing to this design, the perspective camera can be rotated horizontally, which allowed us to create new functions of the sensor. Thus, we describe object tracking that in turn makes it possible to actively select the field of view for the perspective camera, resembling the natural eye fixation mechanism. In our sensor real-time image processing is ensured by a Nvidia Jetson embedded computer. The first prototype was based on

Fig. 2.1 Hybrid field of view passive camera with an actuated central vision sensor

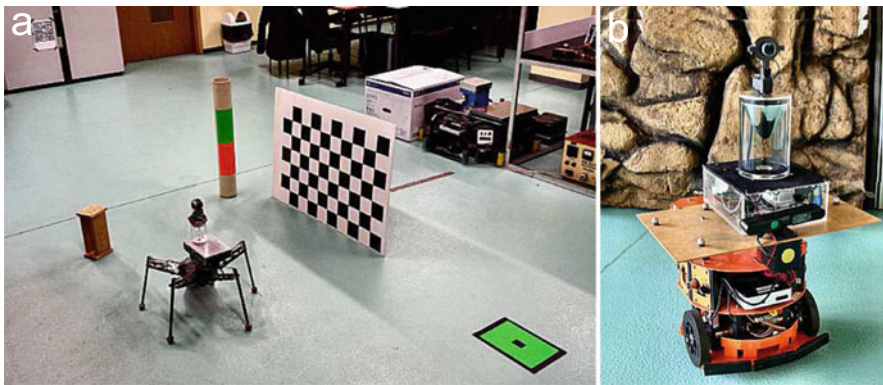
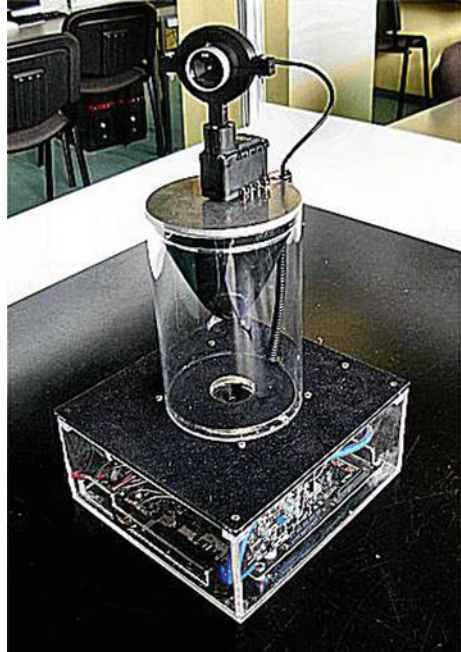


Fig. 2.2 Mobile robots with the self-contained vision sensor: the first prototype on a small walking robot (a), and the improved version on a wheeled robot (b)

the TK-1 board [12, 21], while the improved version uses the more recent Jetson TX-1. Nvidia Jetson computers are energy-efficient, compact, and powerful, making our sensor a self-contained perception unit. Such a design suits small, low-cost, and resource-constrained mobile robots (Fig. 2.2).

2.2 Related Work

Literature brings different examples of visual perception systems inspired by Nature. The system presented in [30] creates a global description of the scene, and calculates a coarse localization of the mobile robot. Then the results are improved based on the extraction of local features from images. This approach takes inspirations from the peripheral/foveal vision scheme, but still relies on a single camera for perception, and does not control the field of view. A different foveation implementation is presented in [34]. This work introduces an active vision system for an anthropomorphic robot with a pair of cameras located in each of its eyes. This design ensures a wide field of view, and the observed objects are always maintained in the foveal vision area. Santos-Victor et al. [24] describe a robot navigation system based on a wide field of view and simple processing of the optical information, which was directly inspired by insect vision.

Omnidirectional cameras are popular in various applications for two decades [38]. They are used for relatively simple robot navigation [39], and more recently, are combined into multi-camera systems, for example to obtain omnidirectional stereo images [23]. The concept of combining an omnidirectional and perspective camera is widely used in the soccer robots. However, in the soccer domain well-defined visual markers are available in the playing field [15], which are not present in other application areas.

Also the problem of stereo-based distance measurements using a hybrid system of cameras was analyzed in the context of mid-size league robotic soccer [11]. It was demonstrated that stereo-based computation of objects' positions often results in highly uncertain measurements if the cameras are poorly calibrated. Hence, a simple object localization scheme was proposed in [11], combining the bearing information from the catadioptric, and the distance to a known object (e.g. a ball used in soccer) from the perspective camera. In [20], we proposed a similar system, which solves the real-time localization task on a small mobile robot. Our system uses QR-code landmarks located on objects to simplify the localization task. Those examples demonstrate that cooperation between cameras of totally different field of view may be beneficial to various robotic tasks.

In the robotic literature only a few works tackle the problem of integrating the omnidirectional and perspective cameras in a more tight and direct manner than it is done in typical soccer robots. Cagnoni et al. [6] present a hybrid omnidirectional pin-hole sensor, but they focus only on the sensor description and calibration procedure. A system which supports obstacle detection for mobile robots is shown in [1]. Our hybrid solution presented in this chapter is conceptually similar to the designs discussed in [6] and [13], but in contrary to our system, the solutions from [6] and [13] require external processing of the images. Such an approach renders real-time processing of the omnidirectional images almost impossible. Hence, these sensors hardly can be applied in mobile robot navigation, which requires real-time response to various visual stimuli. We have applied the first prototype of our sensor on a small legged robot, which does not have enough on-board computing power to build

an environment representation in real time [36]. Although omnidirectional cameras have been already used on few walking robots [19], our application demonstrates gains due to the use of a vision sensor with on-board processing. Although the first version of our peripheral vision software has been already described in our recent conference paper [22], this chapter not only provides a more detailed description, but introduces also an object tracking module, based on particle filtering. This module is a good example of the on-board processing power in our sensor.

2.3 Hardware of the Sensor

Our proposal of the new passive vision sensor consists of three subsystems. The first subsystem is a single-board computer hosting other components and providing the on-board computation resources. The second part is an omnidirectional subsystem consisting of an upward-looking camera with a properly curved mirror. The camera and its mirror are combined by a transparent tube. The last subsystem is a standard perspective camera with the USB interface that is rotated by a servo. Two prototypes of the sensor have been constructed, which share the general design, but differ in the components being used. In both sensors a hyperbolic mirror machined from aluminum alloy and then polished manually is used. The shape of the mirrors is chosen in a way that ensures the single effective viewpoint imaging geometry [2]. With this geometry every pixel in the acquired image receives the light passing through the common point in one particular direction, which is required to produce geometrically correct images.

The omnidirectional subsystems in both variants of the sensor use cameras dedicated to the Jetson single-board computer, and equipped with the CSI-2 MIPI interface. In the first prototype it was E-Cam130 CUTK1 manufactured by E-Con, which yields 1920×1080 images at the frame rate of 30 FPS. The second prototype has the Leopard Imaging LI-IMX274-MIPI-CS camera, with the resolution of 3864×2196 pixels. In both cameras the resolution and frame rate can be changed by software, but we have chosen the parameters that best suit the application, and are a trade-off between the high resolution, high frame rate, and compatibility with the available software.

The most different component in the first and the second prototype is the perspective camera subsystem. In the older version, a simple webcam is fixed to the top of the mirror with a printed plastic part. In the upgraded design, the camera is attached to the small MX12-W servo from Robotis [4]. Thus, the perspective camera can rotate horizontally. The flexible USB cable makes it possible to cover full 360° field of view, but the camera cannot rotate $n \times 360^\circ$. The first version of the design uses a Logitech 500 webcam with the resolution 1280×1024 pixels, while the improved one is equipped with a more compact Microsoft Lifecam, having the resolution of 1280×720 pixels.

The most important aspect of our sensor design is the use of a modern single-board computer as the base. The first prototype uses Nvidia Jetson TK-1 with the

Tegra K1, for which the main computing power is provided by Kepler architecture graphics cores with compute unified device architecture (CUDA) support. Jetson TK-1 achieves 300 GFLOP/s in the single-precision mode. The second version is based on the more recent Jetson TX-1, which has the GPGPU with 256 cores of the improved Maxwell architecture. The improvements allow the TX-1 version to produce the computing power of about one TFLOP/s. However, a drawback of the standard TX-1 model is its increased footprint, which required to increase the size of the whole sensor unit. Both models of the Jetson board facilitate the CSI-2 MIPI interface for cameras allowing direct data transfer between the camera and the GPGPU. This interface is used, however, only for the omnidirectional cameras. The perspective camera is connected via typical USB interface.

2.4 Basic Software and Calibration

The developed sensor requires proper calibration of its components, as well as parameters calibration for the stereo pair that consists of the perspective camera and the virtual camera created by software from the omnidirectional image. All extrinsic parameters (rotations and translations) are estimated with respect to the coordinate frame of the catadioptric camera, which is considered the reference frame of the whole sensor.

Calibration for a hybrid system is much more complicated than in a standard stereo vision systems, because of the geometric distortions in the omnidirectional images. Before extracting metric information from images acquired by the omnidirectional part, the geometric distortions must be removed. Calibration of a stereo pair, where one of the cameras is perspective, and the other one is omnidirectional, is described in [6]. In this method, both cameras observe calibration patterns on several parallel surfaces of known relative positions. This assumption is a drawback of the method, because it makes implementation complicated and vulnerable to errors due to inaccurate location of the calibration surfaces. Hence, we decided to use a simpler method based on the existing open-source calibration tools, which are well-documented and commonly used in vision research. In our method, the omnidirectional camera and the perspective camera are at first calibrated independently, and a virtual camera view is defined from the omnidirectional image. Then, calibration patterns are shown to both cameras (perspective and virtual) in different positions and angles. When the required number of images is collected, these two cameras are calibrated as a standard stereo pair.

The calibration procedures for both subsystems and the stereo pair are highly automated, and do not involve any external equipment other than a simple 63-field chessboard pattern. We have used the same pattern in all the calibration procedures, and the same size of the calibration database (20 images).

2.4.1 Calibration of the Subsystems

Calibration of the perspective camera can be accomplished by using the popular Matlab toolbox [5] or the OpenCV library procedures [16]. In contrast, the known calibration methods for omnidirectional vision systems are often particular to the camera type [3], or require an accurate specification of the mirror geometry and additional equipment to perform the calibration procedure [10]. Among the omnidirectional camera calibration methods known from literature the one proposed by Scaramuzza et al. [26] appears to be the most universal and practical one, as it uses only a standard chessboard pattern, and does not assume any particular mirror or camera type. This method is implemented in the OCamCalib toolbox, which we have used for the presented sensor.

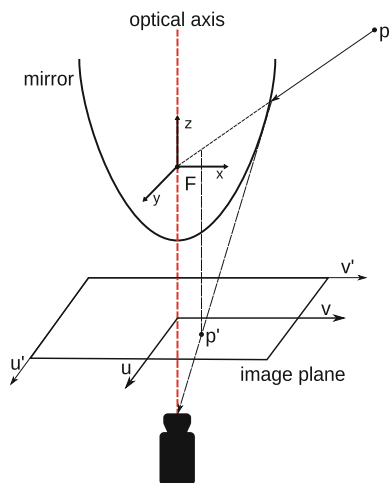
The calibration process starts with determining the model of the perspective camera. The model of distortions with five coefficients, which describe the radial and tangential distortions of the image is used. The perspective camera matrix is calculated from formula:

$$\mathbf{K}_p = \begin{bmatrix} f_{c1} & \alpha_c f_{c1} & c_{c1} \\ 0 & f_{c2} & c_{c2} \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

where f_{c1} is the horizontal and f_{c2} is the vertical focal length, c_{c1} and c_{c2} define the center of the image, and α_c is the pixel skew coefficient.

As already mentioned, the omnidirectional camera is calibrated employing the approach and camera model proposed by Scaramuzza [25]. The geometric model shown in Fig. 2.3 is represented by the formula:

Fig. 2.3 Geometric model of the catadioptric camera



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix}, \quad (2.2)$$

where u and v represent projection of a 3D point \mathbf{p} into the perfect (i.e. undistorted) image, x , y , and z are image coordinates of this point on the mirror surface, while ρ is the distance between the projected point \mathbf{p}' and the undistorted image's center.

To solve the Scaramuzza equations and define the camera model, it is necessary to calculate $z = f(\rho)$, defined as a fourth order polynomial $z = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4$. In order to receive an optimal solution, the coefficients are computed iteratively and the reprojection errors are observed. We use the experimental procedure described in [26]. When preparing the calibration data it is of great importance to cover the whole field of view of the camera by the chessboard patterns, because the calibration data are used also to compensate any misalignments existing between the mirror and the center of the camera. The calibration process has two stages. First, the center of the omnidirectional image $\mathbf{o}_c = [u_c, v_c]^T$ and the affine matrix $\mathbf{A}_{(2 \times 2)}$ are calculated. The affine matrix $\mathbf{A}_{(2 \times 2)}$ determines the relation between (u, v) coordinates of the idealized image and the actual image coordinates (u', v') . Eventually, the calibration results are refined applying the iterative Levenberg-Marquardt non-linear optimization technique.

2.4.2 Panoramic Images

Typically, a view of the environment, which is seen in a picture from the catadioptric camera is highly distorted. While objects can be detected and roughly localized or tracked using raw omnidirectional images, it is not possible to calculate accurate positions of these objects or point features. For the calculation of the accurate distances and geometric relations a rectified (i.e. geometrically corrected) 360° panoramic image is necessary. With such images, the sensor can not only detect obstacles, but also measure accurate distances to objects in the wide field of view.

Scaramuzza [25] presents a simple method of the image rectification based on geometric inverse projection and the calibrated model of the catadioptric camera. Based on geometry and dimensions of the images (Fig. 2.4), the omnidirectional image pixel coordinates (u, v) are calculated:

$$u = \frac{2\pi v_p R_{\max}}{h} \cos\left(\frac{2\pi u_p}{w}\right), \quad v = \frac{2\pi v_p R_{\max}}{h} \sin\left(\frac{2\pi u_p}{w}\right), \quad (2.3)$$

where h is the height of the panorama, and w is the width of the panorama, R_{\max} denotes the radius of the omnidirectional image's outer circle, while (u_p, v_p) are respective pixels of the reconstructed panoramic image.

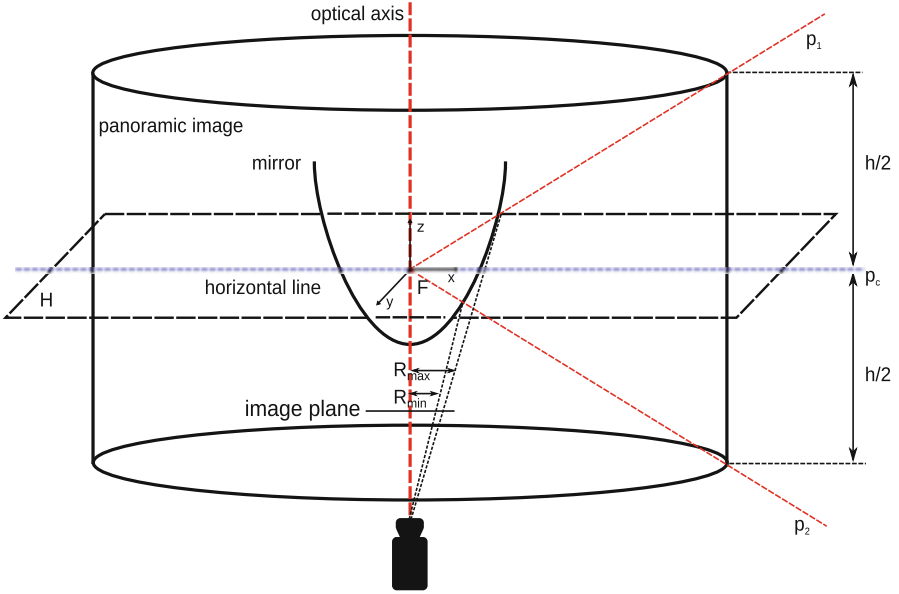


Fig. 2.4 Geometry of the panoramic image surrounding the catadioptric camera

However, the formulas (2.3) do not consider the calibrated parameters of the catadioptric camera. Because of that the panoramic image is not compatible with the field of view of the perspective camera image. To achieve this compatibility, our rectification algorithm has to locate the horizontal line (\mathbf{p}_c point in Fig. 2.4) of the panoramic image. The correctly located horizontal line should be on the same elevation as the optical center of the curved mirror. In practice, it means that pixels taken from the central row of the panoramic image should have zero z coordinates. The half-lines \mathbf{p}_1 and \mathbf{p}_2 go through the upper and the lower rim of the cylinder, respectively. For further processing, especially for creation of stereo-pair, it is very important that the height h of the cylinder (in pixels) equals the height resolution of the perspective camera image. Next, all pixels (i.e. their coordinates) from the panorama’s cylindrical surface are re-projected back into the undistorted omnidirectional image. To achieve this, a method presented in (2.2) is used, which is based on the inverse mapping. The last step in the corrected procedure for panoramic image creation is calculation of the pixel coordinates in the omnidirectional image by formula:

$$u = \rho_v(v_p) \cos\left(\frac{2\pi u_p}{w}\right), \quad v = \rho_v(v_p) \sin\left(\frac{2\pi u_p}{w}\right). \quad (2.4)$$

where $\rho_v = f(v_p)$ denoted the distance between the point’s projection and the center of the omnidirectional image. This parameter is computed separately for each row of the panorama. The coordinates u and v are considered in the range between the R_{min} and R_{max} radius. The minimal radius R_{min} is determined by the blind area in the omnidirectional picture.



Fig. 2.5 Rectified panoramic images: constructed using only the inverse projection (a), and constructed using our improved method (b)

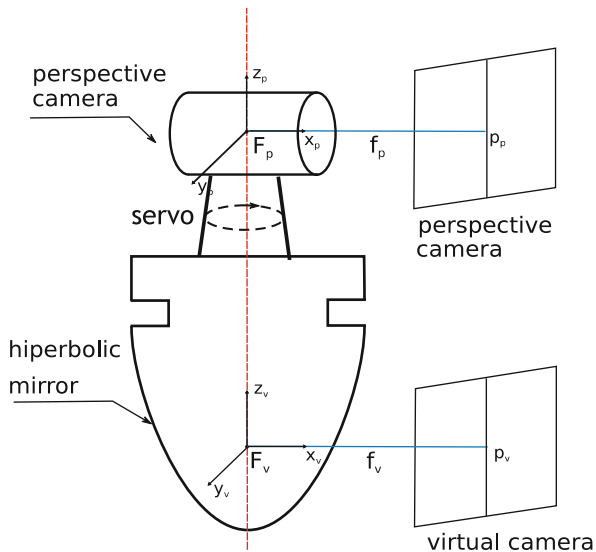
Correct image rectification is a time-consuming process. Hence, to do all calculation in real-time, the OpenCV function `remap()` is used, which is supported by CUDA on the Jetson platform. In this parallelized version the reconstruction of a panoramic image takes only 0.85 ms. Example results of panoramic image reconstruction are shown in Fig. 2.5. The image constructed considering the calibrated parameters of the system and the correct horizontal line location (Fig. 2.5b) looks more natural, and the relations between the height of particular objects seen in the image are preserved, unlike in the simple method (Fig. 2.5a).

2.4.3 Virtual Camera

To calculate the distance between the sensor and an object of unknown size, it is necessary to have two views of this object, which are related by known extrinsic parameters, i.e. the rotation and translation between the cameras that produced these views. To accomplish this task we define a virtual camera, which provides the field of view similar to the actual perspective camera of the hybrid sensor. The coordinate system of the virtual camera has the origin located in the optical center of the curved mirror (Fig. 2.6). The focal length and resolution are chosen purposefully to yield images that are geometrically similar to the perspective ones. While a similar idea was used in [13], our virtual camera image is defined directly from the panoramic image constructed in real-time, which makes the computations much faster. We take advantage from the fact that the panoramic images are reconstructed in real-time in our sensor, and they make a perspective-like view of the scene readily available in the 360° field of view. Thus, we only need to create from the panorama a virtual image that is geometrically compatible with the actual perspective image. The virtual image is created by projecting a ray from the center of the curved mirror towards the cylindrical surface. This ray determines the area on panoramic image defined by the requested resolution of the virtual camera (compatible with the physical one). Pixels from a proper area defined on the panoramic image are taken as a representation of pixels in the virtual camera image.

The calibration procedure for the virtual camera is the same as the one we use for the physical perspective camera. We used the same toolbox [5] to get the camera matrix \mathbf{K}_v .

Fig. 2.6 Geometric relations between our sensor and the virtual camera image



2.4.4 Calibration Between the Subsystems

To know the geometric relations between the two subsystems necessary for the stereo distance measurements we perform extrinsic calibration between the perspective and the virtual camera. The results are extrinsic parameters of the stereo pair. We treat the perspective camera and the virtual camera as a stereo pair, unlike [6], where the perspective and the omnidirectional camera are calibrated together. Our approach avoids the use of any special calibration equipment, and allows us to use the standard calibration software. Therefore, having defined the virtual camera, we assume that we have a pair of cameras, which are properly calibrated in terms of their intrinsic parameters, as the \mathbf{K}_p , \mathbf{K}_v matrices and distortion models are already computed. The next step of the calibration is calculation of the extrinsic parameters of the stereo pair. To accomplish this, we use again the camera calibration toolbox [5]. We take a new series of images, which contains the chessboard pattern seen in many different positions over the common field of view of both cameras. The transformation of the coordinate system between the perspective and the virtual camera is computed from the corresponding points of the calibration pattern. This relation is described by the rotation matrix $\mathbf{R}_s(3 \times 3)$ and the translation vector $\mathbf{t}_s = [t_x, t_y, t_z]^T$. The last step of the hybrid vision sensor calibration is calculation of the essential matrix [9]. The essential matrix is computed based on rotation and translation between the images from both cameras:

$$\mathbf{E} = \mathbf{R}_s \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \quad (2.5)$$

2.5 Peripheral Vision in the Hybrid Sensor

The concept of a stand-alone passive vision sensor comes from the observation, that some mobile robots (e.g. small walking machines) cannot allocate enough computing power to the perception and environment modelling tasks. Therefore, they may benefit from a sensor that provides the robot with pre-processed navigation cues, such as location of obstacles or direction of collision-free motion. The software described so far in this chapter converts raw frames from the omnidirectional camera into geometrically correct panoramic and virtual camera images, but does not support navigation directly. Hence, this section describes how the omnidirectional images can be used to support selected navigation-related functions of the robot, providing the machine with a rough analogy of the peripheral vision in animals. Our peripheral vision enables the robot to detect obstacles and moving objects, track objects, and focus the perspective camera on a selected object. This last function demonstrates benefits from the cooperation between the peripheral and central vision capabilities in our hybrid sensor. Although the algorithms behind these functions are simple in general, we use them to demonstrate that our sensor is an efficient platform to implement various image processing algorithms, also these that benefit a lot from parallel processing on a GPGPU.

2.5.1 *Detection of Objects*

Rapid detection of changes occurring in the environment is crucial to animals, as it is related to their predatory behaviors or the ability to avoid other predators and various natural hazards. Also mobile robots may benefit from the ability to quickly detect changes in the observed scene. Therefore, the main peripheral vision function implemented in our sensor is detection of moving objects in the omnidirectional images. Moving objects are segmented from the background in real-time using the Background Subtraction Library (BGSLibrary) [32]. This function supports human–robot interaction (e.g. detection of people approaching the robot), surveillance applications, and multi-robot systems, where quick detection of obstacles and other robots is required [14, 28]. The BGSLibrary offers an easy-to-use software framework integrated with OpenCV. It makes possible to discriminate moving objects from the background, providing that the camera is static while acquiring a pair of images. The library contains implementation of several algorithms that support different tasks, such as video analysis. From these

algorithms, we chose two techniques of relatively low computational complexity, namely the `FrameDifferenceBGS` and `SigmaDeltaBGS`.

`FrameDifferenceBGS` is very simple, as it only compares the query image to the one acquired in the previous time instance, and then extracts the moving objects by marking areas where the difference of color is larger than a given threshold. `SigmaDeltaBGS` algorithm is somewhat more complicated, as it attempts to estimate parameters of the observed background, which is assumed to be approximately uniform (e.g. a flat floor). This algorithm produces fewer artifacts than the simple frame difference, but is slightly more computationally demanding. By default the `BGSLibrary` is running on a CPU, and it is not compatible with CUDA, hence it cannot benefit from a GPGPU. Because in the robotic applications real-time processing is a must, we adapted the used `BGSLibrary` algorithms to use a version of `OpenCV` that is supported by CUDA. Eventually, we were able to exploit the GPGPU readily available in the Jetson platform.

2.5.2 *Tracking of Objects*

Some of the moving objects that could be extracted from background by our change detection functions may be important enough to be tracked for longer time, e.g. to determine their speed and trajectory. To track an object the sensor has to determine some of its perceptual properties, to make this object distinguishable among others. The simplest property that can be easily distinguished is color. An implementation of this concept on the hybrid field of view sensor was presented in [22]. For the sake of speed, detection of objects having a specific, user-defined color was implemented on the raw omnidirectional images applying thresholding in the Hue-Saturation-Value (HSV) color space. The location of an area of the defined color is then converted to the polar coordinate system, with the origin in the center of rotation of the moving camera servo. This gives the perspective camera its reference angle, which is compared to the current angle of rotation from the camera servo. The computed rotation angle is the one that brings the camera to the target heading in the shortest time.

Because the simple tracking procedure can be applied only to bright-colored objects, its practical use on a mobile robot is limited. Therefore, we implemented on the hybrid field of view sensor also a more advanced tracker employing optical flow and particle filter for the tracking procedure. This function makes it possible to track previously unknown objects having arbitrary shapes and colors, as long as they move at a reasonable speed and stand out visually from the background (Fig. 2.7). The new tracker is based on the algorithm presented in [29] with some improvements. The computation of optical flow is implemented as a parallelized version of the Farnebäck algorithm [7], which quickly yields a vector field with the detected velocity vectors of pixels that have moved between two consecutive frames. Then, a filter with 60 particles is initialized around the target object, which has to be designated by the user with a bounding box. As in the original algorithm [29] the particles are described

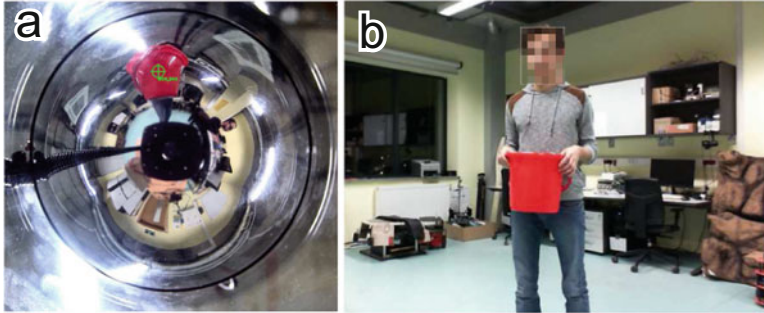


Fig. 2.7 Simple, color-based object tracking: an object (red bucket) detected object in the raw image (a), a perspective camera image taken after focusing on this object's location (b)

in polar coordinates that are natural for an omnidirectional image. The weights of particles are computed taking into account differences between the velocity vectors of the particles (i.e. image pixels where the particles are located) and the velocity vector of the target, which is known from the previous iteration. Moreover, the similarity of the Hue component of the HSV color model is considered when computing the weights. The resampling step draws a new set of particles from the weighted ones, favoring particles with higher weights, which replace those of lowest weights. As a result, the filter converges in few iterations and the best particles track the moving target.

2.5.3 Avoiding Obstacles

Obstacle avoidance is an essential function in every mobile robot. A robot has to detect any objects that may pose a danger when it is attempting to move towards the given goal. Obstacles may be detected by range or visual sensors, but the avoidance task becomes more efficient if the robot perceives the objects that surround its body without a need to rotate. Therefore, peripheral vision provided by the omnidirectional part of our sensor camera is particularly suitable to indicate the presence of obstacles around the robot in real-time. This concept has been implemented on a compact, low-cost legged robot [36], which got equipped with the first prototype of our vision sensor.

The obstacle detection and avoidance method is inspired by the popular vector field histogram (VFH) algorithm [35]. This algorithm can be directly used for sensors which measure the distances between the robot and the surrounding objects, for example sonars or 2D laser scanners. Based on this data, a local map of the local environment is created. However, our version of the algorithm works using only data from the omnidirectional part of the hybrid sensor. All calculations should be performed in real-time, so the robot can pursue the obstacle avoidance task. Potential

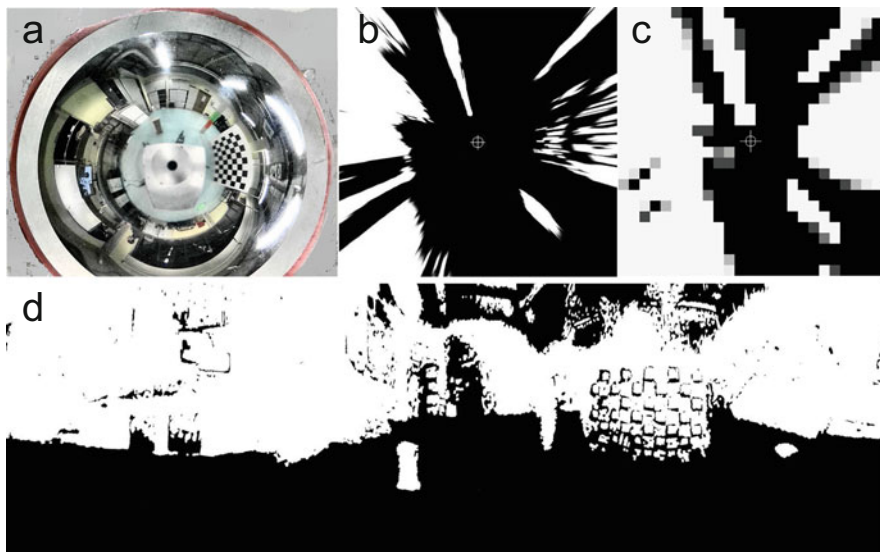


Fig. 2.8 Construction of the occupancy grid from the omnidirectional images: raw image (a), data projected to a horizontal 2D image (b), final occupancy grid (c), and the binary panoramic image after filtration (d)

obstacles are recognized on the panoramas rectified in real-time, through background removal. Next, rough distances to them are calculated directly from these images to create an occupancy grid representation of the surroundings. This is possible, because in the panorama objects that are closer to the sensor are located in the lower part of the image, while the upper part depicts more distant objects.

Also because of speed requirements, the panoramic images are constructed in a slightly different way than in the general case (see Sect. 2.4.2). Namely, the background is removed from raw omnidirectional pictures (Fig. 2.8a) using the color information, and only these pixels that represent obstacles are transferred to the panorama. Using a defined color in the HSV color space is a very background removal fast method, and it does not need a good estimate of the robot's ego-motion, which is unavailable in a legged robot. On the drawbacks side, we have to assume that the background is of approximately uniform appearance. Although combining this function with the BGSLibrary functions is possible, we have found that the simple method works well indoors, while it is much faster and easier to parallelize using CUDA. On the legged robot, the areas that can show the body or the swinging legs of the machine are masked by proper shapes directly on the omnidirectional image [36]. The omnidirectional images with masked background and the robot's element that may be treated as close objects are then binarized and converted to panoramic images (Fig. 2.8d). The binary panoramas are treated with erosion and then dilation morphological operators in order to eliminate small, isolated pixel blobs. The next step is to fill the 2D local occupancy grid of the environment with the information

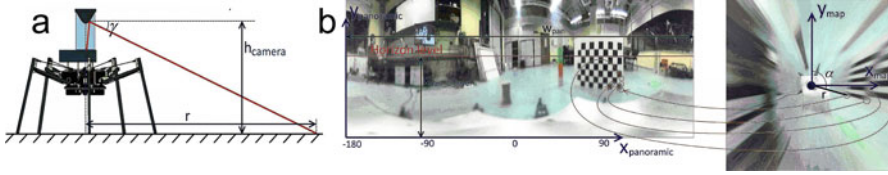


Fig. 2.9 Configuration of our sensor on the walking robot (a), panoramic image data projected into the 2D local environment representation (b)

from the prepared images. In the pictures obstacles are marked as white areas, and free areas are marked black. The panoramic image coordinates that have to be used to find information about the occupancy of the cells in the grid are defined as:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \frac{\alpha w_{\text{pan}}}{360^\circ} \\ x_{\text{pan}0} - g_{\text{pan}} \gamma \end{bmatrix}, \quad (2.6)$$

where: w_{pan} is the width of the panoramic image (pixels), g_{pan} defines the vertical resolution of the panoramic image (pixels/ 1°), and $x_{\text{pan}0}$ defines the elevation of the horizon line (Fig. 2.9a). The angles α and γ are given by formulas:

$$\alpha = \arctan\left(\frac{x_{\text{map}}}{y_{\text{map}}}\right), \quad \gamma = \arctan\left(\frac{h_{\text{camera}}}{\sqrt{x_{\text{map}}^2 + y_{\text{map}}^2}}\right), \quad (2.7)$$

where x_{map} and y_{map} are coordinates of the occupancy grid, and h_{camera} is the elevation of the sensor center measured from the floor. This parameter yielded by the robot's controller, because in a legged machine it depends on the legs configuration (Fig. 2.9b).

The local grid map that is a robot-centric representation of the environment accumulates the occupancy information extracted from the prepared panoramic images. Using this intermediate representation makes it possible to accumulate the information related to obstacles in short time windows, and avoids the necessity to compute the control commands using highly uncertain data. The coordinates of the pixels that represent obstacles are transformed into an instantaneous and local horizontal image—a form of simple map (Fig. 2.8b). Next, the coordinates of obstacles are employed to update the robot-centered occupancy grid. The occupancy grid is attached in the origin of the vision sensor coordinate frame, and its size is 5 m \times 5 m, with 0.2 m \times 0.2 m cells (Fig. 2.8c). Following the idea of original VFH we increase the cell occupancy by a fixed value (in the experiments the value of 3 was applied) whenever a pixel representing an occupied area (i.e. obstacle) is transferred into this particular cell. If the transferred pixel represents an empty area, the cell occupancy is decreased by a smaller value (value of 1 was used). However, the occupancy counter is bounded for each cell: it cannot exceed 25 or drop below zero. Then, the one-dimensional polar histogram is built upon this occupancy grid, exactly

as in the original VFH. This histogram is also attached in the origin of the sensor coordinate frame (coincides with the center of the robot), which makes it possible to select the steering direction that avoids all obstacles, but is the one closest to the direction to the goal.

2.6 Central Vision in the Hybrid Sensor

The central vision in animals and human beings serves mostly the needs of accurate interaction with particular objects in the environment, supporting, e.g. grasping of objects. However, our sensor is dedicated to mobile robots, and the central vision serves mostly navigation tasks, such as landmark-based localization [20]. Hence, the main function is accurate measurement of distances to selected objects.

The distances are measured employing an unorthodox stereo vision setup, in which one image in the stereo pair is yielded by the perspective camera directed towards the chosen object, but the other one is synthesized from the omnidirectional image by our virtual camera. Assuming that both cameras are calibrated with respect to intrinsic parameters, we need to relate the perspective camera coordinates to the coordinate system of the virtual camera by extrinsic calibration, as described in Sect. 2.4.4. Once the extrinsic parameters are known, we compute the projection matrices for both cameras in the stereo pair. The projection matrix of the virtual camera reduces to $\mathbf{P}_v = \mathbf{K}_v[\mathbf{I}|\mathbf{0}]$, because we assume that the coordinate system of this camera is attached in the origin of the coordinate frame of the stereo pair. Then, the projection matrix of the perspective camera is calculated. This matrix accounts for the rotation and translation between the two cameras: $\mathbf{P}_p = \mathbf{K}_p[\mathbf{R}_s|\mathbf{t}_s]$. A point in the 3D scene \mathbf{p} is related by the projection matrices to its counterparts \mathbf{p}'_v and \mathbf{p}'_p in the 2D images obtained from the virtual camera, and the perspective camera, respectively:

$$\mathbf{p}'_v = \mathbf{P}_v \mathbf{p}, \quad \mathbf{p}'_p = \mathbf{P}_p \mathbf{p}. \quad (2.8)$$

Therefore, we can reconstruct the position in 3D of the point \mathbf{p} from its projections on the undistorted images from both cameras. The stereo distance computation is accomplished using the optimal triangulation method [8], which runs in real-time on the Jetson board in our sensor. It should be noticed that this method has been chosen mostly due to its computation speed advantage and simple implementation, while more recent and advanced methods exist, e.g. based on neural networks [18]. The GPGPU available in our sensor makes it possible to implement such a method in the future.

Prior to the stereo computations, we have to determine the matching point features. These features are located on the images from both cameras, but they represent the same points in 3D. The computer vision literature lists a number of methods to determine point correspondences in stereo vision [9]. Taking into account that the images produced by the virtual camera are of relatively low resolution,

because they are only up-sampled to the resolution compatible with the perspective camera, we employ the point descriptors to find the corresponding features. The descriptor vectors catch the appearance of the local neighborhood of each feature. They are commonly used to match point features in robot navigation, and are characterized by a good trade-off between the matching efficiency and the computing power requirements [27]. We have implemented three alternative feature matching procedures using SIFT, SURF, or ORB detector/descriptor pairs. The use of sparse point features yields “sparse” stereo information, as the position is computed only for a certain number of features. This is, however, acceptable for most navigation algorithms that natively employ sparse feature maps [31].

Point features are detected in both images, and then described using one of the detector/descriptor pairs. The coordinates of the feature points are undistorted and normalized. Then, we attempt to match points from both images minimizing the distances (Euclidean in the case of SIFT and SURF, and Hamming in the case of ORB) between the descriptor vectors associated with these points. Once the initial associations are established, the symmetrical reprojection error is calculated:

$$e_{\text{rep}} = \max\{d(e_j, (u_i^v, v_i^v)), d(e_i, (u_j^p, v_j^p))\}, \quad (2.9)$$

where u_i^v and v_i^v denote the normalized coordinates of the i -th \mathbf{p}'_v point, u_j^p and v_j^p are coordinates of the j -th \mathbf{p}'_p point, while the Euclidean distance of a point y to the line x is denoted by $d(x, y)$, and e_i, e_j are epipolar lines computed from the essential matrix (2.5):

$$\begin{aligned} [e_{i_x}, e_{i_y}, e_{i_z}]^T &= \mathbf{E}[(u_i^v, v_i^v, 1)]^T, \\ [e_{j_x}, e_{j_y}, e_{j_z}] &= [(u_j^p, v_j^p, 1)]\mathbf{E}. \end{aligned} \quad (2.10)$$

If the error e_{rep} (2.9) is smaller than a fixed threshold, the match gets accepted. The paired features are then used to calculate the distances in the 3D scene.

2.7 Experimental Results

2.7.1 Peripheral Vision

Peripheral vision functions have been tested in the tasks of obstacle avoidance, detection of dynamic objects, and tracking of both specific color and arbitrary shape/color objects. Some of these experiments are documented on the accompanying video material.¹

¹<http://lrm.cie.put.poznan.pl/bioinspsens.wmv>.

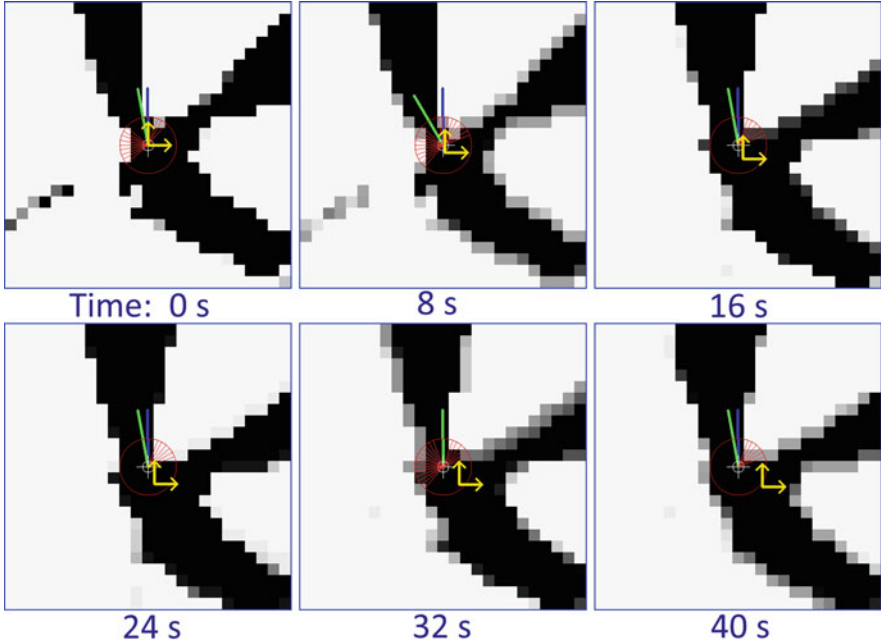


Fig. 2.10 Visualization of the grid maps produced in our sensor while computing a sequence of steering directions to the goal

The more compact version of the sensor, based on the Jetson TK-1 board was tested on a legged robot. The robot walked on a flat floor in a lab, avoiding different types of obstacles, including specially prepared cardboard boxes and tubes, as well as the usual lab equipment. Robot-centered grid maps updated in sequence during this experiment is depicted in Fig. 2.10. The direction to the goal defined by the human operator is shown in Fig. 2.10 by blue arrows, green arrows depict the steering direction obtained from the VFH algorithm, and distances used in the polar histogram to detect obstacles are denoted by red circles. An important improvement in the processing speed has been achieved for this algorithm owing to the use of parallel processing on the Jetson's GPGPU. Namely, if projection of the detected obstacles from the panoramic image to the robot-centered map was implemented on the Jetson's KT-1 CPU, this operation took 3.01 ms for a single image, but the CUDA implementation using the Kepler GPGPU required only 0.45 ms for the same operation.

Also the real-time detection of moving objects through background discrimination was tested on the Jetson TK-1 variant of the sensor. The FrameDifferenceBGS algorithm implemented on the Jetson TK-1 CPU required 19.8 ms per frame. This is enough to detect slowly moving objects, however, the embedded Jetson platform is much slower in this task than a desktop computer (PC with i7 at 2.3 GHz), which took only 8.5 ms per frame. Unfortunately, the execution time decreased only minimally, to 13.7 ms, when the FrameDifferenceBGS algorithm was re-

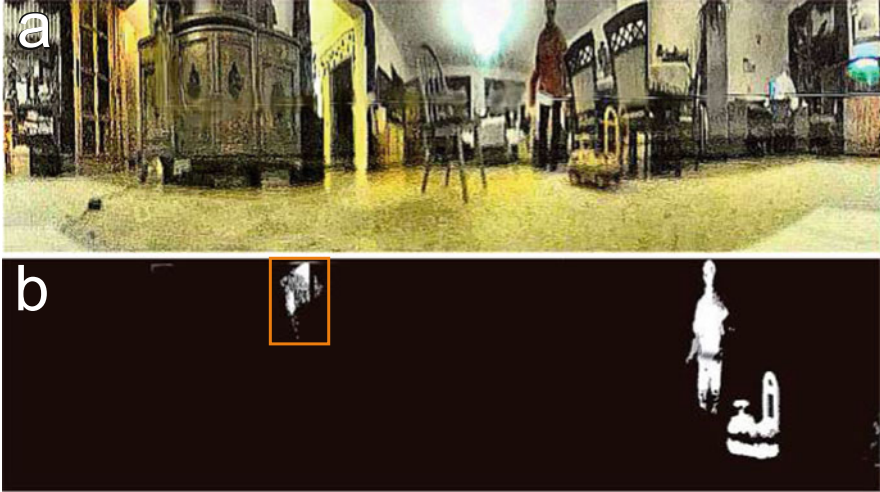


Fig. 2.11 Dynamic objects segmented from the background by the SigmaDeltaBGS method: the panoramic image (a), and the detected objects denoted by white pixels (b). The orange rectangle surrounds some wrongly identified pixels

implemented using CUDA and ran on the Jetson’s GPGPU. The reason for this result is a large number of data transfer operations in the considered algorithm, compared to the relatively simple computations. Such tasks do not benefit much from parallel processing architecture. For the more complex SigmaDeltaBGS the difference in processing speed between the Jetson TK-1 implementation and its desktop PC counterpart was smaller—processing of a single frame took 239.6 and 198.7 ms, respectively. However, in this algorithm data transfer constitutes much smaller fraction of the operations. Figure 2.11 shows example images with a person and a toy cart detected by the SigmaDeltaBGS algorithm. Notice that very few outliers are present (Fig. 2.11b).

Tracking was tested on several objects, including simple balls, toys, and people surrounding the sensor [17]. Figure 2.12a demonstrates the behavior of the particle filter: from the computed optical flow field (left) to the converging particles (shown as pink rectangles). The ability to track an object of complicated shape and color is shown in Fig. 2.12b, where a toy giraffe (pulled on a rope) is tracked by the filter. Real-time performance is achieved due to parallel implementation of both the optical flow and the particle filter on the GPGPU.

2.7.2 Central Vision

For the evaluation of the main central vision function, the stereo-based distance measurements, we performed a series of experiments in a home-like environment.

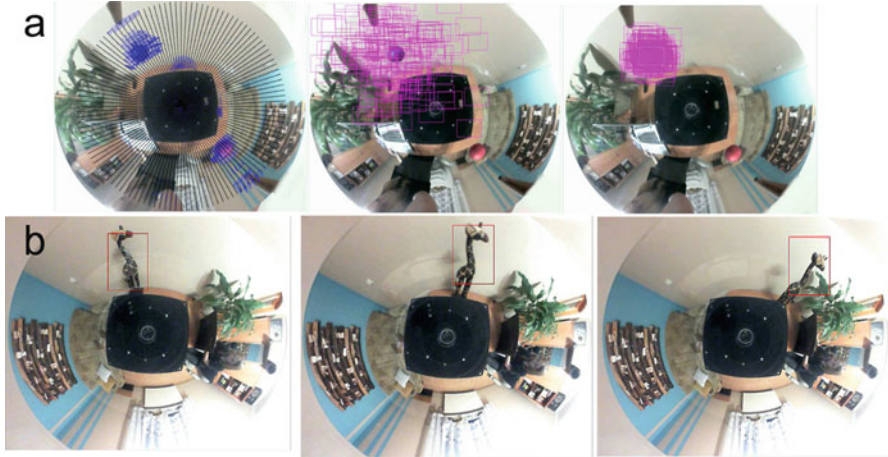


Fig. 2.12 Particle filter tracking a ball (a), and the tracked positions of a toy denoted by bounding boxes (b)



Fig. 2.13 Different images for one of the scenes: perspective camera image (a), omnidirectional camera image (b), and panoramic image (c)

At first, the distance measurements accuracy was determined. Then, we tested extensively the descriptor-based feature matching in sparse stereo to determine which detector/descriptor pair suits best the requirements of our sensor. We have set up four simple scenes, gathering furniture pieces and other common objects (boxes, pillows) into sets of two or three items (Fig. 2.13). The ground truth distances between the scene objects and the sensor were measured using a meter tape, assuming that the origin of the coordinates is located in the center of the curved mirror.

At first we evaluated the distance measurements using the SIFT detector/descriptor pair (Fig. 2.14a), because SIFT is considered the “golden standard” of the point feature descriptors if real-time performance is not a concern [20]. We measured distances to a number of features detected on the observed objects (Fig. 2.14b). The objects in the scene had flat front vertical surfaces. Hence, we averaged the distance measurements for all the features appearing on the particular vertical surface to produce the quantitative results shown in Fig. 2.15.

One can easily deduce from these plots that the range measurement errors depend on the distance to the observed object. It is also visible that these errors become minimal in the mid-range of the measured distance. This result coincides with

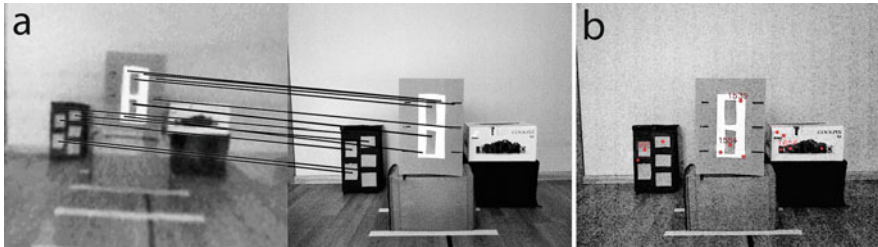


Fig. 2.14 Sparse stereo in the same scene: associated SIFT features (a), and 3D feature points located on the scene (b)

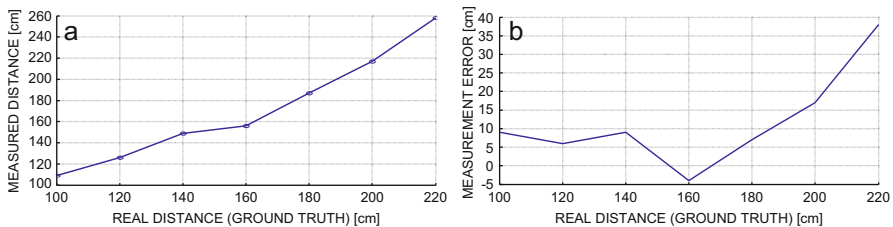


Fig. 2.15 Dependence between the range measurements and the ground truth distance (a), and the dependence between the ground truth distance and the distance measurement error (b)

the range, where the interpolation errors in the panoramic images are minimal. Apparently, the number of correct matches depends on the distance to object. The number of correctly matched SIFT features varied from 5 to 17 in the scene depicted in Fig. 2.13. The largest number of correct matches was observed at 1.8 m to the middle object. The number of matches is largest for the measured distances from 1.6 to 1.8 m, which coincides with the range of minimal distance errors. However, the number of detected features was typically higher for more distant objects.

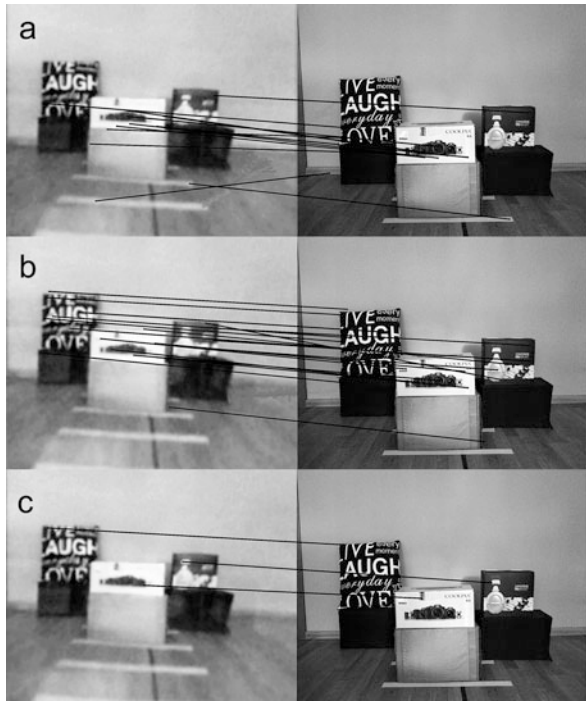
Table 2.1 shows the number of correctly matched features for the SIFT, SURF, and ORB detector/descriptor pairs, four different scenes, and three representative measurement distances. Example matches are visualized in Fig. 2.16 for the scene no. 3 at the distance of 1.4 m.

In the same experiments the time needed to process a pair of images (perspective and virtual) and to compute the distance was evaluated. The measured time (in seconds) includes creation of the virtual image, point feature detection, description and matching, and triangulation, but does not cover the reconstruction of the panoramic image. The processing time t_c shown in Table 2.2 was measured on the Cortex-A15 CPU for all the detector/descriptor pairs. For the SURF and ORB features, which have OpenCV implementations supporting CUDA, the time t_g , also included in Table 2.2, was measured on the GPGPU. Only the use of GPGPU allowed the sensor to accomplish the central vision process in real-time. The processing time depended mostly on the detector/descriptor type, however, the matching time depends also on the number of detected features.

Table 2.1 Performance of the detector/descriptor pairs in sparse stereo measured by the number of correct matches

Detector/descriptor	Scene 1	Scene 2	Scene 3	Scene 4
SIFT 1.2 m	5	22	19	33
SURF 1.2 m	7	10	13	9
ORB 1.2 m	3	0	0	3
SIFT 1.6 m	15	36	25	36
SURF 1.6 m	14	21	51	25
ORB 1.6 m	2	3	4	4
SIFT 2.0 m	16	10	24	17
SURF 2.0 m	40	18	8	3
ORB 2.0 m	2	4	1	2

Fig. 2.16 Features associated using SIFT (a), SURF (b), and ORB (c) in the Scene no. 3



2.8 Conclusions

This chapter presents a stand-alone, passive vision sensor of hybrid field of view that draws inspirations from the vision systems of insects and animals. The sensor is hosted by a recent single-board computer that provides enough computing power to implement a wide palette of image processing algorithms supporting robot navigation. Moreover, the high computing power of the sensor and its open-source software architecture, exploiting the common CUDA and OpenCV libraries, make it possible to implement new functions required by the task at hand.

Table 2.2 Total processing time for stereo-based distance measurements on the Jetson TK-1 CPU and GPGPU

Detector/ descriptor	Scene 1		Scene 2		Scene 3		Scene 4	
	t_c	t_g	t_c	t_g	t_c	t_g	t_c	t_g
SIFT 1.2 m	7.94	–	7.90	–	7.92	–	7.65	–
SURF 1.2 m	5.70	0.44	2.51	0.22	2.72	0.21	2.99	0.23
ORB 1.2 m	2.75	0.11	0.59	0.02	1.02	0.04	1.00	0.04
SIFT 1.6 m	7.64	–	7.79	–	7.62	–	7.52	–
SURF 1.6 m	2.22	0.17	2.86	0.22	2.45	0.19	2.46	0.19
ORB 1.6 m	1.75	0.07	1.50	0.06	1.27	0.05	1.23	0.04
SIFT 2.0 m	7.68	–	7.54	–	7.54	–	7.54	–
SURF 2.0 m	2.34	0.18	2.35	0.18	2.61	0.20	2.33	0.18
ORB 2.0 m	1.72	0.07	1.25	0.05	1.05	0.04	1.29	0.05

The concept of a sensor integrating an omnidirectional camera and a perspective camera is not particularly novel, but we contribute new elements:

- software that implements selected functions of the peripheral and central vision concept on top of the wide field of view vision typical to insects;
- the use of GPGPU for real-time image processing in a low-power, embedded vision system;
- simple yet efficient calibration methodology of the hybrid field of view vision system.

Moreover, this chapter contributes also improved algorithms and results of extensive experimental evaluation. For instance, the reconstruction of panoramic images has been improved to ensure better compatibility between these images and the perspective camera images. On the basis of experiments we have selected the SURF detector/descriptor pair for the sparse stereo vision function in the hybrid field of view sensor. The efficient OpenCV implementation with CUDA support ensures that SURF can be used in real-time on the Jetson platform.

Acknowledgements We would like to thank Nvidia for donating the Jetson TX-1 board within the Academic GPU Grants program. Also the involvement of students pursuing their degree in the Mobile Robots Lab, who assisted with the experiments and preparation of the video material is greatly appreciated.

References

1. Adorni, G., Bolognini, L., Cagnoni, S., & Mordonini, M. (2001). *A non-traditional omnidirectional vision system with stereo capabilities for autonomous robots, AIIA 2001: Advances in artificial intelligence*. LNCS (Vol. 2175, pp. 344–355). Berlin: Springer.
2. Baker, S., & Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2), 175–196.

3. Bakstein, H., & Pajdla, T. (2002). Panoramic mosaicing with a 180 field of view lens. In Proceeding of IEEE Workshop on Omnidirectional Vision (pp. 60–67).
4. Biadala, A., & Czukin, G. (2017). *A hybrid vision system with active field of view selection*. Engineer's degree Thesis, Poznań: Poznań University of Technology (in Polish).
5. Bouguet, J.-Y. *Camera calibration toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/
6. Cagnoni, S., Mordonini, M., & Mussi, L. (2007). Hybrid stereo sensor with omnidirectional vision capabilities: Overview and calibration procedures. In Proceeding International Conference on Image Analysis and Processing, Modena (pp. 99–104).
7. Farneback, G. (2003). Two-frame motion estimation based on polynomial expansion. In J. Bigun & T. Gustavsson (Eds.) *Image analysis, SCIA 2003*. LNCS (Vol. 2749, pp. 363–370). Heidelberg: Springer.
8. Hartley, R. I., & Sturm, P. (1995). *Triangulation, computer analysis of images and patterns*. LNCS (vol. 970, pp. 190–197). Heidelberg: Springer.
9. Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.
10. Kang, S. B. (2000). Catadioptric self-calibration. In *CVPR* (pp. 201–207).
11. Käppeler, U. -P., Höferlin, M., & Levi, P. (2010). 3D object localization via stereo vision using an omnidirectional and a perspective camera. In Proceeding of 2nd Workshop on Omnidirectional Robot Vision (pp. 7–12).
12. Kozłowski, P., & Drankiewicz, W. (2016). *Hybrid field of view vision for a mobile robot*. Engineer's degree Thesis. Poznań: Poznań University of Technology (in Polish).
13. Lin, H. -Y. & Wang, M. -L. (2014). HOPIS: Hybrid omnidirectional and perspective imaging system for mobile robots. *Sensors*, 14, 16508–16531.
14. Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., & et al. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*, 43(4), 360–369.
15. Menegatti, E., & Pagello, E. (2002). Cooperation between omnidirectional vision agents and perspective vision agents for mobile robots. In M. Gini, et al. (Eds.), *Intelligent autonomous systems* (vol. 7, pp. 231–235). Amsterdam: IOS Press.
16. OpenCV Documentation. <http://docs.opencv.org>.
17. Plucinska, N. (2018). *Object tracking on omnidirectional images*. Engineer's degree Thesis. Poznań: Poznań University of Technology (in Polish).
18. Rodríguez-Quinonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-Lopez, M., Hernández-Balbuena, D., Rascon, R., & et al. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
19. Roennau, A., Kerscher, T., Ziegenmeyer, M., Zöllner, J., & Dillmann, R. (2009). Adaptation of a six-legged walking robot to its local environment. In *Robot motion and control*. LNCIS (Vol. 396, pp. 155–164). Heidelberg: Springer.
20. Rostkowska, M., & Skrzypczyński, P. (2015). Improving self-localization efficiency in a small mobile robot by using a hybrid field of view vision system. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 9(4), 28–38.
21. Rostkowska, M., & Skrzypczyński, P. (2016). Hybrid field of view vision: From biological inspirations to integrated sensor design. In *Proceeding IEEE international conference on multisensor fusion and integration for intelligent systems, Baden-Baden* (pp. 653–658).
22. Rostkowska, M., Wąsik, M., & Skrzypczyński, P. (2018) Implementation of peripheral vision in a hybrid field of view sensor. In R. Szewczyk, et al. (Eds.), *Automation 2018, advances in automation, robotics and measurement techniques, AISC 743, Zürich* (pp. 584–594). Heidelberg: Springer.
23. Salinas, C., Montes, H., Fernandez, G., Gonzales de Santos, P., & Armada, M. (2012). Catadioptric panoramic stereovision for humanoid robots. *Robotica*, 30, 799–811.

24. Santos-Victor, J. A., Sandini, G., Curotto, F., & Garibaldi, S. (1995). Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, 14, 159–177.
25. Scaramuzza, D. (2008). Omnidirectional vision: from calibration to robot motion estimation, PhD Dissertation. Zurich: ETH Zürich.
26. Scaramuzza, D., Martinelli, A., & Siegwart, R. (2006). A toolbox for easy calibrating omnidirectional cameras. In *Proceeding IEEE International Conference on Intelligent Robots & Systems, Beijing* (pp. 5695–5701).
27. Schmidt, A., Kraft, M., Fularz, M., & Domagala, Z. (2013). Comparative assessment of point feature detectors and descriptors in the context of robot navigation. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 7(1), 11–20.
28. Sergiyenko, O. Y., Ivanov, M. V., Tyrsa, V. V., Kartashov, V. M., Rivas-Lopez, M., Hernández-Balbuena, D., & et al. (2016). Data transferring model determination in robotic group. *Robotics and Autonomous Systems*, 83, 251–260.
29. Shu-Ying, Y., Wei Min, G., & Cheng, Z. (2009). Tracking unknown moving targets on omnidirectional vision. *Vision Research*, 49, 362–367.
30. Siagian, C., & Itti, L. (2009). Biologically inspired mobile robot vision localization. *IEEE Transaction on Robotics*, 25(4), 861–873.
31. Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach. *International Journal of Applied Mathematics and Computer Science*, 19(4), 575–588.
32. Sobral, A. (2013). BGSLibrary: An OpenCV C++ background subtraction library. In *Proceeding WVC 2013, Rio de Janeiro*.
33. Soria, C., Carelli, R., & Sarcinelli-Filhot, M. (2006). Using panoramic images and optical flow to avoid obstacles in mobile robot navigation. *Proceeding IEEE ISIE 2006, Montreal* (pp. 2902–2907).
34. Ude, A., Gaskett, C., & Cheng, G. (2006). Foveated vision systems with two cameras per eye. In *Proceeding IEEE International Conference Robotics & Automation, Orlando* (pp. 3457–3462).
35. Ulrich, I., & Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots. In *Proceeding IEEE International Conference on Robotics & Automation, Leuven* (pp. 1572–1577).
36. Wąsik, M., Rostkowska, M., & Skrzypczyński, P. (2016). Embedded, GPU-based omnidirectional vision for a walking robot. In M. O. Tokhi & G. S. Virk (Eds.). *Advances in cooperative robotics* (pp. 339–347). Singapore: World-Scientific.
37. Wehner, R., & Wehner, S. (1990). Insect navigation: use of maps or Ariadne’s thread? *Ethology, Ecology, Evolution* 2, 27–48.
38. Yagi, Y. (1999). Omnidirectional sensing and its applications. *IEICE Transaction on Information and Systems*, 82(3), 568–579.
39. Yagi, Y., Kawato, S., & Tsuji, S. (1994). Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. *IEEE Transaction on Robotics and Automation*, 10(1), 11–22.

Chapter 3

Color and Depth Sensing Sensor Technologies for Robotics and Machine Vision



Ali Shahnewaz and Ajay K. Pandey

Abbreviations

3D	Three dimension
AMCW	Amplitude-modulated continuous wave
APD	Avalanche photodiode
CAS	Computer-assisted surgery
CCD	Charge-coupled device
CMOS	Complementary metal oxide semiconductor
CNN	Convolutional neural network
CNN-CRF	Convolutional neural network-conditional random field
DoG	Difference of gradient
DSSC	Dye-sensitized solar cells
FIP	Focus-induced photoluminescence
FMCW	Frequency-modulated continuous wave
FOV	Field of view
FW	Fixed window
LiDAR	Light detection and ranging
MIS	Minimally invasive surgery
RANSAC	Random sample consensus
RGB	Red green blue
RGB-D	Red green blue depth
SAD	Sum of absolute differences
SfM	Structure from motion
SIFT	Scale-invariant feature transformation

A. Shahnewaz · A. K. Pandey (✉)

School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

e-mail: shahnewaz.ali@hdr.qut.edu.au; a2.pandey@qut.edu.au

SLAM	Simultaneous localization and mapping
SPAD	Single-photon avalanche diodes
SURF	Speeded-up robust feature
ToF	Time of flight

3.1 Introduction

Conventional vision technology projects 3D world information into 2D plane with information lacking in Z-axis, that is, the depth information of a scene. Access to depth information is paramount in capturing the real-world space; therefore, 3D vision systems form important research topic for robotic and autonomous systems. For instance, path planning and obstacle avoidance form the key aspects of autonomous vehicles and heavily rely on sensors providing situational awareness for system accuracy. A large body of research is being conducted on safety and obstacle avoidance, and 3D vision technologies remain an integral part of robotic systems [1–3]. It is not surprising that in addition to the usual red, green, and blue (RGB) color vision, most of the advanced robotic vision systems already deploy a form of active or passive depth information using the so-called RGB-D vision technology, where D stands for depth. In robotics, time-of-flight (ToF)-based sensors together with stereo vision systems are widely used to extract the depth information. ToF sensors are particularly suited to self-driving cars and autonomous aerial systems or drones. ToF-based depth sensor is the most promising form of long-range active depth sensing, and tech giants such as Texas Instruments, Sony, Panasonic, STMicroelectronics, AMS, etc. are currently developing micro-depth sensor for range imaging in a form that is compatible with portable device such as smartphones.

Object recognition in real time is yet another active research area in robotic vision, and use of RGB-D sensors for 3D object reconstruction is common. Information contained in voxels is used to compare and identify different objects and features contained within them [4–7]. The advantage of this approach benefits from the fact that a lot of salient features can be extracted from the 3D space to improve object recognition performance [4–7]. No wonder the demand for novel and high-resolution cameras that can provide depth is on the rise. Currently, many commercial 3D image sensors exist in the market, and imaging system providers are developing a new generation of 3D image sensors [8–10]. Surveillance system, vehicle identification, traffic control system, people counting system, activity and gesture identification etc. are the subdomains of this category where 3D information offers improved system efficiency [1, 2, 4, 12, 109]. Access to depth information has a big impact on computer graphics especially in games and in content and image retrieval as well as in archeology [13–15].

In medical robotics, depth information has a great influence on assigning perception. In computer-assisted surgery (CAS) or in robotic-assisted minimally invasive surgery (MIS), depth has an important role. In conventional MIS procedures, 3D surgical world is projected to 2D screen; hence, surgeons performing MIS face more challenges than open surgery. Surgeon has to operate 3D world in a 2D space

without haptic sense that makes MIS system more complicated. Unintentional tissue damage is often reported that may later cause other difficulties such as arthritis or osteoarthritis. In MIS context, vision is the most crucial factor that improves surgical outcomes with respect to safety and unintentional injury [11]. Without the depth information, MIS faces difficulties to track surgical tools within the surgical space. Promising improvement has been reported when 3D vision is incorporated into the tracking system [16]. Recent studies show a significant amount of improvement in MIS procedure by presenting a comprehensive result of 3D MIS versus 2D MIS. According to their records, median error of MIS in 3D surgery versus 2D is 27 and 105, respectively, that reports 25.72% less median error [17]. Another study shows that 3D MIS reduces 71% performance time as well as 63% error rate [18, 19]. Therefore, 3D vision systems offer great advantage in countries where the number of skilled surgeons is limited.

In this chapter, we describe a diverse range of vision technology by reviewing the current scanning technologies specific to application areas of robotic and machine vision. We also aim to extend this discussion to capture the advantages and limitations of active and passive depth sensing technologies using in stereo vision, time of flight, and structured light with a particular focus on how to deal with constrained (indoor) or unconstrained (outdoor) environments.

3.2 3D Image Construction

Depth estimation technique mainly faces two big challenges: (1) depth accuracy and (2) computational cost in terms of time [20–23, 106, 107]. Two different branches, sensor technology and computer vision, actively involved in research to meet these constraints. Based on the imaging technology, current depth estimation technologies can be classified into two main categories, active or passive, as illustrated in Fig. 3.1. Passive estimation technology relies on machine learning algorithms and

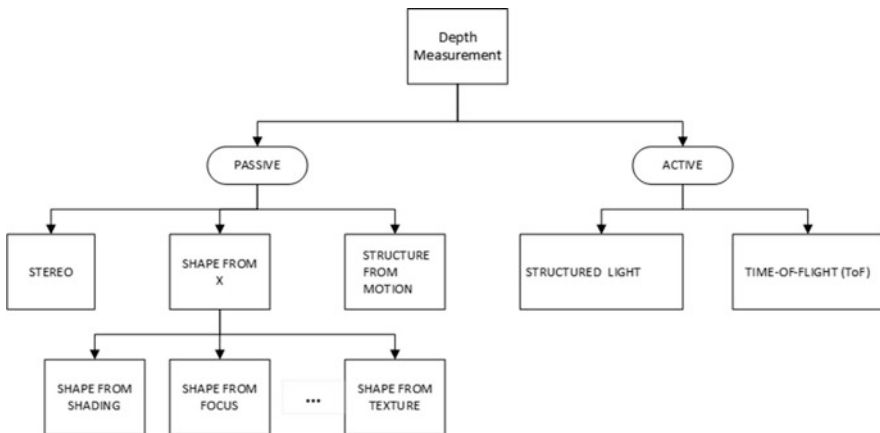


Fig. 3.1 Classification of depth measurement technology

mathematical approaches to extrapolate depth information from 2D image or images. Whereas, the other class is referred to as active depth sensing technology which relies on active controlled signal sources and sensor technology to estimate distance. The aim of this chapter is to provide a comprehensive review of the various depth estimation approaches along with their merits and demerits.

3.2.1 Image Sensor

Imaging technology uses natural or ambient illumination to capture the scene. Most of the image sensors are based on charge-coupled device (CCD) or complementary metal oxide semiconductor (CMOS). On the other hand, optical scanning sensors are used to estimate depth [24]. Wendy Flores-Fuentes et al. proposed a novel electronic sensor that consists of an electronic processing unit along with photodiode. To measure the distance, their proposed work infers the energy centre of an optical received signal. In the next section, we are going to introduce the most promising passive stereo technique to infer 3D structure.

3.2.2 Stereo Vision

Stereo vision is the most common approach to infer depth from a set of images. Computer vision algorithms are used to reconstruct depth from single or multiple images. Single-view 3D reconstruction methodology uses only one image. On the other hand, multi-view 3D construction considers two or more images to reconstruct depth information. It is also known as stereoscopic vision. When two images are used, the system is known as a binocular stereo vision system, and probably it is the most widely focused research area of computer vision.

Binocular stereo vision originally mimics the human vision system. In a binocular stereo vision, two images are taken from two different cameras at the same time [25]. The basic requirement is that two cameras are placed at a known distance. In this arrangement, the left camera is denoted as a reference camera where the right camera is called target camera. The distance between the optical center of these two cameras is referred as a baseline. Stereo vision system uses the concept of parallax and uses disparity as a vision cue. Figure 3.2 provides an overview of binocular stereo vision and how it is used to calculate depth.

As highlighted in Fig. 3.2, stereo matching is the core technique of the stereo vision. Stereo matching is the process that matches each pixel from the reference image to the target image to perceive the depth of each pixel [26]. The resulting output image is often referred as a depth map. An intensive comparison takes place to find the corresponding pixel in the target image. Offline camera calibration and pre-processing always take place before the actual stereo matching process [27–29]. Ideally, the reference and the target cameras capture the same scene point at the same time with a slightly different viewpoint, and this serves the basis for stereo vision algorithms [28]. Therefore, the term synchronization is always used to convey the

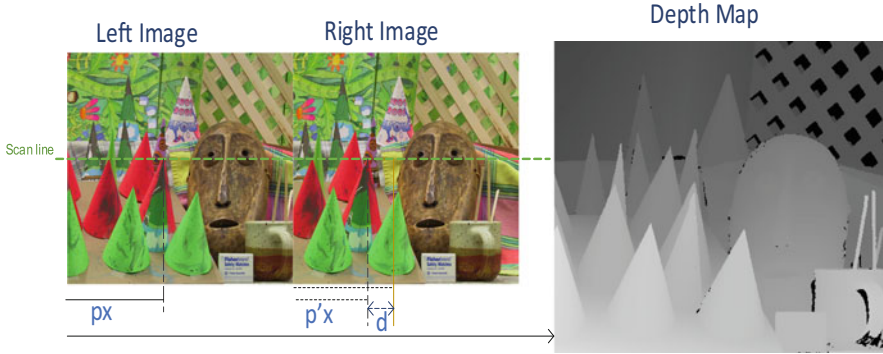


Fig. 3.2 Overview of binocular stereo vision system. Left and right cameras take the same scene image, then stereo matching is performed to find the corresponding points. The resulting corresponding points provide disparity information with respect to the left image. Finally, depth map is calculated from disparity

sense that the image acquisition system captures the same scene point at the same time with no time lag [30, 31]. When the object is in motion, this precondition plays a pivotal role to reduce reconstruction noises, and camera calibration process is used to eliminate image acquisition distortions [27, 32]. Basically, stereo rectification is a transformation process that aligns two images into the same plane, so that same horizontal line becomes parallel to both camera centers [33, 34].

Depth is calculated by finding disparity in a pair of images. Disparity refers the distance between two corresponding points in the left and right images of a stereo pair. It is inversely proportional to the depth and vice versa. In a stereo vision system, the relationship between depth and disparity can be expressed as follows [35–37]:

$$d = \frac{bf}{Z} \quad (3.1)$$

Here b is the baseline and f is the focal length. Z stands for depth, and disparity is expressed by the letter d . When the stereo matching process is completed, the difference of the pixel position in the right image with respect to the left image is referred to disparity of that pixel. The basic idea of the disparity calculation is to match each pixel from the left image to the right image. In some circumstances, parts of a scene may not be visible to one or both cameras. This part of a scene is known as a missing part. In other words, stereo matching process fails to find the best match. These outcomes are often reported as holes [38]. Hence, after calculating depth map, post processing algorithm are used to refine noises [35]. Depth is estimated from disparity by using the geometric principle of triangulation, and some of the common approaches are summarized in Fig. 3.3.

Generally, stereo matching algorithms are classified into two groups. Pixel-wise matching algorithm is categorized as a correlation-based approach. It is

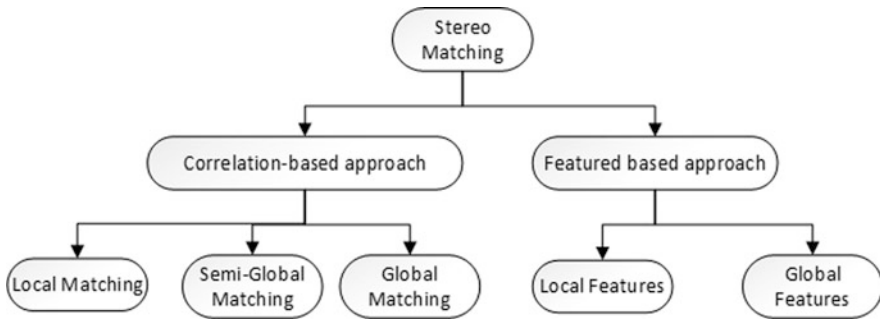


Fig. 3.3 Classification of stereo matching algorithm

further grouped into two main groups: (1) local matching and (2) global matching. Between the local matching and the global matching, semi-global matching resides that combines the upsides of both the algorithms. Stereo matching becomes highly ambiguous especially when a single pixel is considered. To alleviate this characteristic, often fixed or variable length of window is considered. Example of the local matching algorithm is the sum of absolute differences (SAD), fixed window (FW), etc. [39, 40].

Local methods can estimate disparity at high speed, but it compromises estimation accuracy to computational cost. The downside of local matching algorithms is that the disparity map often contains ambiguity. Though window-based approaches improve the overall accuracy, defining universal window size to balance both speed and accuracy is a challenging task. Probably, one of the major limitations of local matching algorithm is that it is incapable to handle occlusion due to lack of global information. Moreover, this group of algorithms is often limited to low texture images because, often, the local windows fail to capture smoothly varying texture features [13, 21] at low frequency. The rudimentary hypothesis of this group of the algorithm is that the corresponding pixel exists on the same horizontal scan line. For this reason, rectification is a crucial step to increase the accuracy of disparity estimation. However, accurate image rectification in practice is a hard task. Some algorithms also consider an additional path to estimate disparity [41]. But this additional path aggregation function again increases the disparity computational cost.

On the other hand, global matching algorithms provide improved and highly accurate depth map [102]. Instead of the local neighbor pixel, the global method takes into account all image pixels. Smoothness function is the most pivotal step of global method. The aim of this step is to minimize the energy cost of the overall depth map. The objective is to reconstruct depth map with the lowest energy. Unlike local stereo matching algorithm, this set of algorithms requires very high computational cost. Generally, the energy minimization function is defined as [1, 42]:

$$E(d) = E_D(d) + E_s(d) \quad (3.2)$$

Here, $E_s(d)$ is known as a smoothing function.

Local methods are not robust to noise, and accuracy gets compromised with respect to speed. On the other hand, global methods consume high computation cost, are robust to noises, and provide highly accurate results. Adaption of a global method in a real-time system is a challenging task. The semi-global method originally proposed by Hirschmuller [41] balances these two approaches. According to this method, the matching process is performed with a set of pixels that is basically a window-based approach. So, initially, the stereo matching approach starts with local stereo matching process. Census transformation and sum of absolute differences (SAD) are probably the most used algorithms to perform this task. Census transformation is more robust than SAD [43]. Here, window size or census kernel size plays an important role in identifying textureless or low texture properties. The drawback of larger window size is that it increases computational cost. In order to estimate the matching cost, generally hamming distances are used. The lowest hamming distance is preferred for each pixel over the total disparity level. This initial matching cost encounters the same problems of the local matching algorithm. Thus, it contains wrong correspondences due to limited or low textures. To alleviate these problems to some extent, the semi-global method introduces further cost aggregation function which is known as a path cost aggregation. Path cost is calculated from several directions, and in practice, 8–16 directions are used. Although the semi-global method improves local method matching accuracy, this method still falls short in fully overcoming the above-described limitations. Path cost aggregation of the semi-global method can be described as follows:

$$E(D) = \sum_p \left(C(p, D_p) + \sum_{q \in \mathcal{N}_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in \mathcal{N}_p} P_2 T[|D_p - D_q| > 1] \right) \quad (3.3)$$

Passive stereo matching technology encounters a set of challenges. Image may be contaminated by noises. Missing point due to occlusions or self-occlusion, the absence of texture, and the perfection of same horizontal scan line alignment are the rudimentary problems to reconstruct 3D structure using stereo images. Principally three pivotal metrics are used to describe a stereo matching algorithm as a whole. These are (1) robustness, (2) accuracy, and (3) computational cost. Feature matching-based algorithms are also widely used to estimate passive depth from images. In this approach, features are calculated to construct feature vectors. This process can be referred to as a feature descriptor process where features are extracted from the images. Then feature matching algorithms are used to find the correspondence feature, and disparity is calculated based on the matching outcomes. The most common image features are edges and corners. But these features are usually susceptible to noise but have less computational cost. Other widely used image features are scale-invariant feature transformation (SIFT), difference of gradient (DoG), and speeded-up robust feature (SURF) [42, 44–47]. Feature selection is a crucial process. Robust features are always preferable, but it increases computational cost. By definition, features are the most interesting points of an image that carries

important image information. Therefore, feature-based methods create a sparse matrix. Only partially reconstructed depth can be achieved from feature-based stereo matching algorithm compared to dense depth map construction.

3.2.3 *Shape from Shading*

Binocular stereo is based on finding the corresponding problem. However, images also contain many visual cues such as shading, texture, etc. In computer vision, these visual cues are used to construct the shape of an object. These are classical approaches used in monocular stereo system. Among them, shape from shading and photometric stereo are the most prominent fields which are still viable and probably the most widely used research approaches to reconstruct 3D structure from a single image. This approach has many applications. Considering the growing interest of this approach, we focused on shape from shading and photometric stereo in more detail.

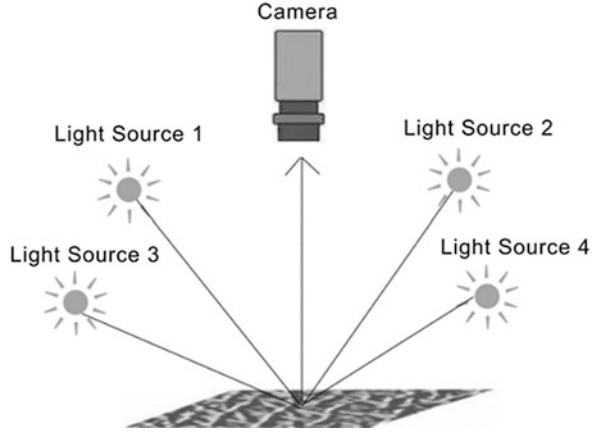
Originally, shape from shading approach was proposed by Horn [48] and later by his Ph.D. student Woodham. In his Ph.D. thesis, Woodham proposed a photometric approach which is the extension of shape from shading [49]. Though it seems a very old approach, it is still an active research area in computer vision to infer depth from monocular or single view.

Shading pattern of an object conveys information and visual cues of its surface. Under controlled lighting source, the reflected light intensity of an object surface has a sharp relation to its surface shape. It creates a bridge between the shading to surface slope. However, shape from shading is often marked as an ill-posed problem that refers to the same numerical solution representing two distinct surfaces, one is inversion of the other one. Photometric stereo, which is one step further from the approach of shape from shading, solved this problem using more light sources [112]. As shown in Fig. 3.4, the idea behind photometric stereo is to estimate surface reflectance coefficient, albedo, and surface normal. When these are estimated, depth of a surface is calculated by integrating surface normals or by solving nonlinear partial derivative equation. One important definition in this context is the surface albedo. It is the reflectance coefficient that tells the amount of light a surface can reflect. The value of albedo is between 1 and 0, and it is denoted by ρ . Limitations of the shape from shading is that this approach is based on some assumptions such as Lambertian surface, surface smoothness, and discontinuity. On the other hand, photometric stereo is often limited to complex lighting environment and specular nature of a surface. Using knowledge of radiometry, numerical solutions, and proper construction of the system especially known lighting source, photometric stereo is able to capture depth from textured, untextured, or textureless images.

Surface intensity or surface irradiance can be expressed as

$$I(x, y) = R(p, q) \quad (3.4)$$

Fig. 3.4 Photometric stereo system. The figure is taken from S. Ali et al. [50]. Four images are taken from four different lighting sources at different angles. z is the depth of the point



From the radiometry and back to the literature, Horn used this relation to model shape from shading. Equation (3.4) tells that the image intensity is directly proportional to its reflectance (R) map which is also known as irradiance map. Reflectance map is a relational map which relates scene radiance, surface reflectance property, surface orientation, and observed brightness [51]. If surface reflectance property is estimated properly, then surface radiance depends on the surface shape. Horn approach p and q in Eq. (3.4) represent the surface gradient points and can be expressed as

$$p = \frac{dz}{dx} \quad (3.5)$$

$$q = \frac{dz}{dy} \quad (3.6)$$

Extension of this shape from shading is photometric stereo. The basic idea is to infer depth of a scene illuminated at different angle. In photometric stereo, a camera is placed in a fixed position. Usually, three or more lighting sources are used to construct photometric stereo. Images are captured one after another by changing lighting direction from one to another. The idea is to capture surface orientation from different illumination direction. Collected images are then processed to construct depth map.

One reflectance map corresponds to one light source. So, l number of light sources produce l number of reflectance maps. Unlike shape from shading, photometric stereo calculates surface property such as albedo. Photometric stereo is an overdetermined system where the number of unknowns is less than the number of equations. Hence, it eliminates the limitations of shape from shading. The surface normal can be defined as a vector on a surface in 3D space which is perpendicular to the surface. The basic principle is based on the radiance by calculating surface normal and the direction of

light. Suppose \mathbf{s} is the light source vector and \mathbf{n} denotes surface normal, then image irradiance can be expressed as follows:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} \mathbf{n} \quad (3.7)$$

Finally, the depth map is calculated from the surface normal through numerical integration. There are some well-known numerical methods that already exist in literature mainly based on fast marching method and the integration methodology [52, 53]. According to the integration method, depth can be calculated from the following equation [54]:

$$z(x, y) = z(x_0, y_0) + \int p(x_0, y_0) dx + q(x_0, y_0) dy \quad (3.8)$$

Lambertian surface property is the preliminary assumption of the shape from shading or photometric stereo. Dynamically estimation of a surface property or photometric stereo for non-Lambertian surface is one of the active areas where a lot of contribution has been reported. In recent years, many contributions are reported where structured light, color image intensity, and fusion of photometric stereo with other approaches are used [55, 56]. The strong point of photometric stereo is that it provides fine surface shape with fine depth information. Recent patent has been reported in 2018 where photometric stereo process has been used to reconstruct 3D environment model [57]. The downside of this approach is that, unlike passive stereo system, it uses external lighting sources to estimate depth. Hence, it is limited to environmental lighting sources, or complex lighting sources make this approach hard to estimate depth.

3.2.4 Dynamic Vision

In contrary to other conventional imaging system or camera, the event camera meets high-speed vision sensor demand. The idea behind the event camera or dynamic vision sensor is to produce an image when an event has occurred. In other words, even if the brightness value of a single pixel changes, it produces an image. Event camera does not produce image at a fixed rate, but based on an event, it generates an image at high speed. An event can be translated into time series tuple of $\langle t_k, (X_k, Y_k), p_k \rangle$ [58, 59], where t_k expresses time, (X_k, Y_k) is the coordinate of a pixel that raises an event, p_k defines priority. Event camera can produce an event in some order of milliseconds [58, 59]. In robotic odometry, event camera provides the ability to solve many feature-based cutting-edge problems such as visual simultaneous localization and mapping (SLAM) [60, 61]. On the other hand, the event camera has great influence on passive depth estimation. Examples of event-based scene detection is shown in Fig. 3.5.

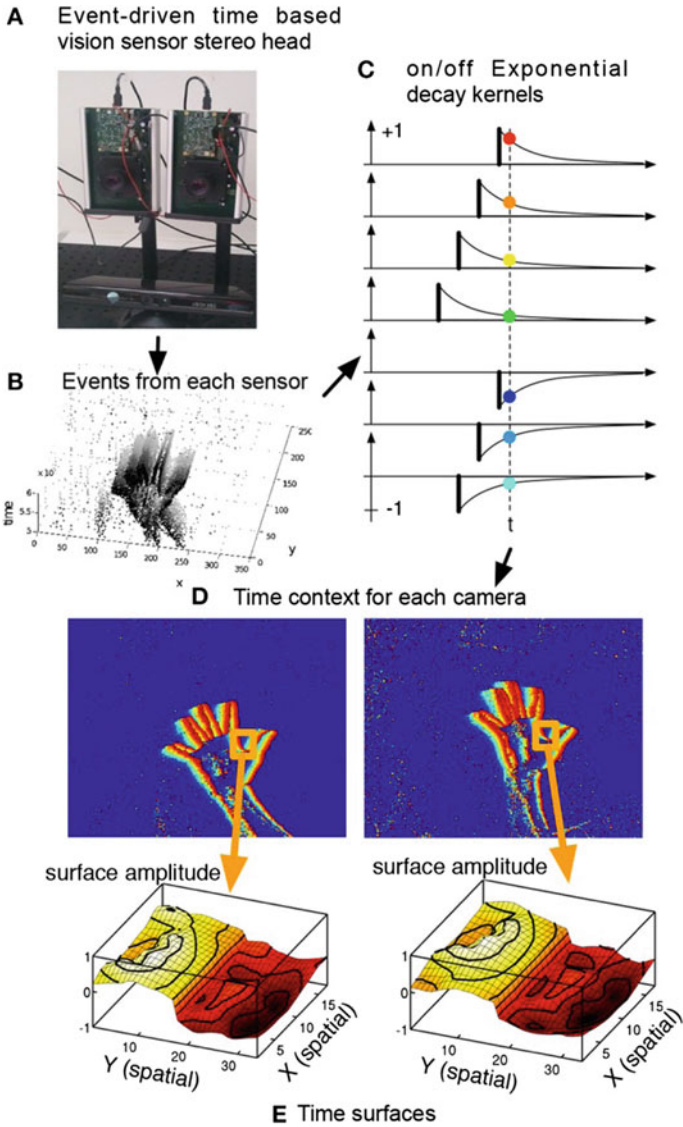
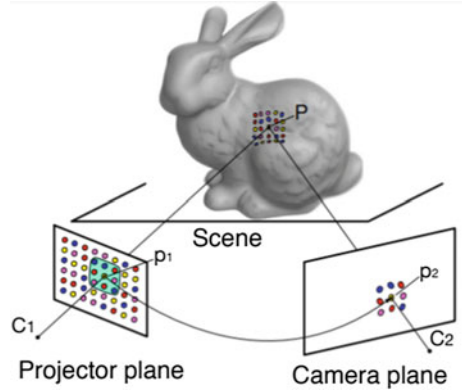


Fig. 3.5 Dynamic vision based on events. Events in the scene are captured through dynamic vision cameras, and depth construction is obtained from the images. (a) Dynamic vision camera-based stereo system. (b) Output events and captured scene. (c) Extracted neighbourhood that allows to build the event context. (d) Time context of the most recent events. (e) Exponential decay kernel for spatial domain. Figure is taken from Ieng et al. [62]

Fig. 3.6 Dynamic vision with structured light. Matching is performed between two view p_1 and p_2 , and depth is recovered through triangulation method from known optical center C_1 and C_2 . Figure is taken from T. Leroux et al. [64]



N. Julien et al. estimated depth using dynamic vision sensor using active approach [63]. In their work, they addressed passive stereo matching problem using event data. They generated events of an observed scene so-called light spots using lens and laser light, and scanning was performed by translating laser beam. The Fig. 3.5 shows the output of a stereo rig consisting of a dynamic vision sensor that produces overlapped stereo images. Stereo matching is performed over the sparse data at each event. It alleviates the stereo matching problem. An active pixel array is used to grab a visual scene. Though this work approaches to solve the stereo matching problem, scanning all the pixels of the field view area consumes time. Moreover, their approach is limited to a range in some meters.

T. Leroux et al. in their method used digitized structured light projection with an event camera to estimate the depth of a scene [64]. Their method as shown in Fig. 3.6 relies on the use of frequency-tagged light pattern. It generates a continuous event. Since structured light has a distinguishable property of pattern at a different frequency, it facilitates matching problem on event-based data.

The fundamental approach is based on the idea that unique projector pixel triggered a unique scene point that is captured by the image sensor. By knowing this two center points say C_1 and C_2 , depth is recovered using the triangulation method.

3.3 Active 3D Imaging

The active 3D imaging system consists of an additional signal source known as a projector. The aim of the projector is to emit signals. Received reflected signals are analyzed to construct the 3D structure of the surrounding environments. The emitted signal can be laser light, ultrasound signal, near infrared light, etc. It is known as a projector, and its responsibility is to fire signals on the surrounding surface. Many terms are used to describe 3D active imaging technology such as a rangefinder and range imaging. Several methods are used to measure distance, but probably the most

practiced methods are based on time of flight (ToF), triangulation and phase shift. This section provides a brief introduction of active sensing methods and technique. Dense depth map with less ambiguity and minimum depth error are the most reported advantages of active 3D imaging technology. However, the resolution of the depth map is limited. Miniaturized, high-resolution, and low-power active depth sensor has a potential demand in various fields like medical robotics.

3.3.1 Time of Flight

Time-of-flight (ToF) systems measure the distance from the scanner to surface points. The basic idea of the active sensing technology is to emit signal such as from a laser. When the signal is emitted by the projector, then the clocking system inside the active imaging system starts counting. This approach is known as direct time of flight. If the object exists within the range of the imaging system, then it reflects a potential amount of signal to the receiver. When the receiver part of the camera receives this signal, it then computes the round-trip time. Then the distance is estimated from the basic principle of the light or electromagnetic source as follows:

$$d = \frac{\Delta t * c}{2} \quad (3.9)$$

Here d expresses the distance of an object from the camera, Δt stands for total travel time, and c is the velocity of the light. Figure 3.7 captures the fundamental working procedure described above. Direct ToF imaging system consists of four basic components: (1) light source or transmitter, (2) optics, (3) light detector or receiver, and (4) electronic timer [65]. Light source or transmitter along with optics produce a signal transmitting unit for this system. Different light or signal source can be used such as near infrared or laser. The optical lens is used to diffuse the signal

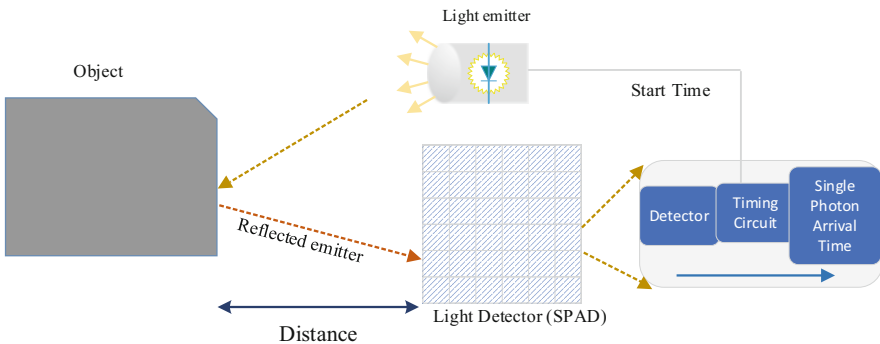


Fig. 3.7 Working principle of ToF-based sensor

over the surface. The optical lens is also used to collect light and project it to the receiver. It creates limited field view to avoid other outdoor lighting such as sunlight.

On the other hand, the receiver unit is generally composed with two system components: (1) photosensor and (2) electronic time counter. Within a defined range, ToF provides high-quality depth map. High-scale precision clock is the challenging part of this approach. When an object is placed very near to camera, for example, in millimeter distance, it is a challenging area to design a clock that can measure a time gap in nano or pico scale. That makes active direct ToF camera limited to very short distance.

Photosensor has to sense reflected light within a very short time. Some semiconductor components such as avalanche photodiode (APD) and single-photon avalanche diodes (SPADs) show their ability to sense signal within the picosecond range, and these components are used to fabricated ToF sensor [66]. To improve the resolution, efficiency, and scale down the dimension of the whole imaging system, currently leading manufacturers are involved in developing solid-state 3D active imaging system [9]. In the indirect approach, a continuous signal is sent by the transmitter or projector instead of the one-shot signal in order to avoid small-scale clock design. The transmitter contains an array of a signal emitter and generates the desired signal. Different kind of signals are used such as sine, square, etc. The received signal is compared to the original signal. Different signal characteristics are used such as signal phase to estimate distance. It is a continuous process and more flexible for silicon technology.

One-shot approach can measure both short and long distance with some range limitations. Long distance measurement requires stronger light source, in most of the cases coherent light sources, which can be hazardous. Moreover, strong and complex lighting source can contaminate the reflected signal. In practice, a multi-shot approach is adopted to overcome this problem. However, still high-power light source is the main drawback considered in this situation. The continuous pulsed signal is used to overcome this crucial problem. From the basic theory of signal processing, the target signal is wrapped into a carrier signal that has relatively low frequency. Often amplitude-modulated continuous wave (AMCW) or frequency-modulated continuous wave (FMCW) are used in this domain. In frequency-modulated continuous wave (FMCW), high-frequency signal is combined to a relatively low-frequency signal and then transmitted. This mechanism increases the system robustness. Suppose an emitted signal St_x is transmitted and a reflected back signal Sr_x is received. If a sine signal is transmitted, then they can be expressed as

$$St_x = \cos 2\pi \omega t \quad (3.10)$$

$$Sr_x = \cos (2\pi \omega t + \varphi) \quad (3.11)$$

where φ contains phase shift information that eventually expresses the amount of time and distance that the signal traversed after its emission. Basic electronics and filter approaches are used that estimate phase shift between transmitted and received

signal. Multi-path propagation is one of the considered problems of ToF technique. When light hit a surface point, the scattered light may fall on the detector plane through different paths. Multiple detection of such event can generate noises.

In single-photon detection, especially in solid state where avalanche photodiodes are widely used to convert light energy to current energy, strong light source generates strong current. When light source traverses a long path, the signal becomes weaker and generates less current. Similarly, when the signal hit a diffuse surface or mat surface, it reflects weaker signal. The signal is scattered into different directions when it hits sharp edges and photodiode response becomes weaker. These are well-addressed problems of ToF technique especially for LiDAR (light detection and ranging). LiDAR-based system uses the fundamental principle of ToF [67]. Laser is a more preferable choice as a light source. When laser light is reflected back from a surface point, LiDAR can estimate the surface distance. Since light is projected on the surface, the surface property and other factors can also contaminate the reflected light as it is mentioned in the ToF section.

Spot scanner scans a single point at one time. This type of LiDAR projects laser light on the surface point. The back-propagated light is captured and projected to the light detection sensor. Single point distance is measured with this approach. To recover whole geometry covered by the field of view (FOV), conventional steering is used to scan all the points.

Apart from the pulsed shot approach, amplitude modulation continuous wave (AMCW), frequency modulation continuous wave (FMCW), and triangulation techniques are also adopted. In recent years, a big volume of literature currently exists which concentrates ToF camera especially LiDAR on solid state. Single photon on distance measurement technique is widely adopted. Some solid-state materials such as avalanche photodiode (APD) and single-photon avalanche photodiode (SPAD) are widely used in this research area to detect incoming light at very small time gap [68]. With the capability of detecting and discriminating incoming light in the range of picoseconds, these materials became the state-of-the-art choices to develop solid-state LiDAR.

Table 3.1 presents a comprehensive list of active depth measurement sensors and their characteristics

3.3.2 Structured Light

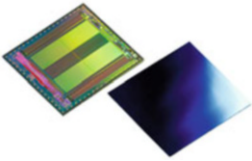
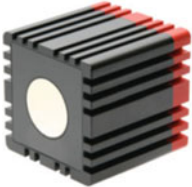


Structured light is used to estimate the depth of a surface. This technology is widely used to construct 3D image [69–71, 105]. Similar to time-of-flight (ToF) mechanism, it uses a projector that generates a pattern of light. Considering pattern generation procedure, structured light can be further categorized into basic two classes: single-shot structured light and multi-shot structured light. When a pattern is projected, the surface scene is captured by the image sensor. Based on the number of the image sensors used, a structured light depth estimation procedure has two well-studied directions: (1) monocular structured light and (2) binocular structured light [71, 72].

Table 3.1 List of active depth measurement sensors

Product name	Characteristics	Vendor
REAL3 	Dimensions 68 mm × 17 mm × 7.25 mm Measurement range 0.1–4 m Frame rate max 45 fps Resolution 224 × 172 pixel (38 k) Viewing angle (H × V) 62° × 45°	Infineon REAL3™
PMD PhotonICs® 19k-S3 	Time-of-flight 3D chip Dimensions 12 × 12 mm ² Pixel array 160 × 120	PMD
OPT9221, OPT8241, OPT3101 OPT8320 	Time of flight Long-range distance Sensor resolution 80 × 60 to 320 × 240 Frame rate 1000–120	Texas Instruments
PX5 	Alternative spatial phase image 3D sensing Up to 5 MP resolution Fame rate 90	Photon-x
BORA 	Time of flight Resolution 1.3 Megapixel Distance various range Minimum range 0.5 m Maximum 500 m	Teledyne e2v
IMX456QL back-illuminated ToF 	ToF image sensor VGA resolution 10 μm square pixel Approx. 30 cm to 10 m distance measurement	Sony DepthSense™

(continued)

Table 3.1 (continued)

Product name	Characteristics	Vendor
Epc660 	Time of flight Resolution 320×240 pixels (QVGA) 1000 ToF images per second in advance operation mode Range millimeter to 100 m	ESPROS Photonics Corporation
MESA 4000 	Time of flight Distance 0.8–8 m Resolution 640×480	MESA Imaging
SR300 	Structured light (IR) Distance 0.2–1.5 m Resolution 640×480	Intel RealSense™
ASUS Xtion 	Structured light (IR) Distance 0.8–4 m Resolution 640×4800	Asus

Depth estimation procedure analyzes a captured structured light pattern to estimate depth. Different methods are already established so far and can be grouped into (1) spatial neighborhood pattern method, (2) time-multiplexing pattern method, and (3) direct coding pattern method [23]. The fundamental approach of structured light depth estimation is to calculate disparity and can be defined as $d = U_a - U_c$. U_a comes from the projector coordinate system and U_c comes from the camera coordinate system.

As shown in Fig. 3.8, in this arrangement, depth estimation can be defined as a pattern matching problem of the scene that is illuminated by a specific light pattern.

Some approaches are based on deformation of the received pattern. Considering the correspondence problem of passive stereo vision, it shows a significantly improved result on a textured and textureless region as well as it reduces ambiguity [19, 74]. In structured light triangulation method, camera calibration can be the first building block that estimates camera intrinsic matrix. It is also important to estimate the extrinsic parameter that maps projector coordinate to camera coordinate system known as stereo calibration. Encoded light patterns are projected on the surface, and reflected patterns are captured by the image sensor. Deformation depends on the surface planar characteristics. Matching is performed on the decoded pattern by using different approaches such as global optimization [75, 76]. Then depth is

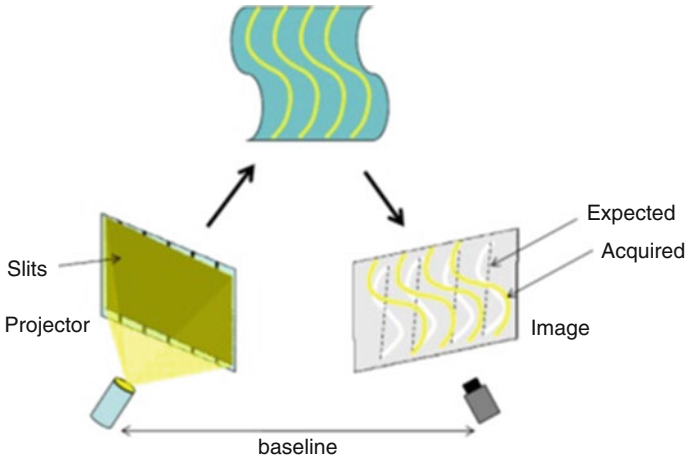


Fig. 3.8 Structured light system architecture [73]

inferred using triangulation [77–79]. Numerous pattern generation and structured light techniques exist in literature and are used in practice. A comprehensive list of structured light techniques has been stated by Jason Geng published in *Advances in Optics and Photonics* [80]. According to their research, structured light technique is categorized into five main categories: (1) sequential projections (multi-shots), (2) continuous varying pattern, (3) stripe indexing, (4) grid indexing, and (5) hybrid methods.

Structured light and photometric stereo both are active depth sensing technology. Through the advances of solid-state physics and micro lens technology, miniaturized depth sensor is now possible, and this technology is expected to improve further in terms of image resolution. A comprehensive comparison of depth sensing imaging technology is presented in Fig. 3.9.

Alternatively, the backbone of the image sensing technology beyond CMOS can also be modified to estimate depth. Recent research shows that new form of image sensor is capable of estimating depth directly from incident light [81]. Pixel aperture and depth from defocus are evaluated to construct a camera sensor. One of the merits of this sensor design is the control of pixel aperture-controlled pixel array design. Within a single die approach can capture blurred and sharp image. At a certain distance, a camera can produce a sharp image, which depends on camera focal setting. Otherwise, it produces a blurred image due to defocus. How much image blurring has occurred gives a cue to estimate depth [82, 83]. This approach is known as the depth from defocus. It requires two images, one sharp image and other one is blur image. Their constructed image sensor uses two different filters. A color filter is used to construct a blurred image, and a white filter is used to produce a sharp image. Both these images are then used to estimate depth. Pixel aperture controls the incident light. It can block, partially block, or pass whole amount of light to the image plane.

Parameter		Active			Passive	
		ToF (Scan)	ToF (Flash)	Structured Light	Stereo	Spatial Phase Imaging (SPI)
Operating Wavelength		NIR	NIR	NIR	Visible	UV to Radio
Resolution	Pixel Count Wavelength	Low: 10k - 100k	Medium: 2MP	Low: 10k - 100k	Low: 10k -100k	High: 5MP - 20MP
Pixel Count Constraints		Number of emitters Emitter Intensity Scanning Speed Processing Speed	Number of emitters Emitter Intensity Scanning Speed Processing Speed	Number of emitters Emitter Intensity Scanning Speed Processing Speed	Number of Aperatures Ambient light intensity Number of corresponding points Processing Speed	Ambient light intensity
Range		2.5m – 50m	10m -100m	50 cm – 6m	2.5cm – 10m	0.01um – 100km
Power Consumption		Medium	High	High	Low	Low
Color		No	No	NO	Yes	Yes
Computational Demand		High/ Medium	High/ Medium	High/Medium	High	Low
Hardware		Simple illumination Complex sensor	Demanding illumination Complex System	Demanding illumination Complex System	Sometimes requires illumination Simple cameras simple system	Rarely requires illumination Simple sensor Simple System

Fig. 3.9 Comparison of 3D imaging techniques. Figure source is Photon-X [8]

Pekkola Oili et al. presented focus-induced photoresponse technique to measure distance [84]. Their approach is based on photoresponse materials such as dye-sensitized solar cells (DSSC) and optics. Photoresponse of a photosensor depends on the amount of incident photons and the surface area in which they fall [84]. The authors referred it as focus-induced photoluminescence (FIP) effect. In their technique, they use this property with the combination of lens, and they successfully derived the distance from the FIP effect. As shown in Fig. 3.10, they presented a single-pixel measurement technique, and to retrieve full geometry of an object, scan needs to be performed over the whole surface.

When lights fall on photodiode, it then generates photocurrent. FIP effect expresses the amount of light in terms of photocurrent. However, ambiguity arises when light radiant power is unknown. Their system arrangement consists of two photodiodes, and one lens is placed in front of the system that collects rays as it is mentioned in the figure. Instead of single photocurrent, their approach uses photocurrent ratio of two sensors to alleviate this situation. Moreover, their work shows the quotient changes with distance.

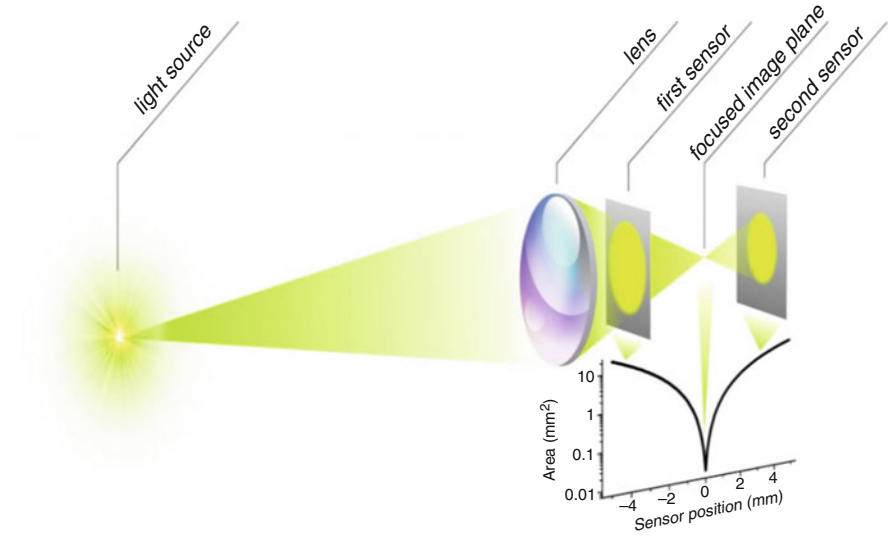


Fig. 3.10 FIP-based distance measurement technique [84]

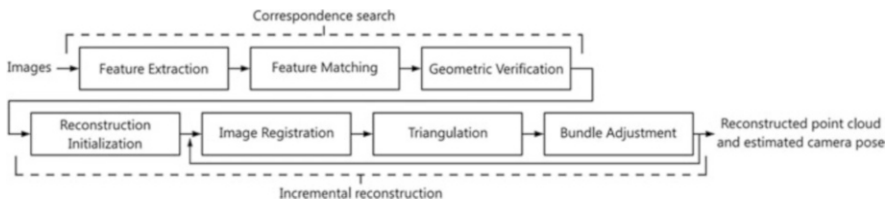


Fig. 3.11 Overview of structure from motion incremental pipeline. Input images are captured from different view angle. Figure is obtained from S. Bianco et al. [87]

3.3.3 Shape from Motion

Structure from motion (SfM) is one of the mature techniques to reconstruct a shape from a sequence of images. Some commercial 3D rendering softwares already adopt this approach to construct the 3D shape of an object [11, 85, 86, 103]. In this technique, motion is used to infer the depth of a scene. The concept behind SfM is shown in Fig. 3.11. Here motion means a scene is observed from a different angle of view. Generally, under orthographic camera model, at least three image sequences are used to estimate depth. Although various shape from motion algorithms exist, this chapter only focuses on the state of the art of shape from motion pipeline.

Using motion, multi-view images, structure from motion is the technique to reconstruct 3D shape of an object and simultaneous estimation of camera pose. SfM takes a series of input images from different camera view (motion). It is a sequential processing pipeline that iteratively estimates motion and shape. The first

stage corresponds to the feature extraction process. Local features are extracted from every frame, and the extracted features matched within the overlapped image pair. Correspondence outliers are filtered out via random sample consensus (RANSAC) and bundle adjustment. Projective geometry is used to verify matched features. Different geometries and parameters such as homography, camera fundamental matrix, epipolar geometry, perspective n point, and triangulation are used to reconstruct shape. Highly overlapped images are a good candidate to improve its efficiency.

Feature extraction and matching action are then performed over the pair of overlapped images. Observed images are taken at different angles; hence, view angle and illumination condition affect the overlapped images. Thus, a feature that is observed from one view angle may not become visible from another view angle due to loss of illumination characteristics such as edge property. Moving from one view angle to another, same features have the probability to compromise its dimensional characteristics. Scaling factors may affect the matching process. Feature points are the key elements that describe scene context; hence, more feature points are desirable. In the last decade, scale-invariant feature transformation [88] approach has been widely adopted in this context because it is robust to noise.

3.4 Deep Learning Approaches to 3D Vision

Deep learning has gained much success in complex computer vision problems [86, 89–94]; recently it has been used to solve 3D reconstruction problem [104, 108, 110, 111]. Multilayer perception and its capacity to infer knowledge in 3D reconstruction domain have been deliberately used to solve different problems in different approaches. Considering shape from motion approach, features of an image sequence has great impact. Often low-textured and salient features are hard to extract. Convolutional neural network (CNN) has been used in this domain, and this approach significantly shows better performance compared to other feature extraction methodologies such as SIFT and DoG in a different environment [86]. Similar problem has been addressed by a deep learning context, and it shows a significant improvement [89, 93] in estimating pose. However, a more sophisticated approach has been developed, and a full network has been developed to solve structure from motion problems [90, 91]. Moreover, these approaches solve conventional structure from motion such as small camera translation problem [91, 95]. CNN is also used to infer depth that comes from the technique such as the depth from defocus [95]. Mainly it improves the depth uncertainty problem. Stereo matching problem also referred as a finding correspondences. Deep learning has been well studied to solve both problems related to passive stereo vision: (1) finding feature and (2) finding correspondences [96–98].

Monocular or single depth image also has a great impact on computer vision as well as on robotics. SLAM is widely used to solve robot localization problem. SLAM depth is conventionally based on structure from motions that are limited to

low texture and small translation [90, 99]. Improvement has been observed when deep learning is used to estimate depth especially when a low-textured region has been considered [99]. Convolutional neural network-conditional random field (CNN-CRF) framework for monocular endoscopy can estimate depth with relative error 0.152 and 0.242 on real endoscopic images [92].

Photometric stereo and structured light have been widely used in many areas. Moreover, object or camera in motion and surface property estimation such as Lambertian or non-Lambertian are the challenges and limitations of photometric stereo. Deep learning has been used to estimate surface normal vector which is the rudimentary step of photometric stereo before calculating depth [51, 90, 94]. Deep learning is also used to estimate depth in a supervised and unsupervised manner [51, 90, 94, 100]. In a supervised manner, the network needs to be trained with known data set and its ground truth depth map. In an unsupervised manner, depth can be estimated from both monocular and binocular views. It opens a freedom in such a way that, even if a stereo arrangement fails, network remains active and provides depth from any single image. The idea of this approach is to predict stereo images in a sense that for an input image say left image L , a network is trained in such a way that it can predict a disparity map [101]. Depth can be calculated from predicted disparity using triangulation method with a known baseline that is used to train the network. Several smoothing functions are used to reduce prediction error and noises.

Though deep network shows high accuracy [101] result, it cannot predict the depth of an unknown object shape which is not used during the training [90]. Also, the deep network needs a well-trained network to estimate depth in real-world environment.

3.5 Conclusion

Current sensors are able to achieve depth resolution from few centimeters to 100 m in real time, and sensor technologies like ToF, structured light, and stereo vision largely form the backbone of object detection and range finding applications in robotics and autonomous systems. Extraction of depth information from computational techniques is yet another growing area of research, and approaches like shape from shading and structure from motion offer some advantages in sensor design. Ambient light spectrum and light intensity planes play an important role in getting a dense depth map, and often lighting conditions experienced in complex environments contaminate depth estimation process. Demands on illumination pattern and computation limit the role of certain depth sensing mechanisms to static or less mobile platforms, and one sensor might not be a good fit. New sensing architectures and neuromorphic approaches to sensor design are already in progress to simplify some of these challenges. Ideally, miniature sensors with low power consumption and computational demands that can combine depth as well as accurate color information are preferred. The ability to add multi-spectral imaging on depth sensors is another area of interest and fusion of depth from different sensor technologies would solve some of the challenges in achieving robust vision for aerial, marine, and medical robotics.

References

1. Schauwecker, K., & Zell, A. (2014). On-board dual-stereo-vision for the navigation of an autonomous MAV. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 74(1–2), 1–16.
2. Di Stefano, L., Clementini, E., & Stagnini, E. (2017). Reactive obstacle avoidance for multicopter UAVs via evaluation of depth maps. In *13th International Conference on Spatial Information Theory*.
3. Massimiliano, I., & Antonio, S. (2018). Path following and obstacle avoidance for an autonomous UAV using a depth camera. *Robotics and Autonomous Systems*, 106, 38–46.
4. Elaiwat, S., Bennamoun, M., Boussaid, F., & El-Sallam, A. (2014). 3-D face recognition using curvelet local features. *IEEE Signal Processing Letters*, 21, 172–175.
5. Maturana, D., & Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 922–928).
6. Schwarz, M., Schulz, H., & Behnke, S. (2015). RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1329–1335).
7. Song, S., Lichtenberg, S. P., & Xiao, J. (2015). SUN RGB-D: A RGB-D scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 567–576).
8. Retrieved from <http://www.photon-x.co/>
9. ToF sensors. Retrieved from <http://www.ti.com/sensors/specialty-sensors/time-of-flight/>
10. NanEye Stereo web. Retrieved from <https://ams.com/3d-sensing>
11. Vélez, A. F. M., Marcinczak, J. M., & Grigat, R. R. (2012). Structure from motion based approaches to 3D reconstruction in minimal invasive laparoscopy. In A. Campilho & M. Kamel (Eds.), *Image analysis and recognition*. Berlin: Springer.
12. Xia, Y., Xu, W., Zhang, L., Shi, X., & Mao, K. (2015). Integrating 3d structure into traffic scene understanding with RGB-D data. *Neurocomputing*, 151, 700–709.
13. Wang, D., Wang, B., Zhao, S., & Yao, H. (2017). View-based 3D object retrieval with discriminative views. *Neurocomputing*, 151, 612–619.
14. Kokkonis, G., Psannis, K. E., Roumeliotis, M., et al. (2017). Real-time wireless multisensory smart surveillance with 3D-HEVC streams for internet-of-things (IoT). *The Journal of Supercomputing*, 73, 1044.
15. Santana, J. M., Wendel, J., Trujillo, A., Suárez, J. P., Simons, A., & Koch, A. (2017). Multimodal location based services—Semantic 3D city data as virtual and augmented reality. In G. Gartner & H. Huang (Eds.), *Progress in location-based services 2016*. Berlin: Springer.
16. Du, X., Allan, M., Dore, A., et al. (2016). Combined 2D and 3D tracking of surgical instruments for minimally invasive and robotic-assisted surgery. *International Journal of Computer Assisted Radiology and Surgery*, 11, 1109–1119.
17. Alaraimi, B., El Bakbak, W., Sarker, S., et al. (2014). A randomized prospective study comparing acquisition of laparoscopic skills in three-dimensional (3D) vs. two-dimensional (2D) laparoscopy. *World Journal of Surgery*, 38, 2746–2752.
18. Sørensen, S. M. D., Savran, M. M., Konge, L., et al. (2016). Three-dimensional versus two-dimensional vision in laparoscopy: A systematic review. *Surgical Endoscopy*, 30, 11–23.
19. Velayutham, V., Fuks, D., Nomi, T., et al. (2016). 3D visualization reduces operating time when compared to high-definition 2D in laparoscopic liver resection: A case-matched study. *Surgical Endoscopy*, 30, 147–153.
20. Hirschmuller, H., & Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis* (pp. 1–8).
21. Hosni, A., Bleyer, M., Rhemann, C., Gelautz, M., & Rother, C. (2011). Real-time local stereo matching using guided image filtering. In *IEEE International Conference on Multimedia and Expo* (pp. 1–6).

22. Domański, M., et al. (2015). Fast depth estimation on mobile platforms and FPGA devices. In *3DTV-Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON)* (pp. 1–4).
23. Fan, Y., Huang, P., & Liu, H. (2015). VLSI design of a depth map estimation circuit based on structured light algorithm. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 23, 2281–2294.
24. Flores-Fuentes, W., Rivas-Lopez, M., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., & Rivera-Castillo, J. (2014). Energy center detection in light scanning sensors for structural health monitoring accuracy enhancement. *IEEE Sensors Journal*, 14(7), 2355–2361.
25. Bleyer, M., & Breiteneder, C. (2013). Stereo matching—State-of-the-art and research challenges. In G. Farinella, S. Battiato, & R. Cipolla (Eds.), *Advanced topics in computer vision. Advances in computer vision and pattern recognition*. London: Springer.
26. Ding, J., Du, X., Wang, X., & Liu, J. (2010). Improved real-time correlation-based FPGA stereo vision system. In *IEEE International Conference on Mechatronics and Automation* (pp. 104–108).
27. Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.
28. Liu, X., Li, D., Liu, X., & Wang, Q. (2010). A method of stereo images rectification and its application in stereo vision measurement. In *Second IITA International Conference on Geoscience and Remote Sensing* (pp. 169–172).
29. Santana-Cedrés, D., et al. (2017). Estimation of the lens distortion model by minimizing a line reprojection error. *IEEE Sensors Journal*, 17, 2848–2855.
30. Sousa, R. M., Wány, M., Santos, P., & Morgado-Dias, F. (2017). NanEye—An endoscopy sensor with 3-D image synchronization. *IEEE Sensors Journal*, 17, 623–631.
31. Ascensão, B., Santos, P., & Dias, M. (2018). Distance measurement system for medical applications based on the NanEye stereo camera. In *International Conference on Biomedical Engineering and Applications (ICBEA)* (pp. 1–6).
32. Rodríguez-Quiñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
33. Fusiello, A., & Trucco, E. (2000). Verri, a compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12, 16–22.
34. Kumar, S., Micheloni, C., Piciarelli, C., & Foresti, G. L. (2010). Stereo rectification of uncalibrated and heterogeneous images. *Pattern Recognition Letters*, 31, 1445–1452.
35. Hamzah, R. A., Ibrahim, H., & Hassan, A. H. A. (2016). Stereo matching algorithm for 3D surface reconstruction based on triangulation principle. In *1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)* (pp. 119–124).
36. Rivera-Castillo, J., Flores-Fuentes, W., Rivas-López, M., Sergiyenko, O., Gonzalez-Navarro, F. F., Rodríguez-Quiñonez, J. C., et al. (2017). Experimental image and range scanner datasets fusion in SHM for displacement detection. *Structural Control and Health Monitoring*, 24(10), e1967.
37. Real-Moreno, O., Rodriguez-Quiñonez, J. C., Sergiyenko, O., Basaca-Preciado, L. C., Hernandez-Balbuena, D., Rivas-Lopez, M., & Flores-Fuentes, W. (2017, June). Accuracy improvement in 3D laser scanner based on dynamic triangulation for autonomous navigation system. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on* (pp. 1602–1608). IEEE.
38. Atapour-Abarghouei, A., & Breckon, T. P. (2018). A comparative review of plausible hole filling strategies in the context of scene depth image completion. *Computers & Graphics*, 72, 39–58.
39. Yoon, K. J., Member, S., & Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 650–656.

40. Hamzah, R. A., Rahim, R. A., & Noh, Z. M. (2010). Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application. In *3rd International Conference on Computer Science and Information Technology* (pp. 652–657).
41. Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 807–814).
42. Joglekar, J., Gedam, S. S., & Mohan, B. K. (2014). Image matching using SIFT features and relaxation labeling technique—A constraint initializing method for dense stereo matching. *IEEE Transactions on Geoscience and Remote Sensing*, *52*, 5643–5652.
43. Hafner, D., Demetz, O., & Weickert, J. (2013). Why is the census transform good for robust optic flow computation? In A. Kuijper, K. Bredies, T. Pock, & H. Bischof (Eds.), *Scale space and variational methods in computer vision*. Berlin: Springer.
44. Huang, F., Huang, S., Ker, J., & Chen, Y. (2012). High-performance SIFT hardware accelerator for Real-time image feature extraction. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*, 340–351.
45. Garstka, J., & Peters, G. (2015). Fast and robust keypoint detection in unstructured 3-D point clouds. In: *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)* (pp. 131–140).
46. Kechagias-Stamatis, O., & Aouf, N. (2016). Histogram of distances for local surface description. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2487–2493).
47. Prakhya, S. M., Lin, J., Chandrasekhar, V., Lin, W., & Liu, B. (2017). 3D HoPD: A fast low-dimensional 3-D descriptor. *IEEE Robotics and Automation Letters*, *2*, 1472–1479.
48. Brooks, M. J., & Horn, B. K. P. (1985). Shape and source from shading. In *Proc. Int. Joint Conf. Artificial Intelligence* (pp. 932–936).
49. Woodham, R. J. (1980). Photometric method for determining surface orientation from multiple images. *Optical Engineering*, *19*, 139–144.
50. Sohaib, A., Farooq, A. R., Atkinson, G. A., Smith, L. N., Smith, M. L., & Warr, R. (2013). In vivo measurement of skin microrelief using photometric stereo in the presence of interreflections. *Journal of the Optical Society of America. A*, *30*, 278–286.
51. Woodham, R. J. (1978). Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In *Image Understanding Systems and Industrial Applications*.
52. Mostafa, M. G., Yamany, S. M., & Farag, A. A. (1999). Integrating stereo and shape from shading. In *Int. Conf. on Image Processing* (pp. 130–134).
53. Prados, E., & Soatto, S. (2005). Fast marching method for generic shape from shading. In N. Paragios, O. Faugeras, T. Chan, & C. Schnörr (Eds.), *Variational, geometric, and level set methods in computer vision. VLSM. Lecture notes in computer science* (Vol. 3752). Berlin: Springer.
54. Lu, S., & Yuanyuan, W. (2017). Three-dimensional reconstruction of macrotecture and microtexture morphology of pavement surface using six light sources–based photometric stereo with low-rank approximation. *Journal of Computing in Civil Engineering*, *31*, 1. 2.
55. Antensteiner, D., Štöle, S., & Pock, T. (2018). Variational fusion of light field and photometric stereo for precise 3D sensing within a multi-line scan framework. In *24th International Conference on Pattern Recognition (ICPR)* (pp. 1036–1042).
56. Ju, Y., Qi, L., Zhou, H., Dong, J., & Lu, L. (2018). Demultiplexing colored images for multispectral photometric stereo via deep neural networks. *IEEE Access*, *6*, 30804–30818.
57. Hilliges, O., Weiss, M. H., Izadi, S., & Kim, D. (2018). *Using photometric stereo for 3D environment modeling*. US Patent.
58. Piatkowska, E., Kogler, J., Belbachir, N., & Gelautz, M. (2017). Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu*.
59. Zhu, A. Z., Chen, Y., & Daniilidis, K. (2018). *Realtime time synchronized event-based stereo*, arxiv. arXiv:1803.09025.

60. Censi, A., & Scaramuzza, D. (2014). Low-latency event-based visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 703–710).
61. Gallego, G., Lund, J. E. A., Mueggler, E., Rebecq, H., Delbruck, T., & Scaramuzza, D. (2018). Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 2402–2412.
62. Ieng, S. H., Carneiro, J., Osswald, M., & Benosman, R. (2018). Neuromorphic event-based generalized time-based stereovision. *Frontiers in Neuroscience*, 12, 442.
63. Martel, J. N. P., Müller, J., Conrath, J., & Sandamirskaya, Y. (2018). An active approach to solving the stereo matching problem using event-based sensors. In *IEEE International Symposium on Circuits and Systems (ISCAS)*.
64. Leroux, T., Ieng, S. H., & Benosman, R. (2018). *Event-based structured light for depth reconstruction using frequency tagged light patterns*, arxiv. arXiv:1811.10771.
65. Piatti, D., Remondino, F., & Stoppa, D. (2013). State-of-the-art of TOF range-imaging sensors. In F. Remondino & D. Stoppa (Eds.), *TOF range-imaging cameras*. Berlin: Springer.
66. Edoardo, C., Matt, F., Richard, W., Robert, K. H., & Cristiano, N. (2013). Spad-based sensors. In *TOF range-imaging cameras* (pp. 11–38). Berlin: Springer.
67. Behroozpour, B., Sandborn, P. A. M., Wu, M. C., & Boser, B. E. (2017). Lidar system architectures and circuits. *IEEE Communications Magazine*, 55, 135–142.
68. Beer, M., Schrey, O. M., Nitta, C., Brockherde, W., Hosticka, B. J., & Kokozinski, R. (2017). 1×80 pixel SPAD-based flash LIDAR sensor with background rejection based on photon coincidence. *IEEE Sensors*, 1–3.
69. Albitar, C., Graebing, P., & Doignon, C. (2007). Robust structured light coding for 3D reconstruction. In *IEEE 11th Int. Conf. on Computer Vision* (pp. 1–6).
70. Lee, D., & Krim, H. (2010). 3D surface reconstruction using structured circular light patterns. In J. Blanc-Talon, D. Bone, W. Philips, D. Popescu, & P. Scheunders (Eds.), *Advanced concepts for intelligent vision systems. ACIVS*. Heidelberg: Springer.
71. Ma, S., Shen, Y., Qian, J., Chen, H., Hao, Z., & Yang, L. (2011). Binocular structured light stereo matching approach for dense facial disparity map. In D. Wang & M. Reynolds (Eds.), *AI 2011: Advances in artificial intelligence. AI 2011. Lecture notes in computer science* (Vol. 7106). Berlin: Springer.
72. Zhao, L., Xu, H., Li, J., & Cai, Q. (2012). Binocular stereo vision measuring system based on structured light extraction algorithm. In *2012 International Conference on Industrial Control and Electronics Engineering, Xi'an* (pp. 644–647).
73. Retrieved from <https://www.aniwaa.com/best-3d-scanner/>
74. Choo, H., Ribera, R. B., Choi, J. S., & Kim, J. (2011). Depth and texture imaging using time-varying color structured lights. In *2011 International Conference on 3D Imaging (IC3D)* (pp. 1–5).
75. Pages, J., Salvi, J., Collewet, C., & Forest, J. (2005). Optimised De Bruijn patterns for one-shot shape acquisition. *Image and Vision Computing*, 23(8), 707–720.
76. Tyler, B., Beiwen, L., & Song Z. (2016). *Structured light techniques and applications*. Wiley Online Library.
77. Slys, R., Moreau, L., & Borouchaki, H. (2013). On uniqueness in triangulation based pattern for structured light reconstruction. In *International Conference on 3D Imaging* (pp. 1–6).
78. Rodrigues, M., Kormann, M., Schuhler, C., & Tomek, P. (2013). Structured light techniques for 3D surface reconstruction in robotic tasks. In R. Burduk, K. Jackowski, M. Kurzynski, M. Wozniak, & A. Zolnierok (Eds.), *8th International Conference on Computer Recognition Systems CORES*.
79. Van, L. T., & Huei, Y. L. (2018). A structured light RGB-D camera system for accurate depth measurement. *Hindawi International Journal of Optics*.
80. Geng, J. (2011). Structured-light 3D surface imaging: A tutorial. *Advances in Optics and Photonics*, 3, 128–160.
81. Choi, B.-S., et al. (2017). Pixel aperture technique in CMOS image sensors for 3D imaging. *Sensors and Materials*, 29(3), 235–241.
82. Pentland, A. P. (1987). A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 523–531.

83. Rajagopalan, A. N., Chaudhuri, S., & Mudenagudi, U. (2004). Depth estimation and image restoration using defocused stereo pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 1521–1525.
84. Oili, P., Christoph, L., Peter, F., Anke, H., Wilfried, H., Stephan, I., Christian, L., Christian, S., Peter, S., Patrick, S., Robert, S., Sebastian, V., Erwin, T., & Ingmar, B. (2017). *Focus-induced photoresponse: A novel optoelectronic distance measurement technique*, arXiv. arXiv:1708.05000.
85. Xu, X., Che, R., Nian, R., He, B., Chen, M., & Lendasse, A. (2016). Underwater 3D object reconstruction with multiple views in video stream via structure from motion. *OCEANS*, 1–5.
86. Aji, R. W., Akihiko, T., & Masatoshi, O. (2018). *Structure-from-Motion using Dense CNN Features with Keypoint Relocalization*, arXiv., arXiv:1805.03879.
87. Bianco, S., Ciocca, G., & Marelli, D. (2018). Evaluating the performance of structure from motion pipelines. *Journal of Imaging*, 4(8), 98.
88. Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE Int. Conf. on Computer Vision* (pp. 1150–1157).
89. Arjun, J., Jonathan, T., & Yann, L., & Christoph, B. (2014). *MoDeep: A deep learning framework using motion features for human pose estimation*, arXiv. arXiv:1409.7963.
90. Benjamin, U., Huizhong, Z., Jonas, U., Nikolaus, M., Eddy, I., Alexey, D., & Thomas, B. (2017). DeMoN: Depth and motion network for learning monocular stereo. *Benjamin Ummerhofer*, arXiv. arXiv:1612.02401.
91. Sudheendra, V., Susanna, R., Cordelia, S., Rahul, S., & Katerina, F. (2017). *SfM-Net: Learning of structure and motion from video*, arXiv. arXiv:1704.07804.
92. Faisal, M., & Nicholas, J. D. (2018). *Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy* (pp. 230–243).
93. Gyeongsik, M., Ju, Y. C., & Kyoung, M. L. (2018). *V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map*, arXiv. arXiv:1711.07399.
94. Liang, L., Lin, Q., Yisong, L., Hengchao, J., & Junyu, D. (2018). Three-dimensional reconstruction from single image base on combination of CNN and multi-spectral photometric stereo. *Sensors*, 18(3).
95. Marcela, C., Bertrand, L. S., Pauline, T.-P., Andrés, A., & Frédéric, C. (2018). *Deep depth from defocus: How can defocus blur improve 3D estimation using dense neural networks?* arXiv. arXiv:1809.01567.
96. Jure, Ž., & Yann, L. (2016). *Stereo matching by training a convolutional neural network to compare image patches*, arXiv. arXiv:1510.05970.
97. Luo, W., Schwing, A. G., & Urtasun, R. (2016). Efficient deep learning for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5695–5703).
98. Sameh, K., Sean, F., & Christoph, R. (2018). *StereoNet: Guided hierarchical refinement for real-time edge-aware depth prediction*, arXiv. arXiv:1807.08865.
99. Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6565–6574).
100. Zhan, H., Garg, R., & Weerasekera, C. S. (2018). *Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction*.
101. Godard, C., Aodha, O. M., & Brostow, G. J. (2017). *Unsupervised monocular depth estimation with left-right consistency clement godard*, arXiv. arXiv:1609.03677.
102. Srivastava, S., Ha, S. J., Lee, S. H., Cho, N. I., & Lee, S. U. (2009). Stereo matching using hierarchical belief propagation along ambiguity gradient. In *16th IEEE International Conference on Image Processing (ICIP)* (pp. 2085–2088).
103. Westoby, M., Brasington, J., Glasser, N. F., & Hambrey, M. J. (2012). Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, 300–314. Elsevier.
104. Yichuan, T., Ruslan, S., & Geoffrey, H. (2012). *Deep Lambertian networks*, arXiv. arXiv:1206.6445

105. Visentini-Scarzanella, M., et al. (2015). Tissue shape acquisition with a hybrid structured light and photometric stereo endoscopic system. In X. Luo, T. Reichl, A. Reiter, & G. L. Mariottini (Eds.), *Computer-assisted and robotic endoscopy*. CARE, Springer.
106. Wang, W., Yan, J., Xu, N., Wang, Y., & Hsu, F. (2015). Real-time high-quality stereo vision system in FPGA. *IEEE Transactions on Circuits and Systems for Video Technology*, 25, 1696–1708.
107. Ttofis, C., Kyrkou, C., & Theocharides, T. (2016). A low-cost Real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering. *IEEE Transactions on Computers*, 65, 2678–2693.
108. Xie, J., Girshick, R., & Farhadi, A. (2016). Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision*. Berlin: Springer.
109. Jalal, A., Kim, Y. H., Kim, Y. J., Kamal, S., & Kim, D. (2017). Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognition*, 61, 295–308. Elsevier.
110. Ma, F., & Karaman, S. (2018). Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation (ICRA), Brisbane* (pp. 1–8).
111. Li, R., Wang, S., Long, Z., & Gu, D. (2018). UnDeepVO: Monocular visual odometry through unsupervised deep learning. In *IEEE International Conference on Robotics and Automation (ICRA), Brisbane* (pp. 7286–7291).
112. Yuanhong, X., Pei, D., Junyu, D., & Lin, Q. (2018). *Combining SLAM with multi-spectral photometric stereo for real-time dense 3D reconstruction*, arxiv. arXiv:1807.02294.

Chapter 4

Design and Simulation of Array Cells of Mixed Sensor Processors for Intensity Transformation and Analog-Digital Coding in Machine Vision



Vladimir G. Krasilenko, Alexander A. Lazarev, and Diana V. Nikitovich

Acronyms

AAM	Auto-associative memory
ABC	Analog-digital basic cell
ADC	Analog-to-digital converter
AM	Associative memory
BC	Basic cell
BIA	Binary image algebra
CCCA	Current-controlled current amplifiers on current mirror multipliers
CDNE	Complementary double NE
CL	Continuous logic
CLC	Continuous logic cell
CLEM	Continuous logical equivalence model
CLF	Continuous logic function
CM	Current mirror
CMM	Current multiplier mirror
CMOS	Complementary metal-oxide-semiconductor
CNN	Convolutional neural network
DAC	Digital-to-analog converter
DC	Digital-analog cell
DOEP	Digital optoelectronic processor
EM	Equivalence model
EQ_CL	Equivalent continuous-logical
FPAA	Field-programmable analog array

V. G. Krasilenko (✉) · A. A. Lazarev · D. V. Nikitovich
Vinnytsia National Technical University, Vinnytsia, Ukraine

G	Gray
HAM	Hetero-associative memory
IP	Image processor
MAAM	Multi-port AAM
MHAM	Multi-port hetero-associative memory
MIMO	Multi-input and multi-output
MLA	Array of microlenses
NE	Neural element
NEq	Normalized equivalence
NEqs	Neuron equivalentors
NN	Neural network
NnEq	Normalized nonequivalence
NSEqF	Normalized spatial equivalence function
OE-VLSI	Optoelectronic very large scale integration
SD_NEF	Spatially dependent normalized equivalence function
SHD	Sample and hold device
SI EM AM	Spatially invariant equivalence model of associative memory
SLE CNS	Self-learning equivalent convolutional neural structure
SMC_ADC	Multichannel sensory analog-to-digital converter
TC	Transfer characteristics
TPCA	Time-pulse-coded architecture
ULE	Universal (multifunctional) logical element
VMO	Vector or matrix organization

4.1 Introduction

To create biometric systems, computer vision systems are needed to solve the problem of recognizing objects in images. There are many known methods and means to address these problems [1, 2]. In most recognition algorithms, the most frequently used operation is the comparison of two different images of the same object or its fragments. The mutual 2D function of correlating a reference fragment with the current offset image fragment is also most often used as a discriminant measure of their mutual comparison. With a strong correlation of images in their set to improve the accuracy and probability, the quality of comparison of the noisy current fragment and the reference image, as shown in [3], it is desirable to use methods of comparison, image selection, based on measures of mutually equivalent two-dimensional spatial functions transformations and adaptive correlation weightings. Various models of neural networks (NN) are actively used as a tool for image recognition and clustering. The latter is also widely used for modeling pattern recognition, biometric identification, associative memory (AM), and control of robotic devices. In [4, 5],

equivalence models (EM) of auto-associative memory (AAM) and hetero-associative memory (HAM) were proposed. The EMs have an increased memory capacity (3–4 times higher than the number of neurons) relative to the number of neurons (4096 and more) and the ability to compare, store, and retrieve to recognize strongly correlated and noisy images of large dimension, as was shown in [6–8]. These models allow you to recognize fragments (64×64 and more) with a significant percentage (up to 25–30%) of damaged pixels [5, 7, 9, 10]. Models of multi-port AAM (MAAM) and multi-port hetero-associative memory (MHAM) for associative image storage and recognition were investigated in [7, 8], the main idea of which was originally published in [4]. Mathematical models and AM implementations based on EMs were initiated in [4] and described in detail in papers [7–9], and their modifications in [11–13]. For analysis and recognition, the problem of clustering various objects must be solved. This previous clustering allows you to organize the correct automated grouping of the processed data, conduct cluster analysis, evaluate each cluster on the basis of a set of attributes, put a class label, and improve subsequent classification and recognition procedures. The significant advantages of the EM for creating MAAM and MHAM on their basis [8, 11, 12] and improved neural networks [5–9], made it possible to suggest new modifications of MHAM for parallel cluster image analysis [11, 12] and their hardware implementations on parallel structures, matrix-tensor multipliers, equivalentors with spatial and temporal integration [8, 9, 12–14]. Spatially non-invariant models and their implementation for image recognition and clustering were considered in [8, 12], and only in [1, 2, 9, 11], spatial-invariant image recognition models were considered, but not clustering. More generalized spatially invariant (SI) equivalence models (EMs) are invariant to spatial displacements and can be used for clustering images and their fragments, and therefore, the study of such models is an urgent task [14–17]. In addition, as our analysis shows, these models, described in our works [1–10] and known for more than 20 years, are very closely related to the operations of convolution of images. In the most promising paradigms of convolutional neural networks (CNN) with deep learning [18–24], the main operation is convolution. But they reveal that regularities on the basis of existing patterns or filters require complex computational procedures in their training. Jim Crutchfield of UC Davis and his team are exploring a new approach to machine learning based on pattern discovery. Scientists create algorithms to identify previously unknown structures in data, including those whose complexity exceeds human understanding. New possible ways of self-learning based on such advanced models were considered in [25]. It explained some important fundamental concepts, mechanisms of associative recognition and modeling processes of transformation and learning on the basis of understanding the principles of biological neural structures functioning. Patterns were identified in such models for binary slices of digitized multilevel images, and their implementations were proposed, and refer article [26] for multilevel images without prior binarization. But, as will be explained below, for all progressive models and concepts and nonlinear transformation of signals, image pixel intensities are necessary.

The bottleneck between the processor and the memory or processors is very slow interconnects. The increase in the integration density of devices further aggravates

the problem, since more channels are required for communication outside the crystal. The use of optical interconnects is discussed as an alternative to solve the problems mentioned. The use of optics or optoelectronics for interconnects outside the crystal and inside the crystal was demonstrated in [27]. This problem in such OE-VLSI is solved by implementing external interconnects not with the edge of the chip, but with arrays of optical detectors and light emitters, which allow implementing the stack architecture of a three-dimensional chip [28]. But in this case, the combination of various passive optical elements with active optoelectronic and electronic circuits in one system is also an unsolved problem. Intelligent detector circuits can be thought of as a subset of OE-VLSI circuits. They consist only of arrays of photodetectors, which can be monolithically integrated with digital electronics in silicon and circuits for analog-digital conversion. This greatly simplifies the design of OE-VLSI circuits, which must additionally contain only light-emitting devices, and the latter can also be implemented in silicon [29]. Such intelligent detectors with a frame [30] show a large scope and market potential. In this regard, our approach also relies on an intelligent pixel-like structure combining parallel detection of signals with parallel processing of signals in a single circuit. To realize the fastest processing, each pixel has its own analog and analog-digital node. One of the important directions for solving various scientific problems is parallel processing of large arrays (1D or 2D) of data using non-traditional computational MIMO-systems and matrix logics (multi-valued, sign, fuzzy, continuous, neuro-fuzzy, etc.) and the corresponding mathematics [31–34]. For realizations of optical learning neural networks (NN) with a two-dimensional structure [31], continuous logical equivalence models (CLEM) NN [32–34] require elements of matrix logic as well as an adequate structure for vector matrix computing procedures. Advanced parallel computing structures and processors with time-pulse signal processing [35] require parallel processing and parallel inputs/outputs. The generalization of scalar two-valued logic on the matrix case led to the intensive development of binary image algebra (BIA) [36] and logical elements of a two-dimensional array for optical and optoelectronic processors [33, 35, 37–39]. One of the promising areas of research is the use of time-pulse-coded architectures (TPCA), which were considered in papers [40, 41], which, through the use of two-level optical signals, not only increase functionality (up to universality), increase noise immunity and stability, and reduce the requirements for alignment and optical system but also simplify the control circuit and settings for the required functions and keep intact the entire methodological basis of these universal elements regardless of the length of the code word and the type of logic. Mathematical and other theoretical foundations of universal (multifunctional) logical elements (ULE) of the matrix logical design with a fast programmable setting, where the unification of functional bases is expedient, and the need to use ADC arrays were considered in [42, 43]. An ADC is a continuous-discrete automation that performs the conversion of an analog value x by its interaction with standard sizes in a discrete output code. Aspects of the theory and practice of designing and using ADCs and

DACs are so broadly outlined that it is even difficult to choose the most general works. At the same time, in the last 20–30 years, optical methods and means for parallel digital image processing have been intensively investigated, which, unlike analog ones, have high accuracy and a number of significant advantages. Certain success has been achieved in the field of creating two-dimensional matrix logic devices, storing image-type data for such parallel information systems and digital optoelectronic processors (DOEP) [38] with a performance of 10^{12} – 10^{14} bits per second. Most vector-matrix and matrix-matrix multipliers [39] use 1D linear and 2D matrix multichannel ADC [43–45]. A bottleneck in parallel DOEP is an ADC, which, unlike traditional input systems with scanning or sequential parallel reading and output, must in parallel fully perform ADC conversion of a large number of signals (256×256 pixels or more) and provide an input speed of up to 10^6 – 10^7 frames per second. Therefore, there is a need of multichannel ADC with several parallel inputs and outputs, with vector or matrix organization (VMO) [43–45], and the channels operate in parallel and provide low power consumption, simple circuit implementation, short transformation time, low input level, acceptable word length, etc. In addition, such a VMO ADC can also perform other functions, for example, the computation of logical matrix operations, digital filtering, and digital addition-subtraction of images. For multichannel multisensor measuring systems, especially for wireless ones, ADCs with very low consumption and high accuracy and speed are required. In papers [46–52] design of ADCs, current comparators, and their applications were considered. But these comparators are very high speed, consist of many transistors, and have high consumption power. Equivalent (EQ) continuous-logical (CL) ADC, which was considered in [43–45, 51], provided high performance with a smaller amount of equipment since it consisted of n-series-connected analog-digital basic cells (ABC). Such cells implement CL-functions on CMOS-transistors operating in the current mode. The parameters and performance of such CL ADCs, including the type of output codes, are influenced by the selection of the required continuous logic functions for the analog-digital conversion and the corresponding ABC scheme. The simplicity of these CL ADCs makes it possible to realize a significant number of multichannel converters for optical parallel and multisensor systems. The proposed CL ADC schemes are more preferable specifically for such applications where parallelism and large size arrays are required. Based on the above, the purpose of our chapter is to design and simulate various variants of the technical implementation of continuously logical basic cells (CL BC) based on current mirrors (CM) and multichannel sensory analog-to-digital converters (SMC ADC) with the functions of preliminary analog and subsequent analog digital processing for image processors (IP) and sensor systems. In addition, in our previous works, the accuracy characteristics of the ADC were not considered, and no estimates of conversion errors were made for different possible modes and modifications of such basic cells and ADCs as a whole. That is why the purpose of the present work is also to evaluate ADC errors, demonstrate them by specific experimental results, and also further enhancements of such ADCs and their basic cells, which significantly expand their functionality and the range of problems they solve.

4.2 Simulation of Array Cells for Image Intensity Transformations and Theoretical Mathematical Background

4.2.1 Substantiation of the Need to Design Devices for Parallel Nonlinear Image Intensity Transformations in Self-Learning Equivalent Convolutional Neural Structures (SLECONS)

For neural networks and associative memory systems, generalized equivalence models using the functions of nonlinear normalized equivalences of matrices and tensors were developed. They use spatially dependent normalized equivalence functions (SD_NEF) [6], which are defined as:

$$\tilde{\mathbf{e}}(A, B) = \frac{A \tilde{*} B}{I \times J} = \frac{1}{I \times J} \sum_{i=1}^I \sum_{j=1}^J (a_{\zeta+i, \eta+j} \sim b_{i,j}) \quad (4.1)$$

where $\tilde{\mathbf{e}} = [e_{\zeta, \eta}] \in [0, 1]^{(N-I+1) \times (M-J+1)}$ and symbol $(\tilde{*})$ indicate a spatial convolution, but with an element-wise operation of not multiplication, but “equivalence.” In accordance with the principle “the strongest survives” and the strengthening of the nonlinear action of the components, depending on the level of their values, the elements of the matrix $\tilde{e}(\tilde{e})$ and of other intermediate SD_NEF are transformed using of auto-equivalence operations [13] with different parameters p_1, p_2 . The higher the parameters p_1, p_2 in p -step auto-equivalence, that is, the more “competing” nonlinear transformations, the faster the process of recognition and state stabilization, as studies show with the help of energy equivalent functions [6, 26, 53]. The number of iterations necessary for successful recognition depends on the model parameters and, as experiments show, is significantly smaller compared to other models and does not exceed just a few. Changing the parameters p_1, p_2 it is possible to obtain all the previously known EMs [5–13]. To implement the proposed new subclass of associative neural systems, certain new or modified known devices are needed that can calculate the normalized spatial equivalence functions (NSEqFs) with the necessary speed and performance [1, 6, 10]. We called such specialized devices the image equivalentors [4, 6, 9, 10, 13], which are, in essence, a doubled correlator [54] or a doubled convolver. For the input image S_{inp} , learning array matrix \mathbf{A} , which is a set of reference images, the general SI EM AM is proposed [11, 14, 15] and modeled, where after the first and the second steps of the algorithm, the element-wise equivalence convolution and nonlinear transformations were calculated. Research results of the generalized SI EM AM confirmed advantages and improved characteristics and the possibility to recognize with interferences up to 20–30% [14, 15]. Works [12, 14] described a clustering method based on the simultaneous calculation of the corresponding distances between all cluster

neurons and all training vectors using such MAAM and MHAM. As metrics, we use generalized nonlinear normalized vector equivalence functions, which gives good convergence and high speed of models, see papers [12, 14–17]. An iterative learning process that uses a learning matrix and consists of calculating the optimal set of weight vectors for all cluster neurons is described by the proposed model. Optimal patterns are formed by such an iterative procedure, based on the search for patterns and tangible fragments of objects that are in the set of trained images. Patterns of recognition and clustering of images that combine the learning process with the process of recognition are proposed in [25, 26]. For our EMs and all known convolutional neural networks, it is necessary to determine the convolutions of a large number of patterns from the set of standards with the current image fragment in each layer in the learning process. Large images require a large number of filters for image processing, as studies show, and the size of filters, as a rule, is also large. Therefore, the acute problem is to significantly improve the computational performance of such CNN. Therefore, the last decade was marked by the activation of work aimed at creating specialized neural accelerators, which calculate the comparison function of 2 two-dimensional arrays and use the multiplication and addition accumulation operations. Unlike most papers, in our works, we use those functions of normalized equivalence in which there is no multiplication operation. But, as our studies show, equivalent models also allow us to construct convolutional equivalence structures and self-learning systems. Therefore, using our approaches to the construction of a one-dimensional neuron equivalentor [55–58], we considered the structure of a two-dimensional neuron equivalentor, generalized for processing two-dimensional arrays. The block diagram of the main unit of self-learning equivalent convolutional neural structures (SLECNS) [26] is shown in Fig. 4.1a. The required number of convolutions $e_0 - e_{n-1}$, depending on the number of filter templates $\mathbf{W}_0 - \mathbf{W}_{n-1}$, is formed from the matrix \mathbf{X} . Convolutions are represented by matrices with multilevel values, unlike binary ones, which we used earlier. Each filter is compared with the current fragment of the matrix, and equivalent measures of proximity or other measures, such as a histogram, are used as a measure of the similarity of the fragment of the matrix \mathbf{X} and the filter. Therefore, interpretation method for spatially invariant case requires the calculation of spatial features convolution-type $\mathbf{E}^m = \mathbf{W}^m \overset{t}{\otimes} \mathbf{X}$, where $E_{k,l}^m = 1 - \text{mean} \left(\left| \text{submatrix}(\mathbf{X}, k, k+r_0-1, l, l+r_0-1) - \mathbf{W}^m \right| \right)$, nonlinear processing by the expression $\text{EN}_{k,l}^m = G(a, E_{k,l}^m) = 0,5 \left[1 + (2E_{k,l}^m - 1)^a \right]$, and comparing each other to determine the winners for indexing expressions: $\text{MAX}_{k,l} = \max_{\text{index } m} \left(\text{EN}_{k,l}^{m=0}, \text{EN}_{k,l}^{m=1} \dots \text{EN}_{k,l}^{m=M-1} \right)$ and $\text{EV}_{k,l}^m = f_{\text{nonlinear}}^{\text{activ}} \left(\text{EN}_{k,l}^m, \text{MAX}_{k,l} \right)$. Thus, in the first and second steps, it is necessary to calculate a large number of convolutions.

From the above formulas, it follows that it is necessary to calculate the average value of the component-wise difference of two matrices. Similarly, normalized nonequivalent functions for all filters are calculated, and their components are nonlinear transformed: $\text{EN}_{0,k,l} = 0.5[1 + (2\text{E}0_{k,l} - 1)^\alpha]255$, where α is the

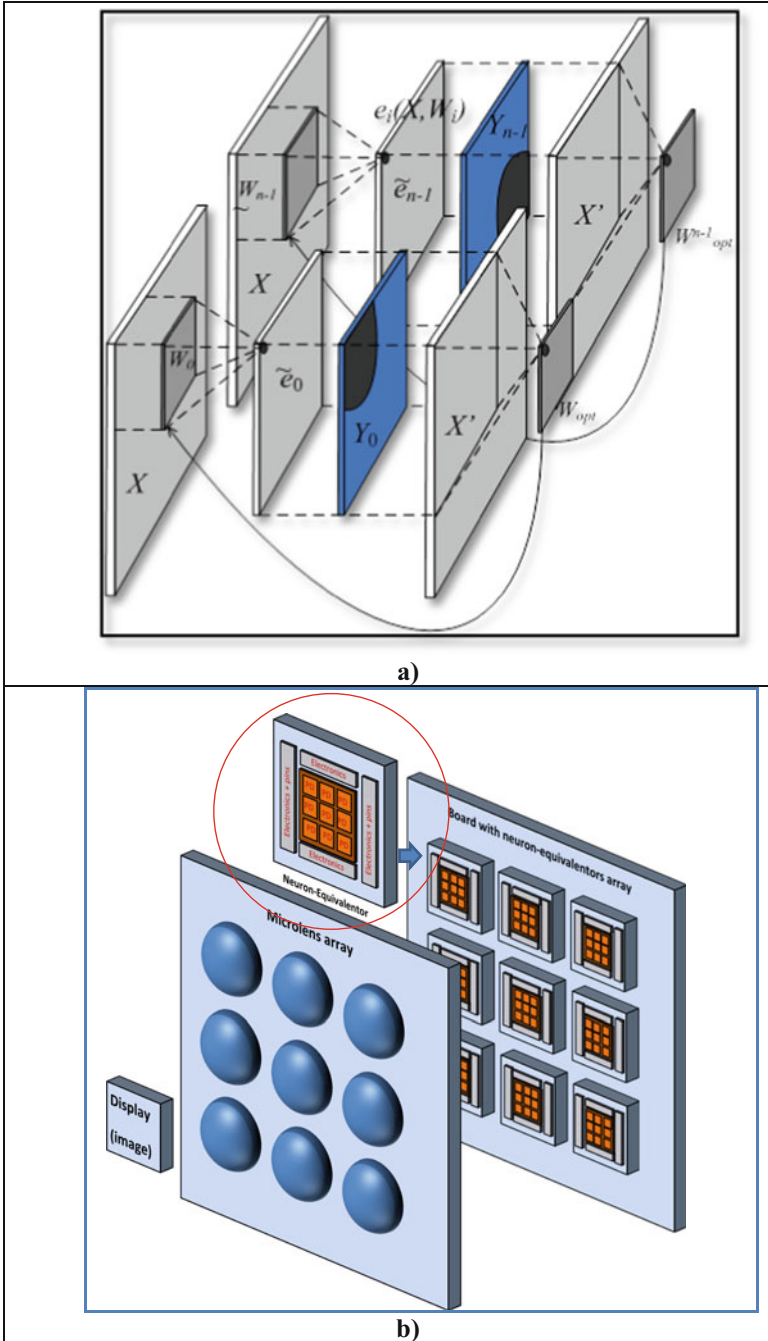


Fig. 4.1 The basic unit (structure) of the SLECONS, which explains the principle of its functioning of learning neural network model based on the multi-port memory to find centroid cluster elements (a); the basic unit that uses an array of neuron equivalentors (b)

nonlinearity coefficient. Based on these transformations for SLECNS, we need to implement nonlinear transformations for different α . The analysis of this expression shows that it is necessary to rise to power and multiply, so we propose to approximate this dependence, for example, by three-piece linear approximation. The experiments conducted in work [26] show great promise of the proposed methods and models of self-learning recognition of images, including multilevel and color images. But for their work in real time, taking into account the large requirements for performance and the amount of calculations, it is necessary to have appropriate high-performance and energy-efficient arrays and image processors with parallel principles of operation and picture input-outputs, whose design was partially considered in papers [9, 12, 13, 58–64]. In Fig. 4.1b a new structure proposed by us in [65] is shown, which may be promising for use in machine vision and artificial intelligence, neural structures, in various high-performance measurement systems [66–68]. The presented structure makes it possible to calculate in parallel the set of all components (elements) of the equivalent convolution at once in a single cycle at high speed. The cycle time is equal to the time of selection from the processed image of its shifted current fragment. The structure of the system that uses an array of neuron equivalents consists of a microdisplay that dynamically displays the current fragments, an optical node as an array of microlenses (MLA) with optical lenses (not shown!), and a two-dimensional array of equivalentors (Eqs) with optical inputs. Each Eq is implemented in a modular hierarchical manner and can consist of similar smaller sub-pixel, also 2D-type, base nodes. The equivalentor has a matrix of photodetectors, on which a halftone image of a fragment is projected using (MLA). The number of electrical analog inputs is equal to the number of photodetectors, to which the filter components are fed from a sampling and holding device (SHD) or analog memory with subsequent digital-analog conversion using any known method.

These components are presented in the form of microcurrents. Each equivalentor has its own filter mask from the filter set, which is formed as a result of training. Simulation on 1.5 μm CMOS transistors in various modes showed that such equivalentors and their basic blocks can operate in low-power and high-speed modes, their energy efficiency is estimated to be at least 10^{12} an. op/sec per watt and can be increased by an order of magnitude, especially considering FPAA [69]. But much depends on the accuracy of the current mirrors and their characteristics. Thus, at the inputs of each equivalentor, there are two arrays (vectors) of analog currents representing the current fragment being compared and the corresponding filter standard, and the output of the equivalentor is an analog current signal that is nonlinearly transformed in accordance with the activation function and represents a measure of their similarity, proximity. Also, as have been shown [65], nonlinear component-wise transformations allow even without WTA network to allocate the most NEs with the greatest activity. From the above described, it follows that for hardware implementations of all the advantages of SIEM, an important issue is the design of parallel nonlinear transformations, transformations of intensity levels. And, as will be shown below, the use of an array of cells that perform hardware and not with PC, nonlinear transformations adequate to auto-equivalence operations, allows the laborious computational process of searching for extremums in SD_NEF (maps

for clustering and learning) not to be performed, but to automatically select these extremums using only several transformation steps and eliminate all unnecessary levels, making these level pixels neutral for subsequent algorithmic steps.

4.2.2 Brief Review and Background of Mathematical Operators, Which Are Implemented by Neurons

Almost all concepts, models, and structures of NN and CNN use informational mathematical models of neurons, which are reduced to the presence of two basic mathematical component operators: the first component computes a function from two vectors \vec{X} and \vec{W} , where \vec{X} is the vector of input signals of a neuron, \vec{W} is the vector of weights, and the second component corresponds to some nonlinear transformation of the output value of the first component to the output signal. The input operator can be implemented as sum, maximal or minimum value, a product of the self-weighted inputs [55, 56]. But recently, the set of such operators has expanded significantly [6, 9, 13, 56]. Equivalence models of neural networks, which have some advantages, require the computation of such operators: normalized equivalence (NEq), nonlinear normalized nonequivalence (NnEq), and autoequivalence of vectors. In [9, 13, 57], we considered how to implement these input operators for the case when the components of the vectors \vec{X} and \vec{W} are both normalized and unipolarly encoded. In work [58], we used just normalized equivalences, but time-pulse coding was used for analog signals. The positive aspect of that work was the use of a modular principle that allowed the calculation of the operator of the normalized equivalence of a vector to the calculation of normalized equivalent subvectors and their output signals. In paper [58], the mathematical basis of the creation of neurons of equivalents calculating the function of NEq is described in detail, using the modular principle. To increase the number of inputs of our complementary double NE (CDNE) or the dimension of the compared vectors, you can use the combination of the basic analog CDNE of a smaller dimension. This greatly expands the functionality of such a basic CDNE, especially when they are combined in complex hierarchical structures. It shows that all algorithmic procedures in the equivalence paradigm of NNs and AM on their basis are reduced to the calculation of NEqs from two vectors or matrices, and the elemental nonlinear transformations that correspond to the activation functions, and for the above EMs of NNs, reduce to the calculation of auto-equivalences (auto-non-equivalences). But in the above works, activation functions were not simulated and shown. A lot of work has been devoted to the design of hardware devices that realize the functions of activation of neurons, but they do not consider the design of exactly the auto-equivalent transformation functions for EMs and the most common arbitrary types and types of nonlinear transformations. Therefore, the goal of this paper is the design of cells for hardware parallel transformation of image intensity levels. In work [65], the question of the simplest approximations of auto-equivalence functions (three-

piece approximation with a floating threshold) was partially solved. The basic cell of this approximation consisted of only 18–20 transistors and allowed to work with a conversion time from 1 to 2.5 μ s. At the same time, the general theoretical approaches to the design of any nonlinear type of intensity transformation were not considered, and this is the objective of the paper. The operations of addition and subtraction of currents are most easily performed on current mirrors.

4.2.3 *Mathematical Models of Nonlinear Transformations of Image Intensities*

The input analog intensity of the pixel is denoted by x , where $x \in [0, D]$, D is the maximum intensity of the selected range, and denotes the output analogized transformed intensity by y , where $y \in [0, D]$. Then the operator of the nonlinear intensity transformation can be written in the form: $y = F_{\text{trans}}(x)$. As such functions can be threshold processing functions, exponential, sigmoid, and many others, which, in particular, are used as activation functions in the construction, synthesis of neural elements and networks are based on them. To form the required nonlinear intensity transformations, it is possible to use a piecewise linear approximation of the chosen functions. For a piecewise-linear approximation, break the range of input levels D into N equal sub-bands, width $p = D/N$. Using the function of bounded difference known from papers [6, 13], defined as $a \dot{-} b = \begin{cases} a - b, & \text{if } a > b \\ 0, & \text{if } a \leq b \end{cases}$. Form for the input signal x and each upper sub-band level $pD_i = i \cdot p$, where $i = 1 \dots N$, the following signals: $s_i = (x \dot{-} (i - 1) p) \dot{-} (x \dot{-} i \cdot p)$. For $i = 1$ we get $s_1 = x \dot{-} (x \dot{-} p)$, and this is the minimum $\min(x, p)$, and there is a step signal with height p . For $i = 2$ we get $s_2 = (x \dot{-} p) \dot{-} (x \dot{-} 2p)$, which corresponds to a step in height p , but which begins at p . For $i = N$ we get $s_N = (x \dot{-} (N - 1) p) \dot{-} (x \dot{-} N \cdot p) = (x \dot{-} (N - 1) p)$, which corresponds to a step in height p , but which begins at $(N - 1)p = D - p$. Summing with the weight coefficients k_i of these steps, we can form a piecewise approximated intensity.

$$y_a = \sum_{i=1}^N k_i s_i = \sum_{i=1}^N k_i [(x \dot{-} (i - 1) p) \dot{-} (x \dot{-} i \cdot p)], \quad (4.2)$$

For forming $y_a \in [0, D]$, that is, the normalized range of its levels, the weighting coefficients of the steps are selected from the condition: $\sum_{i=1}^N k_i = N$. Analysis of formula (4.1) shows that by changing the gain of the steps, we can form any required piecewise continuous intensity conversion function. If the coefficient k_i is negative, it means that the corresponding step is subtracted. Thus, in order to implement the transformations, a set of nodes for realizing operations of bounded difference, weighting (multiplication), and simple summation are needed. If the input pixel intensity is set by the photocurrent, then having the current mirrors (CM), by which

the operations of the limited difference and the summation of the photocurrents are easily realized, it is sufficient to have a plurality of limited difference schemes and the specified upper sub-band levels pD_i . By choosing the parameters of the current mirror transistors, operations of dividing or multiplying are currents by the required fixed k_i . If it is necessary to dynamically change the view, the conversion function, that is, the weight of the components, then you need the coded amplifiers. When working with currents and CM, a set of keys and a multiplying mirror with discrete weights (binary) perform the role of code-controlled amplifiers and are essentially DAC with the only difference that instead of a reference analog signal, an analog signal s_i . After some transformations, formula (4.2) is transformed into this form:

$$y_a = \sum_{i=1}^N k_i [(x \dot{-} pD_{i-1}) \dot{-} (x \dot{-} pD_i)] = \sum_{i=1}^N k_i \cdot \min(x \dot{-} pD_{i-1}, p) \quad (4.3)$$

Formula (4.3) indicates that for the implementation of the intensity conversion, it is necessary to have analogous minimum circuits, but it is realized in the form of two operations of bounded difference: $a \dot{-} (a \dot{-} b) = \min(a, b)$. In addition to the formulas (4.2) and (4.3) considered above, it is possible to realize the required function by means of triangular signals:

$$y_a = \sum_{i=1}^N k_i \cdot t_i = \sum_{i=1}^N k_i [(x \dot{-} (i-1) \cdot p) \dot{-} 2(x \dot{-} i \cdot p)] \quad (4.4)$$

For the formation of the constants s_i or t_i , the input signal x can be multiplied by N , and then all components are simultaneously generated simultaneously in each sub-assembly. On the other hand, in each sub-assembly, a signal $(x \dot{-} pD_i)$, this is fed to the next in the pipeline sub-assembly for the formation of signals and components from it. This corresponds to a conveyor circuit that will have a large delay but does not require the multiplication of the input signal. The choice of this or that scheme and element base depends on the requirements for the synthesized node.

4.2.4 Simulation of Array Cells for Image Intensity Transformation

4.2.4.1 Simulation of Image Intensity Transformation with Mathcad

Using both the basic components for the composition of the lambda function **fspΔs2**, shown in Fig. 4.2 and described by expression:

$$\mathbf{fsp\Delta s2}(\mathbf{xs}, \mathbf{p\Delta x}, \mathbf{p\Delta}, \mathbf{k}) = \mathbf{k} \times \mathbf{obs}(\mathbf{obs}(\mathbf{xs}, \mathbf{p\Delta x}), \mathbf{obs}(\mathbf{xs}, \mathbf{p\Delta}) \times 2) \quad (4.5)$$

where $\mathbf{x}s$ is the function argument, $\mathbf{p}\Delta\mathbf{x}$ is the parameter indicating the lower bound level $\mathbf{x}s$ (beginning), $\mathbf{p}\Delta$ is the second parameter indicating the level for the maximum, k is the third parameter indicating the scalar gain multiplier; and $\text{abs}(a, b) = a^{-b}$, we proposed a function composition $\mathbf{fsp}\Delta\mathbf{sS}$, which is calculated by the expression:

$$\mathbf{fsp}\Delta\mathbf{sS}(\mathbf{x}s, \Delta\mathbf{k}, \mathbf{VK}) = \sum_{i=1}^{\Delta\mathbf{k}} \mathbf{fsp}\Delta\mathbf{s2} \left[\mathbf{x}s, \frac{255}{\Delta\mathbf{k}} \times (i - 1), \frac{255}{\Delta\mathbf{k}} (i), \mathbf{VK}_i \right] \quad (4.6)$$

where $\Delta\mathbf{k}$ is the number of components (lambda functions), $\mathbf{x}s$ is the argument of the function, and \mathbf{VK} is the vector of gain factors. The result of constructing some types of transfer characteristics (TC) using these functions in the Mathcad environment is shown in Fig. 4.2. To approximate auto-equivalence, we also offer simpler (two-step) basic N -functions:

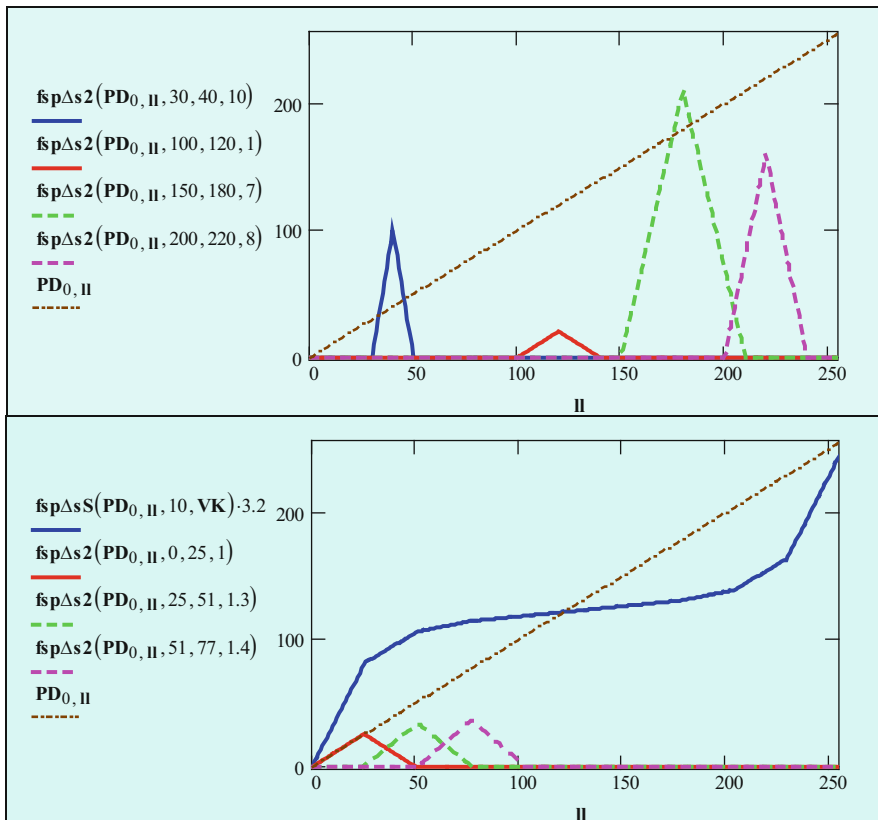


Fig. 4.2 Graphs of synthesized transformation functions

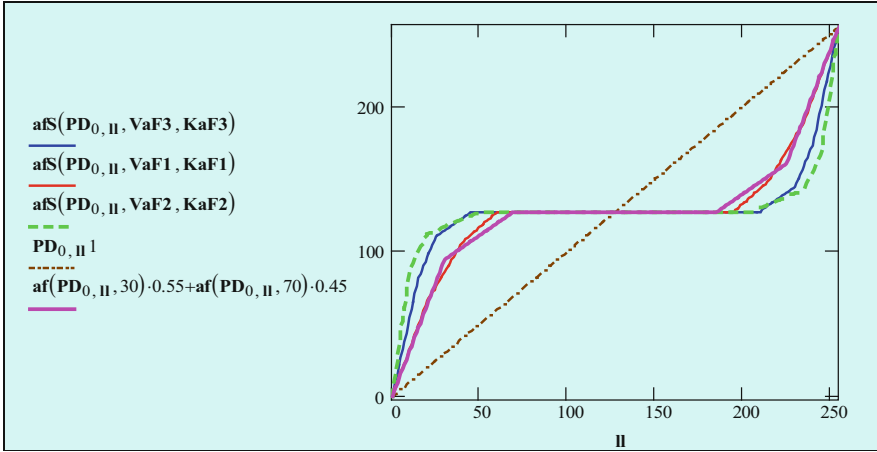


Fig. 4.3 Examples of synthesized transfer characteristics for auto-equivalence functions

$$\mathbf{af}(\mathbf{xs}, \mathbf{xp}) = [\mathbf{obs}(\mathbf{xs}, \mathbf{obs}(\mathbf{xs}, \mathbf{xp})) + \mathbf{obs}(\mathbf{xs}, (\mathbf{DP} - \mathbf{xp}))] \cdot \left(\frac{\mathbf{DP}}{\mathbf{xp} \cdot 2} \right) \quad (4.7)$$

and triple their composition:

$$\mathbf{afS}(\mathbf{xs}, \mathbf{VaF}, \mathbf{KaF}) = \sum_{i_v=0}^2 \mathbf{af}(\mathbf{xs}, \mathbf{VaF}_{i_v}) \cdot (\mathbf{KaF}_{i_v}) \quad (4.8)$$

In general, the number of components in a composition can be arbitrary, but for modeling we used 8- and 16-component compositions and adjustment vectors. Examples of such functions and compositions for the synthesis of TC are shown in Fig. 4.3. Another variety of functions is shown in Fig. 4.4, and the results of using such TCs to prepare the original PIC image are shown in Fig. 4.5.

4.2.4.2 Design and Simulation of Array Cells for Image Intensity Transformation Using OrCad PSpice

Let us first consider the design and simulation of a single base cell for the image intensity of an arbitrary transformation, using the example of a four-piece approximation by triangular signals according to formula (4.4). Figure 4.6 shows the scheme used for modeling, and Fig. 4.7 shows the schematic of the basic sub-node. To form four triangular signals from the input signal, we use four identical sub-nodes, each of which consists of 14 (13) transistors and an additional current mirror (two transistors Q18 and Q19), and for propagation of the input photocurrent and threshold levels, the auxiliary circuit consists of 17 (14) transistors.

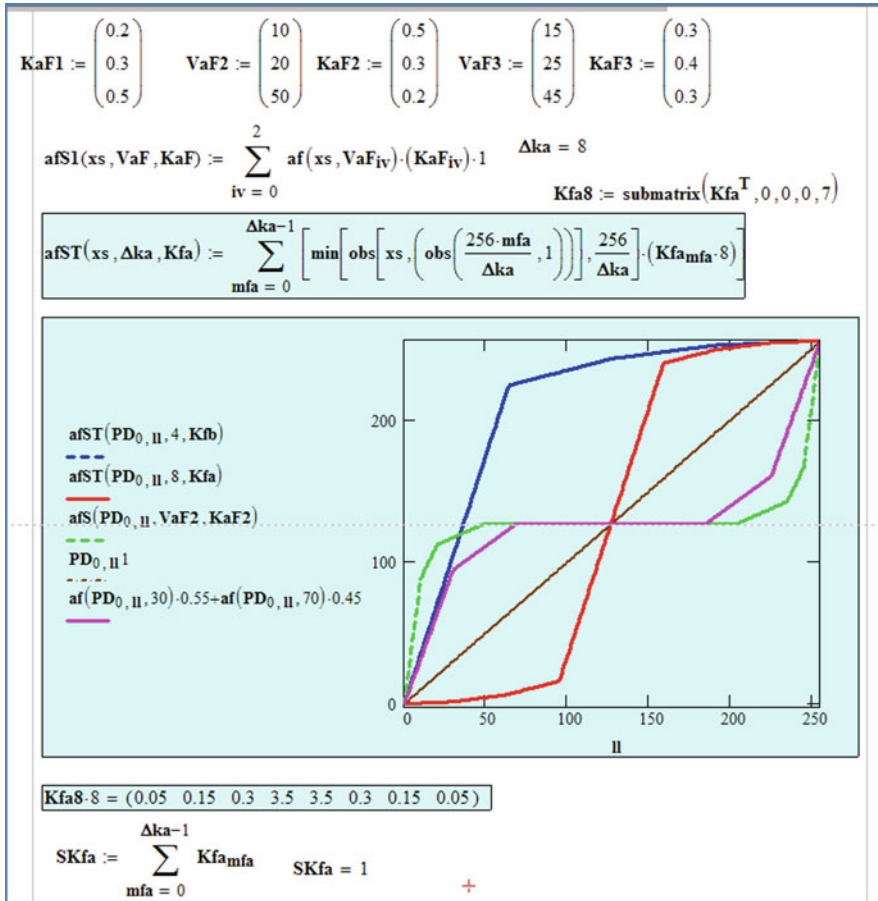


Fig. 4.4 Mathcad windows with the formulas and graphs of synthesized transformation functions

The input photocurrent was simulated by a current generator I2. In general, the cell layout consists of 68 transistors. In this scheme for simulation, we used four fixed different gain values for each triangular signal. To do this, the output signal of the sub-node was multiplied using the current multiplier mirror (CMM), and by fixing the output connections S0–S3 with the summing output current mirror or the power line, we chose the weights k_i . Thus, we modeled different transformation functions by choosing a set of coefficients k_i . The simulation results for various input signals are shown in Fig. 4.8.

Using a linearly increasing input signal (red solid line) and a conversion function, the form of which is shown in the green bold line in Fig. 4.8a, and using auxiliary signals (shown in different colors), we obtained a nonlinear transformation similar to the ReLu function (with saturation). In Fig. 4.8b, the resultant signal (green bold line) is shown after a nonlinear conversion by means of this function of the input

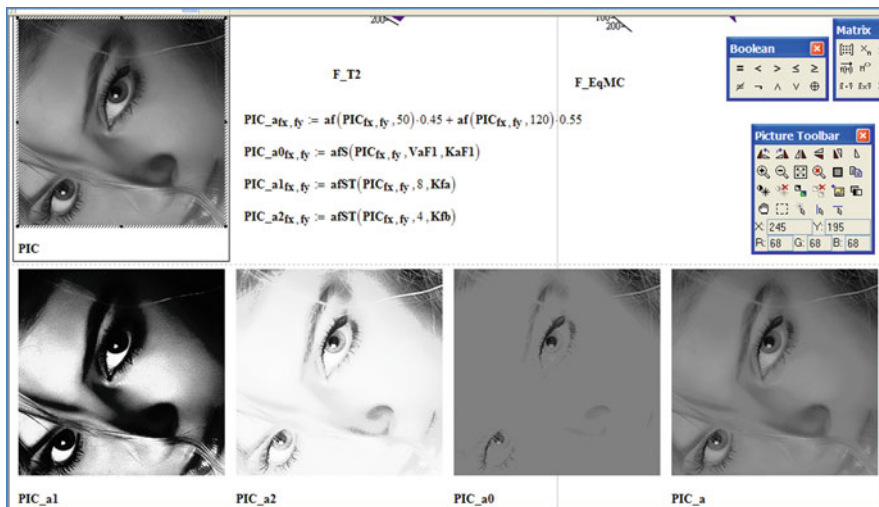


Fig. 4.5 Mathcad windows on which the formulas and results of image intensity transformation are shown, wherein 2D from left to right: input image PIC, the computed auto-equivalence functions, nonlinear (after activation) output images (bottom row)

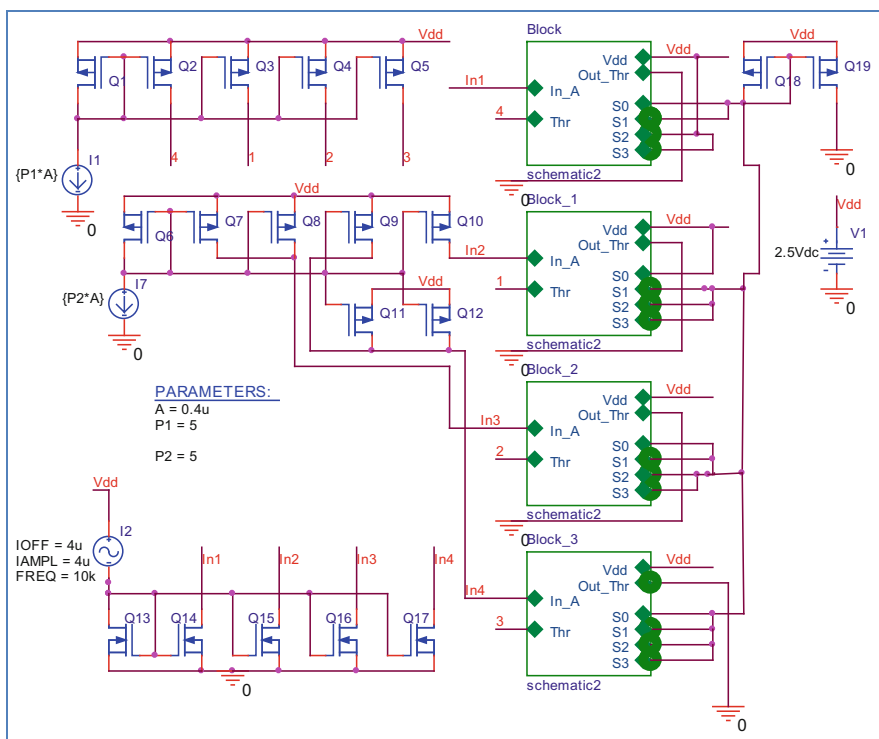


Fig. 4.6 Circuit for simulation of nonlinear converter cell on the base of four-piece linear approximation and four base sub-nodes

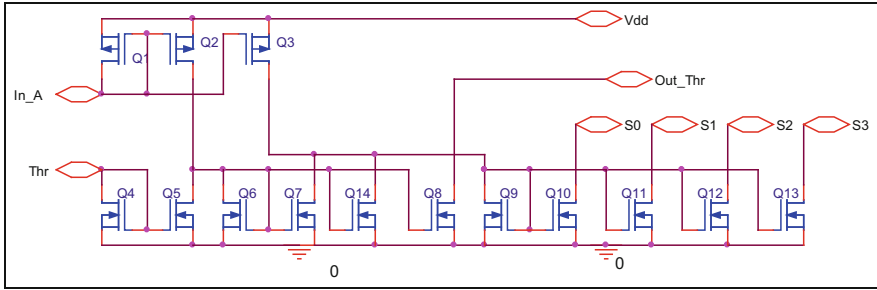
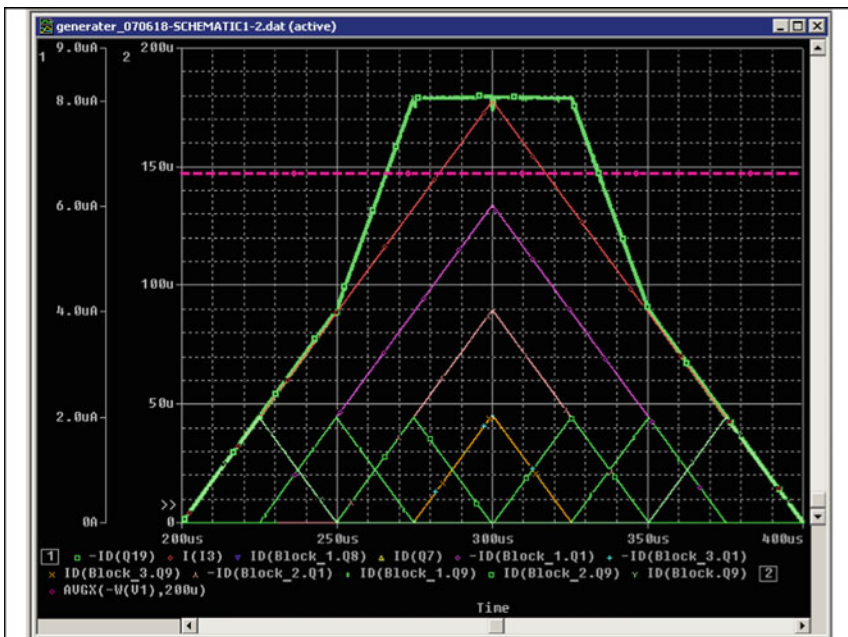


Fig. 4.7 Circuit of base sub-node (schematic 2) for four-piece linear approximations

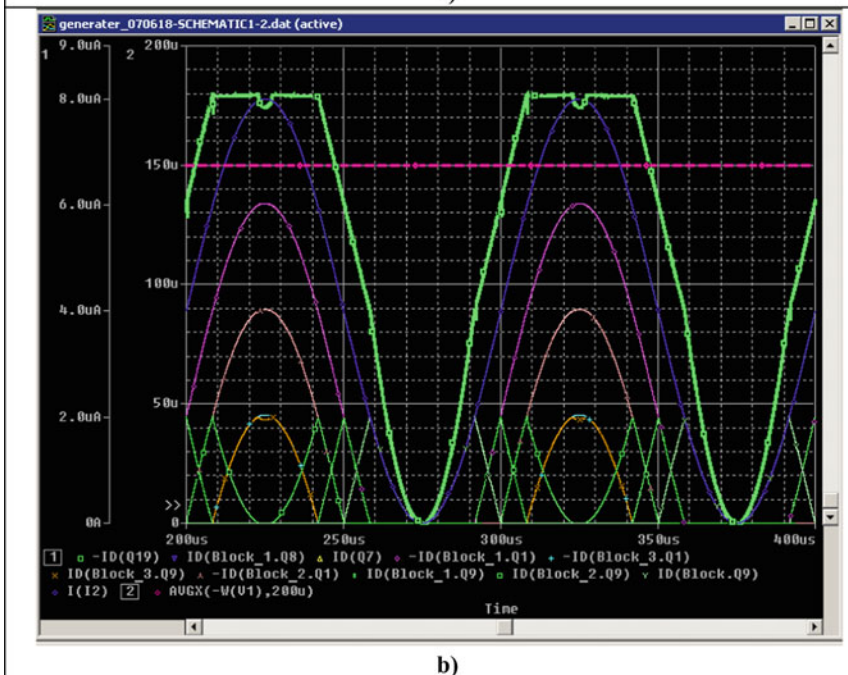
sinusoidal signal (shown in blue). The power consumption of the cell is $150 \mu\text{W}$ at a supply voltage of 2.5 V , $I_{\text{max}} = D = 8 \mu\text{A}$, $N = 4$, $p = 2 \mu\text{A}$, and the periods of the input signals are 200 and $100 \mu\text{s}$. To dynamically switch the view of the image pixel intensity conversion function, we use the current-controlled current amplifiers on current mirror multipliers (CCCA) with binary-weighted current outputs (Fig. 4.9). The general scheme of the cell realizing the dynamic intensity conversion with eight piecewise linear approximation is shown in Fig. 4.10. This circuit contains $170\text{--}200$ transistors and consists of eight basic nodes ($A + \text{CCCA}$). The Node A consists of 8 (7) transistors and generates a triangular signal from the input signal at a given threshold for each sub-band pD_i . The auxiliary circuits for generating upper sub-band levels and subtracting them from the input signals are shown at the left in Fig. 4.10 and can be implemented in different ways depending on the selected element base and approach. The processes of formation from the input signal of all auxiliary components, triangular waveforms, nonlinearly transformed output signal, and simulation results of this circuit for different modes are shown in Figs. 4.11 and 4.12. For a supply voltage of 2.5 V , $I_{\text{max}} = D = 8 \mu\text{A}$, $N = 8$, $p = 1 \mu\text{A}$ and the period of the input linearly increasing-decreasing triangular signal equal to $1000 \mu\text{s}$.

Removing only one transistor in node A of the circuit in Fig. 4.10 allows it to modify and implement on the basis of tunable nonlinear transformations in accordance with the formula (4.1), and not (4.3), that is, with the help of s_i , but not t_i .

The results of modeling such as conversion scheme with the composition of the basic step signals s_i are shown in Figs. 4.13, 4.14, and 4.15, and Figs. 4.14 and 4.15 show the case of four-level approximation and Fig. 4.13 the eight-level approximation. The results confirm the possibility of synthesizing converter cells with specified or required accuracy characteristics of the transformation laws and, in particular, auto-equivalence functions, the microvolt level of power consumed by them, and high speed (microseconds and less). For the simplest and approximate approximation functions, but often quite sufficient for the selection of the winning function by the activation function, the cell circuits consist of only $17\text{--}20$ transistors, have a very high speed ($T = 0.25 \mu\text{s}$), and a small power consumption (less than $100 \mu\text{W}$). The results of simulating such simple ($3\text{--}4$ piecewise approximation) cells



a)



b)

Fig. 4.8 Simulation result for the circuit in Fig. 4.6 for input linear rising signal (a) and for input sinusoidal signal (b)

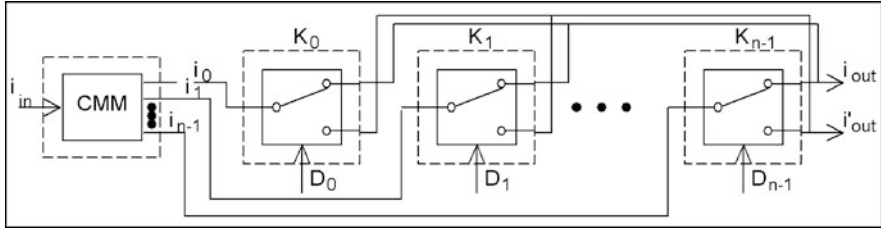


Fig. 4.9 Code-controlled current amplifier (CCCA) that consists of current mirror with multiplication (CMM) and set of n keys (K)

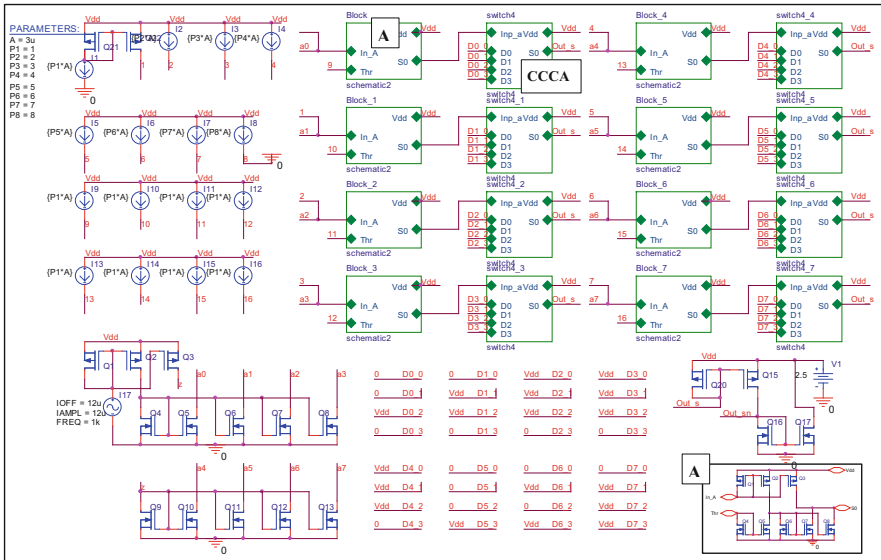


Fig. 4.10 Circuit for simulation of nonlinear converter cell on the base of eight-piece linear approximation and eight base sub-nodes

(see Fig. 4.16) separately and in the composition with nodes for input operators, and in small-sized networks of Eq equivalents are presented in Sect. 4.2.4.3 and are shown in Figs. 4.17 and 4.18. The analysis of the obtained results confirms the correctness of the chosen concept and the possibility of creating CLCs for image intensity transformation and MIMO structures on their basis, as hardware accelerators for compact high-performance systems of machine vision, CNN, and self-learning biologically inspired devices.

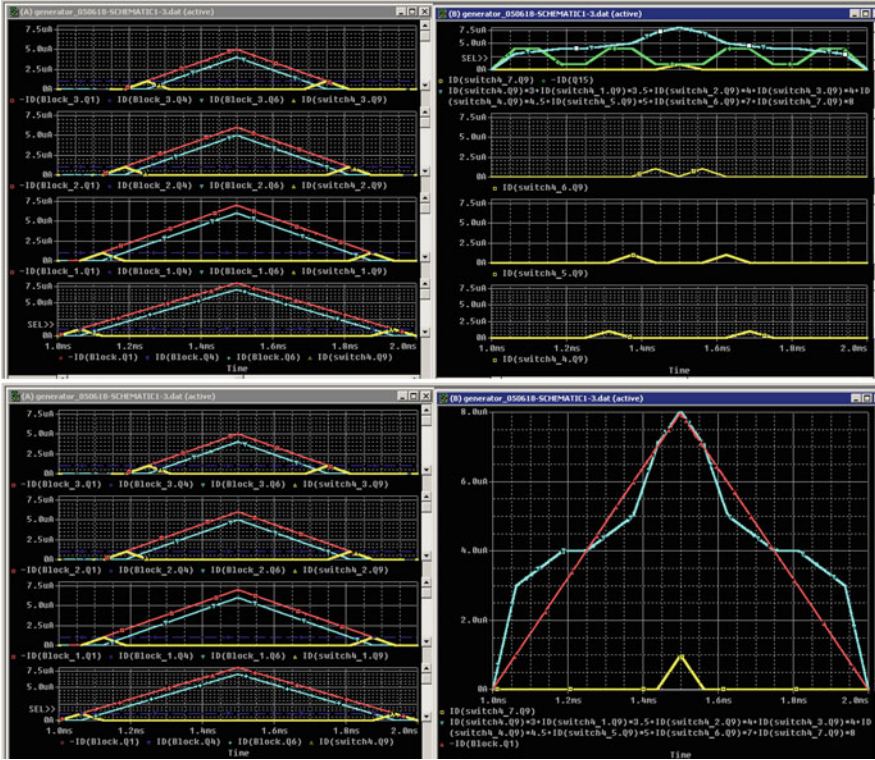


Fig. 4.11 Simulation result for eight sub-node circuit (Fig. 4.10): up left, formation of triangle signals for linear rising input signal (red line), output signal (yellow line) (the first four signals); up right, formation of triangle signals (red line), output signal (yellow line) (the second 4 signals) and two outputs for two characteristics (blue and green lines); down right, input signal (red line), output signal (blue line)

4.2.4.3 Simulation of Nonlinear Transformation in Analog 64-Input and 81-Input Neuron Equivalentor

For the simulation of nonlinear transformation in analog 64-input and 81-input neuron equivalentor [65], we used a node whose circuit is shown in Fig. 4.16, which forms the activation function (autoequivalence) in the form of a piecewise linear approximation. Simulating results of such 64-input NE with the nonlinear conversion of the output signal response for linearly rising (falling) currents with a period $T = 2.5 \mu s$ are shown in Figs. 4.17 and 4.18. In the same place, the results of modeling the formation processes of linear and nonlinear normalized NEq are shown. Comparing two vectors with current signals, the 64-input neuron equivalent has a total power consumption of approximately 2–3 mW at a low supply voltage, contains less than 1000 CMOS transistors, and provides good temporal characteristics.

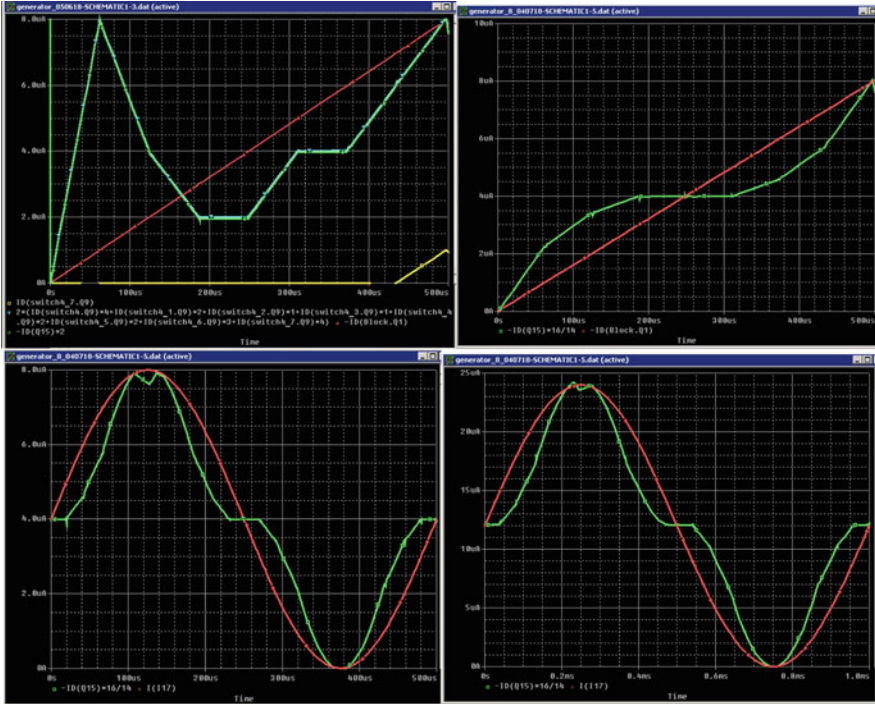


Fig. 4.12 Simulation result for eight sub-node circuit (Fig. 4.10): up left, for linear rising input signal (red), output signal (green), and corresponds to N-shape transfer characteristic; up right, for linear rising input (red), output signal (green), and corresponds to the auto-equivalence transfer characteristic; down left and right, for sinusoidal input signal (red), output signal (green), and corresponds to the auto-equivalence transfer characteristic for input current range 0–8 μ A and period 500 μ s (down left graph), 0–24 μ A, and 1 ms (down right graph)

The circuit performs summation, limited subtraction, and multiplication of analog currents on current mirrors.

4.3 Continuous-Logic (CL) Transformation and the Equivalently CL ADC

4.3.1 Basic Theoretical Foundations, Equivalence Models, and Their Modification for SMC_CL_ADC

These converters significantly reduce (or even eliminate) the error of digitization (quantization) inherent in the classical ADC. The CL transformations are given in [30, 45, 51], in which the transformation CL functions (CLF) are defined, and it is

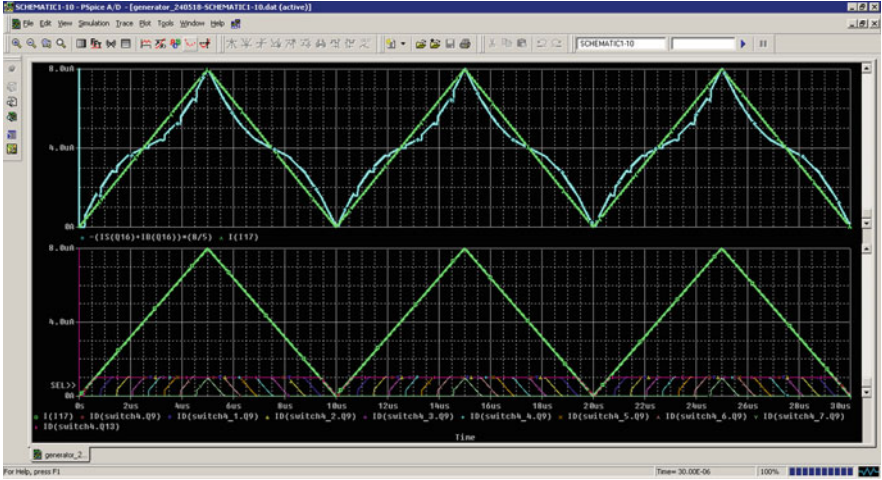


Fig. 4.13 Simulation result for circuit with step signals and eight-level approximation of input current signal: input signal (green line), output signal (blue line), and other signals (color lines)

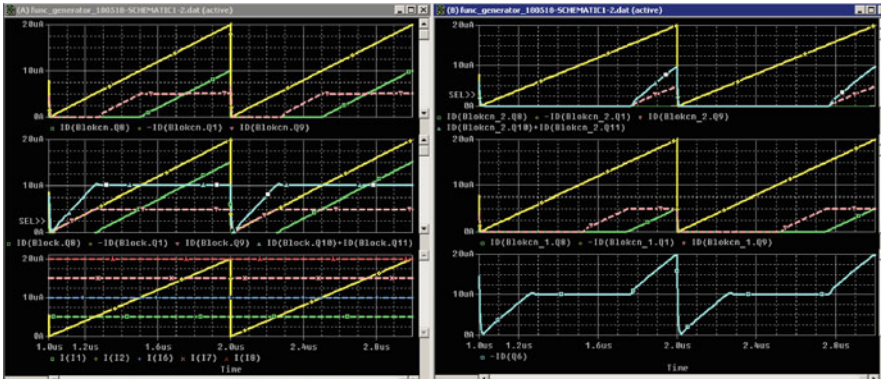


Fig. 4.14 Simulation result for four-level approximation, the realized nonlinear transformation is the normalized auto-equivalence function for self-learning convolutional networks ($I_{max} = 20 \mu A$, $T = 1 \mu s$)

shown that the operation of min and max of continuous logic are the basic operations of the functions. Using operators of hybrid logic for the formation of CLF, it is possible: $D_1[P(x_1, x_2)] = \max(x_1, x_2)$, $D_2[P(x_1, x_2)] = \min(x_1, x_2)$ where P and D are, respectively, threshold and non-threshold operators, which are implemented in various ways. In many models of neural networks for image recognition, especially many graded ones, it is desirable to have binary bit-plane images that encode the image matrix in Gray codes [41]. In addition, in a number of works [32–35, 40, 45, 46], it was shown that some operations of continuous logic, such as equivalence and nonequivalence, and their generalized family, provide a number of advantages in the

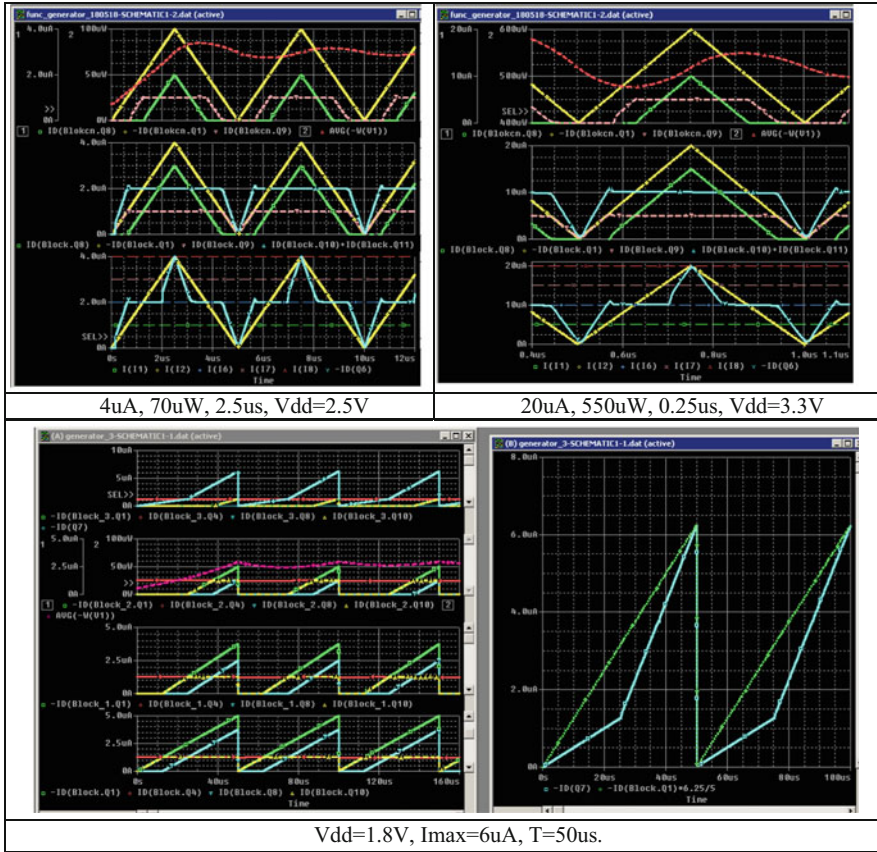


Fig. 4.15 Simulation result for four-level approximation, the realized nonlinear transformation is the normalized auto-equivalence function for self-learning convolutional networks (for different input currents and transformation periods): input signal (yellow line), output signal (blue line), and power consumption (red line)

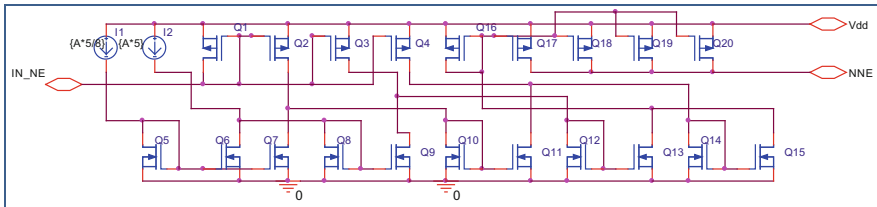


Fig. 4.16 Activation function circuit on current mirrors

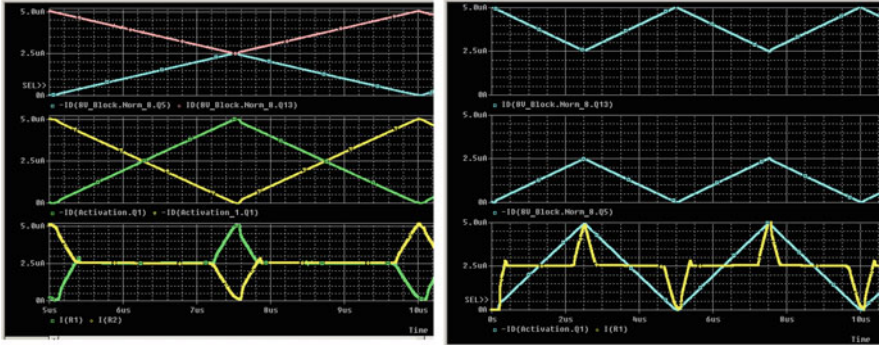


Fig. 4.17 The results of modeling the 64-input E_q for current $I_{max} = 5 \mu A$, and a linearly rising (falling) currents with a period $T = 2.5 \mu s$. On the left two upper signals (pink, maximum; blue, minimum of two input currents), green, equivalent signal; yellow, nonequivalence, below the signals after their nonlinear conversion; on the right, the two upper signals are the maximum and minimum, the lower blue is the normalized equivalence, the yellow is the nonlinear normalized equivalence

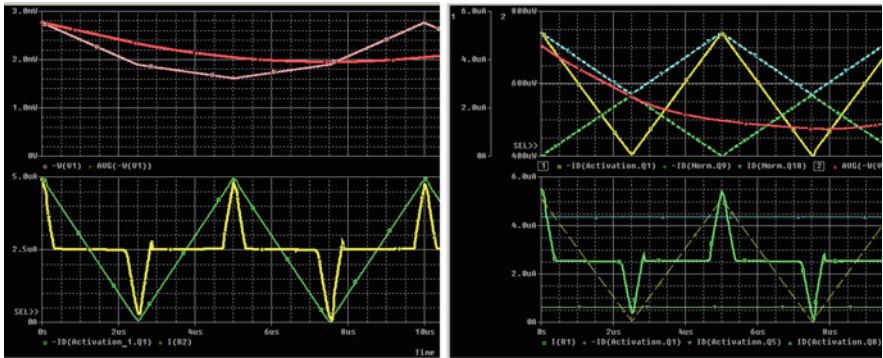


Fig. 4.18 The results of modeling the 64-input E_q for current $I_{max} = 5 \mu A$, and a linearly rising (falling) currents with a period $T = 2.5 \mu s$. On the left: the results of modeling the processes of formation of linear (green) and nonlinear normalized NEqs (yellow); on the upper graph: the peak and average consumption powers; on the right: the results of modeling the processes of formation of linear (yellow on the upper trace) and nonlinear normalized NEqs (green on the bottom trace), red line shows the power of consumption. Blue, maximum of two signals; green, minimum of two signals for $V = 3.3 V$

so-called equivalence paradigm. These scalar operations of equivalence $eq(x,y)$ and nonequivalence $neq(x,y)$ for $x, y \in [0,1]$ are defined in papers [32, 33], namely

$$eq(x, y) = x \wedge y + \bar{x} \wedge \bar{y} = \min(x, y) + \min((1 - x), (1 - y)) = 1 - |x - y| \tag{4.9}$$

$$neq(x, y) = |x - y| = 1 - eq(x, y) = \max(x, y) - \min(x, y) = \max(\bar{x}, \bar{y}) - \min(\bar{x}, \bar{y}) = (x \dot{-} y) + (y \dot{-} x) \tag{4.10}$$

where $(\dot{-})$ is the limited difference operation. If we consider it for $y = 1 - x = \bar{x}$, these functions are transformed to:

$$\text{eq}(x, \bar{x}) = 2(x \wedge \bar{x}) = 2 \min(x, \bar{x}) \quad (4.11)$$

$$\text{neq}(x, \bar{x}) = \max(x, \bar{x}) - \min(x, \bar{x}) = 1 - 2 \min(x, \bar{x}) \quad (4.12)$$

As it has been shown in work [45], these functions can be successfully used in the CL ADC. For the formation of binary bit planes that correspond to the categories of images coded in the Gray code, we used for each pixel an iterative procedure over the matrices of equivalence and nonequivalence obtained in the previous stages: $\text{eq}_{i+1}(\text{eq}_i(\dots), \text{neq}_i(\dots))$ and $\text{neq}_{i+1}(\text{eq}_i(\dots), \text{neq}_i(\dots))$. It is easy to see that the division of the segment $[0, 1]$ into $2^n = N$ subranges sets each of them a set, a vector of signs, which corresponds to the Gray code measured by the scalar size x . Thus, positional digit d_{n-i} of the code is defined as

$$d_{n-i}(\text{eq}_{i-1}, \text{neq}_{i-1}) = \{1, \text{ if } \text{eq}_{i-1} > \text{neq}_{i-1}, 0, \text{ if else}\} \quad (4.13)$$

where $i \in 1 \dots n$, and $\text{eq}_0 = x$, $\text{neq}_0 = \bar{x}$. From this, it is obvious that in order to realize the ADC for optical signals, we needed to synthesize BC CLs that implement the required operations eq_i , neq_i and the threshold operators. We called such ADCs equivalent to continuously logical, complementary dual ones, since the signals x and \bar{x} in them are complementary, and the CL functions are equivalent (nonequivalent), that is, equivalently CL ADC [45]. Since these ADCs were implemented on current mirrors (CM), and the input signals of the ADC are currents, we will designate such an ADC as an ADC CM [30]. In this work, as transformation CLFs, we use the following functions:

$$\begin{aligned} \text{eq}_{i+1}(\text{eq}_i, D/2) &= 2(\text{eq}_i \dot{-} 2(\text{eq}_i \dot{-} D/2)) \quad \text{or} \\ \text{neq}_{i+1}(\text{neq}_i, D/2) &= 2|\text{neq}_i - D/2| \end{aligned} \quad (4.14)$$

where $\text{eq}_0 = x$, $\text{neq}_0 = \bar{x}$, which allow us to work not with two signals, but with one signal, thereby simplifying the implementation of the cells. Structure of SMC CL ADC for IP is shown in Fig. 4.19.

4.3.2 Design of CL ADC CM-6 (8) (G): iv (the Iteration Variant) Based on DC-(G) (with Gray Code)

Figure 4.20 shows a circuit of one channel of SMC_ADC. The structure is shown in Fig. 4.20a and the base cell in Fig. 4.20b. The circuit consists of a sample and hold device (SHD), a single digital-analog DC-(G) cell (block A), and additional elements (block B). The input analog current signal to be converted is recorded in

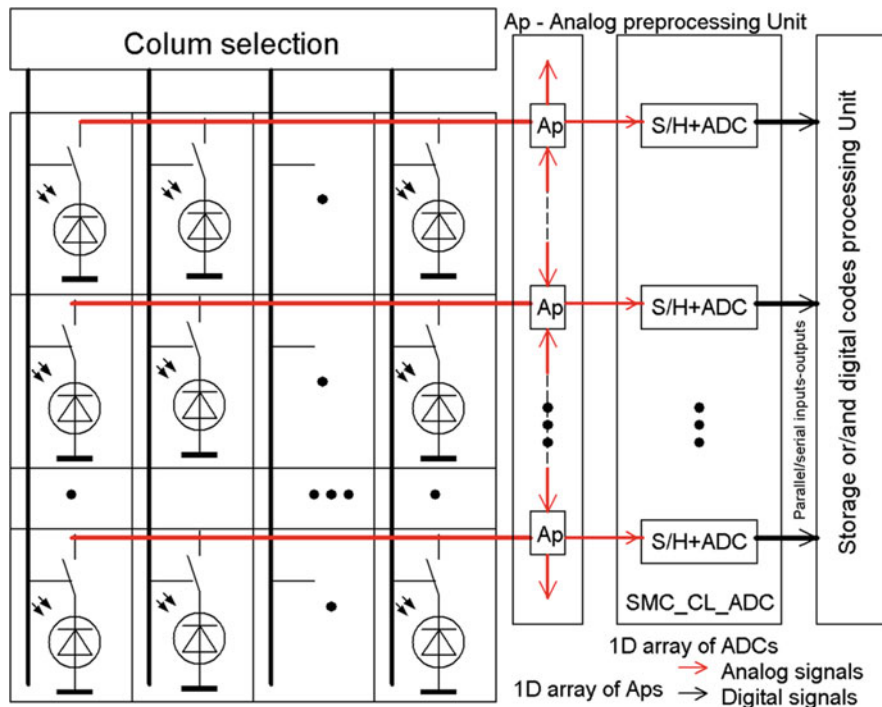
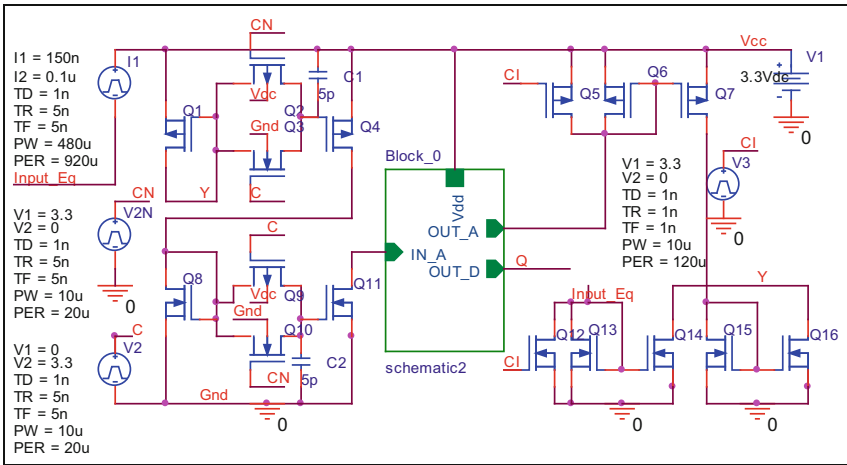


Fig. 4.19 Structure of 2D image sensor with 1D array of CL_ADC and storage or/and digital code processing unit

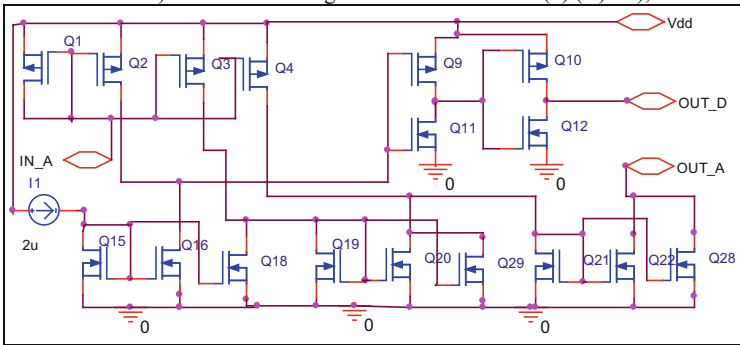
the SHD and then transmitted to the analog DC-(G), which will generate the next digital bit of the output code and the CL function.

This function is fed back to the SHD to form the next consecutive bits. The device selection and hold (SHD) consists of 18 transistors. DC-(G) consists of 15 or 17 transistors and a reference current generator. Since the circuit of one channel consists of only 33 (35) transistors, this makes it promising for multisensory systems. The DC converts the input analog signal to another output current signal, using CLF (Sect. 4.3.1) overcurrent signals and simultaneously compares it with the threshold current. The advantage of such continuous logical transformations is that the form of such transformations can be very diverse, and the operations of continuous logic used for such transformations themselves are also numerous.

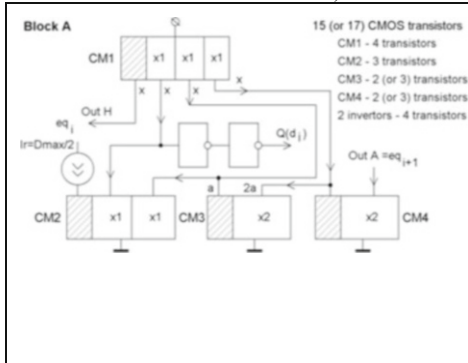
Thus, there is a wide choice for searching and optimizing such cells taking into account the required goal. To minimize the apparatus costs, cells can be very simple and consist of 10–20 transistors. In addition, the use of other known, improved dynamic and accurate indicators of current comparators [50–52], including a floating gate, etc., significantly expands the range of application of such implementations of ADC, reduces power consumption to microwatts, or significantly expands the dynamic range of input signals and maximum conversion frequencies.



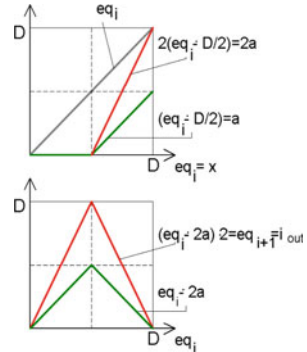
a) Functional diagram of CL ADC CM-6(8) (G) -iv;



b) electrical circuit of block A;



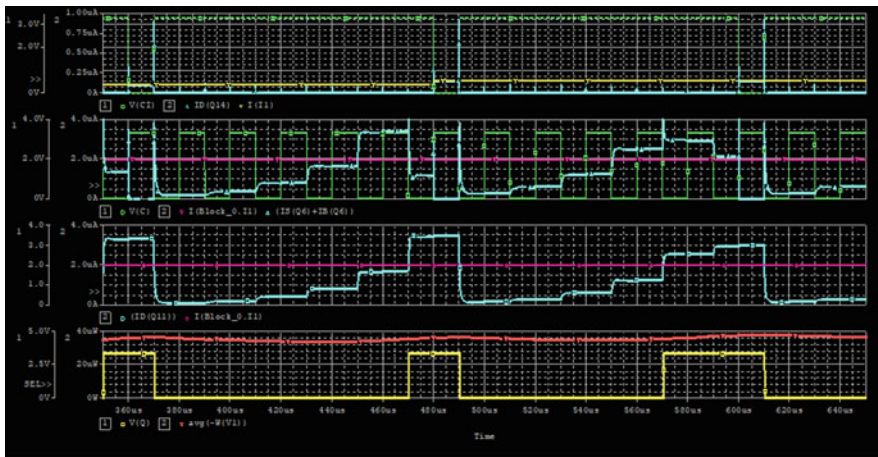
c) functional diagram of block A;



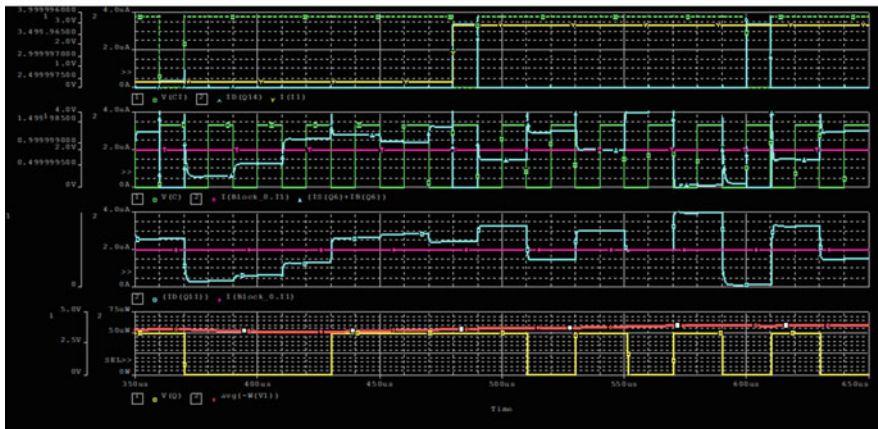
d) diagrams of continuous logical transformation of analog signal by limited difference and multiplication by 2 for realization of the equivalence function over the signal

Fig. 4.20 Circuits of the one channel of multichannel CL ADC CM-6 (8) (G)-iv with iteration transformation and base cell DC-(G). (a) Functional diagram of CL ADC CM-6 (8) (G)-iv. (b) Electrical circuit of block A. (c) Functional diagram of block A. (d) Diagrams of continuous logical transformation of analog signal by limited difference and multiplication by 2 for realization of the equivalence function over the signal

The advantage of the structure with a serial output of the Gray code is that increasing the number of iterations increases the bit ADC with an unchanged structure. To convert a serial Gray code to a binary code, only one modulo adder and one D flip-flop are required. Figures 4.21 and 4.22 show the results of simulation of one channel of six bits CL ADC CM-6 (G)-iv with iteration transformation at

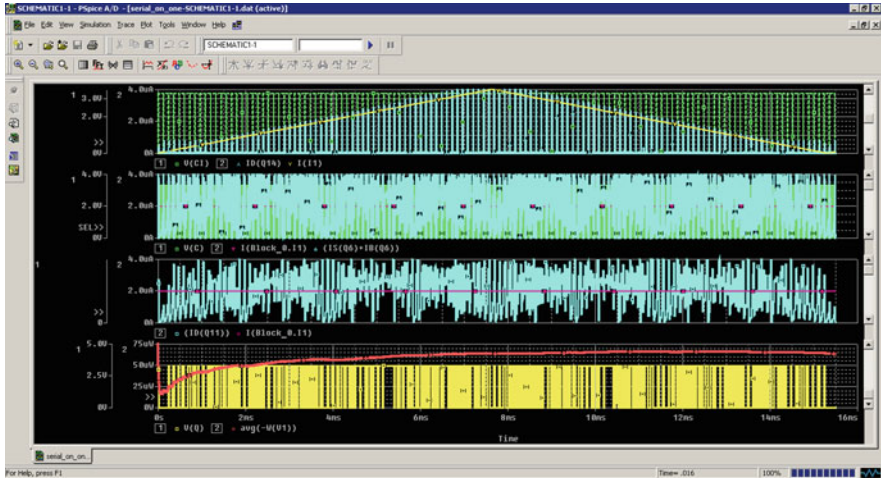


a)

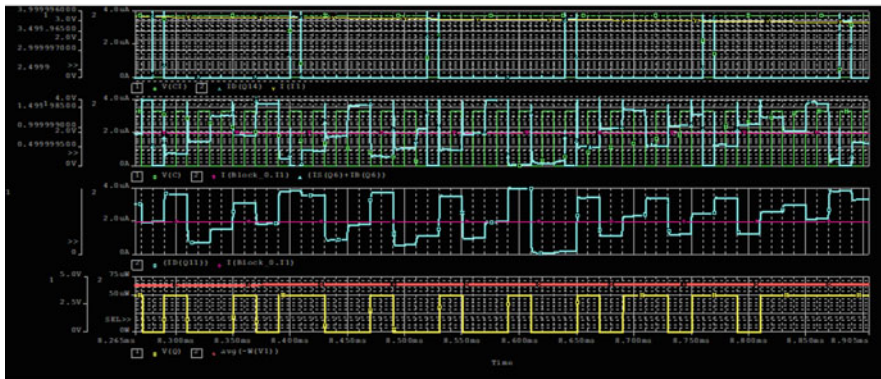


b)

Fig. 4.21 (a) Simulation result for two input currents 100 nA and 150 nA, corresponding output Gray codes {000001} and {000011}; the blue line in the third trace is output current of the block for six-digit ADC, the violet line is the threshold current; the yellow line in the fourth trace is the output voltage of the block that corresponds to output code (a time interval for the first code 370–490 μs (six digits by 20 μs), a time interval for the second code 490–610 μs); the red line is the power consumption of about 40 μW. (b) Simulation result for two input currents and corresponding output Gray codes {000111} and {010101}; the consumption power is about 40 μW



a)



b)

Fig. 4.22 Simulation results of the six-digit ADC for a triangular current signal (the yellow line in the first trace): **(a)** the whole time interval; **(b)** for five time intervals with decreasing input current and corresponding output Gray code {100101}, {100100}, {100100}, {101101}, {101111} (the yellow line in the fourth trace), the red line is the power consumption of about 70 μ W

linearly increasing input current signals. The total power consumption of this ADC-6 (8) (G)-iv did not exceed 70 μ W with a maximum input current of 4 μ A and a conversion period of 120 μ S ($6 \times 20 \mu$ S for 6 bits) and a conversion period of 160 μ S ($8 \times 20 \mu$ S for 8 bits). For operating modes with lower currents and $V_{dd} = 1.5\text{--}1.8$ V, the power consumption of ADC-6 (8) (G)-iv can be reduced to 10–15 μ W.

The drawback of our earlier works is the lack of research on the ultimate capabilities of such structures and their precision characteristics. Therefore, in this paper, we pre-observed such a structure in the formation of eight-digit code,

determined the possibility of operation with very small input currents (10 nA to 1 μ A), and adding to the structure of the DAC and converters from the Gray code to the binary code (Fig. 4.23), and determined the magnitude of ADC errors and its accuracy characteristics for different modes. By reducing the requirements for high speed, the proposed diagram allows using analog-to-digital conversions for small-amplitude input currents, and the power consumption of the such single ADC channel can be less than 50 μ W with $D_{max} = 1 \mu$ A. All the circuits are modeled on 1.5 μ m CMOS transistors. Simulation of analog-to-digital conversion errors is shown in Fig. 4.24. Simulation is performed for $D_{max} = 4 \mu$ A, 6-bit ADC, conversion period $T = 120 \mu$ s. Figure 4.24 shows that the maximum error is about 1 least significant bit (LSB), and only for the maximum input current, the error is about 2 LSB for the 8-bit ADC. Also, the simulation results showed that when reducing the conversion time to 10–20 μ s, the errors will be the same.

In Fig. 4.23, functional diagram of CL ADC CM-(8) (G)-iv with Gray-to-binary code transformation and serial/parallel outputs with code converter and DAC for error calculation is shown. Actually the ADC itself, from which 1D or 2D arrays will be done for sensors or image processors, in contrast to the circuit in Fig. 4.20, may additionally comprise some digital elements, for example, a logic element and a trigger or register. This depends on the possible modes and requirements regarding

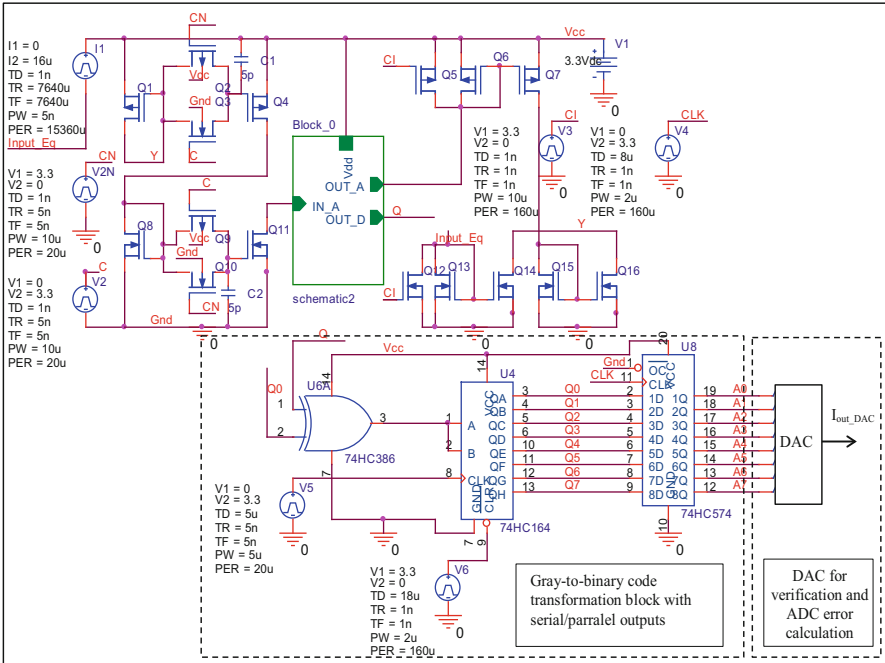


Fig. 4.23 Functional diagram of CL ADC CM-(8) (G)-iv with Gray-to-binary code transformation and serial/parallel outputs with code converter and DAC for errors calculation

the formats of output and storage of code arrays. Therefore, in Fig. 4.23, these additional optional units are marked with a dash-dotted line. To test the accuracy and timing characteristics in the dynamics, we used two registers and DAC. The results of modeling this circuit with OrCAD are partially shown in Fig. 4.24, and they confirm the correct operation and analog-to-digital and code conversion, both when linearly increasing (decreasing) and sinusoidal current signals are applied to the ADC input. They show that for the 8-bit ADC, even in high speed ($I_{max} = 16\text{--}24 \mu\text{A}$) and low-voltage low-frequency energy-efficient modes (with $I_{max} = 1 \mu\text{A}$, $4 \mu\text{A}$), the maximum error does not exceed 4–5 quantization quanta, and the average error does not exceed 2 LSB.

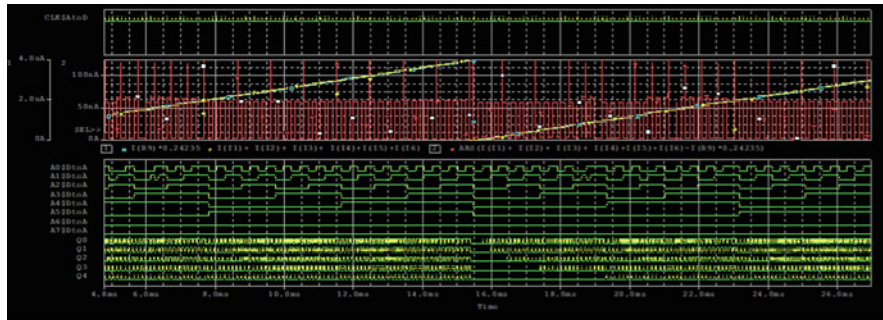
4.3.3 *Simulating Parallel Conveyor CL_ADC (P_C) Based on Eight 8-DC-(G) with Parallel Serial Output*

The block diagram of parallel conveyor CL_ADC (P_C) based on 8-DC-(G) (with Gray code) with a parallel serial output is shown in Fig. 4.25. The simulation results with PSpice OrCAD are shown in Fig. 4.26. Researches have shown that in such CL_ADC (P_C) 6 (8)-DC-(G) at changing I_{max} from 16 to $24 \mu\text{A}$, the power consumption at 3.3 V was from 1 to $2 \mu\text{W}$ (6 bits) and $3 \mu\text{W}$ (8 bits). The conversion frequencies in the experiments were for these currents: 32, 40, and 50 MHz for $16 \mu\text{A}$ and 64 MHz for $24 \mu\text{A}$ and $40 \mu\text{A}$.

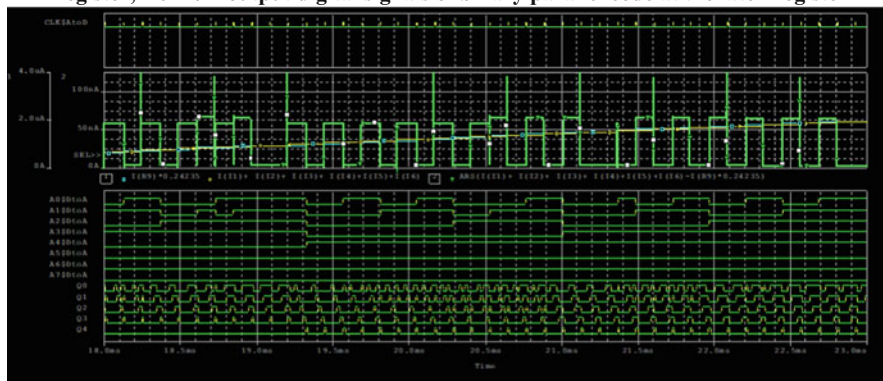
They correspond to different modes: different I_{max} , namely $1 \mu\text{A}$, $4 \mu\text{A}$, $16 \mu\text{A}$, $24 \mu\text{A}$, $40 \mu\text{A}$; various 1.5 V, 1.8 V, 2.5 V, 3.3 V; various transformation periods T ($0.02 \mu\text{s}$, $0.025 \mu\text{s}$, $1 \mu\text{s}$, $20 \mu\text{s}$, $100 \mu\text{s}$), etc. These researches show that power consumption for ADC for the specified values of I_{max} (equal $1 \mu\text{A}$ and 1.8 V, 64 nA and 1.5 V) makes accordingly $40 \mu\text{W}$ and $2 \mu\text{W}$, the quantization step is 15.625 nA for $I_{max} = 4 \mu\text{A}$ and 62.5 nA for $I_{max} = 16 \mu\text{A}$, and quantization frequency = 40 MHz.

The essence of analog preprocessing is to find the function from the signals of several adjacent channels for different 1D and 2D windows. In this case, the 1D window is a size of 3 (may be 5, 7, 9, etc.), and the processing type is the function of finding the average of the three signals. As a function, any continuous logic functions of the type max, min, described in Sect. 4.3.1 and in paper [41], can be used.

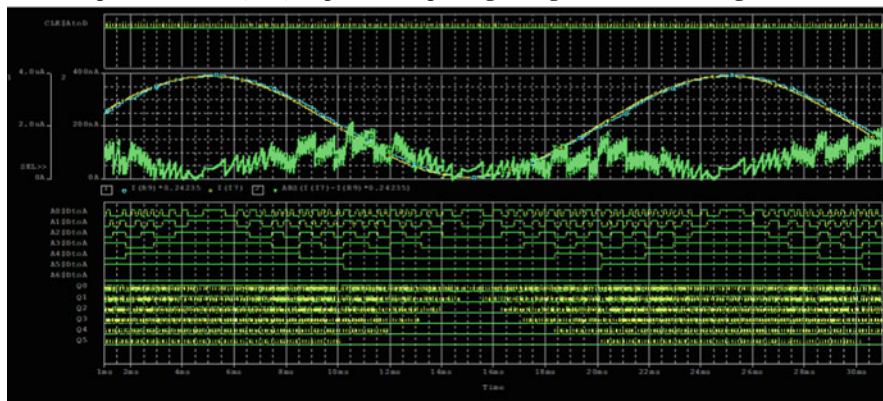
The analog preprocessing unit (Fig. 4.19) consists of 4 (6) CMOS transistors (Fig. 4.27) in this case. For functions min, max, etc., the Ap-unit consists of about 10–20 CMOS transistors. The consumption power of one-channel 8-bit ADC + Ap-unit is less than $250 \mu\text{W}$. Simulation results of analog signal preprocessing (selecting average signal out of three neighbor channel signals) for different input signals (linearly increasing (decreasing) and sinusoidal signals) are shown in Fig. 4.28.



a) The blue line is the DAC output current, yellow line is the 8 bit ADC input current, the red line is the ADC current error (<70nA); Q0..Q7 – output digital signals of the shift register, A0-A7 – output digital signals of binary parallel code at the latch register



b) the blue line is the DAC output current, the yellow line is the ADC input current, the green line is the ADC current error (<70nA), A0-A7 – output digital signals of binary parallel code, Q0..Q4 – part of output digital signals of the shift register



c) the blue line is the DAC output current, the yellow line is the ADC input current, the green line is the ADC current error (<200nA), A0-A6 – part outputs digital signals of 8 binary parallel code, Q0..Q5 – part of outputs 8 digital signals of the shift register

Fig. 4.24 Simulation results of the 8-bit ADC with Gray-to-binary code transformation and serial/parallel outputs. (a) The blue line is the DAC output current, yellow line is the 8-bit ADC

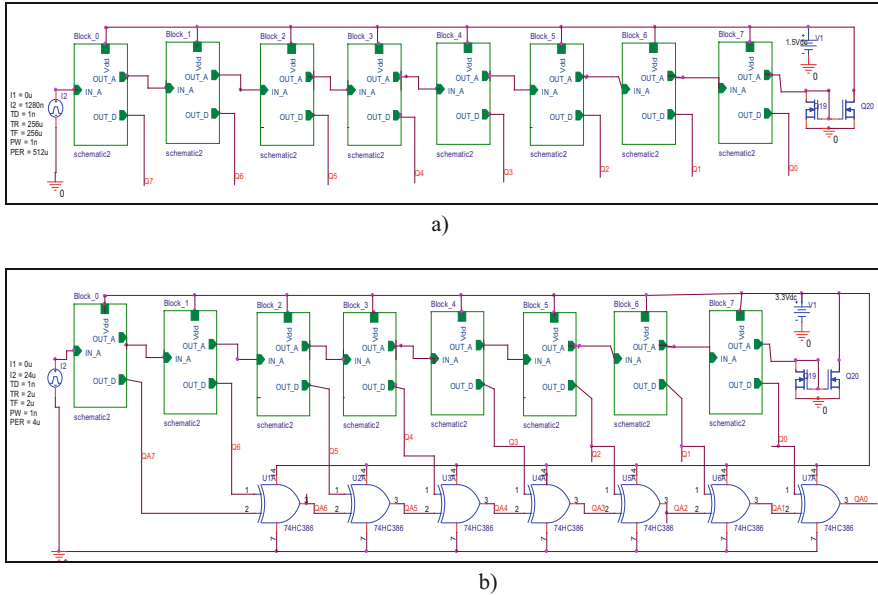
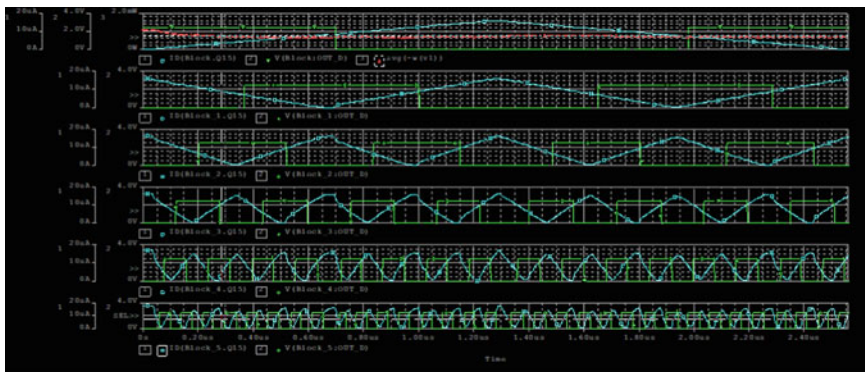


Fig. 4.25 Structure of 8-bit ADC: (a) with Gray code parallel outputs (Q0–Q7); (b) with Gray-to-binary code transformation and parallel outputs (QA0–QA7)

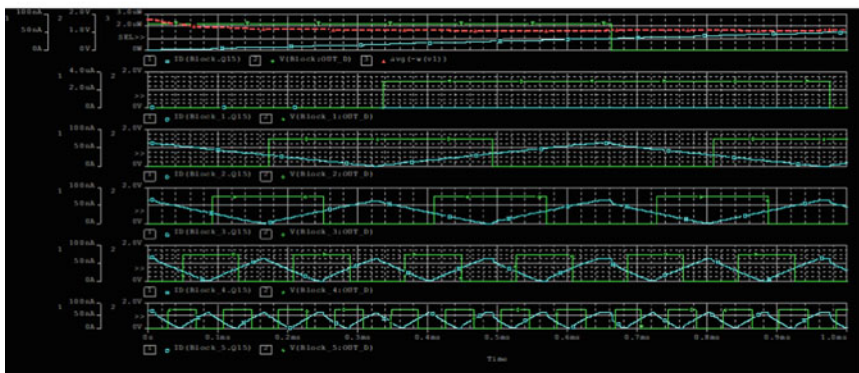
4.4 Conclusions

For the construction of mixed image processor (IP), neural networks (NNs), and image intensity transformation, the fundamentals of continuous logic cell (CLC) design based on current mirrors (CM) with functions of preliminary analog processing are proposed. Several effective schemes have been developed and modeled for CLC and optoelectronic complement dual analog neuron-equivalentors as hardware accelerators SLECNS. The proposed CLC have a modular hierarchical construction principle and are easily scaled. Their main characteristics were measured. They have a low supply voltage of 1.8–3.3 V, small power consumption of no more than 1 mW, processing time-conversion 0.1–1 μ s, insignificant relative calculation errors (1–5%), can work in low-power modes (less than 100 μ W)

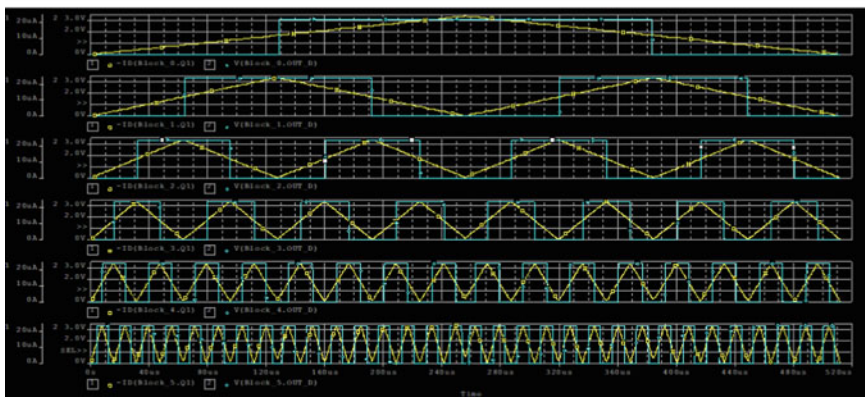
Fig. 4.24 (continued) input current, the red line is the ADC current error (<70 nA); Q0–Q7, output digital signals of the shift register; A0–A7, output digital signals of binary parallel code at the latch register. (b) The blue line is the DAC output current, the yellow line is the ADC input current, the green line is the ADC current error (<70 nA); A0–A7, output digital signals of binary parallel code; Q0–Q4, part of output digital signals of the shift register. (c) The blue line is the DAC output current, the yellow line is the ADC input current, the green line is the ADC current error (<200 nA); A0–A6, part of output digital signals of eight binary parallel code; Q0–Q5, part of output eight digital signals of the shift register



a) Time diagrams of signals of digit converting cells of 6 bit CL_ADC for mode: converting frequency $F = 50\text{MHz}$, input current $I_{\text{max}} = 16\mu\text{A}$, $V_{\text{dd}} = 3.3\text{V}$, consumption power $P \approx 1\text{mW}$

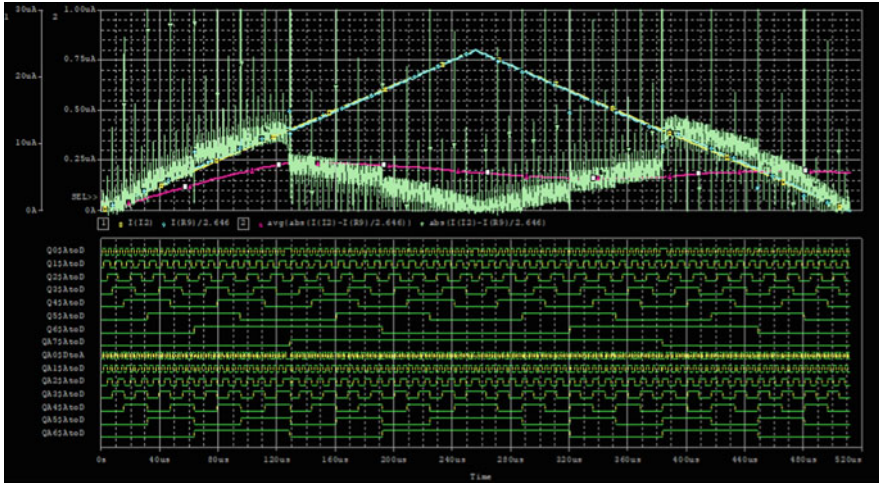


b) Time diagrams of signals of digit converting cells of 6 bit CL_ADC for mode: converting frequency $F = 50\text{kHz}$, input current $I_{\text{max}} = 64\text{nA}$, $V_{\text{dd}} = 1.5\text{V}$, consumption power $P \approx 2\mu\text{W}$

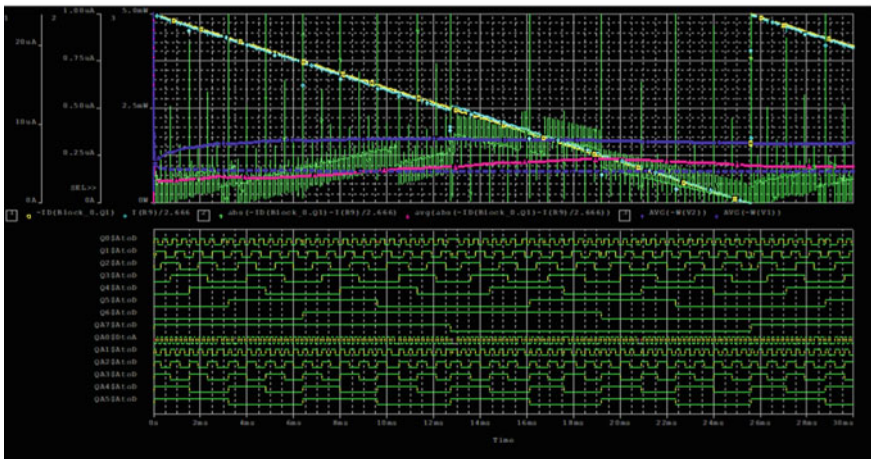


c) Time diagrams of signals of digit converting cells of 8 bit ADC (6 cells out of 8 are shown) for mode: converting frequency $F = 1\text{MHz}$, input current $I_{\text{max}} = 24\mu\text{A}$, $V_{\text{dd}} = 3.3\text{V}$

Fig. 4.26 Structure of multichannel 8-bit ADC (1D array 8-bit CL_ADC) and simulations results. (a) Time diagrams of signals of digit converting cells of 6-bit CL_ADC for mode: converting frequency $F = 50\text{MHz}$, input current $I_{\text{max}} = 16\mu\text{A}$, $V_{\text{dd}} = 3.3\text{V}$, power consumption $P \approx 1\text{mW}$.

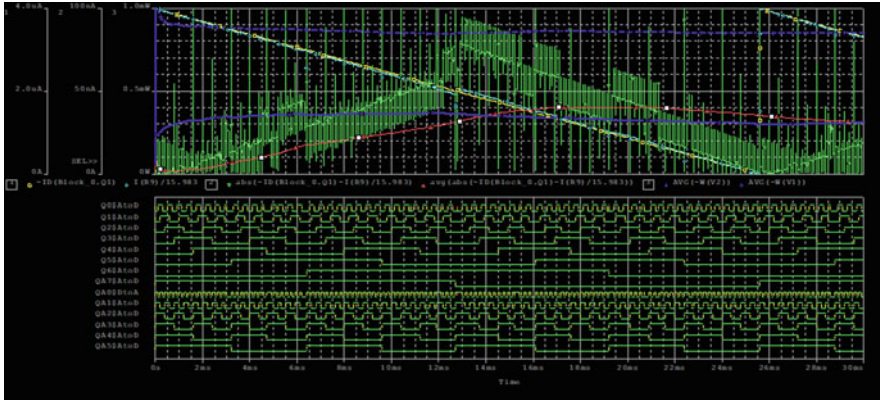


d) Time diagrams of 8 bit parallel CL_ADC signals by simulation for $I_{input_max}=24\mu A$, ADC conversion time is $1\mu s$; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<250nA$), the green line is the ADC current error; QA0-QA7 – output digital signals of binary parallel code, Q0..Q7 (Q7=QA7) – output digital signals of Gray parallel code



e) Time diagrams of 8 bit parallel CL_ADC signals by simulation for $I_{input_max}=24\mu A$; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<250nA$), the green line is the ADC current error, the blue line is the power consumption (3mW)

Fig. 4.26 (continued) (b) Time diagrams of signals of digit converting cells of 6-bit CL_ADC for mode: converting frequency $F = 50\text{ kHz}$, input current $I_{max} = 64\text{ nA}$, $V_{dd} = 1.5\text{ V}$, power consumption $P \approx 2\text{ }\mu W$.



f) Time diagrams of 8 bit parallel CL_ADC signals by simulation for $I_{\text{input_max}}=4\mu\text{A}$, conversion frequency is 10kHz; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<40\text{nA}$), the green line is the ADC current error, the blue line is the power consumption (1.3mW)

Fig. 4.26 (continued) (c) Time diagrams of signals of digit converting cells of 8-bit ADC (6 cells out of 8 are shown) for mode: converting frequency $F = 1 \text{ MHz}$, input current $I_{\text{max}} = 24 \mu\text{A}$, $V_{\text{dd}} = 3.3 \text{ V}$. (d) Time diagrams of 8-bit parallel CL_ADC signals by simulation for $I_{\text{input_max}} = 24 \mu\text{A}$, ADC conversion time is $1 \mu\text{s}$; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<250 \text{ nA}$), the green line is the ADC current error; QA0–QA7, output digital signals of binary parallel code, Q0–Q7 (Q7 = QA7), output digital signals of Gray parallel code. (e) Time diagrams of 8-bit parallel CL_ADC signals by simulation for $I_{\text{input_max}} = 24 \mu\text{A}$; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<250 \text{ nA}$), the green line is the ADC current error, the blue line is the power consumption (3 mW). (f) Time diagrams of 8-bit parallel CL_ADC signals by simulation for $I_{\text{input_max}} = 4 \mu\text{A}$, conversion frequency is 10 kHz; the blue line is the DAC output current, the yellow line is the ADC input current, the violet line is the average ADC current error ($<40 \text{ nA}$), the green line is the ADC current error, the blue line is the power consumption (1.3 mW)

and high-speed (1–2 MHz) modes. The relative energy efficiency of the CLC and equivalentors is estimated at a value of not less than 10^{12} an. op./sec. per watt and can be increased by an order. The correctness of the chosen concept is confirmed by the obtained results of the design and creation of neuron equivalentors (NEqs) and MIMO structures based on them. Such neuron equivalentors can form the basis of promising self-learning biologically inspired devices SLECNS and CNN, in which the number of such parallel-running NEqs is 1000. Thus, we have proposed implementation options for digital-analog cells (DC) and CL structures of the ADC CM. Such ADCs are simple, and only one DC is required for the iteration type, supplemented by a sample and hold device. The advantage of the ADC is the ability to easily implement parallel code, as well as serial parallel output code. Results of circuit simulation using OrCAD are shown. Such simple structure of CL ADC CM with low power consumption $\leq 3 \text{ mW}$ and supply voltage 1.8–3.3 V, and at the same time with good dynamic characteristics (frequency of digitization even for $1.5 \mu\text{m}$

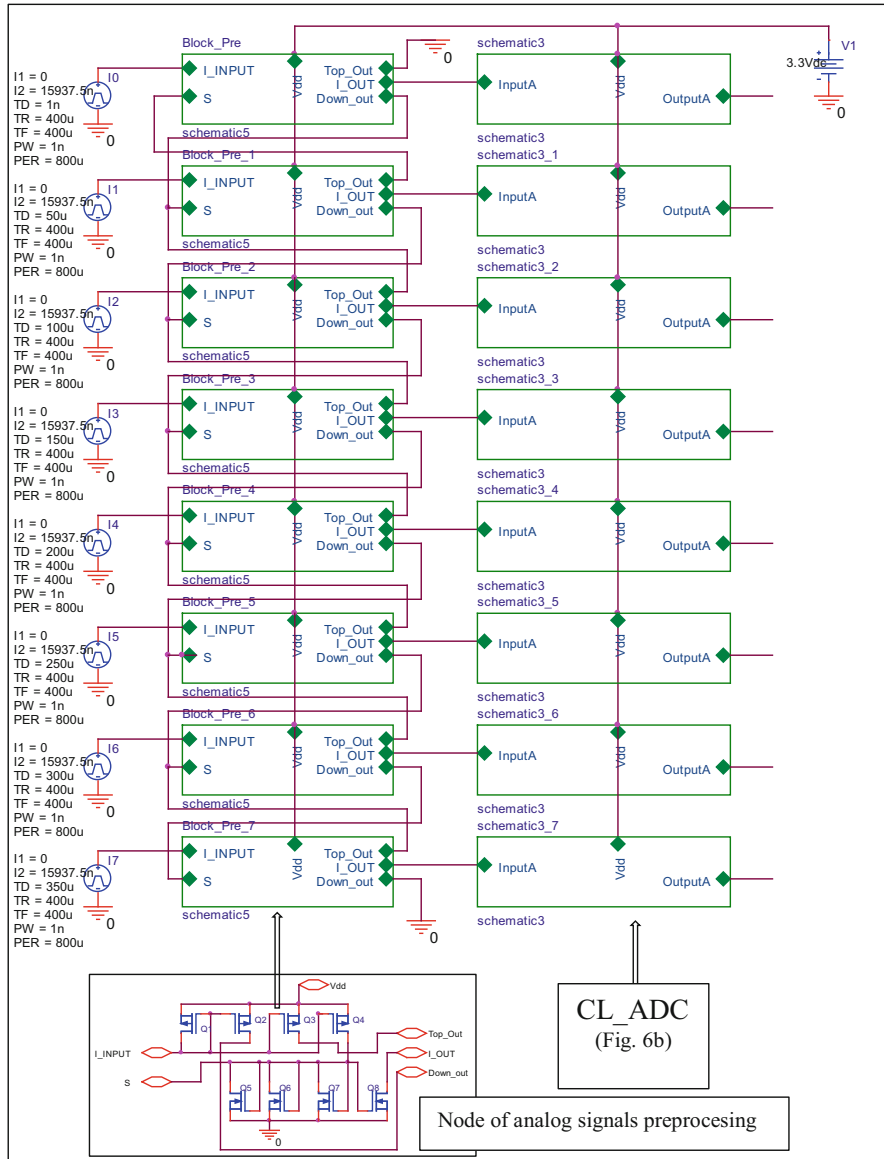
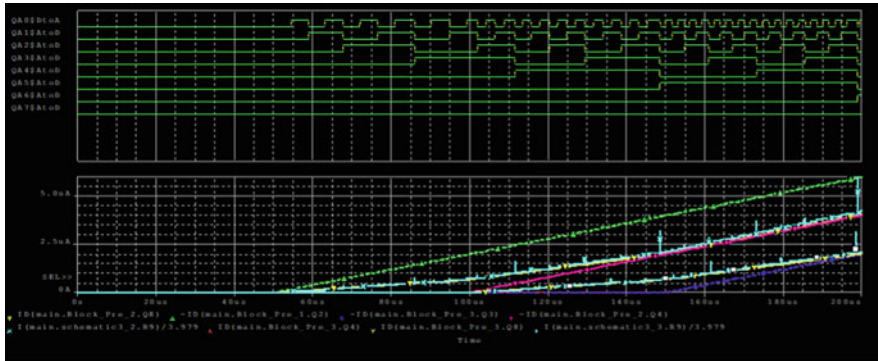
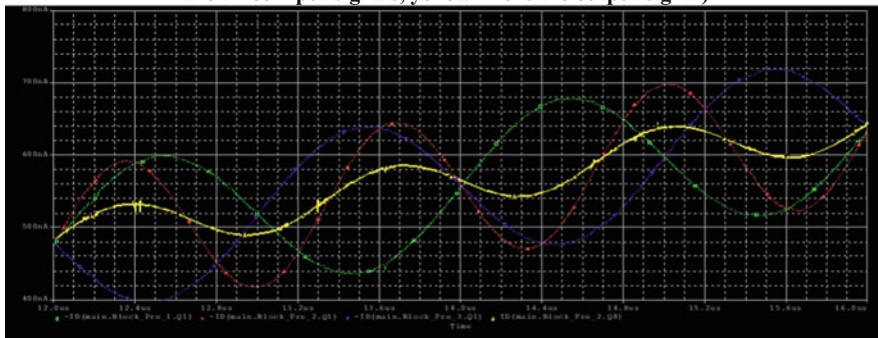


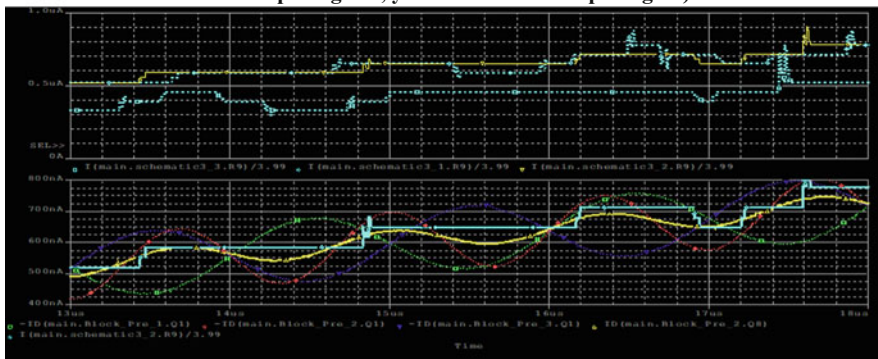
Fig. 4.27 Structure of multichannel 8-bit ADC (1D array 8-bit CL_ADC) with analog signal preprocessing



a) simulation results of analog signals preprocessing (selecting average signal out of three neighbor channels signals): green, blue, violet lines are three input signals, yellow line is the output signal)



b) simulation results of analog signals preprocessing (selecting average signal out of three neighbor channels signals): green, blue, red lines are three input signals, yellow line is the output signal)



c) simulation results of analog signals preprocessing (selecting average signal out of three neighbor channels signals): green, blue, violet lines are three input signals, yellow line is the output signal), light blue is the DAC output signal

Fig. 4.28 Structure of multichannel 8-bit ADC (1D array 8-bit CL_ADC) with analog signal preprocessing. (a) Simulation results of analog signal preprocessing (selecting average signal

CMOS-technologies is 40 MHz, and can be increased up to ten times) and accuracy (Δ quantization = 15.6–62.5 nA for $I_{max} = 4\text{--}16 \mu\text{A}$) characteristics are shown. Taking into account the sensitivity of modern photodetectors, the range of optical signals can be 1–200 μW . For the ADC of iteration type, one channel consists of one DC-(G) and SHD, and it has only 35–40 CMOS transistors. Thus, such 1D and 2D arrays of successive ADCs are very promising for sensors and IP. The general power consumption of one ADC, in this case, is only 50–70 μW , if the maximum input current is 4 μA . For high performance and frequency of conversions, it is preferable to use the parallel pipeline CL_ADC (P_C) scheme based on the set of 8-DC-(G) with parallel serial outputs. The maximal error is about 1 LSB, and only about 2 LSB for 8-bit CL ADC for the maximal input current. CL ADC CM with analog signal preprocessing opens new prospects for realization linear and matrix (with picture operands) micro photo-electronic structures which are necessary for neural networks, digital optoelectronic processors, neuro fuzzy controllers.

References

1. Krasilenko, V. G., Nikolskyy, A. I., & Bozniak, Y. A. (2002). Recognition algorithms of multilevel images of multi-character identification objects based on nonlinear equivalent metrics and analysis of experimental data. *Proceedings of SPIE*, 4731, 154–163.
2. Krasilenko, V. G., Nikolskyy, A. I., & Bozniak, Y. A. (2012). Recognition algorithms of images of multi-character identification objects based in nonlinear equivalent metrics and analysis of experimental data using designed software. In *Proceedings of Eleventh All-Ukrainian International Conference* (pp. 107–110).
3. Krasilenko, V. G., Boznyak, Y. A., & Berozov, G. N. (2009). Modelling and comparative analysis of correlation and mutual alignment equivalent images. Science and learning process: Scientific and methodical. In *Proceedings Scientific Conference of the VSEI Entrepreneurship University "Ukraine"* (pp. 68–70).
4. Krasilenko, V. G., & Magas, A. T. (1997). Multiport optical associative memory based on matrix-matrix equivalentors. *Proceedings of SPIE*, 3055, 137–146.
5. Krasilenko, V. G. (2010). Research and design of equivalence model of heteroassociative memory. *The Scientific Session of MIFI-2010*, 2, 83–90.
6. Krasilenko, V. G., Saletsky, F. M., Yatskovsky, V. I., & Konate, K. (1998). Continuous logic equivalence models of Hamming neural network architectures with adaptive-correlated weighting. *Proceedings of SPIE*, 3402, 398–408.
7. Krasilenko, V. G., Nikolskyy, A. I., Yatskovskaya, R. A., & Yatskovsky, V. I. (2011). The concept models and implementations of multiport neural net associative memory for 2D patterns. *Proceedings of SPIE*, 8055, 80550T.

←
Fig. 4.28 (continued) output of three neighbor channel signals): green, blue, violet lines are three input signals, yellow line is the output signal. (b) Simulation results of analog signals preprocessing (selecting average signal output of three neighbor channels signals): green, blue, red lines are three input signals, yellow line is the output signal. (c) Simulation results of analog signal preprocessing (selecting average signal output of three neighbor channel signals): green, blue, violet lines are three input signals, yellow line is the output signal, light blue is the DAC output signal

8. Krasilenko, V. G., Lazarev, A., & Grabovlyak, S. (2012). Design and simulation of a multiport neural network heteroassociative memory for optical pattern recognitions. *Proceedings of SPIE*, 8398, 83980N-1.
9. Krasilenko, V. G., & Nikolskyy, A. I. (2001). Optical pattern recognition algorithms based on neural-logic equivalent models and demonstration of their prospects and possible implementations. *Proceedings of SPIE*, 4387, 247–260.
10. Krasilenko, V. G., Kolesnitsky, O. K., & Boguhvalsky, A. K. (1997). Application of non-linear correlation functions and equivalence models in advanced neuronets. *Proceedings of SPIE*, 3317, 211–223.
11. Krasilenko, V. G., & Nikitovich, D. V. (2014). Experimental studies of spatially invariant equivalence models of associative and hetero-associative memory 2D images. *Systemy obrobky informaciji Kharkivskijj universytet Povitrjanykh Syl imeni Ivana Kozheduba*, 4(120), 113–120.
12. Krasilenko, V. G., Lazarev, A. A., Grabovlyak, S. K., & Nikitovich, D. V. (2013). Using a multi-port architecture of neural-net associative memory based on the equivalency paradigm for parallel cluster image analysis and self-learning. *Proceedings of SPIE*, 8662, 86620S.
13. Krasilenko, V. G., Nikolskyy, A. I., & Flavitskaya, J. A. (2010). The structures of optical neural nets based on new matrix_tensor equivalently models (MTEMs) and results of modeling. *Optical Memory and Neural Networks (Information Optics)*, 19(1), 31–38.
14. Krasilenko, V. G., Lazarev, A. A., & Nikitovich, D. V. (2014). Experimental research of methods for clustering and selecting image fragments using spatial invariant equivalent models. *Proceedings of SPIE*, 9286, 928650.
15. Krasilenko, V. G., & Nikitovich, D. V. (2015). Researching of clustering methods for selecting and grouping similar patches using two-dimensional nonlinear space-invariant models and functions of normalized equivalence. In *VII Ukrainian-Polish Scientific and Practical Conference Electronics and Information Technologies (ELIT-2015)* (pp. 129–134). Lviv: Ivan Franko National University of Lviv.
16. Krasilenko, V. G., & Nikitovich, D. V. (2014). Modeling combined with self-learning clustering method of image fragments in accordance with their structural and topological features. *Visnyk Khmelnyncjkogho Nacionaljnogho Universytetu*, 2, 165–170.
17. Krasilenko, V. G., & Nikitovich, D. V. (2014). Sumishhenyj z samonavchannjam metod klasteryzaciji fragmentiv zobrazhenj za jikh strukturno-topologichnymy oznakamy ta jogho modeljuvannja. In *Pytannja prykladnoji matematyky i matematychnogho modeljuvannja* (pp. 167–176).
18. LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time-series. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks*. Cambridge, MA: MIT Press.
19. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>.
20. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)* (pp. 1097–1105). New York: Curran Associates Inc.
21. Shafiee, A., et al. (2016). ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* (pp. 14–26). Seoul: IEEE. <https://doi.org/10.1109/ISCA.2016.12>.
22. Zang, D., Chai, Z., Zhang, J., Zhang, D., & Cheng, J. (2015). Vehicle license plate recognition using visual attention model and deep learning. *Journal of Electronic Imaging*, 24(3), 033001. <https://doi.org/10.1117/1.JEI.24.3.033001>.
23. Taylor, G. W., Fergus, R., LeCun, Y., & Bregler, C. (2010). Convolutional learning of spatio-temporal features. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Proceedings of the 11th European Conference on Computer Vision: Part VI (ECCV'10)* (pp. 140–153). Berlin: Springer.
24. Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011, Providence, RI* (pp. 3361–3368). <https://doi.org/10.1109/CVPR.2011.5995496>.

25. Krasilenko, V. G., Lazarev, A. A., & Nikitovich, D. V. (2017). Modeling and possible implementation of self-learning equivalence-convolutional neural structures for auto-encoding-decoding and clusterization of images. *Proceedings of SPIE, 10453*, 104532N.
26. Krasilenko, V. G., Lazarev, A. A., & Nikitovich, D. V. (2018, 8 March). Modeling of biologically motivated self-learning equivalent-convolutional recurrent-multilayer neural structures (BLM_SL_EC_RMNS) for image fragments clustering and recognition. In *Proc. SPIE 10609, MIPPR 2017: Pattern Recognition and Computer Vision, 106091D*. <https://doi.org/10.1117/12.2285797>
27. Fey, D. (2001). Architecture and technologies for an optoelectronic VLSI. *Optic, 112*(7), 274–282.
28. Yi, L., Shan, G., Liu, S., & Xie, C. (2016). High-performance processor design based on 3D on-chip cache. *Microprocessors and Microsystems, 47*, 486–490. ISSN 0141-9331. <https://doi.org/10.1016/j.micpro.2016.07.009>.
29. Maier-Flaig, F., Rinck, J., Stephan, M., Bocksrocker, T., Bruns, M., Kübel, C., Powell, A. K., Ozin, G. A., & Lemmer, U. (2013). Multicolor silicon light-emitting diodes (SiLEDs). *Nano Letters, 13*(2), 475–480. <https://doi.org/10.1021/nl3038689>.
30. Krasilenko, V. G., Nikolskyy, A. I., & Lazarev, A. A. (2013, January 3). Multichannel serial-parallel analog-to-digital converters based on current mirrors for multi-sensor systems. In *Proc. SPIE Vol. 8550, Optical Systems Design 2012*, 855022. <https://doi.org/10.1117/12.2001703>.
31. Mori, M., & Yatagai, T. (1997). Optical learning neural networks with two dimensional structures. In *Proceedings of SPIE* (Vol. 3402, pp. 226–232).
32. Krasilenko, V. G., Bogukhvalskiy, A. K., & Magas, A. T. (1996). Designing and simulation optoelectronic neural networks with help of equivalent models and multivalued logics. *Proceedings of SPIE, 2824*, 135–146.
33. Krasilenko, V. G., Nikolskyy, A. I., & Lazarev, A. A. (2011). [*Design and simulation of time-pulse coded optoelectronic neural elements and devices, optoelectronic devices and properties*]. InTech . ISBN: 978-953-307-204-3. <https://doi.org/10.5772/16175>.
34. Krasilenko, V. G., Nikolskyy, A. I., & Lazarev, A. A. (2013). [*Design and modeling of optoelectronic photocurrent reconfigurable (OPR) multifunctional logic devices (MFLD) as the universal circuitry basis for advanced parallel high-performance processing, optoelectronics—Advanced materials and devices*]. InTech. ISBN: 978-953-51-0922-8. <https://doi.org/10.5772/54540>.
35. Krasilenko, V. G., Bardachenko, V. F., Nikolsky, A. I., & Lazarev, A. A. (2007). Programmed optoelectronic time-pulse coded relational processor as base element for sorting neural networks. In *Proceedings of SPIE* (Vol. 6576, p. 657610). Bellingham, WA: SPIE.
36. Huang, K. S., Yenkin, B., & Sawchuk, A. (1989). Image algebra representation of parallel optical binary arithmetic. *Applied Optics, 28*(6), 1263–1278.
37. Wang, J., & Long, Y. (2017). M-ary optical computing. In *Cloud computing-architecture and applications*. InTech.
38. Guilfoyle, P., & McCallum, D. (1996). High-speed low-energy digital optical processors. *Optical Engineering, 35*(2), 436–442.
39. Pituach, H. (2003). *Enlight256. White paper report*. Israel: Lenslet Ltd.
40. Krasilenko, V. G., Bardachenko, V. F., Nikolsky, A. I., Lazarev, A. A., & Kolesnytsky, O. K. (2005). Design of optoelectronic scalar-relation vector processors with time-pulse coding. *Proceedings of SPIE, 5813*, 333–341.
41. Krasilenko, V. G., Nikolskyy, A. I., Lazarev, A. A., & Lazareva, M. V. (2010). Design and simulation of programmable relational optoelectronic time-pulse coded processors as base elements for sorting neural networks. *Proceedings of SPIE, 7723*, 77231G.
42. Krasilenko, V. G., Nikolskyy, A. I., & Lazarev, A. A. (2014). Simulation of reconfigurable multifunctional continuous logic devices as advanced components of the next generation high-performance MIMO-systems for the processing and interconnection. *Proceedings of SPIE, 9009*, 90090R.
43. Kolesnitsky, O. K., & Krasilenko, V. G. (1992). Analog-to-digital converters with picture organization for digital optoelectronic processors. *Autometric, 2*, 16–29.

44. Kozshemjako, V. P., Krasilenko, V. G., & Kolesnitsky, O. K. (1993). Converters of halftone images in binary slices for digital optoelectronic processors. *Proceedings of SPIE, 1806*, 654–658.
45. Krasilenko, V. G., Nikolsky, A. I., Krasilenko, O. V., & Nikolska, M. A. (2011). Continuously logical complementary: Dual equivalently analog-to-digital converters for the optical systems. *Proceedings of SPIE*, 8001–8030.
46. Chakir, M., Akhamal, H., & Qjidaa, H. (2017). A design of a new column-parallel analog-to-digital converter flash for monolithic active pixel sensor. *The Scientific World Journal, 2017*. Article ID 8418042, 15 pages. <https://doi.org/10.1155/2017/8418042>.
47. Salahuddin, N. S., Wibowo, E. P., Mutiara, A. B., & Paindavoine, M. (2011). Design of thin-film-transistor (TFT) arrays using current mirror circuits. In *Livre/Conférence Journal of Engineering, Computing, Sciences & Technology, Asian Transactions* (Vol. 1, pp. 55–59).
48. Musa, P., Sudiro, S. A., Wibowo, E. P., Harmanto, S., & Paindavoine, M. (2012). Design and implementation of non-linear image processing functions for CMOS image sensor. In *Optoelectronic Imaging and Multimedia Technology II, Proceedings of SPIE* (Vol. 8558). Retrieved from <http://spie.org/Publications/Proceedings/Paper/10.1117/12.2000538>
49. Długosz, R., & Iniewski, K. (2007). Flexible architecture of ultra-low-power current-mode interleaved successive approximation analog-to-digital converter for wireless sensor networks. *VLSI Design, 2007*. Article ID 45269, 13 pages.
50. Roy, I., Biswas, S., & Patro, B. S. (2015). Low power high speed differential current comparator. *International Journal of Innovative Research in Computer and Communication Engineering, 3*(4), 3010–3016. <https://doi.org/10.15680/ijrccce.2015.0304089>.
51. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., Krasilenko, O. V., & Krasilenko, I. A. (2013). Simulation of continuously logical ADC (CL ADC) of photocurrents as a basic cell of image processor and multichannel optical sensor systems. *Proceedings of SPIE, 8774*, 877414.
52. Rath, A., Mandal, S. K., Das, S., & Dash, S. P. (2014). A high speed CMOS current comparator in 90 nm CMOS process technology. *International Journal of Computer Applications*. (0975–8887) International Conference on Microelectronics, Circuits and Systems (MICRO-2014).
53. Krasilenko, V. G., Nikolsky, A. I., & Parashuk, A. V. (2001). Research of dynamic processes in neural networks with help of system energy equivalence functions. In *Proceedings of the 8-th STC Measuring and Computer Devices in Technological Processes №8* (pp. 325–330).
54. Perju, V., & Casasent, D. (2012). Optical multichannel correlators for high-speed targets detection and localization. *Proceedings of SPIE, 8398*, 83980C.
55. Rudenko, O. G., & Bodiansky, E. V. (2005). *Artificial neural networks*. Kharkov: OOO SMIT Company. 408p.
56. Krasilenko, V. G., Nikolsky, A. I., & Pavlov, S. N. (2002). The associative 2D-memories based on matrix-tensor equivalental models. *Radioelektronika Informatics Communication, 2*(8), 45–54.
57. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Lobodzinska, R. F. (2009). Design of neurophysiologically motivated structures of time-pulse coded neurons. *Proceedings of SPIE, 7343*.
58. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Magas, T. E. (2010). Design and simulation of optoelectronic complementary dual neural elements for realizing a family of normalized vector 'equivalence-nonequivalence' operations. *Proceedings of SPIE, 7703*, 77030P.
59. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Sholohov, V. I. (2004). The concept of biologically motivated time-pulse information processing for design and construction of multifunctional devices of neural logic. *Proceedings of SPIE, 5421*, 183–194.
60. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Magas, T. E. (2012). Simulation results of optoelectronic photocurrent reconfigurable (OPR) universal logic devices (ULD) as the universal circuitry basis for advanced parallel high-performance processing. *Proceedings of SPIE, 8559*, 85590K.
61. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Mihalnichenko, N. N. (2004). Smart time-pulse coding photo-converters as basic components 2D-array logic devices for advanced neural networks and optical computers. *Proceedings of SPIE, 5439*.

62. Krasilenko, V. G., Nikolsky, A. I., & Lazarev, A. A. (2015). Designing and simulation smart multifunctional continuous logic device as a basic cell of advanced high-performance sensor systems with MIMO-structure. *Proceedings of SPIE*, 9450, 94500N.
63. Krasilenko, V. G., Ogorodnik, K. V., Nikolsky, A. I., & Dubchak, V. N. (2011). Family of optoelectronic photocurrent reconfigurable universal (or multifunctional) logical elements (OPR ULE) on the basis of continuous logic operations (CLO) and current mirrors (CM). *Proceedings of SPIE*, 8001, 80012Q.
64. Krasilenko, V. G., Nikolsky, A. I., Lazarev, A. A., & Pavlov, S. N. (2005). Design and applications of a family of optoelectronic photocurrent logical elements on the basis of current mirrors and comparators. *Proceedings of SPIE*, 5948, 59481G.
65. Krasilenko, V. G., Lazarev, A. A., & Nikitovich, D. V. (2018). Design and simulation of optoelectronic neuron equivalentors as hardware accelerators of self-learning equivalent convolutional neural structures (SLECNs). *Proceedings of SPIE*, 10689, 106890C.
66. Rodríguez-Quinonez, J. C., Sergiyenko, O., Hernandez-Balbuena, D., Rivas-Lopez, M., Flores-Fuentes, W., & Basaca-Preciado, L. C. (2014). Improve 3D laser scanner measurements accuracy using a FFBP neural network with Widrow-Hoff weight/bias learning function. *Opto-Electronics Review*, 22(4), 224–235.
67. Flores-Fuentes, W., Sergiyenko, O., Gonzalez-Navarro, F. F., Rivas-López, M., Rodríguez-Quinonez, J. C., Hernández-Balbuena, D., et al. (2016). Multivariate outlier mining and regression feedback for 3D measurement improvement in opto-mechanical system. *Optical and Quantum Electronics*, 48(8), 403.
68. Flores-Fuentes, W., Rodriguez-Quinonez, J. C., Hernandez-Balbuena, D., Rivas-Lopez, M., Sergiyenko, O., Gonzalez-Navarro, F. F., & Rivera-Castillo, J. (2014, June). Machine vision supported by artificial intelligence. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on* (pp. 1949–1954). IEEE.
69. Schlottmann, C. R., & Hasler, P. E. (2011). A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1(3), 403–411. <https://doi.org/10.1109/JETCAS.2011.2165755>.

Part II
Detection, Tracking and Stereoscopic
Vision Systems

Chapter 5

Image-Based Target Detection and Tracking Using Image-Assisted Robotic Total Stations



Volker Schwieger, Gabriel Kerekes, and Otto Lerke

Abbreviations

3D	Three dimensional Spatial extent, concerning the three dimensions/axis: x , y , z
ASCII	American Standard Code for Information Interchange
CCD	Charge-coupled device Light-sensitive chip
CMOS	Complementary metal-oxide-semiconductor Light-sensitive chip
EDM	Electronic distance measurement Distance measurements based on a modulated infrared light beam
FoV	Field of vision Visible part by the use of a telescope
Hz	Horizontal Concerns the spatial orientation and extension
IATS	Image-assisted total station Geodetic measurement device, extended by camera and laser scanner
ID	Identifier Unique tag feature
LED	Light-emitting diode —
MSAC	M-estimator sample consensus Filtering algorithm to avoid gross errors and increase the robustness

V. Schwieger (✉) · G. Kerekes · O. Lerke
Institute of Engineering Geodesy, University of Stuttgart, Stuttgart, Germany
e-mail: volker.schwieger@iigs.uni-stuttgart.de; gabriel.kerekes@iigs.uni-stuttgart.de;
otto.lerke@iigs.uni-stuttgart.de

MPixel	Mega pixel 1×10^6 pixel
PC	Personal computer
	–
RMS	Root mean square
	–
RTS	Robotic total station Geodetic measurement device
SIFT	Scale-invariant feature transform Image processing algorithm
SURF	Speeded-up robust feature Image processing algorithm
UAV	Unmanned air vehicle
	–
UGV	Unmanned ground vehicle
	–
V	Vertical Concerns the spatial orientation and extension

5.1 Introduction

Measuring angles and distances was of interest to mankind since ancient times. Transforming the geometry from theory (plans) into practice (field) necessitated instruments and tools that offer a basis or standard for comparison. On the one side, measurement units had to be defined, and on the other, real instruments capable of reproducing measurements had to be created. This process took centuries of research and development; therefore, a brief overview of significant historical steps that led to the state-of-the-art image-assisted total station is offered.

There are many technical developments that lead to the instrument used for angle measurement, known as the theodolite. The first description dates back to 1571, where Leonard Digges used a theodolite for a mapping campaign [1]. Mechanical developments led to constant improvements of these instruments, reaching a milestone in 1787 with Jesse Ramsden's great theodolite. Due to its accurate reading gradation, it is seen as the first modern theodolite [2]. The next significant step was taken in the early nineteenth century with the help of the transit theodolite; vertical and horizontal angle measurement were now verified by two-face measurements. Constant progress in optics and mechanics and the industrialization of the twentieth century lead to mass production of accurate theodolites. Out of the producers that stood out, the Swiss engineer Heinrich Wild [3] is worth mentioning. With the practical implementation of lasers in the late 1960s, the next generation of theodolites was the equipment with a distance measurement unit. This revolutionized surveying in many aspects and gave birth to the so-called tachymeter (total station). One of the first commercially available tachymeter was the Reg Elta 14 produced by

Carl Zeiss in 1969 [4]. Besides measuring all polar coordinates, it could store data and perform rudimentary calculations, thus setting a trend for the upcoming total stations.

Since then, the focus was mainly on improving the accuracy and reducing the size of these instruments, a fact that has been achieved within a short period. In the 1970s, automation of the measuring process gained attention. The idea of target identification and tracking was a highly discussed topic for geodetic instruments since the 1980s [5] and got to be implemented, to our concern, in several research projects of universities and industrial measurement systems [6]. One of the reasons that drove this progress was target identification and tracking for military purposes. First implementations of such instruments already happened during World War II with the so-called kinotheodolite Askania used for tracking smooth moving objects like missiles or airplanes. After the war, similar kinotheodolites were produced by Contravers (EOS) and are still being used today for similar purposes [7].

For civil purposes, however, a passive or active reflector (prism) in combination with motorized tachymeters was used in the tracking process. These have seen applications starting from the mid-1990s in machine control and guidance, deformation monitoring, or robot track definition [8]. Since then, all large manufacturing companies of geodetic instruments started producing robotic total stations (RTS). The next prospective improvements started in the year 2000 at different research institutes by incorporating an external digital camera onto a RTS. This opened a new spectrum of possibilities with an alternative positioning method that originates from photogrammetry. This synergy also reminds of the historical phototheodolite that was developed in Italy in 1865 by Porro [9], which can be considered among the predecessors of modern Image-Assisted Total Stations (IATS).

After receiving positive outcomes from the science community, incorporation of coaxial or overview CCD or CMOS sensors into the instrument's telescope [10] has become standard for a high-end IATS. This led to a new generation of instruments, originally called videotheodolites [6], with current state-of-the-art instruments like the Trimble SX10 that has no eyepiece (optical telescope axis) and totally relies on several telescope-integrated cameras [11]. These combine the precise 3D positioning capability of a tachymeter with image capturing systems [12] and through the known coordinates of the IATS, the orientation of the build-in camera is automatically given all the time [13]. Figure 5.1 represents a time axis with some of the above-mentioned generation of instruments.

These are some of the important milestones that led to the further presented IATS. In the second section, the functional principles of these multi-sensor systems are briefly explained. Section 5.3 presents the methods of automatic target recognition and tracking principles while using a reflector. Section 5.4 shows the methods needed for target identification using images and, further on, target tracking. Afterwards two applications are presented in Sect. 5.5, and the last section provides a quality assessment approach for positioning applications.



Fig. 5.1 Examples of different generation theodolites/total stations

5.2 Principles of Robotic Image-Assisted Total Stations

Robotic total stations are also called tachymeters. According to Joeckel et al. [14], the assembly architecture is subdivided into following component groups: sensors, actuators, storage, power supply, and man–machine interfaces (keyboard, touchscreen, and handhelds). These components are connected to a common microprocessor, which processes and coordinates various data. Each component group consists of multiple system items. For example, the component group “Sensors” contains the angle detection device, distance measurement device, optical sensor for target tracking, electronic levels, compensators, temperature, and pressure sensors. In the last evolutionary stage of total stations, the devices have been extended by cameras and laser scanners. Thus, the total station has become a versatile instrument and may be used for many applications. It can also be defined as a multi-sensor system (Fig. 5.2).

Image-assisted total stations (IATS) combine the standard robotic total station with cameras. Thus, they extend the classical geodetic scope of applications by photogrammetric techniques. Moreover, the camera module can be used for documentation purposes. The integrated CCD (charge-coupled device) chip has been formerly used for the target tracking in order to realize a one-man station. Within the IATS, an additional CCD undertakes camera functionalities.

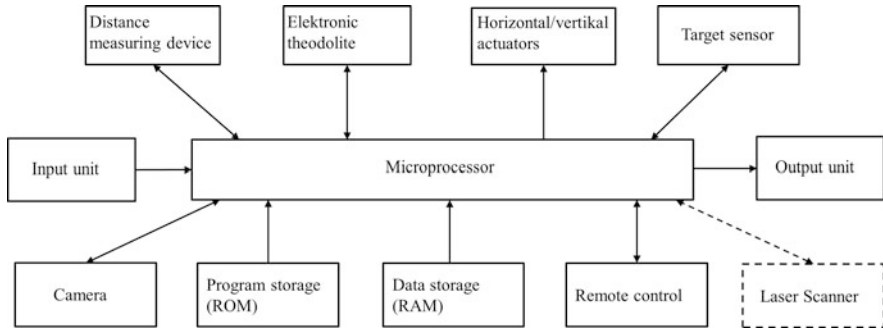


Fig. 5.2 Principle layout of a total station (modified from [14])

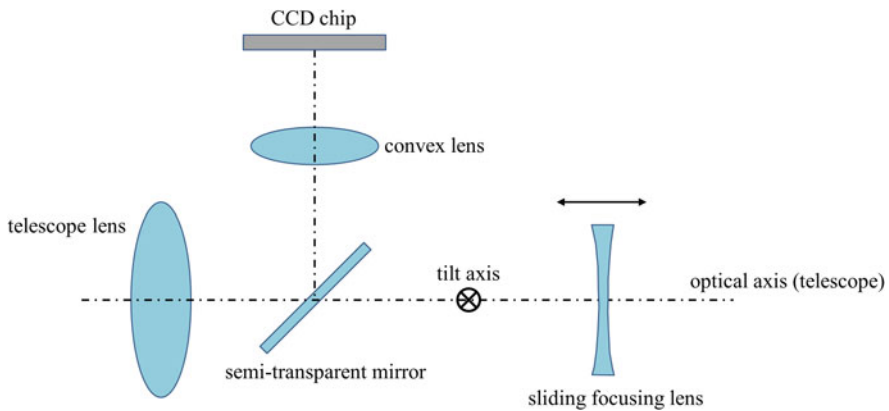


Fig. 5.3 Coaxial layout (according to [15])

Thereby different assembling layouts of the camera module within the housing of the total station exist. These are the coaxial layout, the eccentric layout, and the centric layout, with separate image beam. The layouts are designed by the manufacturers of the instruments. In the following, the three different layout designs are schematically depicted.

In the coaxial layout (Fig. 5.3), the optical axis of the telescope is aligned with the optical axis of the camera. Therefore, a semi-transparent mirror is installed, according to Fig. 5.3, which deflects the camera axis and consequently bundles the two beam paths. The advantage of such a realization is that both fields of vision are centered on each other. The disadvantage is the high optical complexity [15].

In the eccentric layout (Fig. 5.4) the two beam paths are spatially separated. As a result, the optical camera axis is parallel to the optical telescope axis.

The advantage of this layout is the simple optical arrangement, in contrast to the coaxial layout. The disadvantage is caused by the eccentricity, which induces a distance-dependent direction error. To correct this error, the measurements in the pixel domain (Sect. 5.4.5) must be extended by correction terms or functions in row and column directions. These terms and functions are either given by the

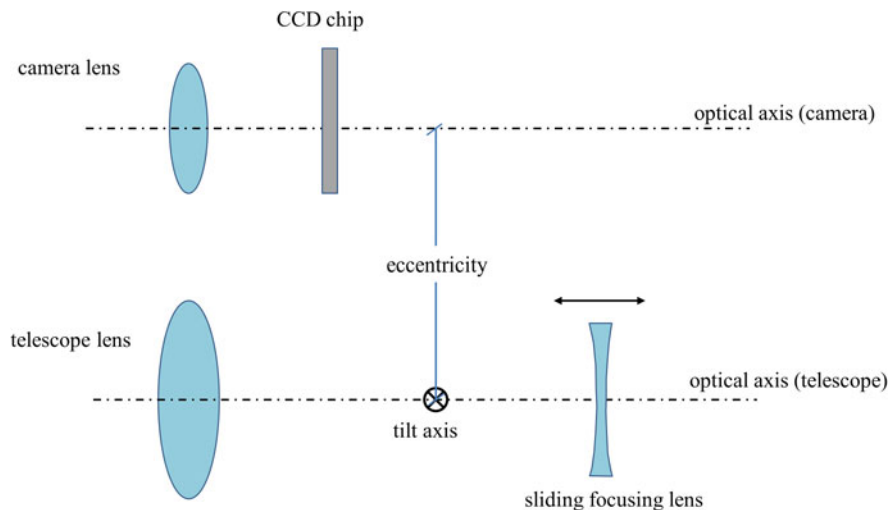


Fig. 5.4 Eccentric layout (according to [15])

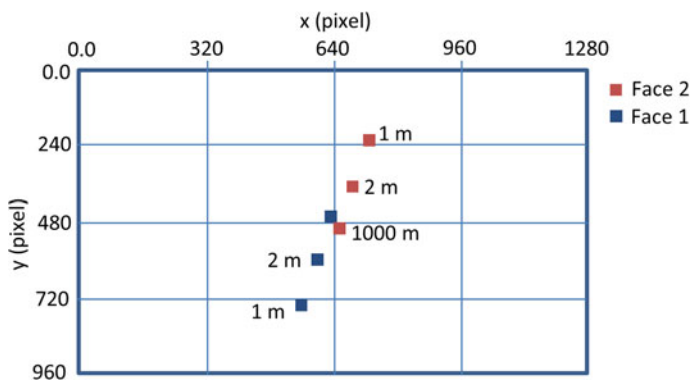


Fig. 5.5 Differences between cross-hair and image center [16]

manufacturer or must be determined by calibration. The calibration is valid for one particular instrument and cannot be generalized.

The difference between the cross-hair targeted point and the camera center has to be known for each zoom level. In Fig. 5.5, an example is given for one resolution of an overview camera and three distances.

It can be seen that with increasing distance, the difference caused by the parallax decreases.

A correction term has been exemplarily determined for a Trimble S7 total station which works with the eccentric layout. The calibration procedure reveals a constant offset $k_h = -6$ pixel in the row-direction (horizontal) and a function of shape $k_v = a \cdot x^b + c$ in the column direction (vertical). After the estimation of the parameters a , b , and c by calibration measurements, the following correction function could be established:

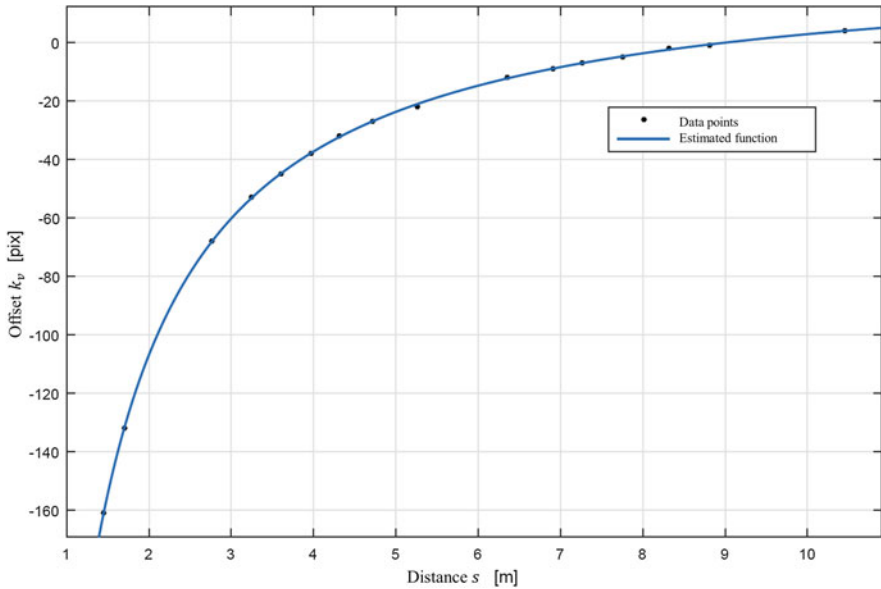


Fig. 5.6 Correction function for vertical direction in the pixel domain

$$k_v(s) = -278.8 \cdot s^{-1.051} + 27.61 \tag{5.1}$$

with

k_v : correction term

s : distance

Figure 5.6 depicts the course of the correction value, dependent on the measured target distance.

In order to gain benefits of the eccentric layout, without dealing with the pre-described distance-dependent direction error, a third layout may be realized. Thereby the two optical axes are tilted to each other by the angle Δ (Fig. 5.7). The intersection point of the two axes is located at the telescope’s tilting point (axis). The included angle Δ induces a height index error. The error is defined as the deviation in the vertical graduated disk reading of the tachymeter if it points to the zenith. A further disadvantage is that the two beam paths imaging different fields of vision (FoV). Depending on the aperture angles, it may happen that the FoVs are different [15].

5.2.1 Working Principles of Standard Total Station

The basic components regarding the classical (not image-based) determination of positions are the electronic theodolite, providing horizontal and vertical angle measurements and the distance measurement device. The combination of angle and

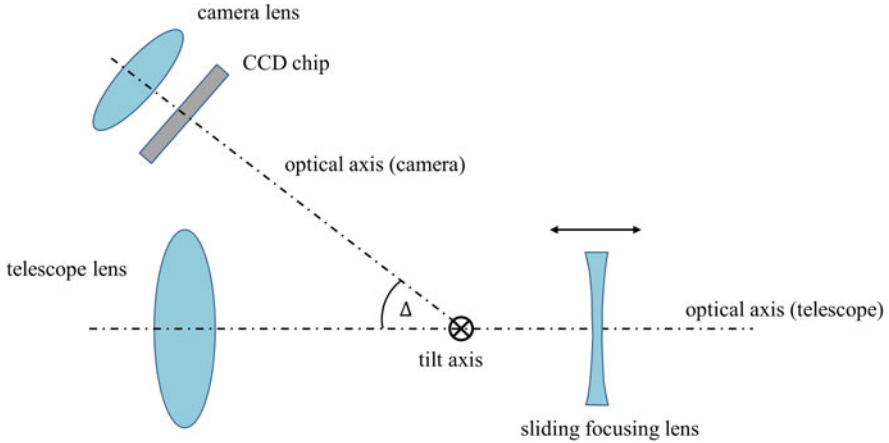


Fig. 5.7 Centric layout, with separate image beam (according to [15])

distance measurements allows the calculation of 3D point coordinates. However, the same principles are valid for IATS.

5.2.1.1 Electronic Distance Measurement

The electronic distance measurement (EDM) is based on time-of-flight measurement of a signal emitted from a transmitter and received by the receiver [8]. The signals may be infrared light, laser light, or micro waves. These signals are modulated with information [14]. Among different distance measurement procedures, two procedures are the most common for total stations: the pulse method and the phase-based method. In modern total stations, both methods are used separately. However, the interference measurement by the use of an interferometer is the most accurate method, but will not be presented in detail within the frame of this contribution, since total stations do not work with this principle.

Pulse Method

The pulse method is based on a sharp defined pulse, which is modulated and emitted. The time difference between the signal emission and its receipt is defined as signal propagation time t (Fig. 5.8). The distance d is calculated by the propagation speed of the wave c and the exactly measured signal propagation time [14].

$$d = \frac{c \cdot t}{2} \quad (5.2)$$

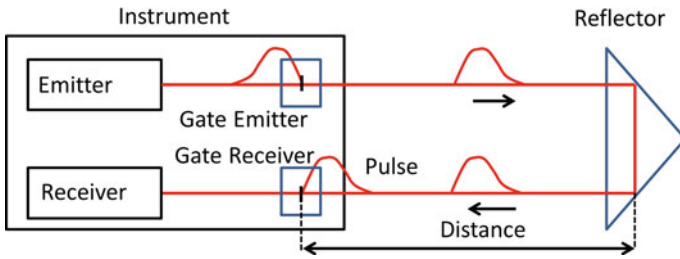


Fig. 5.8 Principle of pulse method [17]

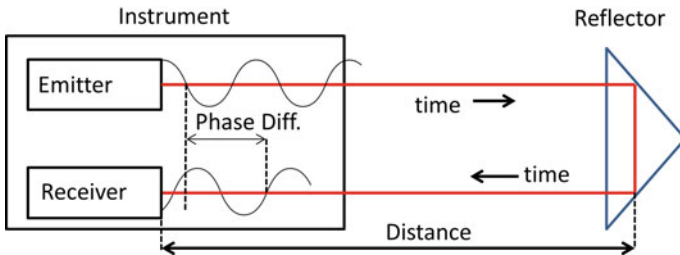


Fig. 5.9 Principle of phase-based method [17]

with

d : distance

c : speed of light

t : propagation time

The denominator factor 2 is justified by the fact that the emitted signal travels to the target and back to the instrument. Hence, the measured result represents the double distance.

Phase-Based Method

In this method, the carrier wave of the signal is modulated by a long-periodic oscillation. The distance d is defined by an unknown number n of complete waves, or oscillations of length λ and the remaining wave piece of length $\Delta\lambda$ (Fig. 5.9). $\Delta\lambda$ is measured by a precise phase meter. Hence, the distance is obtained by Eq. (5.3) [14].

$$d = \frac{n \cdot \lambda + \Delta\lambda}{2} \tag{5.3}$$

where

n : number of waves

λ : wave length

$\Delta\lambda$: remaining wave fraction

The determination of n is based on the combined measure, consisting of rough and fine measurements, with different wavelengths λ_1 and λ_2 . The rough measurement, with the wave fraction $\Delta\lambda_2$, is used to determine the number of complete wavelengths or phase shifts of λ_1 . For the distance, the following relationship is valid [14]:

$$2 \cdot d = n \cdot \lambda_1 + \Delta\lambda_1 \quad (5.4)$$

$$2 \cdot d \approx \Delta\lambda_2 \quad (5.5)$$

Thus the number of waves can be approximated as follows [14]:

$$n \approx \frac{\Delta\lambda_2 - \Delta\lambda_1}{\lambda_1} \quad (5.6)$$

For n , the closest number to the approximated value is valid.

5.2.1.2 Electronic Angle Measurement

The angle measurement takes place by manual or automatic target aiming of the telescope and subsequent readings of the directions at the graduated disk. The reading is realized by different methods, described in the following.

Code-Based Method

Within the code-based method, the graduated disk is encoded (Fig. 5.10). Thus, to each disk position, a unique coded output signal is allocated [8]. The division consists of one or multiple adjoining radial lanes. The lanes consist of alternating translucent and opaque fields. Hence, the division is binary coded. The reading is realized by an above-located light-emitting diode (LED) and a subjacent photodiode, which receives the signal and converts it to an electrical signal.

Incremental Method

This method works with relative angles. The graduated disks are painted with radial gridlines, representing a sequence of transparent and opaque fields (increments).

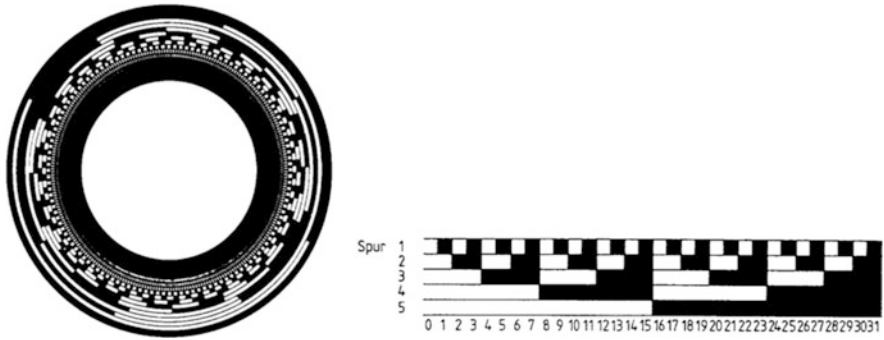
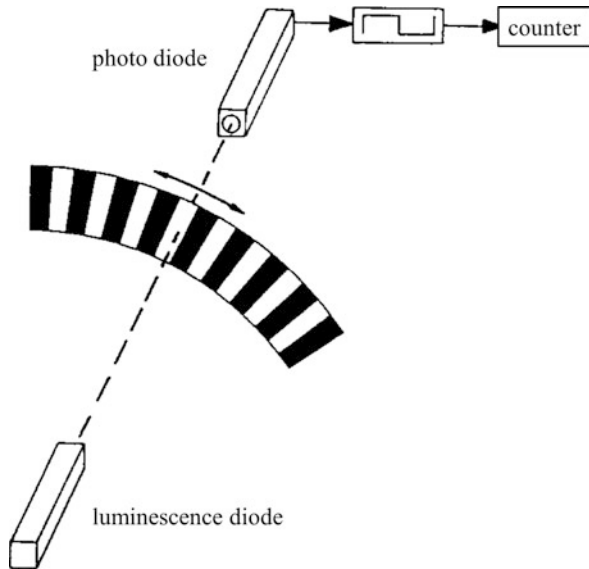


Fig. 5.10 Left: encoded graduated disk; right: binary code example [14]

Fig. 5.11 Incremental method (modified from [14])



The distance between those fields is called the grid constant. The sequenced number of bright and dark fields is captured and counted by the scanning system, consisting of LED and photodiode.

Additionally, the disk turning direction is important, in order to have the unique directions to be set in relation to each other. Thus a direction discriminator, consisting of two reading devices, is required [14] (Fig. 5.11).

Nowadays most of the total stations work with the incremental method.

5.3 Automated Reflector-Based Target Recognition and Target Tracking

5.3.1 Automated Target Recognition and Detection

The aim of automated target recognition is to enable the total station to measure targets autonomously. Thus the main advantage is the revolution of the workflow resulting in a one-man station. Moreover, the observer-dependent focus errors are eliminated. Before the automatic target recognition can be realized, some pre-conditions, listed below, have to be fulfilled up to now:

- The total station has to be motorized.
- The target has to be signalized by the reflector.
- Special sensors (image-based or others) are required.
- In some cases, a wireless connection is required.

The complete procedure consists of two steps which are sequenced in rough pointing/coarse search and fine pointing/fine aiming.

5.3.1.1 Rough Pointing/Coarse Search

The rough pointing requires no pre-information and is finished if the target is in the range of vision of the telescope. There are different realizations of rough pointing methods.

Method A requires special sensors for rough pointing within the tachymeter and a passive target reflector. The technique is based on the reflections of the reflector at the instrument. For this purpose, a laser plane fanned out vertically, with an opening angle of approximately 36° . At the same time, the instrument performs a horizontal rotation. The rotation stops, as soon as the instrument detects the reflected light of the reflector. In the following, the instrument begins with a vertical movement, where the laser beam is focused. The vertical movement stops if the reflection of the reflector is detected by the instrument again. Subsequently, the set horizontal and vertical angles can be tapped.

Method B is similar to method A, but the laser is fanned out with an opening angle of 10° . Thus more revolutions per scan are required. After each horizontal revolution, the vertical angle is changed.

Method C is similar to method A, but an active reflector emits a rectangular fan ($60.3^\circ \times 19.8^\circ$) to the instrument via a modulated laser or infrared signal. The signal also contains information about the reflector's unique ID (Fig. 5.12).

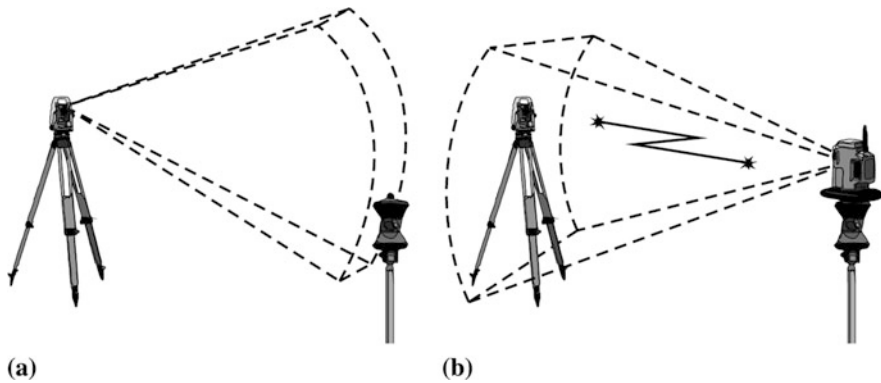


Fig. 5.12 Principle sketch of rough pointing; (a) methods A + B using passive prism; (b) method C using active prism (modified from [18])

5.3.1.2 Fine Pointing/Fine Aiming

The aim of the fine pointing is the determination of deviation between the telescope's crosshair and the center of the reflector. The pre-conditions are that the target is signalized by a reflector and that the reflector is inside the field of vision of the telescope. Thereby the second pre-condition is already fulfilled by rough pointing. Two general procedures, dependent on the manufacturer of the instrument, are implemented. These procedures are the image processing and the quadrant detector.

Procedure A is based on the detection of the reflected infrared light of a passive reflector by the instrument using a CCD array or a complementary metal-oxide-semiconductor (CMOS) array. The localization of the reflector's center is established by the determination of the geometric center within the image of the reflection. The geometric center is derived from the weighted intensity values of the illuminated cells. Therefore, the center of the reflector is determined in the optical system, and the CCD array system X' , Y' respectively. The difference between the crosshairs in the tachymeter's V - H_z system and the CCD array system X' , Y' is known in advance from calibration and is expressed by transformation parameters. This calibration is conducted by the manufacturer beforehand. By the use of the known transformation parameters, a transformation is carried out between the Y' - X' and the V - H_z system. As a result, the deviations ΔV and ΔH_z with respect to the known crosshair positions V and H_z are determined (Fig. 5.13).

Procedure B is based on the detection of the infrared light at the instrument emitted by an active reflector. For this, a quadrant or double-quadrant detector is used. Therefore, each quadrant needs to acquire light. After the light spot is visible in all quadrants, the direction of the prism center can be calculated by a linear function of the detected intensities. The alignment of the telescope and the prism is finished, if all quadrant intensities are equal. The double-quadrant detector enhances the target search and aiming operations. Therefore, the fine quadrants q_1 , q_2 , q_3 , and

Fig. 5.13 Principle sketch of V-Hz system and CCD system (procedure A) (modified from [14])

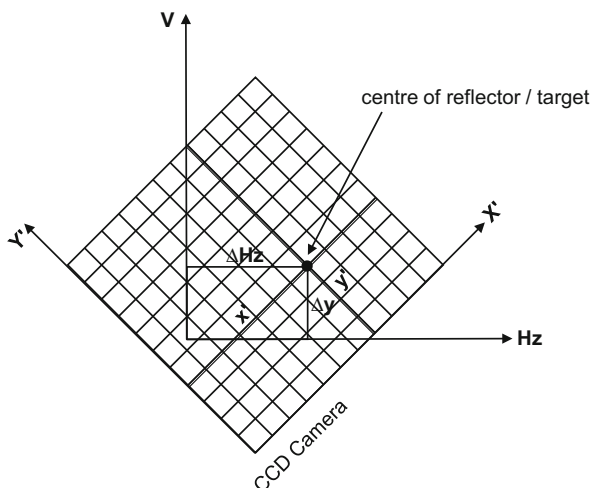
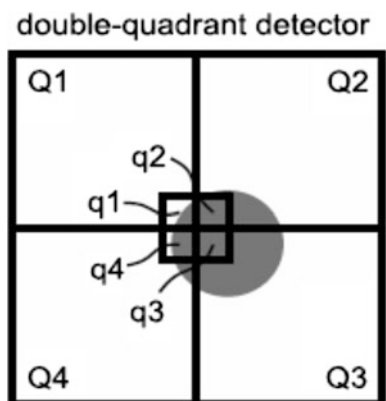


Fig. 5.14 Principle sketch of double-quadrant detector (procedure B) [18]



q4 (cf. Fig. 5.14) allow fine aiming of targets at longer distances (>25 m), because of their narrow field of view. Whereas the coarse quadrants Q1, Q2, Q3, and Q4 (cf. Fig. 5.14) are used for search and fine aiming of targets close to the instrument [18].

The procedure also allows the use of passive reflectors, where an infrared laser light is emitted from the total station and is reflected by the passive reflector [18].

After the determination of ΔV and ΔHz , the reading for V and Hz may be corrected, if deviations are greater than a specified threshold [18].

In general, servomotors realize a precise adjustment of the telescope for the target because first, a visual control by the observer becomes possible and second, distance measurements are possible. This is valid for both presented fine pointing procedures.

5.3.2 Target Tracking

Within the frame of this contribution, the meaning of the term “target tracking” is referred to the ability of the instrument to track the target with a pre-defined sampling rate. This is valid for reflector-based as well as reflector-less tracking procedure.

For target tracking procedure, active or passive targets may be used [8]. During the automatic tracking, the telescope permanently follows the target. The measured deviations ΔV and ΔH_z of the prism referred to the tachymeter’s axes system (crosshair) (Fig. 5.15) are minimized by the horizontal and vertical movements of the telescope, induced by the actuators [8].

The tracking mode is realized by a closed-loop system, including fine pointing procedure and actuators operations.

The sequence of actions within the closed loop, depicted in Fig. 5.16, is as follows: the deviations are determined by use of fine pointing procedures. The controller calculates the regulating variables for the motors to eliminate the deviation, which is the difference between the crosshair and the reflector center.

If the connection between the instrument and the reflector is interrupted and a subsequent loss of the target occurs, caused by obstacles, the deviations cannot be determined anymore. In that case, the controller calculates the regulating variable by the use of a prediction model for the movement, for example, by the use of

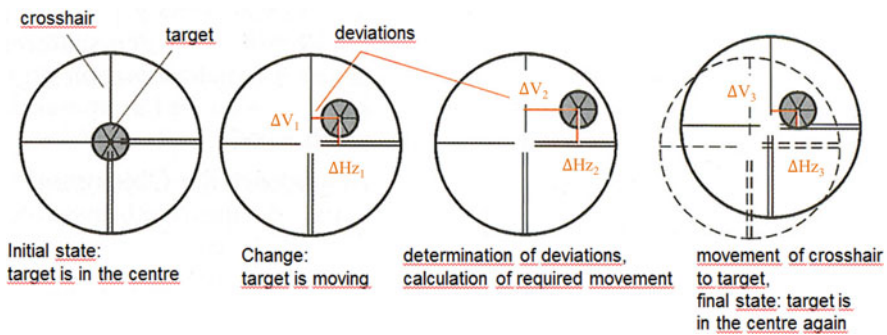


Fig. 5.15 Principle of target tracking (modified from [19])

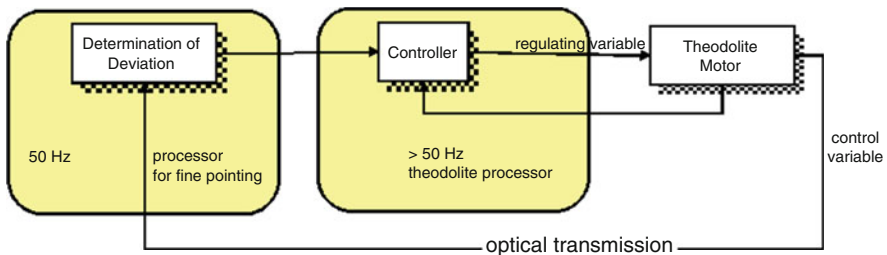


Fig. 5.16 Tracking mode closed-loop system [20]

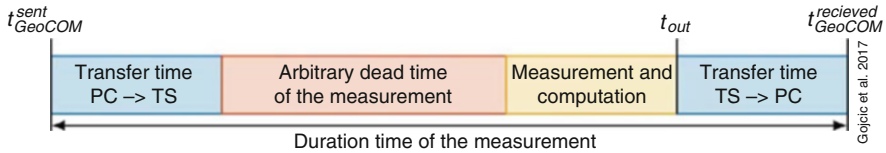


Fig. 5.17 Events order in the tracking process [21]

constant velocity information for the horizontal and vertical direction. The velocity information is derived from movements before target loss. By that procedure, it is possible to conduct a prediction for some seconds. If the prediction has not succeeded, a rough pointing is required a new [8].

5.3.3 Time Aspects in Object Tracking

Most developments make use of reflective corner-cube prisms that are used for finding and tracking an object. Furthermore, if the prism is followed with the aid of a target recognition device, measuring a distance between reflector and instrument is possible; therefore, polar coordinates are directly available.

It has been seen in Sect. 5.3.1 how a reflector is identified and tracked. One important aspect that influences all tracking processes is time synchronization of the individual measurements. Retrieving an angle reading, measuring a distance, and computing 3D coordinates are events that require time, and synchronizing these events is crucial for an accurate position. Similarly, acquiring images, processing the image coordinates, and transforming them into angular increments for the servomotors require time. Nowadays, even if processing power reached astonishing levels, latency still applies for state-of-the-art IATS, and the general latency scheme (Fig. 5.17) is available.

Another aspect is that many kinematic applications involve slow moving objects; therefore, the aforementioned effects become negligible. Some examples that use these principles are mostly found in guidance and control of construction machines.

5.4 Image-Based Object Recognition, Position Determination, and Tracking

This section treats the usage of an image-assisted total station (IATS) in order to perform static object recognition, position determination, and tracking. Within this context, the potential of the integrated photo module of the IATS should be fully exploited and not be limited to documentation purposes only.

5.4.1 *Image Processing Fundamentals*

According to Luhmann [22], the digital image processing can be subdivided into different steps: image capturing, pre-processing, segmentation, detection, clustering, configuration, and result assessment.

The pixel coordinate system has a fundamental function in the image processing task. It is defined as a left-handed two-dimensional x - y coordinate system, described by rows and columns, whereby the x -axis pointing into the direction of rows and the y -axis into the direction of columns [23]. In general, an object in the image covers multiple pixels in the image. These covered pixels are coherent and their gray values are rather similar.

To increase the efficiency of image processing, image pyramids are used. An image pyramid is a series of images, where the next following image is reduced in resolution and size by factor n , compared with its predecessor. Additionally, the reduced image is smoothed by filter. Thus with decreasing resolution, small image structures disappear because of the informational content decreases [22]. This allows to first search for rough features in images with decreased resolution. Afterward the search can be focused on previously found, interesting areas of the image, by the use of images of the pyramid with higher resolution.

Filtering can be realized by convolutions in the spatial domain and multiplications in the frequency domain. For frequency domain operations, the image must be first transformed into the frequency domain, for example, by Fourier transformation [22, 23].

The convolution is realized by a stepwise “sliding” of the convolution kernel across the image. The pixel value beneath the kernel is multiplied by the corresponding filter kernel value. These products are subsequently summed and multiplied by the sum of the kernel values. The resulting value is then allocated to the pixel, which best corresponds to the mean value of the filter kernel [22].

There are different smoothing filter used in digital image processing. The most import smoothing filters are the Box filter and the Gaussian filter. Detailed information on smoothing filters may be found in Luhmann [22].

Besides smoothing filters, edge detector filters play a superior role in image processing. They are based on numerical derivations of gray value functions and are used to locate sharp changes in gray values, which in turn, indicate edges. Edges are pixels, where the first derivative of the gray value function changes abruptly [23].

The simplest edge detector is the Roberts detector, based on the first derivative of the pixel plane in x and y directions [24]. The Sobel operator combines a derivation with a smoothing. This helps to counteract the amplification of noise induced by the derivation. To obtain further information about the edge curvature, the Laplace operator can be applied. It is based on the second derivative of the gray value function. Thereby the edges are represented by significant changes. On the other hand, the second-order derivative negatively affects the noise sensitivity. To counteract this adverse effect, the image can be smoothed by the Gaussian filter before derivation.

The combination of smoothing and differentiation leads to the Laplacian of Gaussian operator.

The main disadvantage of edge detectors is the instability of their position. Edges are stable in only one direction. In contrast, most image processing algorithms need features with stable positions. Therefore, corners are more suitable because of their fix localization in both directions. One of the most common corner detectors is the Hessian detector. The detector is based on the usage of the determinant of the Hessian matrix [25]. The Hessian matrix is defined as follows [26]:

$$\mathcal{H}_f = \frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix} \quad (5.7)$$

The feature detection is accomplished by the analysis of the determinant. The feature is located at the position where the determinant indicates a maximum. According to Merziger and Wirth [26], the determinant is defined as follows:

$$\det \mathcal{H} = D_{xx} \cdot D_{yy} - (w \cdot D_{xy})^2 \quad (5.8)$$

The elements D_{xx} , D_{yy} , and D_{xy} are so-called Blob filter responses at image point \mathbf{x} . They represent a 9×9 Box filter. w represents the weighting factor.

In the next section, the usage of the presented image processing tools will be illustrated within common image processing algorithms.

5.4.2 Image Processing Algorithms for Feature Extraction

There are many possibilities of image processing that originate from the field of computer vision [27] and are used for such purposes, but these can be grouped in three classes: edge-based, template-based, and point-based [12]. As the name already suggests, edge-based implies identifying edges of an object and then computing, if necessary, its geometrical center. In this way, if the geometry does not change and the background shows contrast, objects can be easily detected based on their edges. Template-based involves using a pre-known pattern that the algorithm recognizes. Therefore, patterns are always compared with the reference pattern, and if a match is found, the object is considered to be the searched one. Point-based implies finding certain features (points) that, similarly to the template-based, match a pre-known image. Two well-known point-based feature extraction algorithms will be detailed in the following sections. The process of feature extraction generally consists of two steps: detection and description.

5.4.2.1 SIFT (Scale-Invariant Feature Transform) Algorithm

According to Lowe [28], the requirements for image-based object recognition from real world are that image objects, so-called features, that are randomly arranged in space and partly covered, are identified and detected uniquely. The features shall be invariant with respect to translations, rotations, scaling, and changes in illuminance. Furthermore, the features shall be unaffected by image distortions and noise. To fulfill the requirements, the image features must possess characteristic shapes, in order to be identified uniquely. The SIFT algorithm decomposes the image into a finite number of objects, which are described by local descriptor vectors.

The required processing step after image capturing is the localization of the characteristic image features. Their positions, identified in the space domain, must fulfill the invariance properties. The requisite mathematical steps are comprised in the so-called detector.

The detector is based on two convolutions of the image with the Gaussian kernel and a subsequent forming of Gaussian difference in order to detect curvatures (cf. Sect. 5.4.1). The Gaussian difference approximates the Laplacian of Gaussian in order to shorten the processing time. After resampling the image by the use of bilinear interpolation [22], the search for local maxima and minima within different levels of image pyramids is conducted, where the neighboring pixels of the image are compared with each other.

The detector step is followed by the descriptor step in order to characterize the image. For this purpose, the feature gradient and orientation are calculated. This is done for each pixel $A_{i,j}$ by the calculation of the gradient magnitude $M_{i,j}$ and the orientation $R_{i,j}$.

$$M_{i,j} = \sqrt{(A_{i,j} - A_{i+1,j})^2 + (A_{i,j} - A_{i,j+1})^2} \quad (5.9)$$

$$R_{i,j} = \arctan \left(\frac{A_{i,j} - A_{i+1,j}}{A_{i,j+1} - A_{i,j}} \right) \quad (5.10)$$

Figure 5.18 exemplarily depicts the descriptor.

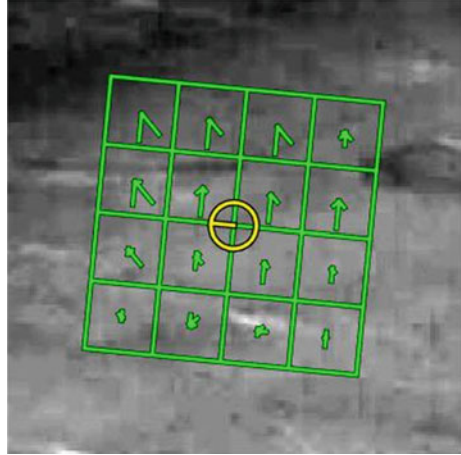
The descriptor vector is created for each feature and is of dimension $n = 128$. It contains the feature's stable position in the image, the scale, and the orientation. Optionally, the descriptor might additionally contain colors or textures.

Detailed inside view into the SIFT algorithm is provided by Lowe [28].

5.4.2.2 SURF (SPEEDED-UP ROBUST FEATURE) ALGORITHM

According to Bay et al. [25], the most important property of the detector is the repeatability. In this context, repeatability means the reliability of the detector

Fig. 5.18 Feature descriptor;
 yellow: orientation $R_{i,j}$;
 green: magnitude $M_{i,j}$



to identify and find the same physical object under changing visual conditions. Therefore, the neighborhood of each relevant image point is described by the descriptor. The descriptor must be distinctive and robust toward noise, translations, as well as geometric deformations and photogrammetric distortions. By the use of feature descriptors from two different images, these features and even the images (usage of multiple features) can be allocated to each other. For example, the allocation is based on the Euclidean distance between the two descriptors. The dimension of the descriptor has a direct impact on the allocation time and thus on the computation duration. Hence, small dimensions of the descriptor vector are desirable on the one hand. On the other hand, small descriptor dimensions are less unique and thus less distinctive [25]. SURF algorithm offers a good compromise between short processing times and sufficient descriptor dimension in order to ensure distinctiveness. SURF uses the scale- and rotation-invariant detectors and descriptors. No color information is used.

The detector is based on the usage of the determinant of the Hessian matrix (cf. Sect. 5.4.1). Therefore, integral images are introduced. Integral images serve for fast computations of pixel sums within rectangular sections [25]. The input of the integral image $I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j)$ at position $\mathbf{x} = (x, y)^T$ is the sum of all pixels of the input image I within a rectangular region, spanned between the image origin and image point \mathbf{x} .

Now the Blob filter responses, which can be used to calculate maxima and minima, are stored in a Blob response map. The map represents the image scale space. The scale space is implemented by image pyramids and is divided into octaves. The octaves represent series of filter responses, determined by convolution. Each octave is subdivided into a constant number of scaling levels. Hence, the detector contains the steps of suppression of the input image and the determination of the features by the described procedure, using the Hessian determinant. The use of integral images increases the computational speed and enhances the robustness [25].

The descriptor uses the intensity values to characterize the features. It is based on the distribution of the first-order responses of the Haar wavelet in x and y directions. Details on Haar wavelets can be extracted from Talukder and Harada [29]. For the descriptor, based on the Haar wavelet, first a rectangular region around the point of interest is built. The determination of orientation of each region is accomplished by the detector step. Then, the regions are subdivided into smaller rectangular regions. The Haar wavelet responses of these smaller regions are calculated, where dx is the response in x direction and dy is the response in y direction. Information about the polarity and change of intensity is obtained from $|dx|$ and $|dy|$. For the description of the intensity structure of each subpixel region, a four-dimensional descriptor vector, shaped as $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$, is then established. The local sub-region descriptor vector is then calculated for all sub-regions, which are $4 \times 4 = 16$ in total. Hence the descriptor vector for each feature is of dimension $n = 16 \times 4 = 64$.

More detailed information on SURF algorithm may be extracted from Bay et al. [25] (Table 5.1).

5.4.3 Object Recognition and Matching

After the extraction of the features is accomplished by one of the pre-described algorithms of Sect. 5.4.2, the next operation, in the sequence of image processing, is object recognition. This procedure is based on a comparison between the reference image and the test image and is defined as matching step. In detail, the extracted key points (cf. Sect. 5.4.2) of the reference and test image are compared with each other. For the comparison, n -dimensional feature vector (SIFT $n = 128$, SURF $n = 64$) is defined as the position of the key points in n -dimensional space. The task to be solved is the finding of next neighbors between the two feature vectors (reference and test image) by the use of the Best Bin First algorithm, according to Muja and Lowe [30]. Best Bin First algorithm is suitable to efficiently find an approximate solution to the nearest neighbor search problem in very-high-dimensional spaces [31]. The procedure is based on a binary-coded description of the feature vector. It has to be stated that this solution provides an approximate solution only.

Table 5.1 Comparison between SIFT and SURF

	SIFT	SURF
Algorithm input	Grayscale images	Intensity images
Used filter	Original filter	Approximated filter
Structure of the scale space pyramid	Different resolutions of the image	Different resolutions of the filter
Base of the descriptor	Gradients	Haar wavelet filter response
Descriptor dimension	128-dimensional	64-dimensional
General property	More reliable	Faster

The key points, expressed by the feature vector created by SIFT or SURF, that are passed to the algorithm are subdivided in k clusters. Therefore, k randomly chosen points form the cluster center. The remaining points are allocated to the particular cluster, to which their distance is minimal. If the cluster is larger than a predefined threshold, new cluster centers are chosen, and the algorithm starts anew. This helps to prevent different cluster sizes, if unfavorable cluster centers have been chosen beforehand. Each cluster forms a k -dimensional tree, the so called k -d tree. The algorithm is advantageous in tree building and during search operations, because of the parallel and simultaneous processing of different trees. The search for the nearest neighbor in the tree is conducted from top to bottom, where at each branch the nearest node to the starting point is marked. Non-marked nodes are stored in a separate priority list. After all trees are searched once, the search proceeds from the next point which is nearest to the starting point. Now the next nearest neighbors of particular k -d trees from the priority list are compared. The points with the minimal distance are chosen. The number of chosen points defines the approximation grade. The higher the grade, the more neighbors are found, though the processing time increases.

By the use of the recognized point pairs, the transformation matrix between the two images can be determined. The matrix allows transformations of pixels/points with respect to points from the reference image into test image. This procedure is called pixel-to-pixel transformation.

To avoid gross errors in the point cloud and to increase the robustness, a filtering by the MSAC (M-estimator sample consensus) algorithm is applied. Detailed information on MSAC may be found in Torr and Zisserman [32]. The transformation itself can be expressed by an affine transformation according to Lowe [28]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (5.11)$$

where

u, v : test image point coordinates

x, y : reference image point coordinates

t_x, t_y : translation parameters

m_1, m_2, m_3, m_4 : rotation and scale parameters

Thus, two tasks have to be solved. The first task is the determination of the transformation parameters. These are determined by the least squares method according to Niemeier [33]. In order to estimate the six parameters, the minimum number of requisite matches between the reference and the test image must be three. In the second task, all points of the reference image can be transformed into the test image by the estimated transformation parameters.

5.4.4 Object Position Determination

After the successful matching and identification of the object in the test image according to Sect. 5.4.3, the next challenge is the position determination of the specific object.

In general, the determination of object positions in the target coordinate system, which is the tachymeter system in this specific case, requires horizontal and vertical telescope angles, as well as a distance measurement, obtained by the reflector-less distance measurement (EDM). Hence, the obtainment of the horizontal and vertical telescope angles from images is necessary and will be elaborated in the following.

In each reference, image key points are defined, of which the pixel coordinates in the reference image, as well as their coordinates in the object coordinate system are known. The geometry of the object is also fully known in the object coordinate system. By using the pre-described pixel-to-pixel transformation from Sect. 5.4.3, the transformation parameters between the reference and the test image can be obtained. The next step is the transition from the image system into the tachymeter system. Therefore, determination of the telescope's aiming direction, expressed by Hz and V angles, from the present pixel coordinate of the appropriate point must be determined. For this, the relation between pixel and angle is required. This relation, described by the transfer factor i , is different for each instrument and is either given or must be determined by calibration. The relation describes the function between a specific telescope angle α and the induced shift p in the pixel system. Thus the transfer factor i can be expressed as follows:

$$i = \frac{p}{\alpha} \quad (5.12)$$

Reconsidering the correction terms, introduced for the eccentric camera-telescope layout in Sect. 5.2 (cf. Fig. 5.4), the calculation of the horizontal and vertical telescope angles from the present image point is carried out by using Eq. (5.13) and Eq. (5.14).

$$\text{Hz} = i \cdot \left(\text{hpix}_g - (\text{hpix}_m + k_h) \right) \quad (5.13)$$

$$V = i \cdot \left(\text{vpix}_g - (\text{vpix}_m + k_v) \right) \quad (5.14)$$

where

$\text{hpix}_g, \text{vpix}_g$: pixel coordinates (row and column) of the measured object in the image

$\text{hpix}_m, \text{vpix}_m$: pixel coordinates (row and column) of the image center

Hz: horizontal telescope angle

V: vertical telescope angle

k_h : horizontal correction term (valid for eccentric layout only)

k_v : vertical correction term (valid for eccentric layout only)

By using Eq. (5.13), the telescope directions of the key point can now be calculated. These directions can be adjusted by the tachymeter's actuators. By the additional use of the reflector-less distance measurement s , all required elements for the coordinate calculation are available. According to Torge [34], the coordinates are calculated as follows:

$$x = s \cdot \cos H_z \cdot \sin V \quad (5.15)$$

$$y = s \cdot \sin H_z \cdot \sin V \quad (5.16)$$

$$z = s \cdot \cos V \quad (5.17)$$

5.4.5 Principles of Image-Based Object Tracking

Basically, when using images, a certain object of interest must be identified, and some features need to be extracted for further processing. In Sects. 5.4.2.1 and 5.4.2.2, SIFT and SURF, which are point-based algorithms, were described in detail and further emphasis will not be put here. The main difference between different algorithms is the computation time, which plays an important role in the image-based tracking process.

Similar to a reflector tracking process, the difference between the crosshair point and object center needs to be constantly minimized. Therefore, if change or movement of the object is detected, the telescope is guided until the before-mentioned difference (cf. Sect. 5.3.2) is reduced to zero. Applying this in a continuous sequence creates the image-based tracking process. The quality of this process is mainly dictated by image resolution, optical zoom capacity, data transfer rate, processing speed, object speed, and telescope rotations speed.

5.5 Applications

5.5.1 Example of Static Object Recognition and Positioning

In this example, the position of an unmanned ground vehicle (UGV) should be determined by the presented image-based method. The concerned object is a model of a tracked loader at scale 1:14 (Fig. 5.19).

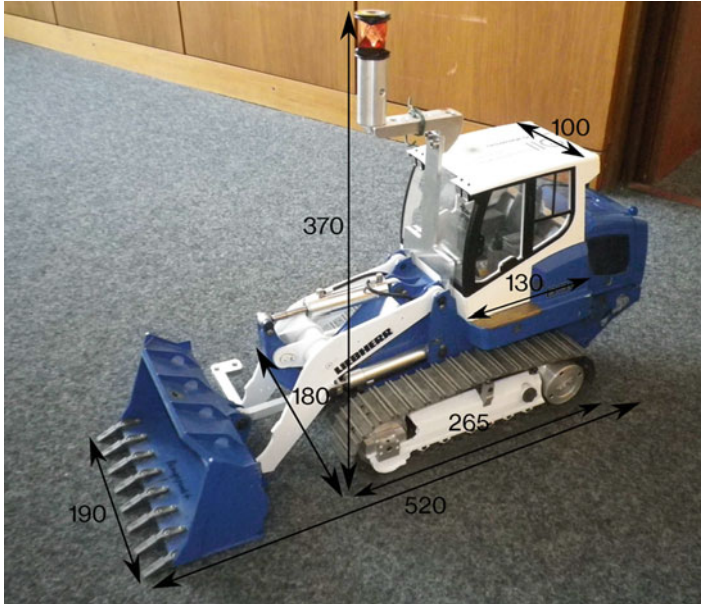


Fig. 5.19 Tracked loader model (units are given in millimeter)

The used instrument is the Trimble S7 robotic total station, including a digital camera which has a field of view of $20.3^\circ \times 15.2^\circ$ [35].

The following presented steps are embedded within a control program, in order to automatically steer the total station. Most state-of-the-art total stations can be steered externally by receiving commands from laptops or PCs via defined interfaces. This allows the user to create application and problem oriented programs. The steering possibilities concern almost every component group of the tachymeter (cf. Sect. 5.2). The interfaces also allow outsourcing image processing algorithms and others to external devices, in order to not overstrain the internal processor of the total station. In the current configuration, the steering program for the Trimble S7 total station is implemented in the programming language C#. The image processing algorithms are implemented in Matlab©. The superordinate control program, which coordinates and synchronizes the data flow between the individual programming components, is realized in the graphical programming language LabView from National Instruments.

The implemented steering program let the total station automatically move the telescope by pre-defined angles in vertical and horizontal directions, capture images, and deploy reflector-less distance measurements.

The flowchart of the specific total station steering program for object recognition and positioning is depicted in Fig. 5.20.

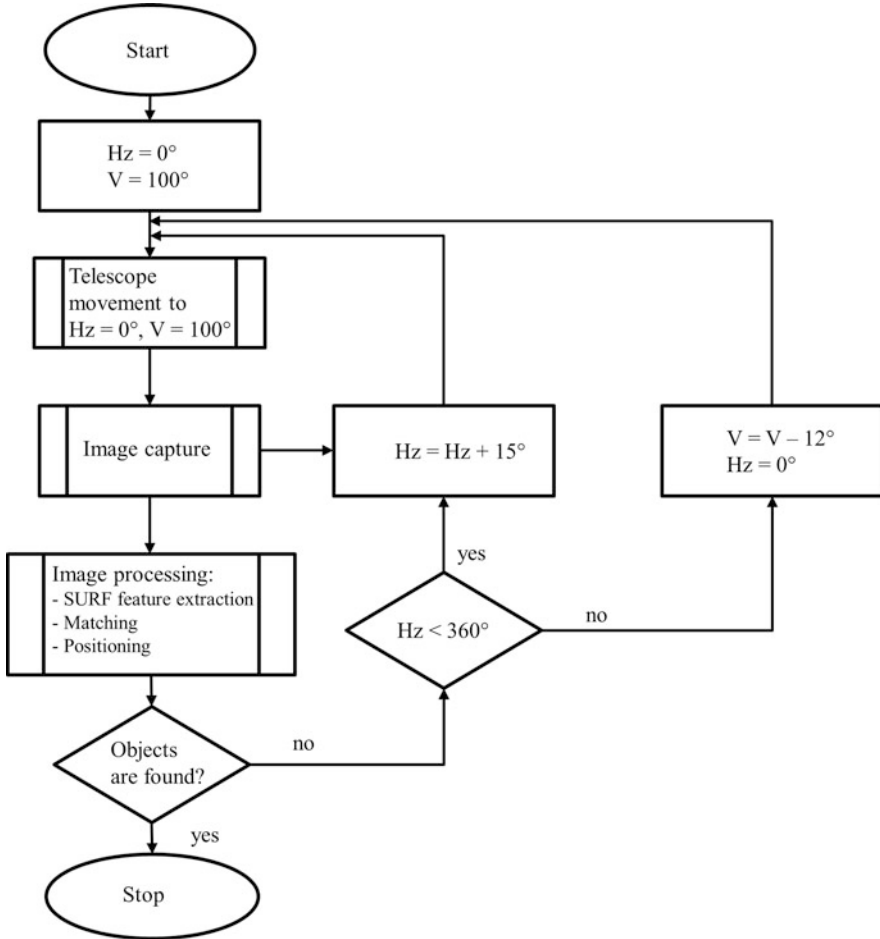


Fig. 5.20 Flowchart of static, image-based object recognition

After the image has been captured by the camera module, the feature extraction by the use of the SURF algorithm, according to Sect. 5.4.2, is conducted. The extraction result in the reference image is depicted in Fig. 5.21.

Accordingly, Fig. 5.22 shows the extraction result in the test image. The image was captured from another perspective than the reference image. This should underline the performance of the SURF algorithm, where test images might be taken from a different perspective, but the matching robustness is still given.

It can also be seen in Fig. 5.22 that features that are neither part of the object nor part of the reference image have been detected and extracted.

Subsequently the matching step, according to Sect. 5.4.3 is performed. The matching result is shown in Fig. 5.23.

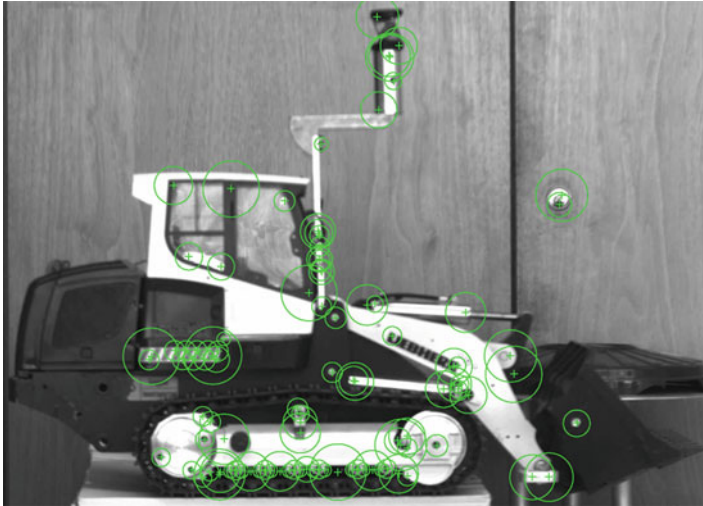


Fig. 5.21 Detected and extracted features in the reference image (green circles)

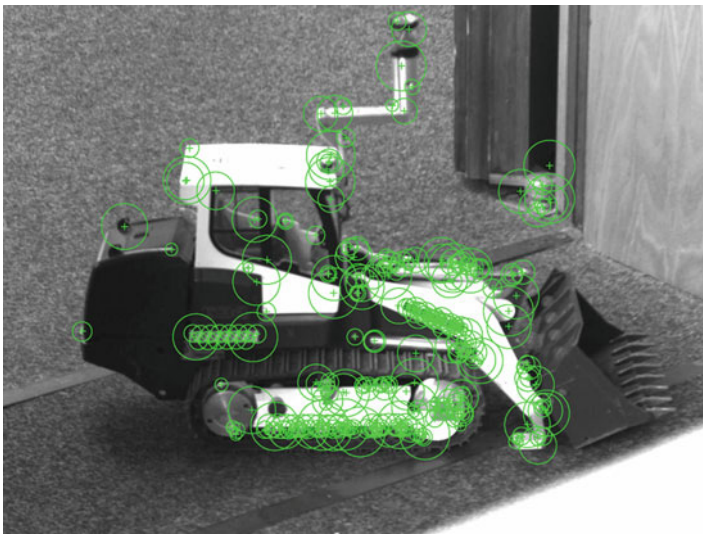


Fig. 5.22 Detected and extracted features in the test image (green circles)

It can be seen that many features are matched faulty. Therefore, the MSAC algorithm is applied to the first matching result. After the MSAC filtering, clearly fewer matches are left. These matches are unique and correct (Fig. 5.24).

Now the object recognition can be followed up by the use of the pixel-to-pixel transformation, described in Sect. 5.4.3. After this step, the object is uniquely identified in the test image (Fig. 5.25).

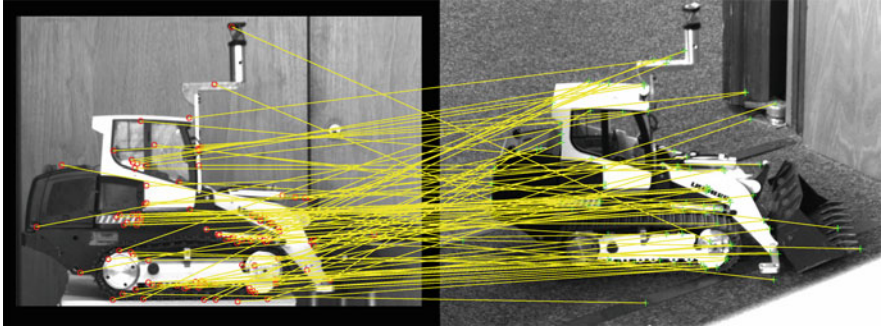


Fig. 5.23 Matching result before MSAC filtering (left: reference image; right: test image)

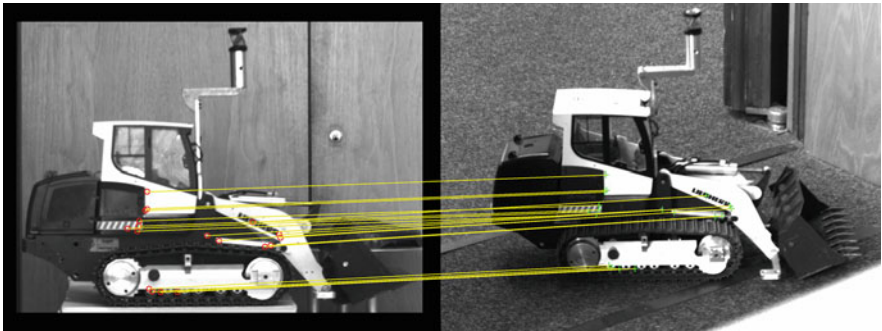


Fig. 5.24 Matching result after MSAC filtering (left: reference image; right: test image)

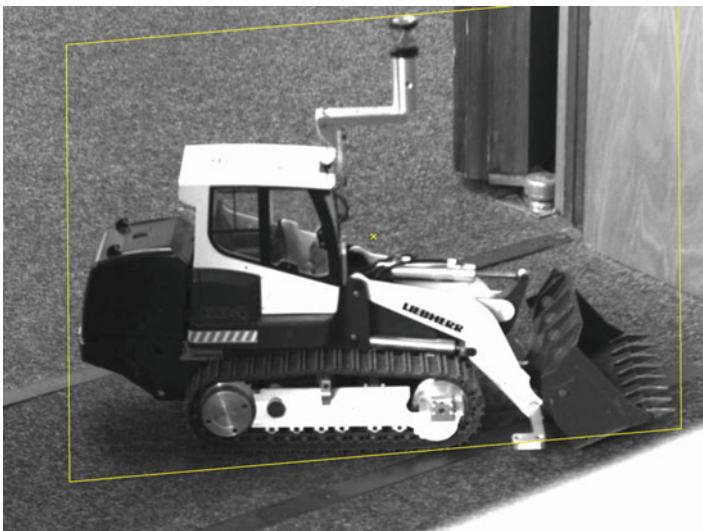


Fig. 5.25 Object identified in the test image

In the last step, telescope H_z and V angles are calculated from the pixel coordinates of the test image, according to Sect. 5.4.4. After the automatic aiming (setting of H_z and V angles by the servomotors), the reflector-less distance measurement is triggered, and the position is obtained by Eqs. (5.4)–(5.9).

5.5.2 Example of Kinematic Image-Based Object Tracking

Tracking is generally understood in engineering geodesy as the process of following moving objects with a certain sampling rate. The complete process is comprehensively explained in Sect. 5.3, and further on, emphasis on the same process will be extended with the use of images. In contrast to using a reflector to signalize the object, using images offers versatility and flexibility of choosing which object should be tracked just by acquiring an image of it. This means that the object does not necessarily need to be accessible. First similar attempts and principle descriptions can be found in Bayer et al. [36].

Specifically, in the case of IATS used for object tracking, images are processed to constantly identify and track the desired object. Further on, this section will provide an insight into the image-based tracking principles that uses a SURF algorithm to identify the object in each frame (image) and then track it.

Recently, a system comprised of a Leica TS16i IATS and the control software running under Matlab© was developed at the Institute of Engineering Geodesy. The TS16i is a high precision tachymeter that includes an overview camera with a 5 MPixel CMOS sensor. For object identification and tracking, a SURF algorithm is used. The camera has a $15.5^\circ \times 11.7^\circ$ field of view and is capable of capturing up to 30 frames per second. Four optical zoom levels are available in this case and can be used for tracking at different distances.

Processing of the frames (images) takes place on an external computer that constantly receives and sends data to the IATS. The physical connection is realized through a wireless network, and the developed program uses functions from the Image Processing Toolbox in Matlab. Examples of some of these functions are image read, detect SURF features, extract features, and match features.

Leica instruments can be controlled from an external source only with the use of special commands, sent as an ASCII message and defined by the GeoCOM Protocol [37]. Depending on the hardware integrated into the IATS, only some commands are available. In the present, the CAM and MOT commands are used for controlling the camera and servomotors of the IATS.

In the first phase, the user needs to select the object that is going to be tracked. This can be done either by directly capturing an image of it and then cropping the area with the object or from a previously taken image.



Fig. 5.26 Original image (left) and identified feature points (right)

Once the object is selected, unique features (points) are identified on the object and will serve as a basis for the tracking loop. Objects with a rich texture and varying geometry are the best suited in this case. This fact may be observed in Fig. 5.26, where the letters in the IIGS logo do not have the same amount of feature points as the building in the logo center.

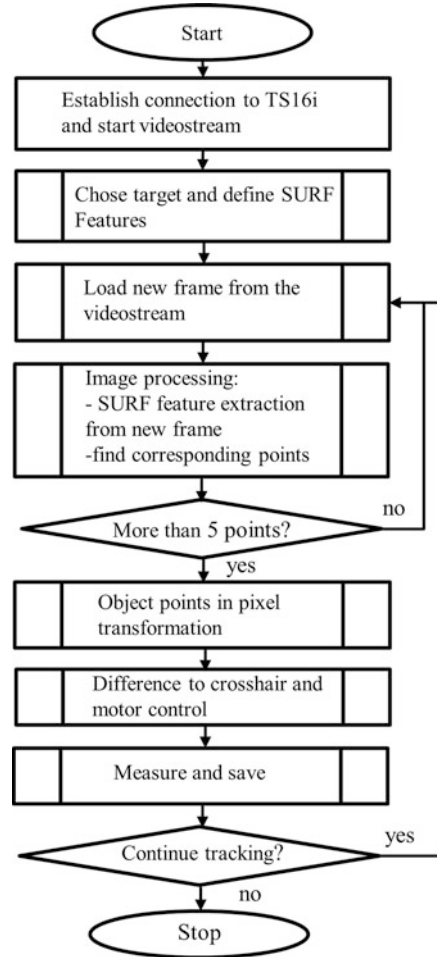
The object center is then determined, and from this point, the telescope is guided based on each processed frame. From the hardware point of view, this can happen at a rate of 30 fps or 30 Hz, but due to practical reasons for processing speed, a rate of 10 fps has been chosen. Additionally, a distance measurement can be made (without reflector) and the absolute coordinates of the object are obtained. This whole tracking process is currently limited to a 0.5 Hz update rate, mainly caused by transfer and processing speed. Finally, the process can be summarized as seen in Fig. 5.27.

5.6 Quality Control of Total Stations in Kinematic Mode Using a Laser Tracker

During total station measurements, the measurement quality and accuracy cannot be internally evaluated. Therefore, external measurements, by instruments with higher accuracy are required. One such instrument is the laser tracker. Different manufacturers offer laser tracker systems, often in combination with an additional accessory.

The laser tracker API Radian has been used for the following experiment examples. The distance measurement accuracy of the laser tracker is 250 times higher in kinematic mode and 500 times higher in static mode than that of the used robotic total station. The angle measurement accuracy is about 1.5 times better. The active target has the ability to permanently align with the tracker's laser beam and thus always keep the line of sight, independently of platform's orientation. The mechanical realization of the alignment is based on two servo actuators for setting the horizontal and vertical directions. Figure 5.28 depicts both instruments and their properties.

Fig. 5.27 Tracking process flowchart



Example 5.1 is conducted in combination with an active target, whereas Example 5.2 uses the regular laser tracker reflector SMR (Spherical Mounted Reflector) [38].

Detailed descriptions on the functionality of the active target are not published by the manufacturer. However, Kyle [40] presents an approach on the alignment functionality. The author describes an optical approach for the determination of the orientation of the active target for indoor scenarios. This method is based on the use of a pinhole reflector and a CCD array. Thereby a part of the incident laser ray passes the pinhole reflector and hits the CCD array. The x, y coordinate of the CCD, which was encountered by the ray, is depending on the direction of the emitted light source. Hereby the position of the reflector as well as the coordinate system of the laser tracker is known and can be determined or measured directly [40].




	Leica TCRP 1201 	API Radian™ Lasertracker 	Active Target 
Measurement Accuracies	Angle: $5 \frac{\mu\text{m}}{\text{m}}$ Distance static mode: 2mm + 2ppm Distance kinematic mode: 5mm + 2ppm	Angle: $3 \frac{\mu\text{m}}{\text{m}}$ Distance IFM (static only): 10 μm or 5 ppm Distance ADM: 10 ppm (Lock – On)	Centering Accuracy: $\pm 3 \mu\text{m}$
	Sampling Rate	$\leq 10 \text{ Hz}$	IFM: continuously ADM: no details
			Angular Velocity: $50 \frac{^\circ}{\text{s}}$
			Weight: 0,9 kg

Fig. 5.28 Overview of the measuring systems used for quality evaluation [38, 39]

Example 5.1

One application scenario is the quality control of a robotic total station in the field of guidance of unmanned ground or air vehicles (UGV, UAV). In such a scenario, an unmanned vehicle is controlled by the total station, which operates in kinematic mode within a closed-loop system. The total station serves as a sensor. In this specific experiment, reflector-based distance measurements have been performed. The total station provides the positions for the control algorithm. Within the control algorithm, these positions serve as controlled variables. The resulting guidance performance is dependent on the control quality, mainly influenced by the guidance algorithm and the measurement accuracy of the total station. The resulting combined accuracy is the quadratic sum of the two quantities. Hence, neither the quality of the guidance algorithm performance nor information about the measurement accuracy is known. Without an external, high accuracy measurement device, it is not possible to split up these two quantities from the combined measure. The laser tracker helps to overcome this drawback.

Figure 5.29 exemplarily depicts the measurement setup to solve the underlying problem.

The loop performs as follows: the tachymeter measures the position of the prism, mounted on the UGV, and sends it to the control computer. The computer calculates the perpendicular distance/lateral deviation between the UGV position and the reference trajectory. Based on this information, the algorithm calculates the best steering angle to get the UGV back on the reference trajectory as fast as possible. The UGV is equipped with an active laser tracker target and additionally with a 360° passive tachymeter reflector (Fig. 5.30).

The 360° reflector in combination with the total station provides the positions for the vehicle’s guidance system. The active target in combination with the laser tracker provides external, independent position measurements. The two reflectors are arranged one above the other in a vertical line. Thus, the recorded horizontal

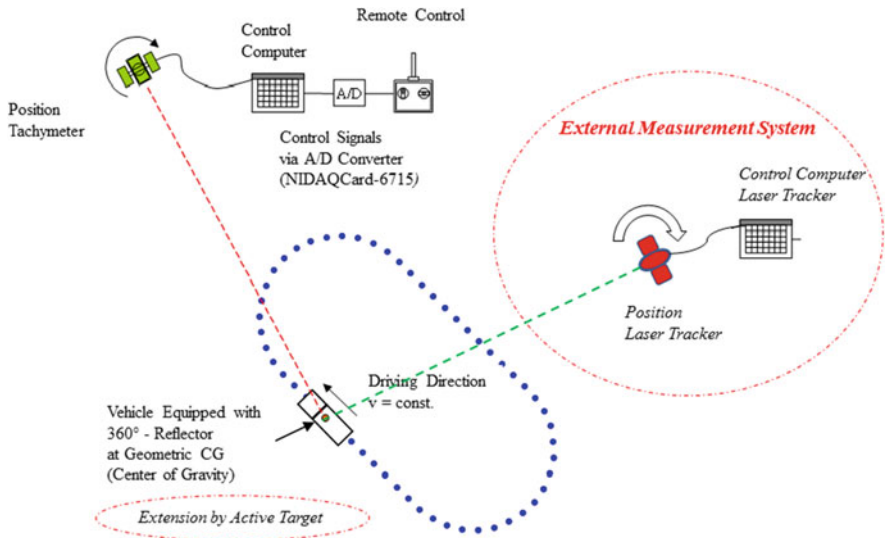


Fig. 5.29 Measurement setup for quality control [41]

Fig. 5.30 Active target and 360° passive reflector combination [41]



coordinates are referenced to one unique reference point. The two quality parameters, namely the control quality and the measurement accuracy, may be expressed as root mean square values (RMS).

To perform the separation between the measurement accuracy and the guidance quality, the following definitions have been made according to Beetz [42]:

- The measurement accuracy is the RMS between the tachymeter trajectory and the laser tracker trajectory.



Fig. 5.31 Moving trolley with laser tracker reflector (SMR) (left) and the target with feature points as identified by the IATS (right)

- The control quality is the RMS between the reference trajectory and the recorded laser tracker trajectory.
- The combined measure, the RMS between the reference trajectory, and the recorded tachymeter trajectory can be determined, too.

The results for the Example 5.1 reveal an RMS of 0.0028 mm for the tachymeter measurement accuracy and an RMS of 0.0031 mm for the control quality for the UGV [41].

Example 5.2

In order to test the system's performance, a target has been placed on a small trolley that is moving on a miniature railway and tracked during the movement. The reference is given by a laser tracker measurement of a reflector placed on the exact same axis of the trolley (Fig. 5.31). For both kinematic measurements, the object was manually shifted.

The results are further presented, and the differences between the coordinates obtained from laser tracker and IATS can be observed in Fig. 5.32. A systematic deviation is first noticeable in comparison to the tracker measurement. After shifting the tracker coordinates by 5 mm in the X direction, which is a manual correction, a plausible comparison may be conducted. Consequently, the tracker coordinates were fitted to a fourth-degree polynomial function, and the individual distances from the IATS coordinates to this regression line were computed. A mean value of 0.6 mm for lateral deviation was achieved.

Future improvements foresee the identification of this systematic effect and the usage of more efficient image processing tools in combination with a real-time industrial controller unit like the CompactRIO System from National Instruments that would help reduce latency time.

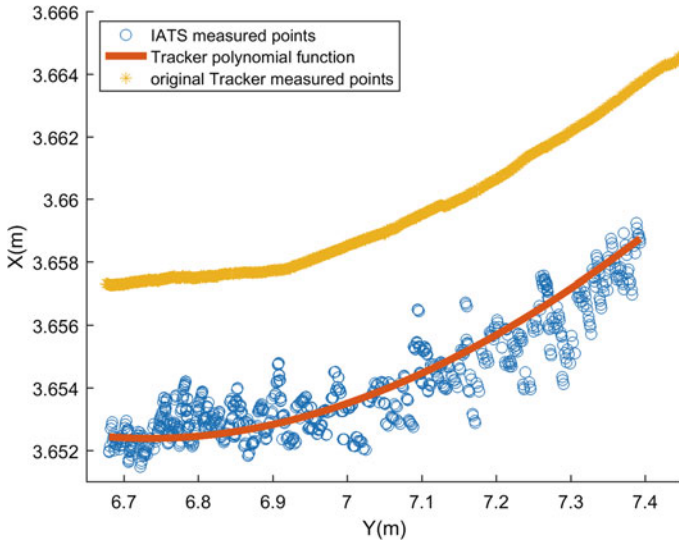


Fig. 5.32 Differences between IATS and laser tracker measurement of the same reference line

5.7 Conclusion

The contribution aimed at advertising the geodetic instrument tachymeter, sometimes also called total station, to the machine vision community. The authors focused on the automated version, the robotic total station that may track moving objects without intervention of the user. The limitation up to now is that the tracked object needs to be equipped with a mirror (reflector) that reflects a transmitted laser beam transmitted by the tachymeter. These principles are described in an overview before the new features like built-in cameras are highlighted. These cameras open the way to image processing and object recognition or, with other words, to machine vision. In this contribution, the advances in object detection and tracking using image processing techniques like the well-known SURF and the SIFT algorithm are presented. The algorithms have been implemented on high-end total stations available on the market. The results are encouraging, especially with respect to the measurement accuracy.

The expected accuracy will be less than 1 mm in tracking mode if the systematic effects of the evaluation procedure are eliminated. The measurement accuracy is determined using a laser tracker delivering accuracies around $10\ \mu\text{m}$ level for distance measurements. The tracking rate needs to be increased in the future to assure a continuous homogeneous real-time tracking. When the algorithms will run in real-time with a tracking rate of 10–20 Hz, any objects, even if they are not equipped with a reflector, may be tracked.

A remaining challenge is the synchronization of the total stations with other sensors within sensor fusion algorithms. The authors will further work on the open issues and see the results as a valuable input for the research within the DFG

Research Cluster of Excellence “Integrative Computational Design and Construction for Architecture” that will start in 2019 at the University of Stuttgart.

References

1. Rogers, L. (2004). *Leonard and Thomas Digges: 16th century mathematical practitioners, script to oral presentation*. Retrieved from November 11, 2018, from <http://numerisation.irem.univ-mrs.fr/>
2. McConnell, A. (2013). Jesse Ramsden: The craftsman who believed that big was beautiful. *The Antiquarian Astronomer*, (7), 41–53.
3. Mark, R.-P. (2009). Von Zeiss zu Trimble – 100 Jahre Entwicklung und Bau geodätischer Instrumente in Jena. *Allgemeine Vermessungsnachrichten*, 3, 83–88. Berlin.
4. Leitz, H. (1969). Zwei elektronische Tachymeter von Zeiss. *Allgemeine Vermessungsnachrichten*, 1/69, 73–79. Karlsruhe.
5. Matthias, H. (1982). Der Topomat, Vermessung, Photogrammetrie. *Kulturtechnik*, 80(4), 123–128.
6. Kahmen, H., & Reiterer, A. (2006). *Videotheodolite Measurement Systems—State of the Art in ISPRS Commission V Symposium ‘Image Engineering and Vision Metrology’*.
7. Wagner, A., Wasmeier, P., Wunderlich, T., & Ingesand, H. (2014). Vom selbstzielenden Theodolit zur Image Assisted Total Station. *Allgemeine Vermessungsnachrichten*, 121(5).
8. Möser, M., Hoffmeister, H., Müller, G., Staiger, R., Schlemmer, H., & Wanninger, L. (2012). *Handbuch Ingenieurgeodäsie*. Berlin: Grundlagen.
9. Luhmann, T., Robson, S., Kyle, S., & Harley, I. (2006). *Close range photogrammetry: Principles, methods and applications*. Caithness, Scotland: Whittles, Dunbeath.
10. Ehrhart, M., & Lienhart, W. (2017a). Accurate measurements with image-assisted total stations and their prerequisites. *Journal of Surveying Engineering*, 143(2), 2017.
11. Lachat, E., Landes, T., & Grussenmeyer, P. (2017). Investigation of a combined surveying and scanning device: The Trimble SX10 scanning total station. *Sensors*, 17(4).
12. Reiterer, A., & Wagner, A. (2012). System considerations of an image assisted total station—Evaluation and assessment. *Allgemeine Vermessungsnachrichten*, 3, 83–94. Berlin.
13. Scherer, M., & Lerma, J. L. (2009). From the conventional total station to the prospective image assisted photogrammetric scanning total station: Comprehensive review. *Journal of Surveying Engineering*, 135(4), 173.
14. Joeckel, R., Stober, M., & Hueb, W. (2008). *Elektronische Entfernungs- und Richtungsmessung und ihre Integration in aktuelle Positionierungsverfahren*, 5. Auflage. Heidelberg: Wichmann-Verlag.
15. Vogel, M. (2006). *Vom Pixel zur Richtung – Die räumlichen Beziehungen zwischen Abbildungsstrahlen und Tachymeterrichtungen*. Dissertation, Technischen Universität Darmstadt.
16. Leica Geosystem AG. (2018c). *The MS60 GeoCOM reference manual*. Retrieved November 11, 2018, from www.myworld.leica-geosystems.com
17. Rüeger, J. M. (1990). *Electronic distance measurement—An introduction*. Berlin/Heidelberg: Springer.
18. Ehrhart, M., & Lienhart, W. (2017b). Object tracking with robotic total stations: Current technologies and improvements based on image data. *Journal of Applied Geodesy*, 11(3). <https://doi.org/10.1515/jag-2016-0043>.
19. Deumlich, F., & Staiger, R. (2002). *Instrumentenkunde der Vermessungstechnik*. Heidelberg: Wichmann. ISBN 3-87907-305-8.
20. Zeiske, K. (1999). TPS 1100 Professional Series—Eine neue Tachymetergeneration von Leica. *VR 61/2*.

21. Gojcic, Z., Kalenjuk, S., & Lienhart, W. (2017). Synchronization routine for real-time synchronization of robotic total stations. In *INGEO 2017 7th International Conference on Engineering Surveying, Lisbon, Portugal* (pp.183–191).
22. Luhmann, T. (2010). *Nahbereichsphotogrammetrie. 3. Auflage, Wichmann*. Berlin und Offenbach: VDE Verlag GmbH.
23. Sonka, M., Vaclav, H., & Boyle, R. (1994). *Image, processing, analysis and machine vision*. London: Chapman & Hall Computing.
24. Girod, B. (2013). *Digital image processing: Edge detection*. Retrieved October 12, 2018, from https://web.stanford.edu/class/ee368/Handouts/Lectures/2016_Autumn/12-Edge-Detection_16x9.pdf
25. Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2006). Speeded-up robust features. In *9th European Conference on Computer Vision, 7–13 May 2006*.
26. Merziger, G., & Wirth, T. (2010). *Repetitorium Höhere Mathematik. 6. Auflage*. Barsinghausen: Binomi Verlag.
27. Shapiro, L. G., & Stockman, G. C. (2001). *Computer vision*. Upper Saddle River: Prentice Hall.
28. Lowe, G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision, Corfu 1999, Proceedings*.
29. Talukder, K. H., & Harada, K. (2007). Haar wavelet based approach for image compression and quality assessment of compressed image. *IAENG International Journal of Applied Mathematics*, 36(1). ISSN 1992-9978.
30. Muja, M., & Lowe, D. (2012). Fast matching of binary features. In *2012 Ninth Conference on Computer and Robot Vision—CRV'12, IEEE Computer Society, Washington D.C., USA. Proceedings*. ISBN 978-0-7695-4683-4.
31. Kybic, J., & Vnucko, I. (2010). *Approximate best bin first k-d tree all nearest neighbor search with incremental updates*. Research Reports of CMP, Czech Technical University in Prague, No. 10, 2010. Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University. ISSN 121 3-2365.
32. Torr, P., & Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding, Jahrgang 78*. ISSN: 1077-3142.
33. Niemeier, W. (2008). *Ausgleichsrechnung*. Berlin/New York: Walter De Gruyter Verlag.
34. Torge, W. (1980). Drei- und zweidimensionale Modellbildung. In H. Pelzer (Ed.), *Geodätische Netze in Landes- und Ingenieurvermessung*. Stuttgart: Konrad Wittwer.
35. Trimble Inc. (2018). *Datasheet Trimble S7 total station*. Retrieved November 11, 2018, from <https://geospatial.trimble.com/products-and-solutions/trimble-s7#product-support>
36. Bayer, G., Heck, U., & Mönicke, H.-J. (1989). Einsatz einer CCD-Kamera bei der Objektführung mittels Motortheodolit. *Allgemeine Vermessungsnachrichten*, 96(11–12), Karlsruhe.
37. Leica Geosystem AG. (2018b). *The GeoCOM reference manual*. Retrieved November 9, 2018, from www.myworld.leica-geosystems.com
38. Automated Precision Inc. (2018). *Laser tracker specifications*. Retrieved November 10, 2018, from <https://apisensor.com/download/radian-spec-sheet/>
39. Leica Geosystems. (2018). *Technische Daten TPS1200*. Retrieved November 10, 2018, from https://w3.leica-geosystems.com/downloads/123/zz/gps/general/brochures/GPS1200_brochure_de.pdf
40. Kyle, S. (2008). *Roll angle in 6DOF tracking. 2008 Coordinate Metrology Society Conference, Charlotte-Concord*.
41. Lерke, O., & Schwieger, V. (2015). Evaluierung der Regelgüte für tachymetrisch gesteuerte Fahrzeuge. *zfv—Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, Heft 4/2015 – 140. Jahrgang. <https://doi.org/10.12902/zfv-0078-2015>.
42. Beetz, A. (2012). *Ein modulares Simulationskonzept zur Evaluierung von Positionssensoren sowie Filter- und Regelalgorithmen am Beispiel des automatisierten Straßenbaus*. Dissertation, Universität Stuttgart.

Chapter 6

The Methods of Radar Detection of Landmarks by Mobile Autonomous Robots



Oleksandr Poliarus and Yevhen Poliakov

Abbreviations

GPS	Global Positioning System
MAR	Mobile autonomous robots
EMW	Electromagnetic waves
AP	Antenna pattern
RCS	Radar cross section

6.1 Introduction

The development of the theory and practice of modern mobile autonomous robots (MAR) involves providing the necessary accuracy of their navigation in unknown terrain of the earth or another planet. The robot's position on the earth surface is qualitatively determined by the GPS or other navigation system, but in some cases, the efficiency of the GPS can be reduced, for example, due to the limited visibility of the satellites. In such situations, it is convenient to use on-board sensors of various types to determine the coordinates of various objects of the environment [1] or important navigating landmarks. These landmarks are used to measure the angular coordinates of a robot and to solve the problem of its localization [2]. The logical approach is to place on-board sensors in different ranges of electromagnetic waves (EMW) (microwave, optical range, etc.). They scan the surrounding space and find obstacles that appear on the way of the robot. Not all objects outside the robot's path may

O. Poliarus (✉) · Y. Poliakov
Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

represent much interest for it, but if some of these objects are clearly distinguished above the ground and have the known coordinates, they are potential landmarks for the robot. If the robot reliably detects them and determines their coordinates with the desired accuracy, then the practical use of landmarks becomes real. As a rule, these landmarks are passive ones, that is, they only reflect the electromagnetic waves generated by the radar transmitter that is located at the robot. Near them, there are many secondary emitters (separate dimensional objects, trees, dense bushes, vegetation, the earth's surface irregularities, etc.) that create the background of reflected signals whose amplitudes may exceed the amplitude of the signal reflected from the landmark. The sounding of the surrounding space by a robot is often performed in the range of light wavelengths [3, 4] or in other bandwidths of the frequencies [5]. The chapter discusses only the radiofrequency range of waves in the interest of detecting landmarks and determining their coordinates. This wavelength range may be basic or additional one depending on the tasks facing the robot. The possibilities of simultaneous use of different wavelength ranges for solving the main objectives of the robot are also discussed in the chapter. In conditions where it is impossible to detect the echo signal from the landmark, a method for detecting the abrupt changes in signal amplitude during the scanning of landmarks of a special form is proposed.

6.2 The Navigation Problem of Mobile Autonomous Robots

Let the mobile autonomous robot navigate in unknown terrain in the absence of GPS signals on it. A radar or several small radars operating at different frequencies, which differ significantly, are mounted on a robot's board. The system can operate together with laser, ultrasound, and other measuring systems and can duplicate the capabilities of the technical vision system. Information from all measurers is processed by the system of intellectual data analysis, which makes the decision on the robot's position on the ground, as well as the type of terrain, the shape of the relief, which is also necessary for navigation. The chapter deals with a comprehensive approach to radio navigation problems and discusses some issues related to the integration of measuring information.

We believe that a robot scans the surrounding space for determining the position of the robot on the terrain, classification of the type of terrain, and the shape of the relief. All surrounding objects are divided into concentrated and distributed ones. The examples of concentrated objects are objects of human activity (cars, pillars, separate buildings), as well as natural ones (a separate tree, a hill in plain terrain). The distributed objects are almost all the continuous buildings in the city, forest, cross-country, etc.). Radio navigation of mobile robots, as a rule, can be done using concentrated objects with known coordinates. If the coordinates of a single object are determined in the process of measurement, then in some cases, this object can be considered as reference one. The distributed objects of a specific form that clearly stood out against a background of the environment can also be used to navigate

the robots. The radar method involves radiation of the objects by EMW of different frequency. Let us consider the top view on the edge of an even surface without vegetation, which is irradiated by a radar of a robot (Fig. 6.1).

Reflected electromagnetic waves are formed on even surfaces, the centers of which are points A, B, C, D, but in the direction toward the radar antenna, the reflected signal (echo signal) comes only from point B. It carries information about the distance from the radar to point B, but for the conditions of Fig. 6.1, this is the distance to the entire surface.

If the flat (in global sense) surface has a small-scale irregularity with mean height h or vegetation (grass, bushes), etc., the reflected signals reach the antenna radar from all points of this surface (Fig. 6.2).

The signal reflected at point B has the greatest amplitude. At the input of the receiver of the radar in the process of scanning, there will be a random process of amplitudes of echo-signals with an abrupt increase of amplitude at the moment of passing the direction of the main lobe of the radar antenna pattern (AP) through this point. Such a sharp increase of random amplitude at a certain point in time is called the “jump” of the amplitude. The analysis of such jumps is carried out further, where the possibility of their application for navigating robots is also estimated.

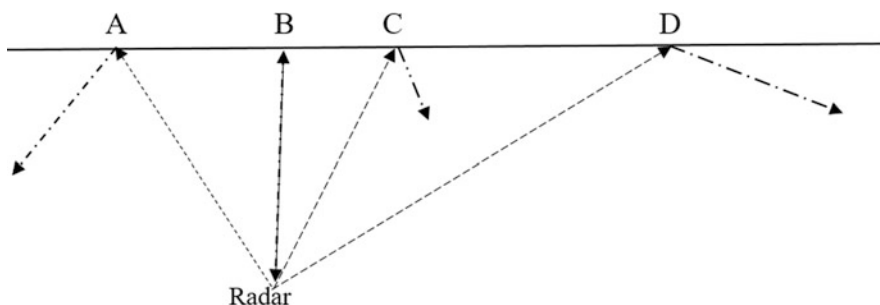


Fig. 6.1 The scheme of reflection signal formation during scanning an even surface

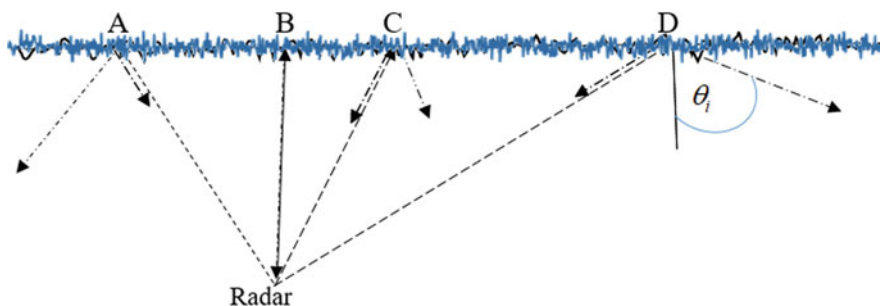


Fig. 6.2 The scheme of the echo signal formation from the even surface with small-scale irregularities

The scheme, shown in Fig. 6.2, is transformed into a scheme (Fig. 6.1) in cases where Rayleigh's criterion is fulfilled [6]

$$h < \frac{\lambda}{8 \cos \theta_i}, \quad (6.1)$$

where h is the largest height of small-scale surface roughness, λ is the wavelength of radiation, θ_i is the incidence angle of the wave on the surface at a certain point i (Fig. 6.2).

At point B, the incidence angle of the wave $\theta_i = 0$ and then $h < \frac{\lambda}{8}$. Hence, the EMW reflection at point B is a mirrored one if the maximum height of the surface irregularities does not exceed eighth of the wavelength. At $\lambda = 1$ m, this height reaches 12.5 cm, and from such an uneven surface, there is a mirror reflection. The domains around other points A, C, and D (usually this is the first Fresnel zone) form a diffuse reflection of the EMW in the direction toward the receiving radar antenna of the robot. For convenience and simplicity of the terminology, we will call the domain around point B, which reflects EMW in the direction of the radar antenna, a mirror point. If the Rayleigh criterion is not satisfied, the EMW reflection at point B is diffuse and the amplitude of the echo signal, as well as the probability of an amplitude jump, can be substantially reduced. Reflection of waves with small lengths from a flat surface, as a rule, is diffuse, since on a real surface, small-scale irregularities and vegetation exist almost always. Thus, in accordance with the Rayleigh criterion, it is necessary to use radiation with a relatively long wavelength $\lambda > 8h \cos \theta_i$ to obtain a mirror reflection of the EMW from a rough surface. For the conditions shown in Figs. 6.1 and 6.2, the mirror points from the domains A, C, and D cannot be obtained; however, if the earth or other surface is a curved one, then the appearance of echo signals from similar points becomes possible if $\theta_i = 0$, that is, the part of the curved surface is flat and perpendicular to the wave vector of the incident waves. Moreover, there may be several points B_1 , B_2 , and B_3 , in which there is a mirror EMW reflection in the direction toward the radar antenna, which is installed on the mobile robot (Fig. 6.3).

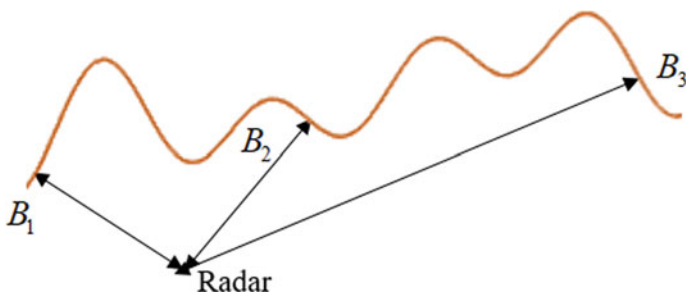


Fig. 6.3 The scheme of formation of echo signals from an uneven smooth surface

There are two main cases here. In the first case, the robot scans the surrounding space by antenna with a narrow antenna pattern (AP). This can be achieved using a small wavelength (centimeter or millimeter wavelength range). The irradiation of the points B_1 , B_2 , and B_3 is carried out sequentially, and the radar receiver takes the echo signal from each mirror point at different moments of time, and these points are distinguished by angular coordinates. In the second case, EMW with a large wavelength is used and the antenna has a wide AP.

EMW reflection from the points B_1 , B_2 , and B_3 becomes mirrored, but the echo signals are not distinguished by angular coordinates. In the antenna aperture, there is an interference electromagnetic field. By using this field, it is difficult to distinguish information about the reflected signal from each mirror point, the number of which may not be known. Determining the angular position in many cases needs a radar with a narrow AP antenna. Although the diffusive EMW scattering prevails, the mirror component of the signal's amplitude may be enough to distinguish a mirror point in the process of scanning against the background of diffuse reflection of the electromagnetic waves.

Previously, a heuristic analysis of the process of waves scattering from the earth's surface was carried out. There are various methods for calculating the characteristics of scattered waves, but it is impossible to use them in practice since it is difficult to describe the estimated situation because of changes in location conditions in the process of scanning and movement of the robot. However, during robot's navigation, it is expedient to get the maximum information about the surrounding area, especially when a priori information about the area is limited, for example, when a robot moves on another planet. The echo signals from the terrain contain information about the shape of the relief, the presence of vegetation, forests, and concentrated objects of artificial and natural origin. The characteristics of signals can be calculated in advance for the typical conditions of robot navigation using modern methods of analysis. Further, the intelligent systems that are installed at a robot will likely be able to determine the nature of the area on which it moves.

According to Rischka and Conrad [7], the landmark is a physical object created by man or nature, which is easily recognized by technical means. A landmark recognition by mobile robots involves the use of sequential comparison of the landmark video images with the reference images previously recognized [8]. Strict geometric methods of determining the landmark shape are not rational, since its real form can be blurred, for example, a pillar covered with trees, and the number of different landmarks can be large. In [7], it is reported that a database of 900 landmarks has been created. It is important here to determine the characteristic features of the landmark and to attribute it to any group, unless, of course, it is not unique. Consequently, there is a problem of constructing landmark models, as well as models of unknown terrain, which can be used to construct a map. The landmarks and unknown environmental models can be built not only based on video observations but also with radar observations, because in many cases it is difficult to obtain high-quality video images of landmarks and the surrounding terrain, for example, at night, fog, etc.

Currently, self-controlled robots have already been created. They can detect obstacles and automatically eliminate the possibility of collision with them. To do this, at a robot, ultrasound and other sensors are used together with the corresponding software [9]. The range of these sensors is small, and, therefore, they are compact and consume insignificant energy. Increasing the range of sensors is not required, which is due to the low speed of robots. For high-speed robots, for example, automatic cars, the requirements for the range of obstacles are increasing. A reliable means of implementing these requirements is the use of radars at robots.

A more powerful means of obtaining reliable and qualitative information about the surrounding space is the association of measurement information obtained by different types of sensors, for example, ultrasonic, radar, and mechanical. This is the problem of collecting measurers or mixing information from sensors built on various physical principles [10]. Localization of mobile robots on the ground most often uses triangulation methods [11], which also use radiation in different wavelengths, in particular laser, based on dynamic triangulation [12] and neural network for improving 3D laser scanner measurements [13]. Naturally, for the analysis of radar information, it is necessary to know the reflection properties of the surrounding area.

6.3 EMW Reflection from the Surrounding Area in Different Frequency Ranges

There are many scientific papers in which methods of calculating the parameters of scattered waves from the objects are analyzed. Let us consider them briefly and make the main conclusions from the results of theoretical and experimental studies, which may be useful for the robots' navigation.

The method of calculating the characteristics of scattered waves should be chosen based on the frequency range of the EMW, the polarization of the waves, the shape and state of the reflection surface, and some other factors that are not considered here. To calculate the characteristics of scattered waves on a surface with small-scale irregularities, the method of small perturbations is used, and if the radius of curvature of the surface is considerably greater than the wavelength, Kirchhoff's approximation is preferred [6]. To determine the required characteristics, it is necessary to have information about the coefficients of reflection of the EMW from the surface, which, in turn, depend on the complex dielectric permittivity $\hat{\epsilon} = \epsilon - j60\lambda\sigma$ of the surface, where ϵ is the real value of the dielectric permittivity of the soil, λ is the wavelength, and σ is the specific conductivity of the soil. For different types of soil, they are presented in Table 6.1 [14].

Table 6.1 presents the electrical characteristics of some homogeneous soils. In the presence of heterogeneous distributed objects on the ground, the electrical characteristics of a complex system are replaced by equivalent or effective values. If there is a vegetative cover on the black earth surface, then the effective value of the permittivity for the wavelength $\lambda = 3.2$ cm is $\epsilon_{ef} = 4-9.5$ in summer and $\epsilon_{ef} = 12$ in

Table 6.1 The values of components of the complex permittivity for different types of soils

No.	Kind of soil	The values of the components of dielectric permittivity	
		The real part of dielectric permittivity, ε	Specific conductivity of the soil σ , $\frac{\text{Cm}}{\text{m}}$
1	Snow	1.2	$2 \cdot 10^{-4}$
2	Dry soil	2.5–4	10^{-2} – 10^{-1}
3	Wet soil	4–20	10^{-2} –3
4	Crystalline rocks	5–10	10^{-6} – 10^{-4}
5	Water	60	10^{-3} –10
6	Seawater	80	4–6.6

winter. Similar characteristics at the same wavelength for meadows with shrubs are $\varepsilon_{\text{ef}} = 10$ –12 in summer and $\varepsilon_{\text{ef}} = 2.3$ –2.7 in winter, and for forest areas $\varepsilon_{\text{ef}} = 2.5$ –5 at wavelengths from 1.25 to 70 cm. The specific conductivity of soil with abundant vegetation is very difficult to simulate, as many factors are affected on it.

From the given data, the electrical characteristics of the soil are diverse, and therefore, it is difficult to create a single model. As a result, after applying Maxwell's equations, we usually pass to the estimation of the fields of scattered waves, using the transport equation, the Green's functions, and approximate solutions of integral equations for the surface current [14]. The electromagnetic field of waves reflected from an arbitrary surface is determined as the sum of the field reflected from some average smooth surface (not necessarily equal), and the perturbation fields caused by the scattering of waves by small-scale irregularities. The boundary conditions are transferred from the general surface to an averaging smooth surface, which is a complicated procedure. Thus, in determining the general scattered field, the method of small perturbations and the Kirchhoff method are used simultaneously.

The concepts of small-scale and large-scale irregularities are closely related to the wavelength, as can be seen from the Rayleigh criterion. The surface shape, which is determined by a certain complex function $h(x, y)$, can be represented as the sum of the products of orthogonal functions on random coefficients, for example, coefficients of the Fourier series. For each type of surface, the random coefficients are distributed according to some law. For practical purposes, a function that describes the shape of a surface is often represented as the sum of three functions [15]. The first function describes large-scale irregularities, the second one small-scale, and the third one the effective height of the structure, which is formed by elements of vegetation. For navigation of robots, the reflection of the waves from the surface toward the radar antenna requires information on the reflection coefficient of the EMW in this direction. The surface in this case is modeled in the form of a set of facets, each of which is covered by small-scale irregularities [15]. The resulting field is the sum of the coherent and incoherent representations of all facets. The main factor determining the wave field is the phase relationship between the partial reflected waves, especially when the number of these waves is small.

The specific effective EMW reflection surface (scattering surface) of statistically rough isotropic surfaces at angles of falling close to zero (the wave vector

perpendicular to the plane tangent to the surface at a given point) is determined by the expression [15]

$$\sigma_0 = K_{f_0}^2 \frac{l_h^2}{4\sigma_h^2} e^{-\frac{l_h^2}{4\sigma_h^2} \operatorname{tg}^2 \theta}, \quad (6.2)$$

where K_{f_0} is the complex coefficient of the mirror reflection of EMW with frequency f_0 , σ_h^2 is the variance of heights of small-scale irregularities, l_h is the radius of correlation of these irregularities, θ is the incidence angle of the wave to the surface (it is known that $\theta = 0$, if the waves reflect in the direction of the radar antenna).

The analysis of scattered waves by different structures of the surface has shown [15] that σ_0 depends on a complicated way on the frequency of the EMW. At large-scale irregularities of the surface, the specific reflection area is practically independent of frequency. Since the scale of irregularities substantially exceeds the wavelength, the mirror reflection of the waves dominates. It is the reason that with the increase of the angle of incidence of the wave on the surface, the part of mirror reflection toward radar and σ_0 sharply decreases rapidly. The presence of small-scale irregularities causes the change in the specific effective surface of the scattering, depending on the frequency of the laws from λ^0 to λ^{-4} . Thus, the radar of a robot can distinguish the presence of type of irregularities on the surface if it is equipped with transmitters and receivers operating at different frequencies. After calibration of the radar equipment, the reflected signals at different frequencies will be close in amplitude, if the waves reflect from mirror points, and on the surface, there are large-scale irregularities of the relief. During the robot movement, the nature of the terrain and the amplitude of the echo signals can vary, which is due to a significant dependence σ_0 on the presence of distributed objects on the surface. Table 6.2 describes some specific effective areas of wave reflection from the surface of the earth for different wavelengths [15].

Table 6.2 Specific effective areas of wave reflection from the earth's surface

No.	Type of terrain	Parameter values	
		Wavelength, λ , cm	Specific effective area, σ_0
1	Dense forest	3.2	0.1–0.8 (in summer) 0.6–0.7 (in winter)
2	Woodland	0.86	0.08
		1.25	0.02–0.05
		3.3	0.003–0.06
3	Forest	8	0.8
		70	0.6
4	Meadow with shrubs	3.2	3–7 (in summer)
5	Inhomogeneous terrain	0.32	0.4
		0.86	0.9
6	Desert	8	2.2
		70	0.5

Thus, on the basis of the frequency dependence of the amplitudes of echo signals from the surface type, the nature of the roughness on it can be determined (large-scale and small-scale). More information on irregularities based on this approach is difficult to obtain. If during the scanning the domains of the surface are heterogeneous, then the analysis of the echo signals can lead to erroneous conclusions about the change of the irregularity's nature, whereas really the type of the terrain has changed. For example, the specific area of scattering at a wavelength of 0.86 cm is an order of magnitude smaller for wooded area than for inhomogeneous terrain without forest [15]. There are some important features for the practice of reflecting millimeter waves from the surface, which are considered, for example, in [16]).

In the optical range, waves dissipate diffusely from objects which are important for robot's navigation. The energy of the reflected waves is directly proportional to the diffuse scattering coefficient ρ_d , that is, to the ratio of the reflected and falling light flux [17].

In most cases, the lower values ρ_d correspond to smaller wavelengths of the optical range, and the upper values to the largest lengths of this range. Reflection from coniferous trees does not have such dependence. So, in the range $\lambda = 0.6\text{--}0.7\ \mu\text{m}$ with increasing wavelength, the coefficient ρ_d first decreases, and then increases again.

The summarized information of this section can be used to construct models of random processes of echo signal amplitudes that arise during the scanning of surface by a radar antenna of a robot.

6.4 Mathematical Models of Random Processes Describing the Amplitudes of Echo Signals from the Distributed Objects

Random processes of amplitude of echo signals at the input of the radar receiver are formed during scanning the surrounding space by an antenna of the radar while a robot moves. Even for a static robot, a random process is created after the radar antenna irradiates a certain part of the surface of distributed and concentrated objects. During scanning, the type of the surface is changed randomly for the observer due to changes in the relief, vegetation, and so on. Correspondingly, the characteristics of random echo signals are changed at the input and output of the radar receiver.

The size of the irradiated part of the surface depends on the width of the radar AP and distance between the robot and surface (Fig. 6.4). Reflected signals come to the antenna from all irradiated areas of the surface. Their amplitude is determined by the width of the AP and the type of the surface with small-scale irregularities and vegetation. With uniformly distributed characteristics of irregularities and vegetation, there is a certain stationary random process (Fig. 6.5) of amplitudes of echo signals at the receiver input. The wider the AP, the more reflective elements of the irradiated surface are involved into the echo signal creation in a certain direction, which leads

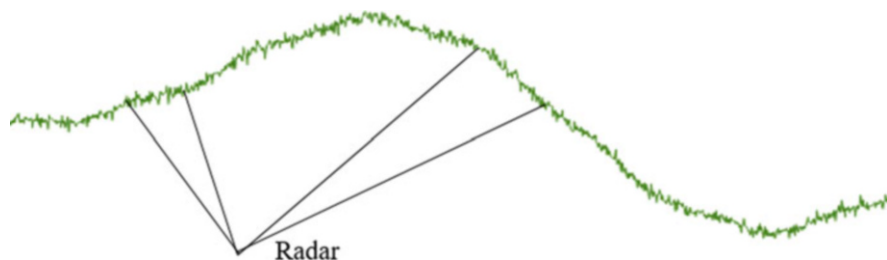


Fig. 6.4 A scheme of rough surface area irradiation by radar antenna with narrow antenna pattern

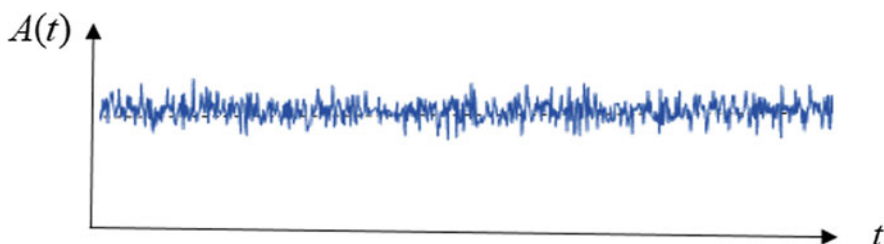


Fig. 6.5 An example of the realization of a stationary random process of amplitudes of echo signals that reflect from the rough surface during its scanning

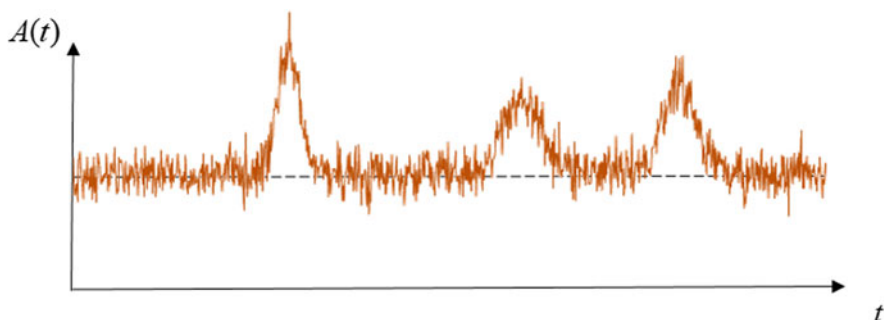


Fig. 6.6 An example of the realization of a random process of amplitudes of echo signals reflected from the rough surface during its scanning in the presence of mirror points on the surface

to the averaging of the amplitudes and a decrease of variance of the echo signal fluctuations. If the terrain changes radically, for example, there is a transition from the steppe to the wooded area, the variance of the echo signals can be substantially changed. This is an important feature for binding the robot to landmarks in navigation of the robots.

The process described above refers to the diffuse reflection of signals from the surface. In the presence of mirror points on the surface at certain moments of time due to the mirror component, the amplitude of the echo signal from the surface in a certain direction may increase sharply, that is, a jump of amplitude (Fig. 6.6) occurs.

This phenomenon can also be used for robot navigation. The amplitude of the jump is completely determined by the mirror component of the reflected signal, and the latter depends on the type of the surrounding area. The duration of the jump is determined by the scanning speed of space by the antenna. The shape of amplitude jump in simulation can be described by the Gaussian law. For reliable detection of a jump, it must have good energy characteristics, that is, amplitude and duration. The duration of the jump is determined by the nature of the transition from the completely diffuse reflection of the EMW from the surface to the mixed diffuse-mirror reflection and, naturally, the speed of space scanning. With a fast scan, the amplitude jump is difficult to detect against the background of diffuse reflected signals and interference. Slow scanning allows us to create conditions for detecting a amplitude jump.

Thus, in the scanning sector of an antenna with a narrow AP, the random process is represented by the sum of the stationary random process and random functions that at random moments describe the jumps of the amplitudes (Fig. 6.7).

If the length of the EMW is large (decimeter, meter waves), that is, the condition of mirror reflection (6.1) is executed, diffusely reflected echo signals will have a small amplitude in comparison with mirror signal amplitude, and in some cases,

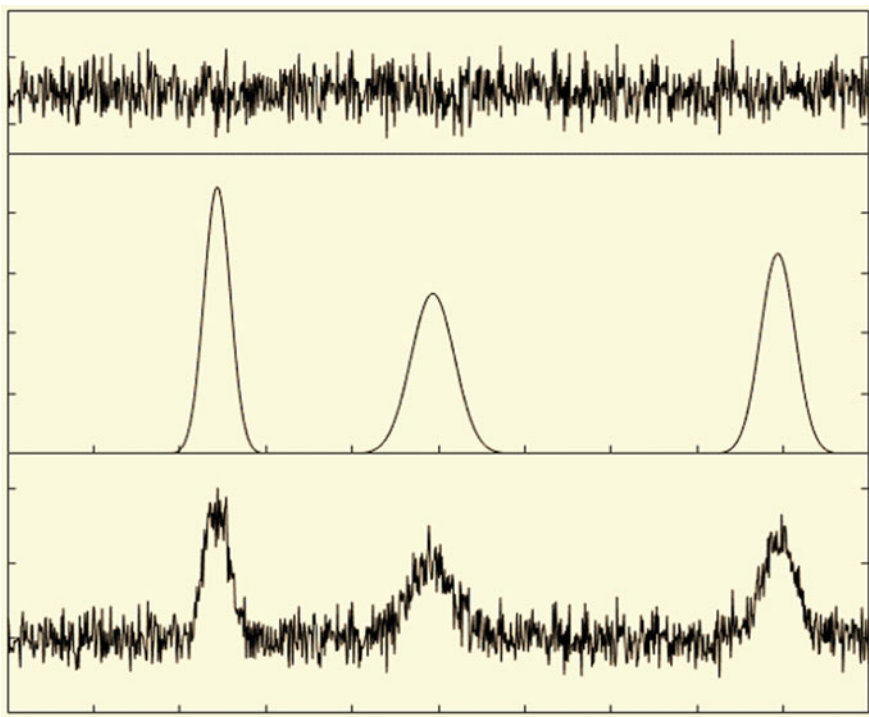


Fig. 6.7 A scheme of formation of a total random process of amplitudes of echo signals, reflected from a rough surface during its scanning, in the presence of three mirror points on the surface

they can be ignored. The radar AP at a robot of small size cannot be narrow, since its width is proportional to the wavelength and inversely proportional to the linear antenna size. Consequently, the antenna irradiates a large area of the surface. The echo signals from the surface are formed in the direction of the radar only from the mirror points. The number of these points is determined by the width of the AP and the type of large-scale irregularities on the surface, which, in general, determine its shape. For a flat surface, one should expect one mirror point. For a complex form surface, the number of mirror points can reach several units. If the mirror point is one, then diffuse signals are observed at the receiver input, the amplitude of which depends on the roughness, and on a large amplitude jump that can be easily found. In the presence of two mirror points, an interference pattern of two oscillations with different phases is formed, that is, the total oscillation may have amplitudes from zero to double amplitude from two mirror points. In this case, in the process of scanning within the width of AP, we can first get one mirror point, then two, then again one mirror point. Other variants of forming a total echo signal are possible. A similar picture is observed for cases of falling several mirror points into the width of the AP. A random process describing the echo signal from a rough surface in this case may be nonstationary. It can have a significant change in the mean value, but variance can behave in a complicated way: on the one hand, due to small diffuse scattering, it should be small, and on the other hand, due to random interference effects with deep fading, it can reach large values.

There is another option of EMW reflection from the surface when the number of mirror points is zero. This corresponds to the situation (Fig. 6.8) of scanning an oblique surface, when at the input of the receiver only diffuse reflected signals will be present. This is another feature of surface recognition by a radar of robot, but it is possible only with the use of large wavelengths in relation to the size of surface roughness.

Note that the characteristics of such waves are not distorted in the troposphere of the earth, that is, the influence of the medium on the propagation of radio waves can be neglected. Otherwise, it occurs with optical, millimeter, and even centimeter EMW, which are distorted in the troposphere due to the influence of precipitation,

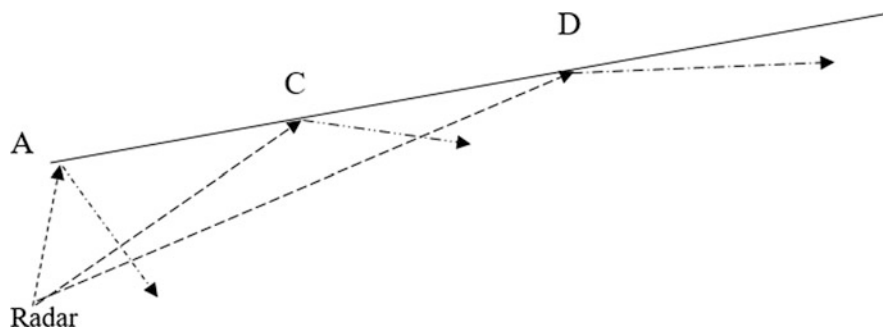


Fig. 6.8 Illustration of the absence of mirror points on an irradiated flat surface

fog, gases, etc. We will assume that the range of the location is relatively small, so that at distances such distortions of the characteristics of the echo signals do not reach large values, that is, they should not be considered.

Now let us consider the effective surface area of distributed and concentrated (point) objects, that is, backscatter echo area of these objects. The radar of a robot irradiates the surrounding area in angular range, which is determined by the angles θ , φ in two orthogonal planes. The transmitter power is P , transmitter antenna gain is $G = G_m F^2(\theta, \varphi)$, where G_m is the maximum antenna gain and $F^2(\theta, \varphi)$ is the normalized antenna pattern. The receiving antenna has the maximum value of the effective aperture area A_{ef_m} .

Using [18], one can obtain the formula for the radar cross section (RCS) σ_e of a diffusely scattering surface when it irradiates by EMW along the normal. At an angle θ from the normal, we have

$$\sigma_e(\theta) = 4S_e R_{h,v} \cos^2 \theta, \quad (6.3)$$

where irradiated area on the scattering surface on the range r within the width of the AP in two orthogonal planes $2\theta_{0.5P}$ and $2\varphi_{0.5P}$

$$S_e \approx r^2 \cdot 2\theta_{0.5P} \cdot 2\varphi_{0.5P} \quad (6.4)$$

The coefficients of EMW reflection from the surface $R_{h,v}$ are the effective coefficients, that is, they consider the presence of vegetation on the surface, etc. If the maximum value of the effective aperture area of the receiving antenna is A_{ef_m} , the signal power at the receiver output

$$P_r = \frac{P G_m A_{\text{ef}_m} F^4(\theta, \varphi) R_{h,v} 2\theta_{0.5P} 2\varphi_{0.5P} \cos \theta}{(4\pi r)^2}. \quad (6.5)$$

We consider that the domain of the surrounding area is simultaneously irradiated by two antennas at different frequencies with the same width of the antenna. It is easy to pick up by selecting the size of the antennas and the amplitude–phase distribution of the electromagnetic field in their aperture. Before use, the measuring channels at both frequencies are calibrated by changing the parameters P , G_m , A_{ef_m} during irradiation of an even conductive surface from the same range. The result of the calibration is the uniformity of the amplitudes of the reflected signals in both frequency channels from the direction of the main lobes of the APs.

Let us consider the following model situation. The robot scans the surrounding area at two different frequencies that are significantly different. Let, for example, the wavelength of horizontal polarization in the first frequency channel does not exceed 3 cm, and the second frequency channel is in the meter range. As a result of scanning a flat surface by EMW at two frequencies, the time dependences of the amplitude realization of the reflected signal at the input of a radar receiver are shown in Fig. 6.9.

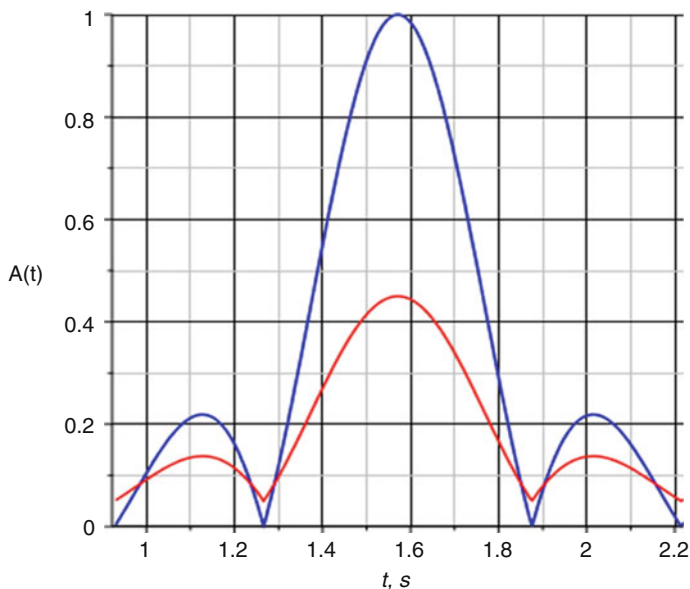


Fig. 6.9 The examples of the time dependences of the normalized amplitudes of signals, reflected from the even rough surface: the red line shows the amplitude of the signal in the first frequency channel; blue, in the second one

The time dependence of the amplitudes completely repeats the form of APs, and in the meter range (second channel), the field amplitude is greater than that in the first one, since the module of the coefficient of EMW reflection from the soil in the meter range is higher than that in the centimeter one. When the soil moisture increases, the difference between the red and the blue curves in the amplitude decreases. Because of the diffuse scattering of the centimeter EMW, the zero amplitude values in the first frequency channel (red line) disappear and the amplitude falls in the direction of the main maximum of the AP. Thus, the presence of small-scale roughness on a surface can be recognized qualitatively by the presented feature.

In the presence of mirror points on the surface, the above dependence $A(t)$ is destroyed (Fig. 6.10). In the meter range, the condition for mirror reflection of the EMW (Rayleigh criterion) is still preserved, and therefore, in some directions, there arise interference extrema of the field. In the centimeter band of the EMW, the dependence of the total field on time is complex and differs essentially from the previously deduced dependencies (Fig. 6.10).

It is easy to see that the reflection of an EMW from an uneven rough surface forms the complicated structure of electromagnetic field at both frequencies. This kind of dependence is exactly the evidence of a substantially uneven surface. Here, however, it should be noted that if the correlation radius of large-scale irregularities is significantly greater than the value $r \cdot 2\theta_{0.5P}$ or $r \cdot 2\varphi_{0.5P}$, for example, in Fig.

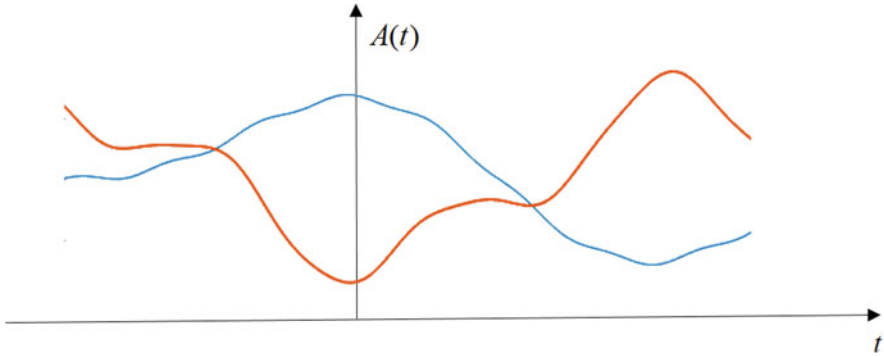


Fig. 6.10 The examples of the time dependences of the signal amplitudes reflected from the rough surface with three mirror points: the brown signal shows the amplitude of the signal in the first frequency channel; blue, in the second one

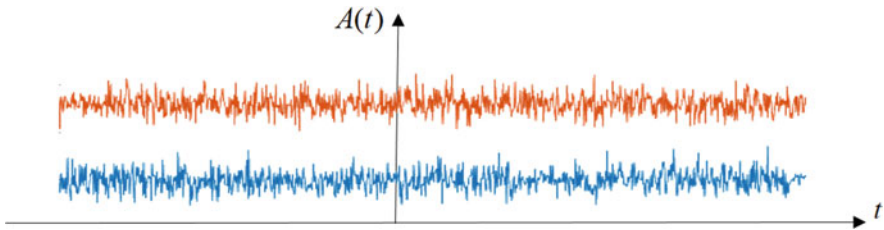


Fig. 6.11 The examples of the time dependences of the signal amplitudes reflected from the rough surface: the red signal shows the amplitude of the signal in the first frequency channel; blue, in the second frequency

6.4, this case is similar to the previous situations, since the number of mirror points tends to one.

In practice, the situation is often encountered, as shown in Fig. 6.8. In this case, the EMW of meter band almost does not reflect in the direction of the radar if the Rayleigh criterion is fulfilled. On the contrary, the centimeter EMW diffusely scatter in the direction of the radar, as shown in Fig. 6.11.

The analysis of the figures shows that the comparison of the amplitudes of the signals of two frequencies reflected from the surface allows to estimate approximately the type of terrain of the surrounding area. The obtained results form a priori information for the robots about the environment.

Until now, the time dependences of the amplitudes of signals reflected from the terrain during the process of its scanning were considered. Echo signals can also be obtained without scanning the environment. In this case, the antenna of a robot is stationary, and the maximum of the AP is directed perpendicular to the line of the robot's motion. As a result, it should be expected that dependencies (Fig. 6.9) would be converted into other ones similar to Fig. 6.5.

Consequently, the presence on the ground of concentrated objects, which can serve as landmarks for autonomous mobile robots, challenges the detection of such objects and their coordinates with respect to other landmarks. Immobility of most landmarks prevents the use of Doppler methods for separating echo signals from landmarks in the presence of signals that are reflected from stationary terrain. The only approach is to develop methods for detecting abrupt changes in the energy characteristics of echo signals from the terrain that arise during scanning the surrounding area or moving the robot along this area with fixed antenna. If the landmark is located on the background of a complicated relief of the terrain, then scanning the terrain will indeed lead to sharp changes in the amplitude of the echo signals (“jumps”) as under the influence of interference effects in the adding of signals, reflected from the mirror points and as a result of appearance of concentrated objects within the irradiated region that can be used as landmarks. Reliable detection of amplitude jump signals in this case is impossible. However, for the case presented in Fig. 6.10, the level of the reflected signals from the terrain in both frequency channels is significantly reduced. Thus, the most convenient situation in which the possibility of detecting ground landmarks is possible is the location in the direction of the sloping terrain, which has little forest and shrubs. This case can be considered as the main one for navigating the robots. Hence, it is necessary to estimate the radar cross section of concentrated objects and decide whether their choice can be used as landmarks.

Let us divide conditionally all practical situations into two groups. The first group includes all cases in which a significant part of the energy of EMW from the reflecting surface is directed to the antenna radar of a robot (e.g., Figs. 6.1 and 6.2). The second group covers cases of only diffuse reflection of EMW from the surface (e.g., Fig. 6.8). The simplest mathematical model of the complex amplitude of reflected monochromatic signals is

$$\dot{A}(t) = \sum_{k=1}^m \dot{A}_k(t) e^{j\varphi_k(t)} + n(t) \quad (6.6)$$

where $\dot{A}_k(t)$, $\varphi_k(t)$ are respectively random complex amplitudes and phases of the echo signals, reflected from the k th mirror point at the input of the radar receiver, m is the random number of such mirrored points, $n(t)$ is the white noise.

The model (6.3) describes the amplitude of the reflected signals in the first group. In the second group, there is a sum of many diffuse components with low energetic characteristics. For physical reasons, reliable detection of abrupt changes in the amplitude of echo signals is possible only in the second group. For a numerical estimation of this possibility, it is necessary to know the model of the echo signals from the landmark as a concentrated object.

6.5 Mathematical Models of Random Processes Describing Amplitudes of Echo Signals from Concentrated Objects

The different concentrated objects of natural and artificial origin with various scattering properties can be used as a landmark. Sometimes they are called the point objects, because their size is much smaller than the size of distributed objects.

The radar cross section (RCS) of concentrated objects is given, for example, in [19]. Therefore, RCS of a ball with perfectly conducting surface and radius $r \gg \lambda$ is

$$\sigma = \pi r^2 \quad (6.7)$$

The RCS of a circular cylindrical metal pillar with radius r and length L is determined by the formula [20]

$$\sigma_{\max} = \frac{2\pi r L^2}{\lambda} \quad (6.8)$$

RCS of a metal rectangular plate with dimensions a and b , which is much larger wavelength, is

$$\sigma_p = \frac{4\pi S^2}{\lambda^2} \quad (6.9)$$

where S is the area of the plate, the largest size of which is significantly smaller than the distance r between the robot and the plate. The formula (6.9) is given for the direction of the radar—the domain of the surface.

The analysis of formulas (6.7)–(6.9) shows that the RCS of the concentrated objects varies depending on the wavelength of the EMW. This may be the basis for the previous robot recognition of the landmark type. However, the main task is to recognize the situation that is peculiar for the first and second group of reflected signals. As previously indicated, this can be done on the basis of the use of essential differences in random processes describing echo signals at different frequencies that are significantly different. The robot's decision about existence of the second group in the given time range leads to the need for analysis of echo signals from useful objects, that is, landmarks. It turns out that the RCS of these objects in the centimeter range exceeds a similar index in the meter range, as can be seen from the formulas (6.4)–(6.9). For the signals of the first group, everything was the opposite. Consequently, this feature of RCS can be used to identify a situation that is characteristic of the first group.

The dependence of the RCS of the concentrated objects on the angular coordinates most often takes the form of a type $\frac{\sin \alpha}{\alpha}$, that is, it has a main lobe and several side lobes. To detect the amplitude jumps of the echo signals, only the main lobe has the actual value. Note that the AP of the radar also has a similar shape. Then the angle α can be represented as a product $\Omega \cdot t$, where Ω is an angular scanning speed of the

antenna. That is why the amplitude of the echo signal from the landmark should be approximated by the Gaussian dependence

$$A(t) = \frac{A_0}{\sigma_\alpha \sqrt{2\pi}} e^{-\frac{(\Omega \cdot t)^2}{2\sigma_\alpha^2}} \quad (6.10)$$

where A_0 is some amplitude of the reflected signal, chosen during the simulation or determined in the process of experimental research, and σ_α is a parameter characterizing the width of the Gaussian dependence. It should also be considered that $\alpha \leq \alpha_{\max}$, where α_{\max} is the maximum value of the angle within which the radiation and the reception of echo signals occur.

Hence, the maximum jump of the amplitude of the echo signal is proportional to the maximum RCS value σ_{\max} and is frequency dependent. In the first and second frequency channels of the receiver, the statistical characteristics of the echo signals from the environment should be described by white noise, the level of which in the centimeter range will most often be higher than in the meter range. The ratio of amplitude jumps of the echo signals from the cylindrical metal pillar in the first and second frequency channels are evaluated by the receiver measuring system and estimated by the formula

$$\frac{\Delta A_1}{\Delta A_2} \approx \frac{f_1}{f_2} \quad (6.11)$$

where f_1 and f_2 are the frequency of signals in the first and second frequency channels, respectively. If the landmark is a flat rectangular plate, the ratio of the amplitudes of these echo signals in both frequency channels is

$$\frac{\Delta A_1}{\Delta A_2} \approx \left(\frac{f_1}{f_2} \right)^2 \quad (6.12)$$

which follows from a formula like (6.11) [20].

The described method makes it possible not only to recognize the landmarks in some cases but also to find a connection of the MAR coordinates with these landmarks. If the robot's trajectory lies not far from the landmarks L_1 and L_2 (Fig. 6.12), then at the points S_1 and S_2 at different times the robot can detect these landmarks. Since the distance d between these points and the angles β_{ij} are known in advance, the distance to the landmarks at each point of the robot's trajectory is determined by the methods of triangulation. Here the index i shows the number of the current position on the trajectory of the robot, and the index j is the landmark number.

Assume that the error of determining the distance d between the current points of the robot's trajectory is small. Then, the precision of estimating the position of the robot relative to the landmark is determined by the errors of measuring the angle coordinates β_{ij} . As a result, the spatial errors of the determination of the working

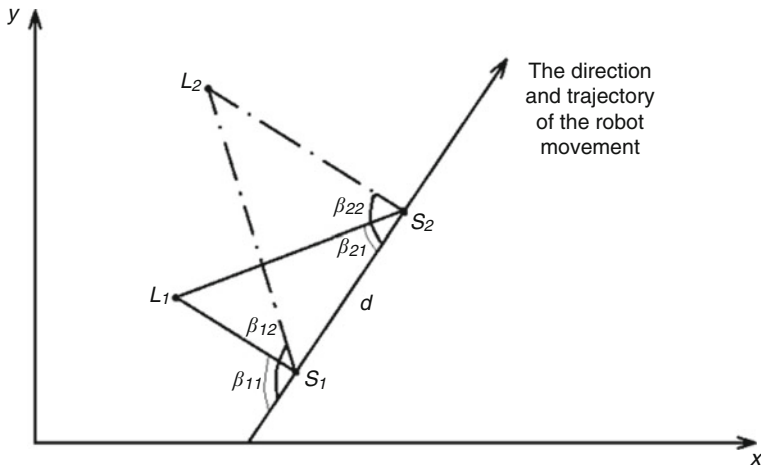


Fig. 6.12 Trajectory of the robot’s motion relative to two landmarks

robot’s position will be distributed within the ellipse with the axes $k\sigma_\xi$ and $k\sigma_\eta$ [21], where $k = \sqrt{-2 \ln(1 - p_0)}$ and p_0 is a probability of getting the robot’s position errors into the ellipse with the indicated axes. The coefficient $k = 2.15$ for $p_0 = 0.90$ and $k = 3$ for $p_0 = 0.99$. The dimension of the error ellipse for one landmark is estimated by the formulas [21]

$$\sigma_\xi = \left(\sum_{i=1}^2 \frac{\cos^2 \beta_i}{\rho_i^2 \sigma_i^2} \right)^{-0.5} \tag{6.13}$$

$$\sigma_\eta = \left(\sum_{i=1}^2 \frac{\sin^2 \beta_i}{\rho_i^2 \sigma_i^2} \right)^{-0.5} \tag{6.14}$$

where ρ_i is the distance from the robot’s antenna (point S1 or S2) to the landmark, σ_i^2 is the variance of the errors of angle coordinate determination by the robot antenna.

The spatial errors σ_ξ, σ_η of robot’s positions essentially depend on the distance ρ_i between the robot and the landmark and are reduced at short distances. If, at a distance $\rho_i \approx 300$ m, the landmark orientation is determined with errors not exceeding 10 m, then the measurement error of the angular coordinate is about 1°, which is not a problem for the radio engineering system of the robot.

The use of two frequency channels or channels with the processing of fundamentally different signals (microwave, ultrasound, laser, etc.) provides the mutual processing of measurement information that is hidden in the echo signals. The development of optimal systems for measuring the echo signal parameters and

the object coordinates useful for the robots is a very important problem, but usually the classical schemes perceive the signal amplitude jumps as interfering spikes and smooth them out.

6.6 Measurement of Amplitude Jump of Signals for Landmark Detection by Mobile Autonomous Robots

Since the signal reflected from the landmark often has no peculiarities compared to the background signal, its reliable detection by traditional methods is practically impossible. However, there are opportunities to detect some landmarks. During scanning the surrounding space by an antenna of the robot, a random process of the amplitude of echo signal from the area irradiated by the transmitter is observed at the input of the on-board receiver. The realizations of this process contain fluctuation components, the nature of which is due to the conditions of EMW reflection from the background elements of the terrain. The reflected signal from a landmark may be hidden in these fluctuations, unless there is a resonance scattering of the EMW from this landmark. In this case, during the terrain scanning, a amplitude jump may occur that exceeds the background reflection of the waves and the internal and external noise acting on the receiver input. The duration of the jump depends on the speed of the scanning. The jump of the echo signal amplitude can also occur if, in a short time, the nature of the area on which the robot moves essentially changes. The section deals with the method of detecting jumps of the echo signals, which allows, in some cases, to use this jump to identify a possible landmark by a mobile robot.

Detailed approaches to detecting abrupt changes in the dynamical system occurring in unknown moments of time are set out in [1] (abrupt changes at an unknown time point). In order to determine the abrupt (sharp) changes in [1], a unified approach is proposed within the general statistical theory of quality control, signal processing, automatic signal segmentation, and navigation monitoring systems. It is based on the use of the likelihood ratio algorithm and estimation of the statistical properties of the system. Before and after an abrupt change, two major models of the stochastic process are analyzed, and Kullbek's information is determined. To detect abrupt changes, a non-parametric Bayesian approach is used. The value of the signal parameter after the change is considered as known, and the probabilistic characteristics of this change are not evaluated.

In the general case, obtaining an estimation of the signal parameter is carried out by observing the maximum of the conditional a posteriori probability density of the parameter described, for example, by the Fokker–Planck–Kolmogorov equation. This stochastic equation in partial derivatives describes the evolution of the conditional a posteriori probability density for the Markov process. It is very complex and cannot be solved analytically [22]. A similar equation for high-frequency signals was obtained by [23]. Maltsev and Silaev [24, 25] developed optimal algorithms for evaluating the state of a dynamic system and identifying random jump-like

variations of its parameters and determining the moments of their occurrence. The systems of differential equations for a posteriori density of the probabilities of the parameters of random processes, the solution of which by the approximate methods leads to obtaining current estimates of parameters in a real time scale and the median estimation of the jump moment, are obtained. The analysis is carried out for the system described by the autoregressive process with a correlation coefficient jump at a random moment of time. In [26], an optimal system for detecting and estimating the sharp changes (jumps) of the amplitude of vibrations of machines in real time was developed. The main aspects of this approach to detecting landmarks for mobile robots are published in [27–29].

According to [1], the change in the signal parameter can be considered as abrupt change if it occurs almost instantaneously or less than the sampling period, which for MAR depends on the antenna rotational speed. The robot's antenna scans the terrain in a wide range of angles. The angular size of a landmark is small, and the time of its radiation by this antenna is also little. This indicates that the signal reflected from the landmark exists for a short time interval, that is, it generates a signal with a sharp amplitude change, which we call the jump. The form of the amplitude jump is like the shape of the antenna pattern, and, in the simulation, it will be represented by a Gaussian impulse reflected from a metal pillar, for example. The electromagnetic wave will be observed against the background of echo signals from the environment. Their amplitude in time is described by some random processes. For simulation, we will use reference random processes that characterize the reflection properties of a certain terrain and are described by known stochastic differential equations (SDRs), for example, of the following type:

$$\frac{dA(t)}{dt} = -\alpha \cdot t + n(t), \quad (6.15)$$

$$\frac{dA(t)}{dt} = a(A, t) + b(A, t) \cdot n(t). \quad (6.16)$$

In these equations, the parameter α characterizes the correlation properties and the spectrum width of the random process, $n(t)$ is a white noise, and the functions $a(A, t)$, $b(A, t)$ are used to form a nonstationary random process. All functions must be selected so that the random process is like the behavior of the reflected signals from the terrain.

The amplitude of the reflected signals from the terrain depends on many factors, for example, transmitter power and receiver sensitivity, antenna characteristics, range to reflecting elements of the terrain, effective surface of the scattering of objects in the area, etc. In order not to deal with a wide range of amplitudes of echo signals, we will normalize the obtained amplitude values to unity and accordingly simulate jumps of different intensity, duration, which occur at random moments of time at different levels of noise $n(t)$. The jumps should not significantly exceed the echo

signal background. The time scale should be consistent with the speed of scanning the surrounding area. A various number of jumps can be used to conduct the study.

The usual optimal system of signal amplitude measurement is not capable of qualitatively estimating rapid sudden changes of amplitudes (jumps of amplitudes). It is necessary to synthesize an optimal system that considers the peculiarities of the jumps of amplitudes in an unknown (in advance) instant of time. Such a system is based on the Fokker–Planck differential equation. Derivation of the system of stochastic differential equations for estimating amplitudes of jumps and their variances is presented in [27], and the final result is described by Eqs. (6.17)–(6.20).

$$\begin{aligned} \frac{dp_1}{dt} = & P_{\tau_{\text{jump}}}(t) \cdot e^{-z} \\ & + \frac{1}{N} \cdot p_1 \cdot (1 - p_1) \left\{ \begin{aligned} & A \cdot \Delta A_1 [1 - \cos(\varphi_0 - \varphi_1)] + \frac{1}{2} (A_1^2) - \\ & - \sigma_{\Delta A_1}^2 + 2 \cdot n(t) \cdot \Delta A_1 \cdot \sin(\omega \cdot t + \varphi_1) \end{aligned} \right\} \end{aligned} \quad (6.17)$$

$$\frac{dz}{dt} = \frac{p_1}{N} \left\{ \begin{aligned} & A \cdot \Delta A_1 [1 - \cos(\varphi_0 - \varphi_1)] + \\ & + \frac{1}{2} (\Delta A_1^2 A - \sigma_{\Delta A_1}^2) + 2 \cdot n(t) \cdot (A_1) \cdot \sin(\omega \cdot t + \varphi_1) \end{aligned} \right\} \quad (6.18)$$

$$\begin{aligned} \frac{d\Delta A_1}{dt} = & \frac{1}{p_1} \cdot P_{\tau_{\text{jump}}}(t) \cdot e^{-z} \cdot (\Delta A_0 - \Delta A_1) \\ & + V_1(t) \cdot \frac{1}{N} \cdot [2 \cdot y(t) \cdot \sin(\omega \cdot t + \varphi_1) - A \cdot \cos(\varphi_0 - \varphi_1) - \Delta A_1] \end{aligned} \quad (6.19)$$

$$\frac{dV_1}{dt} = \frac{1}{p_1} \cdot P_{\tau_{\text{jump}}}(t) \cdot e^{-z} \cdot [(\Delta A_0 - \Delta A_1)^2 + V_0 - V_1] - \frac{1}{N} \cdot V_1^2 \quad (6.20)$$

where

p_1 —a posteriori probability of detecting the signal amplitude jump

z —relative speed operation of the system

$\Delta A_0, \Delta A_1$ —a priori and a posteriori amplitude jump estimations

$V_1(t)$ —variance of a posteriori amplitude jump distribution

$V_0(t)$ —variance of a priori amplitude jump distribution

t —time

τ_{jump} —the moment of the signal amplitude jump from the value $A_0(\tau_{\text{jump}})$ to another value $A_1(\tau_{\text{jump}})$

$\varphi_0(t), \varphi_1(t)$ —the phase of the signal before and after the jump, respectively

$n(t)$ —white Gaussian noise with zero mean and spectral intensity N

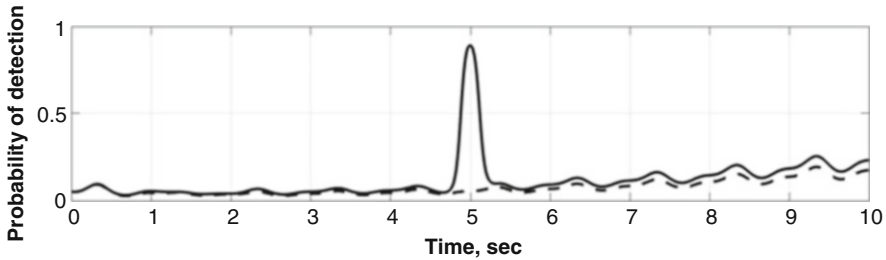


Fig. 6.13 The probability of detecting the amplitude jump for the random process when the amplitude jump is absent (dotted line) and if it exists (solid line)

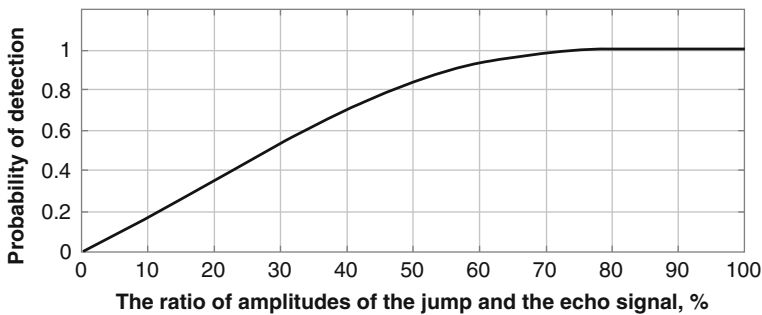


Fig. 6.14 The probability of detecting a jump of amplitude from the ratio of the amplitude of this jump to the mean amplitude of the echo signals from the terrain

The solution of the system of differential Eqs. (6.17)–(6.20) under the corresponding initial conditions allows to obtain the important characteristics of the landmarks detection system. The dependence of the probability of detecting the amplitude of the signal jump from time is shown in Fig. 6.13. If the jump is absent, then the probability is close to zero (Fig. 6.13, dotted line), and if it really exists, this probability for this example is closer to 1 (Fig. 6.13, solid line).

The probability of amplitude jump detection essentially depends on the energy characteristics of the jump, that is, its amplitude and duration. Figure 6.14 describes the dependence of this probability on the amplitude jump of constant duration. For the conditions given earlier, the probability 0.8 is achieved even when the amplitude jump exceeds half the mean amplitude of the signal that is reflected from the terrain (Fig. 6.14).

The estimation of the amplitude jump obtained as a result of the solution of the system of Eqs. (6.17)–(6.20) is presented in Fig. 6.15. The system is not able to determine the jump shape, but the value of this jump determines well.

The accuracy of estimating the jump of the amplitude is determined from the variance equation (fourth Eq. 6.20) of the system (6.17)–(6.20). The time dependence of the variance is shown in Fig. 6.16.

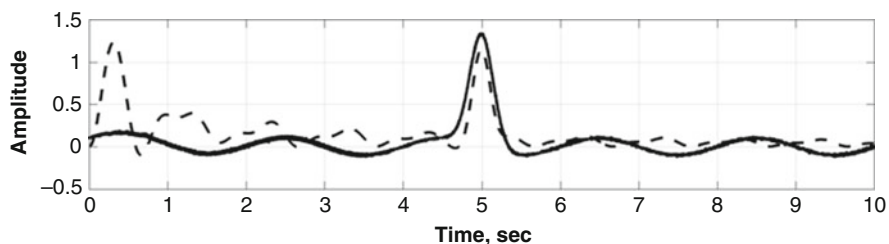


Fig. 6.15 The realization of random process with amplitude jump (solid line) and result of its estimate (dotted line)

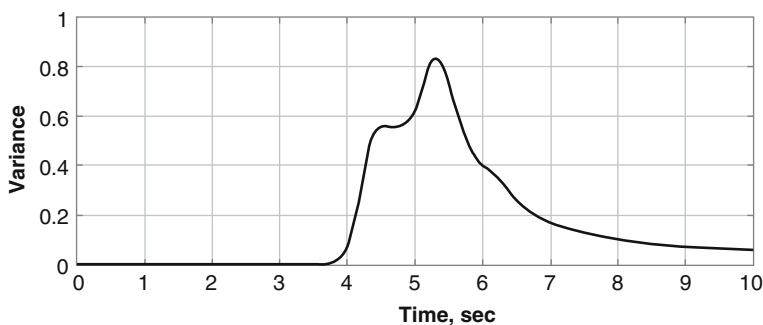


Fig. 6.16 An example of the time dependence of the amplitude jump variance

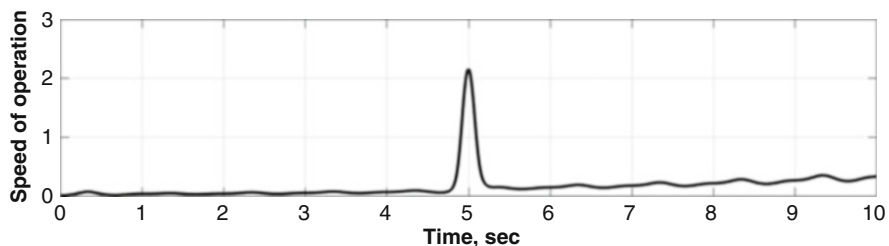


Fig. 6.17 An example of time dependence of operation system rate

Only after a while the variance of the amplitude jump decreases to small values. This time depends on the operating speed of the system, which is determined from the second equation of the system (6.17)–(6.20) and is shown in Fig. 6.17.

So, determining the amplitude jump takes some time which is not critical for relatively slow MAR. The results of the simulation are obtained for the case of the absence of a priori information about the time of an amplitude jump occurrence, which means that the given numerical results characterize the worst properties of the system of detection and evaluation of amplitude jumps of echo signals from landmarks. The presence of any information about jumps increases the quality of their detection and extends the scope of the method. For example, the navigation

systems described in [1, 4] can provide a priori information about the robot's position, which increases the probability of detecting landmarks and increases the accuracy of the estimation of the amplitudes of jumps and the accuracy of the coordinate measurement of the robot.

It should be emphasized that the detection of amplitude jumps of echo signals from landmarks is based on the evaluation of energy characteristics, regardless of the physical nature of such jumps. That is why it is possible to detect noise surges that are similar to jumps in signal amplitudes and, as a result, to reduce the probability of correct identification of landmarks. Hence, the requirements for the reliabilities of detected jump parameters, which differ significantly from the parameters of noise surges, increase.

The developed approach to detecting jumps of the echo signal amplitude is based on the use of the Fokker–Planck–Kolmogorov equations that have a wide application. For example, we have developed a similar method for detecting sudden changes in economic processes [27].

References

1. Colle, E., & Galerne, S. (2017). A multihypothesis set approach for mobile robot localization using heterogeneous measurements provided by the internet of things. *Robotics and Autonomous Systems*, 96, 102–113. Elsevier.
2. Garulli, A., & Vicino, A. (2001). Set membership localization of mobile robots via angle measurements. *IEEE Transactions on Robotics and Automation*, 17(4), 450–463.
3. Lindner, L., Sergiyenko, O., Rivas-Lopez, V., Hernandez-Babluena, D., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., Murrieta-Rico, F. N., Ivanov, M., Tyrsa, V., & Basaca, L. C. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot*, 44(4), 532–541.
4. Sergiyenko, O. Y. (2010). Optoelectronic system for mobile robot navigation. *Optoelectronics, Instrumentation and Data Processing*, 46(5), 414–428.
5. Prorok, A., Gonon, L., & Martinoli, A. (2012). Online model estimation of ultra-wideband TDOA measurements for mobile robot localization. In *IEEE International Conference on Robotics and Automation (ICRA)* (8 p). Saint Paul, USA.
6. Ishimaru, A. (1978). *Wave propagation and scattering in random media. Vol. 2: Multiple scattering, turbulence, rough surfaces and remote sensing* (317 p). New York: Academic.
7. Rischka, M., & Conrad, S. (2014). Landmark recognition: State-of-the-art methods in a large-scale scenario. In *Proceedings of the 16th LWA Workshops: KDML, IR and FGWM* (pp. 10–17). Aachen, Germany.
8. Schmid, C., & Mohr, R. (1996). Combining greyvalue invariants with local constraints for object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (pp. 872–877). San Francisco, CA, USA.
9. Hanumante, V., Roy, S., & Maity, S. (2013). Low cost obstacle avoidance robot. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(4), 52–55.
10. Kandylakis, Z., Karantzalos, K., Doulamis, A., & Karagiannidis L. (2017). Multimodal data fusion for effective surveillance of critical infrastructure. In *Frontiers in spectral imaging and 3D technologies for geospatial solutions*, 25–27 October 2017 (pp. 87–93). Jyväskylä, Finland.
11. Borenstein, J., Everett, H. R., Feng, L., & Wehe, D. (1997). Mobile robot positioning- sensors and techniques. *Journal of Robotic Systems*, 14(4), 231–249.

12. Real-Moreno, O., Rodriguez-Quiñonez, J. C., Sergiyenko, O., Basaca-Preciado, L. C., Hernandez-Balbuena, D., Rivas-Lopez, M., & Flores-Fuentes, W. (2017). Accuracy improvement in 3D laser scanner based on dynamic triangulation for autonomous navigation system. In *Industrial Electronics (ISIE). 2017 IEEE 26th International Symposium on IEEE* (pp. 1602–1608).
13. Rodriguez-Quiñonez, J. C., Sergiyenko, O., Basaca-Preciado, L. C., Hernandez-Balbuena, D., Rivas-Lopez, M., Flores-Fuentes, W., & Basaca-Preciado, L. C. (2014). Improve 3D laser scanner measurements accuracy using a FFBP neural network with Widrow-Hoff weight/bias learning function. *Opto-Electronics Review*, 22(4), 224–235.
14. Krasiuk, N. P., Koblov, V. L., & Krasiuk, V. N. (1988). Influence of the troposphere and underlying surface on radar. In *Radio and communication* (216 p). (in Russian).
15. Zubkovich, S. G. (1968). Statistical characteristics of radio signals reflected from the earth's surface. In *Sov radio* (224 p). (in Russian).
16. Kulemin, G. P., & Razskazovsky, V. B. (1987). The scattering of millimeter radio waves by the earth at low angles. (*Scientific thought*) (232 p). (in Russian).
17. Lukianov, D. P., et al. (1981). Laser measuring system. In *Radio and communication* (456 p). (in Russian).
18. Skolnik, M. I. (1990). *Radar handbook* (846 p). New York: McGraw-Hill.
19. Grishin, J. P., Ignatov, V. D., Kazarinov, J. M., & Ulianitskiy, J. A. (1990). Radio engineering systems. In *High school* (496 p). (in Russian).
20. Rajyalakshmi, P., & Raju, G. S. N. (2011). Characteristics of radar cross section with different objects. *International Journal of Electronics and Communication Engineering*, 4(2), 205–216.
21. Shirman, J. D. (1970). Theoretical foundation of radar. *Sov radio* (560 p). (in Russian).
22. Sharma, S. N. (2008). A Kolmogorov-Fokker-Planck approach for a stochastic Duffing-van der pol system. *Differential Equations and Dynamical Systems*, 16, 351–377.
23. Stratonovich, R. L. (1968). *Conditional Markov process and their application to the theory of optimal control* (367 p). Amsterdam: Elsevier.
24. Maltsev, A. A., & Silaev, A. V. (1985). Detection of jump-shaped parameter changes and optimal estimation of the state of discrete dynamic systems. *Automation and Telemekhanic*, 45–58. in Russian.
25. Maltsev, A. A., & Silaev, A. V. (1989). Optimal estimation of moments of random jump changes of signal parameters. *Radio Engineering and Electronics*, 34(5), 1023–1033. in Russian.
26. Poliarus, O. V., Barchan, V. V., Poliakov, Y. O., & Koval, A. O. (2009). The optimal system for detecting and estimating the jumps of amplitudes of dynamic objects vibrations. *East European Journal of Advanced Technology*, 6/6(42), 21–23. (in Ukrainian).
27. Poliarus, O. V., Poliakov, Y. O., Nazarenko, I. L., Borovyk, Y. T., & Kondratiuk, M. V. (2018). Detection of jumps parameters in economic processes (on the example of modelling profitability). *International Journal of Engineering & Technology*, 7(4.3), 488–496.
28. Poliarus, O. V., Poliakov, Y. O., & Lindner, L. (2018). Determination of landmarks by mobile robot's vision system based on detecting abrupt changes of echo signals. In *Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society* (pp. 3165–3170). Washington, DC, USA.
29. Poliarus, O., Poliakov, Y., Sergiyenko, O., Tyrsa, V., Hernandez, W., & Nechitailo, Y. (2019). Azimuth estimation of landmarks by mobile autonomous robots using one scanning antenna. In *Proceedings of IEEE 28th International Symposium on Industrial Electronics* (pp. 1682–1687). Vancouver, BC, Canada.

Chapter 7

Machine Vision System for Orchard Management



Duke M. Bulanon, Tyler Hestand, Connor Nogales, Brice Allen,
and Jason Colwell

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
Cov	Covariance
GPS	Global Positioning System
IR	Infrared
K_I	Integral gain
K_P	Proportional gain
LIDAR	Light detection and ranging
NIR	Near infrared
PI	Proportional-plus-integral
RGB	Red, Green, and Blue
RMS	Root mean square
UAV	Unmanned arial vehicle
UGV	Unmanned ground vehicle
Var	Variance

D. M. Bulanon (✉) · T. Hestand · C. Nogales · B. Allen
Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA
e-mail: dbulanon@nnu.edu; theastand@nnu.edu; cnogales@nnu.edu; beallen@nnu.edu

J. Colwell
Department of Mathematics and Computer Science, Northwest Nazarene University, Nampa, ID,
USA
e-mail: jcolwell@nnu.edu

7.1 Introduction

Machine vision is a technology that provides a visual sensor to machine systems. There are several industries where machine vision can be applied, such as agriculture [1], automotive [2], and industrial [3], each with its own set of applications. The first industrial applications focused on quality inspection and manipulator control while agriculture used machine vision for tractor navigation, product inspection, and fruit harvesting. This chapter focuses on the development of machine vision for orchard management applications and is ordered as follows:

1. Definition of machine vision

The four main elements of a machine vision system are:

- **Scene Constraints**—the physical constraints of the environment in which the machine vision system operates. There are several factors to consider when evaluating the scene constraints which include lighting and the color of the work plane as well as other factors [4].
- **Image Acquisition**—the properties and characteristics of the camera being used. There include color cameras, stereo cameras, NIR cameras, IR cameras, and others. Each of these camera types has different characteristics, so the decision of which camera employed depends on the application.
- **Image Processing and Analysis**—the process of modifying the acquired image to extract the desired information. There are several sub-steps in the image processing and analysis element, which include preprocessing the image, segmenting the region to useful regions, extracting useful features and classifying those features.
- **Actuation**—the physical action the system will take in response to an identified object. In agricultural applications, examples include picking fruit off the tree, sorting already picked fruit by their grade and weed control [5, 6].

2. Machine vision applications in different areas of agriculture

As mentioned in the abstract, there are several different fields for which machine vision can be applied and different applications within each field. Specifically, within the field of agriculture, three primary applications exist. They are:

- **Plant Identification**—analyzes the color, size, and shape of the object within the image to classify the plant type.
- **Process Control**—common application for machine vision and in agriculture tends to focus on evaluating fruit. This evaluation uses the size, shape, and color of the fruit to determine quality for grading and sorting.
- **Machine Guidance and Control**—the most common thought of process for an application of machine vision. Though many forms exist, a common example is a ground vehicle, which could be either manned or unmanned. In an agricultural application, this vehicle would run through a field or an orchard employing

several different inputs and sensors such as GPS, ultrasonic sensors, and a vision system [5]. The vision system could determine what object is in front of the ground vehicle and helps determine what action the ground vehicle should take.

3. Orchard management machine vision system

This example of a machine vision system in an orchard application demonstrates different analysis techniques of images in order to extract useful information. There are several different steps involved when analyzing images for a machine vision system, and these are dependent on the task at hand. The goal, for the example discussed in this chapter, is a machine vision system that can predict a fruit yield of apple and peach trees when the tree is in full blossom. It was hypothesized that the crop yield could be estimated by determining the number of blossoms on the tree, so a machine vision system needed to be created which could count the number of blossoms on a tree. Because every blossom on a fruit tree was essentially the same color, the RGB data from an acquired image could be used to filter out of the scene everything but the blossoms. The remaining information, the blossoms, are then counted and processed to determine the overall yield of the particular tree. Again, depending on the goal of the project, a different application of machine vision may need to be applied, but the method reviewed in this chapter provides various applications which can be extended outside of agriculture.

4. Stereo imaging to identify tree structure and improve individual tree detection

One of the main problems with the machine vision system used to estimate the yield of apple and peach trees is the inclusion of blossoms from multiple trees as each blossom in the acquired image would be counted even if it was on a tree behind the tree of interest. This is because the vision system developed used only the image RGB data to remove the scene. An additional filter, one which focuses on distance, needs to be added. Stereo imaging can be used to address this problem because now the acquired image has both the RGB parameter and the distance parameter. With this distance parameter, trees which are farther away from the tree of interest can be eliminated, so blossoms only from the tree of interest can be counted, thereby increasing the accuracy of the yield estimation.

5. Machine vision system that navigates a robot within orchard rows

Another application for a machine vision system is implementation on a ground vehicle which can navigate through the rows of an orchard. When applying machine vision for this configuration, the scene constraints become extremely important. To successfully navigate the rows of an orchard, the system must account for the symmetry of the rows, the size of the trees, and row separation. The ground vehicle is sized to fit and operate within these constraints. For example, a small ground vehicle can use the sky visible between rows of trees for navigation if the trees are relatively large and the rows are adequately spaced.

7.2 The Machine Vision System

Awcock and Thomas [7] defined a general machine vision system that is shown in Fig. 7.1. The defined system consists of four elements that could be found in a typical machine vision system in any field of application. The four elements are scene constraints, image acquisition, image processing and analysis, and actuation.

7.2.1 Scene Constraints

The scene constraint refers to the environment wherein the machine vision equipment is to be placed, and it is where the information is to be taken. The main aim of this system is to extract from the environment the desired information by the proper controlling of factors that affect the acquisition of data like lighting and the proper installation of the machine vision equipment. Some of the environment may be controlled such as in the sorting lines for product inspection [8] while other environmental parameters such as lighting conditions, fruit location on a tree, and the unstructured nature of the branches are difficult to control in an apple orchard [9].

7.2.2 Image Acquisition

Image acquisition is the element that converts light falling onto the photosensors of a camera into a digitized data, typically a 512×512 pixel image, which is then able to

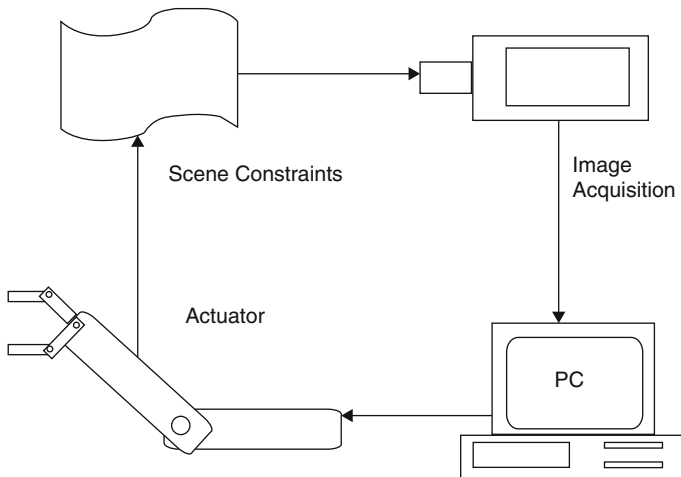


Fig. 7.1 A generic machine vision system

be processed. The camera may be a black and white video camera that is dependent on light intensity, a colored video camera that is dependent on the visible spectrum, or an infrared camera with the selection being based on the relevant information needed. With the advancement of sensor technologies, cameras sensitive beyond the visible spectrum are also available. Hyperspectral and multispectral imaging techniques have emerged as important tools for the safety and quality inspection of food and agricultural products [1].

7.2.3 Image Processing

Image processing deals with the acquired digital image as input and outputs an image that has been enhanced so that the desired information can be extracted. Several steps are involved in the extraction of the data each of which is discussed in the following sections.

7.2.3.1 Preprocessing

Images are preprocessed to modify and prepare the pixel values of the digitized image to produce an output that can be more easily analyzed in subsequent operations. This may consist of contrast enhancement, filtering to remove the noise of the hardware, and correction for camera distortion [10].

7.2.3.2 Segmentation

Segmentation is the process wherein the digitized image is broken down into meaningful regions. It is considered the first step of image analysis because the decision-making process of identifying the foreground and background has already been conducted. The simplest segmentation process is the identification of the foreground and background regions which is often easily achieved by thresholding. A very popular thresholding technique is the Ohtsu method [11].

7.2.3.3 Feature Extraction

After the image is divided into regions, the feature extraction process identifies objects in the region using descriptors. Basic descriptors are typically scalars that include area, centroid, perimeter, major diameter, compactness, and thinness [12]. These descriptors are often used simultaneously providing a good description of the object of interest.

7.2.3.4 Classification

Classification is the task of putting the objects in the image into some predefined categories. This process may be done by template matching or by a statistical method. Template matching is the comparison of the unknown objects to a set of known templates so as to identify the object.

Artificial intelligence, or machine learning, is being used more frequently in image classification in agricultural applications. In many of these applications, supervised machine learning is being used, where the user would enter and label several “training samples,” then the neural network would recognize connections between them. The neural network is then tested with “test samples” which it has not seen before, and the network is evaluated. Agricultural applications so far have primarily been concerned with plant identification, where plants were segmented under different lighting conditions [13], and it has been applied in weed management [14].

7.2.4 Actuation

Once the machine has identified the object, the decision on what the machine will do is known as the actuation process. This is the interaction of the machine with the environment or the original scene either directly or indirectly. This closes the machine vision system shown in Fig. 7.1. Usually, the machine vision is linked to a robotic system which is the basic component of automated operations [15].

7.3 Agricultural Machine Vision Applications

Machine vision systems typically use complex electronic sensors. The rapid development of computer technology and the photosensor has widened the field for machine vision applications. Currently, industries occupy most of the field of usage focusing mainly on product inspection, but other areas such as military science, astronomy, medicine, and the field of agriculture are now investigating the uses for machine vision [16]. For agriculture, researchers have been studying the potentials of machine vision in enhancing production which can be classified into three categories:

1. Plant identification
2. Process control
3. Machine guidance and control

The recent applications developed in these categories are described below.

7.3.1 Plant Identification

Plant identification refers to the process of classifying a certain plant by accurately identifying its component geometrical shape, size, and color. Figure 7.2 shows a schematic diagram of a machine vision system that is used for plant identification. Important parameters analyzed by the system are size, color, and shape, and surface temperatures. Making measurements of these parameters by noncontact visual means is an advantage of machine vision as identification and classification can be done without the risk of damage to the plant.

Several research projects on plant identification using machine vision have been conducted. Guyer et al. [17] developed a machine vision system that identifies plant species such as corn, soybeans, tomatoes, and some weed species at early growth stages using spatial parameters. The image processing stage evaluated the differences in the reflection of radiation from leaves and soil surface and the differences in the number of leaves and the shapes of the different weed species. This plant identification visual system could thus be used for selective herbicide application. A robotic vision-based system was developed to detect crop and weed locations, kill weeds, and thin crop plants [18]. This vision system identified different plant leaves using shape features that included area, major axis, minor axis, area to length ratio, compactness, elongation, length to perimeter ratio, and perimeter to broadness ratio. The system could then differentiate between tomato cotyledons and weeds when attached to a ground vehicle such as a tractor, and the prototype robotic weed control system could identify and treat weeds simultaneously.

With the advancement of aerial systems, a machine vision system can also be attached to an unmanned aerial vehicle (UAV) for the purpose of plant identification. Crop monitoring and assessment platform were developed to identify apple trees

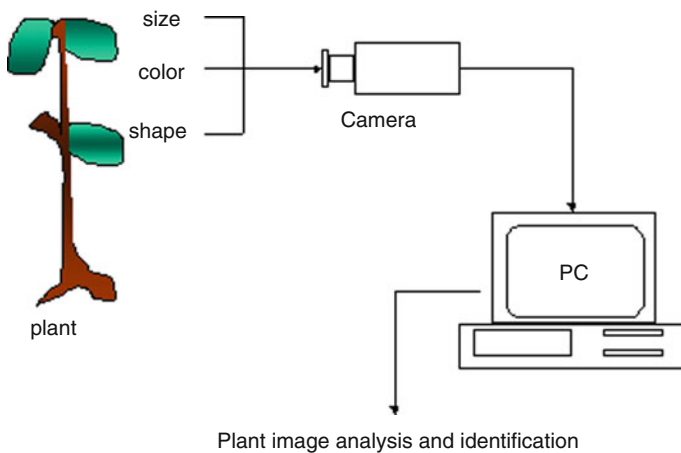


Fig. 7.2 Machine vision system for plant identification

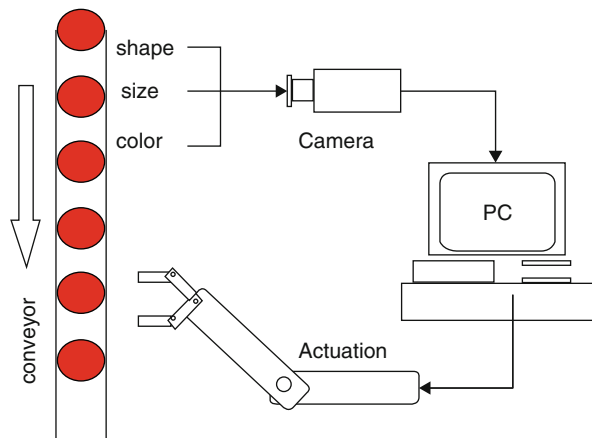
and monitor irrigation types [19]. This UAV machine vision system was composed of a multispectral camera (near-infrared, green, blue) and an image processing and analysis unit. The image processing calculated the enhanced normalized difference vegetation index to identify the tree crops and estimate the irrigation level and was able to differentiate between the full drip and 50% sprinkler irrigated trees. When identifying plants on a UAV system, the images will likely be acquired with a color camera, but using a color camera for the navigation of the UAV has potential problems. Concerning the navigation, using a laser triangulation system has several advantages compared to the color camera navigation. The main advantage is distance measurement, which can be measured to a high degree of accuracy, where a color camera system would estimate the distance [20]. Of course, there are errors when using a laser triangulation system for UAV navigation, such as the static and dynamic friction within the DC motors used in the system; but these errors can be estimated and accounted for, thus increasing the overall accuracy of the system [21].

7.3.2 Process Control

Industrial applications rely on visual systems for process control when the control is dependent on a visual parameter, for example, the inspection of circuit boards in a production line [22]. The system is able to make an intelligent action spotting and removing abnormal products. Generally, in visual sensing, the parameters being assessed are color, shape, and size.

In agriculture, evaluation of the color information indicates qualities such as maturity, sweetness, and wholesomeness. As shown in Fig. 7.3, a machine vision system may be used for the inspection of fruits by allowing the fruits to pass in front of a camera so that its quality may be evaluated.

Fig. 7.3 Machine vision for process control



Miller and Delwiche [23] studied a color vision system that inspects and grades fresh market peaches. Digital color images of the peaches taken as the fruit moved on a conveyor belt analyzed the peach for color, size, and surface characteristics. Compared to visual inspection by human senses, this system gave a high output rate, high reliability, and high uniformity and was additionally capable to make critical measurements.

There are machine vision systems that can detect wavelengths outside the visible electromagnetic spectrum. Bulanon et al. [24] developed a machine vision system to detect citrus black spot using hyperspectral imaging. Hyperspectral imaging allows the acquisition of spatial information across a sequence of individual bands covering a broad wavelength range, resulting in a three-dimensional image data with a very high spectral resolution. In the study, five different surface conditions including citrus black spot were evaluated. Linear discriminant analysis and an artificial neural network were then developed using wavelengths of 493 nm, 629 nm, 713 nm, and 781 nm. Both pattern recognition approaches had an overall detection accuracy of 92%. Rehkugler and Throop [25] developed a machine vision system that detects the defects in an apple.

In addition to the spectral properties of agricultural products, size, shape, form, freshness condition, and absence of visual defects are normally evaluated. Costa et al. [26] developed an automated shape processing system which could be used for both scientific and industrial purposes. This tool would be very useful for grading and sorting agricultural products especially if they were coupled with pattern recognition techniques [27]. It offers many advantages over conventional and mechanical sorting devices. Furthermore, evaluating the shape of agricultural products is a key parameter for allocating packaging and shipping resources [28].

7.3.3 Machine Guidance and Control

One of the important features of a robotic harvesting system is recognizing and locating a fruit. The commonly used camera gives a two-dimensional picture. Since three coordinates are required to fully locate the object, the distance dimension is lacking. This third dimension is typically acquired through the use of another sensor such as a range finding sensor, acoustics, radio frequency, or a stereographic vision system.

Researchers are trying to eliminate the need for an additional sensor by developing the range of information utilizing the object's geometric shape property, reflectance intensity, chrominance, and emissivity. The goal is to take a digital image of the object and then use image processing to identify and locate the position of the objects. Parrish and Goksel [29] conducted the first experimental system for an apple harvesting system. A black and white video camera was used to detect the apples. The image coordinates of the apple and its centroid were determined by image processing, and then the trajectory planning and the actuation routines directed the robotic arm to the apple. Figure 7.4 shows the generic machine vision system applied to fruit

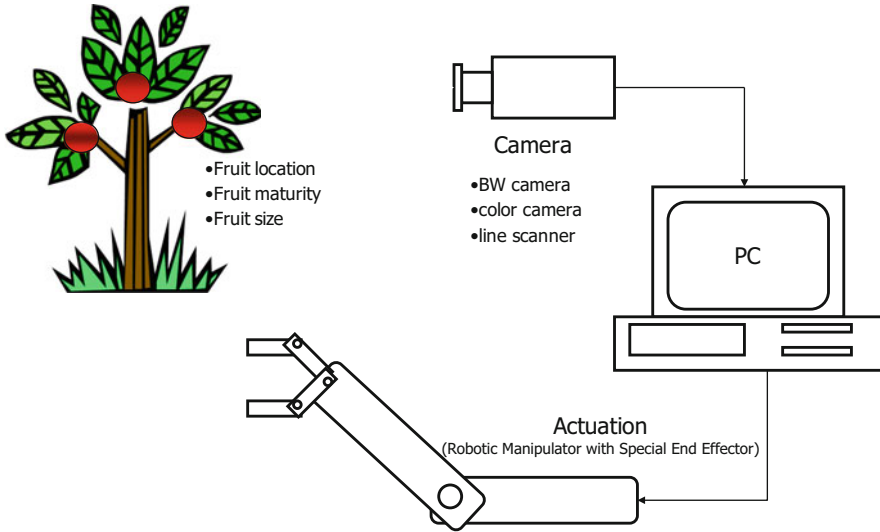


Fig. 7.4 Machine vision for fruit harvesting

harvesting. Similar to the development made by Parrish and Goksel, the features that are extracted from the image included color, shape, centroid location, and depth information. These features were then used to guide a robotic arm toward the fruit and pick it. Slaughter and Harrel [30] later improved the black and white video camera by using a colored video camera. This time, the detection of apples was dependent not only on gray-level intensity but also on color. The color factor is an important parameter in differentiating the object from its background. Another example of machine vision-based fruit harvesting is the apple robotic harvesting system developed by Bulanon and Kataoka [9]. The segmentation method was based on the chromaticity coefficients red and green combined with a decision-theoretic approach method to threshold the apple fruits from the background under variable lighting conditions. The vision system was used to guide a customized end effector that picked the fruit in a manner similar to the way a human would pick the apple.

One of the problems encountered in a robotic vision system is the similarity of the spectral reflectance between the object and its background specifically the leaves of the tree. Recent studies have focused on using the thermal characteristics of the fruit to separate it from the foliage. Bulanon et al. [31] studied the thermal characteristics of the citrus tree. A 24-hour temperature profile between the fruits and the leaves was obtained, and it was found that the fruits had a higher surface temperature than the leaves during the nighttime. Thus a unique image processing approach which combined color and thermal images using fuzzy logic was developed.

Another robotic system that could be guided by machine vision is an agricultural ground vehicle. The vehicle could be manned or unmanned. If the vehicle is manned, the machine vision system is used to assist the driver in steering the system while

an unmanned vehicle would be fully autonomous. The last section of this chapter discusses the development of a machine vision system for steering an unmanned ground vehicle in a commercial peach orchard.

7.4 Machine Vision Development for Fruit Yield Estimation: An Example of Plant Identification Application

Section 7.3 discussed the different applications of machine vision, those being plant identification, process control, and machine guidance and control. This section will discuss a plant identification application of machine vision specific to orchard management. This development was created under a research project of the Robotics Vision Lab at Northwest Nazarene University, where the goal of the project was early fruit yield estimation. Yield estimates are important for growers to help in the production planning and marketing of the fruits. There are several ways of estimating fruit yield [32, 33], and machine vision is one of the popular tools available [34–38]. Most of these vision-based yield estimators [39] count the fruits when they are almost ready to harvest; however, an early yield estimate [40] is more important to the growers. The hypothesis of the project was that by counting the number of blossoms of a fruit tree, an early yield estimate could be derived. The fruits of interest in this project were apples and peaches: specifically, Pink Lady apples grown in a high-density orchard and Snow Giant peaches grown in a standard orchard. Both orchards were located in Caldwell, Idaho, and were planted in a north-south direction. Thirty trees were selected randomly from a block in each orchard and photographed throughout the blossom period during the 2018 growing season. A 12-megapixel 24-bit digital color camera was used to photograph each tree on the east and west sides. Later in the season when the fruits were mature, a ground truth yield was obtained by manually counting the fruits on the selected trees.

The images were processed using MATLAB and its Digital Image Processing Toolbox [41]. Figure 7.5 displays a sample image of a blossoming apple tree in a high-density orchard. The height of each apple tree is approximately 8 ft, and there are approximately 4 ft between each tree. In this orchard, images were acquired approximately 10 ft from the tree. Figure 7.6 displays a blossoming peach tree in a standard orchard. The height of each peach tree is approximately 15 ft, and there is approximately 10 ft between each tree. In this orchard, images were acquired approximately 13 ft from the tree.

7.4.1 Image Processing for Blossom Isolation

With image acquisition completed, the next step is to isolate and count the blossoms for each apple and peach tree.



Fig. 7.5 Sample image of a blossoming apple tree in a high-density orchard



Fig. 7.6 Sample image of a blossoming peach tree in a standard orchard

7.4.1.1 Methods of Data Transformations

Before the blossoms could be isolated, a set of sample data needed to be collected to determine the color properties of each category in the image, so that a color filter can be created from this data, which will isolate the blossoms. This sample data was collected manually, where 600 different pixels for each category of the image were

manually selected, and the RGB values of those pixels were recorded. Within the 600 pixels selected for each category, 300 pixels were selected from images taken from the east side of the tree, and 300 pixels were selected from the images taken from the west side of the tree. The five main categories of classification in each image for both the apple and peach images are the sky, blossoms, leaves/grass, branches, and dirt. A 3D scatterplot displaying the recorded RGB values for the apple images is displayed in Fig. 7.7. As seen in Fig. 7.7, the RGB values of the sky is not included. This is because by manually analyzing the images, it has been noticed that the sky is a relatively large category in size and that the pixels are all connected. Because the pixels of the sky are all connected, an area feature extraction method, which is explained later in this section, can easily be implemented, which will remove the sky from the image.

The goal is to isolate the red circle data points, which represent the blossom’s RGB values, so that when analyzing the entire image, the blossoms can be isolated. There are several image analysis functions in MATLAB that could be used to isolate the blossoms, but because MATLAB is not an open source software, it is preferred to use a method of isolation not using these functions.

One method of blossom isolation investigated was to apply a transformation matrix to each sample data point, written mathematically as

$$Ax = b \tag{7.1}$$

where b is the new value of the sample data point, A is the transformation matrix, and x is the red, green, and blue values of the sample data point. In this form, these matrices have the form:

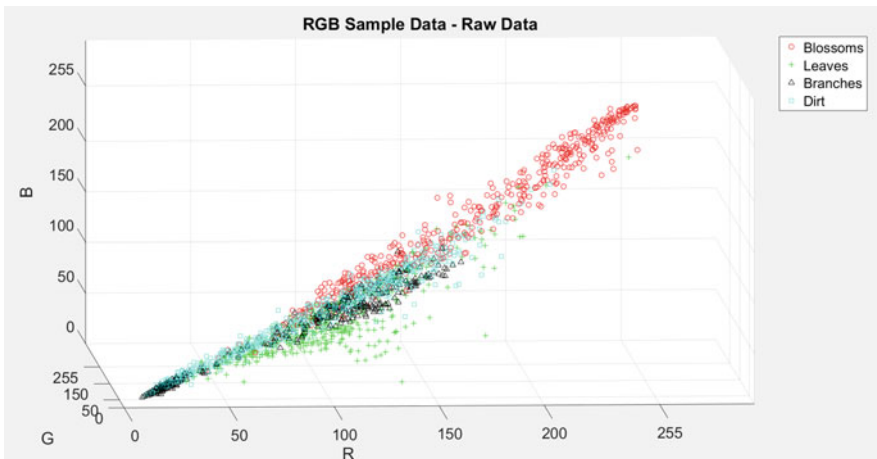


Fig. 7.7 Sample RGB values of objects in the apple orchard

$$A = \begin{bmatrix} a_{1,1} & a_{1,1} & a_{1,3} \\ \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,1} & a_{n,3} \end{bmatrix} \quad (7.2)$$

$$\mathbf{x} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (7.3)$$

$$\mathbf{b} = \begin{bmatrix} Ra_{1,1} + Ga_{1,2} + Ba_{1,3} \\ Ra_{2,1} + Ga_{2,2} + Ba_{2,3} \\ \vdots \\ Ra_{n,1} + Ga_{n,2} + Ba_{n,3} \end{bmatrix} \quad (7.4)$$

where the element $a_{n,3}$ is an element in A occupying the n th row and 3rd column. When transformation matrix A is applied to the sample data matrix \mathbf{x} , the image is \mathbb{R}^n .

An example of a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^1$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ would be a summative transformation that adds the red, green, and blue values of each pixel. The matrix A would take the form seen in Eq. (7.5).

$$A = [1, 1, 1] \quad (7.5)$$

Applying this matrix to the scatterplot displayed in Fig. 7.7 results in the data points being transformed to a single axis. This is difficult to display because the data points are clustered, so the results transformation are displayed with a histogram in Fig. 7.8.

An example of a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ is to rotate the 3D scatterplot displayed in Fig. 7.7 such that only two of the axes can be seen. If it was desired to display the red and blue axes, matrix A would take the form

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.6)$$

Applying this matrix to the scatterplot displayed in Fig. 7.7, the result is a 2D scatterplot displayed in Fig. 7.9.

A transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ is to move the data points in the 3D scatterplot to a different location on the same 3D scatterplot. An example of this is to take a ratio transformation, that is, to take the red, green, and blue components of each pixel and divide it by the sum of its respective red, green, and blue components. The matrix A would take the form

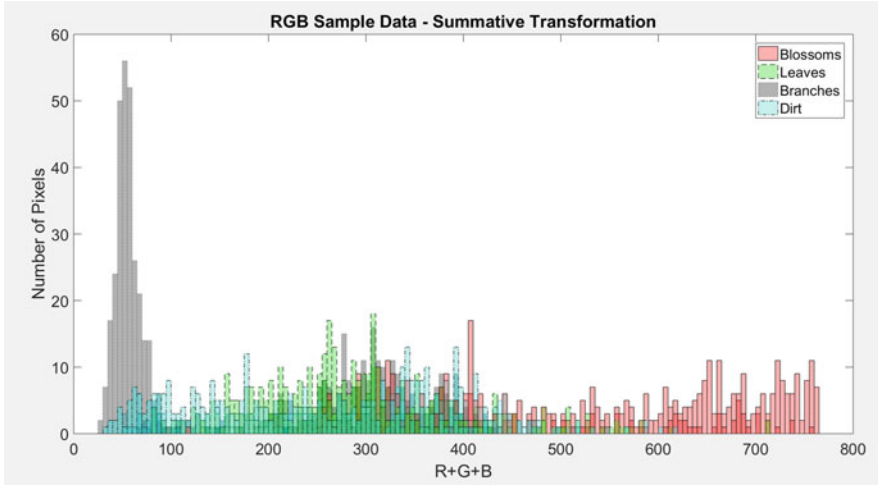


Fig. 7.8 Histogram of the summative transformation of the sample RGB values

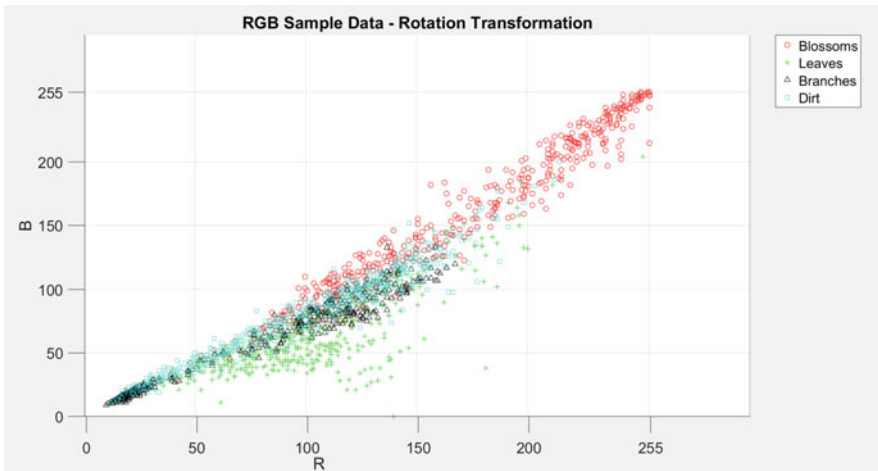


Fig. 7.9 Rotation transformation of the sample RGB values

$$A = \begin{bmatrix} (R + G + B)^{-1} & 0 & 0 \\ 0 & (R + G + B)^{-1} & 0 \\ 0 & 0 & (R + G + B)^{-1} \end{bmatrix} \tag{7.7}$$

This transforms each sample data point onto the plane

$$x + y + z = 1 \tag{7.8}$$

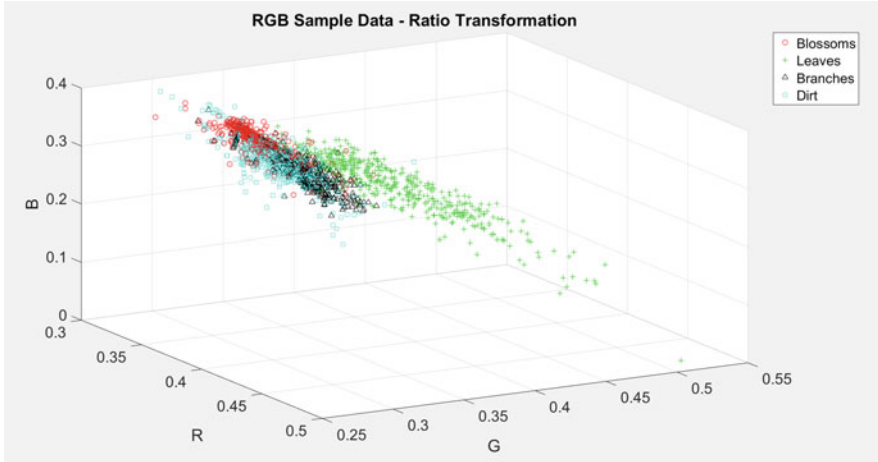


Fig. 7.10 Ratio transformation of the sample RGB values

Applying this matrix to the scatterplot displayed in Fig. 7.7, the result is the 3D scatterplot displayed in Fig. 7.10.

There are infinitely many transformations that can be applied to the set of sample data, such as transformation matrix A yielding a new data point \mathbf{b}

$$A = \begin{bmatrix} 2 & 3 & 7 \\ 5 & 8 & 1 \\ 4 & 6 & 9 \end{bmatrix} \tag{7.9}$$

$$\mathbf{b} = \begin{bmatrix} 2R + 3G + 7B \\ 5R + 8G + 1B \\ 4R + 6G + 9B \end{bmatrix} \tag{7.10}$$

As mentioned previously, the dimension of \mathbf{b} can extend past three. If A is a 4×3 matrix, then \mathbf{b} is in \mathbb{R}^4 . Though these can often be difficult to describe graphically, so examples of this and higher dimensions of \mathbf{b} will not be presented in this chapter.

7.4.1.2 Testing Blossom Isolation

Recall that the goal of applying a transformation matrix is to isolate the red circle data points in the sample data. Looking back at Fig. 7.9, two lines can be drawn which separates the blossom sample data points from the other categories, as seen in Fig. 7.11.

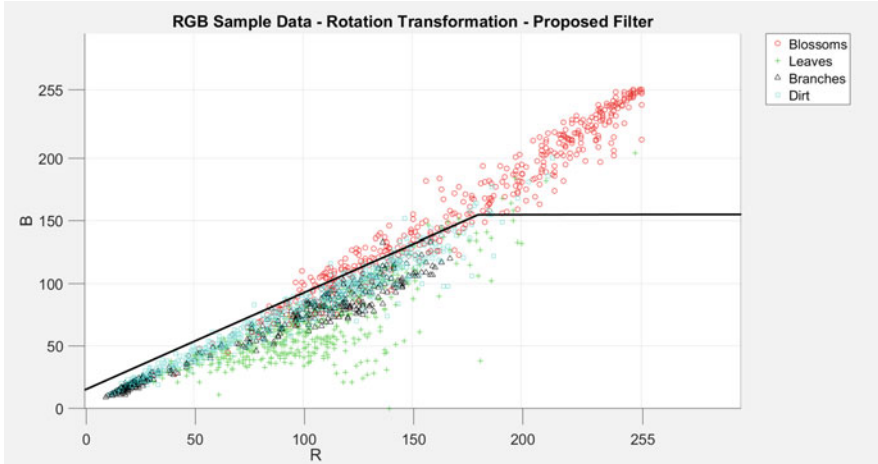


Fig. 7.11 Blossom isolation in the rotation transformation

The equation of these lines is the color filter that will be used to isolate the blossoms from the other objects when filtering an entire image. By using the equations, the points above or below the line can be set to zero, thus isolating a section of the data.

For example, the equations of the lines in Fig. 7.11 are

$$7 \times \text{Red} - 9 \times \text{Blue} - 135 = 0 \tag{7.11}$$

and

$$\text{Blue} = 155 \tag{7.12}$$

Thus, the red circle data points can be isolated by applying the pseudocode:

```

if((7 × Red - 9 × Blue - 135 > 0) && (Blue < 155)) {
  Red = 0
  Blue = 0
}
    
```

Applying this code to the data set on Fig. 7.11 yields the plot displayed in Fig. 7.12.

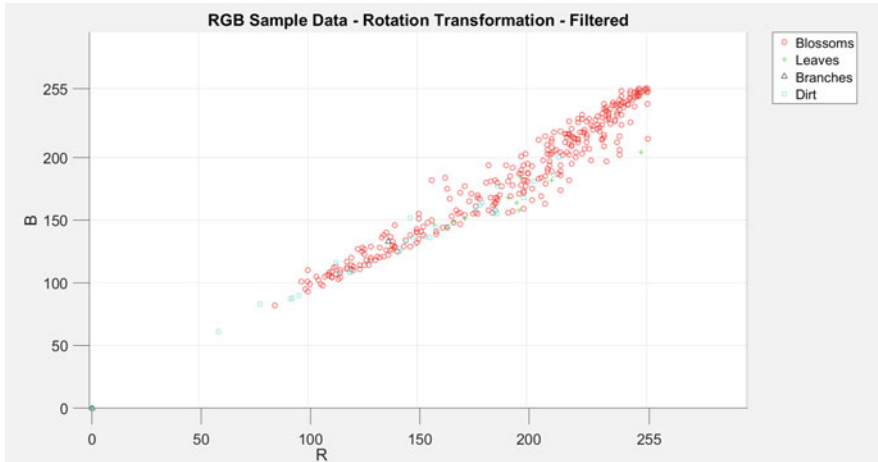


Fig. 7.12 Results of the blossom isolation color filter

7.4.1.3 Tree Isolation

As noticed by the sample image of the apple tree displayed in Fig. 7.5, there are three trees present in the foreground of the image with multiple others in the background. This is common for this style of high-density orchard as the trees are only separated by approximately 4 ft. Because the goal is to count the blossoms on the center tree, a method of isolating the center tree must be derived. Figure 7.8 shows a large clustering of data in the lower region of the histogram, RGB values less than 100 after applying the summative transformation, which is classified as either branches or dirt. This implies the branches can be used as a method of tree isolation, specifically the trunks can be used because they are the most isolated from each other.

Using a copy of the image, a tree isolation algorithm can be created. The results of this algorithm will be applied to the original image before the blossom isolation algorithm is applied. Because the trunks are the means of tree isolation, the first step of the trunk isolation algorithm is to crop out the top two thirds of the copied image. As explained in Sect. 7.2, this first step is the preprocessing in the tree trunk image processing algorithm. The next step of the algorithm is to isolate the trunks, which can be done by applying the following pseudocode to each pixel of the image:

```

if (Red + Green + Blue < 100) {
  Red = 0
  Green = 0
  Blue = 0
}

```

This results in the image displaying the trunks of each tree, but there is some noise, as some dirt samples remain. To remove these samples, a size filter can be applied because the number of dirt pixels passing through the transformation filter is much less than the number of branch pixels passing through the transformation filter. With this process what mostly remains are the three foreground tree trunks.

The next step then is to isolate the central tree trunk. For this task, a MATLAB data structure called “regionprops” was used. This function was used even though it was earlier stated that the use of MATLAB specific features was undesirable. This is because this function is also available through open source methods, such as the OpenCV library [42] or the ImageJ package, Fiji [43].

MATLAB’s regionprops measured the properties of an image’s regions—area, centroid, major and minor axis lengths—and then applied a bounding box to the region. The centroid feature of regionprops can be used to determine the location of each trunk, thereby giving the center position of each tree. Due to the nature of high-density orchards, where the trees tend to be vertical with little overlap of branches, the center tree can now be isolated. Using the position of the trees, the left and right trees can be cropped out by drawing a vertical line at the midpoint of the center tree and trees to the left and right of center.

This method works with the apple trees because they are in a high-density orchard, but peach trees are planted farther apart. Three trunks are not always seen in each image as previously shown in Fig. 7.6. Vertical line trunk isolation is not a viable option as there is no “center” tree. Instead, the natural geometry of the peach tree is used as a method of isolation.

Peach trees have four main branches that stem from the trunk, which makes a shape similar to an upside-down pyramid in the empty space within the four main branches. So, when an image of the tree is taken, the four main branches have a “V” shape. Thus, the tree of interest can be isolated by drawing a line from the top corners of the image to the bottom center, cropping out the bottom two corners. Figure 7.13 displays the result of the tree isolation algorithm applied to the original apple and peach tree images.

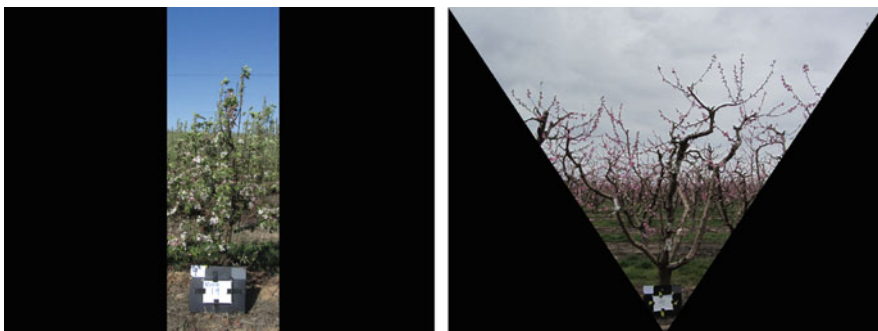


Fig. 7.13 Tree isolation process results for an apple and peach trees

7.4.1.4 Blossom Isolation and Counting for Apple Trees

Now that the center tree is isolated, the blossom filter described in Sect. 7.4.1.2 can be applied to each pixel in the apple image. Referring to Fig. 7.11, the line that isolates the blossoms from the other categories in the image is very close to including data points of the other categories. Because there is no significant gap in the separation of the blossoms and the other categories, a significant amount of noise in the resulting image after applying the color filter can be anticipated, which is exactly what is seen in Fig. 7.14, after the blossom isolation color filter is applied to the image.

As seen in Fig. 7.14, several of the pixels from the leaves passed through the color filter as has the entire sky. Both issues can be resolved by applying a size filter focusing on removing small and large groups of pixels. This size filter uses the regionprops data structure mentioned earlier in Sect. 7.4.1.3, where if the area is outside of the range of a specified pixel count, then the pixels are set to zero. It should be noted that the specific range which will allow an area to pass through the size filter varies depending on the size of the image. There are more pixels in a 12-megapixel camera (which was used in acquiring these images) than an 8-megapixel camera, so the allowable area of an image from a 12-megapixel camera should be higher than the allowable area of an image from an 8-megapixel camera. Because of this, care needs to be taken to match the filter parameters to the number of pixels in an image. After applying the size filter, the resulting binary image is displayed in Fig. 7.15.

The remaining areas are the identified blossoms on the tree. The regionprops data structure will now be used to label each area and to obtain a count for the total blossoms. In addition, a bounding box around each area can be applied, and the boxes can be overlaid over the original image to visually check the accuracy of the

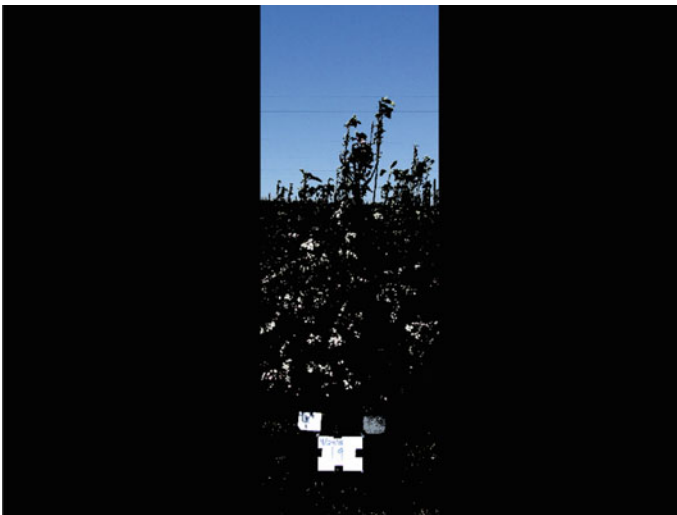


Fig. 7.14 Color filter applied to an image of an apple tree

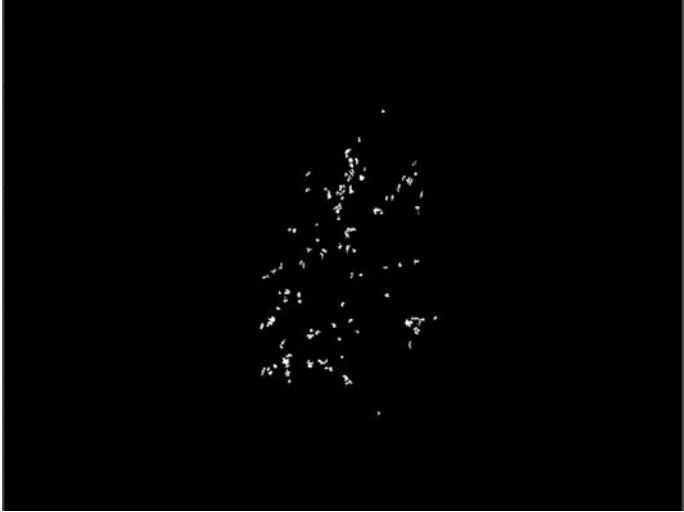


Fig. 7.15 Size filter applied to the apple tree image



Fig. 7.16 Bounding boxes of the identified blossoms overlaid on the original image

program. This image is displayed in Fig. 7.16, where it can be seen that there are very few false positives and false negatives in the image.

7.4.1.5 Blossom Isolation and Counting for Peach Trees

The process for identifying the blossoms on a peach tree is virtually the same as the process for the apple tree. The largest difference is that different transformation matrices may be applied and there will be a different equation applied to each pixel to apply the color filter. In the case of blossom identification, the transformation matrix which was applied is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{7.13}$$

which rotated the 3D scatterplot to display the red and green color values. Figure 7.17 through Fig. 7.20 displays the overall process of blossom isolation for a peach tree. The rotation transformation of the sample RGB data and the line displaying the color filter are displayed in Fig. 7.17, the tree isolation algorithm and the color filter applied to the sample image seen in Fig. 7.6 are displayed in Fig. 7.18, the result of the size filter to remove the noise is displayed in Fig. 7.19, and the bounding boxes overlaid on the original image is displayed in Fig. 7.20.

After analyzing Fig. 7.20, there appears to be a significant number of false negatives. This observation may lead to the conclusion that the algorithm is not very successful in identifying peach blossoms; however, the false negatives seen in Fig. 7.20 were intentionally produced. This particular type of peach undergoes an intensive thinning process. Thus, to produce a better final yield estimate, fewer regions were desired. If an orchard were to not be significantly thinned, then a different size filter should be applied.



Fig. 7.17 Blossom isolation process for a peach tree

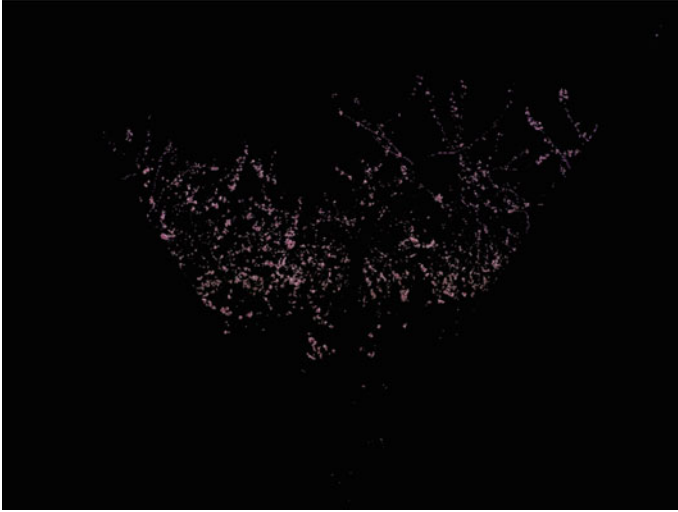


Fig. 7.18 Color filter applied to the peach tree

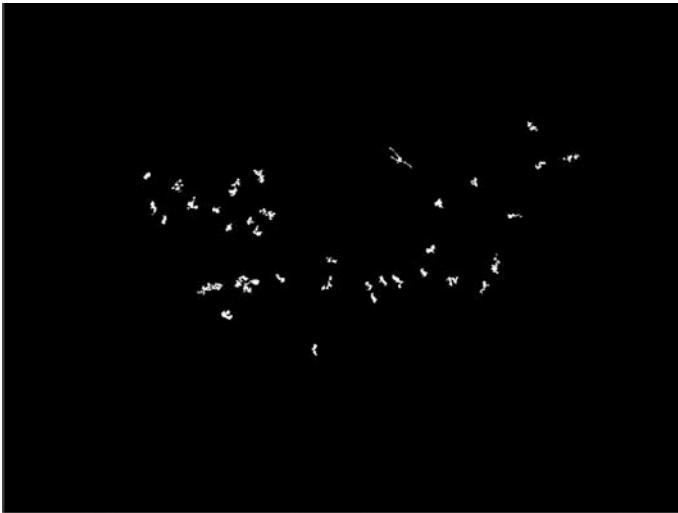


Fig. 7.19 Size filter applied to the peach tree

7.4.2 Results of Yield Estimation

So far, an algorithm has been developed which isolates and counts the number of blossoms on an apple and peach trees. There still remains steps to produce the final result of fruit yield estimate.



Fig. 7.20 Bounding boxes of the identified blossoms overlaid on the original image of the peach tree

7.4.2.1 Transition from Blossom Count to Yield Estimation

Once the blossom isolation process for each image has been applied, a blossom count for each side of the tree has been obtained. Recall from the beginning of Sect. 7.4, it was explained that two images of the tree of interest were acquired, one image from the east side and one from the west side. This is significant because the blossom count from each side of the tree cannot simply be added together to obtain a total fruit yield estimation, because there is a significant risk of double-counting blossoms.

Consider this question, “What if on the east side of the tree, the blossom count is consistent between each tree, but there is a large variance in the blossom count between each tree on the west side?” Intuition would say the blossom count from the east side should play a larger role in the yield estimation, because of the more consistent blossom count. Consistency in the blossom count is important because hypothetically there should only be a small difference in the total blossom count from tree to tree as each tree in a section of an orchard is of the same age and same size.

This is a case where intuition is correct, because the correct method of determining a yield from a set of two blossom counts is to calculate a weight that will be applied to blossom counts from the east side, and a different weight from blossom counts taken on the west side. The derivation for the two weights is described in Sect. 7.4.2.2, and it will be seen that the two weights depend on the variances and covariances [44] of the blossom counts from the east side, blossom count from the west side, and ground truth number of fruits.

7.4.2.2 Derivation of Weight Values

In the following derivation, random variables are denoted by capital letters, actual values of those variables by lowercase letters, and vectors by boldface type. The correlation between the blossom count from the images and the actual fruit count was determined. There are 30 selected trees, which are numbered #1 through #30. For tree # i , there are

$$X_E = x_{E,i} \quad (7.14)$$

blossoms visible from the East and

$$X_W = x_{W,i} \quad (7.15)$$

blossoms visible from the West. The eventual fruit yield of tree # i is

$$Y = y_i. \quad (7.16)$$

The data is represented by vectors in \mathbb{R}^n

$$\mathbf{x}_E = (x_{E,1}, x_{E,2}, \dots, x_{E,n}), \quad (7.17)$$

$$\mathbf{x}_W = (x_{W,1}, x_{W,2}, \dots, x_{W,n}), \quad (7.18)$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n). \quad (7.19)$$

Choose weights α_E and α_W , with $\alpha_E, \alpha_W \geq 0$, and with $\alpha_E + \alpha_W = 1$. Then construct an equation of the form

$$Y' = m(\alpha_E X_E + \alpha_W X_W) + c \quad (7.20)$$

which will be the least-squares regression line of

$$X = \alpha_E X_E + \alpha_W X_W. \quad (7.21)$$

The RMS error of $Y \approx Y'$ will be

$$s_Y \sqrt{1 - r_{\mathbf{x}, \mathbf{y}'}^2} \quad (7.22)$$

where $r_{\mathbf{x}, \mathbf{y}}$ is the correlation coefficient of the data $\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W$ with \mathbf{y} . Accordingly, the RMS error of the linear model will be minimized if α_E and α_W are chosen to maximize the correlation coefficient $r_{\mathbf{x}, \mathbf{y}}$.

Writing

$$x_i = \alpha_E x_{E,i} + \alpha_W x_{W,i}, \quad (7.23)$$

$$\mathbf{x} = (x_1, \dots, x_n), \quad (7.24)$$

and

$$\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W, \quad (7.25)$$

define the means

$$\bar{x}_E = \frac{\sum_{i=1}^n x_{E,i}}{n} \quad (7.26)$$

$$\bar{x}_W = \frac{\sum_{i=1}^n x_{W,i}}{n} \quad (7.27)$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (7.28)$$

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (7.29)$$

the mean vectors

$$\bar{\mathbf{x}}_E = (\bar{x}_E, \dots, \bar{x}_E) \quad (7.30)$$

$$\bar{\mathbf{x}}_W = (\bar{x}_W, \dots, \bar{x}_W) \quad (7.31)$$

$$\bar{\mathbf{x}} = (\bar{x}, \dots, \bar{x}) \quad (7.32)$$

$$\bar{\mathbf{y}} = (\bar{y}, \dots, \bar{y}), \quad (7.33)$$

and the deviation vectors

$$\tilde{\mathbf{x}}_E = \mathbf{x}_E - \bar{\mathbf{x}}_E \quad (7.34)$$

$$\tilde{\mathbf{x}}_W = \mathbf{x}_W - \bar{\mathbf{x}}_W \quad (7.35)$$

$$\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}} \quad (7.36)$$

$$\tilde{\mathbf{y}} = \mathbf{y} - \bar{\mathbf{y}} \quad (7.37)$$

so that

$$\bar{\mathbf{x}} = \alpha_E \bar{\mathbf{x}}_E + \alpha_W \bar{\mathbf{x}}_W, \quad (7.38)$$

and

$$\tilde{\mathbf{x}} = \alpha_E \tilde{\mathbf{x}}_E + \alpha_W \tilde{\mathbf{x}}_W. \quad (7.39)$$

Then

$$r_{\mathbf{x},\mathbf{y}} = \frac{\tilde{\mathbf{x}} \bullet \tilde{\mathbf{y}}}{\sqrt{\tilde{\mathbf{x}} \bullet \tilde{\mathbf{x}}} \sqrt{\tilde{\mathbf{y}} \bullet \tilde{\mathbf{y}}}}, \quad (7.40)$$

while

$$\tilde{\mathbf{x}} = \alpha_E \tilde{\mathbf{x}}_E + \alpha_W \tilde{\mathbf{x}}_W. \quad (7.41)$$

The first expression is the cosine of the angle

$$\theta_{\tilde{\mathbf{x}},\tilde{\mathbf{y}}} \quad (7.42)$$

between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, all vectors $\tilde{\mathbf{x}}$ in the plane spanned by $\tilde{\mathbf{x}}_E$ and $\tilde{\mathbf{x}}_W$ need to be maximized. To do this, $\tilde{\mathbf{y}}$ must be projected into this plane, and an orthogonal basis for the plane is desired. Using the Gram–Schmidt Method [45] on the $\tilde{\mathbf{x}}_E$ and $\tilde{\mathbf{x}}_W$ an orthogonal basis

$$\left\{ \tilde{\mathbf{x}}_E, \left(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E \right) \tilde{\mathbf{x}}_W - \left(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W \right) \tilde{\mathbf{x}}_E \right\} \quad (7.43)$$

is obtained for

$$\text{span} \left\{ \tilde{\mathbf{x}}_E, \tilde{\mathbf{x}}_W \right\}. \quad (7.44)$$

Project $\tilde{\mathbf{y}}$ onto each of these two orthogonal basis vectors, and add the projections to obtain

$$\begin{aligned}
& \frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E} \tilde{\mathbf{x}}_E \\
& + \frac{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}})}{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - 2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2 + (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)} \\
& \quad \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E) \tilde{\mathbf{x}}_W - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W) \tilde{\mathbf{x}}_E \right) \\
& = \frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E} \tilde{\mathbf{x}}_E + \frac{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}})}{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2} \\
& \quad \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E) \tilde{\mathbf{x}}_W - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W) \tilde{\mathbf{x}}_E \right). \tag{7.45}
\end{aligned}$$

Since $r_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}}$ will be unaffected by multiplying $\tilde{\mathbf{x}}$ by a scalar, multiply the last vector by the denominator

$$(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2 \tag{7.46}$$

to simplify the expression

$$\begin{aligned}
& \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right. \\
& \quad \left. - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) + (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_E \\
& + \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_W \tag{7.47}
\end{aligned}$$

Two terms may be canceled to obtain the following:

$$\begin{aligned}
& \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_E \\
& + \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_W \tag{7.48}
\end{aligned}$$

Finally, divide by the scalar $n^2 \tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E$

$$\begin{aligned} & \left(\left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) \right) \tilde{\mathbf{x}}_E \\ & + \left(\left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) \right) \tilde{\mathbf{x}}_W \end{aligned} \quad (7.49)$$

Then let

$$\begin{aligned} \beta_E &= \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) \\ &= \text{var} [\mathbf{x}_W] \text{cov} [\mathbf{x}_E, \mathbf{y}] - \text{cov} [\mathbf{x}_E, \mathbf{x}_W] \text{cov} [\mathbf{x}_W, \mathbf{y}] \end{aligned} \quad (7.50)$$

and

$$\begin{aligned} \beta_W &= \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) \\ &= \text{var} [\mathbf{x}_E] \text{cov} [\mathbf{x}_W, \mathbf{y}] - \text{cov} [\mathbf{x}_E, \mathbf{x}_W] \text{cov} [\mathbf{x}_E, \mathbf{y}] \end{aligned} \quad (7.51)$$

Setting

$$\alpha_E = \frac{\beta_E}{\beta_E + \beta_W} \quad (7.52)$$

and

$$\alpha_W = \frac{\beta_W}{\beta_E + \beta_W} \quad (7.53)$$

yields the desired vector

$$\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W, \quad (7.54)$$

and of course

$$\bar{x} = \alpha_E \bar{x}_E + \alpha_W \bar{x}_W, \quad (7.55)$$

with $\alpha_E + \alpha_W = 1$. To finish, take the least-squares regression line of \mathbf{y} on \mathbf{x} . Writing

$$m = \frac{\text{cov} [\mathbf{x}, \mathbf{y}]}{\text{var} [\mathbf{x}]} \quad (7.56)$$

for its slope, the line is

$$y' = m(x - \bar{x}) + \bar{y}. \tag{7.57}$$

Given the counts X_E from the East and X_W from the West, the prediction for the fruit yield Y is

$$Y' = m(\alpha_E X_E + \alpha_W X_W - \bar{x}) + \bar{y}. \tag{7.58}$$

7.4.2.3 Statistical and Probabilistic Results

The values of the weights (α_E and α_W) for the apple and peach orchard are displayed in Table 7.1. As seen from this table, the weights applied to the peach blossom count are very similar, where the weights applied to the apple blossom count have a large difference. This is because the images were acquired at 9 a.m., and there was slight overcast when acquiring the peach orchard images, where there were clear skies when acquiring the apple orchard images. So for the apple orchard, images from the west side were looking into the sun, thus increasing the blossom count variance between each tree, and lowering the weight. For the peach orchard, there was overcast, so the images from the east and west sides were consistent, thus resulting in consistent weights.

Table 7.2 elaborates on the results for the apple and peach orchards. In this table, the probability of underestimation is the confidence which the program will not overestimate the number of fruit. This number was calculated by performing a right-tailed binary hypothesis test, by using the two averages of each orchard along with the sample standard deviation to determine a Z-score, which then can be used to determine a significance level [44]. It is important to have a high probability of underestimation, so the farmer is not misled in the number of fruit he/she has, while maintaining a low percent error, so the farmer can have an accurate revenue estimation and be accurate in resource allocation.

Table 7.1 Value of weight applied to the blossom counts

	Apple orchard	Peach orchard
East side	0.639	0.475
West side	0.361	0.525

Table 7.2 Additional results of the early yield estimation program

	Apple orchard	Peach orchard
Coefficient of correlation between ground truth yield and predicted yield	0.699	0.606
Ground truth average fruit per tree	114.17	66.733
Estimated average fruit per tree	103.47	61.440
Percent error	-9.37%	-7.93%
Probability of underestimation	97.19%	95.25%

7.4.3 General Image Processing Techniques for Other Projects

What are the main points which can be taken away from the process which was described in the previous sections? If the images which are being processed have the same set of objects, the RGB data can be used to isolate the objects in the image. By manually collecting a set of sample data and applying different transformations to them, the transformation which best isolates the object of interest can be determined. Then after applying an n -dimensional transformation matrix to the sample data, an equation with n independent variables can be determined that filters out the other objects in the image.

In the example of isolating the blossoms in apple and peach trees, which was presented in Sect. 7.4.1, a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ was applied to the data set. This transformation greatly depends on the scene constraints and the application of the system. Zhang et al. [46] used a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^1$, where they monitored tea leaves to determine the optimal time for harvesting. In this study, the blue plane was subtracted from the green plane. The primary reason why the $\mathbb{R}^3 \rightarrow \mathbb{R}^1$ transformation worked better than the $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ or a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ transformation is the scene constraints. So when applying machine vision to a system, the scene constraints should have a large influence on which types of transformation is applied.

7.4.3.1 Potential Problems with Over-Constraining the Sample Data

Elaborating more on the equation with n independent variables to isolate the object of interest, this equation can be very involved, but it is often better to use a simple, linear equation. As with the example that isolates blossoms, a linear equation was used because if a parabolic equation with a high degree is used, there is a chance of over constraining the sample data, which may not be useful when applied to the entire image.

Suppose there is a set of images with two objects. A set of RGB sample data is taken, and a rotation transformation to display the red and green values is applied. In this hypothetical situation, this 2D scatterplot may look like the scatterplot displayed in Fig. 7.21. At this point, a line separating the two objects would be drawn, and the equation of this line is the color filter which would be applied to the entire image.

If a straight line is drawn in the sample data set in Fig. 7.21, there is some error, which can be seen in Fig. 7.22. So it would be very tempting to draw a high-degree parabolic function, to get complete separation between the two objects. This high degree polynomial line, which also can also be seen in Fig. 7.22, is an example of over constraining the sample data, and it may have less success when applied to the image because of its complexity.

Suppose the RGB values of each pixel of both objects were known. Of course, this is an impossible task because each individual pixel would have to be analyzed for each image in the set, determining which category the pixel belonged to, and this is potentially millions of pixels. Regardless, suppose these data set exists. If the linear

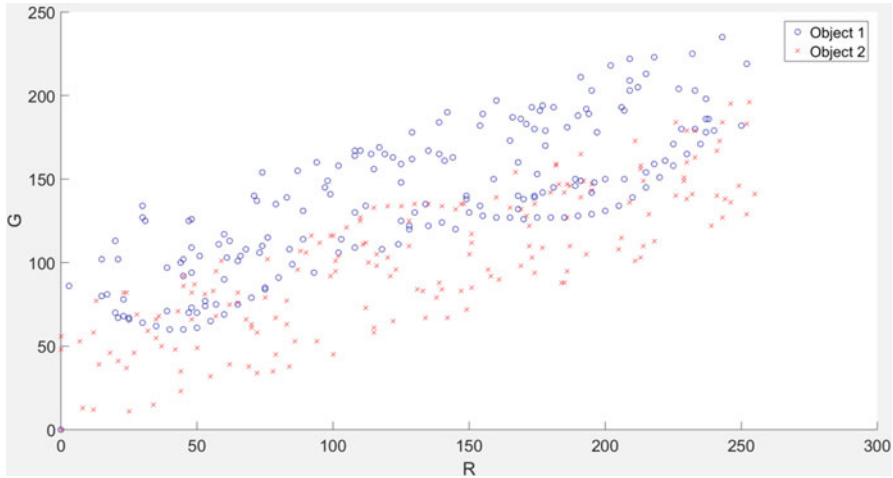


Fig. 7.21 Sample data set of a hypothetical situation

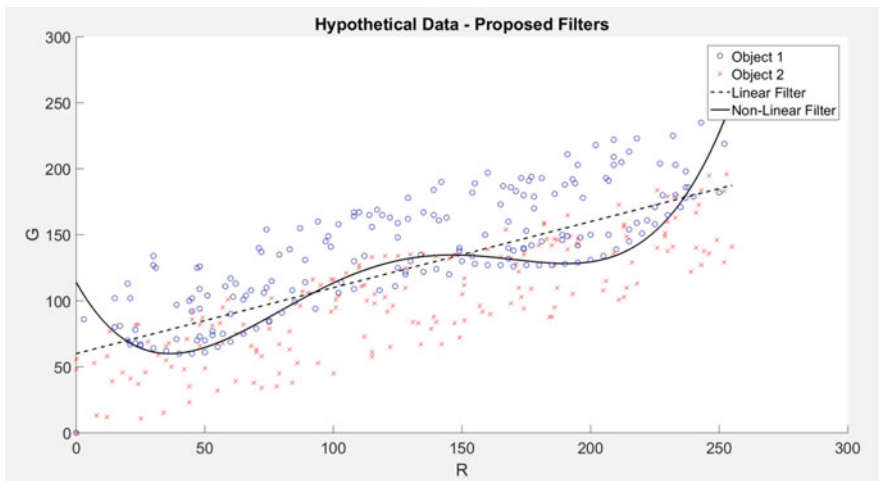


Fig. 7.22 Over constrained and linear equations applied to the sample data set

and parabolic function was applied to this complete data set, seen in Fig. 7.23, it is seen why it is better to use the linear equation compared to the parabolic equation.

But this is just a hypothetical situation used to prove the point of how linear equations are sometimes better than a parabolic equation. How is this proved? Looking back at Fig. 7.21, there are not a lot of sample data points, compared to the complete data set seen in Fig. 7.23. The small sample data size is the reason for the error when applied to the complete data set. The more sample data points which are collected, the higher the confidence which can be had in a complex, nonlinear filter. Going back to the apple and peach trees, there are 12 million pixels in the image and

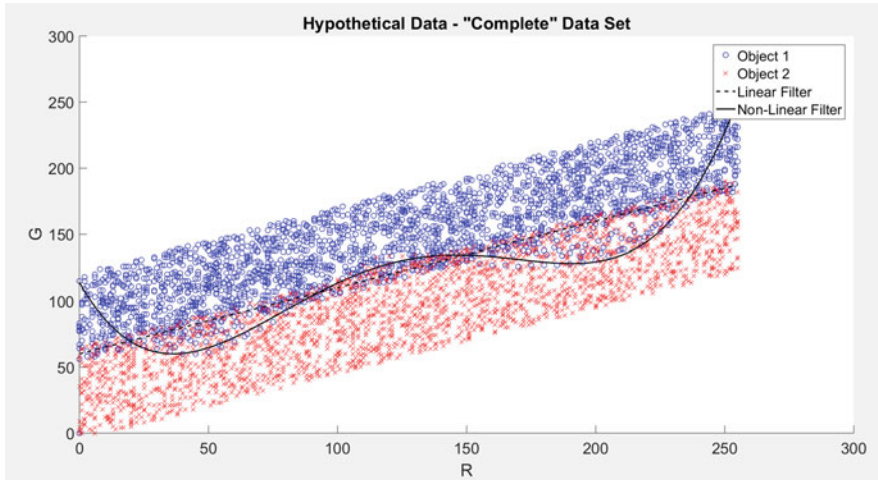


Fig. 7.23 Over-constrained and linear equations applied to the “complete” dataset

60 pictures for each orchard. From this, it is very easy to see it would be extremely time intensive to gather enough pixels to have enough confidence to use an advanced function, which is why it is better to use a simple, linear function.

7.5 An Alternate Method of Object Isolation

7.5.1 Introduction

One of the biggest challenges in using machine vision for agriculture is object isolation. The background for an image or video captured outdoors is seldom uniform. There are always objects and features surrounding the object of interest. For example, when capturing an image of an orchard tree, adjacent trees or trees from different rows will appear in the background. If one was taking a picture of a corn field, adjacent fields with different crops could appear in the background. In any case most of the images will have the sky or the ground in them, and these unwanted elements present unique challenges to image processing.

Humans are easily able to identify and isolate objects in an image; however, machines must be given a little more guidance. For instance, consider the car in Fig. 7.24. A simple method of isolating the car from the building in the background is to apply a color threshold filter. However, this method requires fixed RGB threshold values, and depending on the lighting (or the color of the car and building), the thresholds might need to be adjusted for each image to effectively isolate the object. Furthermore, if the car and the background are similar colors, it is even harder to distinguish the two. Notice that in the top right corner of the image there is a portion



Fig. 7.24 Car in front of a building

of gray sky. The sky color closely matches that of the car, and a simple color filter likely would not differentiate the two. For the car in Fig. 7.24, isolation would be possible by using an area or size filter of some kind. However, if the sky and car had similar pixel areas, image processing might classify the sky as a silver car. Of course, in agriculture, a corn stalk or orchard tree might be pictured instead of a car, but the concepts are the same.

7.5.2 *Spatial Mapping*

A more effective way to isolate an object from its background than using a color filter is by using spatial mapping. Spatial mapping is the process of creating a three-dimensional map of a given environment from sensor data. This sensor data most often come in the form of an image or an array of distance measurements from some arbitrary point to different points in the environment. For example, a stereo camera would produce sensor data in the form of two images and a light detection and ranging device (LIDAR) would produce an array of distance measurements [47].

Spatial mapping can be used to isolate an object from its background by using the physical geometry of the object. For instance, the car in the image is closer to the camera than the building in the background, thus if a 3D map of the image was available, a distance filter could be applied, and the car could be isolated. Furthermore, the dimensions of the car such as length, width, and height could be used to further isolate or to classify the car.

In order to use spatial mapping, a 3D map of the image must be acquired. As previously mentioned, a 3D map can be obtained from a stereo camera or by using LIDAR technology. Both technologies are useful, depending on the application.



Fig. 7.25 ZED stereo camera

However, for this section, the focus will be on the stereo camera. Figure 7.25 shows an example of a stereo camera. The camera pictured is the ZED camera designed by Stereolabs.

7.5.3 Stereo Camera Operation

Stereo cameras are devices that use two fixed RGB cameras to generate a 3D map of an image. The general concept behind stereo cameras is that objects that are close will have a large pixel shift between the two cameras, and objects that are far away will have very little pixel shift between the two cameras. In addition to the 3D map, an RGB image is obtained from a stereo camera. Normally each pixel in the RGB image will be assigned an X-Y-Z Cartesian values, and from those values, a 3D map in the form of a point cloud can be generated. Stereo cameras can perform 3D rendering very quickly, and so they are favorable in real-time applications such as robotics and machine vision [48, 49].

Stereo cameras are particularly effective at isolating trees in a fruit orchard. When photographing a tree in an orchard, the center tree (the tree of interest), as well as the tree to the left and to the right, and parts of trees in one row down are unintentionally included in the image as seen in Fig. 7.26, which shows an unfiltered picture of an apple orchard. By applying a simple distance filter, the sky and background trees can be removed from the image without any manual selecting. The filtered image can then be processed without encountering any negative effects from the background. Stereo cameras are advantageous because they do not rely on pixel color values for their object isolation. If image processing was being used to isolate the trees in Fig. 7.26, then the color thresholds might have to be adjusted to account for the

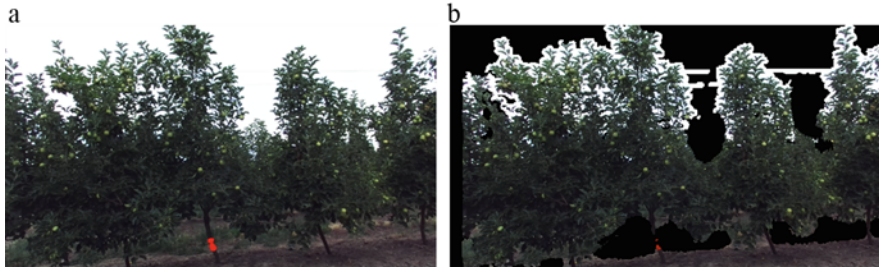


Fig. 7.26 (a) Unfiltered orchard photo. (b) Background filtered orchard photo

overcast sky or for the way the sun reflects off the leaves at different times during the day. Using a distance filter with a stereo camera, however, depends solely on the distance from the camera to the tree and operates independently of the changing light conditions.

Notice in Fig. 7.26b how the trees in the background have been filtered out leaving just the trees in the row of interest. It would have been difficult to isolate the two rows using color and area filters. This is because the two rows are the same color and the trees in the two rows blend together, making it very hard to differentiate them. However, the stereo camera provides spatial information on the location of the trees and makes differentiating between the two rows a relatively simple task.

A further advantage of the stereo camera is that it can obtain basic dimensions of the trees such as height and width. This information can be used to determine the health and canopy volume of a given tree [50]. Tree geometry is also useful when calculating a fruit yield estimate.

7.5.4 Difficulties of Using Spatial Mapping to Isolate Objects

Using stereo cameras for special mapping and object isolation presents a few unique challenges. For instance, there are intrinsic errors associated with using stereo cameras, and one should always check the device specifications and preform some basic tests to verify that the camera meets the design criteria. Mapping objects that are far away will introduce larger error than objects that are close.

In addition, stereo cameras will occasionally measure a few points that have a dramatically large error. This will create data spikes in the 3D array and if not accounted for can have a huge detrimental effect. To reduce the effect of these data spikes, it is recommended to take many snapshots or even a video of the environment and average together the data that is collected. Some stereo cameras are even able to track the location of the camera with respect to the environment. If that capability is available, it is helpful to move the camera and take measurements at different angles.

Furthermore, stereo cameras work best when imaging objects that have a lot of complexity. Complexity helps the stereo camera determine how much an object

shifts between its two RGB cameras. For example, consider using a stereo camera to measure the distance to a blank white wall. If the two RGB cameras on the stereo imager collect identical images, then the camera will determine that the wall is far away since it was not able to detect any shift between the two RGB cameras. This is a completely unreliable measurement because camera could be an inch away from the wall, and the camera would still determine that the wall is far away. Complexity is important; however, it does not have to come in the form of geometric or texture complexity. Simple color complexity will enable a stereo camera to work properly. For instance, if there were green stripes painted on the white wall, the camera would be able to detect how the stripes shifted between the two RGB cameras and an accurate 3D map of the wall could be made.

7.5.5 Object Isolation Conclusion

In this section, a novel method of object isolation was introduced. Traditional methods of object isolation use color and area filters to uncover objects of interest. The method proposed in this section uses the spatial information of an environment to differentiate between objects. This method isolates objects without relying on the color of the object. It is especially advantageous when the object of interest is a similar color to its background as occurs often in agriculture applications.

7.6 A Machine Vision for Peach Orchard Navigation

7.6.1 Introduction

The process of automating an agricultural operation such as pruning or harvesting in fruit orchards requires a platform that is able to autonomously navigate the orchard. Research on autonomous navigation for an agricultural application was started by using guides that are embedded on the ground [51, 52]. With the development of computer and sensor technologies, autonomous navigation utilized sensors such as limit switches, ultrasonic sensor, lidar, machine vision, and Global Positioning System [53]. In this section, the development of a machine vision system for autonomously navigating a peach orchard is presented. Previous research on navigation using machine vision relied on ground features to use as navigation guides. In this study, a unique approach of an upward looking camera was used to take advantage of the sky features to use as directrix for the unmanned ground vehicle.

7.6.2 Visual Feedback System for Navigation

The block diagram of the visual feedback system of the unmanned ground vehicle (UGV) seen in Fig. 7.27 has three main components: the unmanned ground vehicle platform, the vision sensor, and the controller. The input for navigation is the desired vehicle position, and the visual feedback system is used to correct the position of the vehicle. The error between the desired position and the current position is used by the controller to calculate the position of the vehicle as it moves in between the tree rows.

The visual feedback control system was an image-based position servoing system [12], where features of the image were used as control variables to estimate the vehicle's heading. The image processing of this visual feedback system did not rely on ground features but on sky features. This method is a sky-based approach [54], and the image processing is shown in Fig. 7.28. After acquiring an image, the image was cropped to remove the portion of the sky in the field of view that was closest to the camera. This was done to improve the sensitivity of the control system. It was found that slight changes in the direction of the vehicle were magnified when using the centroid of a point that was further away from the camera. Furthermore, cropping the image reduced the data needing to be processed, resulting in faster processing time and more rapid response of the ground vehicle platform. Since the green color plane provided the higher contrast between the sky and the tree canopy, the green plane was extracted to use for segmentation. A simple thresholding approach was employed to extract the path plane of the vehicle because of the high contrast between the canopy and the sky. The "salt-and-pepper" noise was removed by filtering the thresholded image. Finally, the vehicle's heading was calculated by finding the centroid of the path plane.

After the path plane was extracted, the path plane was inverted and used the position of the difference between the centroid and the set point to find the vehicle's heading and used it to drive the motor actuators, seen in Fig. 7.29. The Proportional-plus-Integral (PI) controller was used to handle the position error and used it to differentially steer the vehicle. The proportional and integral constants, K_P and K_I , of the PI controller were determined by first setting the integral gain to zero and adjusting the proportional gain until the system's response was slightly overdamped

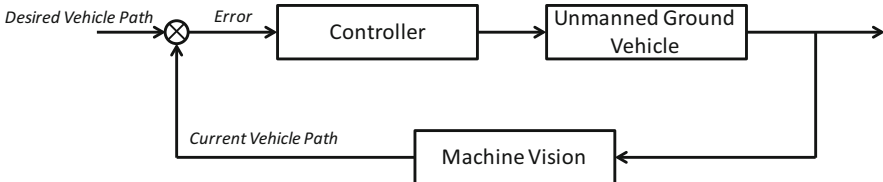


Fig. 7.27 Visual feedback system for unmanned ground vehicle navigation

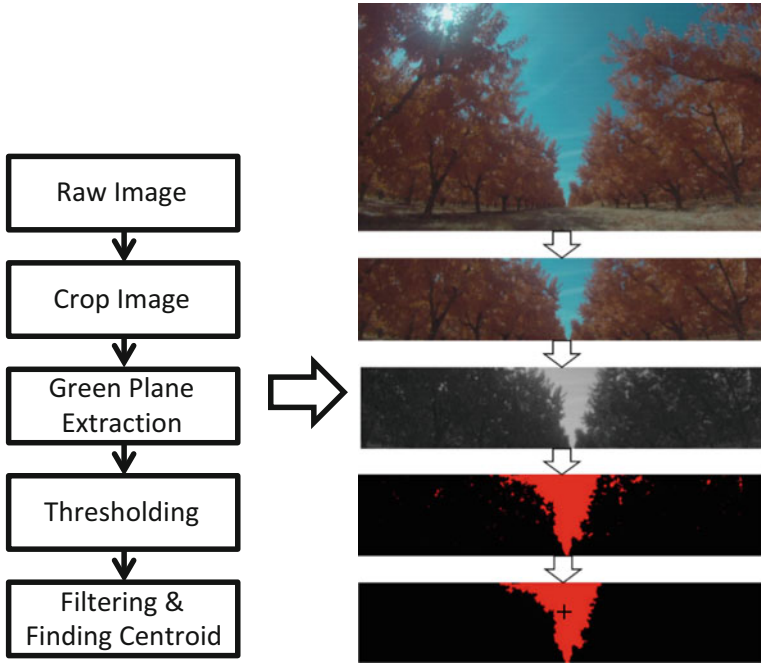


Fig. 7.28 Image processing for finding the centroid of path plane

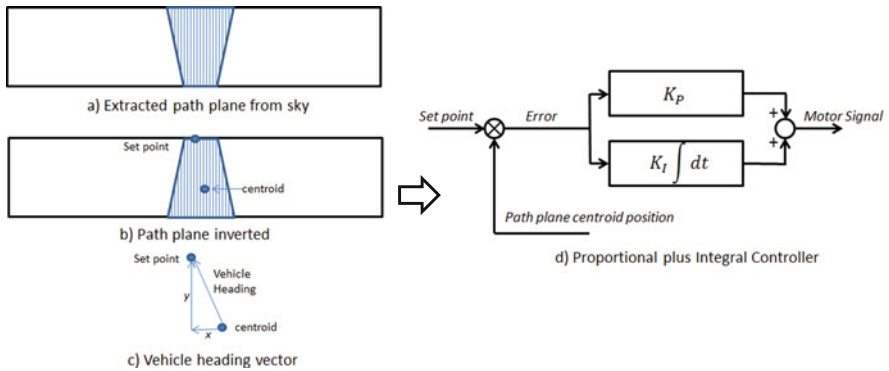


Fig. 7.29 Path plane manipulation and the PI controller for navigation

[55]. The integral gain was adjusted to remove the steady-state error. Once the PI controller had been tuned, a forward speed adjusted to 30% of the maximum value was used as the forward control signal.

7.6.3 Experimental Ground Vehicle Platform

The navigation control system was evaluated in a commercial peach orchard located in Caldwell, Idaho, USA. The orchard is well maintained, and one of the rows was randomly selected as a test row. To evaluate the performance of the visual feedback system, the distance from the vehicle to one of the tree rows was measured using an ultrasonic sensor, and cardboard boxes were positioned at a fixed distance from one row of trees. As the vehicle traveled down the row, the distance from the cardboard box was measured via the ultrasonic sensor. Ultrasonic measurements were taken over the first 27 m of travel, and visual observation of the vehicle was done as it finished the whole length of the row.

Figure 7.30 shows that the UGV deviated a maximum of 3.5 cm from its starting point over the 27 m traveled. Based on the test results, it was determined that the image processing algorithm for the vehicle guidance system was sufficient for guiding the vehicle down the orchard row.

The challenges in developing a machine vision system for outdoor application include inconsistent lighting, shadows, and color similarities in features. These difficulties were eliminated by using the sky-based approach where the image contained only the canopy and the sky, thus simplifying the segmentation process. This is a very good example of simplifying the scene constraint, the first component of the machine vision model, to aid segmentation. A simple and effective image segmentation facilitates feature detection. In addition to the test in which ultrasonic data was taken over a set distance, the vehicle was allowed to run the entire length of the row. The vehicle completed the entire row with very little error; however, it was observed that there were larger deviations from the center of the row when the vehicle approached sections where there was a break in the canopy either due to a missing tree or a tree with limited leaf growth. These breaks in the canopy caused the UGV to move away from the center of the row, but when the vehicle would move past that section, it would correct itself and return to the center. The result of a missing tree affected the shape of the path plane. This means that the shape of the path plane

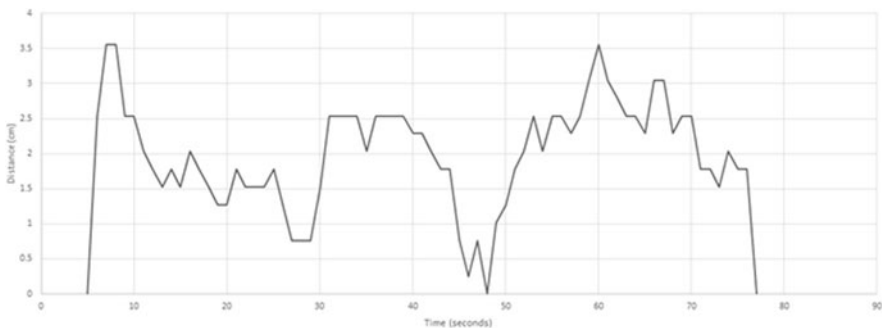


Fig. 7.30 Deviation from starting point for the peach orchard evaluation

could be used to determine a missing tree or end of the row conditions. The result of the test run in the field showed that the sky-based approach in combination with the PI controller was effective in guiding the vehicle down the row.

The sky-based approach machine vision for orchard navigation demonstrated the potential of guiding a ground vehicle in a straightline motion. However, there are some drawbacks to the sky-based approach. It is only effective when the trees have fully developed canopies. Fruit trees that have canopies year-round such as citrus will greatly benefit from this system. On the other hand, fruit trees that lose their leaves in the winter and remain dormant until the spring season will have no canopy during this season. Pruning and other orchard operations are conducted during the dormant period of the trees. To help automate these operations, the ground vehicle should rely on ground features. In this case, a ground-based image processing would be effective. Furthermore, the problems with shadows when the canopy is present can be disregarded. Therefore, for orchards that have deciduous trees, an adaptive image processing approach could be developed to deal with the changing environmental condition. For example, a sky-based image processing will be used when canopies are present, and a ground-based approach will be employed when there are no leaves. The other drawback of the proposed approach is that it only tackles the straightline motion down the row but not the end-of-the-row condition. The end-of-the-row condition could be handled in several ways. An ultrasonic sensor could be used to detect the absence of a series of trees. Another approach would be to observe the path plane of the sky-based approach. The shape of the path plane will be different at the end of the row, and this can be used to trigger the vehicle that it is at the end of the row. Future research would include dealing with a changing environment such as with canopy and without canopy conditions, detecting end-of-the-row conditions, and translating to the next row.

7.7 Conclusion

In this chapter, the different applications of machine vision in agriculture were presented. The vision applications are classified into the following groups: plant identification, process control, and machine control. Concerning plant identification, a machine vision system developed to estimate fruit yield early in the season was discussed. The developed fruit yield estimator identified and counted blossoms and correlated it with the total number of fruits on the tree. A coefficient of correlation of approximately 0.70 was obtained for both apple and peach orchards. An individual tree recognition algorithm combined with stereo-imaging was also discussed. This algorithm removed trees in the background, which could provide false positives of the blossom count. Concerning machine control, a machine vision system to navigate an unmanned ground vehicle prototype was described. The ground vehicle was able to successfully navigate an entire row of commercial peach trees autonomously. These application examples display the potential of machine vision in the field of orchard production. The future of the automation of production agriculture is very bright with machine vision as one of its tools.

References

1. Chen, Y., Chao, K., & Kim, M. S. (2002). Machine vision technology for agricultural applications. *Computers and Electronics in Agriculture*, *36*, 173–191. [https://doi.org/10.1016/s0168-1699\(02\)00100-x](https://doi.org/10.1016/s0168-1699(02)00100-x).
2. Buluswar, S. D., & Draper, B. A. (1998). Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*, *11*(2), 245–256. [https://doi.org/10.1016/s0952-1976\(97\)00079-1](https://doi.org/10.1016/s0952-1976(97)00079-1).
3. Deac, G. C., Deac, C. N., Popa, C. L., Ghinea, M., & Cotet, C. E. (2017). Machine vision in manufacturing processes and the digital twin of manufacturing architectures. *DAAAM Proceedings of the 28th International DAAAM Symposium, 2017*, 0733–0736. <https://doi.org/10.2507/28th.daaam.proceedings.103>.
4. Sharan, R. V., & Onwubolu, G. C. (2014). Automating the process of work-piece recognition and location for a pick-and-place robot in a SFMS. *International Journal of Image, Graphics and Signal Processing*, *6*(4), 9–17. <https://doi.org/10.5815/ijigsp.2014.04.02>.
5. Emmi, L., Gonzalez-De-Soto, M., Pajares, G., & Gonzalez-De-Santos, P. (2014). Integrating sensory/actuation systems in agricultural vehicles. *Sensors*, *14*(3), 4014–4049. <https://doi.org/10.3390/s140304014>.
6. Matache, M., Persu, C., Nitu, M., & Gabriel, G. (2017). Vision system for spraying machines adaptive control. *Engineering for Rural Development—International Scientific Conference*, *24*, 358–363. <https://doi.org/10.22616/erdev2017.16.n071>.
7. Awcock, G. J., & Thomas, R. (1995). *Applied image processing*. New York: McGraw-Hill.
8. Chao, K., Park, B., Chen, Y. R., Hruschka, W. R., & Wheaton, F. W. (2000). Design of a dual-camera system for poultry carcasses inspection. *Applied Engineering in Agriculture*, *16*(5), 581–587.
9. Bulanon, D. M., & Kataoka, T. (2010). Fruit detection system and an end effector for robotic harvesting of Fuji apples', International commission of Agricultural and. *Biosystems Engineering Journal*, *12*(1), 203–210.
10. Ni, Z., & Burks, T. F. (2013). Plant or tree reconstruction based on stereo vision. In *Annual meeting of the American Society of Agricultural and Biological Engineers*, 2013.
11. Bulanon, D. M., Burks, T. F., & Alchanatis, V. (2008). Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection. *Biosystems Engineering*, *101*(2), 161–171.
12. Corke, P. (2011). Robotics, Vision and Control: Fundamental Algorithms in MATLAB. *Springer Tracts in Advanced Robotics*, *73*(6), 2011.
13. Sabzi, S., Abbaspour-Gilandeh, Y., & Javadikia, H. (2017). Machine vision system for the automatic segmentation of plants under different lighting conditions. *Biosystems Engineering*, *161*, 157–173. <https://doi.org/10.1016/j.biosystemseng.2017.06.021>.
14. Partel, V., Kakarla, S. C., & Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture*, *157*, 339–350. <https://doi.org/10.1016/j.compag.2018.12.048>.
15. Ruiz-Altisent, N., Ruiz-Garcia, L., Moreda, G. P., Lu, R., HernandezSanchez, N., Correa, E. C., et al. (2010). Sensors for product characterization and quality of specialty crops—A review. *Computers and Electronics in Agriculture*, *74*, 176–194.
16. Davies, E. R. (1997). *Machine vision: Theory, algorithms, practicalities*. New York: Academic Press.
17. Guyer, D. E., Miles, G. E., Schreiber, M. M., Mitchell, O. R., & Vanderbilt, V. C. (1986). Machine vision and image processing for plant identification. *Transactions of ASAE*, *29*(6), 1500–1507.
18. Lee, W. S., Slaughter, D. C., & Giles, D. K. (1999). Robotic weed control system for tomatoes. *Precision Agriculture*, *1*, 95–113.
19. Bulanon, D. M., Lonai, J., Skovgard, H., & Fallahi, E. (2016). Evaluation of different irrigation methods for an apple orchard using an aerial imaging system. *ISPRS International Journal of*

Geo-Information, 5, 79.

20. Lindner, L., Sergiyenko, O., Rivas-López, M., Valdez-Salas, B., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (2016, November). Machine vision system for UAV navigation. In *Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), International Conference on IEEE* (pp. 1–6).
21. Lindner, L., Sergiyenko, O., Rivas-López, M., Ivanov, M., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (2017, June). Machine vision system errors for unmanned aerial vehicle navigation. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on IEEE* (pp. 1615–1620).
22. Wu, W. Y., Wang, M. J., & Liu, C. M. (1996). Automated inspection of printed circuit boards through machine vision. *Computers in Industry*, 28(2), 103–111.
23. Miller, M. K., & Delwiche, M. J. (1989). A color vision system for peach grading. *Transactions of ASAE*, 32(4), 1484–1490.
24. Bulanon, D. M., Burks, T. F., Kim, D. G., & Ritenour, M. A. (2013). Citrus black spot detection using hyperspectral image analysis. *Agricultural Engineering International: CIGR Journal*, 15(3), 171–180.
25. Rehkugler, G. E., & Throop, J. A. (1986). Apple sorting with machine vision. *Transactions of ASAE*, 29(10), 1388–1397.
26. Costa, C., Antonucci, F., Pallottino, F., Aguzzi, J., Sun, D. W., & Menesatti, P. (2011). Shape analysis of agricultural products: A review of recent research advances and potential to computer vision. *Food and Bioprocess Technology*, 4, 673–692.
27. Morimoto, T., Takeuchi, T., Miyata, H., & Hashimoto, Y. (2000). Pattern recognition of fruit shape based on the concept of chaos and neural networks. *Computers and Electronics in Agriculture*, 26, 171–186.
28. Pallottino, F., Menesatti, C., Costa, C., Paglia, G., De Salvador, F. R., & Lolletti, D. (2010). Image analysis techniques for automated hazelnut peeling determination. *Food and Bioprocess Technology*, 3(1), 155–159.
29. Parrish, E., & Goksel, A. K. (1977). Pictorial pattern recognition applied to fruit harvesting. *Transactions of ASAE*, 20, 822–827.
30. Slaughter, D., & Harrel, R. C. (1987). Color vision in robotic fruit harvesting. *Transactions of ASAE*, 30(4), 1144–1148.
31. Bulanon, D. M., Burks, T., & Alchanatis, V. (2009). Image fusion of visible and thermal images for fruit detection. *Biosystems Engineering*, 103(1), 12–22.
32. Ehsani, R., & Karimi, D. (2010). Yield monitors for specialty crops. *Landbauforschung Völkenrode, Special Issue*, 340, 31–43.
33. Maja, J. M., Campbell, T., Neto, J. C., & Astillo, P. (2016). Predicting cotton yield of small field plots in a cotton breeding program using UAV imagery data. In *Proc SPIE 986, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*, 98660C. Retrieved May 17, 2016.
34. Tumbo, S. D., Whitney, J. D., Miller, W. M., & Wheaton, T. A. (2002). Development and testing of a citrus yield monitor. *Applied Engineering in Agriculture*, 18, 399–403.
35. Annamalai, P., & Lee, W. S. (2003). *Citrus yield mapping system using machine vision, 2003 ASAE Annual Meeting*. American Society of Agricultural and Biological Engineers.
36. Stajanko, D., Rakun, J., & Blanke, M. (2009). Modelling apple fruit yield using image analysis for fruit colour, shape and texture. *European Journal of Horticultural Science*, 74(6), 260–267.
37. Lee, W. S., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., & Li, C. (2010). Sensing technologies for precision specialty crop production. *Computers and Electronics in Agriculture*, 74, 2–33.
38. Zaman, Q. U., Swain, K. C., Schumann, A. W., & Percival, D. C. (2010). Automated, low-cost yield mapping of wild blueberry fruit. *Applied Engineering in Agriculture*, 26(2), 225–232.
39. Wang, Q., Nuske, S., Bergerman, M., & Singh, S. (2012). Automated crop yield estimation for apple orchards. In *Proceedings of the International symposium on Experimental Robotics*, June 2012, Quebec City.

40. Cheng, H., Damerow, L., Sun, Y., & Blanke, M. (2017). Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *Journal of Imaging*, 3, 6.
41. MATLAB. (2016). *MATLAB and image processing toolbox release*. Natick, MA: The MathWorks.
42. Khodaskar, H. V., & Mane, S. (2017). Human face detection & recognition using raspberry Pi. *International Journal of Advanced Engineering, Management and Science*, 1–2. <https://doi.org/10.24001/icsesd2017.50>.
43. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>.
44. Bertsekas, D. P., & Tsitsiklis, J. N. (2008). *Introduction to probability* (2nd ed.). Belmont: Athena Scientific.
45. Lay, D. C. (2003). *Linear algebra and its applications* (3rd ed.). Boston: Addison-Wesley.
46. Zhang, L., Zhang, H., Chen, Y., Dai, S., Li, X., Imou, K., Liu, Z., & Li, M. (2019). Real-time monitoring of optimum timing for harvesting fresh tea leaves based on machine vision. *International Journal of Agricultural & Biological Engineering*, 12(1), 6–9.
47. Llorens, J., Gil, E., Llop, J., & Escolà, A. (2011). Ultrasonic and LIDAR sensors for electronic canopy characterization in vineyards: Advances to improve pesticide application methods. *Sensors*, 11(2), 2177–2194. <https://doi.org/10.3390/s110202177>.
48. Geiger, A., Ziegler, J., & Stiller, C. (2011). StereoScan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*. <https://doi.org/10.1109/ivs.2011.5940405>.
49. Rovira-Más, F., Zhang, Q., & Reid, J. F. (2008). Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2), 133–143. <https://doi.org/10.1016/j.compag.2007.07.007>.
50. Lee, K., & Ehsani, R. (2008). A laser-scanning system for quantification of tree-geometric characteristics. In *2008 Providence*, Rhode Island, June 29–July 2, 2008. <https://doi.org/10.13031/2013.25003>
51. Richey, C. B. (1959). “Automatic pilot” for farm tractors. *Agricultural Engineering*, 40(2), 78–79, 93.
52. Tillet, N. D. (1991). Automatic guidance sensors for agricultural field machines: A review. *Journal of Agricultural Engineering Research*, 50, 167–187.
53. Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance in North America. *Computers and Electronics in Agriculture*, 25(1/2), 154–168.
54. Radcliffe, J., Cox, J., & Bulanon, D. M. (2018). Machine vision for orchard navigation. *Computers in Industry*, 98, 165–171. <https://doi.org/10.1016/j.compind.2018.03.008>.
55. Bolton, W. (2001). *Mechatronics, electronic control systems in mechanical and electrical engineering* (2nd ed.). Boston: Addison Wesley Longman.

Chapter 8

Stereoscopic Vision Systems in Machine Vision, Models, and Applications



Luis Roberto Ramírez-Hernández, Julio Cesar Rodríguez-Quiñonez, Moisés J. Castro-Toscano, Daniel Hernández-Balbuena, Wendy Flores-Fuentes, Moisés Rivas-López, Lars Lindner, Danilo Cáceres-Hernández, Marina Kolendovska, and Fabián N. Murrieta-Rico

Acronyms

FOV	Field of view
GPS	Global positioning system
HD	High definition
IMU	Inertial measurement units
OC	Opposite cathetus
SHM	Structural health monitoring

L. R. Ramírez-Hernández · J. C. Rodríguez-Quiñonez (✉) · M. J. Castro-Toscano
D. Hernández-Balbuena · W. Flores-Fuentes
Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico
e-mail: luis.ramirez16@uabc.edu.mx; julio.rodriguez81@uabc.edu.mx;
moises.jesus.castro.toscano1@uabc.edu.mx; dhernan@uabc.edu.mx; flores.wendy@uabc.edu.mx

M. Rivas-López · L. Lindner
Instituto de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico
e-mail: mrivas@uabc.edu.mx; lindner.lars@uabc.edu.mx

D. Cáceres-Hernández
Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá, Panamá, Republic of Panama
e-mail: Danilo.caceres@utp.ac.pa

M. Kolendovska
National University of Radioelectronics, Kharkiv, Ukraine
e-mail: marina.kolendovska@nure.ua

F. N. Murrieta-Rico
Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico
e-mail: fabian.murrieta@uabc.edu.mx

SLAM	Simultaneous localization and mapping
SVM	Support vector machine
SVS	Stereoscopic vision systems

8.1 Introduction

Nowadays, there are several methods for digital scene reconstruction to perform accurate measurements of surface geometry, object detection, and other application in the area of research and engineering. Most of these methods can be identified as active or passive methods, where active methods act externally over the scene and passive methods do not act over the scene [1]. One example of the active methods is the flight time rangefinders; based on a laser or ultrasound, it finds the distance of the objects by measuring the return time of the wave. An example of the passive methods is the vision systems, which are used in applications that require to analyze specific objectives in particular scenes, where the analysis objects can leave the scene or not, for example, pedestrian detection and traffic control, among others. Vision systems have been used on different areas such as robot vision, automatic navigation, object recognition, material inspection, structural health monitoring, and especially in 3D coordinates measuring, which has gained considerable attention over the last decades [2].

In 3D coordinate measurement applications, different technologies are used such as inertial measurement units (IMU), which use inertial sensors (accelerometers and gyroscopes) to calculate 3D coordinates from mobile objects [3, 4], dynamics laser scanners, which applies the principle of dynamic triangulation [5–7], and multi-camera technique, where the images are analyzed and processed to get the shape of objects, which is determined by the distance and angles of the objects with respect to the vision system [8]. One of the particular vision systems which constitute the study of this chapter is the stereoscopic vision systems (SVS). The SVS get three-dimensional information from multiple visual inputs (two or more cameras). This is performed through the search of similarities in the digital images and determining corresponding points or homologous objects between scenes [9].

The process of SVS consists of five main steps [10]:

Images Acquisition

Consists of taking images of each camera used in a particular SVS and considering the same scene in each image, and the images will have a slight difference of the scene due to the relative displacement between the cameras.

System Geometry

Consists of the SVS design determined by the number of cameras and the position where they are located.

Feature Extraction

Consists of finding similarities between the images that will be used in the pattern matching. The techniques of feature extraction can be area based, where similarities are sought in the pixels intensity of the images. Another technique is based on

characteristics, where the information is extracted from isolated edge points, edge point's chains, or regions delimited by edges in the images.

Pattern Matching

It is the process by which the projection of a point in the 3D scene is localized in each image taken from different cameras of the particular SVS.

Distance Calculation

It is the process where the pattern matching information is used to find the relations of measurements between the points of the 3D scene and the two-dimensional projections of those points in the images.

This chapter provides useful information about the currently developed SVS, dividing them into two main groups: the binocular vision systems and the multivision systems. The theory, geometry, and basic equations of each system are explained, as well as the applications that these systems have had in the different areas of engineering.

8.2 Binocular Vision Systems

The depth information is an important feature in the scene reconstructions. It loses when capturing the scene through a camera due to the integration that occurs in each pixel of solid angle information and the physical limits of the pixel in the photosensitive device. This depth information can be retrieved through SVS. This section explains the SVS where the visual information is obtained by two cameras, beginning with the artificial biological vision model and continuing with other binocular vision model, often used in the literature. Binocular vision systems have been the most used SVS with several applications in the areas of research and engineering that will be shown in the binocular vision system applications section.

8.2.1 Artificial Biological Vision Model

The SVS simulates the biological stereopsis processes of depth perception derived from two eyes, normally developed by biological organisms with binocular vision. However, organisms with binocular vision have particular differences between them, mainly in the horizontal position of the eyes on the head, resulting in different perceptions of the image. An example is described in Fig. 8.1a, where the biological SVS obtain the images of the objects at different depths in each eye. If the images projected to the eyes' retinas are overlapped (Fig. 8.1a), an inverse relation of relative position is obtained, as can be appreciated in Fig. 8.1b [11].

To simulate the biological SVS, the artificial biological SVS uses two cameras that are placed in different points of view, causing changes in the system characteristics,

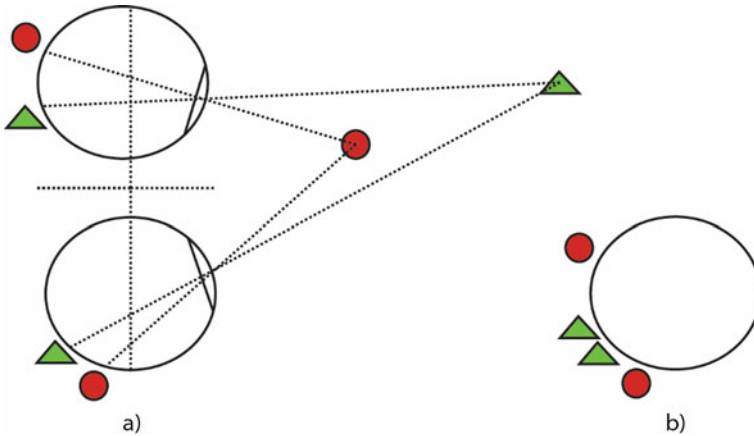


Fig. 8.1 (a) Biological SVS observing objects at different depths of a 3D scene; (b) images overlap of both eyes

for example, field of view, yield depth perception, and horizontal disparities, among others. Commonly, the artificial biological SVS use the pinhole camera geometry model. Consider an enclosure that only light coming through the pinhole can reach the image plane in such a way that the rays of light emitted or reflected by an object pass through the pinhole and form an inverted image of that object on the image plane.

This model defines the geometric relationship between a 3D point and its 2D point projection on the image plane. Figure 8.2 shows the geometry model of a pinhole camera. The geometric mapping from 3D to 2D in the pinhole camera model is called a perspective projection which consists of an image plane and the optical center point C on the focal plane. The distance between the point C and the image plane is the focal length f , and the optical axis is the line going through point C and perpendicular to the image plane [11]. The pinhole camera model is used to find the projection of the 3D coordinates in each point of the image by passing them through the optical center point C .

To analyze the geometry of the pinhole camera model with one particular point in the scene, Fig. 8.3 shows the model in a 2D plane. Imagine a real point P in the scene and its projection point Q in the image which are united by a projection line between the scene and the image plane. In this projection, it is possible to appreciate two similar triangles sharing the projection line as hypotenuse. The cathetus of the triangle on the left side of C is given by f and oc' (opposite cathetus), while on the right side of C is given by d and oc . Equation (8.1) shows the relation in the scene of the triangle on the left side of C and the triangle on the right side of C , where the negative sign represents the inverted image. The length oc depends on the distance d and the distance CP (hypotenuse), while the length oc' depends on the focal length f and the distance between the optical center C and the projection point Q .

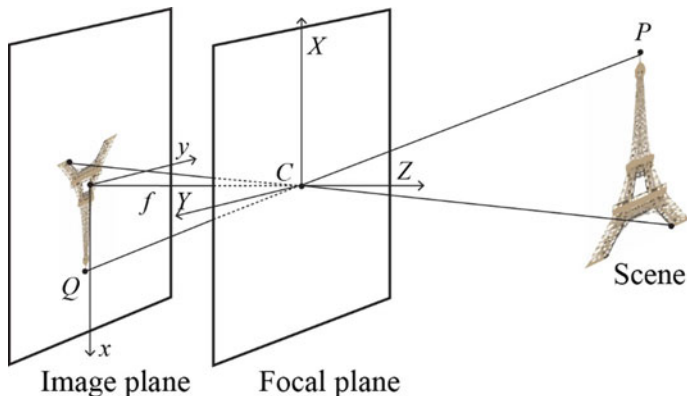
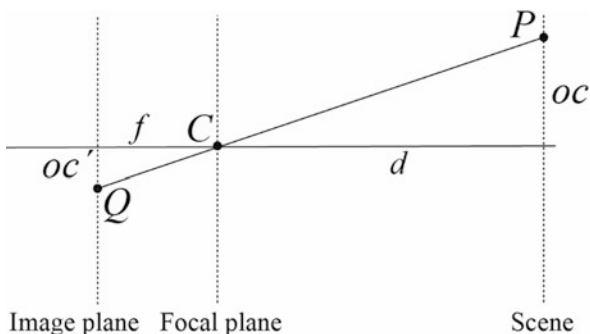


Fig. 8.2 The pinhole camera model

Fig. 8.3 Pinhole camera model in 2D perspective



$$\frac{oc}{d} = -\frac{oc'}{f} \tag{8.1}$$

Considering a homogeneous coordinate system, Fig. 8.4 shows a perspective projection of a 3D point using an ideal pinhole camera model. The optical center C is located at the origin of a 3D coordinate system with (u, v) representing the projection point coordinates of a point with coordinates (X, Y, Z) in the scene.

Equations (8.2) and (8.3) show the projection of a 3D point (X, Y, Z) onto the image plane at pixel position (u, v) [12].

$$u = \frac{Xf}{Z} \tag{8.2}$$

$$v = \frac{Yf}{Z} \tag{8.3}$$

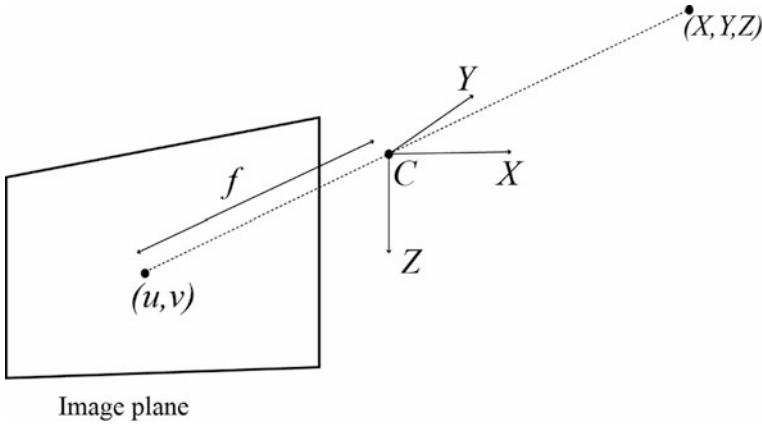


Fig. 8.4 Perspective projection of a 3D point using an ideal pinhole camera model

Using the projective geometry, the previous relation can be formulated and expressed in matrix notation as shown in Eq. (8.4).

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}_n \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{8.4}$$

where λ is the homogeneous scaling factor, and \mathbf{P}_n is the perspective projection matrix of the camera [12].

Another geometry to consider is the epipolar geometry shown in Fig. 8.5. Epipolar geometry is the intrinsic projective geometry between two images, which is independent of the scene structure and depends only on the internal parameters of the camera and its relative position. Essentially, the epipolar geometry between two images is given by the intersection of the plane of each image with a set of planes, where all the planes of this set contain a baseline that joins the optical centers of the cameras.

This geometry is used in the artificial biological SVS, and it consists of the optical centers C_1 and C_2 of the first and second cameras, respectively. Points m_1 and m_2 denote the projection of a point M on the 3D real scene, and the distance b is the baseline. Most important feature in the epipolar geometry is that the projection points m_1 and m_2 , optical centers C_1 and C_2 , and the point M are in a same plane called epipolar plane [13].

The artificial biological SVS have their optical axes parallel or convergent between them. In these systems, the displacement among cameras allows obtaining the depth of the objects through a triangulation process with visual information generated from the scene objects in each camera [14]. Figure 8.6 shows an original pair of images

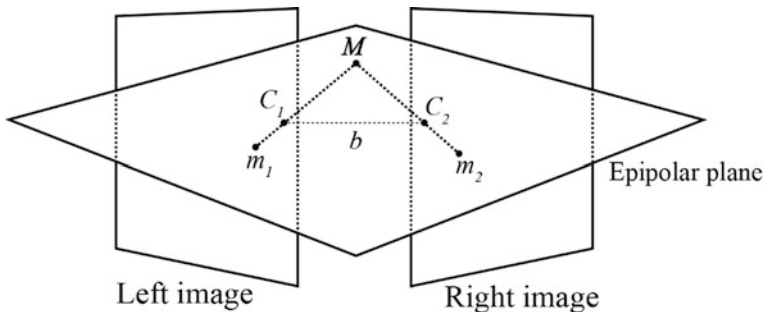


Fig. 8.5 The epipolar geometry

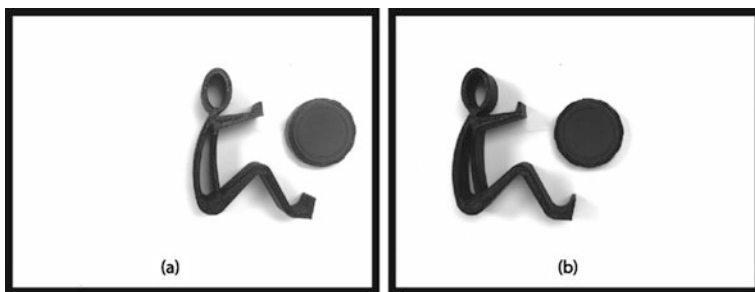


Fig. 8.6 (a) Original stereoscopic left image; (b) original stereoscopic right image

from an artificial biological SVS with two cameras aligned horizontally in which only a horizontal displacement of the objects and not vertical is appreciated.

These displacements are called disparities and are used by artificial biological SVS to find the depth of a real 3D point [15]. Figure 8.7 shows the localization of a particular 3D point using an artificial biological SVS and epipolar geometry. The coordinates \$(x_L, y_L)\$ and \$(x_R, y_R)\$ are the locations of the projection points in the left and right cameras, respectively, over the 3D point denoted by the coordinates \$(X, Y, Z)\$. The geometric model of this SVS makes possible a triangulation process for the location of the 3D point coordinates on the real scene.

Equations (8.5)–(8.7) show the relation of the angles \$B\$, \$C\$, and \$\beta\$ with the coordinates \$X\$, \$Y\$, and \$Z\$ of the 3D point which are derived from the law of sines [16].

$$X = a \frac{\sin C * \sin B}{\sin (B + C)} \tag{8.5}$$

$$Y = a \left(\frac{1}{2} - \frac{\sin C * \cos B}{\sin (B + C)} \right) \tag{8.6}$$

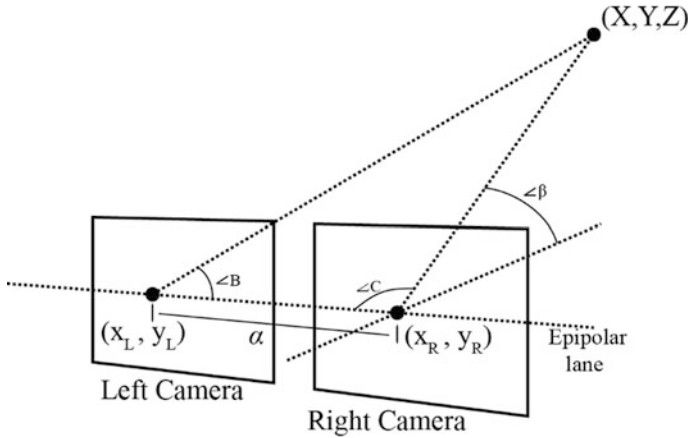


Fig. 8.7 Angles and coordinates obtained by artificial biological SVS

$$Z = a \frac{\sin C * \sin B * \tan \beta}{\sin (B + C)} \tag{8.7}$$

8.2.2 Other Binocular Vision Model

A different approach for SVS is using cameras in two different planes to obtain 3D point clouds with reference to a Cartesian space (Fig. 8.8). Consider a point in the Cartesian space given by (x, y, z) projected over a screen in an image plane. The projections will have two coordinates of the real 3D point corresponding at one camera located parallel to the plane (x, z) and a second camera located parallel to the plane (x, y) . In this way, with the first camera, the projection coordinates (y_1, y_2) and, with the second camera, the projection coordinates (y'_1, y_3) of the real 3D point are obtained, notice y_1 and y'_1 share the same projection axis [17]. The advantage of this configuration is a wide field of view avoiding occlusions in the scene. Occlusions occur in SVS when a particular region of the scene is observed in one image but not in the other [18].

With the geometry model of the camera 1, the projection coordinates (y_1, y_2) of the real 3D point is shown in Eq. (8.8).

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \alpha \lambda_1 \begin{bmatrix} -\cos(\theta_1) & \sin(\theta_1) \\ -\sin(\theta_1) & -\cos(\theta_1) \end{bmatrix} \begin{bmatrix} x - O_{c11} \\ z - O_{c13} \end{bmatrix} + \begin{bmatrix} u_{01} \\ v_{01} \end{bmatrix} \tag{8.8}$$

where α is a factor conversion of meters to pixels, λ_1 is the focal length in image 1, θ_1 is the angle rotation of camera 1, and the vector $[u_{01} \ v_{01}]$ is the shift in the image of the optical center of camera 1. Therefore, $\alpha \lambda_1$ is calculated using Eq. (8.9).

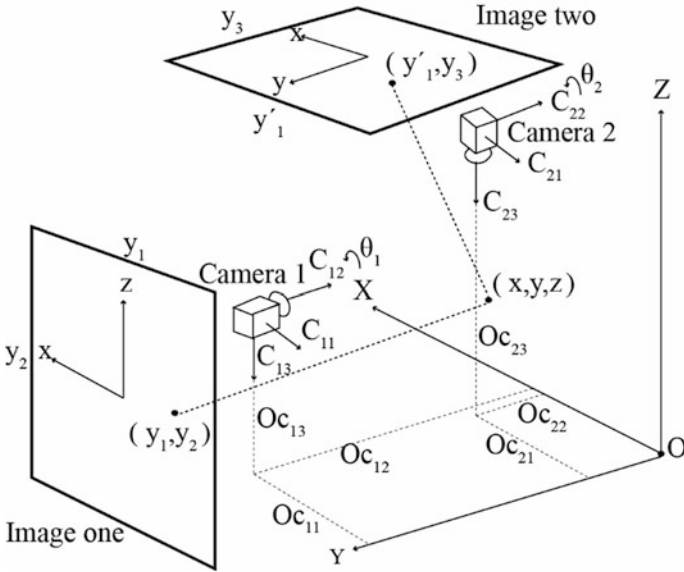


Fig. 8.8 Binocular vision model with two cameras in different planes

$$\alpha\lambda_1 = \frac{\alpha_1\lambda_1}{O_{C12} - y - \lambda_1} \tag{8.9}$$

The projection coordinates (y'_1, y_3) of the real 3D point with respect to the geometry model of camera 2 is shown in Eq. (8.10).

$$\begin{bmatrix} y'_1 \\ y_3 \end{bmatrix} = \alpha\lambda_2 \begin{bmatrix} -\cos(\theta_2) & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x - O_{c21} \\ y - O_{c23} \end{bmatrix} + \begin{bmatrix} v_{02} \\ u_{02} \end{bmatrix} \tag{8.10}$$

where λ_2 is the focal length in image 1, θ_2 is the rotation angle of camera 2, and the vector $[u_{02} \ v_{02}]$ is the shift in the image of the optical center of camera 2. Therefore, $\alpha\lambda_2$ is calculated using Eq. (8.11).

$$\alpha\lambda_2 = \frac{\alpha_2\lambda_2}{O_{C23} - z - \lambda_2} \tag{8.11}$$

Combining Eqs. (8.8) and (8.10), Eqs. (8.12)–(8.14) are obtained to describe the geometry model of the two cameras and the relation of the projection coordinates with the coordinates of the real 3D point [19].

$$y_1 = -\alpha\lambda_1 \cos(\theta_1)(x - O_{c11}) + \alpha\lambda_1 \text{sen}(\theta_1)(z - O_{c13}) + u_{01} \tag{8.12}$$

$$y_2 = -\alpha\lambda_1 \text{sen}(\theta_1)(x - O_{c11}) + \alpha\lambda_1 \cos(\theta_1)(z - O_{c13}) + v_{01} \quad (8.13)$$

$$y_3 = -\alpha\lambda_2(y - O_{c22})u_{02} \quad (8.14)$$

where, λ_1 , λ_2 , and λ_3 represent the coordinates x , z , and y in the image planes, respectively.

8.3 Multivision Systems

This section explains the SVS in which the visual information is obtained from three or more cameras. In the literature, a great number of multivision models can be found; this is due to the number of geometrical configurations. This section divides the multivision systems into two groups: trinocular vision models that consist of SVS of three cameras and multivision systems that consist of SVS with more than three cameras.

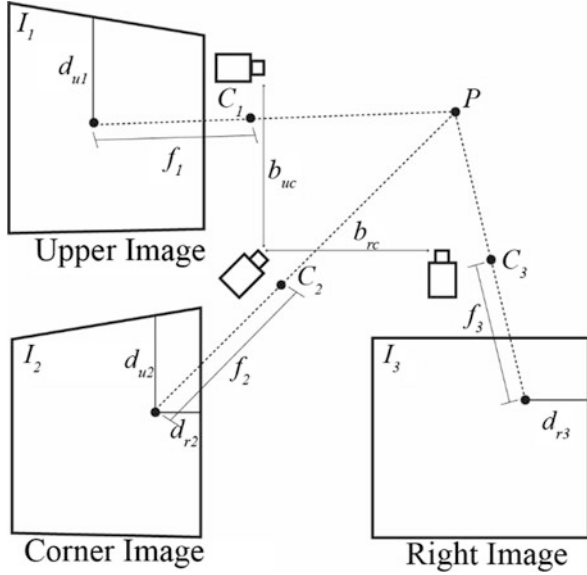
8.3.1 Trinocular Vision Models

Trinocular vision models are SVS that use three cameras to obtain the visual information of a real scene. Unlike binocular SVS, trinocular SVS makes the correspondence problem easier to solve and overcome the limitations of the binocular systems in the pattern matching due to the acquisition of more visual information by the use of the third camera [20]. This section shows trinocular vision models which are divided into five groups: right triangular, parallel, surrounding, divergent, and arbitrary.

8.3.1.1 Right Triangular Model

The right triangular model sets three cameras on the vertices of a right angle triangle. This ensures that the epipolar lines for the center image are perpendicular to the right and the top cameras [21]. Figure 8.9 shows the right triangular vision model, where C_1 , C_2 , and C_3 are the optical center of the upper, corner, and right cameras, respectively. I_1 , I_2 , and I_3 are the projection points of a real 3D point P . f_1 , f_2 , and f_3 are the focal length of the cameras. The projection of each point P in the corner image I_2 has disparities d_r and d_u with respect to the right and the upper cameras, where d_r is the difference between d_{r3} and d_{r2} . Otherwise, d_u is the difference between d_{u1} and d_{u2} . The baseline distances b_{uc} and b_{rc} of the upper-corner and the right-corner cameras, respectively, are equal.

Fig. 8.9 Right triangular vision model



In this configuration, a 3D point of the scene produces three pairs of homologous epipolar lines. The main features of the right triangular vision model are its flexibility to allow arbitrary positions of three different cameras and the reliability due to the use of a third camera, which reinforces the geometric constraints, reducing the influences of heuristics in the matching process, improving system accuracy [22].

8.3.1.2 Parallel Model

Parallel trinocular vision model considers three cameras with a distance between them, and their optical axes are parallel to the X -axis of a spatial reference system [23]. Figure 8.10 shows the parallel trinocular vision model used for the localization of a 3D point in a real scene. Distances h_1 and h_2 are the gaps of the cameras. The coordinates (x_l, y_l) , (x_m, y_m) , and (x_r, y_r) are the projection points in the left, middle, and right image planes, respectively. The variable f is the focal length of the cameras, and P is a particular 3D point in the real scene. Since the cameras are arranged in a parallel trinocular configuration, the three images lie in the same XZ plane, and the y -coordinate values of corresponding projection points on each image plane are identical.

Equations (8.15)–(8.18) show the relation of any point P with coordinates (X, Y, Z) on the left, middle, and right image planes.

$$x_m - x_l = \frac{h_l f}{Z} \tag{8.15}$$

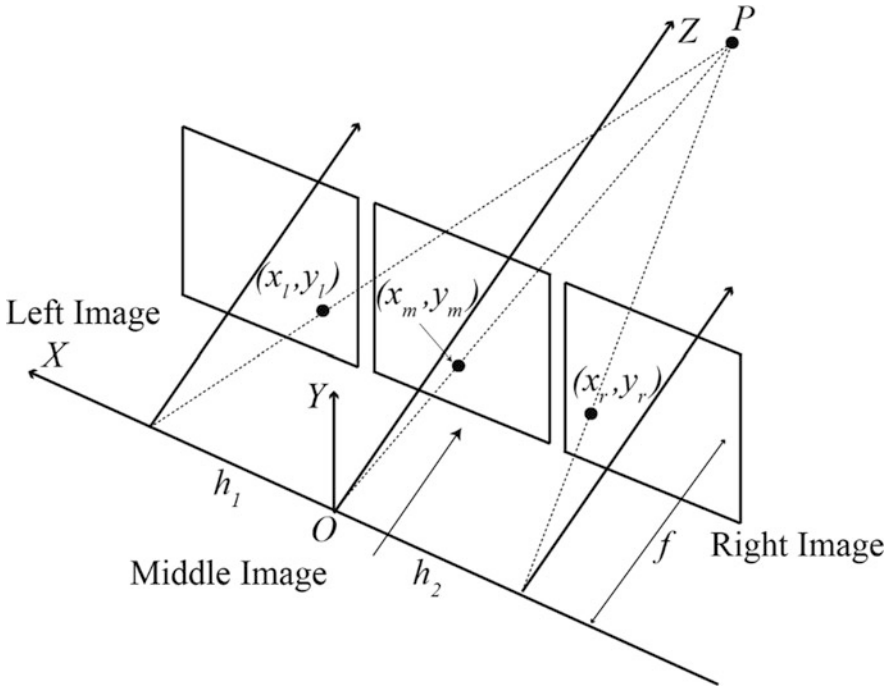


Fig. 8.10 Parallel trinocular vision model

$$x_r - x_m = \frac{h_2 f}{Z} \quad (8.16)$$

$$x_r - x_l = \frac{h_3 f}{Z} \quad (8.17)$$

$$y_l = y_m = y_r = y \quad (8.18)$$

where the distance h_3 is equal to the sum of the distances h_1 and h_2 . Z is the depth of the 3D point [24].

8.3.1.3 Surrounding Model

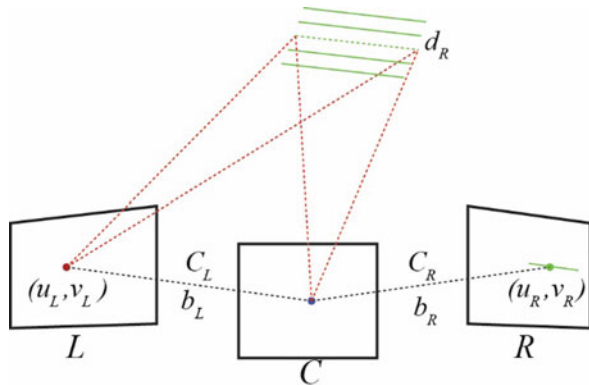
Surrounding trinocular vision model consists of a sequence of three cameras that surrounds an object to be modeled. To generate dense accurate depth maps of the scene, this model is divided into two independent pairs, using a left and center cameras like pair 1 and the center and right cameras like pair 2, where each pair of cameras have their epipolar lines parallel. Figure 8.11 shows the surrounding trinocular vision model. Pair 1 is denoted by L and C_L representing the left and center cameras, respectively, and pair 2 is denoted by R_L and R representing the center and right cameras, respectively. The variable b_L is the baseline between the left and the center cameras, and b_R is the baseline between the center and the left cameras. The coordinates (u_L, v_L) and (u_R, v_R) are the projection points in the left and right cameras. A disparity d_L in the left camera pair and a disparity d_R in the right camera pair can be projected into the C and C_L images. The objective of the model is to compute the corresponding $[u_R, v_R, d_R]$ for each $[u_L, v_L, d_L]$ concerning at the points of projection and disparities of each camera pairs [25].

8.3.1.4 Divergent Model

This camera system consists of three cameras, where the edge cameras have divergent optical axes and wide-angle lenses that provide an overview of the scene. The center camera has a long focus lens and is used to obtain more detailed information in the central field of view. Figure 8.12 shows the divergent trinocular vision model. FOV_L , FOV_C , and FOV_R denote the field of vision in the left, center, and right cameras, respectively. FOV_{LCR} is the overlapping area seen by the three cameras and where the stereo triangulation is possible [26].

This model improves the accuracy of the 3D point localization in the scene because features used for triangulation are closer in the image border than in the center. Pixels in the image border achieve a smaller sector in the field of view which improves the measurements and calculation [27].

Fig. 8.11 Surrounding trinocular vision model



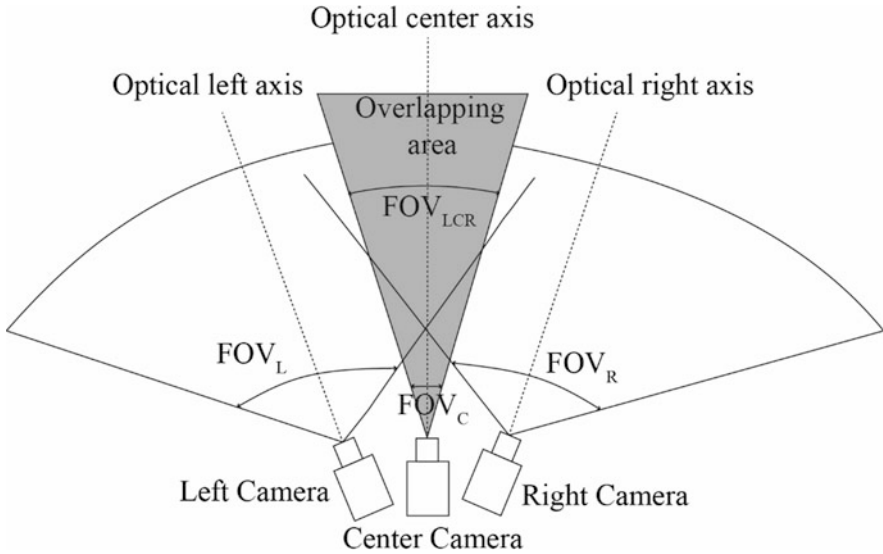


Fig. 8.12 Divergent trinocular vision model

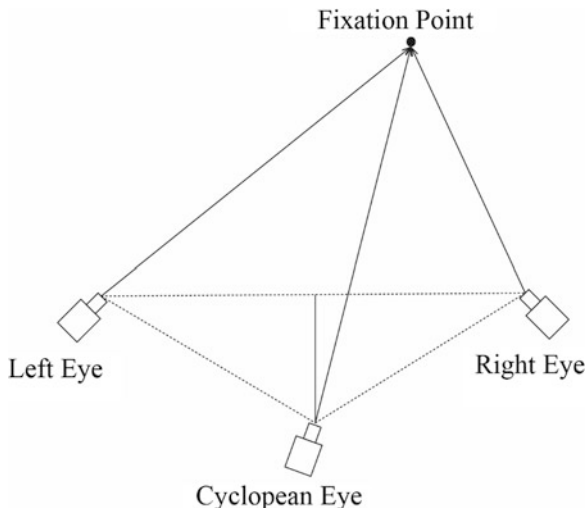
8.3.1.5 Arbitrary Model

Trinocular vision systems seen above have a defined geometry in comparison with the arbitrary trinocular vision model. The geometry of these models is specific for each particular purpose. In [28], an arbitrary trinocular vision model called CardEye system is used to mimic the functionality of the human vision system without being restricted to its components. The geometry model of the CardEye system is shown in Fig. 8.13. The system uses an active lighting device to assist in surface reconstruction process and employs techniques to improve the machine perception. The three cameras are coupled together to perform the same motion, to fixate to a point, or to change the baseline of the cameras. Applications using arbitrary trinocular models are explained in the “Trinocular SVS Applications” section.

8.3.2 Multi-Camera Models

Unlike the binocular or trinocular SVS, the multivision systems with four or more cameras can be set in a configuration that maximizes the field of view. To visualize all the environment means that more image features can be detected and tracked. The metric scale in the multi-camera system can be directly obtained from the epipolar geometry, where there is a great flexibility in this configuration because an overlapping field of view is not necessary to retrieve the metric scale [29].

Fig. 8.13 CardEye system model



The main difference between the multi-camera system and the standard pinhole camera is the absence of a single center of projection, so the light rays that pass through these systems have a non-central projection. Vision systems with non-central projection cameras allow a greater freedom in system design because they eliminate the challenging task of constructing a system of cameras which share a nodal point. Figure 8.14 shows a multivision system that covers a panoramic environment of 360°.

For a scene point P_{Fc} with coordinates $[X\ Y\ Z]^T$ in reference to frame F_c , its projection can be transformed into another reference frame F_o with magnitude M , as shown in Eq. (8.19).

$$P_{F0} = \frac{P_{Fc}}{P_{Fc}} + [0\ 0\ M]^T \tag{8.19}$$

By projecting P_{F0} onto the normalized undistorted image plane and computing the coordinates (x_u, y_u) , the image point is obtained, as shown in Eq. (8.20).

$$[X\ Y\ Z]^T = \frac{P_{F0}}{Z_{P_{F0}}} \tag{8.20}$$

where $Z_{P_{F0}}$ is the z component of P_{F0} . Thereby, the point coordinates are projected to a panoramic environment of 360°.

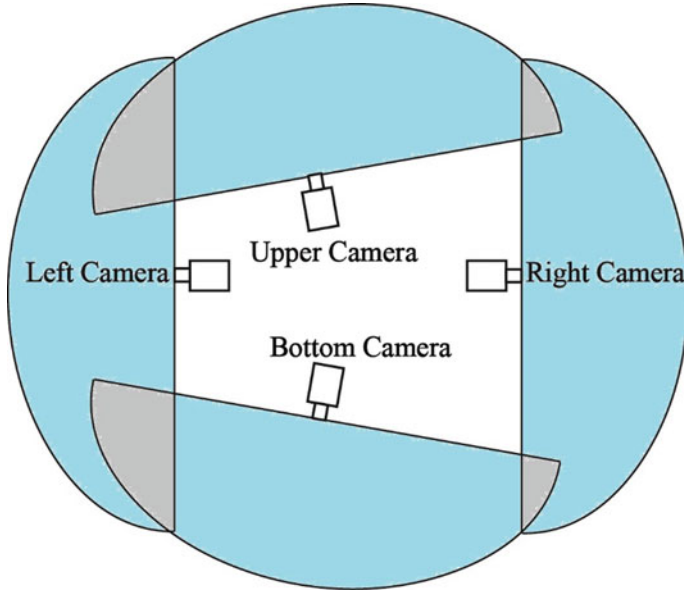


Fig. 8.14 Multivision system with 360° panoramic coverage

8.4 Applications

In this section, some applications of the SVS mentioned in this chapter are presented. For each application, the research area, the researchers who participated, and a brief description of how they implemented the SVS are mentioned.

8.4.1 Binocular Vision System Applications

8.4.1.1 Artificial Biological SVS Applications

Currently, applications in automatic navigation use the binocular SVS to solve different problems. The researchers Emiliano Statello et al. realized a method for real-time visual odometry, using stereo images combined with information obtained by a GPS system, performing localization in a geo-referenced map [30]. Also, the researchers Monica Ballesta et al. created a multi-robot map alignment in visual simultaneous localization and mapping (SLAM). The researchers use SVS in multiple robots, and each robot builds its own map of the scene, then the data maps are merged into a global one to obtain a wide information of the environment [31].

In the area of artificial vision for robotics applications, the researchers Kwang Ho Park et al. realized a multi-range approach of stereo vision that is useful in

uncertain environments where the detection of obstacles is an essential function of visual-based mobile robot [32]. The researchers Calixia Cai, Nikhil Somani, and Alois Knoll present an approach to control a six degrees of freedom manipulator using an active SVS that addresses the challenges of choosing proper image features for object detection and perform the tracking process [33].

In 3D measuring, the researchers J. C. Rodríguez Quiñonez et al. presented a 3D distance measurement accuracy improvement for SVS using optimization methods. They extracted 3D information from multiple 2D images, performing a pattern match to find the corresponding points between the images taken by the left and right cameras. The corresponding point position was used to perform a triangulation process to obtain the real 3D coordinates, and with neural networks, the measurement accuracy of the 3D coordinates is improved [16].

In computer vision, the researchers U. Castellani et al. presented a new method using SVS for 3D face recognition, where the 3D face image was modeled using multilevel B-splines coefficients and was used subsequently for a classification system with support vector machine [34].

In object recognition, the researchers Ling Cai et al. presented a novel method for SVS for multi-object detection and object tracking for surveillance application. The model solves problems of illumination variations, shadow interference, and object occlusion, locates objects by the foreground volume projected on the ground plane, and searches the density space of projected points [35].

In manufacturing process, the researchers Sotiris Malassiotis and Michael G. Stryntzis presented an SVS for the high-accuracy 3D measurement of holes on the surface of industrial components. This application is useful for inline quality inspection in assembly plant [36].

In structural health monitoring, SVS is used to measure 3D structural displacement, as is the case of the researchers Kim et al. who monitor vibration measurement in the civil engineering constructions [37]. In the same area, researchers P.F. Luo and F.C. Huang presented an SVS to evaluate the fracture parameters of materials subjected to stress [38].

In agriculture, the researchers Francisco Rovira-Más, Qin Zhang, and John F. Reid described a method to create 3D terrain maps by combining the information captured with an SVS, a localization sensor, and an inertial measurement unit, all installed on a mobile equipment platform. The perception engine comprises a compact stereo camera that captures field scenes and generates 3D point clouds, which are transformed to geodetic coordinates and assembled in a global field map [39].

In automatic navigation, the researchers Zhang Shuo et al. presented a tractor path tracking control using binocular SVS. The system controls a tractor which realizes crop recognition and path planning [40].

For microsurgeries, the researchers S. Rodríguez Palma, B. C. Becker, and C. N. Riviere implemented a simultaneous calibration of stereo vision and 3D optical tracker for robotic microsurgery. They applied the calibration in two cameras with approximately 2×3 mm mounted in a handheld micromanipulator with piezoelectric actuators called micron to realize the microsurgery [41].

In robotics, the researchers Walter A. Aprile et al. built a mobile robotic platform that features an active robotic head with two high-resolution cameras that produce a panoramic virtual environment and stereoscopic images [42].

In biomedical, the researchers Ki-Chul Kwon et al. developed a high-definition three-dimensional stereoscopic imaging display system for operating a microscope or experimenting on animals. The system consists of a stereoscopic camera part, an image processing device for stereoscopic video recording, and stereoscopic display [43].

For laparoscopic surgery, the researchers Xin Kang et al. developed a real-time stereoscopic augmented reality system which realizes a live laparoscopic ultrasound with stereoscopic video. The system creates the perception of true depth with an understanding of 3D spatial relationships among anatomical structures and visualizes internal structures along with a more comprehensive visualization of the operating field [44].

8.4.1.2 Other Binocular Vision Model Applications

A model with different geometry that the biological artificial SVS is used in the area of visual servoing control. In this area, there are systems that use the visual information obtained from one or several cameras to control the position or orientation of a robot with respect to an object or a set of visual characteristics according to the task that it must perform. The researchers M. A. Pérez and M. Bueno developed a 3D visual servoing control for robot manipulators. The main characteristic of the proposed model is that it does not need any dynamic model of the system, and image coordinates are employed directly for feedback, while an observer is designed for velocity estimation [17].

8.4.2 Multivision System Applications

8.4.2.1 Trinocular SVS Applications

In automatic navigation, the researchers Don Murray and James J. Little presented a vision-based mobile robot that autonomously explores its environment while is building the occupancy grid maps of the environment using trinocular stereo algorithm. The trinocular SVS developed by Don Murray and James J. Little has three identical wide-angle cameras. The camera coordinate frames are co-planar and aligned, so that the epipolar lines of the camera pairs are along the rows and columns of the images [45]. In the same area, the researcher A. Rieder creates a trinocular divergent SVS which is mounted on a pan and tilt platform over a vehicle. The SVS covers all the important parts of the environment, has a wide look-ahead range, high resolution in special areas of interest, and a good range estimation for all distances [26].

In object recognition, the researchers Yasushi Sumi et al. proposed a new method for 3D object recognition which uses SVS with three cameras. The recognition can be planar, polyhedral, or free forms, and the matching is performed by calculating candidates for the object position and orientation using local features, achieving the position and orientation by an iteration method. Trinocular stereo images are obtained from a stereo camera module which has three precisely calibrated CCD cameras. The third camera is for testing stereo correspondences and for improving the accuracy of 3D measurements using a double-baseline stereo algorithm [46].

In robot vision, the researchers Jens Christian Andersen, Nils A. Andersen, and Ole Ravn describe a vision sensor that extracts visible features of objects using a trinocular SVS for intelligent robot navigation. They focused on autonomous robot navigation in an indoor human environment, where the robot is able to find and recognize some basic types of objects. The project uses three cameras in a right angle triangular configuration to remove false correlations, pre-extracts a number of features, and stores these points for the ongoing navigation [47].

In computer vision, the object location and object tracking are major issues; therefore, the trinocular SVS has been used to solve these problems. The researchers Rafael Garcia, Joan Batlle, and Joaquim Salvi considered a trinocular SVS to estimate both the position and velocity of known objects with no use of the image-plane coordinates of the object's features. A position measurement tool uses the apparent area captured by every camera to locate the object, and finally, a prediction tool improves the estimation in locating the object [48].

In robotics, the researchers Elsayed Hemayed, Moumen Ahmed, and Aly Farag design an arbitrary trinocular vision model called CardEye which is used to control a robotic arm. The system has the basic mechanical properties of active vision platforms: pan, tilt, roll, focus, zoom, aperture, vergence, and baseline [28].

In metrology, the researchers A. Blake et al. designed a trinocular SVS for mechanical scanning and rapid image capture. The advantage of this SVS is to minimize the incidence of false matching, predict image measurement tolerances, and maintain the elimination of ambiguity [49].

In human motion analysis, the researchers Shoichiro Iwasawa et al. proposed a new real-time method for estimating human postures in 3D using trinocular SVS. They estimated the major joint positions based on a genetic algorithm by learning procedure [50].

8.4.2.2 Multivision SVS Applications

In autonomous navigation, the researchers Christian Häne et al. implemented a multi-camera system to cover the full 360° field of view around the car. They adapted standard vision pipelines for 3D mapping, visual localization, and obstacle detection to take full advantage of the availability of multiple cameras. The vision system is able to precisely calibrate multi-camera systems, build sparse 3D maps for visual navigation, visually localize the car with respect to these maps, generate accurate

dense maps, as well as detect obstacles based on real-time depth map extraction [29]. Other applications on autonomous navigation are found in [51–55].

In computer vision, the researchers Andreas Geiger et al. presented a toolbox with a web interface for automatic camera-to-range calibration. The system is easy to set up and recovers intrinsic and extrinsic camera parameters as well as the transformation between cameras and range sensors [56]. In the same area, the researchers Megumu Ueda et al. proposed a system generating free viewpoint video using multiple cameras and a PC cluster in real time. The system reconstructs a shape model of objects by the visual cone intersection method, transforms the shape model represented in terms of a voxel form into a triangular patch form, and finally displays the shape-color model from the virtual viewpoint directed by a user [57].

In 3D reconstruction, the researchers Kensuke Hisatomi et al. realized a method of 3D reconstruction using graph cuts to preserve the intangible cultural heritage. They used a system of 24 HD cameras to capture multi-videos, and then they performed a 3D reconstruction method based on the graph-cut that uses photo consistency with constraints by silhouette edges and the core obtained by the adaptive erosion process on the volume intersection [58].

In object recognition, the researchers Kuan-Wen Chen et al. proposed an adaptative learning method for target tracking across multiple cameras. The method performs target tracking across multiple cameras while also considering the environment changes, such as sudden lighting changes. Also, the method improves the estimation of spatio-temporal relationships by using the prior knowledge of camera network topology [59].

8.5 Conclusion

There are many SVS that have been developed at the moment, all with the objective to use the visual information to model-specific features of a real 3D scene. In this chapter, different SVS models seen in the literature focusing on the basic configuration schemes and principal geometry equations have been provided. It is important to note that the geometry of the presented SVS is ideal. In these cases, factors that affect the accuracy of the measurements are not considered. To improve the accuracy of the measurement and calculation, it is necessary to perform calibration processes in which the external and internal parameters of the cameras are modeled. In the internal parameters, the focal distance and optical distortions of the cameras (radial and tangential distortion) are modeled, while in the external parameters, the position and orientation of the cameras are modeled. To know more about camera calibration methods, Refs. [60–63] are recommended for binocular SVS and [64, 65] for multivision SVS.

A fundamental problem in the SVS is how to identify and solve the occlusion relationships, so that image regions are correctly partitioned between surfaces at different depths. In [66], the authors focus on the problem of occlusion and provide a solution to avoid them through a novel modified geometric mapping technique.

Other problems that occur in the SVS are the illumination factors, image quality, and digital noise which can cause difficulties in the search of pattern matching between the images. These problems are solved using pre-processing image techniques; an example is shown in [67] where a decomposition of the image is made, separating its structure, texture, and digital noise to independently improve each one of them and finally joining them to get a corrected resulting image. As it could be appreciated, there are several advantages and disadvantages of each SVS, for example, the binocular models have a simpler geometry with respect to the trinocular or multivision systems, which leads to a simpler analysis and a lower computational processing; however, the presence of occlusions in these systems make it very difficult to locate the characteristics at a certain moment in the scene. On the other hand, trinocular and multivision models cover the problems of occlusions due to the greater amount of visual information in comparison with the binocular models.

Another advantage of multivision system is the wide field of view that can cover up to 360° of the environment. However, the used geometry leads to a more exhaustive analysis and high levels of processing. As seen above, SVS have been included in many applications, solving problems in a practical way, making it a common topic in research and engineering areas, where the performance of these systems is continuously updated and improved.

Acknowledgments This work was partially supported by the Universidad Autónoma de Baja California, México, the Consejo Nacional de Ciencia y Tecnología (CONACYT), México, and the Secretaría Nacional de Ciencia, Tecnología e Innovación (SENACYT), Panamá.

References

1. Gonzalo, P. M., & de la CruzJesus, M. (2008). *Vision por computador. Imágenes digitales y aplicaciones* (No. 006.37 P 15100637 P 151).
2. Li, J., Zhao, H., Fu, Q., Zhang, P., & Zhou, X. (2009, June). New 3D high-accuracy optical coordinates measuring technique based on an infrared target and binocular stereo vision. In *Optical Measurement Systems for Industrial Inspection VI* (Vol. 7389, p. 738925). International Society for Optics and Photonics.
3. Castro-Toscano, M. J., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., Lindner, L., Sergiyenko, O., Rivas-Lopez, M., & Flores-Fuentes, W. (2017, June). A methodological use of inertial navigation systems for strapdown navigation task. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on* (pp. 1589–1595). IEEE.
4. Castro-Toscano, M. J., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., Rivas-Lopez, M., Sergiyenko, O., & Flores-Fuentes, W. (2018). Obtención de Trayectorias Empleando el Marco Strapdown INS/KF: Propuesta Metodológica. *Revista Iberoamericana de Automática e Informática Industrial*, 15(4), 391–403.
5. Lindner, L., Sergiyenko, O., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., et al. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 44(4), 532–541.
6. Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., et al. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*, 43(4), 360–369.

7. Real-Moreno, O., Rodríguez-Quiñonez, J. C., Sergiyenko, O., Basaca-Preciado, L. C., Hernandez-Balbuena, D., Rivas-Lopez, M., & Flores-Fuentes, W. (2017, June). Accuracy improvement in 3D laser scanner based on dynamic triangulation for autonomous navigation system. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on* (pp. 1602–1608). IEEE.
8. Rivera-Castillo, J., Flores-Fuentes, W., Rivas-López, M., Sergiyenko, O., Gonzalez-Navarro, F. F., Rodríguez-Quiñonez, J. C., et al. (2017). Experimental image and range scanner datasets fusion in shm for displacement detection. *Structural Control and Health Monitoring*, 24(10), e1967.
9. López Valles, J. M., Fernández Caballero, A., & Fernández, M. A. (2005). Conceptos y técnicas de estereovisión por computador. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 9(27), 35–62.
10. Barnard, S. T., & Fischler, M. A. (1982). Computational stereo. *ACM Computing Surveys*, 14(4), 553–572.
11. Hernández, J. M., Sanz, G. P., & Guijarro, M. (2011). *Técnicas de procesamiento de imágenes estereoscópicas*. Revista del CES Felipe II.
12. Xu, G., & Zhang, Z. (2013). *Epipolar geometry in stereo, motion and object recognition: A unified approach* (Vol. 6). Springer.
13. Zhang, Z., Deriche, R., Faugeras, O., & Luong, Q. T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2), 87–119.
14. Quiroga, E. A. C., Martín, L. Y. M., & Caycedo, A. U. (2015). La estereoscopia, métodos y aplicaciones en diferentes áreas del conocimiento. *Revista Científica General José María Córdova*, 13(16), 201–219.
15. Carabias, D. M., Garcia, R. R., & Salor, J. A. R. (2010). Sistema de Visión Estereoscópica para Navegación Autónoma de vehículos no tripulados.
16. Rodríguez-Quiñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
17. Pérez, M. A., & López, M. (2015). 3D visual servoing control for robot manipulators without parametric identification. *IEEE Latin America Transactions*, 13(3), 569–577.
18. Anderson, B. L. (1999). Stereoscopic occlusion and the aperture problem for motion: a new solution. *Vision Research*, 39(7), 1273–1284.
19. López, M. B., Pérez, M. A., & Leite, A. C. (2013). Modelado de sistemas de visión en 2D y 3D: Un enfoque hacia el control de robots manipuladores. *Tecnura: Tecnología y Cultura Afirmando el Conocimiento*, 17(37), 12–21.
20. Gurewitz, E., Dinstein, I., & Sarusi, B. (1986). More on the benefit of a third eye for machine stereo perception. In *Proceedings of the 8th International Conference on Pattern Recognition, Paris, France* (pp. 966–968).
21. Agrawal, M., & Davis, L. S. (2002). Trinocular stereo using shortest paths and the ordering constraint. *International Journal of Computer Vision*, 47(1–3), 43–50.
22. Ayache, N., & Lustman, F. (1991). Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1).
23. Ohya, A., Miyazaki, Y., & Yuta, S. I. (2001). Autonomous navigation of mobile robot based on teaching and playback using trinocular vision. In *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE* (Vol. 1, pp. 398–403). IEEE.
24. Cheng, C. C., & Lin, G. L. (2008). Acquisition of translational motion by the parallel trinocular. *Information Sciences*, 178(1), 137–151.
25. Mulligan, J., & Kaniilidis, K. (2000). Trinocular stereo for non-parallel configurations. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (Vol. 1, pp. 567–570). IEEE.
26. Rieder, A. (1996, August). Trinocular divergent stereo vision. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on* (Vol. 1, pp. 859–863). IEEE.

27. Fang, H., & Nurre, J. H. (1993, April). Analysis of three-dimensional point position for skewed-axes stereo vision systems. In *Vision geometry* (Vol. 1832, pp. 256–266). International Society for Optics and Photonics.
28. Hemayed, E. E., Ahmed, M. T., & Farag, A. A. (2001, July). The CardEye: A trinocular active vision system. In *International conference on computer vision systems* (pp. 157–173). Berlin, Heidelberg: Springer.
29. Häne, C., Heng, L., Lee, G. H., Fraundorfer, F., Furgale, P., Sattler, T., & Pollefeys, M. (2017). 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68, 14–27.
30. Statello, E., Verrastro, R., Robino, B., Gomez, J. C., & Tapino, S. (2016). Navegación por Visión Estereoscópica Asistida por GPS. In *IEEE Argencon 2016 Congreso Bienal de IEEE Argentina*. Buenos Aires, Argentina: Universidad Tecnológica Nacional Facultad Regional de Buenos Aires.
31. Ballesta, M., Gil, A., Reinoso, O., Juliá, M., & Jiménez, L. M. (2010). Multi-robot map alignment in visual SLAM. *WSEAS Transactions on Systems*, 9(2), 213–222.
32. Park, K. H., Kim, H. O., Baek, M. Y., & Kee, C. D. (2003). Multi-range approach of stereo vision for mobile robot navigation in uncertain environments. *KSME International Journal*, 17(10), 1411.
33. Cai, C., Somani, N., & Knoll, A. (2016). Orthogonal image features for visual servoing of a 6-DOF manipulator with uncalibrated stereo cameras. *IEEE Transactions on Robotics*, 32(2), 452–461.
34. Castellani, U., Bicego, M., Iacono, G., & Murino, V. (2005). 3D face recognition using stereoscopic vision. In *Advanced studies in biometrics* (pp. 126–137). Berlin, Heidelberg: Springer.
35. Cai, L., He, L., Xu, Y., Zhao, Y., & Yang, X. (2010). Multi-object detection and tracking by stereo vision. *Pattern Recognition*, 43(12), 4028–4041.
36. Malassiotis, S., & Srinivas, M. G. (2003). Stereo vision system for precision dimensional inspection of 3D holes. *Machine Vision and Applications*, 15(2), 101–113.
37. Kim, S. C., Kim, H. K., Lee, C. G., & Kim, S. B. (2006, October). A vision system for identifying structural vibration in civil engineering constructions. In *SICE-ICASE, 2006. International Joint Conference* (pp. 5813–5818). IEEE.
38. Luo, P. F., & Huang, F. C. (2000). Application of stereo vision to the study of mixed-mode crack-tip deformations. *Optics and Lasers in Engineering*, 33(5), 349–368.
39. Rovira-Más, F., Zhang, Q., & Reid, J. F. (2008). Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2), 133–143.
40. Zhang, S., Wang, Y., Zhu, Z., Li, Z., Du, Y., & Mao, E. (2018). Tractor path tracking control based on binocular vision. *Information Processing in Agriculture*, 5(4), 422–432.
41. Palma, S. R., Becker, B. C., & Riviere, C. N. (2012, March). Simultaneous calibration of stereo vision and 3D optical tracker for robotic microsurgery. In *Bioengineering Conference (NEBEC), 2012 38th Annual Northeast* (pp. 351–352). IEEE.
42. Aprile, W. A., Ruffaldi, E., Sotgiu, E., Frisoli, A., & Bergamasco, M. (2008). A dynamically reconfigurable stereoscopic/panoramic vision mobile robot head controlled from a virtual environment. *The Visual Computer*, 24(11), 941–946.
43. Kwon, K. C., Lim, Y. T., Kim, N., Yoo, K. H., Hong, J. M., & Park, G. C. (2010). High-definition 3D stereoscopic microscope display system for biomedical applications. *Journal on Image and Video Processing*, 2010, 2.
44. Kang, X., Azizian, M., Wilson, E., Wu, K., Martin, A. D., Kane, T. D., et al. (2014). Stereoscopic augmented reality for laparoscopic surgery. *Surgical Endoscopy*, 28(7), 2227–2235.
45. Murray, D., & Little, J. J. (2000). Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2), 161–171.
46. Sumi, Y., Kawai, Y., Yoshimi, T., & Tomita, F. (2002). 3D object recognition in cluttered environments by segment-based stereo vision. *International Journal of Computer Vision*, 46(1), 5–23.

47. Andersen, J. C., Andersen, N. A., & Ravn, O. (2004). Trinocular stereo vision for intelligent robot navigation. *IFAC Proceedings*, 37(8), 502–507.
48. Garcia, R., Batlle, J., & Salvi, J. (2002). A new approach to pose detection using a trinocular stereovision system. *Real-Time Imaging*, 8(2), 73–93.
49. Blake, A., McCowen, D., Lo, H. R., & Lindsey, P. J. (1993). Trinocular active range-sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5), 477–483.
50. Iwasawa, S., Ohya, J., Takahashi, K., Sakaguchi, T., Ebihara, K., & Morishima, S. (2000). Human body postures from trinocular camera images. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on* (pp. 326–331). IEEE.
51. Häne, C., Sattler, T., & Pollefeys, M. (2015, September). Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (pp. 5101–5108). IEEE.
52. Lee, G. H., Pollefeys, M., & Fraundorfer, F. (2014). Relative pose estimation for a multi-camera system with known vertical direction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 540–547).
53. Heng, L., Bürki, M., Lee, G. H., Furgale, P., Siegart, R., & Pollefeys, M. (2014, May). Infrastructure-based calibration of a multi-camera rig. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 4912–4919). IEEE.
54. Heng, L., Furgale, P., & Pollefeys, M. (2015). Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. *Journal of Field Robotics*, 32(5), 775–802.
55. Lee, G. H., Fraundorfer, F., & Pollefeys, M. (2013, November). Structureless pose-graph loop-closure with a multi-camera system on a self-driving car. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (pp. 564–571). IEEE.
56. Geiger, A., Moosmann, F., Car, Ö., & Schuster, B. (2012, May). Automatic camera and range sensor calibration using a single shot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (pp. 3936–3943). IEEE.
57. Ueda, M., Arita, D., & Taniguchi, R. I. (2004, November). Real-time free-viewpoint video generation using multiple cameras and a PC-cluster. In *Pacific-Rim Conference on Multimedia* (pp. 418–425). Berlin, Heidelberg: Springer.
58. Hisatomi, K., Tomiyama, K., Katayama, M., & Iwadata, Y. (2009, September). Method of 3D reconstruction using graph cuts, and its application to preserving intangible cultural heritage. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on* (pp. 923–930). IEEE.
59. Chen, K. W., Lai, C. C., Hung, Y. P., & Chen, C. S. (2008, June). An adaptive learning method for target tracking across multiple cameras. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1–8). IEEE.
60. Chen, S., Zuo, W., & Zheng, L. (2009, March). Camera calibration via stereo vision using tsai's method. In *Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on* (Vol. 3, pp. 273–277). IEEE.
61. Liliang, L., Ping, A., Zhuan, Z., & Zhaoyang, Z. (2011). Effective camera calibration in free-viewpoint systems. In *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2011), Shanghai, China*. IET.
62. Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22.
63. Zhao, J., Zhao, D., & Zhang, Z. (2013). Calibration and correction of lens distortion for two-dimensional digital speckle correlation measurement. *Optik-International Journal for Light and Electron Optics*, 124(23), 6042–6047.
64. Li, B., Heng, L., Koser, K., & Pollefeys, M. (2013, November). A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (pp. 1301–1307). IEEE.
65. Svoboda, T., Martinec, D., & Pajdla, T. (2005). A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & Virtual Environments*, 14(4), 407–422.

66. Priya, L., & Anand, S. (2017). Object recognition and 3D reconstruction of occluded objects using binocular stereo. *Cluster Computing*, 1–10.
67. Lim, J., Heo, M., Lee, C., & Kim, C. S. (2017). Contrast enhancement of noisy low-light images based on structure-texture-noise decomposition. *Journal of Visual Communication and Image Representation*, 45, 107–121.

Chapter 9

UKF-Based Image Filtering and 3D Reconstruction



Abdulkader Joukhadar, Dalia Kass Hanna, and Etezaz Abo Al-Izam

Acronyms

2D, 3D	Two- and three-dimensional space
4WDDMR	Four-wheeled differential drive mobile robot
AO	Absolute orientation
BA	Bundle adjustment
Bel	Degree of belief
EKF	Extended Kalman filter
EO	Exterior orientation
GRV	Gaussian random variable
IO	Interior orientation
KF	Standard Kalman filter
MAP	Maximize the posterior estimation
MLE	Maximum likelihood estimator
pdf	Probability density function
RANSAC	Random sample consensus
RO	Relative orientation
SFM	Structure from motion problem
SIFT	Scale invariant feature transform
SURF	Speeded up robust features
UKF	Unscented Kalman filter
VSLAM	Visual simultaneous localization and mapping

A. Joukhadar (✉) · D. Kass Hanna
Department of Mechatronics Engineering, Faculty of Electrical and Electronic Engineering,
University of Aleppo, Aleppo, Syria
e-mail: ajoukhadar@alepuniv.edu.sy; daliakasshanna@alepuniv.edu.sy

E. Abo Al-Izam
Department of Computer Engineering, Faculty of Electrical and Electronic Engineering,
University of Aleppo, Aleppo, Syria

9.1 Introduction

Recently, techniques for solving the problem of reconstructing a 3D model of the environment have drawn significant attention from both the robotics vision communities and computer vision. Many variants of these techniques have started to make an impact in a wide range of robotics applications such as robot navigation, obstacle avoidance, and visual simultaneous localization and mapping (VSLAM) based on stereo vision [1]. Structure from motion problem (SFM) is defined as a technique for 3D reconstruction from a set of 2D images and some geometrical constraints [2].

SFM methods perform a *bundle adjustment (BA)* optimization of the total geometry in order to obtain an accurate 3D model of the scene. However, this is computationally very expensive and cannot be implemented in a real-time application [3]. On the other hand, *simultaneous localization and mapping (SLAM)* is a problem faced recently in robotics community, essentially addressing the hard real-time mapping and navigation problem by “sequential” interactive local estimation of the structure and motion [4].

The SLAM problem tries to answer the following central question: “*Is it possible for an autonomous robot starting at an unknown location in an unknown environment to build a map of the environment while simultaneously using this map to compute the vehicle’s location?*” Solving this problem allows to develop a truly autonomous robot and navigate safely around the environment [5].

The stochastic nature of the mobile robot motion with noisy measurement data complicates the coupling between navigation and mapping that is inherent SLAM [6]. Many successful SLAM algorithms address these issues by formulating the problem in a probabilistic manner, tracking the joint posterior over the robot pose and map.

Probabilistic Bayesian filter is the strategy to incorporate uncertainty for all possible robot poses with a probability density function (pdf), which is the degree of belief (Bel) of the robot moves [7].

The estimation problem can be solved by KF-based approaches or particle filter (i.e., online VSLAM) for real-time application or bundle adjustment (i.e., standard SFM) for an offline application [7]. In the medical field, stereo vision obtained a 3D vision, which improved the accuracy of surgery and reduced the time required for surgery and errors that may occur. A novel SLAM algorithm proposed by [8] aimed at advancing the state-of-art in image-guided surgery using stochastic models and KF framework to recursively estimate the configuration of the high degree of freedom snake surgical robot using stereo vision. VSLAM estimates the camera pose by implementing *epipolar geometry* on the static feature correspondences as shown in Fig. 9.1. The dynamic features are regarded as outliers and excluded from the computation.

From Fig. 9.1, the static corner has been extracted as a correspondence point in order to estimate the camera orientation. The computer vision community has developed a large number of feature extraction techniques (e.g., Harris corner



Fig. 9.1 Feature correspondence extraction

detector, scale invariant feature transform (SIFT), and speeded-up robust features (SURF)). Unfortunately, these feature-matching techniques do not guarantee perfect correspondences, especially when the data contains outliers [3]. Implementation of robust estimators RANSAC (random sample consensus) is useful to reject outliers and handle false correspondences. On the other hand, deep learning techniques can process the image sequences directly to compute the correspondences in real time [3].

This chapter is structured as follows: Sect. 9.1 introduces the basic estimation techniques; Sect. 9.2 presents the brief of stereo vision, camera calibration, projection, and epipolar function; Sect. 9.3 exhibits the uncertainties and error source in vision system; finally, Sect. 9.4 shows examples of VSLAM based on UKF algorithm and stereo vision.

9.2 Kalman Filter Framework-Based Probabilistic Inference

The following problem commonly recurs in Computer Vision and Autonomous Robotics algorithms: estimate the values of unknown parameters (robot posture, camera orientation, etc.), given a number of measurements (sensory data, images, feature points, etc.). These kinds of problems are called *inverse problem* because they involve in estimating unknown model parameters instead of simulating the forward formation equations [9]. However, a model of uncertainty sources needs to be introduced in order to have a reasonable algorithm. Such inference problems from noisy data are called *probabilistic inference* [10]. In this section, the measurement update equation for the KF, EKF, and UKF is derived starting from the maximum

likelihood (the joint probability density function (pdf)), work through Bayesian rule, Gauss–Newton iterated nonlinear least squares method, and practical nonlinear Bayes filter (i.e., standard Kalman filter (SKF), extended Kalman filter (EKF)-based Taylor linearization, and unscented Kalman filter (UKF)-based stochastic linearization).

9.2.1 Maximum Likelihood Estimator (MLE)

The general form of noisy measurement model is given by (9.1):

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (9.1)$$

where \mathbf{z}_k is the noisy measurement vector, \mathbf{x}_k is the unknown state vector, $h(\cdot)$ is the associated nonlinear measurement model, which maps the unknown into that particular measurement, and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$ is a normal *Gaussian random variable* (GRV) with zero mean and covariance matrix \mathbf{R} . Given all of the noisy measurements $\mathbf{z} = \{\mathbf{z}_k\}$, the likelihood of having the observed $\{\mathbf{z}_k\}$ given a particular value of \mathbf{x} is given by (9.2):

$$L = p(\mathbf{z}|\mathbf{x}) = \prod_k p(\mathbf{z}_k|\mathbf{x}_k) = \prod_k p(\mathbf{z}_k|h(\mathbf{x}_k)) = \prod_k p(\mathbf{v}_k) \quad (9.2)$$

where $p(\mathbf{z}|\mathbf{x})$ is the joint probability distribution of the measurements \mathbf{z} with the unknown vector \mathbf{x} . To solve the inverse problem (if the distribution is unimodal Gaussian), the optimal estimate value for the unknown state vector \mathbf{x} in the absence of any prior model is that maximizes the likelihood function. if the distribution is unimodal Gaussian. However, if the probability is multimodal, it has several local maxima in likelihood, much more care is required [11].

The likelihood function can be written for the Gaussian noise as (9.3):

$$\begin{aligned} L &= \prod_k |2\pi\mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k))\right) \\ &= \prod_k |2\pi\mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}\|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2\right) \end{aligned} \quad (9.3)$$

The norm $\|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2$ is called the *Mahalanobis distance* [9]. It is used to measure the distance between the measurement \mathbf{z}_k with step time \mathbf{k} and the mean of Gaussian distribution $\bar{\mathbf{z}}_k$. Usually, it is more convenient to work with the negative log-likelihood, as a cost function [12] (9.4):

$$\begin{aligned} E = -\log L &= \frac{1}{2} \sum_k (\mathbf{z}_k - \bar{\mathbf{z}}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \bar{\mathbf{z}}_k) + \log |2\pi\mathbf{R}| \\ &= \frac{1}{2} \sum_k \|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2 + K \end{aligned} \quad (9.4)$$

where $K = \sum_k \log |2\pi \mathbf{R}|$ is a constant independent of \mathbf{x} and can be dropped. The inverse covariance matrix \mathbf{R}^{-1} weights each measurement error residuals (i.e., the difference between the real measurement \mathbf{z}_k and the predicted mean value $\hat{\mathbf{z}}_k = \bar{\mathbf{z}}_k$).

Another form of negative log-likelihood can be written as (9.5):

$$E = -\log L = \sum_k \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_{\mathbf{R}^{-1}} \quad (9.5)$$

Consider the measurement noise is Gaussian and the measurement equation is linear (9.6):

$$\mathbf{z}_k = h(\mathbf{x}_k) = \mathbf{H}\mathbf{x}_k \quad (9.6)$$

where \mathbf{H} is the measurement matrix; in this case, the maximum likelihood estimate is given by the minimization of the quadratic function (9.7), which is a simple quadratic form in \mathbf{x}_k solved using linear least square algorithm:

$$E = \sum_k \|\mathbf{z}_k - h(\mathbf{x}_k)\|_{\mathbf{R}^{-1}} = \sum_k (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k) \quad (9.7)$$

9.2.2 Probabilistic Inference and Bayesian Rule

In some cases, the range of possible solution consistent with the measurements is too large to be useful, and any progress cannot be made [11]. For example, MLE estimates each pixel separately based on just its noisy version to solve the problem of image filtering [13]. The difference between the Bayesian inference and the MLE method is that the starting point of Bayesian inference is to formally consider the unknown vector \mathbf{x}_k as a random vector with a prior distribution $p(\mathbf{x}_k)$, which is called the degree of belief (Bel), then the posterior distribution of \mathbf{x}_k can be computed by multiplying the measurements likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ by the prior Bel [14].

Consider the noisy measurement model given by (9.8):

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (9.8)$$

where $\mathbf{x}_k \sim N(\bar{\mathbf{x}}, \mathbf{P})$ is the unknown Gaussian random state vector with mean $\bar{\mathbf{x}}$ and state covariance matrix \mathbf{P} . The form of Bayesian rule is given by (9.9):

$$p(\mathbf{x}_k | \mathbf{z}_k) = \eta p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k) \quad (9.9)$$

where η is the normalizing constant [10]. The problem is to find \mathbf{x}_k that maximizes the posterior estimation (MAP).

Assume that the distributions of \mathbf{x}_k and \mathbf{z}_k are Gaussian (9.10):

$$\begin{aligned} p(\mathbf{z}_k | \mathbf{x}_k) &= |2\pi \mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k))\right) \\ p(\mathbf{x}_k) &= |2\pi \mathbf{P}_k|^{-1/2} \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)\right) \end{aligned} \quad (9.10)$$

The solution that maximizes $p(\mathbf{x}_k | \mathbf{z}_k)$ is the most probable value of the random vector and is equivalent to minimize its negative log, which reduces to the quadratic form [11] (9.11):

$$L = \frac{1}{2} \left((\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) + (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-) \right) \quad (9.11)$$

An algebraic equivalent way to maximize the posterior likelihood is to consider the prior estimate as a pseudo-observation and write a new observation vector [12] (9.12):

$$\mathbf{Z}_k = \begin{bmatrix} \mathbf{z}_k \\ \hat{\mathbf{x}}_k^- \end{bmatrix}, g(\mathbf{x}_k) = \begin{bmatrix} h(\mathbf{x}_k) \\ \hat{\mathbf{x}}_k^- \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_k \end{bmatrix} \quad (9.12)$$

which gives (9.13):

$$L = \frac{1}{2} (\mathbf{Z}_k - g(\mathbf{x}_k))^T \mathbf{C}^{-1} (\mathbf{Z}_k - g(\mathbf{x}_k)) \quad (9.13)$$

This is a nonlinear least squares problem of the form (9.14):

$$E = -\log L = \sum_k \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2 \quad (9.14)$$

A useful approximation for small residual problems is the online stochastic Gauss-Newton method, which defines the sequence of iterates as [12] (9.15):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- - \left(\mathbf{G}_k^T \mathbf{C}^{-1} \mathbf{G}_k \right)^{-1} \mathbf{G}_k^T \mathbf{C}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) \quad (9.15)$$

where $\mathbf{G}_k = \left. \frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}=\bar{\mathbf{x}}}$ the Jacobian of $g(\mathbf{x}_k)$ with respect to the state vector \mathbf{x}_k .

The Gauss-Newton method is simply using the *matrix inversion lemma* [11] (9.16):

$$\begin{aligned} \left(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}_k^{-1} \right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} &= \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \\ \left(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}_k^{-1} \right)^{-1} &= \mathbf{P}_k - \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \mathbf{P}_k \end{aligned} \quad (9.16)$$

Using (9.16), the Kalman equation is given by (9.17):

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k}}^- - \mathbf{K}(\mathbf{z}_{\mathbf{k}} - h(\mathbf{x}_{\mathbf{k}})) \quad (9.17)$$

where $\mathbf{K}_{\mathbf{k}}$ is the Kalman gain (9.18):

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k}}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{\mathbf{k}}\mathbf{H}^T + \mathbf{R})^{-1} \quad (9.18)$$

The update covariance matrix is approximated using (9.19):

$$\mathbf{P}_{\mathbf{k}+1} = (\mathbf{G}_{\mathbf{k}}^T\mathbf{C}^{-1}\mathbf{G}_{\mathbf{k}})^{-1} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{k}}^{-1})^{-1} \quad (9.19)$$

where the Hessian $\mathbf{H}_{\mathbf{k}} = \left. \frac{\partial h(\mathbf{x}_{\mathbf{k}})}{\partial \mathbf{x}_{\mathbf{k}}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}$ is the Jacobian of $h(\mathbf{x}_{\mathbf{k}})$ with respect to the state vector $\mathbf{x}_{\mathbf{k}}$.

The posterior covariance matrix using *matrix inversion lemma* (9.16) is given by (9.20):

$$\mathbf{P}_{\mathbf{k}+1} = (\mathbf{I} - \mathbf{K}_{\mathbf{k}}\mathbf{H})\mathbf{P}_{\mathbf{k}} \quad (9.20)$$

Many applications require an estimate for the uncertainty in this estimate such as KF, which require the computation of this uncertainty as posterior covariance matrix in order to optimally integrate new measurements with previously computed estimates [9].

9.2.3 Bayes Filter and Belief Update

In this section, the formulation of optimal recursive discrete time Bayesian filters (e.g., KF, EKF, and UKF) is presented as a practical estimator in terms of Bayes filter.

The basic elements of Bayesian filter are the initial belief $\text{Bel}(\mathbf{x}_{\mathbf{k}-1})$ containing preliminary information on the unknown vector $\mathbf{x}_{\mathbf{k}-1}$, the motion model $p(\mathbf{x}_{\mathbf{k}}|\mathbf{x}_{\mathbf{k}-1}, \mathbf{u}_{\mathbf{k}})$ as a probabilistic model of the discrete time state space, and the measurement model $p(\mathbf{z}_{\mathbf{k}}|\mathbf{x}_{\mathbf{k}})$ determining the stochastic mapping from the state vector to the measurement, where

- $\mathbf{x}_{\mathbf{k}} \in \mathfrak{N}^n$ is the unknown state space vector on time step \mathbf{k} .
- $\mathbf{u}_{\mathbf{k}} \in \mathfrak{N}^L$ is the control vector on time step \mathbf{k} .
- $\mathbf{z}_{\mathbf{k}} \in \mathfrak{N}^m$ is the observation vector on time step \mathbf{k} .

Using Markov assumption, these vectors are conditionally independent of past values. Bayes filter applies two rules successively to predict the system state [10] (9.21):

$$\begin{aligned}
\text{Prediction : } \overline{\text{Bel}}(\mathbf{x}_k) &= \sum_k p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \text{Bel}(\mathbf{x}_{k-1}) \\
\text{correction : } \text{Bel}(\mathbf{x}_k) &= \eta p(\mathbf{z}_k | \mathbf{x}_k) \overline{\text{Bel}}(\mathbf{x}_k)
\end{aligned} \tag{9.21}$$

The prior predictive $\overline{\text{Bel}}(\mathbf{x}_k)$ is calculated just before the measurement \mathbf{z}_k using the control vector \mathbf{u}_k ; this step is called prediction (or *action phase*). Next, the state estimate belief given in the action phase is corrected using sensor measurement; this step is called correction (or *perception phase*).

9.2.3.1 KF Framework

In many vision applications, the object is tracked from frame to frame as it moves. *Kalman Filter* (1960) has been regarded as the optimal solution to many visual motion tracking and data prediction tasks [13].

The standard KF derivation is given here in the practical use of probabilistic inference [15]. Consider a noisy linear system given by (9.22):

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}_1\mathbf{u}_k + \mathbf{B}_2\boldsymbol{\omega}_k \\
\mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}_k
\end{aligned} \tag{9.22}$$

where $\mathbf{x}_k, \mathbf{x}_{k-1}$ are the current and previous state vector, $\mathbf{A}_{n \times n}$ is the linear state transition matrix of the dynamic model, $\mathbf{B}_{1n \times L}$ is the control matrix, $\mathbf{B}_{2n \times L}$ is the input noise matrix, $\mathbf{H}_{n \times m}$ is the measurement model matrix, $\boldsymbol{\omega}_k \sim N(\mathbf{0}, \mathbf{Q})$ is an additive Gaussian state noise, and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$ is the Gaussian measurement noise. The KF equations can be derived as follows [16]:

1. **Prediction phase:** The motion model causes a drift in the previous estimate, while the additive noise increases the system disbelief.

First, apply the motion model and compute the joint distribution of the Gaussian state \mathbf{x}_k given the initial state $\mathbf{x}_{k-1} \sim N(\bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})$ and the input \mathbf{u}_k by (9.23):

$$\hat{\mathbf{x}}_k^- \sim N\left(\mathbf{A}\bar{\mathbf{x}}_{k-1} + \mathbf{B}\bar{\mathbf{u}}_k, \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}\right) \tag{9.23}$$

Then, apply the measurement model and compute the joint distribution of the measurement \mathbf{z}_k given the predicted state $\hat{\mathbf{x}}_k^-$ by (9.24):

$$\hat{\mathbf{z}}_k \sim N\left(\mathbf{H}\hat{\mathbf{x}}_k^-, \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right) \tag{9.24}$$

2. **Correction phase:** New measurements from the current frame introduce additional information that updates the prior estimate $\hat{\mathbf{x}}_k^-$ and restores some of the belief, by computing Kalman gain \mathbf{K}_k and updating the covariance matrix (9.25):

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T [\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}]^{-1} \\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-
\end{aligned} \tag{9.25}$$

9.2.3.2 EKF Linearization Technique

For nonlinear problems, such as SLAM, an EKF, which linearizes the motion and measurement model around the current estimate, is used. The important drawback of the EKF approach is that it uses the Taylor linearization dynamic model [17]. However, if the robot drives along a straight path, the distribution of mobile model in a plane has been observed by “Banana Shape distribution.” As uncertainty increases, the algorithm becomes inconsistent due to the normality assumption breaking down [18]. Consider a noisy nonlinear system given by (9.26):

$$\begin{aligned}
\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, \boldsymbol{\omega}_k) \\
\mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k
\end{aligned} \tag{9.26}$$

where $f(\cdot)$ is the nonlinear motion model, noisy by non-additive Gaussian noise $\boldsymbol{\omega}_k \sim N(\mathbf{0}, \mathbf{Q})$ and $h(\cdot)$ is the nonlinear measurement model, noisy by $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$.

Table 9.1 illustrates the pseudo code for EKF algorithm as Bayes filter.

9.2.3.3 UKF Stochastic Linearization Technique

The UKF is a Gaussian recursive Bayesian filtering algorithm to solve the probabilistic inference practically. It propagates and updates the system state using a set of deterministically chosen points called sigma points [11]. These points capture the mean and covariance of the state distribution. Filter each point using unscented transform through the nonlinear motion and measurement models [14], and determine the posterior state mean and state covariance to the third order of the nonlinear system. This is a form of statistical local linearization, which produces more accurate estimates than the analytic local linearization employed by the EKF [19]. The UKF algorithm includes three main stages. Table 9.2 shows the pseudo code for the UKF algorithm [16].

9.3 Stereo Vision System

Stereo vision is considered one of the most important recent applications [21]. It is still developing especially in the robotics vision application [22]. Stereo vision is used to form a 3D map of the robot environment, and landmarks are used in this map for localization and exploration [21].

Table 9.1 Extended Kalman filter ($\mathbf{x}_k - 1, \mathbf{P}_k - 1, \mathbf{u}_k, \mathbf{z}_k$)

Initialization:

Initialize the prior knowledge $\mathbf{x}_k - 1, \mathbf{P}_k - 1, \mathbf{Q}, \mathbf{R}$:

$$\mathbf{x}_{k-1} = [\mathbf{x}_0]^T$$

$$\mathbf{P}_{k-1} = \text{diag}(\mathbf{P}_0) \in \mathfrak{N}^n, \mathbf{Q} = \text{diag}(\mathbf{Q}) \in \mathfrak{N}^L, \mathbf{R} = \text{diag}(\mathbf{R}) \in \mathfrak{N}^m$$

For each sample time k , do:

Prediction:

1. Apply the motion model and compute the mean and covariance:

$$\hat{\mathbf{x}}_k^- = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k); \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q})$$

$$\mathbf{P}_k^- = \mathbf{J}_x \mathbf{P}_{k-1} \mathbf{J}_x^T + \mathbf{J}_u \mathbf{Q} \mathbf{J}_u^T$$
 where the Jacobian matrices $\mathbf{J}_x, \mathbf{J}_u$ are obtained by differentiating $f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$ with respect to \mathbf{x}_{k-1} and \mathbf{u}_k , respectively:

$$\mathbf{J}_x = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{x}_{k-1}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}, \mathbf{J}_u = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{u}_k} \right|_{\mathbf{u}=\hat{\mathbf{u}}}$$
2. Apply the measurement model:

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_k^-, \mathbf{v}_k); \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$$

Correction:

The EKF gain \mathbf{K}_k which minimizes the errors and updates the posterior mean and covariance ($\hat{\mathbf{x}}_k, \mathbf{P}_k$) is given by:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \left[\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R} \right]^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-$$

return ($\hat{\mathbf{x}}_k, \mathbf{P}_k$)

Compute the Degree of Belief (Bel):

The determinant \mathbf{P}_k provides a good measure of uncertainty [10].

$$\text{Bel}_k = 1 - \|\mathbf{P}_k\|_2^{0.5}; \quad 0 \leq \text{Bel} \leq 1$$

end function

Figure 9.2 explains a low- and high-level image processing stages. In the low-level image processing stage, a camera calibration with distortion removal is carried out. Camera calibration includes the determination of the camera's intrinsic and extrinsic parameters. Accurate estimates of this geometry are necessary in order to relate image information to an external world coordinate system. On the other hand, in the high-level image processing stage, certain correspondence points are determined based on advanced algorithms for feature extraction [23]. These points are hence used to calculate the relative orientation (RO) as well as the absolute orientation (AO) using a set of control points whose relative coordinates and corresponding image points are known. Stereo vision-based system with SLAM algorithm is used to enable the robot to percept the environment around and within the robot playing area [24]. The computation of relative camera pose can be done using 7-point correspondences for an uncalibrated camera or 5-point for a calibrated camera from two views under by enforcing *epipolar geometry*. If the image correspondences are known, the relative pose between two images can be recovered up to a scale factor [13]. When the camera pose is recovered, one can easily reconstruct 3D points of the scene by intersecting two projection ray lines through triangulation [25]. As the rays do not

Table 9.2 Unscented Kalman filter (χ_{k-1} , \mathbf{P}_{k-1} , \mathbf{u}_k , \mathbf{z}_k)**Initialization:**

Initialize the state space vector χ_0 and covariance matrix \mathbf{P}_{k-1} , \mathbf{Q} , \mathbf{R} :

$$\chi_0 = [\mathbf{x}_0 \ \mathbf{0} \ \mathbf{0}]^T$$

$$\mathbf{P}_{k-1} = \text{diag}(\mathbf{P}_0, \mathbf{Q}, \mathbf{R})$$

For each sample time \mathbf{k} do:

Prediction:

1. Compute $2n - 1$ sigma points:

$$\chi_{k-1} = \left[\mathbf{x}_{k-1} \ \mathbf{x}_{k-1} + \gamma \sqrt{\mathbf{P}_{k-1}} \ \mathbf{x}_{k-1} - \gamma \sqrt{\mathbf{P}_{k-1}} \right]^T$$

where γ is a scalar parameter that determines how far the sigma points are dispersed away from the mean, and $\sqrt{\mathbf{P}_{k-1}}$ is computed using *Cholesky decomposition* [20].

2. Apply the motion model:

$$\chi_{\mathbf{k}} = f(\chi_{k-1}, \mathbf{u}_k, \mathbf{w}_k); \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q})$$

3. Compute the predicted sigma points mean $\hat{\chi}_{\mathbf{k}}^-$ and covariance $\mathbf{P}_{\chi\chi}^-$:

$$\hat{\chi}_{\mathbf{k}}^- = \sum_{i=0}^{2n} \omega_m^{[i]} \chi_{\mathbf{k}}^{[i]}$$

$$\mathbf{P}_{\chi\chi}^- = \sum_{i=0}^{2n} \omega_c^{[i]} (\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^-) (\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^-)^T$$

where $\omega_m^{[i]}$, $\omega_c^{[i]}$ are defined by the algorithm.

4. Propagate the new sigma points through the measurement model:

$$\mathbf{z}_{\mathbf{k}} = h(\chi_{\mathbf{k}}^{[i]}, \mathbf{v}_k); \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$$

5. Compute the new sigma points mean $\hat{\mathbf{z}}_{\mathbf{k}}$, the predicted measurement covariance \mathbf{P}_{zz} , and the state and measurement cross-covariance $\mathbf{P}_{\chi z}$:

$$\hat{\mathbf{z}}_{\mathbf{k}} = \sum_{i=0}^{2n} \omega_m^{[i]} \mathbf{z}_{\mathbf{k}}^{[i]}$$

$$\mathbf{P}_{zz} = \sum_{i=0}^{2n} \omega_c^{[i]} (\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}}) (\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}})^T$$

$$\mathbf{P}_{\chi z} = \sum_{i=0}^{2n} \omega_c^{[i]} (\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^-) (\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}})^T$$

Correction:

1. Compute the innovation $\mathbf{v}_{\mathbf{k}}$ from the current and predicted measurement $\mathbf{z}_{\mathbf{k}}$, $\hat{\mathbf{z}}_{\mathbf{k}}$, respectively:

$$\mathbf{v}_{\mathbf{k}} = \mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k}}$$

2. Update Kalman gain matrix:

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\chi z} \mathbf{P}_{zz}^{-1}$$

3. Update the posterior mean and covariance ($\hat{\chi}_{\mathbf{k}}$, $\mathbf{P}_{\chi\chi}$):

$$\hat{\chi}_{\mathbf{k}} = \hat{\chi}_{\mathbf{k}}^- + \mathbf{K}_{\mathbf{k}} (\mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k}})$$

$$\mathbf{P}_{\chi\chi} = \mathbf{P}_{\chi\chi}^- - \mathbf{K}_{\mathbf{k}} \mathbf{P}_{zz} \mathbf{K}_{\mathbf{k}}^T$$

return ($\hat{\chi}_{\mathbf{k}}$, $\mathbf{P}_{\chi\chi}$)

Compute the Degree of Belief (Bel):

$$\text{Bel}_k = 1 - \|\mathbf{P}_{\chi\chi}\|_2^{0.5}; \quad 0 \leq \text{Bel} \leq 1$$

end function

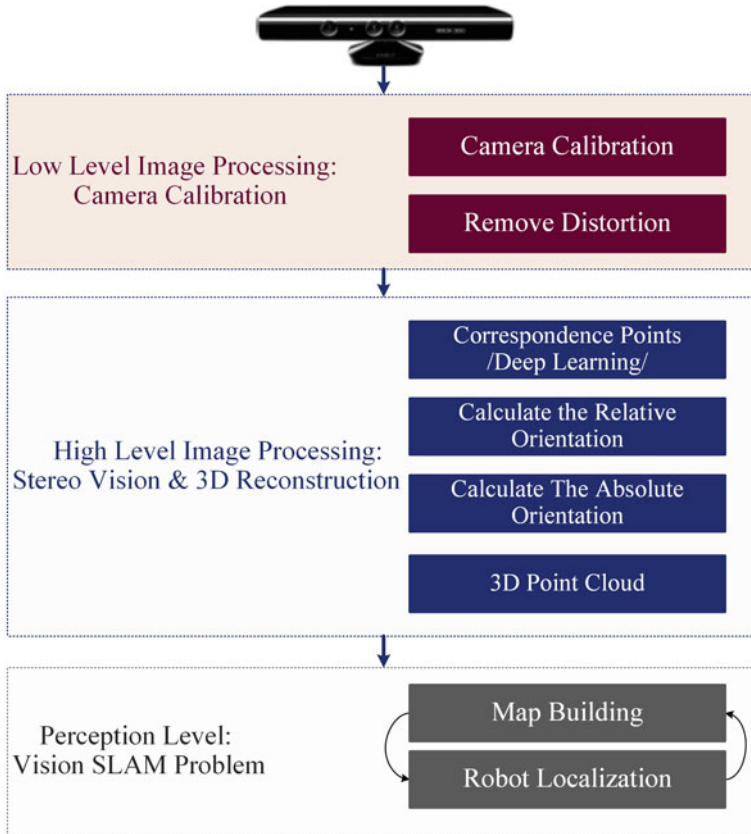


Fig. 9.2 Stereo vision 3D reconstruction stage

always intersect due to erroneous correspondences, the midpoint method or least square based method is proposed to estimate the intersection. Then to avoid the drifting problem, UKF is employed to refine both the camera pose and 3D points by minimizing re-projection errors [26].

9.3.1 Perspective Projection and Collinearity Constraint

The process by which the 3D objects are mapped onto an image by a camera is approximated by *collinearity constraint* [13].

Figure 9.3 shows the perspective projection geometry.

As seen from Fig. 9.3, light falling on the image plane is assumed to have passed through a small pinhole. Therefore, each object point P_ω maps to a single point on the image plane P_u . Three-coordinate systems are necessary to define a perspective camera model [23]: (a) The 3D world coordinate system $\{W\}$. (b) The camera

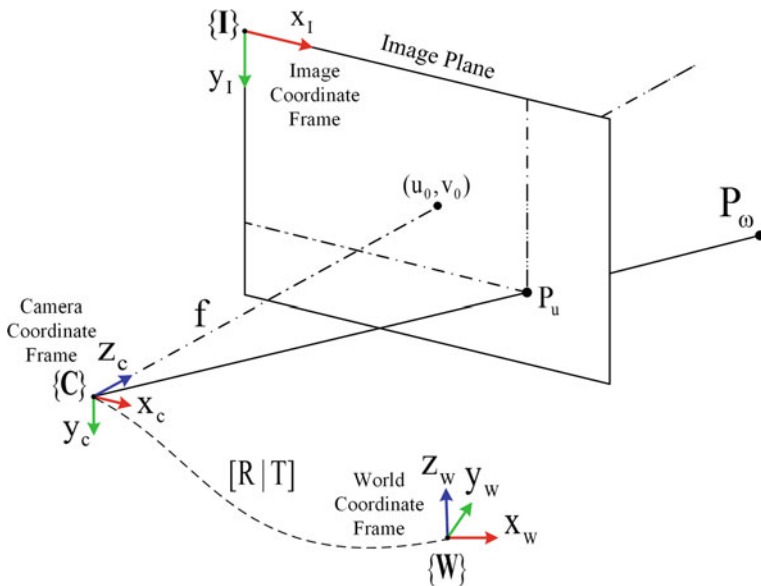


Fig. 9.3 Perspective projection model

coordinate system $\{C\}$; it is attached to the projection center of the camera. The sensor plane is parallel to its xy plane and displaced in a positive z direction. This axis pierces the image plane at the principal point (u_0, v_0) which acts as the origin of the image plane. (c) The 2D image coordinate system $\{I\}$; its origin lies at the upper left corner of the image. Two sets of parameters are used for perspective camera modeling [13]:

- **Extrinsic Parameters (Extrinsics):** These parameters describe the camera pose in the environment. Extrinsics contain six parameters of the exterior orientation (EO) of the projection center (i.e., three parameters for the translation and three other parameters for the rotation). They all vary with the camera motion in the environment.
- **Intrinsic Parameters (Intrinsic):** These parameters model the camera physics and describe the interior orientation (IO) of the camera. The intrinsic parameters are determined by calibration and are usually fixed. The parameters are now in place to define perspective projection mathematically. The mapping with an ideal perspective camera can be decomposed into two steps [23]:

1. Exterior orientation: Given the 3D object's position in world frame ${}^wP_\omega = [{}^w x_\omega \ {}^w y_\omega \ {}^w z_\omega]^T$, the 3D object's position with respect to the camera frame using homogeneous notation ${}^C T_{W4 \times 4}$ (rotation matrix ${}^C R_{W3 \times 3}$ and translation matrix ${}^C D_{W3 \times 1}$) is given by (9.27):

$${}^C\mathbf{P}_\omega = {}^C\mathbf{T}_W {}^W\mathbf{P}_\omega \quad (9.27)$$

2. Interior orientation: A projection from camera frame to image frame using the calibration matrix κ and intrinsic parameters (the focal length \mathbf{f} , the horizontal and vertical scale vector $\mathbf{k}_u, \mathbf{k}_v$) is given by (9.28):

$$\kappa = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (9.28)$$

where $\alpha_u = -k_{uf}$, $\alpha_v = -k_{vf}$. If the shear parameter s and the scale difference m are present, they amount to an affine distortion of the image frame. It is useful to model the distortions as corrections $\Delta u, \Delta v$ of the image coordinates of a perspective camera, and the calibration matrix becomes (9.29):

$$\kappa = \begin{pmatrix} \alpha_u & s\alpha_v & u_0 + \Delta u & 0 \\ 0 & \alpha_v(1+m) & v_0 + \Delta v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (9.29)$$

If κ is known, then the camera is considered to be calibrated. The final mathematical formation of the collinearity constraint for a perspective projection with distortion from object to image frame is given by (9.30):

$${}^I\mathbf{P}_u = \kappa {}^C\mathbf{P}_\omega \quad (9.30)$$

9.3.2 Epipolar Geometry and Coplanarity Constraint

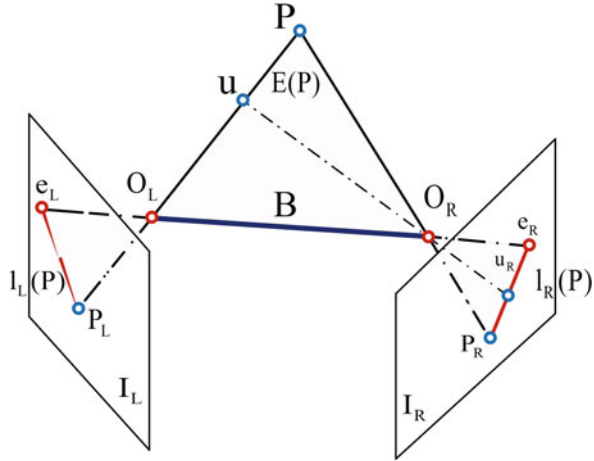
A 3D measurement cannot be derived from a single image of an unknown scene, because the depth along the Z axis is lost during projection. The principle to solve this problem is to measure the corresponding points acquired from two different viewpoints and reconstruct the 3D coordinates via triangulation. Some of these points are considered as control points [27]. Now, two problems need to be solved:

1. Determination of the image pair orientation (relative and absolute orientations).
2. Reconstruction of the 3D scene coordinates.

Figure 9.4 shows what is known as epipolar geometry. \mathbf{e}_L is the *epipole* which is the image of the right camera center in the left camera. \mathbf{e}_R is the epipole of the left camera center in the right camera. The plane formed by \mathbf{P} and the two camera centers O_L, O_R is the *epipolar plane*.

This plane intersects the image planes in the *epipolar line* l_L, l_R ; these lines can be used for matching points, and \mathbf{B} is the baseline, which is the distance between the projection center, O_R and O_L . All epipolar lines converge at the epipole. First,

Fig. 9.4 Epipolar geometry



it seems that the matching process requires cross image searching, but epipolar restrictions reduce this search to a single line [13]. The two projection rays must be coplanar because they intersect in the 3D point \mathbf{P} . For any corresponding point \mathbf{P} , triple product coplanarity can be expressed by (9.31):

$$[O_L P_L \ O_L O_R \ O_R P_R] = 0 \tag{9.31}$$

That is, the three rays are in one plane. The third dimension is extracted from a pair of images using the coplanarity constraint given for the uncalibrated camera as follow (9.32):

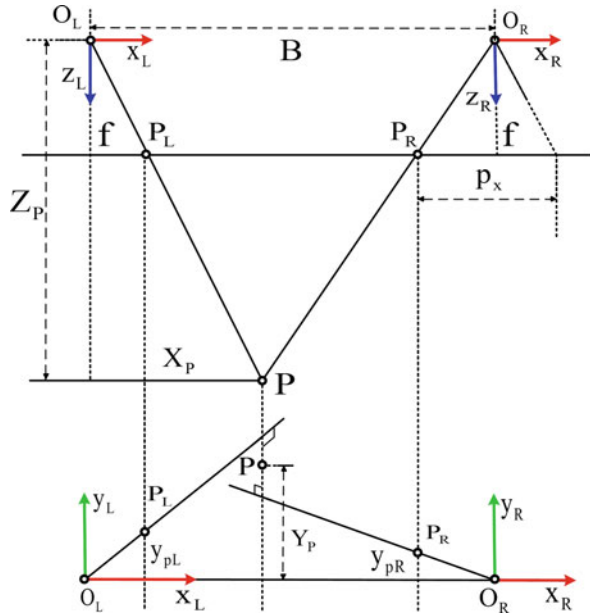
$$p_R^T \underbrace{\kappa_R^{-T} \mathbf{R}_R^{-T} \mathbf{S}_b \mathbf{R}_L^{-1} \kappa_L^{-1}}_{\mathbf{F}} p_L = 0 \tag{9.32}$$

where \mathbf{S}_b is a skew-symmetric matrix resulting from the triple product given in (9.31), κ_R, κ_L are the right and left calibration matrix, and $\mathbf{R}_R, \mathbf{R}_L$ are the rotation matrix of the right and left camera. The *fundamental matrix* $\mathbf{F}_{3 \times 3}$ sums up everything that can be known about the relationship between two uncalibrated cameras. Using $\mathbf{F}_{3 \times 3}$, it is possible to calculate the positions of the epipoles and the epipolar line in one image associated with a point in the other one. An alternative form for the epipolar geometry in case of the calibrated camera using the *essential matrix* $\mathbf{E}_{3 \times 3}$ is given by (9.33):

$$p_R^T \underbrace{\mathbf{R}_R \mathbf{S}_b \mathbf{R}_L^T}_{\mathbf{E}} p_L = 0 \tag{9.33}$$

The depth of the 3D point is calculated using the triangulation principle as shown in Fig. 9.5.

Fig. 9.5 Stereo vision triangulation [13]



The 3D coordinates of a landmark can be computed from two matched points in the left and right images by (9.34):

$$X = x_R \frac{B}{p_x}, Y = \frac{y_L + y_R}{2} \frac{B}{p_x}, Z = f \frac{B}{p_x} \tag{9.34}$$

where $x_{R, L}$, $y_{R, L}$ are the interest coordinate points in the images, X, Y, Z represent the object coordinates in the world frame, B is the baseline, and f is the focal length. The x-parallax $p_x = x_R - x_L$ is the distance between identical pixels when two images are mounted on top of each other [13].

9.4 Uncertainties in Stereo Vision System

Uncertainties are always present in the image acquisition and processing steps. Images are distorted due to various types of random noises such as Gaussian noise, Poisson noise, Quantization noise, Salt and paper noise, etc. These noises may be introduced from noise sources, for example, inaccurate image capturing devices like cameras, misaligned lenses, weak focal length, faulty memory location, etc. [28]. There are two main error categories, namely deterministic and nondeterministic [29]. The camera intrinsic parameter uncertainties are deterministic since the camera is supposed to be calibrated. In a stereo vision system with parallel optical axis as shown in Fig. 9.6, the epipolar constraint reduces to check that both the features

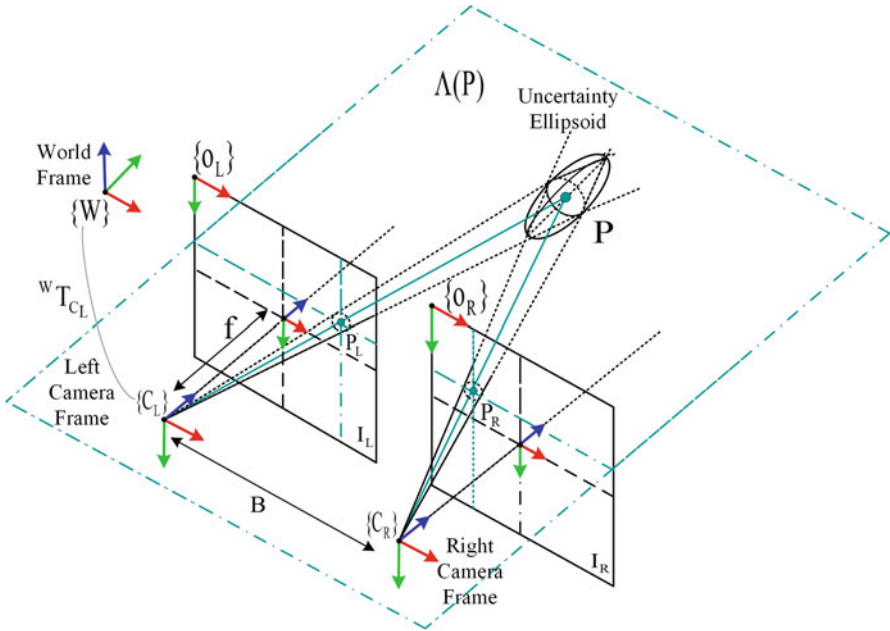


Fig. 9.6 Stereo vision uncertainty

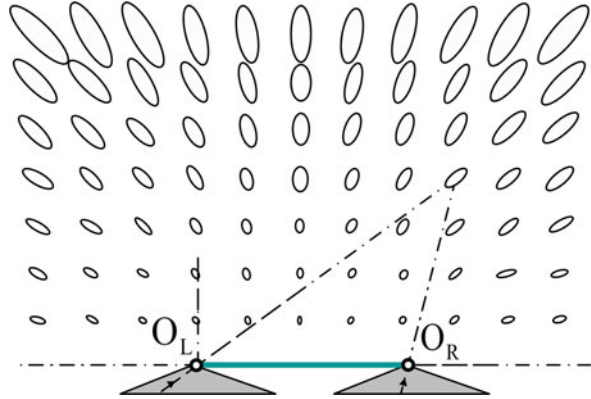
are in the same row of the image. Consider the uncertainty in the 3D landmark position due to errors in the image quantization and in the feature detection process. Once the matching has been established, the most likely 3D coordinates of the landmark are estimated by projecting them back to the environment [30]. Refer to Eq. (9.34), error in the variables x , y , p_x are usually modeled as uncorrelated zero-mean Gaussian random variables [13]. Using the first-order error propagation to approximate the distribution of the variables in (9.34) as multivariable Gaussian, the following covariance matrix for the x , y , z coordinates has been obtained (9.35):

$$\Sigma \approx \mathbf{J} \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_x^2) \mathbf{J}^T \tag{9.35}$$

where \mathbf{J} represents the Jacobian matrix of the functions in (9.34), and $(\sigma_x^2, \sigma_y^2, \sigma_x^2)$ are the variances of the corresponding variables.

The theoretical precision of 3D points depends on the uncertainty of the relative orientation and the uncertainty of measured corresponding points. Assume that the uncertainty of the relative orientation is negligible. By variance propagation using (9.36)

$$\sigma_X^2 = \frac{Z}{p_x} \sigma_{p_x} = \frac{fB}{p_x^2} \sigma_{p_x} = \frac{Z^2}{fB} \sigma_{p_x} = \frac{Z}{f} \frac{1}{B/Z} \sigma_{p_x} \tag{9.36}$$

Fig. 9.7 Stereo vision

where Z coordinate is the distance of the point \mathbf{P} from the principal plane, \mathbf{B} is the baseline, \mathbf{f} is the focal length, \mathbf{p}_x is the x -parallax, and $\sigma_{\mathbf{p}_x}$ is the standard deviation of the point \mathbf{P} . The result shows the uncertainty proportional to Z^2 of the distance from the baseline \mathbf{B} for a given geometry point \mathbf{P} as shown in Fig. 9.7.

9.5 Examples

9.5.1 Pose Tracking Using UKF and Stereo Vision

An important improvement of the surgical robot is to extract pose information about the robot relative position to the patient. The absolute pose of the surgical robot relative to the patient cannot be observed directly due to the highly dynamic nature of the operating environment and uncertainties in the robot kinematics model [31]. To solve this problem, VSLAM is applied using the endoscopic stereo camera to estimate the robot motion. In many applications, KF framework is used to estimate the motion of the target object from the previous frame to the new frame [32]. In this example, a UKF approach is used to estimate the pose of the robot. Assuming that the feature points are observable throughout the sequence, the formulation of the UKF is as follows:

The state vector \mathbf{X}_k is given by (9.37):

$$\mathbf{X}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ z_k \ \dot{z}_k \ \alpha_k \ \dot{\alpha}_k \ \beta_k \ \dot{\beta}_k \ \varphi_k \ \dot{\varphi}_k]^T \quad (9.37)$$

where $x_k, y_k, z_k, \alpha_k, \beta_k, \varphi_k$ are the object's pose and orientation along the $x, y,$ and z axes, respectively, and $\dot{x}_k, \dot{y}_k, \dot{z}_k, \dot{\alpha}_k, \dot{\beta}_k, \dot{\varphi}_k$ are their corresponding velocities.

The dynamic system equation is given by (9.38):

$$\begin{aligned} \mathbf{X}_k &= \mathbf{A}\mathbf{X}_{k-1} + \boldsymbol{\omega}_k \\ \mathbf{A} &= \text{diag} \left[\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \right] \end{aligned} \quad (9.38)$$

where T_s is the sample time, and $\boldsymbol{\omega}_k$ is zero-mean Gaussian noise.

The nonlinear measurement model is defined as (9.39):

$$\mathbf{z}_k = h(\mathbf{X}_k) + \mathbf{v}_k \quad (9.39)$$

where \mathbf{v}_k is a $4m \times 1$ zero-mean Gaussian noise vector imposed on the images captured. m is the number of feature points extracted from the tracked robot. $h(\mathbf{X}_k)$ is the $4m \times 1$ output stereo image pair point transfer function. The estimated coordinates of the feature points at the sample time k is given by (9.40):

$$h(\mathbf{X}_k) = \left[u_{1,k}^L \ v_{1,k}^L \ \cdots \ u_{m,k}^L \ v_{m,k}^L \ \cdots \ u_{1,k}^R \ v_{1,k}^R \ \cdots \ u_{m,k}^R \ v_{m,k}^R \right]^T \quad (9.40)$$

The corresponding points have the following coplanarity constraint (9.41):

$$p_R^T \mathbf{E} p_L = 0 \quad (9.41)$$

The standard perspective projection for a single feature point is given by (9.42):

$$\mathbf{z}_k = \begin{bmatrix} u_k^L \\ v_k^L \\ u_k^R \\ v_k^R \end{bmatrix} = \begin{bmatrix} f x_x / z_x \\ f y_x / z_x \\ f (B - x_x) / z_x \\ f y_x / z_x \end{bmatrix} \quad (9.42)$$

where $\mathbf{z}_k = [u_k^L \ v_k^L \ u_k^R \ v_k^R]^T$ are the measurement pixels in the left and right images, \mathbf{B} is the baseline, and f is the focal length. The UKF algorithm can be derived using Table 9.2, giving the motion model (9.38) and the measurement model (9.39). As shown in Fig. 9.8, the robot is captured using the stereo camera with a location sensor.

The important features of the robot are then extracted and passed to the deep learning network. The robot pose with respect to the image frame is then matched with the predicted pose using UKF approach. The algorithm is initialized using epipolar geometry of the first two images and computes the essential matrix \mathbf{E} using 5-point algorithm plus robust estimator deep learning network. The initial pose parameters are then extracted from \mathbf{E} [33]. This is an initial guess of the pose and will be used in UKF approach.

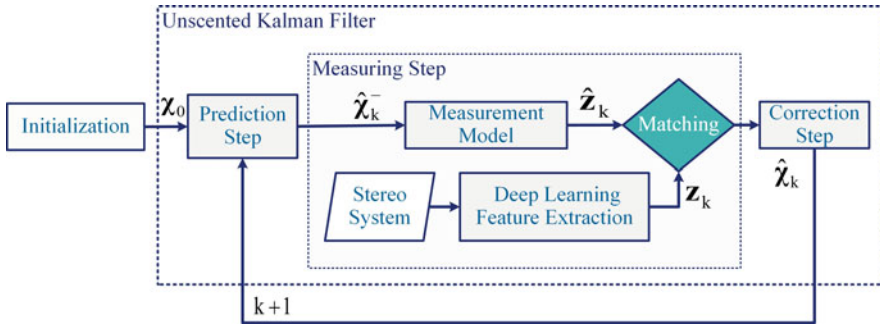
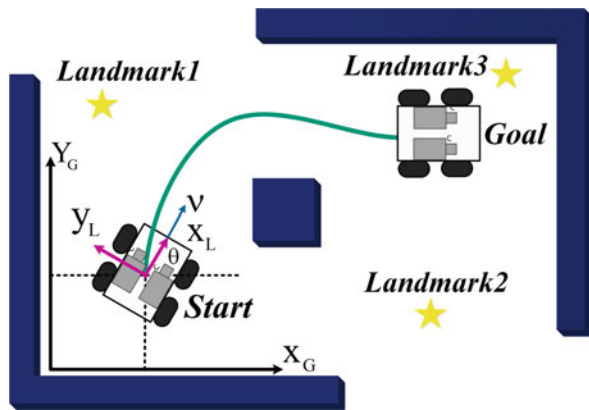


Fig. 9.8 Using UKF and stereo vision for robot tracking

Fig. 9.9 Robot localization-based landmarks [35]



9.5.2 Localization Approach-Based 2D Landmark Map

Landmarks are natural or artificial environment features which a robot can recognize from its sensory input and keep the uncertainty bounded [34]. Consider 4-WDDMR moves in the predefined environment in the global frame $\{G\}$ as shown in Fig. 9.9. Stereo vision system is used as an exterior receptive sensor to enable the robot to recognize the landmarks [36]. The predefined landmarks in the global frame $\{G\}$ help the robot to localize itself correctly. The 4-WDDMR discrete kinematic equation is given by (9.43):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_{k-1} + v_k T_s \cos(\theta_{k-1}) \\ y_{k-1} + v_k T_s \sin(\theta_{k-1}) \\ \theta_{k-1} + \omega_k T_s \end{bmatrix} \quad (9.43)$$

where T_s is the sample time and $f(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is a nonlinear function and relates the 4-WDDMR's pose $\mathbf{x}_k = [x_k \ y_k \ \theta_k]^T$ in $\{G\}$ frame, with the input vector $\mathbf{u}_k =$

$[v_k \ \omega_k]^T$, which represents the translation and angular velocities in the robot frame $\{L\}$. However, the noisy 4-WDDMR motion model with non-additive Gaussian noise w_k is written as follows (9.44):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (9.44)$$

In the authors' previous work [37], it is assumed that landmarks have a fixed and known position (m_x, m_y) .

The measurement model of the location z_k of the landmark m_k from the viewpoint of the robot given the location of the robot x_k is defined as follows (9.45):

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{m}_k) = \begin{bmatrix} h_x(\mathbf{x}_k, \mathbf{m}_k) \\ h_y(\mathbf{x}_k, \mathbf{m}_k) \end{bmatrix} \\ &= \begin{bmatrix} (m_{xk} - x_k) \cos(\theta_k) + (m_{yk} - y_k) \sin(\theta_k) \\ -(m_{xk} - x_k) \sin(\theta_k) + (m_{yk} - y_k) \cos(\theta_k) \end{bmatrix} \end{aligned} \quad (9.45)$$

where $h(\mathbf{x}_k, \mathbf{m}_k)$ is the nonlinear measurement function for the correction stage.

9.6 Conclusion

This book chapter has provided modern and advanced strategies for image filtering and image feature extraction. Kalman filter including EKF and UKF has been described in detail and implemented for pose tracking and localization-assisted 2D landmark map of a mobile robot. This chapter has contributed to the community of mobile robot localization and mapping, with detailed information on different modern approaches for vision-based localization and mapping. It has been pointed out that UKF algorithm is superior to EKF one in terms of bias cancellation and providing higher accuracy of mobile robot pose estimation.

References

1. Goliaei, S., Ghorshi, S., Manzuri, M. T., & Mortazavi, M. (2011). A Kalman filter techniques applied for medical image reconstruction. In *8th International Multi-Conference on Systems, Signals & Devices*.
2. Ling, L. (2013). *Dense real time 3D reconstruction from multiple images*. PhD thesis, College of Science, Engineering and Health, RMIT University.
3. Saputra, M. R. U., Markham, A., & Trigoni, N. (2018). Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys*, 51, 2.
4. Klančar, G., Teslic, L., & Skrjanc, I. (2014). Mobile robot pose estimation and environment mapping using an extended Kalman filter. *International Journal of Systems Science*, 45(12), 2603–2618.

5. Basaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., Garcia, X., Tyrsa, V. V., Rivas-Lopez, M., & Tabakova, I. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
6. Park, J., & Lee, S. (2009). Correction robot pose for SLAM based on extended Kalman filter in rough surface environment. *International Journal of Advanced Robotic System*, 6(2), 67–72.
7. Moreno, F. A., Blanco, J. L., & Gonzalez, J. (2009). Stereo vision-specific models for particle filter-based SLAM. *Robotics and Autonomous Systems*, 57(9), 955–970.
8. Tully, S. T. (2012). *BodySLAM: Localization and mapping for surgical guidance*. PhD thesis, Garnegir Mellon University, Pittsburgh.
9. Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Springer.
10. Thrun, S., Fox, D., & Burgard, W. (2006). *Probabilistic robotics*. Massachusetts Institute of Technology.
11. Merwe, R. (2004). *Sigma point Kalman Filter for probabilistic inference in dynamic state space models*. PhD thesis, Oregon Health & Science University.
12. Sibley, G., Sukhatme, G., & Matthies, L. (2006). *The iterated sigma point Kalman filter with applications to long range stereo*. Robotics Science and Systems.
13. Förstner, W., & Wrobel, B. P. (2016). *Photogrammetric computer vision statistics, geometry, orientation and reconstruction*. Cham, Switzerland: Springer.
14. Särkkä, S. (2011). *Bayesian filtering and smoothing* (Vol. 3). Cambridge University Press.
15. Corke, P. (2011). Robotics vision and control fundamental algorithms in MATLAB. In *Springer tracts in advanced robotics* (Vol. 73). Springer.
16. Haykin, S. M. (2001). *Kalman filtering and neural networks*. Wiley.
17. Joukhadar, A., & Kass Hanna, D., (2018). UKF and adaptive optimal control-based localization enhancement of 4WDDMR, ROS framework-based design and implementation. In *Cogent engineering, System and Control Research Article*.
18. Long, A. W., Wolfe, K. C., Mashner, M. J., & Chirikjian, G. S. (2013). *The banana distribution is Gaussian: A localization study with exponential coordinates* (pp. 265–272). Cambridge, MA: Robotics: Science and Systems VIII; MIT Press.
19. Mahmoudi, Z., Poulsen, N. K., Madsen, H., & Jorgensen, J. B. (2017). Adaptive unscented Kalman filter using maximum likelihood estimation. *IFAC-Papers Online*, 50(1), 3859–3864. <https://doi.org/10.1016/j.ifacol.2017.08.356>.
20. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes. The art of scientific computing*. Cambridge, UK: Cambridge University Press.
21. Rodríguez-Quirfónez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision system using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
22. Hu, Y., Chen, Q., Feng, S., Tao, T., Asundi, A., & Zuo, C. (2019). A new microscopic telecentric stereo vision system—Calibration, rectification, and three-dimensional reconstruction. *Optics and Lasers in Engineering*, 113, 14–22.
23. Bergamini, M. L., Ansaldo, F. A., Bright, G., & Zelasco, J. F. (2017). Digital camera calibration, relative orientation and essential matrix parameters. *WSEAS Transaction on Signal Processing*, 13.
24. Hayet, J. B., Lerasle, F., & Devy, M. (2002). A visual landmark framework for indoor mobile robot navigation. In *International Conference on Robotics & Automation*. Washington, DC.
25. Florczyk, S. (2005). *Video-based indoor exploration with autonomous and mobile robots*. Wiley.
26. Hartmann, G., Huang, F., & Klette, R. (2013). *Landmark initialization for unscented Kalman filter sensor fusion in monocular camera localization*. Auckland, New Zealand: The University of Auckland.
27. Parnian, N., & Golnaraghi, F. (2010). Integration of multi-camera vision system and strap down inertial navigation system (SDINS) with a modified Kalman filter. *Sensors Journal*, 10(6), 5378–5394.

28. Boyat, A. K., & Joshi, B. K. (2015). A review paper: Noise models in digital image processing. *Signal & Image Processing: An International Journal (SIPIJ)*, 6(2).
29. Siegwart, R., & Noubakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. Cambridge, MA, London: The MIT Press.
30. Beinhofer, M. Müller, J., Krause, A., & Burgard, W. (2013). Robust landmark selection for mobile robot navigation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference*.
31. Haghighipanah, M., Miyasaka, M., Li, Y., & Hannaford, B. (2016). Unscented Kalman filter and vision to improve cable driven surgical robot joint angle estimation. In *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*.
32. Yu, Y. K., Wong, K. H., Or, S. H., & Chang, M. M. Y. (2006). Recursive camera motion estimation with trifocal tensor. *IEEE Transaction on System, Man and Cybernetics B*, 36(5), 1081–1090.
33. Yu, Y. K., Wong, K. H., Or, S. H., & Chang, M. M. Y. (2008). Robust 3D motion tracking from stereo images: A model-less method. *IEEE Transaction on Instrumentation and Measurement*, 57(3).
34. Negenborn, R. (2003). *Robot localization and Kalman Filters on finding your position in a noisy word*. MSc thesis, Utrecht University.
35. Joukhadar, A., Kass Hanna, D., Müller, A., & Stöger, C., (2017). UKF-Assisted SLAM for 4WDDMR Localization and Mapping. In *1st International Congress for the Advancement of Mechanism, Machine, Robotics and Mechatronics Sciences, Beirut, Lebanon*, 17–19 October.
36. Kelly, J., & Sukhatme, G. S. (2009). *Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration*. Korea: CIRA.
37. Kass Hanna, D., & Joukhadar, A. (2015). A novel control-navigation system-based adaptive optimal controller & EKF localization of DDMR. *International Journal of Advance Research in Artificial Intelligence*, 4(25), 29–37.

Part III
Pose Estimation, Avoidance of Objects,
Control and Data Exchange for Navigation

Chapter 10

Lie Algebra Method for Pose Optimization Computation



Kenichi Kanatani

Acronyms

- FNS Fundamental Numerical Scheme
- GPS Ground Positioning System
- SVD Singular Value Decomposition

10.1 Introduction

Computing 3D pose from data provided by camera images and 3D sensors is one of the most fundamental problems of 3D analysis involving 3D data, including computer vision and robot control. The problem is usually formulated as minimization of a function of the form

$$J = J(\dots, \mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_M), \tag{10.1}$$

where $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_M$ are rotation matrices, and “...” denotes other parameters that specify translations, object shapes, and other properties. Hereafter, we use bold uppercases to denote matrices (3×3 unless otherwise specified) and bold lowercases to denote vectors (3D unless otherwise specified). For a matrix \mathbf{A} , we write its determinant and Frobenius norm as $|\mathbf{A}|$ and $\|\mathbf{A}\|$, respectively. For vectors \mathbf{a} and \mathbf{b} , we write $\langle \mathbf{a}, \mathbf{b} \rangle$ for their inner product and $\mathbf{a} \times \mathbf{b}$ for their vector product.

K. Kanatani (✉)
Professor Emeritus, Okayama University, Okayama, Japan
e-mail: kanata-k@okayama-u.ac.jp

For minimizing a function J in the form of Eq. (10.1), the standard approach one can immediately think of is: we first parameterize the rotation matrices in terms of, say, axis-angle, Euler angles, or quaternions; then we differentiate J with respect to the parameters and increment them so that J decreases; we iterate this. This approach is generally known as the “gradient method,” and many variations have been proposed for improving convergence, including “steepest descent,” “conjugate gradient,” “Newton iterations,” “Gauss–Newton iterations,” and the “Levenberg–Marquardt method.”

The purpose of this chapter is to show that for this type of optimization, *parameterization of rotation is not necessary*. After all, “differentiation” means evaluation of the change of the function value for a small variation of the variable. Hence, for differentiation with respect to rotation \mathbf{R} , we only need to evaluate the change of the function value when a small rotation is added to \mathbf{R} . To do this, it is sufficient to *parameterize a small rotation*. To be specific, we compute a small rotation that reduces the function J , add it to the current rotation \mathbf{R} , regard the resulting rotation as a new current rotation \mathbf{R} , and iterate this process. As a result, the matrix \mathbf{R} is updated at each iteration in the computer memory, so that there is no need to parameterize the matrix \mathbf{R} itself. We call this the “Lie algebra method” (this terminology is explained later).

This method has a big advantage over the parameterization approach, because any parameterization of rotation, such as axis-angle, Euler angles, and quaternions, has some singularities; if the parameter values happen to be at singularities, though very rarely, computational problems such as numerical instability may occur. Using the Lie algebra method, we need not worry about any singularities of the parameterization, because all we do is to update the current rotation by adding a small rotation. In a sense, this is obvious, but not many people understand this fact.

We first study the relationship between small rotations and angular velocities. Then, we derive the exponential expression of rotation and formalize the concept of “Lie algebra.” We describe the actual computational procedure of some computer vision problems to demonstrate how the Lie algebra method works in practice. Finally, we overview the role of Lie algebra in various computer vision applications.

10.2 Small Rotations and Angular Velocity

If \mathbf{R} represents a small rotation around some axis by a small angle $\Delta\Omega$, we can Taylor-expand it in the form

$$\mathbf{R} = \mathbf{I} + \mathbf{A}\Delta\Omega + O(\Delta\Omega)^2, \quad (10.2)$$

for some matrix \mathbf{A} , where \mathbf{I} is the identity and $O(\Delta\Omega)^2$ denotes terms of second or higher orders in $\Delta\Omega$. Since \mathbf{R} is a rotation matrix,

$$\begin{aligned} \mathbf{R}\mathbf{R}^\top &= (\mathbf{I} + \mathbf{A}\Delta\Omega + O(\Delta\Omega)^2)(\mathbf{I} + \mathbf{A}\Delta\Omega + O(\Delta\Omega)^2)^\top \\ &= \mathbf{I} + (\mathbf{A} + \mathbf{A}^\top)\Delta\Omega + O(\Delta\Omega)^2 \end{aligned} \quad (10.3)$$

must be identically equal to \mathbf{I} for any $\Delta\Omega$. Hence, $\mathbf{A} + \mathbf{A}^\top = \mathbf{O}$, or

$$\mathbf{A}^\top = -\mathbf{A}. \quad (10.4)$$

This means that \mathbf{A} is an antisymmetric matrix, so we can write it as

$$\mathbf{A} = \begin{pmatrix} 0 & -l_3 & l_2 \\ l_3 & 0 & -l_1 \\ -l_2 & l_1 & 0 \end{pmatrix} \quad (10.5)$$

for some l_1 , l_2 , and l_3 . If a vector $\mathbf{a} = (a_i)$ (abbreviation of a vector whose i th component is a_i) is rotated to \mathbf{a}' by the rotation of Eq. (10.2), we obtain

$$\begin{aligned} \mathbf{a}' &= (\mathbf{I} + \mathbf{A}\Delta\Omega + O(\Delta\Omega)^2)\mathbf{a} = \mathbf{a} + \begin{pmatrix} 0 & -l_3 & l_2 \\ l_3 & 0 & -l_1 \\ -l_2 & l_1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \Delta\Omega + O(\Delta\Omega)^2 \\ &= \mathbf{a} + \begin{pmatrix} l_2a_3 - l_3a_2 \\ l_3a_1 - l_1a_3 \\ l_1a_2 - l_2a_1 \end{pmatrix} \Delta\Omega + O(\Delta\Omega)^2 = \mathbf{a} + \mathbf{l} \times \mathbf{a}\Delta\Omega + O(\Delta\Omega)^2, \end{aligned} \quad (10.6)$$

where we let $\mathbf{l} = (l_i)$. Suppose this describes a continuous rotational motion over a small time interval Δt . Its velocity is given by

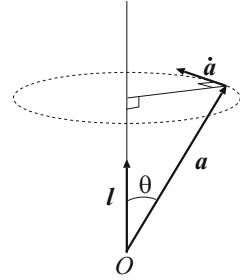
$$\dot{\mathbf{a}} = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{a}' - \mathbf{a}}{\Delta t} = \omega \mathbf{l} \times \mathbf{a}, \quad (10.7)$$

where we define the *angular velocity* ω by

$$\omega = \lim_{\Delta t \rightarrow 0} \frac{\Delta\Omega}{\Delta t}. \quad (10.8)$$

Equation (10.7) states that the velocity $\dot{\mathbf{a}}$ is orthogonal to both \mathbf{l} and \mathbf{a} and that its magnitude equals ω times the area of the parallelogram made by \mathbf{l} and \mathbf{a} . From the geometric consideration, the velocity $\dot{\mathbf{a}}$ is orthogonal to the axis of rotation and \mathbf{a} itself (Fig. 10.1). If we let θ be the angle made by \mathbf{a} and that axis, the distance of the endpoint of \mathbf{a} from the axis is $\|\mathbf{a}\| \sin \theta$, and the definition of the angular velocity ω implies $\|\dot{\mathbf{a}}\| = \omega \|\mathbf{a}\| \sin \theta$. Since $\dot{\mathbf{a}}$ is orthogonal to \mathbf{l} and \mathbf{a} and since $\|\dot{\mathbf{a}}\| = \omega \|\mathbf{a}\| \sin \theta$ equals the area of the parallelogram made by \mathbf{l} and \mathbf{a} , we conclude that \mathbf{l} is the unit vector along the axis of rotation. In physics, the vector $\boldsymbol{\omega} = \omega \mathbf{l}$ is known

Fig. 10.1 Vector \mathbf{a} is rotating around an axis in the direction of the unit vector \mathbf{l} with angular velocity ω . Its velocity vector is $\dot{\mathbf{a}}$.



as the *angular velocity vector*. Using this notation, we can write Eq. (10.7) as

$$\dot{\mathbf{a}} = \boldsymbol{\omega} \times \mathbf{a}. \tag{10.9}$$

10.3 Exponential Expression of Rotation

If we write $\mathbf{R}_l(\Omega)$ to denote the rotation around axis \mathbf{l} (unit vector) by angle Ω , Eq. (10.2) equals $\mathbf{R}_l(\Delta\Omega)$. If we add it to rotation $\mathbf{R}_l(\Omega)$, their composition is $\mathbf{R}_l(\Delta\Omega)\mathbf{R}_l(\Omega) = \mathbf{R}_l(\Omega + \Delta\Omega)$. Hence, the derivative of $\mathbf{R}_l(\Omega)$ with respect to Ω is

$$\begin{aligned} \frac{d\mathbf{R}_l(\Omega)}{d\Omega} &= \lim_{\Delta\Omega \rightarrow 0} \frac{\mathbf{R}_l(\Omega + \Delta\Omega) - \mathbf{R}_l(\Omega)}{\Delta\Omega} = \lim_{\Delta\Omega \rightarrow 0} \frac{\mathbf{R}_l(\Delta\Omega)\mathbf{R}_l(\Omega) - \mathbf{R}_l(\Omega)}{\Delta\Omega} \\ &= \lim_{\Delta\Omega \rightarrow 0} \frac{\mathbf{R}_l(\Delta\Omega) - \mathbf{I}}{\Delta\Omega} \mathbf{R}_l(\Omega) = \mathbf{A} \mathbf{R}_l(\Omega). \end{aligned} \tag{10.10}$$

Differentiating this repeatedly, we obtain

$$\frac{d^2\mathbf{R}_l}{d\Omega^2} = \mathbf{A} \frac{d\mathbf{R}_l}{d\Omega} = \mathbf{A}^2 \mathbf{R}_l, \quad \frac{d^3\mathbf{R}_l}{d\Omega^3} = \mathbf{A}^2 \frac{d\mathbf{R}_l}{d\Omega} = \mathbf{A}^3 \mathbf{R}_l, \quad \dots, \tag{10.11}$$

where the argument (Ω) is omitted. Since $\mathbf{R}_l(0) = \mathbf{I}$, the Taylor expansion of $\mathbf{R}_l(\Omega)$ around $\Omega = 0$ is given by

$$\begin{aligned} \mathbf{R}_l(\Omega) &= \mathbf{I} + \left. \frac{d\mathbf{R}}{d\Omega} \right|_{\Omega=0} \Omega + \frac{1}{2} \left. \frac{d^2\mathbf{R}}{d\Omega^2} \right|_{\Omega=0} \Omega^2 + \frac{1}{3!} \left. \frac{d^3\mathbf{R}}{d\Omega^3} \right|_{\Omega=0} \Omega^3 + \dots \\ &= \mathbf{I} + \Omega \mathbf{A} + \frac{\Omega^2}{2} \mathbf{A}^2 + \frac{\Omega^3}{3!} \mathbf{A}^3 + \dots = e^{\Omega \mathbf{A}}, \end{aligned} \tag{10.12}$$

where we define the exponential of matrix by the following series expansion:

$$e^{\mathbf{X}} = \sum_{k=0}^{\infty} \frac{\mathbf{X}^k}{k!}. \tag{10.13}$$

In Eq. (10.12), the matrix \mathbf{A} specifies the axis direction in the form of Eq. (10.5). Hence, Eq. (10.12) expresses the rotation $\mathbf{R}_l(\Omega)$ in terms of its axis \mathbf{l} and angle Ω . An explicit expression for such a rotation, called the *Rodrigues formula*, is well known (see, e.g., [11, 14]):

$$\begin{aligned} \mathbf{R}_l(\Omega) &= \begin{pmatrix} \cos \Omega + l_1^2(1 - \cos \Omega) & l_1 l_2(1 - \cos \Omega) - l_3 \sin \Omega & l_1 l_3(1 - \cos \Omega) + l_2 \sin \Omega \\ l_2 l_1(1 - \cos \Omega) + l_3 \sin \Omega & \cos \Omega + l_2^2(1 - \cos \Omega) & l_2 l_3(1 - \cos \Omega) - l_1 \sin \Omega \\ l_3 l_1(1 - \cos \Omega) - l_2 \sin \Omega & l_3 l_2(1 - \cos \Omega) + l_1 \sin \Omega & \cos \Omega + l_3^2(1 - \cos \Omega) \end{pmatrix}. \end{aligned} \quad (10.14)$$

In the following, we combine the axis \mathbf{l} and angle Ω , as in the case of the angular velocity vector, as a single vector in the form of

$$\boldsymbol{\Omega} = \Omega \mathbf{l}, \quad (10.15)$$

which we call the *rotation vector*. We also write the matrix that represents the corresponding rotation as $\mathbf{R}(\boldsymbol{\Omega})$. Since $\Omega_1 = \Omega l_1$, $\Omega_2 = \Omega l_2$, and $\Omega_3 = \Omega l_3$, Eq. (10.5) is rewritten as

$$\boldsymbol{\Omega} \mathbf{A} = \Omega_1 \mathbf{A}_1 + \Omega_2 \mathbf{A}_2 + \Omega_3 \mathbf{A}_3, \quad (10.16)$$

where we define the matrices \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 by

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (10.17)$$

Hence, Eq. (10.12) is also written as

$$\mathbf{R}(\boldsymbol{\Omega}) = e^{\Omega_1 \mathbf{A}_1 + \Omega_2 \mathbf{A}_2 + \Omega_3 \mathbf{A}_3}, \quad (10.18)$$

which express the Rodrigues formula of Eq. (10.14).

10.4 Lie Algebra of Infinitesimal Rotations

Consider a rotation $\mathbf{R}(t)$ continuously changing with parameter t , which can be interpreted as time or angle of rotation or some control parameter. We assume it as a dimensionless parameter with appropriate normalization. We regard $t = 0$ as corresponding to the identity \mathbf{I} . We call a “linear” variation of $\mathbf{R}(t)$ around $t = 0$ an *infinitesimal rotation*. To be specific, we expand $\mathbf{R}(t)$ for the small change δt of

t and ignore terms of order two and higher in δt . From Eq. (10.2), we see that an infinitesimal rotation is expressed in the form

$$\mathbf{I} + \mathbf{A}\delta t, \quad (10.19)$$

for some antisymmetric matrix \mathbf{A} , which we call the *generator* of the infinitesimal rotation. If we accumulate this infinitesimal rotation continuously, we obtain a finite rotation $e^{t\mathbf{A}}$ as shown in the preceding section.

Note that any multiple of an infinitesimal rotation is also an infinitesimal rotation. This may sound counterintuitive, but this is the consequence of our defining infinitesimal rotations as “linear” variations of rotations. If the parameter t is regarded as time, multiplication of a generator by a constant c means multiplication of the instantaneous velocity by c .

We also see that the composition of infinitesimal rotations is also an infinitesimal rotation. In fact, if infinitesimal rotations $\mathbf{I} + \mathbf{A}\delta t$ and $\mathbf{I} + \mathbf{A}'\delta t$ are composed, we obtain

$$(\mathbf{I} + \mathbf{A}'\delta t)(\mathbf{I} + \mathbf{A}\delta t) = \mathbf{I} + (\mathbf{A} + \mathbf{A}')\delta t \quad (= (\mathbf{I} + \mathbf{A}\delta t)(\mathbf{I} + \mathbf{A}'\delta t)). \quad (10.20)$$

Recall that terms of order two and higher in δt are always ignored. From this, we see that, unlike finite rotations, the composition of infinitesimal rotations is *commutative*, i.e., it does not depend on the order of composition; the generator of the composed infinitesimal rotation is the *sum* of their generators. If we identify an infinitesimal rotation with its generator, we see that *the set of infinitesimal rotations constitutes a linear space*.

A linear space is called an *algebra* if it is closed under some product operation. The set of all the generators of infinitesimal rotations can be regarded as an algebra if we define a product of generators \mathbf{A} and \mathbf{B} by

$$[\mathbf{A}, \mathbf{B}] = \mathbf{A}\mathbf{B} - \mathbf{B}\mathbf{A}, \quad (10.21)$$

called the *commutator* of \mathbf{A} and \mathbf{B} . By definition, this is *anticommutative*:

$$[\mathbf{A}, \mathbf{B}] = -[\mathbf{B}, \mathbf{A}]. \quad (10.22)$$

The commutator is *bilinear*:

$$[\mathbf{A} + \mathbf{B}, \mathbf{C}] = [\mathbf{A}, \mathbf{C}] + [\mathbf{B}, \mathbf{C}], \quad [c\mathbf{A}, \mathbf{B}] = c[\mathbf{A}, \mathbf{B}], \quad c \in \mathcal{R}, \quad (10.23)$$

and the following *Jacobi identity* holds:

$$[\mathbf{A}, [\mathbf{B}, \mathbf{C}]] + [\mathbf{B}, [\mathbf{C}, \mathbf{A}]] + [\mathbf{C}, [\mathbf{A}, \mathbf{B}]] = \mathbf{O}. \quad (10.24)$$

An operation $[\cdot, \cdot]$ which maps two elements to another element is called a *Lie bracket* if the identities of Eqs. (10.22), (10.23), and (10.24) hold. Evidently, the

commutator of Eq. (10.21) defines a Lie bracket. An algebra equipped with a Lie bracket is called a *Lie algebra*.

Thus, the set of infinitesimal rotations is a Lie algebra under the commutator. Since the generator \mathbf{A} is an antisymmetric matrix, it has three degrees of freedom. Hence, the Lie algebra of infinitesimal rotations is three-dimensional with the matrices \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 in Eq. (10.17) as its basis. It is easy to see that they satisfy

$$[\mathbf{A}_2, \mathbf{A}_3] = \mathbf{A}_1, \quad [\mathbf{A}_3, \mathbf{A}_1] = \mathbf{A}_2, \quad [\mathbf{A}_1, \mathbf{A}_2] = \mathbf{A}_3. \quad (10.25)$$

In terms of this basis, an arbitrary generator \mathbf{A} is expressed in the form

$$\mathbf{A} = \omega_1 \mathbf{A}_1 + \omega_2 \mathbf{A}_2 + \omega_3 \mathbf{A}_3, \quad (10.26)$$

for some ω_1 , ω_2 , and ω_3 . From the definition of \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 in Eq. (10.17), Eq. (10.26) is rewritten as

$$\mathbf{A} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \quad (10.27)$$

This defines a 1-to-1 correspondence between a generator \mathbf{A} and a vector $\boldsymbol{\omega} = (\omega_i)$. Let $\boldsymbol{\omega}' = (\omega'_i)$ be the vector that corresponds to generator \mathbf{A}' . Then, the commutator of \mathbf{A} and \mathbf{A}' is

$$\begin{aligned} [\mathbf{A}, \mathbf{A}'] &= \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -\omega'_3 & \omega'_2 \\ \omega'_3 & 0 & -\omega'_1 \\ -\omega'_2 & \omega'_1 & 0 \end{pmatrix} \\ &\quad - \begin{pmatrix} 0 & -\omega'_3 & \omega'_2 \\ \omega'_3 & 0 & -\omega'_1 \\ -\omega'_2 & \omega'_1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -(\omega_1\omega'_2 - \omega_2\omega'_1) & \omega_3\omega'_1 - \omega_1\omega'_3 \\ \omega_1\omega'_2 - \omega_2\omega'_1 & 0 & -(\omega_2\omega'_3 - \omega_3\omega'_2) \\ -(\omega_3\omega'_1 - \omega_1\omega'_3) & \omega_2\omega'_3 - \omega_3\omega'_2 & 0 \end{pmatrix}, \quad (10.28) \end{aligned}$$

which shows that the *vector product* $\boldsymbol{\omega} \times \boldsymbol{\omega}'$ corresponds to the commutator $[\mathbf{A}, \mathbf{A}']$.

Evidently, all the relations of Eqs. (10.22), (10.23), and (10.24) hold if the commutator $[\mathbf{A}, \mathbf{B}]$ is replaced by the vector product $\mathbf{a} \times \mathbf{b}$. In other words, the vector product is a Lie bracket, and the set of vectors is also a Lie algebra under the Lie bracket $[\mathbf{a}, \mathbf{b}] = \mathbf{a} \times \mathbf{b}$. As shown above, the Lie algebra of vectors is the same as or, to be precise, *isomorphic* to the Lie algebra of infinitesimal rotations. Indeed, the matrices \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 in Eq. (10.17) represent infinitesimal rotations around

the x -, y -, and z -axes, respectively, and Eq. (10.25) corresponds to the relationships $\mathbf{e}_2 \times \mathbf{e}_3 = \mathbf{e}_1$, $\mathbf{e}_3 \times \mathbf{e}_1 = \mathbf{e}_2$, and $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{e}_3$ among the coordinate basis vectors $\mathbf{e}_1 = (1, 0, 0)^\top$, $\mathbf{e}_2 = (0, 1, 0)^\top$, and $\mathbf{e}_3 = (0, 0, 1)^\top$. The argument in Sects. 10.2 and 10.3 implies that identifying the generator \mathbf{A} of Eq. (10.27) with the vector $\boldsymbol{\omega} = (\omega_i)$ is nothing but *identifying an infinitesimal rotation with an instantaneous angular velocity vector*. In other words, we can think of the Lie algebra of infinitesimal rotations as *the set of all angular velocity vectors*. For more general treatments of Lie algebras, see [11].

10.5 Optimization of Rotation

Given a function $J(\mathbf{R})$ of rotation \mathbf{R} , we now consider how to minimize it, assuming that the minimum exists. In general, the solution can be obtained by differentiating $J(\mathbf{R})$ with respect to \mathbf{R} and finding the value of \mathbf{R} for which the derivative vanishes. But how should we interpret *differentiating with respect to \mathbf{R}* ?

As is well known, the derivative of a function $f(x)$ is the rate of change of the function value $f(x)$ when the argument x is infinitesimally incremented to $x + \delta x$. By “infinitesimal increment,” we mean considering the “linear” variation, ignoring higher order terms in δx . In other words, if the function value changes to $f(x + \delta x) = f(x) + a\delta x + \dots$, we call the coefficient a of δx the *differential coefficient*, or the *derivative*, of $f(x)$ with respect to x and write $a = f'(x)$. This is equivalently written as $a = \lim_{\delta x \rightarrow 0} (f(x + \delta x) - f(x)) / \delta x$. Evidently, if a function $f(x)$ takes its minimum at x , the function value does not change by infinitesimally incrementing x ; the resulting change is of a high order in the increment. This is the principle of how we can minimize (or maximize) a function by finding the zero of its derivative. Thus, in order to minimize $J(\mathbf{R})$, we only need to find an \mathbf{R} such that its infinitesimal variation does not change the value of $J(\mathbf{R})$ except for high order terms.

This consideration implies that “differentiation” of $J(\mathbf{R})$ with respect to \mathbf{R} means evaluation of the rate of the change of $J(\mathbf{R})$ when an infinitesimal rotation is added to \mathbf{R} . If an infinitesimal rotation of Eq. (10.19) is added to \mathbf{R} , we obtain

$$(\mathbf{I} + \mathbf{A}\delta t)\mathbf{R} = \mathbf{R} + \mathbf{A}\mathbf{R}\delta t. \quad (10.29)$$

The generator \mathbf{A} is represented by a vector $\boldsymbol{\omega}$ via Eq. (10.27). In the following, we combine the vector $\boldsymbol{\omega}$ and the parameter δt of infinitesimal variation as a single vector

$$\Delta\boldsymbol{\omega} = \boldsymbol{\omega}\delta t, \quad (10.30)$$

which we call the *small rotation vector*, an infinitesimal version of the finite rotation vector $\boldsymbol{\Omega}$ of Eq. (10.15). We also denote the antisymmetric matrix \mathbf{A} corresponding

to vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\top$ via Eq. (10.27) by¹ $\mathbf{A}(\boldsymbol{\omega})$. As shown in Eq. (10.6), the following identity holds for an arbitrary vector \mathbf{a} :

$$\mathbf{A}(\boldsymbol{\omega})\mathbf{a} = \boldsymbol{\omega} \times \mathbf{a}. \quad (10.31)$$

Using this notation, we can write Eq. (10.29) as $\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R}$ in terms of a small rotation vector $\Delta\boldsymbol{\omega}$. We substitute this into $J(\mathbf{R})$. If $J(\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R})$ is written in the form

$$J(\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R}) = J(\mathbf{R}) + \langle \mathbf{g}, \Delta\boldsymbol{\omega} \rangle, \quad (10.32)$$

for some vector \mathbf{g} by ignoring higher order terms in $\Delta\boldsymbol{\omega}$ (recall that $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product of vectors \mathbf{a} and \mathbf{b}), we call \mathbf{g} the *gradient*, or the *first derivative*, of $J(\mathbf{R})$ with respect to \mathbf{R} .

Since \mathbf{g} should vanish at \mathbf{R} for which $J(\mathbf{R})$ takes its minimum, we need to solve $\mathbf{g} = \mathbf{0}$, but this is not easy in general. So, we do iterative search, starting from an initial value \mathbf{R} and successively modifying it so that $J(\mathbf{R})$ reduces. Note that the value of the gradient \mathbf{g} depends on \mathbf{R} , i.e., \mathbf{g} is a function of \mathbf{R} . If, after substituting $\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R}$ for \mathbf{R} in $\mathbf{g}(\mathbf{R})$, we can write

$$\mathbf{g}(\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R}) = \mathbf{g}(\mathbf{R}) + \mathbf{H}\Delta\boldsymbol{\omega}, \quad (10.33)$$

for some symmetric matrix \mathbf{H} by ignoring higher order terms in $\Delta\boldsymbol{\omega}$, we call the matrix \mathbf{H} the *Hessian*, or the *second derivative*, of $J(\mathbf{R})$ with respect to \mathbf{R} . If the gradient \mathbf{g} and the Hessian \mathbf{H} are given, the value of $J(\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R})$ is approximated in the form

$$J(\mathbf{R} + \mathbf{A}(\Delta\boldsymbol{\omega})\mathbf{R}) = J(\mathbf{R}) + \langle \mathbf{g}, \Delta\boldsymbol{\omega} \rangle + \frac{1}{2}\langle \Delta\boldsymbol{\omega}, \mathbf{H}\Delta\boldsymbol{\omega} \rangle \quad (10.34)$$

by ignoring higher order terms in $\Delta\boldsymbol{\omega}$.

Now, we regard the “current” \mathbf{R} as a fixed constant and regard the above expression as a function of $\Delta\boldsymbol{\omega}$. Since this is a quadratic polynomial in $\Delta\boldsymbol{\omega}$, its derivative with respect to $\Delta\boldsymbol{\omega}$ is $\mathbf{g} + \mathbf{H}\Delta\boldsymbol{\omega}$. Hence, this polynomial in $\Delta\boldsymbol{\omega}$ takes its minimum for

$$\Delta\boldsymbol{\omega} = -\mathbf{H}^{-1}\mathbf{g}. \quad (10.35)$$

Namely, the rotation for which Eq. (10.34) takes its minimum is approximately $(\mathbf{I} + \mathbf{A}(\Delta\boldsymbol{\omega}))\mathbf{R}$ for that $\Delta\boldsymbol{\omega}$ (recall that the current value \mathbf{R} is regarded as a fixed constant). However, $\mathbf{I} + \mathbf{A}(\Delta\boldsymbol{\omega})$ is not an exact rotation matrix, although the discrepancy is of higher order in δt . To make it an exact rotation matrix, we add higher order correction terms as an infinite series expansion in the form of Eq. (10.12). Thus,

¹Some authors write this as $[\boldsymbol{\omega}]_\times$ or $(\boldsymbol{\omega} \times)$.

the rotation matrix for which Eq. (10.34) takes its minimum is approximated by $e^{A(\Delta\omega)}\mathbf{R}$. Regarding this as the “new” value of the current rotation, we repeat this process. The procedure is described as follows.

1. Provide an initial value for \mathbf{R} .
2. Compute the gradient \mathbf{g} and the Hessian \mathbf{H} of $J(\mathbf{R})$.
3. Solve the following linear equation in $\Delta\omega$:

$$\mathbf{H}\Delta\omega = -\mathbf{g}. \quad (10.36)$$

4. Update \mathbf{R} in the form

$$\mathbf{R} \leftarrow e^{A(\Delta\omega)}\mathbf{R}. \quad (10.37)$$

5. If $\|\Delta\omega\| \approx 0$, return \mathbf{R} and stop. Else, go back to Step 2.

This is nothing but the well-known Newton iterations. For Newton iterations, we approximate the object function by a quadratic polynomial in the neighborhood of the current argument, proceed to the value that gives the minimum of that quadratic approximation, and repeat this. The difference of the above procedure from the usual Newton iterations is that we analyze the minimum of the quadratic approximation *not* in the space of the rotation \mathbf{R} but *in the Lie algebra of infinitesimal rotations*. As we noted earlier, the space of \mathbf{R} and its Lie algebra are not the same, having higher order discrepancies.

We can think of this situation as follows. Imagine the set of all rotations, defined by the “nonlinear” constraint $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ and $|\mathbf{R}| = 1$ (recall that $|\mathbf{R}|$ denotes the determinant), which is called the *special orthogonal group*² of dimension 3, or the *group of rotations* for short, and denoted by $SO(3)$. This is a “curved space” in the 9-dimensional space of the elements of \mathbf{R} . The Lie algebra of infinitesimal rotations defined by the “linear” constraint $\mathbf{A} + \mathbf{A}^\top = \mathbf{O}$ can be thought of as a “flat” *tangent space* to it at the current \mathbf{R} , which we denote by $T_{\mathbf{R}}(SO(3))$, parameterized by $(\Delta\omega_1, \Delta\omega_2, \Delta\omega_3)$ with the origin $(0, 0, 0)$ at \mathbf{R} . We “project” a point in the Lie algebra $T_{\mathbf{R}}(SO(3))$ to a nearby point of $SO(3)$ by the exponential mapping $e^{A(\Delta\omega)}$ of Eq. (10.12) (Fig. 10.2) (see, e.g., [11]). Hereafter, we call this scheme of optimization the *Lie algebra method*.

Note that in actual computation, we need not compute the series expansion of Eq. (10.12) in Eq. (10.37). Let $\Delta\Omega = \|\Delta\omega\|$ and $\mathbf{l} = \mathcal{N}[\Delta\omega]$, where $\mathcal{N}[\mathbf{a}]$ denotes normalization to unit norm: $\mathcal{N}[\mathbf{a}] \equiv \mathbf{a}/\|\mathbf{a}\|$. As mentioned in Sect. 10.3, we can write $e^{A(\Delta\omega)} = \mathbf{R}_{\mathbf{l}}(\Delta\Omega)$, i.e., the rotation of angle $\Delta\Omega$ around axis \mathbf{l} , which can be computed using the Rodrigues formula of Eq. (10.14).

The criterion $\|\Delta\omega\| \approx 0$ for convergence is set by a predetermined threshold. If $\Delta\omega$ is $\mathbf{0}$, Eq. (10.35) implies $\mathbf{g} = \mathbf{0}$, producing a local minimum of $J(\mathbf{R})$. In general, iterative methods of this type are not necessarily guaranteed to converge when started

²The term “special” means that the determinant is constrained to be 1.

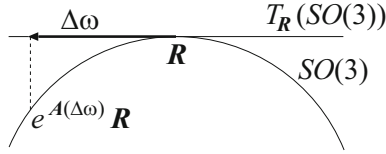
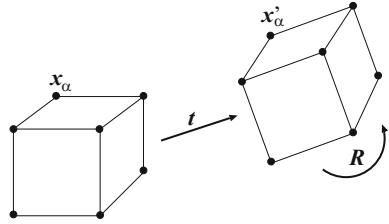


Fig. 10.2 The Lie algebra of infinitesimal rotations can be thought of as the tangent space $T_R(SO(3))$ to the group of rotations $SO(3)$ at R . The increment $\Delta\omega$ in the Lie algebra is projected to the point $e^{A(\Delta\omega)}R$ of $SO(3)$

Fig. 10.3 Observing N points $\{x_\alpha\}$ moving to $\{x'_\alpha\}$, we want to know their translation t and the rotation R



from an arbitrary initial value (some methods are guaranteed, though). Hence, we need to start the iterations from a value close to the desired solution.

10.6 Rotation Estimation by Maximum Likelihood

Given two sets of 3D points x_1, \dots, x_N and x'_1, \dots, x'_N obtained by 3D sensing, we want to know the rigid (or Euclidean) motion between them (Fig. 10.3). A rigid motion consists of a translation t and a rotation R . Translation is easily computed by comparing the centroids of the N points before and after the motion:

$$x_C = \frac{1}{N} \sum_{\alpha=1}^N x_\alpha, \quad x'_C = \frac{1}{N} \sum_{\alpha=1}^N x'_\alpha. \tag{10.38}$$

Let a_α and a'_α be the displacements of x_α and x'_α from their respective centroids:

$$a_\alpha = x_\alpha - x_C, \quad a'_\alpha = x'_\alpha - x'_C. \tag{10.39}$$

The translation is given by $t = x'_C - x_C$, and the rotation R is estimated so that $a'_\alpha \approx R a_\alpha, \alpha = 1, \dots, N$, holds as accurately as possible. We formulate this problem as follows.

We regard the data vectors a_α and a'_α as displaced from their true values \bar{a}_α and \bar{a}'_α by noise and write

$$a_\alpha = \bar{a}_\alpha + \Delta a_\alpha, \quad a'_\alpha = \bar{a}'_\alpha + \Delta a'_\alpha. \quad \alpha = 1, \dots, N. \tag{10.40}$$

We view $\Delta \mathbf{a}_\alpha$ and $\Delta \mathbf{a}'_\alpha$ as independent Gaussian random variables with mean $\mathbf{0}$ and covariance matrices $V[\mathbf{a}_\alpha]$ and $V[\mathbf{a}'_\alpha]$, respectively. We write

$$V[\mathbf{a}_\alpha] = \sigma^2 V_0[\mathbf{a}_\alpha], \quad V[\mathbf{a}'_\alpha] = \sigma^2 V_0[\mathbf{a}'_\alpha], \quad (10.41)$$

and call $V_0[\mathbf{a}_\alpha]$ and $V_0[\mathbf{a}'_\alpha]$ the *normalized covariance matrices* and σ the *noise level*. The normalized covariance matrices describe the directional noise properties that reflect the characteristics of the 3D sensing, which we assume is known, while the noise level, which indicates the absolute noise magnitude, is unknown. Thus, the probability density of $\Delta \mathbf{a}_\alpha$, $\Delta \mathbf{a}'_\alpha$, $\alpha = 1, \dots, N$, is written as

$$\begin{aligned} p &= \prod_{\alpha=1}^N \frac{e^{-\langle \Delta \mathbf{a}_\alpha, V_0[\mathbf{a}_\alpha]^{-1} \Delta \mathbf{a}_\alpha \rangle / 2\sigma^2}}{\sqrt{(2\pi)^3 |V_0[\mathbf{a}_\alpha]|} \sigma^3} \frac{e^{-\langle \Delta \mathbf{a}'_\alpha, V_0[\mathbf{a}'_\alpha]^{-1} \Delta \mathbf{a}'_\alpha \rangle / 2\sigma^2}}{\sqrt{(2\pi)^3 |V_0[\mathbf{a}'_\alpha]|} \sigma^3} \\ &= \frac{e^{-\sum_{\alpha=1}^N (\langle \mathbf{a}_\alpha - \bar{\mathbf{a}}_\alpha, V_0[\mathbf{a}_\alpha](\mathbf{a}_\alpha - \bar{\mathbf{a}}_\alpha) \rangle + \langle \mathbf{a}'_\alpha - \bar{\mathbf{a}}'_\alpha, V_0[\mathbf{a}'_\alpha](\mathbf{a}'_\alpha - \bar{\mathbf{a}}'_\alpha) \rangle) / 2\sigma^2}}{\prod_{\alpha=1}^N (2\pi)^3 \sqrt{|V_0[\mathbf{a}_\alpha]| |V_0[\mathbf{a}'_\alpha]|} \sigma^6}. \end{aligned} \quad (10.42)$$

When regarded as a function of observations \mathbf{a}_α , \mathbf{a}'_α , $\alpha = 1, \dots, N$, this expression is called their *likelihood*. *Maximum likelihood estimation* means computing the values $\bar{\mathbf{a}}_\alpha$, $\bar{\mathbf{a}}'_\alpha$, $\alpha = 1, \dots, N$, and \mathbf{R} that minimize this subject to

$$\bar{\mathbf{a}}'_\alpha = \mathbf{R} \bar{\mathbf{a}}_\alpha, \quad \alpha = 1, \dots, N. \quad (10.43)$$

This is equivalent to minimizing

$$J = \frac{1}{2} \sum_{\alpha=1}^N (\langle \mathbf{a}_\alpha - \bar{\mathbf{a}}_\alpha, V_0[\mathbf{a}_\alpha](\mathbf{a}_\alpha - \bar{\mathbf{a}}_\alpha) \rangle + \langle \mathbf{a}'_\alpha - \bar{\mathbf{a}}'_\alpha, V_0[\mathbf{a}'_\alpha](\mathbf{a}'_\alpha - \bar{\mathbf{a}}'_\alpha) \rangle), \quad (10.44)$$

which is called the *Mahalanobis distance*, often called the *reprojection error* in the computer vision community, subject to Eq. (10.43). Introducing Lagrange multipliers for the constraint of Eq. (10.43) and eliminating $\bar{\mathbf{a}}_\alpha$ and $\bar{\mathbf{a}}'_\alpha$, we can rewrite Eq. (10.44) in the form

$$J = \frac{1}{2} \sum_{\alpha=1}^N \langle \mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha, \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle, \quad (10.45)$$

where we put

$$\mathbf{V}_\alpha = \mathbf{R} V_0[\mathbf{a}_\alpha] \mathbf{R}^\top + V_0[\mathbf{a}'_\alpha], \quad (10.46)$$

and define the matrix \mathbf{W}_α by

$$\mathbf{W}_\alpha = \mathbf{V}_\alpha^{-1}. \quad (10.47)$$

We see that for maximum likelihood estimation, *we need not know the unknown noise level σ* , i.e., it is sufficient to know the covariance matrices up to scale.

If the noise characteristics are the same for all the data, the distribution is said to be *homogeneous*, otherwise it is *inhomogeneous*. If the noise occurrence is the same in all directions, the distribution is said to be *isotropic*, otherwise it is *anisotropic*. When the noise distribution is homogeneous and isotropic, we can let $V_0[\mathbf{a}_\alpha] = V_0[\mathbf{a}'_\alpha] = \mathbf{I}$, which means $\mathbf{V}_\alpha = 2\mathbf{I}$ and $\mathbf{W}_\alpha = \mathbf{I}/2$. Hence, minimizing Eq. (10.45) is equivalent to minimizing $\sum_{\alpha=1}^N \|\mathbf{a}'_\alpha - \mathbf{R}\mathbf{a}_\alpha\|^2$, which is known as *least-squares estimation* or the *Procrustes problem*. In this case, the solution can be analytically obtained. For nondegenerate data distributions, Arun et al. [1] showed that the solution is directly given using the singular value decomposition (SVD), and Kanatani [12] generalized it to include degenerate distributions. Horn [10] showed an alternative method, using the quaternion representation of rotations, which also works for degenerate distributions.

However, the noise distribution of 3D sensing for computer vision applications is hardly homogeneous or isotropic. Today, various types of 3D sensor are available including stereo vision and laser or ultrasonic emission, and they are used in such applications as manufacturing inspection, human body measurement, archeological measurement, camera autofocusing, and autonomous navigation [3, 20, 21]. Recently, an easy-to-use device called “kinect” is popular. For all such devices, the accuracy in the depth direction (e.g., the direction of the camera lens axis or laser/ultrasonic emission) is different from that in the direction orthogonal to it. The covariance matrix of 3D sensing by stereo vision can be analytically evaluated from the camera setting configuration. Many 3D sensor manufacturers provide the covariance of their devices. Here, we consider minimization of Eq. (10.45) for inhomogeneous and anisotropic noise distribution with known (up to scale) covariance matrices.

This problem was first solved by Ohta and Kanatani [18] by combining the quaternion representation of rotations and a scheme of iterating eigenvalue computation called *renormalization*. Later, Kanatani and Matsunaga [15] solved the same problem by a method called *extended FNS (Fundamental Numerical Scheme)*, which also iterates eigenvalue computation but can be applied not to just rotation but to all subgroups of affine transformations including rigid motions and similarities. They used their scheme for land deformation analysis, using GPS measurement. The GPS land surface measurement data and their covariance matrices are available on the websites of government agencies. Here, we show how the Lie algebra method works for minimizing Eq. (10.45).

Replacing \mathbf{R} by $\mathbf{R} + \mathbf{A}(\Delta\omega)\mathbf{R}$ in Eq. (10.45), we see that the linear increment of J is given by

$$\begin{aligned} \Delta J = & - \sum_{\alpha=1}^N \langle A(\Delta\omega) \mathbf{R} \mathbf{a}_\alpha, \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle \\ & + \frac{1}{2} \sum_{\alpha=1}^N \langle \mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha, \Delta \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle, \end{aligned} \quad (10.48)$$

where we have noted that the right side of Eq. (10.45) is symmetric with respect to the two \mathbf{R} 's in the expression so that we only need to consider the increment of one \mathbf{R} and multiply the result by 2. Using the identity of Eq. (10.31), we can write the first term on the right side of Eq. (10.48) as

$$- \sum_{\alpha=1}^N \langle \Delta\omega \times \mathbf{R} \mathbf{a}_\alpha, \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle = - \langle \Delta\omega, \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle, \quad (10.49)$$

where we have used the identity $\langle \mathbf{a} \times \mathbf{b}, \mathbf{c} \rangle = \langle \mathbf{a}, \mathbf{b} \times \mathbf{c} \rangle$. For evaluating $\Delta \mathbf{W}_\alpha$ in the second term on the right side of Eq. (10.48), we rewrite Eq. (10.47) as $\mathbf{W}_\alpha \mathbf{V}_\alpha = \mathbf{I}$, from which we obtain $\Delta \mathbf{W}_\alpha \mathbf{V}_\alpha + \mathbf{W}_\alpha \Delta \mathbf{V}_\alpha = \mathbf{O}$. Using Eq. (10.47) again, we can write $\Delta \mathbf{W}_\alpha$ as

$$\Delta \mathbf{W}_\alpha = -\mathbf{W}_\alpha \Delta \mathbf{V}_\alpha \mathbf{W}_\alpha. \quad (10.50)$$

From Eq. (10.46), we obtain

$$\Delta \mathbf{W}_\alpha = -\mathbf{W}_\alpha (A(\Delta\omega) \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top + \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] (A(\Delta\omega) \mathbf{R})^\top) \mathbf{W}_\alpha, \quad (10.51)$$

which we substitute into the second term on the right side of Eq. (10.48). Note that the two terms on the right side of Eq. (10.51) are transpose of each other and that the second term on the right side of Eq. (10.48) is a quadratic form in $\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha$. Hence, we only need to consider one term of Eq. (10.51) and multiply the result by 2. Then, the second term on the right side of Eq. (10.48) is written as

$$\begin{aligned} & - \sum_{\alpha=1}^N \langle \mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha, \mathbf{W}_\alpha A(\Delta\omega) \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle \\ = & - \sum_{\alpha=1}^N \langle \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha), \Delta\omega \times \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle \\ = & \sum_{\alpha=1}^N \langle \Delta\omega, (\mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha)) \times \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle. \end{aligned} \quad (10.52)$$

Combining this with Eq. (10.49), we can write Eq. (10.48) as

$$\begin{aligned} \Delta J = & - \sum_{\alpha=1}^N \langle \Delta \boldsymbol{\omega}, (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \\ & - (\mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha)) \times \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \rangle \end{aligned} \quad (10.53)$$

Hence, from Eq. (10.32), the gradient of the function $J(\mathbf{R})$ of Eq. (10.45) is given by

$$\begin{aligned} \mathbf{g} = & - \sum_{\alpha=1}^N \left((\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) - (\mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha)) \right. \\ & \left. \times \mathbf{R} \mathbf{V}[\mathbf{a}_\alpha] \mathbf{R}^\top \mathbf{W}_\alpha (\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha) \right). \end{aligned} \quad (10.54)$$

Next, we consider the linear increment resulting from replacing \mathbf{R} by $\mathbf{R} + \mathbf{A}(\Delta \boldsymbol{\omega}) \mathbf{R}$ in this equation. Since we are computing an \mathbf{R} such that $\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha \approx \mathbf{0}$, we can ignore the increment of the first \mathbf{R} in the first term on the right side of Eq. (10.54), assuming that $\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha \approx \mathbf{0}$ as the iterations proceed. The second term is quadratic in $\mathbf{a}'_\alpha - \mathbf{R} \mathbf{a}_\alpha$, so we can ignore it. Only considering the increment of the second \mathbf{R} in the first term, we obtain

$$\begin{aligned} \Delta \mathbf{g} &= \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha \mathbf{A}(\Delta \boldsymbol{\omega}) \mathbf{R} \mathbf{a}_\alpha = \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha (\Delta \boldsymbol{\omega} \times (\mathbf{R} \mathbf{a}_\alpha)) \\ &= - \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha ((\mathbf{R} \mathbf{a}_\alpha) \times \Delta \boldsymbol{\omega}). \end{aligned} \quad (10.55)$$

Now, we introduce new notations. For a vector $\boldsymbol{\omega}$ and a matrix \mathbf{T} , we define

$$\boldsymbol{\omega} \times \mathbf{T} \equiv \mathbf{A}(\boldsymbol{\omega}) \mathbf{T}, \quad \mathbf{T} \times \boldsymbol{\omega} \equiv \mathbf{T} \mathbf{A}(\boldsymbol{\omega})^\top, \quad \boldsymbol{\omega} \times \mathbf{T} \times \boldsymbol{\omega} \equiv \mathbf{A}(\boldsymbol{\omega}) \mathbf{T} \mathbf{A}(\boldsymbol{\omega})^\top. \quad (10.56)$$

The last one is the combination of the first two; whichever \times we evaluate first, we obtain the same result. From Eq. (10.31), it is easily seen that $\boldsymbol{\omega} \times \mathbf{T}$ is “the matrix whose columns are the vector products of $\boldsymbol{\omega}$ and the three columns of \mathbf{T} ” and that $\mathbf{T} \times \boldsymbol{\omega}$ is “the matrix whose rows are the vector products of the three rows of \mathbf{T} and $\boldsymbol{\omega}$ ” (see [13, 16] for more about this notation). Using this notation and Eq. (10.31), we can write Eq. (10.55) as

$$\Delta \mathbf{g} = - \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha \mathbf{A}(\mathbf{R} \mathbf{a}_\alpha) \Delta \boldsymbol{\omega} = \sum_{\alpha=1}^N (\mathbf{R} \mathbf{a}_\alpha) \times \mathbf{W}_\alpha \times (\mathbf{R} \mathbf{a}_\alpha) \Delta \boldsymbol{\omega}, \quad (10.57)$$

where we have noted that $\mathbf{A}(\boldsymbol{\omega})$ is antisymmetric: $\mathbf{A}(\boldsymbol{\omega})^\top = -\mathbf{A}(\boldsymbol{\omega})$. Comparing this and Eq. (10.33), we obtain the Hessian in the form

$$\mathbf{H} = \sum_{\alpha=1}^N (\mathbf{R}\mathbf{a}_{\alpha}) \times \mathbf{W}_{\alpha} \times (\mathbf{R}\mathbf{a}_{\alpha}). \quad (10.58)$$

Now that the gradient \mathbf{g} and the Hessian \mathbf{H} are given by Eqs. (10.54) and (10.58), we can minimize $J(\mathbf{R})$ by Newton iterations as described in the preceding section.

However, we have approximated the Hessian \mathbf{H} by letting some quantities be zero in the course of the computation for minimizing those quantities. This convention is called *Gauss–Newton approximation*, and the Newton iterations using Gauss–Newton approximation are called *Gauss–Newton iterations*. From Eq. (10.35), we see that if $\Delta\omega$ is $\mathbf{0}$ at the time of convergence, $\mathbf{g} = \mathbf{0}$ holds irrespective of the value of \mathbf{H} , returning an exact solution. In other words, *as long as the gradient \mathbf{g} is correctly computed*, the Hessian \mathbf{H} need not be exact. However, the value of \mathbf{H} affects the speed of convergence.

If the Hessian \mathbf{H} is not appropriate, we may overstep the minimum of $J(\mathbf{R})$ and the value of $J(\mathbf{R})$ may increase. Or we may proceed too slowly to reduce $J(\mathbf{R})$ meaningfully. A well-known measure to cope with this is add to \mathbf{H} a multiple of the identity matrix \mathbf{I} and adjust the constant c of $\mathbf{H} + c\mathbf{I}$. To be specific, we decrease c as long as $J(\mathbf{R})$ decreases and increase c if $J(\mathbf{R})$ increases. This modification is known as the *Levenberg–Marquardt method*. The procedure is written as follows (see, e.g., [19]).

1. Initialize \mathbf{R} , and let $c = 0.0001$.
2. Compute the gradient \mathbf{g} and the (Gauss–Newton approximated) Hessian \mathbf{H} of $J(\mathbf{R})$.
3. Solve the following linear equation in $\Delta\omega$:

$$(\mathbf{H} + c\mathbf{I})\Delta\omega = -\mathbf{g}. \quad (10.59)$$

4. Tentatively update \mathbf{R} to

$$\tilde{\mathbf{R}} = e^{A(\Delta\omega)} \mathbf{R}. \quad (10.60)$$

5. If $J(\tilde{\mathbf{R}}) < J(\mathbf{R})$ or $J(\tilde{\mathbf{R}}) \approx J(\mathbf{R})$ is not satisfied, let $c \leftarrow 10c$ and go back to Step 3.
6. If $\|\Delta\omega\| \approx 0$, return $\tilde{\mathbf{R}}$ and stop. Else, update $\mathbf{R} \leftarrow \tilde{\mathbf{R}}$, $c \leftarrow c/10$ and go back to Step 2.

If we let $c = 0$, this reduces to Gauss–Newton iterations. In Steps 1, 5, and 6, the values 0.0001, $10c$, and $c/10$ are all empirical. To start the iterations, we need appropriate initial values, for which we can use the analytical homogeneous and isotropic noise solution [1, 10, 12]. The initial solution is sufficiently accurate in most practical applications, so the above Levenberg–Marquardt iterations usually converge after a few iterations.

10.7 Fundamental Matrix Computation

Consider two images of the scene taken by two cameras. Suppose a point in the scene is imaged at (x, y) in the first camera image and at (x', y') in the second camera image. From the geometry of perspective imaging, the following *epipolar equation* holds [9]:

$$\left\langle \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \mathbf{F} \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix} \right\rangle = 0, \tag{10.61}$$

where f_0 is an arbitrary scale constant; theoretically, we could set it to one, but it is better to let it have the magnitude of x/f and y/f for numerical stability of finite length computation [8]. The matrix \mathbf{F} is called the *fundamental matrix* and is determined from the relative configuration of the two cameras and their internal parameters such as the focal length.

Computing the fundamental matrix \mathbf{F} from point correspondences (x_α, y_α) and (x'_α, y'_α) , $\alpha = 1, \dots, N$, is one of the most fundamental steps of computer vision (Fig. 10.4). From the computed \mathbf{F} , we can reconstruct the 3D structure of the scene (see, e.g., [9, 16]). The basic principle of its computation is minimizing the following function:

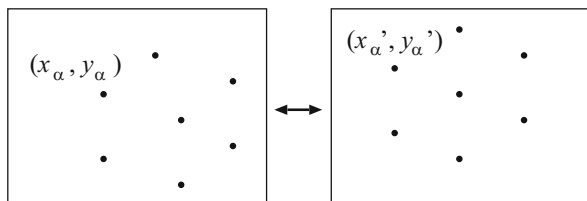
$$J(\mathbf{F}) = \frac{f_0^2}{2} \sum_{\alpha=1}^N \frac{\langle \mathbf{x}_\alpha, \mathbf{F} \mathbf{x}'_\alpha \rangle^2}{\|\mathbf{P}_k \mathbf{F} \mathbf{x}'_\alpha\|^2 + \|\mathbf{P}_k \mathbf{F}^\top \mathbf{x}_\alpha\|^2}, \tag{10.62}$$

where we define

$$\mathbf{x}_\alpha = \begin{pmatrix} x_\alpha/f_0 \\ y_\alpha/f_0 \\ 1 \end{pmatrix}, \quad \mathbf{x}'_\alpha = \begin{pmatrix} x'_\alpha/f_0 \\ y'_\alpha/f_0 \\ 1 \end{pmatrix}, \quad \mathbf{P}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{10.63}$$

By minimizing Eq. (10.62), we can obtain a maximum likelihood solution to a high accuracy, assuming that the noise terms $\Delta x_\alpha, \Delta y_\alpha, \Delta x'_\alpha,$ and $\Delta y'_\alpha$ in the coordinates (x_α, y_α) and (x'_α, y'_α) are Gaussian variables of mean 0 with a constant variance. The function $J(\mathbf{F})$ of Eq. (10.62) is called the *Sampson error* [9, 16].

Fig. 10.4 We compute the fundamental matrix \mathbf{F} from point correspondences of two images



Evidently, the fundamental matrix \mathbf{F} has scale indeterminacy: Eqs. (10.61) and (10.62) are unchanged if \mathbf{F} is multiplied by an arbitrary nonzero constant. We normalized it to $\|\mathbf{F}\|^2 (\equiv \sum_{i,j=1}^3 F_{ij}^2) = 1$. Besides, there is an important requirement, called the *rank constraint* ([9, 16]): \mathbf{F} must have rank 2. Many strategies have been proposed to impose this constraint (see [16]), but the most straightforward one is to express \mathbf{F} via the SVD in the form

$$\mathbf{F} = \mathbf{U} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^\top, \quad (10.64)$$

where \mathbf{U} and \mathbf{V} are orthogonal matrices, and $\sigma_1 \geq \sigma_2 (> 0)$ are the singular values; letting the third singular value σ_3 be 0 is the rank constraint. From the normalization $\|\mathbf{F}\|^2 = 1$, we have $\sigma_1^2 + \sigma_2^2 = 1$, so we can let

$$\sigma_1 = \cos \phi, \quad \sigma_2 = \sin \phi. \quad (10.65)$$

Substituting Eq. (10.64) into Eq. (10.62), we minimize $J(\mathbf{F})$ with respect to \mathbf{U} , \mathbf{V} , and ϕ . This parameterization was first proposed by Bartoli and Sturm [2], to which Sugaya and Kanatani [25] applied the Lie algebra method.

Note that \mathbf{U} and \mathbf{V} are orthogonal matrices; they may not represent rotations depending on the sign of the determinant. However, *a small variation of an orthogonal matrix is a small rotation*. Hence, we can express the small variations of \mathbf{U} and \mathbf{V} in the form

$$\Delta \mathbf{U} = \mathbf{A}(\Delta \omega_U) \mathbf{U}, \quad \Delta \mathbf{V} = \mathbf{A}(\Delta \omega_V) \mathbf{V}, \quad (10.66)$$

in terms of small rotation vectors $\Delta \omega_U = (\Delta \omega_{iU})$ and $\Delta \omega_V = (\Delta \omega_{iV})$. Incrementing \mathbf{U} , \mathbf{V} , and ϕ to $\mathbf{U} + \Delta \mathbf{U}$, $\mathbf{V} + \Delta \mathbf{V}$, and $\phi + \Delta \phi$ in Eq. (10.64), we can write the linear increment of \mathbf{F} , ignoring higher order terms, in the form

$$\begin{aligned} \Delta \mathbf{F} &= \mathbf{A}(\Delta \omega_U) \mathbf{U} \text{diag}(\cos \phi, \sin \phi, 0) \mathbf{V}^\top + \mathbf{U} \text{diag}(\cos \phi, \sin \phi, 0) (\mathbf{A}(\Delta \omega_V) \mathbf{V})^\top \\ &\quad + \mathbf{U} \text{diag}(-\sin \phi, \cos \phi, 0) \mathbf{V}^\top \Delta \phi. \end{aligned} \quad (10.67)$$

Taking out individual elements, we obtain

$$\begin{aligned} \Delta F_{11} &= \Delta \omega_{2U} F_{31} - \Delta \omega_{3U} F_{21} + \Delta \omega_{2V} F_{13} - \Delta \omega_{3V} F_{12} \\ &\quad + (U_{12} V_{12} \cos \phi - U_{11} V_{11} \sin \phi) \Delta \phi, \\ \Delta F_{12} &= \Delta \omega_{2U} F_{32} - \Delta \omega_{3U} F_{22} + \Delta \omega_{3V} F_{11} - \Delta \omega_{1V} F_{13} \\ &\quad + (U_{12} V_{22} \cos \phi - U_{11} V_{21} \sin \phi) \Delta \phi, \\ &\quad \vdots \end{aligned}$$

$$\begin{aligned} \Delta F_{33} &= \Delta\omega_{1U}F_{23} - \Delta\omega_{2U}F_{13} + \Delta\omega_{1V}F_{32} - \Delta\omega_{2V}F_{31} \\ &\quad + (U_{32}V_{32} \cos \phi - U_{31}V_{31} \sin \phi)\Delta\phi. \end{aligned} \quad (10.68)$$

We identify $\Delta \mathbf{F}$ with a 9-dimensional vector consisting of components $\Delta F_{11}, \Delta F_{12}, \dots, \Delta F_{33}$ and write

$$\Delta \mathbf{F} = \mathbf{F}_U \Delta \boldsymbol{\omega}_U + \mathbf{F}_V \Delta \boldsymbol{\omega}_V + \boldsymbol{\theta}_\phi \Delta \phi, \quad (10.69)$$

where we define the 9×3 matrices \mathbf{F}_U and \mathbf{F}_V and the 9-dimensional vector $\boldsymbol{\theta}_\phi$ by

$$\mathbf{F}_U = \begin{pmatrix} 0 & F_{31} & -F_{21} \\ 0 & F_{32} & -F_{22} \\ 0 & F_{33} & -F_{23} \\ -F_{31} & 0 & F_{11} \\ -F_{32} & 0 & F_{12} \\ -F_{33} & 0 & F_{13} \\ F_{21} & -F_{11} & 0 \\ F_{22} & -F_{12} & 0 \\ F_{23} & -F_{13} & 0 \end{pmatrix}, \quad \mathbf{F}_V = \begin{pmatrix} 0 & F_{13} & -F_{12} \\ -F_{13} & 0 & F_{11} \\ F_{12} & -F_{11} & 0 \\ 0 & F_{23} & -F_{22} \\ -F_{23} & 0 & F_{21} \\ F_{22} & -F_{21} & 0 \\ 0 & F_{33} & -F_{32} \\ -F_{33} & 0 & F_{31} \\ F_{32} & -F_{31} & 0 \end{pmatrix}, \quad (10.70)$$

$$\boldsymbol{\theta}_\phi = \begin{pmatrix} \sigma_1 U_{12} V_{12} - \sigma_2 U_{11} V_{11} \\ \sigma_1 U_{12} V_{22} - \sigma_2 U_{11} V_{21} \\ \sigma_1 U_{12} V_{32} - \sigma_2 U_{11} V_{31} \\ \sigma_1 U_{22} V_{12} - \sigma_2 U_{21} V_{11} \\ \sigma_1 U_{22} V_{22} - \sigma_2 U_{21} V_{21} \\ \sigma_1 U_{22} V_{32} - \sigma_2 U_{21} V_{31} \\ \sigma_1 U_{32} V_{12} - \sigma_2 U_{31} V_{11} \\ \sigma_1 U_{32} V_{22} - \sigma_2 U_{31} V_{21} \\ \sigma_1 U_{32} V_{32} - \sigma_2 U_{31} V_{31} \end{pmatrix}. \quad (10.71)$$

Then, the linear increment ΔJ of the function $J(\mathbf{F})$ of Eq. (10.62) is given by

$$\begin{aligned} \Delta J &= \langle \nabla_{\mathbf{F}} J, \Delta \mathbf{F} \rangle = \langle \nabla_{\mathbf{F}} J, \mathbf{F}_U \Delta \boldsymbol{\omega}_U \rangle + \langle \nabla_{\mathbf{F}} J, \mathbf{F}_V \Delta \boldsymbol{\omega}_V \rangle + \langle \nabla_{\mathbf{F}} J, \boldsymbol{\theta}_\phi \Delta \phi \rangle \\ &= \langle \mathbf{F}_U^\top \nabla_{\mathbf{F}} J, \Delta \boldsymbol{\omega}_U \rangle + \langle \mathbf{F}_V^\top \nabla_{\mathbf{F}} J, \Delta \boldsymbol{\omega}_V \rangle + \langle \nabla_{\mathbf{F}} J, \boldsymbol{\theta}_\phi \rangle \Delta \phi, \end{aligned} \quad (10.72)$$

where $\nabla_{\mathbf{F}} J$ is the 9-dimensional vector consisting of components $\partial J / \partial F_{ij}$. From this, we obtain the gradients of J with respect to $\mathbf{U}_U, \mathbf{U}_V$, and ϕ as follows:

$$\nabla_{\boldsymbol{\omega}_U} J = \mathbf{F}_U^\top \nabla_{\mathbf{F}} J, \quad \nabla_{\boldsymbol{\omega}_V} J = \mathbf{F}_V^\top \nabla_{\mathbf{F}} J, \quad \frac{\partial J}{\partial \phi} = \langle \nabla_{\mathbf{F}} J, \boldsymbol{\theta}_\phi \rangle. \quad (10.73)$$

Next, consider the second derivatives $\partial^2 J / \partial F_{ij} \partial F_{kl}$ of Eq. (10.62). We adopt the Gauss–Newton approximation of ignoring terms containing $\langle \mathbf{x}_\alpha, \mathbf{F} \mathbf{x}'_\alpha \rangle$, i.e., the left side of the epipolar equation of Eq. (10.61). It follows that we need not consider terms containing $\langle \mathbf{x}_\alpha, \mathbf{F} \mathbf{x}'_\alpha \rangle^2$ in the first derivative, i.e., we need not differentiate the denominator in Eq. (10.62). Hence, the first derivative is approximated to be

$$\frac{\partial J}{\partial F_{ij}} \approx \sum_{\alpha=1}^2 \frac{f_0^2 x_{i\alpha} x'_{j\alpha} \langle \mathbf{x}_\alpha, \mathbf{F} \mathbf{x}'_\alpha \rangle}{\|\mathbf{P}_k \mathbf{F} \mathbf{x}'_\alpha\|^2 + \|\mathbf{P}_k \mathbf{F}^\top \mathbf{x}'_\alpha\|^2}, \quad (10.74)$$

where $x_{i\alpha}$ and $x'_{j\alpha}$ denote the i th components of \mathbf{x}_α and \mathbf{x}'_α , respectively. For differentiating this with respect to F_{kl} , we need not differentiate the denominator because the numerator contains $\langle \mathbf{x}_\alpha, \mathbf{F} \mathbf{x}'_\alpha \rangle$. Differentiating only the numerator, we obtain

$$\frac{\partial^2 J}{\partial F_{ij} \partial F_{kl}} \approx \sum_{\alpha=1}^2 \frac{f_0^2 x_{i\alpha} x'_{j\alpha} x_{k\alpha} x'_{l\alpha}}{\|\mathbf{P}_k \mathbf{F} \mathbf{x}'_\alpha\|^2 + \|\mathbf{P}_k \mathbf{F}^\top \mathbf{x}'_\alpha\|^2}. \quad (10.75)$$

Let us count the pairs of indices $(i, j) = (1, 1), (1, 2), \dots, (3, 3)$, using a single running index $I = 1, \dots, 9$. Similarly, we use a single running index $J = 1, \dots, 9$ for pairs (k, l) and regard the right side of the above equation as the (I, J) element of a 9×9 matrix, which we write as $\nabla_{\mathbf{F}}^2 J$. Then, as in Eq. (10.72), we can write, using Eq. (10.69), the second derivation of J with respect to \mathbf{U} , \mathbf{V} , and ϕ in the form

$$\begin{aligned} \Delta^2 J &= \langle \Delta \mathbf{F}, \nabla_{\mathbf{F}}^2 J \Delta \mathbf{F} \rangle \\ &= \langle \mathbf{F}_U \Delta \omega_U + \mathbf{F}_V \Delta \omega_V + \boldsymbol{\theta}_\phi \Delta \phi, \nabla_{\mathbf{F}}^2 J (\mathbf{F}_U \Delta \omega_U + \mathbf{F}_V \Delta \omega_V + \boldsymbol{\theta}_\phi \Delta \phi) \rangle \\ &= \langle \Delta \omega_U, \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_U \Delta \omega_U \rangle + \langle \Delta \omega_U, \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_V \Delta \omega_V \rangle \\ &\quad + \langle \Delta \omega_V, \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_U \Delta \omega_U \rangle + \langle \Delta \omega_V, \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_V \Delta \omega_V \rangle \\ &\quad + \langle \Delta \omega_U, \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle \Delta \phi + \langle \Delta \omega_V, \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle \Delta \phi \\ &\quad + \langle \Delta \omega_U, \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle \Delta \phi + \langle \Delta \omega_V, \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle \Delta \phi \\ &\quad + \langle \boldsymbol{\theta}_\phi, \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle \Delta \phi^2, \end{aligned} \quad (10.76)$$

from which we obtain the following second derivatives of J :

$$\begin{aligned} \nabla_{\omega_U \omega_U} J &= \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_U, & \nabla_{\omega_V \omega_V} J &= \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_V, & \nabla_{\omega_U \omega_V} J &= \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \mathbf{F}_V, \\ \frac{\partial \nabla_{\omega_U} J}{\partial \phi} &= \mathbf{F}_U^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi, & \frac{\partial \nabla_{\omega_V} J}{\partial \phi} &= \mathbf{F}_V^\top \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi, & \frac{\partial^2 J}{\partial \phi^2} &= \langle \boldsymbol{\theta}_\phi, \nabla_{\mathbf{F}}^2 J \boldsymbol{\theta}_\phi \rangle. \end{aligned} \quad (10.77)$$

Now that the first and second derivatives are given, the Levenberg–Marquardt procedure for minimizing J goes as follows:

1. Provide an initial value of \mathbf{F} such that $|\mathbf{F}| = 0$ and $\|\mathbf{F}\| = 1$, and compute the SVD of Eq. (10.64). Evaluate the value J of Eq. (10.62), and let $c = 0.0001$.
2. Compute the first and second derivatives $\nabla_{\mathbf{F}} J$ and (Gauss–Newton approximated) $\nabla_{\mathbf{F}}^2 J$ of J with respect to \mathbf{F} .
3. Compute the 9×3 matrices \mathbf{F}_U and \mathbf{F}_V of Eq. (10.70) and the 9-dimensional vector $\boldsymbol{\theta}_\phi$ of Eq. (10.71).
4. Compute the first derivatives $\nabla_{\omega_U} J$, $\nabla_{\omega_V} J$, and $\partial J / \partial \phi$ in Eq. (10.73) and the second derivatives $\nabla_{\omega_U \omega_U} J$, $\nabla_{\omega_V \omega_V} J$, $\nabla_{\omega_U \omega_V} J$, $\partial \nabla_{\omega_U} J / \partial \phi$, $\partial \nabla_{\omega_V} J / \partial \phi$, and $\partial^2 J / \partial \phi^2$ in Eq. (10.77) of J .
5. Solve the following linear equation in $\Delta \omega_U$, $\Delta \omega_V$, and $\Delta \phi$:

$$\begin{aligned} & \left(\begin{pmatrix} \nabla_{\omega_U \omega_U} J & \nabla_{\omega_U \omega_V} J & \partial \nabla_{\omega_U} J / \partial \phi \\ (\nabla_{\omega_U \omega_U} J)^\top & \nabla_{\omega_V \omega_V} J & \partial \nabla_{\omega_V} J / \partial \phi \\ (\partial \nabla_{\omega_U} J / \partial \phi)^\top & (\partial \nabla_{\omega_V} J / \partial \phi)^\top & \partial^2 J / \partial \phi^2 \end{pmatrix} + c\mathbf{I} \right) \begin{pmatrix} \Delta \omega_U \\ \Delta \omega_V \\ \Delta \phi \end{pmatrix} \\ & = - \begin{pmatrix} \nabla_{\omega_U} J \\ \nabla_{\omega_V} J \\ \partial J / \partial \phi \end{pmatrix}. \end{aligned} \quad (10.78)$$

6. Tentatively update \mathbf{U} , \mathbf{V} , and ϕ to

$$\tilde{\mathbf{U}} = e^{A(\Delta \omega_U)} \mathbf{U}, \quad \tilde{\mathbf{V}} = e^{A(\Delta \omega_V)} \mathbf{V}, \quad \tilde{\phi} = \phi + \Delta \phi. \quad (10.79)$$

7. Tentatively update \mathbf{F} to

$$\tilde{\mathbf{F}} = \tilde{\mathbf{U}} \begin{pmatrix} \cos \tilde{\phi} & 0 & 0 \\ 0 & \sin \tilde{\phi} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tilde{\mathbf{V}}^\top. \quad (10.80)$$

8. Let \tilde{J} be the value of Eq. (10.62) for $\tilde{\mathbf{F}}$.
9. If $\tilde{J} < J$ or $\tilde{J} \approx J$ is not satisfied, let $c \leftarrow 10c$ and go back to Step 5.
10. If $\tilde{\mathbf{F}} \approx \mathbf{F}$, return $\tilde{\mathbf{F}}$ and stop. Else, update $\mathbf{F} \leftarrow \tilde{\mathbf{F}}$, $\mathbf{U} \leftarrow \tilde{\mathbf{U}}$, $\mathbf{V} \leftarrow \tilde{\mathbf{V}}$, $\phi \leftarrow \tilde{\phi}$, $c \leftarrow c/10$, and $J \leftarrow \tilde{J}$ and go back to Step 2.

We need an initial value of \mathbf{F} for starting these iterations. Various simple schemes are known. The simplest one is the “least squares” that minimizes the square sum of the left side of the epipolar equation of Eq. (10.61), which is equivalent to ignoring the denominator on the left side of Eq. (10.62). Since the square sum is quadratic in \mathbf{F} , the solution is immediately obtained by eigenanalysis if the rank constraint is not considered. The rank constraint can be imposed by computing the SVD of the resulting \mathbf{F} and replacing the smallest singular value by 0. This scheme is known as *Hartley’s 8-point method* [8]. Hartley’s 8-point method is sufficiently accurate in most practical applications, so the above iterations usually converge after a few iterations. See Kanatani et al. [16] for experimental comparisons of how the above

method improves the accuracy over Hartley’s 8-point method; often the number of significant digits increases at least by one.

10.8 Bundle Adjustment

We consider the problem of reconstructing the 3D structure of the scene from multiple images taken by multiple cameras. One of the most fundamental methods is *bundle adjustment*: we optimally estimate all the 3D positions of the points we are viewing and all the postures of the cameras as well as their internal parameters, in such a way that the bundle of rays, or lines of sight, will piece through the images appropriately.

Consider points $(X_\alpha, Y_\alpha, Z_\alpha)$, $\alpha = 1, \dots, N$, in the scene. Suppose the α th point is viewed at $(x_{\alpha\kappa}, y_{\alpha\kappa})$ in the image of the κ th camera, $\kappa = 1, \dots, M$ (Fig. 10.5). The imaging geometry of most of today’s cameras is sufficiently modeled by perspective projection, for which the following relations hold [9]:

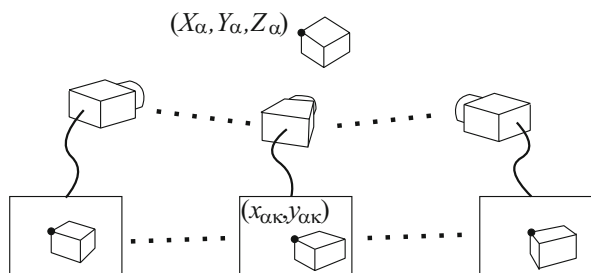
$$\begin{aligned} x_{\alpha\kappa} &= f_0 \frac{P_{\kappa(11)}X_\alpha + P_{\kappa(12)}Y_\alpha + P_{\kappa(13)}Z_\alpha + P_{\kappa(14)}}{P_{\kappa(31)}X_\alpha + P_{\kappa(32)}Y_\alpha + P_{\kappa(33)}Z_\alpha + P_{\kappa(34)}}, \\ y_{\alpha\kappa} &= f_0 \frac{P_{\kappa(21)}X_\alpha + P_{\kappa(22)}Y_\alpha + P_{\kappa(23)}Z_\alpha + P_{\kappa(24)}}{P_{\kappa(31)}X_\alpha + P_{\kappa(32)}Y_\alpha + P_{\kappa(33)}Z_\alpha + P_{\kappa(34)}}, \end{aligned} \tag{10.81}$$

where f_0 is the scale constant we used in Eq. (10.61), and $P_{\kappa(ij)}$ are constants determined by the position, orientation, and internal parameters (e.g., the focal length, the principal point position, and the image distortion description) of the κ th camera. We write the 3×4 matrix whose (i, j) element is $P_{\kappa(ij)}$ as \mathbf{P}_κ and call it the *camera matrix* of the κ th camera. From the geometry of perspective projection, we can write this in the form

$$\mathbf{P}_\kappa = \mathbf{K}_\kappa \mathbf{R}_\kappa^\top (\mathbf{I} - \mathbf{t}_\kappa), \tag{10.82}$$

where \mathbf{K}_κ is the 3×3 matrix, called the *intrinsic parameter matrix*, consisting of the internal parameters of the κ th camera [9]. The matrix \mathbf{R}_κ specifies the rotation

Fig. 10.5 N points in the scene are viewed by M cameras. The α th point $(X_\alpha, Y_\alpha, Z_\alpha)$ is imaged at point $(x_{\alpha\kappa}, y_{\alpha\kappa})$ in the κ th camera image



of the κ th camera relative to the world coordinate system fixed to the scene, and \mathbf{t}_κ is the position of the lens center of the κ th camera. The principle of bundle adjustment is to minimize

$$E = \sum_{\alpha=1}^N \sum_{\kappa=1}^M \left(\left(\frac{x_{\alpha\kappa}}{f_0} - \frac{P_{\kappa(11)}X_\alpha + P_{\kappa(12)}Y_\alpha + P_{\kappa(13)}Z_\alpha + P_{\kappa(14)}}{P_{\kappa(31)}X_\alpha + P_{\kappa(32)}Y_\alpha + P_{\kappa(33)}Z_\alpha + P_{\kappa(34)}} \right)^2 + \left(\frac{y_{\alpha\kappa}}{f_0} - \frac{P_{\kappa(21)}X_\alpha + P_{\kappa(22)}Y_\alpha + P_{\kappa(23)}Z_\alpha + P_{\kappa(24)}}{P_{\kappa(31)}X_\alpha + P_{\kappa(32)}Y_\alpha + P_{\kappa(33)}Z_\alpha + P_{\kappa(34)}} \right)^2 \right), \quad (10.83)$$

with respect to all the 3D positions $(X_\alpha, Y_\alpha, Z_\alpha)$ and all the camera matrices \mathbf{P}_κ from observed $(x_{\alpha\kappa}, y_{\alpha\kappa})$, $\alpha = 1, \dots, N$, $\kappa = 1, \dots, M$, as the input so that Eq. (10.81) holds as accurately as possible. The expression E , called the *reprojection error* [9], measures the square sum of the discrepancies between the image positions predicted by the perspective projection geometry and their actually observed image positions.

Various algorithms have been proposed for bundle adjustment and are now available on the Web. The best known is the *SBA* of Lourakis and Argyros [17]. Snavely et al. [23, 24] combined it with an image correspondence extraction process and offered a tool called *bundler*. Here, we slightly modify these algorithms, based on Kanatani et al. [16], to explicitly use the Lie algebra method for camera rotation optimization.

Letting

$$\begin{aligned} p_{\alpha\kappa} &= P_{\kappa(11)}X_\alpha + P_{\kappa(12)}Y_\alpha + P_{\kappa(13)}Z_\alpha + P_{\kappa(14)}, \\ q_{\alpha\kappa} &= P_{\kappa(21)}X_\alpha + P_{\kappa(22)}Y_\alpha + P_{\kappa(23)}Z_\alpha + P_{\kappa(24)}, \\ r_{\alpha\kappa} &= P_{\kappa(31)}X_\alpha + P_{\kappa(32)}Y_\alpha + P_{\kappa(33)}Z_\alpha + P_{\kappa(34)}, \end{aligned} \quad (10.84)$$

we rewrite Eq. (10.83) in the form

$$E = \sum_{\alpha=1}^N \sum_{\kappa=1}^M \left(\left(\frac{p_{\alpha\kappa}}{r_{\alpha\kappa}} - \frac{x_{\alpha\kappa}}{f_0} \right)^2 + \left(\frac{q_{\alpha\kappa}}{r_{\alpha\kappa}} - \frac{y_{\alpha\kappa}}{f_0} \right)^2 \right). \quad (10.85)$$

Using a single running index $k = 1, 2, \dots$ for all the unknowns, i.e., all the 3D positions $(X_\alpha, Y_\alpha, Z_\alpha)$, $\alpha = 1, \dots, N$, and all the camera matrices \mathbf{P}_κ , $\kappa = 1, \dots, M$, we write all the unknowns as ξ_1, ξ_2, \dots . The first derivative of the reprojection error E with respect to ξ_k is

$$\begin{aligned} \frac{\partial E}{\partial \xi_k} &= \sum_{\alpha=1}^N \sum_{\kappa=1}^M \frac{2}{r_{\alpha\kappa}^2} \left(\left(\frac{p_{\alpha\kappa}}{r_{\alpha\kappa}} - \frac{x_{\alpha\kappa}}{f_0} \right) \left(r_{\alpha\kappa} \frac{\partial p_{\alpha\kappa}}{\partial \xi_k} - p_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_k} \right) \right. \\ &\quad \left. + \left(\frac{q_{\alpha\kappa}}{r_{\alpha\kappa}} - \frac{y_{\alpha\kappa}}{f_0} \right) \left(r_{\alpha\kappa} \frac{\partial q_{\alpha\kappa}}{\partial \xi_k} - q_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_k} \right) \right). \end{aligned} \quad (10.86)$$

Next, we consider second derivatives. Noting that as Eq. (10.85) decreases in the course of iterations, we expect that $p_{\alpha\kappa}/r_{\alpha\kappa} - x_{\alpha\kappa}/f_0 \approx 0$ and $q_{\alpha\kappa}/r_{\alpha\kappa} - y_{\alpha\kappa}/f_0 \approx 0$. So, we adopt the Gauss–Newton approximation of ignoring them. Then, the second derivative of E is written as

$$\begin{aligned} \frac{\partial^2 E}{\partial \xi_k \partial \xi_l} &= 2 \sum_{\alpha=1}^N \sum_{\kappa=1}^M \frac{1}{r_{\alpha\kappa}^4} \left(\left(r_{\alpha\kappa} \frac{\partial p_{\alpha\kappa}}{\partial \xi_k} - p_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_k} \right) \left(r_{\alpha\kappa} \frac{\partial p_{\alpha\kappa}}{\partial \xi_l} - p_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_l} \right) \right. \\ &\quad \left. + \left(r_{\alpha\kappa} \frac{\partial q_{\alpha\kappa}}{\partial \xi_k} - q_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_k} \right) \left(r_{\alpha\kappa} \frac{\partial q_{\alpha\kappa}}{\partial \xi_l} - q_{\alpha\kappa} \frac{\partial r_{\alpha\kappa}}{\partial \xi_l} \right) \right). \end{aligned} \quad (10.87)$$

As a result, for computing the first and second derivatives $\partial E/\partial \xi_k$ and $\partial^2 E/\partial \xi_k \partial \xi_l$ of E , we only need to evaluate the first derivatives $\partial p_{\alpha\kappa}/\partial \xi_k$, $\partial q_{\alpha\kappa}/\partial \xi_k$, and $\partial r_{\alpha\kappa}/\partial \xi_k$ of $p_{\alpha\kappa}$, $q_{\alpha\kappa}$, and $r_{\alpha\kappa}$.

Now, we apply the Lie algebra method to differentiation with respect to the rotation \mathbf{R}_κ in Eq. (10.82)³; to other unknowns (the 3D positions $(X_\alpha, Y_\alpha, Z_\alpha)$, the camera positions \mathbf{t}_κ , and all the parameters contained in the intrinsic parameter matrix \mathbf{K}_κ), we can apply the usual chain rule straightforwardly.

The linear increment $\Delta \mathbf{P}_\kappa$ of Eq. (10.82) caused by a small change $\mathbf{A}(\Delta \boldsymbol{\omega}_\kappa) \mathbf{R}_\kappa$ of \mathbf{R}_κ is written as

$$\begin{aligned} \Delta \mathbf{P}_\kappa &= \mathbf{K}_\kappa (\mathbf{A}(\Delta \boldsymbol{\omega}_\kappa) \mathbf{R}_\kappa)^\top (\mathbf{I} - \mathbf{t}_\kappa) = \mathbf{K}_\kappa \mathbf{R}_\kappa^\top (\mathbf{A}(\Delta \boldsymbol{\omega}_\kappa)^\top - \mathbf{A}(\Delta \boldsymbol{\omega}_\kappa)^\top \mathbf{t}_\kappa) \\ &= \mathbf{K}_\kappa \mathbf{R}_\kappa^\top \begin{pmatrix} 0 & \Delta \omega_{\kappa 3} & -\Delta \omega_{\kappa 2} & \Delta \omega_{\kappa 2} t_{\kappa 3} - \Delta \omega_{\kappa 3} t_{\kappa 2} \\ -\Delta \omega_{\kappa 3} & 0 & \Delta \omega_{\kappa 1} & \Delta \omega_{\kappa 3} t_{\kappa 1} - \Delta \omega_{\kappa 1} t_{\kappa 3} \\ \Delta \omega_{\kappa 2} & -\Delta \omega_{\kappa 1} & 0 & \Delta \omega_{\kappa 1} t_{\kappa 2} - \Delta \omega_{\kappa 2} t_{\kappa 1} \end{pmatrix}, \end{aligned} \quad (10.88)$$

where $\Delta \omega_{\kappa i}$ and $t_{\kappa i}$ are the i th components of $\Delta \boldsymbol{\omega}_\kappa$ and \mathbf{t}_κ , respectively. Rewriting the above equation in the form

$$\Delta \mathbf{P}_\kappa = \frac{\partial \mathbf{P}_\kappa}{\partial \omega_{\kappa 1}} \Delta \omega_{\kappa 1} + \frac{\partial \mathbf{P}_\kappa}{\partial \omega_{\kappa 2}} \Delta \omega_{\kappa 2} + \frac{\partial \mathbf{P}_\kappa}{\partial \omega_{\kappa 3}} \Delta \omega_{\kappa 3}, \quad (10.89)$$

we obtain the gradients $\partial \mathbf{P}_\kappa/\partial \omega_{\kappa 1}$, $\partial \mathbf{P}_\kappa/\partial \omega_{\kappa 2}$, and $\partial \mathbf{P}_\kappa/\partial \omega_{\kappa 3}$ of \mathbf{P}_κ with respect to the small rotation vector $\Delta \boldsymbol{\omega}_\kappa$. Letting the components of the vector $\boldsymbol{\omega}_\kappa$ be included in the set of ξ_i , we obtain the first derivatives $\partial p_{\alpha\kappa}/\partial \xi_k$, $\partial q_{\alpha\kappa}/\partial \xi_k$, and $\partial r_{\alpha\kappa}/\partial \xi_k$ of Eq. (10.84) for the rotation. Note that the value of $\boldsymbol{\omega}_\kappa$ is not defined but its differential is defined. Using Eqs. (10.86) and (10.87), we can compute the first and second derivatives $\partial E/\partial \xi_k$ and $\partial^2 E/\partial \xi_k \partial \xi_l$ of the reprojection error E . The Levenberg–Marquardt bundle adjustment procedure has the following form:

³The quaternion representation of rotations is used in most of the currently available open software.

1. Initialize the 3D positions $(X_\alpha, Y_\alpha, Z_\alpha)$ and the camera matrices \mathbf{P}_κ , and compute the associated reprojection error E . Let $c = 0.0001$.
2. Compute the first and second derivatives $\partial E/\partial \xi_k$ and $\partial^2 E/\partial \xi_k \partial \xi_l$ for all the unknowns.
3. Solve the following linear equation for $\Delta \xi_k, k = 1, 2, \dots$:

$$\begin{pmatrix} \partial^2 E/\partial \xi_1^2 + c & \partial^2 E/\partial \xi_1 \partial \xi_2 & \partial^2 E/\partial \xi_1 \partial \xi_3 & \dots \\ \partial^2 E/\partial \xi_2 \partial \xi_1 & \partial^2 E/\partial \xi_2^2 + c & \partial^2 E/\partial \xi_2 \partial \xi_3 & \dots \\ \partial^2 E/\partial \xi_3 \partial \xi_1 & \partial^2 E/\partial \xi_3 \partial \xi_2 & \partial^2 E/\partial \xi_3^2 + c & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Delta \xi_1 \\ \Delta \xi_2 \\ \Delta \xi_3 \\ \vdots \end{pmatrix} = - \begin{pmatrix} \partial E/\partial \xi_1 \\ \partial E/\partial \xi_2 \\ \partial E/\partial \xi_3 \\ \vdots \end{pmatrix}. \quad (10.90)$$

4. Tentatively update the unknowns ξ_k to $\tilde{\xi}_k = \xi_k + \Delta \xi_k$ *except the rotations* \mathbf{R}_κ , which are updated to $\tilde{\mathbf{R}}_\kappa = e^{A(\Delta \omega_\kappa)} \mathbf{R}_\kappa$.
5. Compute the corresponding reprojection error \tilde{E} . If $\tilde{E} > E$, let $c \leftarrow 10c$ and go back to Step 3.
6. Update the unknowns to $\xi_k \leftarrow \tilde{\xi}_k$. If $|\tilde{E} - E| \leq \delta$, then stop (δ is a small constant). Else, let $E \leftarrow \tilde{E}$ and $c \leftarrow c/10$ and go back to Step 2.

In usual numerical iterations, the variables are successively updated until they no longer change. However, the number of unknowns for bundle adjustment is thousands or even tens of thousands, so an impractically long computation time would be necessary if all variables were required to converge over significant digits. On the other hand, the purpose of bundle adjustment is to find a solution with a small reprojection error. So, it is a practical compromise to stop if the reprojection error almost ceases to decrease, as we describe in the above procedure.

For actual implementation, many issues arise. One of them is the scale and orientation indeterminacy. This is a consequence of the fact that the world coordinate system can be arbitrarily defined and that imaging a small object by a nearby camera will produce the same image as imaging a large object by a faraway camera. To resolve this indeterminacy, we usually define the world coordinate system so that it coincides with the first camera frame and fix the scale so that the distance between the first and second cameras is unity. Normalization like this reduces the number of unknowns of Eq. (10.90). Also, all the points in the scene are not necessarily seen in all the images, so we must adjust the number of equations and unknowns of Eq. (10.90), accordingly.

Another issue is the computation time. Directly solving Eq. (10.90) would require hours or days of computation. One of the well-known techniques for reducing this is to separate the unknowns to the 3D point part and the camera matrix part; we solve for the unknowns of one part in terms of the unknowns of the other part and substitute the result into the remaining linear equations, which results in a smaller-size coefficient matrix known as the *Schur complement* [26]. The memory space is another issue; we need to retain all relevant information in the course of the iterations without writing all intermediate values in memory arrays, which might exhaust the memory resource. See [16] for implementation details and numerical examples using real image data.

10.9 Summary

We have described how we can optimize the pose computation involving rotations using image and sensor data. We have pointed out that *we do not need any parameterization of the rotation* (axis–angle, Euler angles, quaternions, etc.); we only need to parameterize *infinitesimal rotations*, which form a linear space called the *Lie algebra*. We have shown how the rotation matrix \mathbf{R} is successively updated without involving any parameterization in the Levenberg–Marquardt framework. We have demonstrated our Lie algebra method for maximum likelihood rotation estimation, fundamental matrix computation, and bundle adjustment for 3D reconstruction.

The problems we have shown here have been well known and solved by many other methods, often with heuristics and ad-hoc treatment. Software tools for them are available on the Web, and their performance is usually satisfactory. We are not asserting that the use of Lie algebra improves their performance greatly. Our aim here is to emphasize the role Lie algebra plays in vision applications, because it is a fundamental mathematical principle that can be applied to a wide range of nonlinear optimization problems.

Lie algebra has been used for robotics control of continuously changing 3D postures [4, 6]. Recently, some researchers are using the Lie algebra method for “motion averaging”: the 3D posture is computed by different methods and sensors, resulting in different values, and their best average is computed by iterative optimization [5, 7]. A similar approach was used to create a seamless circular panorama by optimizing the camera orientations [22]. In Sect. 10.7, we showed how to optimally compute the fundamental matrix. If the camera internal parameters are all known, the fundamental matrix is called the “essential matrix,” and the Lie algebra method is also used to optimize it [27].

Thus, Lie algebra plays an important role in a wide range of computer vision problems. This chapter is aimed to help deepening its understanding.

References

1. Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5), 698–700.
2. Bartoli, A., & Sturm, P. (2004). Nonlinear estimation of fundamental matrix with minimal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3), 426–432.
3. Básaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., García, X., Tyrsa, V. V., Rivas-Lopez, M., & et al. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
4. Benhimane, S., & Malis, E. (2007). Homography-based 2D Visual tracking and servoing. *International Journal of Robotics Research*, 26(7), 661–676.
5. Chatterjee, A., & Govindu, V. M. (2018). Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 958–972.
6. Drummond, T., & Cipolla, R. (2000). Application of Lie Algebra to visual servoing. *International Journal of Computer Vision*, 37(1), 65–78.

7. Govindu, V. M. (2018). Motion averaging in 3D reconstruction problems. In P. K. Turaga & A. Srivastava (Eds.), *Riemannian computing in computer vision*, (pp. 145–186). Cham: Springer.
8. Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6), 580–593
9. Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.
10. Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4), 629–642.
11. Kanatani, K. (1990). *Group-theoretical methods in image understanding*. Berlin: Springer.
12. Kanatani, K. (1994). Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5), 543–549.
13. Kanatani, K. (1996). *Statistical optimization for geometric computation: Theory and practice*. Amsterdam: Elsevier. Reprinted Dover, New York, 2005.
14. Kanatani, K. (2015). *Understanding geometric algebra: Hamilton, Grassmann, and Clifford for computer vision and graphics*. CRC: Boca Raton
15. Kanatani, K., & Matsunaga, C. (2013). Computing internally constrained motion of 3-D sensor data for motion interpretation. *Pattern Recognition*, 46(6), 1700–1709.
16. Kanatani, K., Sugaya, Y., & Kanazawa, Y. (2016). *Guide to 3D vision computation: Geometric analysis and implementation*. Springer: Cham.
17. Lourakis, M. I. A., & Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1), 21–30.
18. Ohta, N., & Kanatani, K. (1998). Optimal estimation of three-dimensional rotation and reliability evaluation. *IEICE Transactions on Information and Systems*, E81-D(11), 1243–1252.
19. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge: Cambridge University Press.
20. Rodriguez-Quinonez, J. C., Sergiyenko, O., Gonzalez-Navarro, F. F., Basaca-Preciado, L., Tyrsa, V. (2013). Surface recognition improvement in 3D medical laser scanner using Levenberg Marquardt method. *Signal Processing*, 93(2), 378–386.
21. Rodríguez-Quinonez, J. C., Sergiyenko, O., Flores-Feuntes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
22. Sakamoto, M., Sugaya, Y., & Kanatani, K. (2006). Homography optimization for consistent circular panorama generation. In *Proceeding 2006 IEEE Pacific-Rim Symposium Image Video Technology, Hsinchu, Taiwan* (pp. 1195–1205)
23. Snavely, N., Seitz, S., & Szeliski, R. (1995). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(8), 835–846.
24. Snavely, N., Seitz, S., & Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(22), 189–210
25. Sugaya, Y., & Kanatani, K. (2007). High accuracy computation of rank-constrained fundamental matrix. In *Proceeding 18th British Machine Vision Conference, Coventry, U.K.* (vol. 1, pp. 282–291).
26. Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. (2000). Bundle adjustment—A modern synthesis. In B. Triggs, A. Zisserman, & R. Szeliski (Eds.), *Vision algorithms: Theory and practice* (pp. 298–375). Berlin: Springer.
27. Tron, R. & Daniilidis, K. (2017). The space of essential matrices as a Riemannian quotient manifold. *SIAM Journal on Imaging Sciences*, 10(3), 1416–1445.

Chapter 11

Optimal Generation of Closed Trajectories over Large, Arbitrary Surfaces Based on Non-calibrated Vision



Emilio J. Gonzalez-Galvan, Ambrocio Loredo-Flores, Isela Bonilla-Gutierrez, Marco O. Mendoza-Gutierrez, Cesar Chavez-Olivares, Luis A. Raygoza, and Sergio Rolando Cruz-Ramírez

Acronyms

3D	Three dimensional
CAD	Computer-aided design
CSM	Camera-space manipulation
TCP/IP	Transmission Control Protocol/Internet Protocol
TM	Trademark

11.1 Introduction

A large number of relevant robotic manufacturing applications demand a known trajectory to be precisely traced over a large surface whose geometry is not known in advance. This type of demanding tasks includes cutting and welding over the surface

E. J. Gonzalez-Galvan (✉) · A. Loredo-Flores · I. Bonilla-Gutierrez · M. O. Mendoza-Gutierrez
Autonomous University of San Luis Potosi (UASLP), San Luis Potosi, SLP, Mexico
e-mail: egonzale@uaslp.mx; ambrocio.loredo@uaslp.mx; isela.bonilla@uaslp.mx;
marco.mendoza@uaslp.mx

C. Chavez-Olivares
Campus Sur, UAA, Av. Universidad 940. Aguascalientes, AGS, Mexico
e-mail: cesar.chavez@edu.uaa.mx

L. A. Raygoza
Centro de Ciencias Basicas, UAA, Av. Universidad 940. Aguascalientes, AGS, Mexico

S. R. Cruz-Ramírez
SLP Campus, ITESM, Eugenio Garza Sada 300, San Luis Potosi, SLP, Mexico
e-mail: rolando.cruz@tec.mx

of large containers, required for installing attachments like man-holes or flanges. Additionally, they include the cutting of arbitrary shapes over extensive commercial plates and the covering of melted metal over a worn surface [24]. Contributions related to the problem of automating the teach stage, required in a typical industrial-robot application [17], rely on a precise geometrical description of the part to be processed. However, in the cases presented above, such an accurate knowledge of the workpiece geometry is not a viable alternative. As such, the considerable disparity in geometry of a commercially-available plate, produced mainly by bending, the uncertainty in the contour of a worn surface or the permissible range of variation in a dimension, allowed in the fabrication of a large vessel, larger than the one allowed for the placement of a cutting or welding tool, requires a demanding robot-programming procedure, different for each task. In the case of laser and plasma-cutting using robots, the level of precision required for the location of the tool over the surface is about 1° for the orientation and 1 mm for positioning of the cutting-nozzle, relative to the surface (Cisneros, 2003, Engineering manager, Yaskawa-Motoman Mexico, Personal communication). The ideas presented herein can be used to reduce the complexity involved in the robot programming process.

The problem of tracking a given path over large surfaces is found in the case of autonomous mobile robots, which are expected to travel in unknown environments without supervision [8]. Such a capability is enabled by using vision-based techniques like the one referred to as visual simultaneous localization and mapping [2]. Other applications, such as train cab front cleaning [13], require accurate robot positioning over large surfaces of unknown geometry. Although accurate robot positioning in large workspace areas is a challenging maneuver, the opposite in terms of trajectory planning and maneuvering in a reduced workspace represents also a relevant task, as presented in [10].

Related to the ideas presented herein, the accuracy of a robotic vision-based maneuver, when performed over large surfaces, depends on several factors. One of them is the camera resolution per unit physical space. Another aspect is related to the distortion of a previously-planned trajectory when mapped over the arbitrarily curved surface. When the tracking of a closed path is required, such a distortion may restrict the robot from achieving closure. For the maneuvers mentioned above, especially cutting, achieving closure represents an indispensable requirement. Reported works [4] have demonstrated successful implementation of a method based on the use of cameras applied to the solution of the problem of generating and tracking paths over an arbitrarily curved surface. This methodology considered the idea of applying geodesic-mapping with the purpose of transferring a known trajectory stored as a CAD model, over a curved surface. Such a methodology was successfully developed and validated using an industrial robot, which achieved closure of a closed path in the case of a developable surface. Initial results in the case of a closed path being traced over non-developable surfaces were also reported [6]. In this context, several aspects that have a negative effect in achieving closure of a closed path, when traced over this kind of surfaces, were analyzed. Among them, the proposed geodesic mapping is incapable of achieving closure [11, 15] when a mapping is sought between a virtual, flat surface (a CAD model in this case) and a non-developable surface.

Furthermore, the integration process involved in the mapping procedure caused an error accumulation as the path advances. In order to reduce these negative effects, two algorithms are proposed herein in the following sections. In order to face the problem associated with camera resolution, several solutions can be considered. For example, the application of a reduced number of sensors in combination with mirrors is presented in [12]. Other solutions [3] consider multiple cameras or the use of several, calibrated sensors located on pan/tilt units [25], applied to traffic monitoring [26]. In this chapter, multiple, fixed sensors are applied in order to perform path-tracking tasks over extensive surfaces.

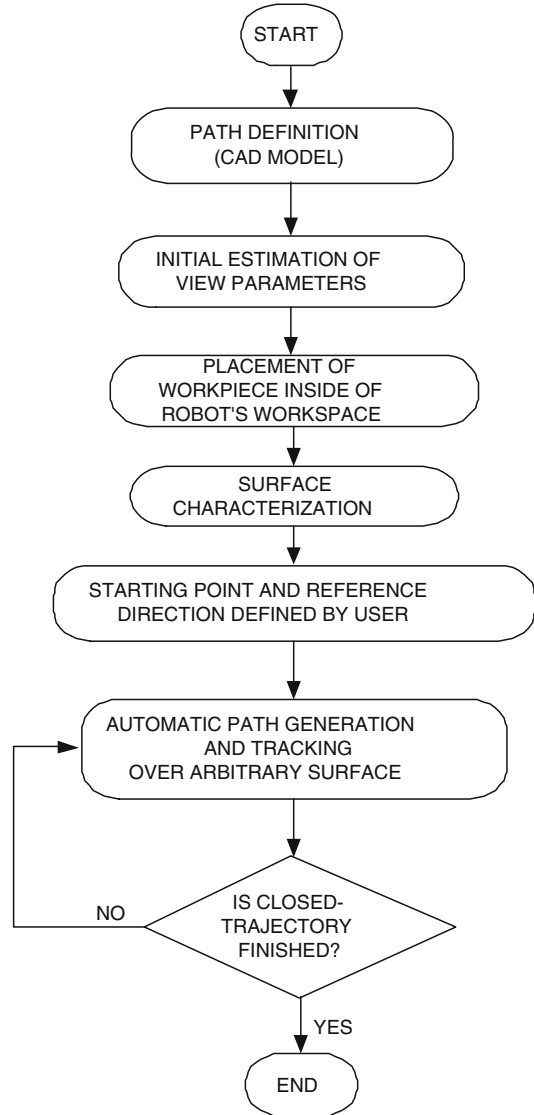
It is worth mentioning that some of the maneuvers cited above tend to produce heat-induced deformations in the workpiece. This effect may affect the manufacturing maneuver and restrict the use of the vision-control strategy while the task is executed, mainly because of the smoke and light produced during, for instance, a welding or cutting process. In this context, the relevance of determining an optimal robot trajectory prior to the actual execution of the industrial maneuver may be underestimated. Nevertheless, there exist in the market devices that take into consideration not only the deformations produced by heat but also the effect of smoke and light emitted during the execution of the task. For example, Fanuc Robotics developed the *Adaptive Welding*[™] tool (<https://www.fanucamerica.com/products/robots/other-robot-options>. Accessed in 2018). Details regarding this particular device are not available, however, it is known that the tool uses a laser-joint scanner in order to adjust welding parameters like travel speed, weaving, voltage, and wire-feed speed, with the purpose of filling the joints to be welded. These commercially-available tools require a robot path defined in advance, which is modified according to the requirements of the welding procedure. Such a path is the one that results from the use of the methodology presented herein. Therefore, the proposed technique is capable of working together with commercial tools designed for correcting the robot-path, according to the changing geometry of the workpiece during task execution. Additionally, the proposed strategy can be included as part of a library for a robot-trajectory generation, as proposed in [16] for a particular brand of robots, different from the robot used as our experimental platform.

11.2 General Aspects Associated with a Path-Generation and Tracking Maneuver

The vision-control methodology proposed herein demands the path-generation and path-tracking processes to be performed simultaneously. This is presented in the flow diagram of Fig. 11.1, which depicts the event in which a trajectory known in advance and stored as a CAD file is followed only once.

A sequence of robot configurations that enables the successful execution of the maneuver is obtained once the depicted algorithm is finished. Prior to the execution of the program, a set of vision parameters must first be defined. These parameters

Fig. 11.1 Flow-diagram of a path generation and tracking task



pertain to a camera model that correlates the arm configuration with the presence, in the images obtained from the camera sensors, of a series of visual marks that are fixed to the tool of the robot. These marks are easily identified by using an image-analysis algorithm. Such a relationship is known as *camera-space kinematics* and is presented in the following section.

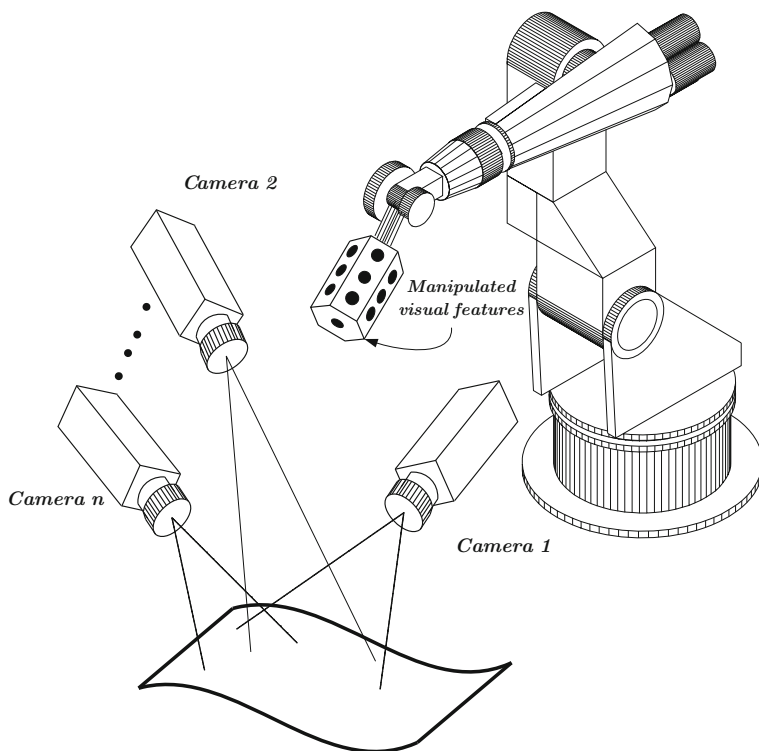


Fig. 11.2 Control cameras used in the maneuver

Next, the workpiece is placed in an arbitrary location and orientation, inside of the workspace of the robot. The cameras used to control the maneuver point to different regions within the workpiece, as depicted in Fig. 11.2.

The following sections will address additional aspects associated with the maneuver for tracing the path, presented in the flow diagram of Fig. 11.1.

11.3 Camera-Space Kinematics

The camera-space manipulation (CSM) method is based on the use of non-calibrated cameras as primary sensors. This method has been applied to robot positioning and path-tracking tasks that require high precision. The objectives of the maneuver are defined within the images obtained from the sensors which remain fixed between the instant in which the target is detected until maneuver termination. The correlation between robot configuration and the location of manipulated visual features within

Fig. 11.3 Industrial robot used to validate the proposed approach



the images obtained from the cameras, like those shown in Fig. 11.3, is known as camera-space kinematics. Such a relationship is achieved by determining seven parameters included in the following model of camera:

$$f_x(x, y, z, x_c; \mathbf{C}) = \mathcal{O}_x(x, y, z; \mathbf{C}) - \mathcal{P}_x(x, y, z, x_c; \mathbf{C}) \quad (11.1)$$

$$f_y(x, y, z, y_c; \mathbf{C}) = \mathcal{O}_y(x, y, z; \mathbf{C}) - \mathcal{P}_y(x, y, z, y_c; \mathbf{C})$$

where (f_x, f_y) represents the perspective projection of a point with coordinates (x, y, z) . Each expression f_x and f_y consists of two components. The first $\mathcal{O}_x(x, y, z; \mathbf{C})$ and $\mathcal{O}_y(x, y, z; \mathbf{C})$ constitutes an orthographic projection, while the second component $\mathcal{P}_x(x, y, z, x_c; \mathbf{C})$ and $\mathcal{P}_y(x, y, z, y_c; \mathbf{C})$ constitutes a correction of the orthographic component, which completes the perspective projection, as follows:

$$\begin{aligned} \mathcal{O}_x(x, y, z; \mathbf{C}) &= g_1(\mathbf{C})x + g_2(\mathbf{C})y + g_3(\mathbf{C})z + g_4(\mathbf{C}) \\ \mathcal{P}_x(x, y, z, x_c; \mathbf{C}) &= g_9(\mathbf{C})xx_c + g_{10}(\mathbf{C})yx_c + g_{11}(\mathbf{C})zx_c \\ \mathcal{O}_y(x, y, z; \mathbf{C}) &= g_5(\mathbf{C})x + g_6(\mathbf{C})y + g_7(\mathbf{C})z + g_8(\mathbf{C}) \\ \mathcal{P}_y(x, y, z, y_c; \mathbf{C}) &= g_9(\mathbf{C})xy_c + g_{10}(\mathbf{C})yy_c + g_{11}(\mathbf{C})zy_c \end{aligned} \quad (11.2)$$

where g_1, \dots, g_{11} depend on \mathbf{C} which is a vector containing seven independent parameters, as follows:

$$\begin{aligned}
 g_1(\mathbf{C}) &= C_1^2 + C_2^2 - C_3^2 - C_4^2 \\
 g_2(\mathbf{C}) &= 2(C_2C_3 + C_1C_4) \\
 g_3(\mathbf{C}) &= 2(C_2C_4 - C_1C_3) \\
 g_4(\mathbf{C}) &= C_5 \\
 g_5(\mathbf{C}) &= 2(C_2C_3 - C_1C_4) \\
 g_6(\mathbf{C}) &= C_1^2 - C_2^2 + C_3^2 - C_4^2 \\
 g_7(\mathbf{C}) &= 2(C_1C_2 + C_3C_4) \\
 g_8(\mathbf{C}) &= C_6 \\
 g_9(\mathbf{C}) &= 2C_7(C_2C_4 + C_1C_3) \\
 g_{10}(\mathbf{C}) &= 2C_7(C_3C_4 - C_1C_2) \\
 g_{11}(\mathbf{C}) &= C_7(C_1^2 - C_2^2 - C_3^2 + C_4^2)
 \end{aligned}
 \tag{11.3}$$

The parameters in \mathbf{C} are defined as follows:

$$\begin{aligned}
 C_i^2 &= \frac{f}{Z_0} e_i^2 ; \quad i = 1, \dots, 4 \\
 C_5 &= f \frac{X_0}{Z_0} ; \quad C_6 = f \frac{Y_0}{Z_0} ; \quad C_7 = -\frac{1}{f}
 \end{aligned}
 \tag{11.4}$$

where, as shown in Fig. 11.4, xyz is a coordinate frame associated with the base of the robot whose origin is (X_0, Y_0, Z_0) with respect to the coordinate system attached to the camera; $e_1 \dots, e_4$ are the Euler parameters that define the relative orientation

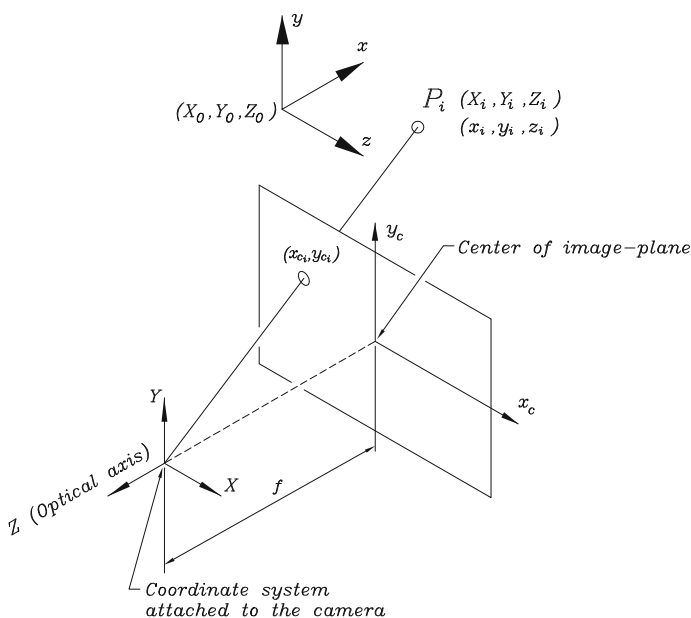


Fig. 11.4 Reference coordinate systems associated with the camera model

between these two coordinate systems. Finally, f represents the focal distance of the camera. The coordinate frames are related as follows:

$$[X \ Y \ Z \ 1]^T = \mathbf{T}(e_1, e_2, e_3, e_4, X_0, Y_0, Z_0)[x \ y \ z \ 1]^T \quad (11.5)$$

where

$$\mathbf{T}(e_1, e_2, e_3, e_4, X_0, Y_0, Z_0) = \begin{bmatrix} t_{11} & t_{12} & t_{13} & X_0 \\ t_{21} & t_{22} & t_{23} & Y_0 \\ t_{31} & t_{32} & t_{33} & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.6)$$

and

$$\begin{aligned} t_{11} &= e_1^2 + e_2^2 - e_3^2 - e_4^2 & t_{23} &= 2(e_1e_2 + e_3e_4) \\ t_{12} &= 2(e_2e_3 + e_1e_4) & t_{31} &= 2(e_2e_4 + e_1e_3) \\ t_{13} &= 2(e_2e_4 - e_1e_3) & t_{32} &= 2(e_3e_4 - e_1e_2) \\ t_{21} &= 2(e_2e_3 - e_1e_4) & t_{33} &= e_1^2 - e_2^2 - e_3^2 + e_4^2 \\ t_{22} &= e_1^2 - e_2^2 + e_3^2 - e_4^2 \end{aligned} \quad (11.7)$$

The camera parameters are calculated in such a way that they minimize the following scalar function J :

$$J(\mathbf{C}) = \sum_{i=1}^m \left\{ [x_{c_i} - f_x(x_i, y_i, z_i, x_{c_i}; \mathbf{C})]^2 + [y_{c_i} - f_y(x_i, y_i, z_i, y_{c_i}; \mathbf{C})]^2 \right\} W_i \quad (11.8)$$

for m camera-space samples (x_{c_i}, y_{c_i}) whose three-dimensional location is (x_i, y_i, z_i) with respect to a coordinate system attached to the base of the robot. These coordinates are obtained by considering the nominal kinematic model of the robot; $f_x(\dots)$ and $f_y(\dots)$ are included in Eq. (11.1), while W_i is a weight given to each sample. It is worth mentioning that the parameters within the camera model are continuously updated, by considering samples obtained during the execution of the task. The weight assigned to each sample is increased as the robot approaches maneuver culmination. This measure is applied in order to ensure that the camera model is locally valid in the region where the task is ended. The configuration of the robot is estimated by using the most updated parameters. The procedure for estimating the view parameters is presented in [19].

11.4 Characterization of Surface

The characterization of the surface requires, among other aspects, the employment of structured lighting as a matrix array of laser beams. The use of laser has, among others, the advantage that the detection of the center of laser spots in the images obtained from the sensors can be made through simple image-analysis algorithms like those presented in [14, 23]. Figure 11.5 presents a picture of the structured lighting used for surface characterization. The image-plane information obtained from the projection of structured lighting is restricted, as presented in [4], based on ideas presented in [27]. Figure 11.6 depicts the detected laser spots that are projected within a region defined by the user, which corresponds to the surface of interest.

As mentioned before, a perspective camera model is used for the experiments reported herein. The estimation of the 3D location of the laser spots projected over the surface is performed by considering this model and the parameters associated with each control camera. Since the geometry of the surface is not previously known, the visual features projected over the workpiece are used for determining such a geometry. Applications for accurate three-dimensional distance measurement [21] have been developed and successfully applied to surface measurement [22]. In the case of the surface characterization presented here, an approximate location is obtained and accuracy is only required relative to the spots near the location where the task is being executed. This is because the parameters associated with the control cameras, used for the estimation of the 3D coordinates, are continuously updated while the task is performed and have a local validity, near the location where the maneuver is performed.

Laser spots have been used as visual features [3] because they have the advantage of not being permanent while enabling detection by using relatively simple image-analysis processes. When a laser spot is seen by at least two cameras, the model in Eq. (11.1) can be used to produce a linear estimation of the nominal 3D coordinates of

Fig. 11.5 Projection of a matrix array of laser spots



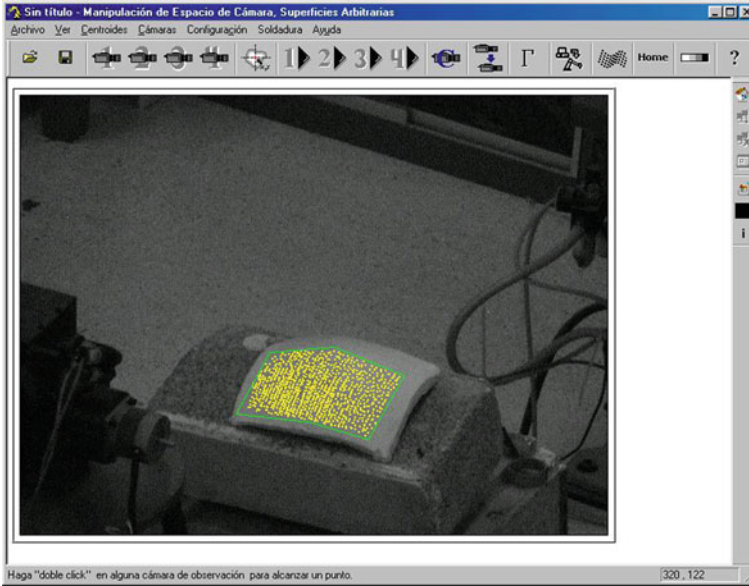


Fig. 11.6 Polygon defined by the user used to delimit the surface of interest

the centroid of the laser spot, relative to a coordinate frame attached to the robot. The approach used for matching the location of a given laser point among the different images obtained from the cameras is detailed in [20]. The linear estimation of the 3D coordinates consists of minimizing the following scalar function:

$$\psi = \sum_{i=1}^{n_d} \left[x_c^{(i)} - f_x(x, y, z, x_c^{(i)}; \mathbf{C}^{(i)}) \right]^2 + \left[y_c^{(i)} - f_y(x, y, z, y_c^{(i)}; \mathbf{C}^{(i)}) \right]^2 \quad (11.9)$$

which considers n_d ($n_d \geq 2$) control cameras that detect a single laser spot, whose 3D coordinates are (x, y, z) . The definition of f_x and f_y is given in Eq. (11.1). The corresponding two-dimensional location of the spot is $(x_c^{(i)}, y_c^{(i)})$ for the i th camera, while the associated parameters are $\mathbf{C}^{(i)}$. This procedure is equivalent to a triangulation method when the number of control cameras is 2, however it has the advantage that the three-dimensional coordinates of the laser spot can be easily determined when more than two cameras are involved in the positioning task. The necessary condition for the minimization of ψ produces the following solution for (x, y, z) :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1}B \quad (11.10)$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} ; B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (11.11)$$

Each term in these two matrices is given by:

$$\begin{aligned} a_{11} &= \sum_{i=1}^{n_d} \left(g_1^{(i)} + g_9^{(i)} x_c^{(i)} \right)^2 + \left(g_5^{(i)} + g_9^{(i)} y_c^{(i)} \right)^2 \\ a_{12} &= \sum_{i=1}^{n_d} \left(g_1^{(i)} + g_9^{(i)} x_c^{(i)} \right) \left(g_2^{(i)} + g_{10}^{(i)} x_c^{(i)} \right) \\ &\quad + \left(g_5^{(i)} + g_9^{(i)} y_c^{(i)} \right) \left(g_6^{(i)} + g_{10}^{(i)} y_c^{(i)} \right) \\ a_{13} &= \sum_{i=1}^{n_d} \left(g_1^{(i)} + g_9^{(i)} x_c^{(i)} \right) \left(g_3^{(i)} + g_{11}^{(i)} x_c^{(i)} \right) \\ &\quad + \left(g_5^{(i)} + g_9^{(i)} y_c^{(i)} \right) \left(g_7^{(i)} + g_{11}^{(i)} y_c^{(i)} \right) \\ a_{22} &= \sum_{i=1}^{n_d} \left(g_2^{(i)} + g_{10}^{(i)} x_c^{(i)} \right)^2 + \left(g_6^{(i)} + g_{10}^{(i)} y_c^{(i)} \right)^2 \\ a_{23} &= \sum_{i=1}^{n_d} \left(g_2^{(i)} + g_{10}^{(i)} x_c^{(i)} \right) \left(g_3^{(i)} + g_{11}^{(i)} x_c^{(i)} \right) \\ &\quad + \left(g_6^{(i)} + g_{10}^{(i)} y_c^{(i)} \right) \left(g_7^{(i)} + g_{11}^{(i)} y_c^{(i)} \right) \\ a_{33} &= \sum_{i=1}^{n_d} \left(g_3^{(i)} + g_{11}^{(i)} x_c^{(i)} \right)^2 + \left(g_7^{(i)} + g_{11}^{(i)} y_c^{(i)} \right)^2 \end{aligned} \quad (11.12)$$

and

$$\begin{aligned} b_1 &= \sum_{i=1}^{n_d} \left(x_c^{(i)} - g_4^{(i)} \right) \left(g_1^{(i)} + g_9^{(i)} x_c^{(i)} \right) \\ &\quad + \left(y_c^{(i)} - g_8^{(i)} \right) \left(g_5^{(i)} + g_9^{(i)} y_c^{(i)} \right) \\ b_2 &= \sum_{i=1}^{n_d} \left(x_c^{(i)} - g_4^{(i)} \right) \left(g_2^{(i)} + g_{10}^{(i)} x_c^{(i)} \right) \\ &\quad + \left(y_c^{(i)} - g_8^{(i)} \right) \left(g_6^{(i)} + g_{10}^{(i)} y_c^{(i)} \right) \\ b_3 &= \sum_{i=1}^{n_d} \left(x_c^{(i)} - g_4^{(i)} \right) \left(g_3^{(i)} + g_{11}^{(i)} x_c^{(i)} \right) \\ &\quad + \left(y_c^{(i)} - g_8^{(i)} \right) \left(g_7^{(i)} + g_{11}^{(i)} y_c^{(i)} \right) \end{aligned} \quad (11.13)$$

in these expressions, the parameters corresponding to the i th camera are included in $g_1^{(i)}, \dots, g_{11}^{(i)}$, as presented in Eq. (11.3). Such an estimation procedure is performed for each of the laser spots projected over the surface. As described in [5], the accuracy in the positioning of the robot depends on a detailed representation of the geometry of the surface, which in turn is enabled by the projection of a large amount of these visual features over the surface.

11.5 Path Tracking

An effective means used to maintain a given ratio between camera-resolution and observed workspace size is the use of multiple cameras. In the proposed methodology, the cameras remain static during the execution of the maneuver, being the only restriction that all different regions within the workspace of the robot have to be observable by at least two cameras. The following scalar function is defined in order to take into consideration the presence of maneuver objectives in several camera-images used to encompass a large region,

$$J(\Theta) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_t(i)} \left\{ \left[x_{t_i}^{(j)} - f_x \left(t_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), x_{t_i}^{(j)}; \mathbf{C}^{(i)} \right) \right]^2 + \left[y_{t_i}^{(j)} - f_y \left(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), y_{t_i}^{(j)}; \mathbf{C}^{(i)} \right) \right]^2 \right\} \delta_i^j \quad (11.14)$$

this function is minimized for the joint coordinates of the robot included in Θ . In the previous function, δ_i^j can be defined as follows:

$$\delta_i^j = \begin{cases} 1 & \text{if the } j\text{th target is to be achieved in the } i\text{th camera} \\ 0 & \text{otherwise} \end{cases} \quad (11.15)$$

As detailed in [4], the procedure for defining maneuver objectives in each of the control cameras is facilitated by the use of laser points projected over the surface. In this case, a path defined in advance as a database is projected over an arbitrary surface. The information in this database is schematically shown in Fig. 11.7. In this figure, the path to be projected is separated into a series of straight-line segments with associated coordinates (r_i, θ_i, z_i) , relative to the preceding segment. Also, associated with each portion of the path is the tool speed and a signal used for establishing if a given action, like cutting, welding, etc., is executed or not in the segment.

Once the surface-characterization process described in the preceding section is completed, a starting point and a direction of reference over the surface is selected by the user in one image obtained from a camera. This selection is made by using a graphical user interface like the one shown in Fig. 11.8. Such a starting point and direction are compatible with those stored in the database, as depicted in Fig. 11.7.

The projection of the path over the arbitrary surface is based on a geodesic projection. As such, it can be considered optimal as it satisfies the optimality conditions and restrictions that a line has to satisfy in order to be considered a geodesic line. In this case, each straight-line segment like the ones shown in Fig. 11.7 is projected as a geodesic line along the arbitrary surface. As discussed in [1], if the equation of the surface is presented in the form $z = f(x, y)$, the differential equation that represents a geodesic line is

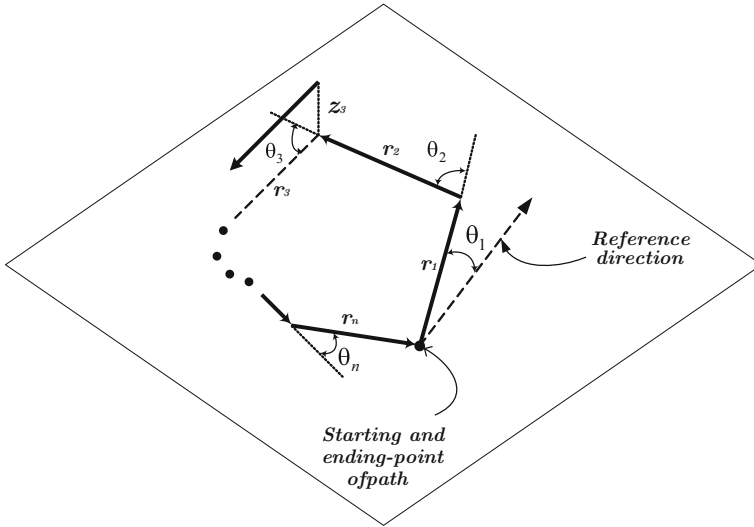


Fig. 11.7 Structure of path stored as a database (CAD model)

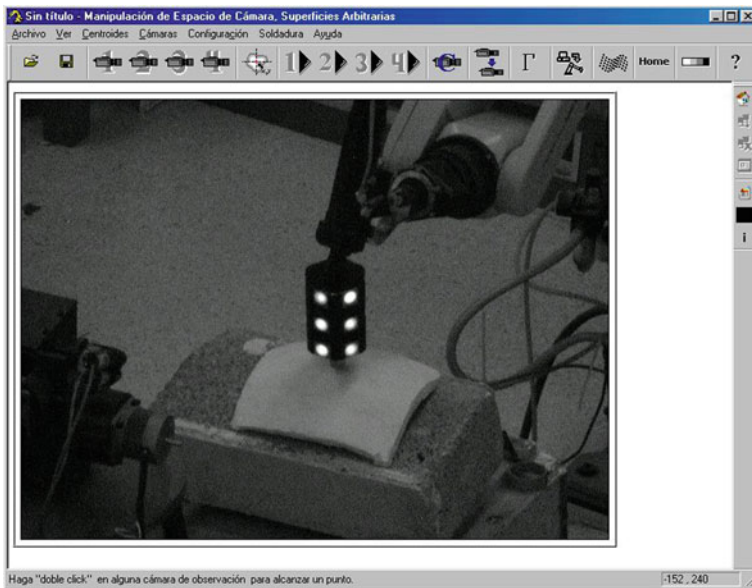


Fig. 11.8 Graphical user interface

$$(1 + s^2 + t^2) \frac{d^2y}{dx^2} = sw \left(\frac{dy}{dx} \right)^3 + (2sv - tw) \left(\frac{dy}{dx} \right)^2 + (su - 2tv) \frac{dy}{dx} - tu \quad (11.16)$$

where

$$s = \frac{\partial z}{\partial x}; t = \frac{\partial z}{\partial y}; u = \frac{\partial^2 z}{\partial x^2}; v = \frac{\partial^2 z}{\partial x \partial y}; w = \frac{\partial^2 z}{\partial y^2} \quad (11.17)$$

Function $z = f(x, y)$ is locally defined as a second-order polynomial as

$$z = p_0 + p_1x + p_2y + p_3x^2 + p_4xy + p_5y^2 \quad (11.18)$$

where parameters p_0 to p_5 are estimated in such a way that they best fit the 3D coordinates of the laser points located within a restricted area, like the one presented in Fig. 11.6. Once this fit is achieved, the second-order differential equation (11.16) is solved using the fourth-order Runge-Kutta integration method [18]. Considering Fig. 11.7, the integration is performed along the direction of the i th segment, until length r_i is achieved. Once the integration is finished along this segment, the 3D location of the ending point becomes the initial point of the following segment. The next integration is performed along a direction rotated an angle θ_i with respect to the ending direction of the previous segment. The process is performed for all the straight-line segments included in the database. Reference [6] presents the procedure for evaluating the robot configuration that permits accurate positioning of the tool held by the robot along the geodesic lines defined above. This procedure requires the selection of coordinate systems attached both to the work surface and to the robot-held tool, as depicted in Fig. 11.9.

A fundamental requirement for the mapping of the previously-defined closed trajectory, stored in a database, over the arbitrary surface, is that closure must be accomplished. This condition is particularly important when applied to an industrial maneuver like metal-cutting. The solution presented herein for achieving such a closure restriction resides in the mapping of two different paths over the surface. One in the direction indicated in the CAD model, as depicted in Fig. 11.7 and the second one in the opposite direction. This is presented schematically in Fig. 11.10. The second path in the reverse direction is defined by using the following recursive formulas:

$$\begin{aligned} r'_i &= r_{n-i+1} & ; i &= 1 \dots n \\ z'_i &= z_{n-i+1} & ; i &= 1 \dots n \\ \theta'_1 &= \sum_{j=1}^n \theta_j - 180^\circ \\ \theta'_i &= 360^\circ - \theta_{n-i+2} & ; i &= 2 \dots n \end{aligned} \quad (11.19)$$

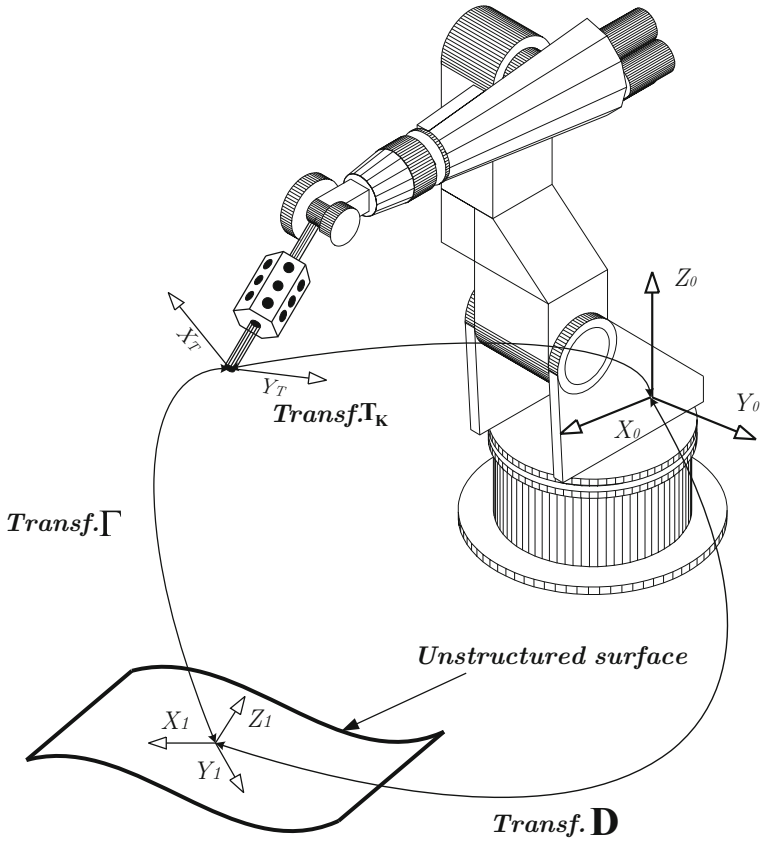


Fig. 11.9 Coordinate systems associated with the path-following task

each element in these formulas corresponds to the i th segment, as shown in Fig. 11.7 while the $()'$ values correspond to the second trajectory in the opposite direction. Since the surface is non-developable, both trajectories do not overlap. An interpolation of the vertexes associated with the same point of each trajectory is used to generate a third path. For example, Fig. 11.10 shows the interpolated points between vertexes A and A' , B and B' , etc. If (X_i, Y_i, Z_i) are the three-dimensional coordinates of a given vertex in the first trajectory while (X'_i, Y'_i, Z'_i) the coordinates of the corresponding vertex in the second trajectory, a linear interpolation is proposed as follows:

$$\begin{aligned}
 X_{p_i} &= X_i + K_i(X'_i - X_i) \\
 Y_{p_i} &= Y_i + K_i(Y'_i - Y_i) \\
 Z_{p_i} &= Z_i + K_i(Z'_i - Z_i)
 \end{aligned}
 \tag{11.20}$$

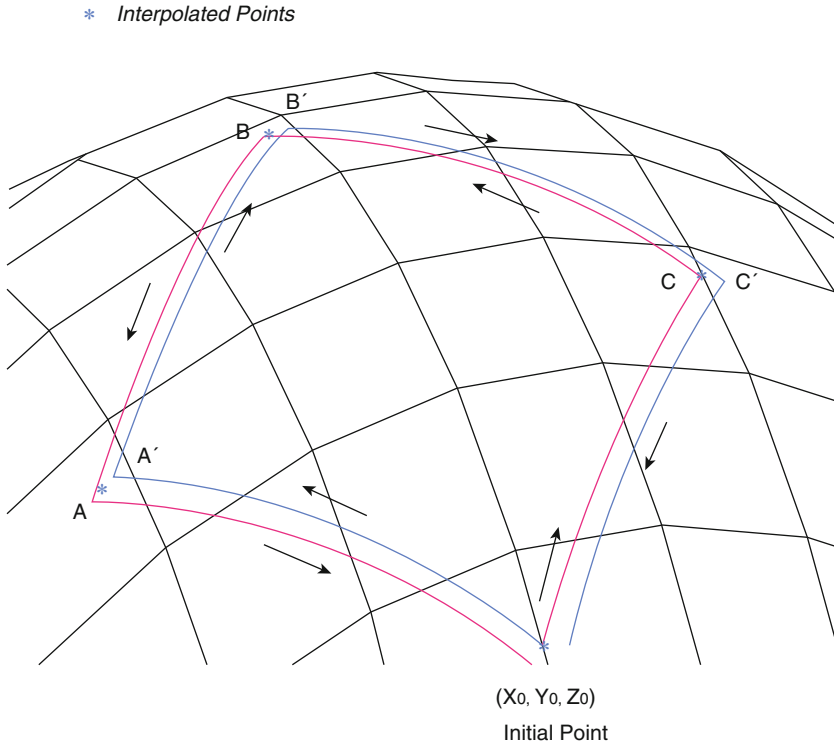


Fig. 11.10 Interpolation between two paths in opposite directions

The three-dimensional coordinates of the vertexes are obtained by minimizing (11.9), which will produce a position with respect to a coordinate frame fixed to the base of the robot. In the previous formula ($X_{p_i}, Y_{p_i}, Z_{p_i}$) represents the coordinates of the interpolated vertex. The interpolation factor K_i is related with the i th segment and takes into consideration the error that accumulates with the integration of the geodesic lines. In this chapter a simple approach is defined and consists of calculating the interpolation factor as

$$K_i = \frac{i - 1}{n} \quad (11.21)$$

where n is the number of segments of the path. In essence, this factor takes into consideration that the interpolated vertex is closer to the corresponding location of the trajectory with the smaller accumulation of integration error. It is considered that such error growth is proportional to the i th segment along the trajectory.

Once the interpolated vertexes are calculated, the final path over the surface is evaluated by solving the geodesic projection equation (11.16) as a boundary-value problem. The integration begins at the starting point (X_0, Y_0, Z_0), shown in

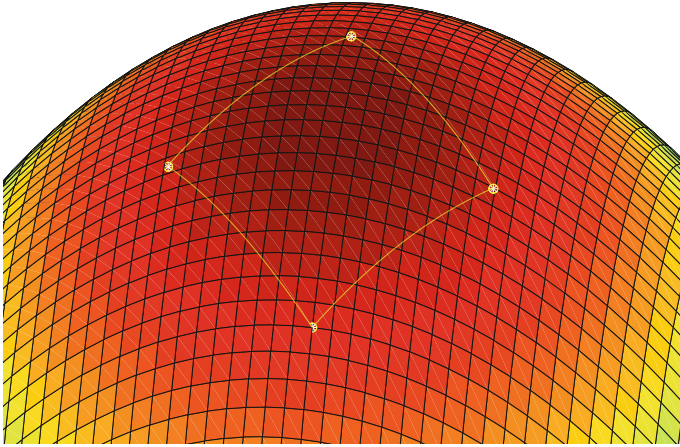


Fig. 11.11 Simulated path through interpolated points

Fig. 11.10. The shooting method [18] is used with the purpose of reaching the first interpolated point (X_{p1}, Y_{p1}, Z_{p1}) as close as possible. The integration continues with the following segment of the path by considering as a starting location of the integration procedure, the ending point of the previous segment. Finally, the last segment is integrated considering the end point as the initial location (X_0, Y_0, Z_0) . This procedure guarantees closure of the path while maintaining a reduced amount of distortion in the resulting closed path. Using this procedure, Fig. 11.11 shows a simulation of how a simple path is produced.

In addition to the geodesic projection, a mapping procedure that considers a virtual projection is used with the purpose of mapping a previously-defined path onto the arbitrary surface. Such a procedure is schematically shown in Fig. 11.12 and is based on the determination of a plane that fits the location of the laser points placed over the surface, using a least-squares estimation procedure. The 3D coordinates of the laser points projected over the surface are obtained as presented in the section where the surface-characterization process is explained. Each vertex of the path stored in the database is projected and the mapped path is obtained as the intersection of the arbitrary surface, whose equation is obtained locally by using Eq. (11.18) and an imaginary line at each vertex whose orientation is perpendicular to the estimated plane.

11.6 Experimental Validation

With the purpose of validating the methodology proposed in this chapter, an industrial Fanuc M16iB 20T robot was used. This device has a tool with known geometry, depicted in Fig. 11.3, with manipulated features that are used to facilitate the image-

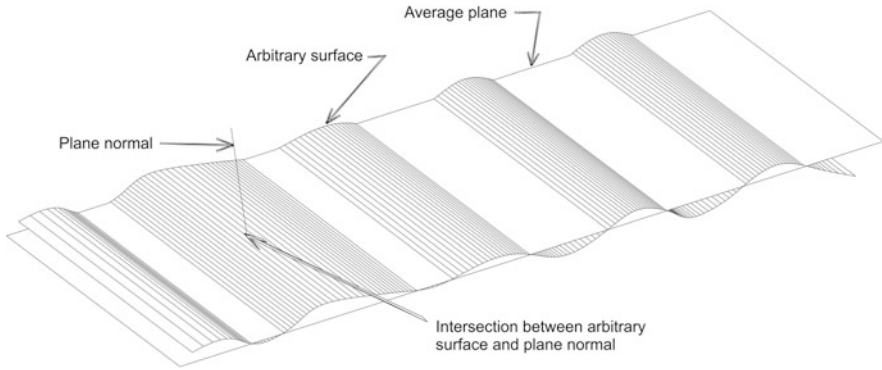


Fig. 11.12 A graphical representation of the virtual-projection mapping

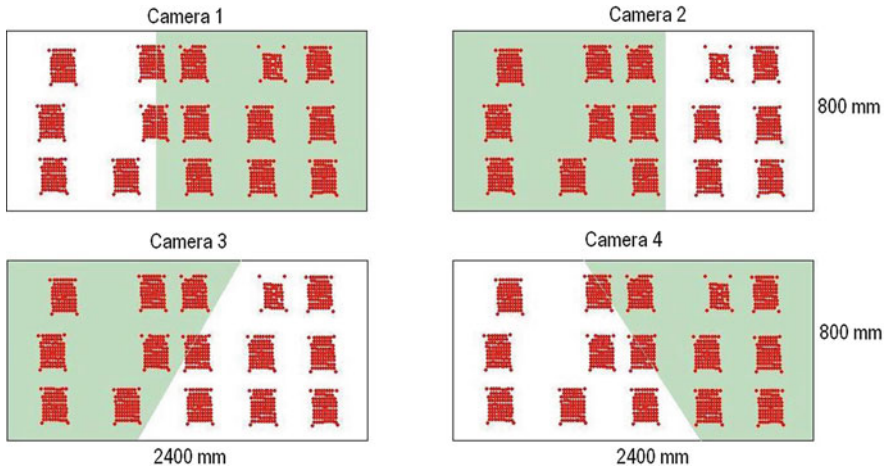


Fig. 11.13 Regions encompassed by each control camera

analysis process. The manipulator has six joints; the first one is prismatic, capable of a 3.6 m displacement. The rest of the joints permits that the robot reaches 1.5 m in a plane perpendicular to the first axis. With this configuration, the large workspace of the robot is roughly a cylinder with a diameter of 3 m and a length of 3.6 m.

The communication that links the computer that includes the control algorithms with the robot is based on a TCP/IP protocol. For the experiments presented herein, such a communication link is needed to obtain samples of the robot pose while approaching its destination. It is also needed for commanding the robot to the estimated configuration obtained from the minimization of the scalar function J included in Eq. (11.14).

Four uEye UI-1540-C control cameras were used and located at a distance of about 5 meters away from the testing surface. Also, a Lasiris laser diode of 5 mW and 635 nm is used with the purpose of projecting marks over the surface which

in turn, as explained in [4], are used for characterizing the working surface. This laser works together with a diffraction head that splits the laser beam into a matrix of 7×7 laser beams. This approach for structured lighting provides a measure for placing a large number of laser spots over the surface. However, there exist in the literature other approaches [7, 9] that take advantage of continuous laser scanning. Such technology has the advantage of improving the resolution associated with the characterization of the surface, however, it is left as a future consideration for the case of the experiments presented herein.

With the purpose of testing the use of a number of fixed camera sensors to encompass a large region of the workspace of the robot, a total of 150 positioning tasks were executed by using the four cameras cited above. As mentioned, the sensors are located in such a way that at least two cameras have visual contact with a particular region of the surface. This surface has a rectangular shape with a length of 2400 mm and a width of 800 mm, and is made out of clay. The approximate region covered by the cameras is depicted in Fig. 11.13 as a shaded area. In this figure, the 150 laser points used for testing the positioning of the robot are seen as dots. In this case, the objective of the task consisted of reaching the centroid of each projected laser spot. The results of the maneuver are shown in Fig. 11.14, which presents the Euclidean distance between the tip of a needle located at the end of the robot-held tool and the centroid of the laser, for each positioning task. In the experiments, we obtained a mean error of 0.62 mm and a standard deviation of 0.34 mm.

In order to validate the capability of the proposed methodology to trace a given path over a large, arbitrary surface, a rectangle of 2000 mm long and 400 mm wide was used. Such a path is comprised of 240, 20 mm long straight-line segments included in a database. As a measure of the error present in the maneuver, the

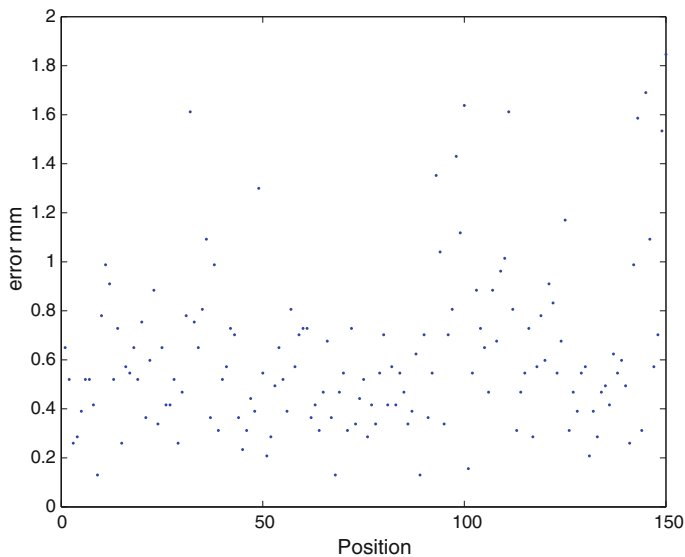


Fig. 11.14 Positioning errors measured during positioning maneuvers

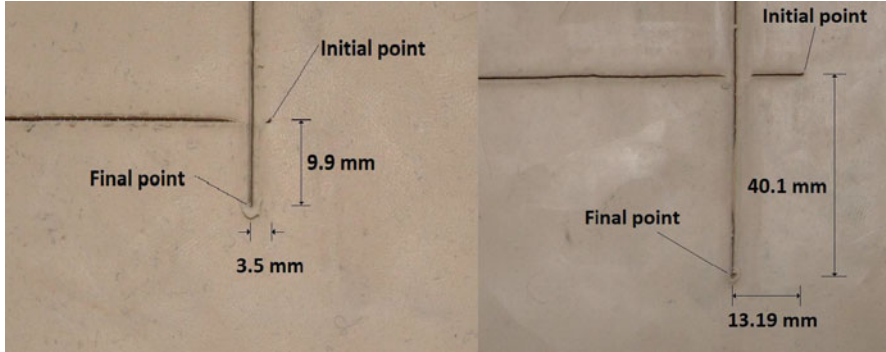


Fig. 11.15 Testing path traced over a soft surface. Left, flat surface, right, curved surface

4800 mm perimeter of the path was used. The path was traced first on top of the surface made of clay that was previously used for the positioning tasks. Then, the geometry of the surface was modified in such a way that it is no longer flat and the difference between the top and the bottom of the surface is approximately 60 mm. In both cases, the corner of the path where it begins and ends is shown in Fig. 11.15 and the path was traced using only a simple geodesic mapping. In the left, the figure presents the trace over the flat surface and in the right, over the curved surface. From the figure is evident that closure is not achieved and also that the error in the closure is aggravated in the case of the non-flat surface. This result is consistent with the fact that in the case of non-developable surfaces, geodesic mapping does not enable closure. Also, it illustrates the effect of integration errors increasing during the tracing of paths over large surfaces. In contrast, as seen in Fig. 11.16, closure is achieved when a combination of geodesic mapping and a weighted interpolation between two opposite paths is used. The closure was also achieved in the case of virtual projection. This result was validated with a total of 20 paths that were traced over the deformed surface, achieving a mean perimeter of 4814.7 mm with a standard deviation of 4.0 mm, when the non-closure simple geodesic mapping is used. In contrast, the closure was achieved in every attempt, with a mean perimeter of 4808.9 mm and standard deviation of 5.4 mm, when using the interpolated-path approach. In the case of virtual projection, a mean perimeter of 4811.2 mm with standard deviation of 3.8 mm was obtained. As can be seen from these tests, more accurate results were obtained from the combination of interpolated-path with the geodesic-mapping approach. Nevertheless, although slightly less accurate, the virtual-projection approach was easier to implement. It is important to mention that, in the reported experiments, the minimum radius of curvature of the deformed surface was approximately 30 cm. It was also observed that the regions behind the bent surface were not visible to the cameras when larger curvatures were considered, which is an aspect that may restrict the application of the proposed vision method.



Fig. 11.16 Comparison of closed paths traced by simple geodesic mapping (1), interpolated geodesic mapping (2), and virtual projection (3)

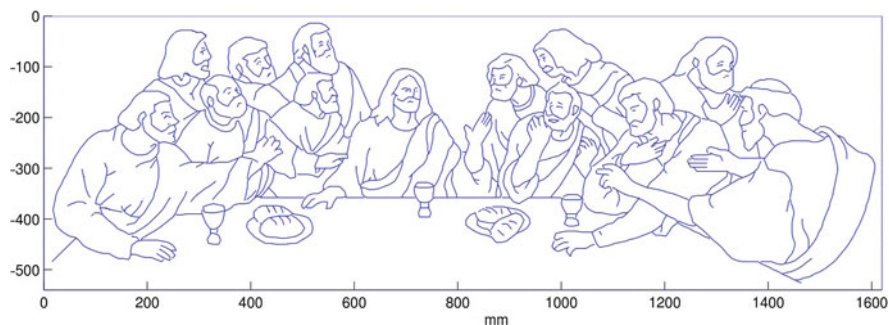


Fig. 11.17 A CAD representation of a Last Supper painting

As a final test, a challenging maneuver consisting of the tracking of a contoured version of the *Last Supper* painting (<http://www.coloringpages101.com/religions-coloring-pages/4450-jesus-is-talking-at-last-supper-coloring-page>. Accessed in 2018), over the large non-flat surface cited above, was performed. A video presenting the execution of this maneuver can be found in (<https://youtu.be/AxvLrqk12Hg>. Published on November 2018). The painting divided into 10 mm straight-line segments is shown in Fig. 11.17. The 2400 mm long and 800 mm wide surface made of clay, with an elevation of about 60 mm, is shown in Fig. 11.18; such a surface representation was obtained after completing the characterization process. For this maneuver, the virtual projection approach was used in order to project the drawing over the surface and the results are depicted in Figs. 11.19 and 11.20. It took approximately five hours, from the initial calculation of camera parameters, surface characterization, etc., until the *Last Supper* drawing is fully transferred over the large, curved surface. Such an extent of time, which may seem excessive,

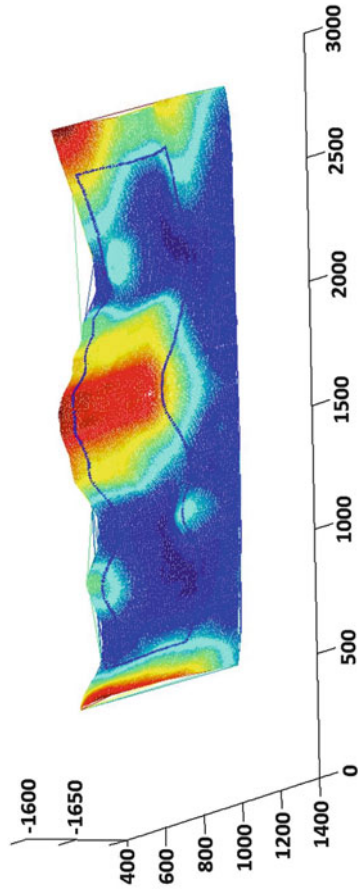


Fig. 11.18 An approximate representation of the curved surface



Fig. 11.19 A picture of the Last Supper painting



Fig. 11.20 Another picture of the Last Supper painting, obtained from a different perspective

can be contrasted with the time that programming such a maneuver may take which, according to experts in the programming of industrial robots, is in the order of days, without the advantages of optimal tracking with reduced distortion, as enabled by the method proposed herein.

11.7 Conclusions

The experiments presented in this chapter show that the proposed methodology is a feasible alternative to project an arbitrary, closed path over a large, curved surface of unknown geometry. As presented, the use of fixed cameras pointing at different regions of the large work-envelope of the robot also enables accurate positioning and path-tracking maneuvers that involve the closure of a closed path.

From a broader perspective, the method proposed herein can be considered as a viable alternative for the self-programming of an industrial manipulator. In this case, the methodology presented here facilitates the programming of the robot, and the maneuver can be repeated once or multiple times. This fact contrasts with the applications given to most industrial robots where a single task is repeated on numerous occasions even during the whole useful operating life of the robot.

Finally, once an initial set of robot configurations that enable a closed path is defined, it is expected that an industrial maneuver like the welding or cutting of metal plates would follow. Such maneuvers can be executed by defining in advance adequate process parameters like traveling speed, voltage, etc. This is especially important in cases where a user, without special qualifications in robotics or in the particular industrial task, will be able to successfully achieve the maneuver.

References

1. Bronshtein, I. N., Semendyayev, K. A., Musiol, G., & Muehlig, H. (2007). *Handbook of mathematics* (5th ed.). Berlin: Springer.
2. Deng, X., Zhang, Z., Sintov, A., Huang, J., & Bretl, T. (2018). Feature-constrained active visual SLAM for mobile robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018 (pp. 7233–7238).
3. González-Gálvan, E. J., Cruz-Ramírez, S. R., Seelinger, M. J., & Cervantes-Sanchez, J. J. (2003). An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision. *Robotics and Computer-Integrated Manufacturing*, 19(5), 387–400.
4. González-Gálvan, E. J., Loredó-Flores, A., Pazos-Flores, F., & Cervantes-Sánchez, J. (2005). An optimal path-tracking algorithm for unstructured environment based on uncalibrated vision. In *IEEE International Conference on Robotics and Automation*, April 2005 (pp. 2547–2552).
5. González-Gálvan, E. J., Loredó-Flores, A., Cervantes-Sánchez, J. J., Aguilera-Cortés, L. A., & Skaar, S. B. (2008). An optimal path-generation algorithm for surface manufacturing of arbitrarily curved surfaces using uncalibrated vision. *Robotics and Computer-Integrated Manufacturing*, 24(1), 77–91.
6. González-Gálvan, E. J., Chavez, C. A., Bonilla, I., Mendoza, M., Raygoza, L. A., Loredó Flores, A., et al. (2011). Precise industrial robot positioning and path-tracking over large surfaces using non-calibrated vision. In *IEEE International Conference on Robotics and Automation*, May 2011 (pp. 5160–5166).
7. Ivanov, M., Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Flores-Fuentes, W., & Rivas-Lopez, M. (2019). Mobile robot path planning using continuous laser scanning. In *Optoelectronics in machine vision-based theories and applications* (pp. 338–372). Hershey: IGI Global.

8. Kalla, P., Koon, R., Ravindranath, P., & Sudhakar, I. (2018). Coordinate reference frame technique for robotic planar path planning. *Materials Today: Proceedings*, 5(9), Part 3, 19073–19079.
9. Lindner, L., Sergiyenko, O., Rodríguez-Quinonez, J. C., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., et al. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*, 43(4), 360–369.
10. Lu, B., Chu, H. K., Huang, K. C., & Cheng, L. (2018). Vision-Based Surgical Suture Looping Through Trajectory Planning for Wound Suturing. *IEEE Transactions on Automation Science and Engineering*, 16(2), 542–556.
11. McInerney, A. (2013). *First steps in differential geometry: Riemannian, contact, symplectic. Undergraduate texts in mathematics*. New York: Springer.
12. Mouaddib, E. M., Sagawa, R., Echigo, T., & Yagi, Y. (2005). Stereovision with a single camera and multiple mirror. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (pp. 800–805).
13. Moura, J., McColl, W., Taykaldiranian, G., Tomiyama, T., & Erden, M. S. (2018). Automation of train cab front cleaning with a robot manipulator. *IEEE Robotics and Automation Letters*, 3(4), 3058–3065.
14. O’Gorman, L., Sammon, M. J., & Seul, M. (2008). *Practical algorithms for image analysis: Description, examples programs and projects* (2nd ed.). Cambridge: Cambridge University Press.
15. O’Neill, B. (2006). *Elementary differential geometry* (Revised 2nd ed.). Cambridge/Amsterdam: Academic/Elsevier.
16. Palenta, K., & Babiarz, A. (2014). KUKA robot motion planning using the 1742 smart camera. In A. Guca et al. (Eds.), *Man-Machine Interactions 3. Advances in intelligent systems and computing* (Vol. 242, pp. 115–122). Basel: Springer.
17. Perrollaz, M., Khorbotly, S., Cool, A., Yoder, J.-D., & Baumgartner, E. (2012). Teachless teach-repeat: Toward vision-based programming of industrial robots. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, May 2012 (pp. 409,414, 14–18).
18. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge: Cambridge University Press.
19. Raygoza-Perez, L. A., González-Gálvan, E. J., Loredó-Flores, A., Pastor, J. J., & Baumgartner, E. (2010). An enabling vision-based approach for non-calibrated, robot positioning task. *International Review of Automatic Control. Theory and Applications*, 3(6), 710–722.
20. Robinson, M. L. (2001). A structured lighting approach to image analysis for robotic applications using camera-space manipulation (Ph.D. Dissertation, University of Notre Dame)
21. Rodríguez-Quinonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., et al. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods’ approach. *Opto-Electronics Review*, 25(1), 24–32.
22. Real, O. R., Castro-Toscano, M. J., Rodríguez-Quinonez, J. C., Serginyenko, O., Hernández-Balbuena, D., Rivas-Lopez, M., et al. (2019) Surface measurement techniques in machine vision: Operation, applications, and trends. In *Optoelectronics in machine vision-based theories and applications* (pp. 79–104). Hershey: IGI Global.
23. Russ, J. C. (2007). *The image processing handbook* (5th ed.). Boca Raton: CRC Press.
24. Seelinger, M., Gonzalez-Galvan, E., Robinson, M., & Skaar, S. B. (1998). Towards a robotic plasma spraying operation using vision. *IEEE Robotics and Automation (Special Issue on Visual Servoing)*, 5(4), 33–36.
25. Senior, A. W., & Hampapur, M. L. (2005). Acquiring multi-scale images by Pan-Tilt-Zoom control and automatic multi-camera calibration. In *Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION’05)* (pp. 1–6).
26. Song, K.-T., & Tai, J.-C. (2006). Dynamic calibration of Pan-Tilt-Zoom cameras for traffic monitoring. *IEEE Transactions on Systems, Man and Cybernetics*, 36(5), 1091–1103.
27. Stücker, D. (1999). *Elementary geometric methods: Line segment intersection and inclusion in a polygon*. Technical Report, Department of Computer Science, University of Oldenburg.

Chapter 12

Unified Passivity-Based Visual Control for Moving Object Tracking



Flavio Roberti, Juan Marcos Toibero, Jorge A. Sarapura, Víctor Andaluz,
Ricardo Carelli, and José María Sebastián

Acronym

DOF Degree of freedom

12.1 Introduction

Robots can be defined as mechatronic devices, capable to perform some specific tasks in a workspace in an autonomous or semiautonomous way. Autonomy is related to the skill of getting environmental information by using some kind of exteroceptive sensors. Examples of these external sensors are proximity sensors, range finders, or vision systems. Present-day researches in the field of robot control are orientated to combine exteroceptive sensors with advanced control algorithms, making the robots capable of carrying out different tasks in unknown or semistructured environments, thus expanding the possible applications. Even when laser rangefinders are the most usually used sensors [1–3], visual sensing has increased in last years because of the quality and amount of information that images can bring [4–6].

F. Roberti (✉) · J. M. Toibero · J. A. Sarapura · R. Carelli
Instituto de Automática, UNSJ-CONICET, San Juan, Argentina
e-mail: froberti@inaut.unsj.edu.ar; mtoibero@inaut.unsj.edu.ar; jsarapura@inaut.unsj.edu.ar;
rcarelli@inaut.unsj.edu.ar

V. Andaluz
Universidad de las Fuerzas Armadas, Sangolquí, Ecuador
e-mail: vhandaluz1@espe.edu.ec

J. M. Sebastián
Centro de Automática y Robótica, Universidad Politécnica de Madrid, Madrid, Spain
e-mail: jsebas@etsii.upm.es

In accordance with the classification presented in Weiss et al. [7], visual servoing systems can be classified into image-based [8–10], when definition of control errors is made on the image plane; or position-based [6, 11], when definition of control errors is made on the 3D Cartesian space. Another classification of the visual servoing systems is based on the location of the vision system. Cameras can be “on-board” [2, 5, 11, 12], when they are located on the robot; or they can be located on a fixed position in the workspace [8, 13, 14]. Regarding the control algorithm design, visual servoing systems are classified into kinematics-based when the control algorithm design considers only the kinematic models [2, 5, 10, 15] or dynamics-based when the development of the control law considers also the dynamic model of the robot [8, 13, 16]. Even when kinematics-based control laws generally have an adequate performance, dynamic models have to be included in the control law design in order to achieve design specifications in high-speed motion tasks or transportation of weighty loads.

Usually, stability analysis of visual servoing systems is addressed in the context of Lyapunov theory [16–18]. Analysis based on the passivity properties represents an alternative possibility, habitually used in the analysis of manipulators’ controllers [19–22]. However, in [23], the passivity property of an interesting dynamic model of the unicycle-like mobile robots has been proved. In addition, some articles present passivity-based control laws for mobile robots [24, 25]. Solutions for the problem of path following control [26], pose problem [27], multi-robot coordination problem [28–30], and biped robot locomotion problem [31] have been proposed. Regarding artificial vision for mobile robot applications, Fujita et al. [32] and Kawai et al. [33] propose a visual motion observer based on passivity properties for being used in mobile platform control systems.

Concerning previously mentioned articles about passivity-based proposals, Fujita et al. [19] is one of the most valuable articles in visual servoing control based on passivity theory. Nevertheless, even when the stability analysis uses the system passivity properties, it is addressed in the Lyapunov theory framework. In the field of mobile robots, in [26] the problem of path planning control is addressed. One of the principal shortcomings of the work is that it represents a solution only for circular paths, and validation is made only through numerical simulations. As cited, Igarashi et al. [29] proposes an interesting solution for 3D attitude coordination of multi-robot systems. A heading velocity controller, allowing the robots to reach the desired relative orientation among them in 3D space, is presented. Validation is performed by both real experiments and numerical simulations. Kawai et al. [33] has reported a significant contribution in the area of visual sensing applied to mobile robots. Even when the visual motion observer represents the principal contribution, a pose control law for a mobile robot with a catadioptric on-board camera is also reported. The stability analysis is made in the context of Lyapunov theory by supposing a static target. Conclusion is supported by experimental data only for the visual observer.

This chapter presents a unified passivity-based visual control structure for robots with on-board camera. The proposed control law makes the robot able to perform a moving target tracking in its workspace. The control structure presented in this chapter can be applied not only for robotic arms but also for mobile robots and for mobile manipulators. Taking advantage of the passivity properties of the control

system and considering exact knowledge of the target velocity, the asymptotic convergence of the control errors to zero is proved. Later, a robustness analysis based on L_2 -gain performance is performed, thus proving that control errors are ultimately bounded even when bounded errors exist in the estimation of the target velocity. A key aspect of the chapter is that the complete theoretical analysis is made in an input-output system context, in lieu of using the passivity properties in a Lyapunov framework. In addition, the presented control system consists of a real tracker instead of a robust position control law; therefore, the velocity of the target is considered in whole system analysis. Finally, dynamic model of the robot is included in the control design, making the proposed control structure appropriate for robotic applications where high performances are required.

The rest of the chapter is organized as follows. Section 12.2 presents the kinematic and the dynamic models used along this chapter. Section 12.3 describes the proposed unified control structure, where the stability and the robustness analysis are included. Section 12.4 presents the numerical simulation and experimental results for all considered robotic systems, and finally, Sect. 12.5 states the conclusions.

12.2 System Models

This section presents the dynamic models of the considered robotic systems, as well as the kinematic model of the vision system when it is mounted on-board the robots.

Before starting with the chapter development, it would be useful to define the main variables that will be used in the next sections. In Table 12.1, a list of symbols adopted for these main variables is presented.

Table 12.1 Nomenclature used for main variables

Symbol	Variable description
ξ	Vector of image features
\mathbf{J}	Total image Jacobian matrix
μ	Vector of the robot velocities
\mathbf{J}_T	Target Jacobian matrix
\mathbf{V}_T	Cartesian velocities of the target
$\tilde{\xi}$	Vector of image feature errors defined as $\tilde{\xi}(t) = \xi(t) - \xi_d$
\mathbf{J}^\dagger	Inverse or pseudo-inverse of \mathbf{J}
\mathbf{v}_ξ	Auxiliary variable defined as $\mathbf{v}_\xi = \mathbf{J}^T \xi$
$\mathbf{v}_{\tilde{\xi}}$	Auxiliary variable defined as $\mathbf{v}_{\tilde{\xi}} = \mathbf{J}^T \mathbf{K} \begin{pmatrix} \tilde{\xi} \\ \xi \end{pmatrix}$
μ_r^c	Kinematic control law
$\tilde{\mu}$	Robot velocity error defined as $\tilde{\mu} = \mu - \mu_r^c$
\mathbf{w}	External disturbance
$\hat{\mathbf{v}}_T$	Estimation of \mathbf{v}_T
$\tilde{\mathbf{v}}_T$	Estimation errors of \mathbf{v}_T

12.2.1 *Dynamic Model of the Robotic Manipulator*

If friction or any other disturbances are not considered, the dynamic model of an n -link rigid robot manipulator can be written as

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_r \quad (12.1)$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times 1}$ is the vector of centripetal and Coriolis torques; $\mathbf{q} \in \mathbb{R}^{n \times 1}$ is the vector of joint displacements; $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ is the vector of gravitational torques; $\boldsymbol{\tau}_r \in \mathbb{R}^{n \times 1}$ is the vector of applied joint torques.

12.2.2 *Dynamic Model of the Mobile Robot*

The mathematical model that represents the dynamics of a mobile robot is [23]

$$\mathbf{H}\dot{\mathbf{u}} + \mathbf{C}(\mathbf{u})\mathbf{u} + \mathbf{F}(\mathbf{u})\mathbf{u} = \mathbf{u}_r \quad (12.2)$$

where $\mathbf{u} \in \mathbb{R}^{2 \times 1}$ is a vector containing the linear velocity and rotational velocity of the mobile robot and $\mathbf{u}_r \in \mathbb{R}^{2 \times 1}$ represents the vector of velocity input signals. Matrix $\mathbf{H} \in \mathbb{R}^{2 \times 2}$ is diagonal, constant, and positive definite; matrix $\mathbf{C}(\mathbf{u}) \in \mathbb{R}^{2 \times 2}$ is skew symmetric; matrix $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^{2 \times 2}$ is (considering realistic assumptions [23]) symmetric, lower bounded, and positive definite. Additionally, mapping $\mathbf{u}_r \rightarrow \mathbf{u}$ is strictly output passive. For extra details related to model (12.2), readers can refer to [23].

12.2.3 *Dynamic Model of the Mobile Manipulator*

The mathematical representation of the mobile manipulator's dynamics used along this chapter is obtained from the dynamic model presented in [34], but including low-level controllers for the actuators. These controllers consist of PD-like velocity controllers, thus obtaining velocity references as input of this new dynamic model [35]

$$\mathbf{M}(\mathbf{q}) \dot{\mathbf{u}} + \mathbf{C}(\mathbf{q}, \mathbf{u})\mathbf{u} + \mathbf{g}(\mathbf{q}) = \mathbf{u}_r \quad (12.3)$$

where $\mathbf{u} \in \mathbb{R}^n$ is the vector of velocities of the mobile manipulator (which includes the linear velocity and the angular velocity of the mobile platform and the arm's joint velocities), $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is a positive definite matrix, $\mathbf{C}(\mathbf{q}, \mathbf{u})\mathbf{u} \in \mathbb{R}^n$, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$, and $\mathbf{u}_r \in \mathbb{R}^n$ is the vector of input signals (velocity references). $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_m]^T =$

$[\mathbf{q}_p^T \mathbf{q}_a^T]^T \in \mathbb{R}^m$ represents the generalized coordinates of the mobile manipulator, where $\mathbf{q}_p \in \mathbb{R}^{mp}$ represents the generalized coordinates of the mobile platform and $\mathbf{q}_a \in \mathbb{R}^{ma}$ represents the generalized coordinates of the robotic arm. For extra details related to model (12.3), readers can refer to [35].

12.2.4 Kinematic Model of the Vision System

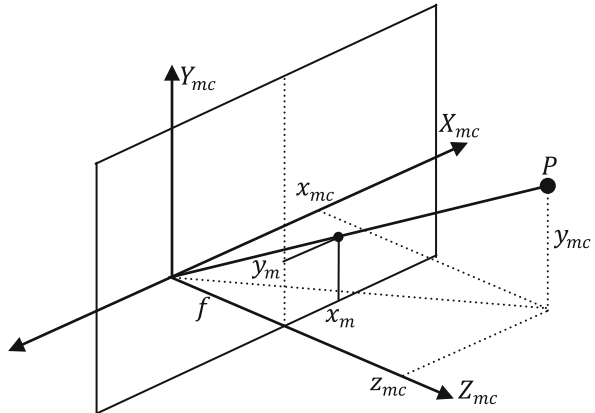
A single camera projects the 3D space onto the 2D image plane, where the vision sensor is placed. Researchers have proposed numerous projection models that represent the process of image formation [36]. Perspective projection model is one of the most used. In this projection model, a 3D orthogonal coordinate system associated with the camera is defined such that X_{mc} and Y_{mc} axes are a base for the image plane, and Z_{mc} axis coincides with the optical axis, as depicted in Fig. 12.1. The origin of the orthogonal Cartesian system is placed at the camera lens focus. Therefore, a point \mathbf{P} in the 3D space described by $\mathbf{P} = [x_{mc} \ y_{mc} \ z_{mc}]^T$ on the coordinate system associated with the vision camera will be mapped onto the image plane as a 2D point with coordinates $(x_m \ y_m)$ given by [36]

$$x_m = f \frac{x_{mc}}{z_{mc}}; \quad y_m = f \frac{y_{mc}}{z_{mc}} \quad (12.4)$$

where f is the camera's focal length, expressed in pixels.

Therefore, if we consider an on-board vision system, taking into account the pin-hole camera model (12.4) and the kinematic model of the used robot and defining some convenient set of image features ξ , the time variation of these image features can be expressed as a function of the movement of both the target and the robot as

Fig. 12.1 Camera projection model



$$\dot{\tilde{\xi}} = \mathbf{J}\boldsymbol{\mu} + \mathbf{J}_T\mathbf{v}_T \quad (12.5)$$

where $\boldsymbol{\mu}$ represents the vector of the robot's velocities (linear velocity and angular velocity of the mobile platform; and/or joint velocities of the robotic arm); \mathbf{v}_T represents the Cartesian velocities of the target; \mathbf{J} is the total Jacobian matrix, and \mathbf{J}_T is the object Jacobian matrix. It is important to remark that \mathbf{J} includes not only the image Jacobian of the vision camera but also the kinematic models of the robot (with its corresponding nonholonomic or holonomic constraints). As an example, Appendix 1 shows mathematical derivations to obtain the kinematic model (12.5) for the mobile robot with an on-board camera.

12.3 Passivity-Based Visual Controller Design

This section of the chapter addresses the design of a visual controller based on image feature errors (image-based control) to make a robot with an on-board camera able to perform a moving object tracking task on the 3D workspace, while the image feature errors $\tilde{\xi}(t)$ asymptotically converge to zero. Therefore, control objective can be defined as

$$\lim_{t \rightarrow \infty} \tilde{\xi}(t) = 0 \quad (12.6)$$

First, a visual controller based on the kinematic model is designed, and by assuming that the velocity of the moving object is perfectly known and perfect velocity tracking, it is proved that image feature errors asymptotically converge to zero. Next, assumption of perfect velocity tracking is disregarded, and new conditions for the dynamics-based controller are obtained to prove that image feature errors asymptotically convergence to zero as well under this realistic consideration.

As mentioned before, full design of the proposed controller is addressed in the framework of the input-output system theory, specifically by taking advantage of the system's passivity properties. These passivity properties have been generally used for the nonlinear system stability analysis [37], principally for interconnected and cascade structured systems [38–40], and represent an alternative analysis to the Lyapunov theory. Formal definitions associated with input-output system theory and passivity properties of functional space operators used in this chapter are given in Appendix 2.

12.3.1 Passivity Property of the Vision System

Previous works have proved that a perspective projection vision system placed at the end effector of a robotic arm is passive when considering a static object in the 3D

space [19]. Now, if a moving object is considered, that is, by modeling the vision system as in (12.5), the passivity property of the vision system can also be proved.

Proposition 1 Mapping $(\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) \rightarrow \mathbf{v}_\xi$, which represents the vision system (12.5), is passive.

Proof Let us consider the positive function $V_\xi = \frac{1}{2} \boldsymbol{\xi}^T \boldsymbol{\xi}$. Then, calculating its derivative and introducing (12.5), the following expression is obtained

$$\dot{V}_\xi = \frac{1}{2} \boldsymbol{\xi}^T \dot{\boldsymbol{\xi}} = \boldsymbol{\xi}^T (\mathbf{J}\boldsymbol{\mu} + \mathbf{J}_T \mathbf{v}_T) \tag{12.7}$$

Now, integrating (12.7) over $[0, T]$,

$$\int_0^T \dot{V}_\xi dt = \int_0^T (\boldsymbol{\xi}^T (\mathbf{J}\boldsymbol{\mu} + \mathbf{J}_T \mathbf{v}_T)) dt \tag{12.8}$$

$$\int_0^T (\mathbf{J}^T \boldsymbol{\xi})^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt = V_\xi(T) - V_\xi(0) \geq -V_\xi(0)$$

where \mathbf{J}^\dagger represents the inverse or pseudo-inverse of \mathbf{J} . Defining $\mathbf{v}_\xi = \mathbf{J}^T \boldsymbol{\xi}$ and according to Definition 6, it can be concluded that mapping $(\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) \rightarrow \mathbf{v}_\xi$ is passive. •

12.3.2 Design of the Kinematic-Based Controller

Considering at this point the regulation problem on the image plane, the vector of image features error is defined as $\tilde{\boldsymbol{\xi}}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi}_d$, where $\boldsymbol{\xi}_d$ represents the desired feature vector on the image plane. In this context, the proposed kinematic control structure is shown in the block diagram of Fig. 12.2.

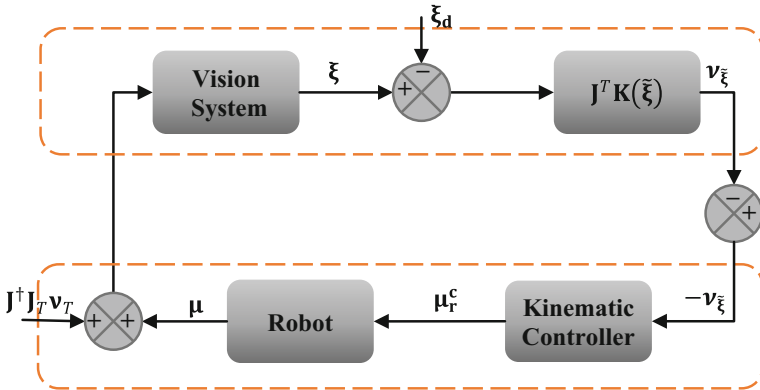


Fig. 12.2 Block diagram of the proposed kinematic visual control system

Proposition 2 *Let us consider the kinematic control system represented in the block diagram of Fig. 12.2, where the “vision system” block is modeled as (12.5) and assuming perfect velocity tracking by the robot, that is, $\boldsymbol{\mu} \equiv \boldsymbol{\mu}_r^c$ (kinematic-based controller). If $\mathbf{K}(\tilde{\boldsymbol{\xi}})$ is a positive definite gain matrix designed such that $\mathbf{v}_{\tilde{\xi}} \in L_\infty$ for any values of image features and $\boldsymbol{\mu}_r^c = -\mathbf{K}_c \mathbf{v}_{\tilde{\xi}} - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T = -\mathbf{K}_c \mathbf{J}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) \tilde{\boldsymbol{\xi}} - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T$ with $\mathbf{K}_c > 0$, then mapping $(\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) \rightarrow \mathbf{v}_{\tilde{\xi}}$ (representing the upper sub-system of block diagram of Fig. 12.2) is passive, and mapping $-\mathbf{v}_{\tilde{\xi}} \rightarrow (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T)$ (representing the lower sub-system of block diagram of Fig. 12.2) is strictly input passive.*

Proof First, the passivity property of the vision system when considering the regulation problem on the image plane is addressed. With this aim, let us take the following positive definite function:

$$V_{\tilde{\xi}} = \int_0^{\tilde{\xi}^T} \boldsymbol{\eta}^T \mathbf{K}(\boldsymbol{\eta}) d\boldsymbol{\eta} \quad (12.9)$$

Then, the time derivative of function $V_{\tilde{\xi}}$ is $\dot{V}_{\tilde{\xi}} = \tilde{\boldsymbol{\xi}}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) \dot{\tilde{\boldsymbol{\xi}}} = \tilde{\boldsymbol{\xi}}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) (\mathbf{J}\boldsymbol{\mu} + \mathbf{J}_T \mathbf{v}_T)$. Integrating $\dot{V}_{\tilde{\xi}}$ over the interval $[0, T]$,

$$\int_0^T \dot{V}_{\tilde{\xi}} dt = \int_0^T \tilde{\boldsymbol{\xi}}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) (\mathbf{J}\boldsymbol{\mu} + \mathbf{J}_T \mathbf{v}_T) dt \quad (12.10)$$

and defining

$$\mathbf{v}_{\tilde{\xi}} = \mathbf{J}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) \tilde{\boldsymbol{\xi}} \quad (12.11)$$

the following expression is obtained:

$$\int_0^T \mathbf{v}_{\tilde{\xi}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt \geq -V_{\tilde{\xi}}(0) \quad (12.12)$$

concluding that the mapping $(\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) \rightarrow \mathbf{v}_{\tilde{\xi}}$ is passive.

Now, recalling the perfect velocity tracking assumption, that is, $\boldsymbol{\mu} \equiv \boldsymbol{\mu}_r^c$, and the definition of the control law,

$$\boldsymbol{\mu}_r^c = -\mathbf{K}_c \mathbf{v}_{\tilde{\xi}} - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T = -\mathbf{K}_c \mathbf{J}^T \mathbf{K}(\tilde{\boldsymbol{\xi}}) \tilde{\boldsymbol{\xi}} - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T; \mathbf{K}_c > 0 \quad (12.13)$$

and substituting (12.13) in (12.12), the following expression is obtained

$$\begin{aligned} \int_0^T \mathbf{v}_{\tilde{\xi}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt &= \int_0^T \mathbf{v}_{\tilde{\xi}}^T (-\mathbf{K}_c \mathbf{v}_{\tilde{\xi}} - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt = \\ &- \int_0^T \mathbf{v}_{\tilde{\xi}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\xi}} dt \leq -\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\xi}}^T \mathbf{v}_{\tilde{\xi}} dt \end{aligned} \quad (12.14)$$

or

$$\int_0^T -\mathbf{v}_{\tilde{\xi}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt \geq \lambda_{\min}(\mathbf{K}_c) \|\mathbf{v}_{\tilde{\xi}}\|_{2,T}^2 \quad (12.15)$$

concluding that the mapping $-\mathbf{v}_{\tilde{\xi}} \rightarrow (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T)$, that is, control law defined in (12.13), is strictly input passive (see Definition 7). •

Therefore, the control system proposed, represented in Fig. 12.2, is composed of the interconnections of passive subsystems.

12.3.2.1 Analysis of the Kinematic Control System

In order to analyze the proposed kinematic control system, let us add the expressions that define its passivity properties (12.12) and (12.15)

$$\begin{aligned} 0 &\geq -V_{\tilde{\xi}}(0) + \lambda_{\min}(\mathbf{K}_c) \|\mathbf{v}_{\tilde{\xi}}\|_{2,T}^2 \\ \|\mathbf{v}_{\tilde{\xi}}\|_{2,T}^2 &\leq \frac{V_{\tilde{\xi}}(0)}{\lambda_{\min}(\mathbf{K}_c)} \end{aligned} \quad (12.16)$$

which implies that $\mathbf{v}_{\tilde{\xi}} \in L_{2e}$. Also, recall that $\mathbf{K}(\tilde{\xi})$ is designed such that $\mathbf{v}_{\tilde{\xi}} \in L_\infty$ and $\dot{\mathbf{v}}_{\tilde{\xi}} \in L_\infty$ since robot velocities are also bounded (by definition of the control law (12.13)). Then, Barbalat's lemma [41] allows to conclude that

$$\mathbf{v}_{\tilde{\xi}} \rightarrow 0 \quad \text{with} \quad t \rightarrow \infty \quad (12.17)$$

Then, assuming that the target is out of any singular position and recalling that $\mathbf{v}_{\tilde{\xi}} = \mathbf{J}^T \mathbf{K}(\tilde{\xi}) \tilde{\xi}$, the condition (12.17) implies that

$$\tilde{\xi} \rightarrow 0 \quad \text{with} \quad t \rightarrow \infty \quad (12.18)$$

thus achieving the control objective.

12.3.2.2 Particular Consideration for the Mobile Manipulator

In case of considering a mobile manipulator, an extra term can be added to the kinematic control law (12.13) without altering previous analysis, since $\mathbf{J}(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) = \mathbf{0}$

$$\boldsymbol{\mu}_r^c = -\mathbf{K}_c \mathbf{v}_{\xi}^c - \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \mathbf{k}_1 \tan \mathbf{h}(\mathbf{k}_2 \boldsymbol{\Lambda}) \quad (12.19)$$

So, taking advantage of the redundancy of mobile manipulators, additional performances can be successfully achieved, namely obstacle avoidance, singular configuration avoidance, keeping image features in the field of view, or several performance criteria optimization.

With the aim of illustrating the previous concept, this chapter considers two independent secondary objectives: obstacle avoidance by the mobile platform and the prevention of singular configuration by controlling the system's manipulability [42]. Therefore, a suitable definition for $\boldsymbol{\Lambda}$ is [35]

$$\boldsymbol{\Lambda} = \left[-u_{\text{obs}} \ \omega_{\text{obs}} \ f_1(\tilde{\theta}_1) \ \dots \ f_{n_a}(\tilde{\theta}_{n_a}) \right]^T \quad (12.20)$$

u_{obs} and ω_{obs} are the linear velocity and the rotational velocity of the mobile platform which make the robot able to achieve the obstacle avoidance objective, when the obstacles do not interfere with the robotic arm. So, the arm can deal with the main objective while the platform deals with the obstacle avoidance by resourcing to the null space configuration. u_{obs} and ω_{obs} are calculated as a function of a fictitious repulsion force [35]. This fictitious force is defined by the relative distance between the robot and the obstacle d , and the angle of incidence to the obstacle α . Therefore, control laws are defined as [35]

$$\begin{aligned} u_{\text{obs}} &= Z^{-1} (k_{u_{\text{obs}}} (d_0 - d) \left[\frac{\pi}{2} - |\alpha| \right]) \\ \omega_{\text{obs}} &= Z^{-1} (k_{\omega_{\text{obs}}} (d_0 - d) \text{sgn}(\alpha) \left[\frac{\pi}{2} - |\alpha| \right]) \end{aligned} \quad (12.21)$$

where d represents the relative distance from robot to obstacle, d_0 is the distance at which the obstacle starts to be avoided, $k_{u_{\text{obs}}}$ and $k_{\omega_{\text{obs}}}$ are positive design constants, Z is the mechanical impedance that defines the interaction between the robot and the environment, and α is the angle of incidence to the obstacle.

12.3.3 Dynamic Compensation Controller

Passivity properties of previous kinematic control system have been obtained by assuming perfect velocity tracking, that is, $\boldsymbol{\mu} \equiv \boldsymbol{\mu}_r^c$. However, when the robotic task requires, for example, high-speed movements or transportation of heavy loads, dynamic effects could not be ignored making previous assumption to be invalid. In this context, robot dynamics has to be considered in the control system design to achieve acceptable performances. Hence, the main goal of this new control law is to compensate robot dynamics in order to reduce the difference between the desired velocity and the robot's real velocity. Inputs of this dynamic control law are the desired velocities generated by the kinematic control law, and it calculates the new control actions to be sent to the robot (Fig. 12.3).

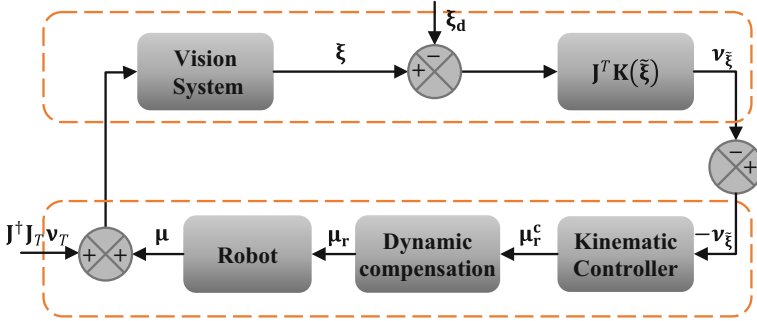


Fig. 12.3 Block diagram of the proposed control system with dynamic compensation

Therefore, disregarding perfect velocity tracking assumption, a velocity error will exist, which is expressed as $\tilde{\boldsymbol{\mu}}(t) = \boldsymbol{\mu}(t) - \boldsymbol{\mu}_r^c(t)$, motivating the design of a dynamic compensation control law.

The dynamic compensation controller $\boldsymbol{\mu}_r$ has to be designed for $\tilde{\boldsymbol{\mu}} \in L_2 \cap L_\infty$, which implies that $\boldsymbol{\mu} \in L_\infty$ since $\boldsymbol{\mu}_r^c \in L_\infty$ has been proved. For example, a controller based on feedback linearization can be used, which allows to easily prove the conditions for $\tilde{\boldsymbol{\mu}}$.

Then, the behavior of control error $\tilde{\boldsymbol{\xi}}$ should be studied. With this purpose, introducing the error $\tilde{\boldsymbol{\mu}}(t)$ previously defined and the kinematic control law (12.13) into (12.14), the following expression is obtained:

$$-\int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (\boldsymbol{\mu} + \mathbf{J}^T \mathbf{J}_T \mathbf{v}_T) dt = \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt - \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \tilde{\boldsymbol{\mu}} dt \quad (12.22)$$

Then, by adding (12.22) with (12.12) and after some mathematical operations, we obtain

$$\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt \leq V_{\tilde{\boldsymbol{\xi}}}(0) + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \tilde{\boldsymbol{\mu}} dt \quad (12.23)$$

$$\lambda_{\min}(\mathbf{K}_c) \left\| \mathbf{v}_{\tilde{\boldsymbol{\xi}}} \right\|_{2,T}^2 \leq V_{\tilde{\boldsymbol{\xi}}}(0) + \left\| \mathbf{v}_{\tilde{\boldsymbol{\xi}}} \right\|_{2,T} \left\| \tilde{\boldsymbol{\mu}} \right\|_{2,T} \quad \forall T \in [0, \infty) \quad (12.24)$$

By recalling the restriction imposed on the dynamic compensation controller $\tilde{\boldsymbol{\mu}} \in L_2$, inequality (12.24) holds only for $\left\| \mathbf{v}_{\tilde{\boldsymbol{\xi}}} \right\|_{2,T} < \infty$, which implies that $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_2$. This result implies that the property $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_2$ holds after including the dynamical controller. So, recalling that $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_\infty$ and $\dot{\mathbf{v}}_{\tilde{\boldsymbol{\xi}}} \in L_\infty$, the control objective is fulfilled, that is,

$$\tilde{\boldsymbol{\xi}} \rightarrow \mathbf{0} \quad \text{with} \quad t \rightarrow \infty \quad (12.25)$$

12.3.4 Robustness Analysis

Kinematic controller (12.13) supposes the exact knowledge of target velocity \mathbf{v}_T ; however, this is not true in practice, where it should be estimated from visual sensing of the target by using, for example, an $\alpha - \beta$ filter [43]. This estimation of \mathbf{v}_T can produce undesirable effects on the control error behavior. Therefore, it is necessary to carry out a robustness analysis in the context of the L_2 -gain performance criterion. Hence, an external disturbance \mathbf{w} is defined as a function of the estimation error of $\tilde{\mathbf{v}}_T$ and velocity error $\tilde{\boldsymbol{\mu}}$, and it has to be proved that [44]

$$\int_0^T \|\tilde{\boldsymbol{\xi}}\|^2 dt \leq \gamma^2 \int_0^T \|\mathbf{w}\|^2 dt; \quad \forall T > 0 \quad (12.26)$$

which means that mapping from \mathbf{w} to $\tilde{\boldsymbol{\xi}}$ has finite L_2 -gain.

Let us define $\mathbf{w} = \tilde{\boldsymbol{\mu}} + \mathbf{J}^\dagger \mathbf{J}_T \tilde{\mathbf{v}}_T$, where $\tilde{\mathbf{v}}_T = \mathbf{v}_T - \hat{\mathbf{v}}_T$, and suppose that \mathbf{w} is bounded. Considering now that the control law (12.13) is calculated with the estimated value of the object velocity $\hat{\mathbf{v}}_T$, (12.22) is modified as follows:

$$\begin{aligned} \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt &= \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (\tilde{\boldsymbol{\mu}}(t) + \boldsymbol{\mu}_r^c(t) + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt = \\ \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \tilde{\boldsymbol{\mu}} dt &- \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (-\mathbf{J}^\dagger \mathbf{J}_T \hat{\mathbf{v}}_T + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt \end{aligned} \quad (12.27)$$

$$\int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt = - \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \tilde{\boldsymbol{\mu}} dt + \int_0^T \tilde{\mathbf{v}}_T^T \mathbf{J}^\dagger \mathbf{J}_T \tilde{\mathbf{v}}_T dt \quad (12.28)$$

$$\int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T (\boldsymbol{\mu} + \mathbf{J}^\dagger \mathbf{J}_T \mathbf{v}_T) dt = - \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{w} dt \quad (12.29)$$

By subtracting (12.29) from the expression that defines the passivity property of the vision system (12.12), we obtain

$$0 \geq -V_{\tilde{\boldsymbol{\xi}}}(0) + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}_c \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt - \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{w} dt \quad (12.30)$$

Then

$$\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}} dt \leq V_{\tilde{\boldsymbol{\xi}}}(0) + \int_0^T \mathbf{v}_{\tilde{\boldsymbol{\xi}}}^T \mathbf{w} dt \quad (12.31)$$

or by defining $\varepsilon = \lambda_{\min}(\mathbf{K}_c)$ and reminding the inner product in the space L_{2e} (Definition 5),

$$\varepsilon \left\| \mathbf{v}_{\tilde{\xi}} \right\|_{2,T}^2 \leq \left\langle \mathbf{v}_{\tilde{\xi}}^T, \mathbf{w} \right\rangle_T + V_{\tilde{\xi}}(0) \quad (12.32)$$

Now, by adding the positive term $\frac{1}{2} \left\langle \frac{1}{\sqrt{\varepsilon}} \mathbf{w} - \sqrt{\varepsilon} \mathbf{v}_{\tilde{\xi}}^T, \frac{1}{\sqrt{\varepsilon}} \mathbf{w} - \sqrt{\varepsilon} \mathbf{v}_{\tilde{\xi}}^T \right\rangle_T$ to the second member of (12.32), the inequality holds. After some mathematical operations, the following expression is obtained:

$$\varepsilon \left\| \mathbf{v}_{\tilde{\xi}} \right\|_{2,T}^2 \leq \left\langle \mathbf{v}_{\tilde{\xi}}^T, \mathbf{w} \right\rangle_T + \frac{1}{2\varepsilon} \langle \mathbf{w}, \mathbf{w} \rangle_T + \frac{\varepsilon}{2} \left\langle \mathbf{v}_{\tilde{\xi}}^T, \mathbf{v}_{\tilde{\xi}}^T \right\rangle_T - \left\langle \mathbf{v}_{\tilde{\xi}}^T, \mathbf{w} \right\rangle_T + V_{\tilde{\xi}}(0) \quad (12.33)$$

$$\left\| \mathbf{v}_{\tilde{\xi}} \right\|_{2,T}^2 \leq \frac{1}{\varepsilon^2} \|\mathbf{w}\|_{2,T}^2 + V_{\tilde{\xi}}(0) \quad (12.34)$$

Now, for $\|\mathbf{w}\|_2^2$ such that $\left\| \mathbf{v}_{\tilde{\xi}} \right\|_2^2$ be bounded away from its saturation value, and after introducing (12.11) into (12.34) it can be concluded that

$$\left\| \tilde{\xi}^T \right\|_{2,T}^2 \leq \frac{1}{\lambda_{\min}(\mathbf{M}) \varepsilon^2} \|\mathbf{w}\|_{2,T}^2 + V_{\tilde{\xi}}(0) \quad (12.35)$$

where $\mathbf{M} = \mathbf{K}^T \left(\tilde{\xi} \right) \mathbf{J} \mathbf{J}^T \mathbf{K} \left(\tilde{\xi} \right)$.

Finally, after integrating (12.35) over the interval $[0, T]$, it can be concluded that the mapping $\mathbf{w} \rightarrow \tilde{\xi}$ has finite L_2 -gain $\leq \gamma$, with $\gamma = \frac{1}{\varepsilon \sqrt{\lambda_{\min}(\mathbf{M})}}$. That is, the proposed control system is robust to disturbance \mathbf{w} in accordance with L_2 -performance criterion (disturbance attenuation in L_2 -gain norm or energy attenuation). In this context, the parameter γ can be understood as a quantitative performance index of the proposed control system when estimation errors exist.

12.4 Simulation and Experimental Results

The passivity-based control structure showed in this chapter has been evaluated through both simulations and real experiments in the three different considered robots. The first robotic system is a mobile robot Pioneer 3DX manufactured by MobileRobots Inc., with a monocular vision system ($f = 850$ pixels) mounted on-board. The moving object consists of a cylindrical target placed on another mobile robot Pioneer 3AT. The experimental equipment is shown in Fig. 12.4a.

Then, some experiments were performed with a mobile manipulator. It consists of a mobile robot Pioneer 3AT and a robotic arm CYTON Alpha 7 degrees of freedom (DOF) (only 3 DOF are used in the experiment). The mobile base has a laser rangefinder only for the obstacle detection, and a JMK Mini Camera (model: JK-805) is placed at the end effector of the robotic arm. The moving object to be

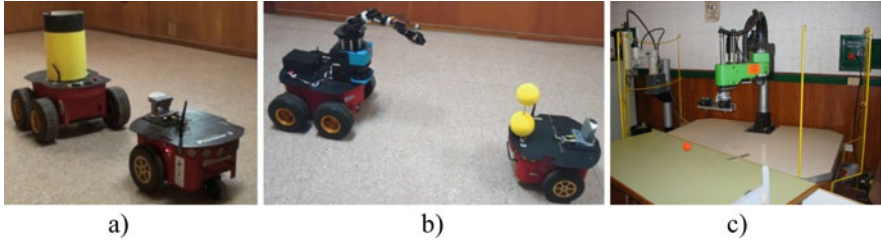


Fig. 12.4 Experimental setup. (a) Laboratory robots used in experiments described in Sect. 12.4.1. (b) Laboratory robots used in experiments described in Sect. 12.4.2. (c) Robotic manipulator Bosch SR-800 considered in simulation results described in Sect. 12.4.3

tracked is placed on another mobile robot Pioneer 3AT. The experimental equipment is shown in Fig. 12.4b.

Finally, the dynamic and kinematic models of the Bosch SR-800 robot manipulator (Fig. 12.4b) with an eye-in-hand stereo vision system is considered.

12.4.1 Mobile Robot

In order to implement and evaluate the passivity-based controller in a mobile robot, the feature selection described in Appendix 1 is used. In all experiments, \mathbf{v}_T is unknown for the control system and its estimation is carried out by an $\alpha - \beta$ filter [43]. However, any other adequate estimation algorithm could be used.

Matrix $\mathbf{K}(\tilde{\boldsymbol{\xi}})$ is defined such that $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_\infty$

$$\mathbf{K}(\tilde{\boldsymbol{\xi}}) = \text{diag} \left(\frac{k_1}{(a_1 + |\tilde{\xi}_1|)(b_1 + d_m)(c_1 + |x_m|^2)}, \frac{k_2}{(a_2 + |\tilde{\xi}_2|)(b_2 + d_m^2)(c_2 + |x_m|)} \right) \quad (12.36)$$

$$k_i, a_i, b_i, c_i > 0$$

Note that definition of matrix $\mathbf{K}(\tilde{\boldsymbol{\xi}})$ is related to the Jacobian matrix associated with the definition of image features.

The design constants of the control law are $k_1 = 0.25, k_2 = 15, a_1 = 70, b_1 = 20, c_1 = 30, a_2 = 100, b_2 = 30, c_2 = 20, \mathbf{K}_c = \text{diag}(70, 4)$; and the desired image features are $\boldsymbol{\xi}_d = [0 \ 270]^T$ for the first experiment and $\boldsymbol{\xi}_d = [0 \ 170]^T$ for the second and

third experiments. Note that these desired image features can be transformed into robot–target relative posture obtaining $d = 0.63m$ and $\varphi = 0^\circ$ for the first experiment; and $d = 1m$ and $\varphi = 0^\circ$ for the second and third ones. Although relative posture values are not needed in control laws, they are useful for interpreting the experimental results.

Experiment 12.4.1.1 In this experiment, the mobile robot has to navigate from an initial position to a final destination relative to the target, since a static object is considered. This final position is defined by the image features. Experimental results are shown in Figs. 12.5, 12.6, 12.7, and 12.8. Figure 12.5 illustrates control errors, in terms of image features, Fig. 12.6 represents the velocity commands sent to the mobile robot, Fig. 12.7 shows the robot trajectory during the experiment, and finally, Fig. 12.8 depicts the robot–target relative position.

Experiment 12.4.1.2 In this experiment, the object moves along a straight-line path with a fixed linear velocity. Hence, the mobile platform has to track the object while maintaining a constant robot–object relative position. Similar to previous experiment, the results are shown in Figs. 12.9, 12.10, 12.11, and 12.12. Figure 12.9 illustrates control errors, in terms of image features, Fig. 12.10 represents the velocity commands sent to the mobile robot, Fig. 12.11 shows the robot trajectory during the experiment, and finally, Fig. 12.12 depicts the robot–target relative position.

Experiment 12.4.1.3 In this final test, the object moves with nonconstant linear and angular velocities. Figure 12.13 illustrates control errors, in terms of image features,

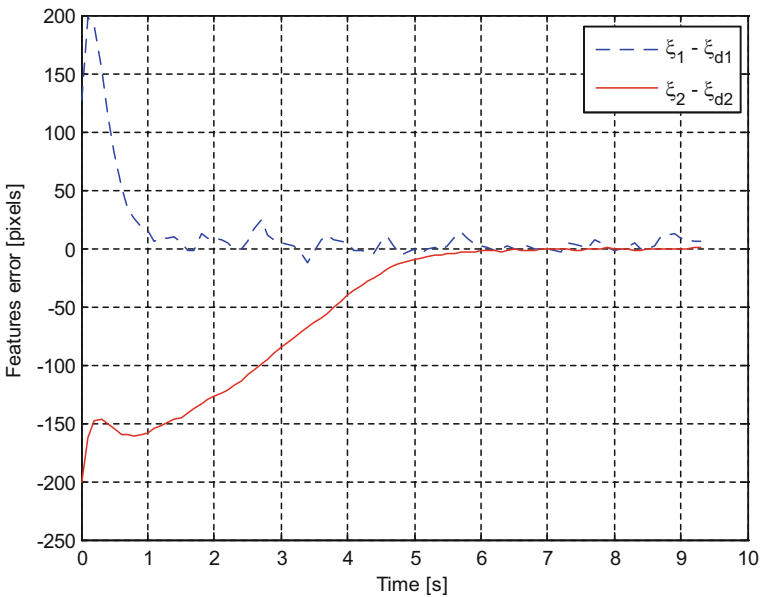


Fig. 12.5 Time evolution of the image features error

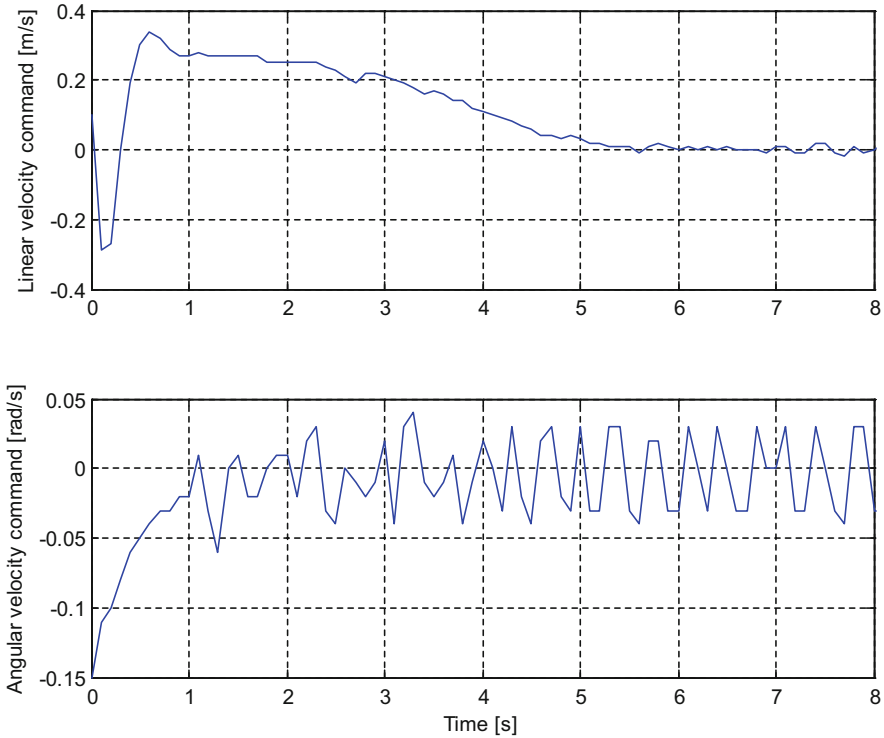


Fig. 12.6 Time evolution of the control commands

Fig. 12.14 represents the velocity commands sent to the mobile robot, Fig. 12.15 shows the robot trajectory during the experiment, and finally, Fig. 12.16 depicts the robot–target relative position.

12.4.2 Mobile Manipulator

In order to implement and evaluate the passivity-based controller in a mobile manipulator, it is considered as the object to be tracked consisting of two spheres vertically aligned. Diameter of these spheres is known and denoted as D . Additionally, the image characteristics are defined as $\xi = [u_1 \ v_1 \ v_2]^T$ (Fig. 12.17).

In all experiments, \mathbf{v}_T is unknown for the control system, and its estimation is carried out by an $\alpha - \beta$ filter [43]. In these experiments, controller (12.19) is used in order to consider two secondary objectives: obstacle avoidance by the mobile base, and the prevention of singular configuration by controlling the manipulability of the system.

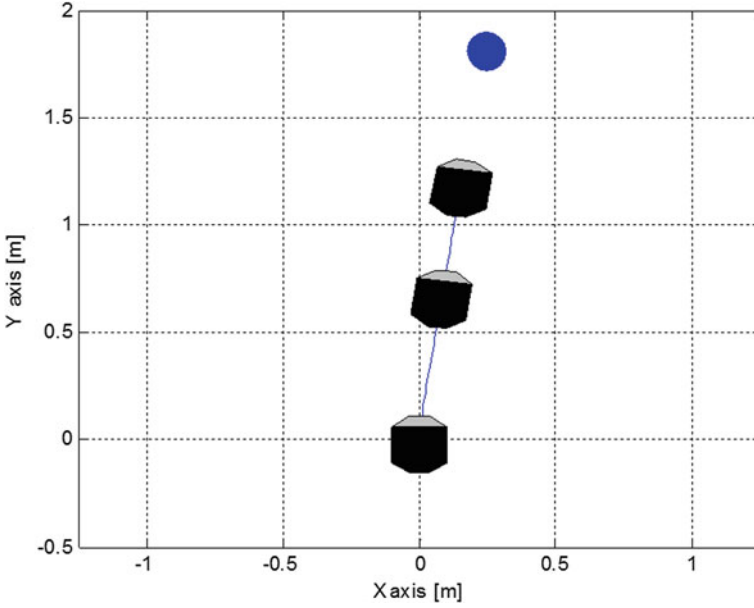


Fig. 12.7 Trajectory described by the mobile robot in the workspace

Matrix $\mathbf{K}(\tilde{\boldsymbol{\xi}})$ was defined such that $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_{\infty}$

$$\mathbf{K}(\tilde{\boldsymbol{\xi}}) = \text{diag} \left(\frac{k_i}{1 + |\tilde{\xi}_i|} \right) \quad (12.37)$$

The arm joints' angles which maximize the manipulability of the arm are found through offline simulation. Hence, the desired joint positions are $\theta_{1d} = 0$ rad, $\theta_{2d} = 0.6065$ rad, and $\theta_{3d} = 1.2346$ rad. For all the experiments the initial robot configuration is $\mathbf{q} = [0 \text{ m } 0 \text{ m } 0 \text{ rad } 0 \text{ rad } 0 \text{ rad } 0 \text{ rad}]^T$. Finally, gain matrices and gain constants of the controller are set as $\mathbf{K}_c = \mathbf{I}$; $k_{\Delta i} = 1$; $k_{u_{\text{obs}}} = 0.5$; $k_{\omega_{\text{obs}}} = 0.9$; $\mathbf{k}_1 = \text{diag} \left(0.7 \ 1 \ 0.1 \ 0.1 \ 0.1 \right)$; $\mathbf{k}_2 = \text{diag} \left(0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2 \right)$; and $k_i = 0.15$.

Experiment 12.4.2.1 In this experiment, the target moves to describe a rectilinear path for about 20 s and after that it abruptly stops. This experiment intends to test the control system performance when high errors in the estimation of \mathbf{v}_T exist. Initial and desired image feature vectors are $\boldsymbol{\xi}(0) = [-150 \ 20 \ -100]^T$ and $\boldsymbol{\xi}_d = [0 \ 60 \ -60]^T$, respectively, both of them expressed in pixels. Results of this experiment are shown in Figs. 12.18, 12.19, 12.20, and 12.21. Behavior of control

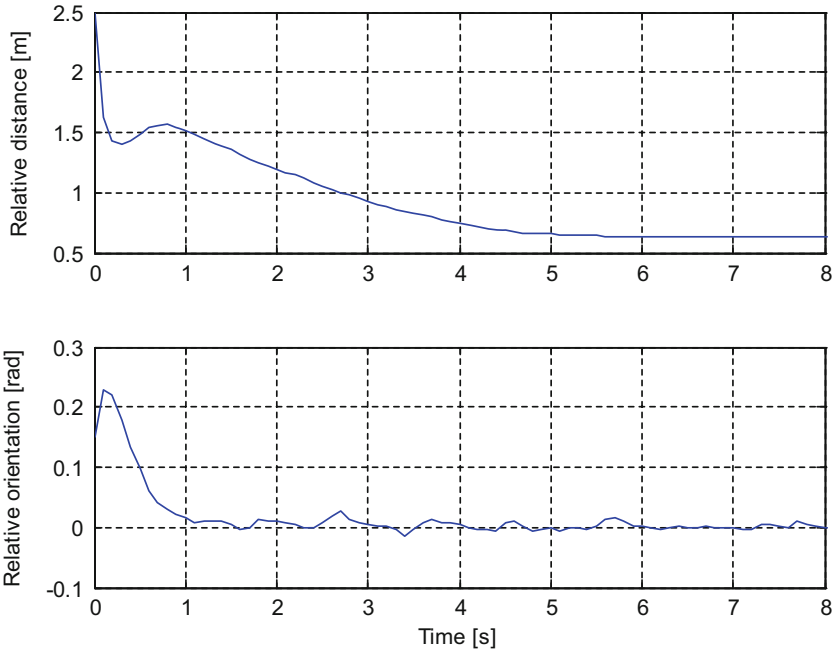


Fig. 12.8 Relative posture between the mobile robot and the target

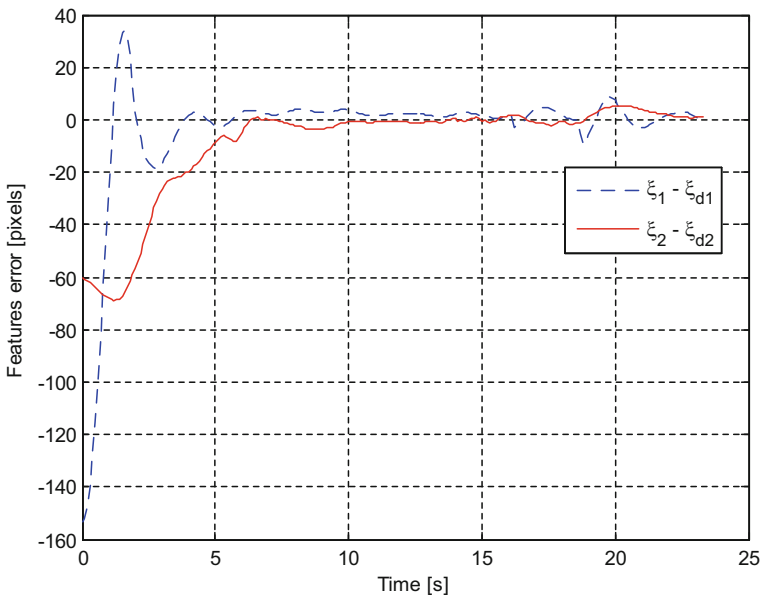


Fig. 12.9 Time evolution of the image features error

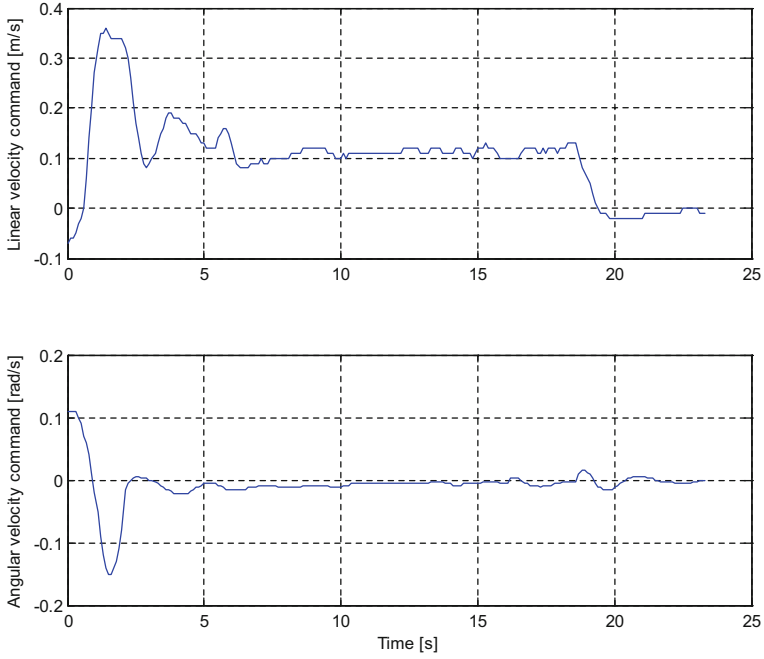


Fig. 12.10 Time evolution of the control commands

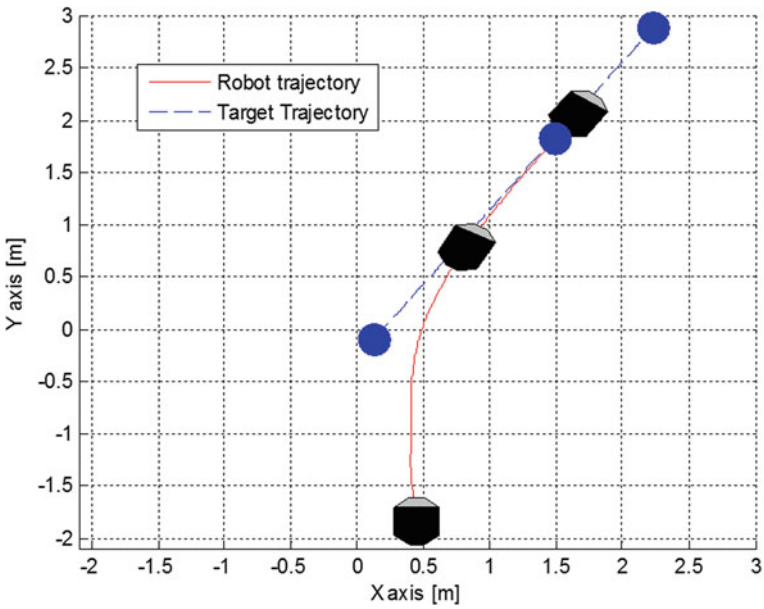


Fig. 12.11 Trajectory described by the mobile robot in the workspace

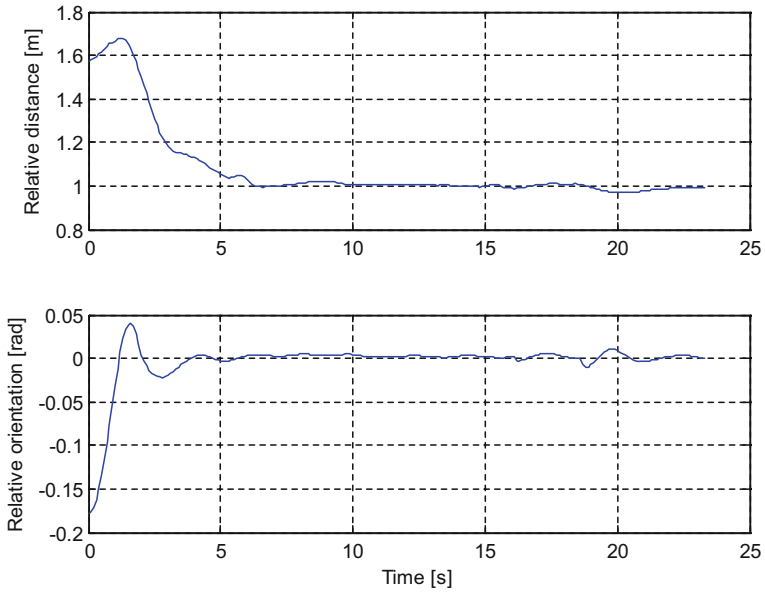


Fig. 12.12 Relative posture between the mobile robot and the target

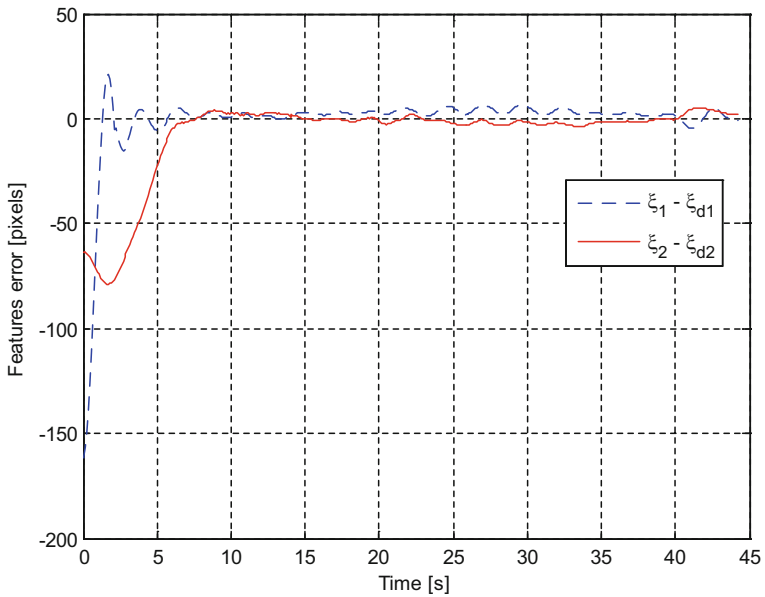


Fig. 12.13 Time evolution of the image feature error

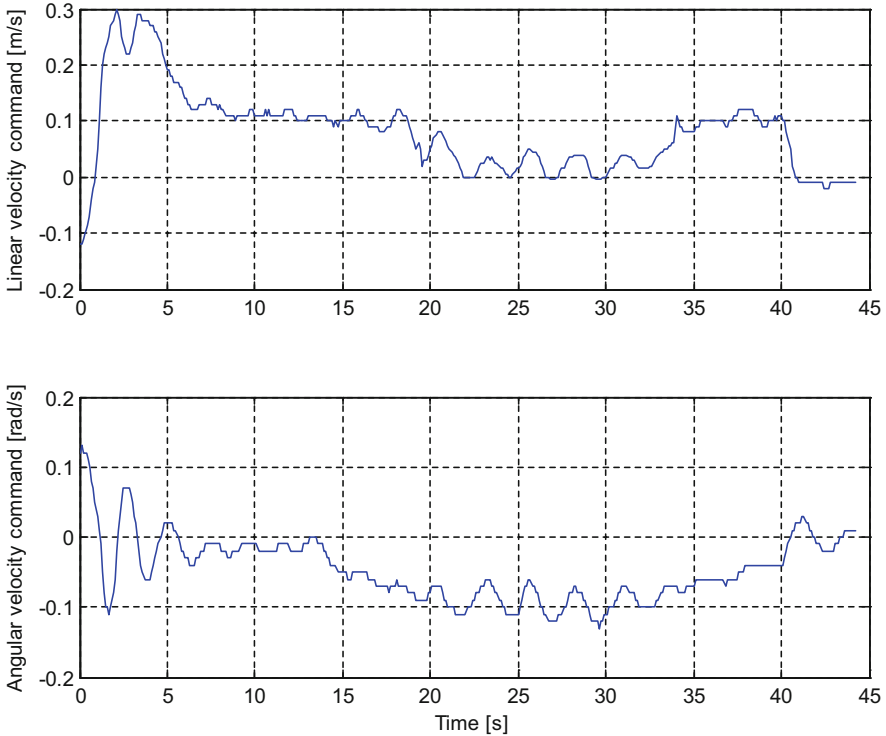


Fig. 12.14 Time evolution of the control commands

error $\tilde{\xi}(t)$ is shown in Fig. 12.18. It can be seen that $\tilde{\xi}(t)$ is ultimately bounded, reaching final values of $\max(|\tilde{\xi}_i|) < 8$ pix. Figures 12.19 and 12.20 illustrate the velocity commands for the mobile platform as well as for the robotic arm; finally, Fig. 12.21 shows the norm of the estimated object velocity. It is important to highlight that image feature errors remain bounded even when high errors in velocity estimation exist, such as the estimation errors which can be appreciated when the experiment begins.

Experiment 12.4.2.2 The complexity of the experiment is now increased. The object moves to describe a non-rectilinear path, and the mobile platform has to avoid an unexpected static obstacle. The obstacle does not restrict visual sensing; therefore, the robotic arm is able to continue performing the object tracking while the mobile base avoids the obstacle. Initial and desired image feature vectors are $\xi(0) = [-150 \ 20 \ -115]^T$ and $\xi_d = [0 \ 60 \ -60]^T$, respectively, both of them expressed in pixels. Experimental results are shown in Figs. 12.22, 12.23, 12.24, and 12.25. Behavior of control error $\xi(t)$ is shown in Fig. 12.22. Figures 12.23 and 12.24 illustrate the velocity commands for the mobile platform as well as for the robotic arm. Figure 12.25 illustrate trajectories described by both the robot and the target in

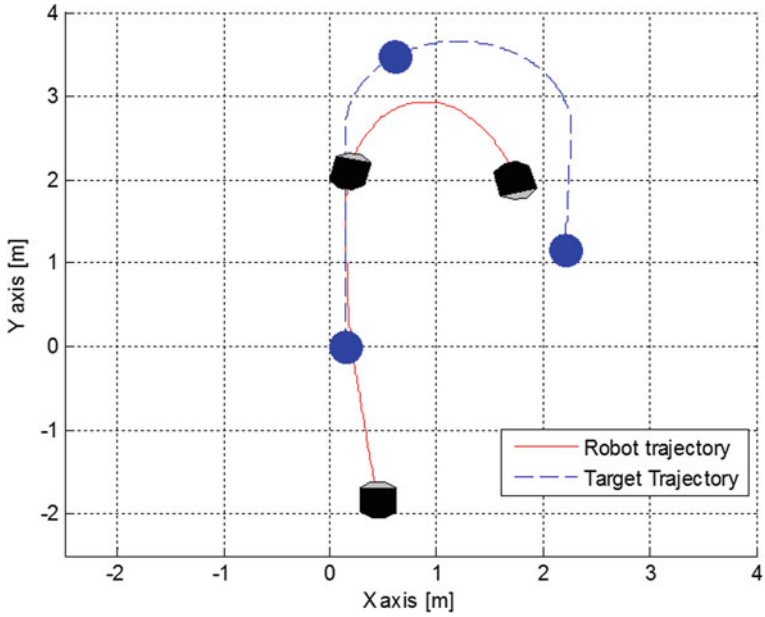


Fig. 12.15 Trajectory described by the mobile robot in the workspace

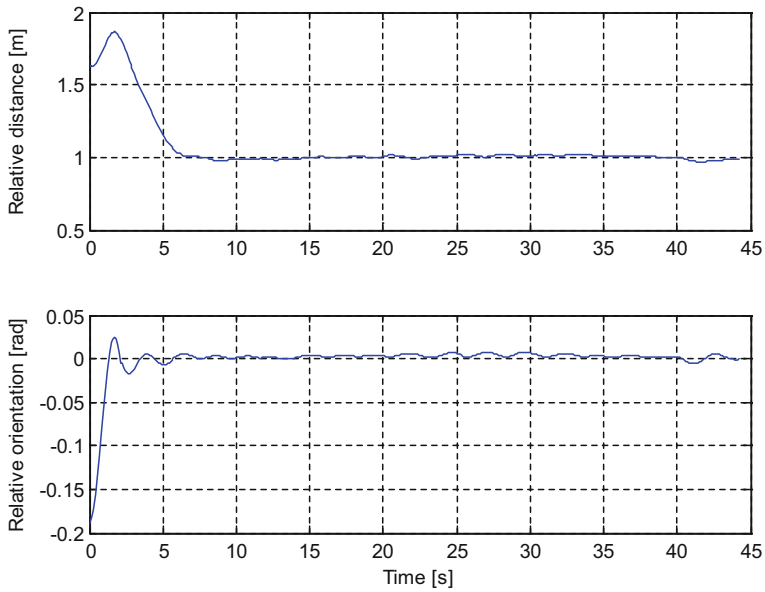


Fig. 12.16 Relative posture between the mobile robot and the target

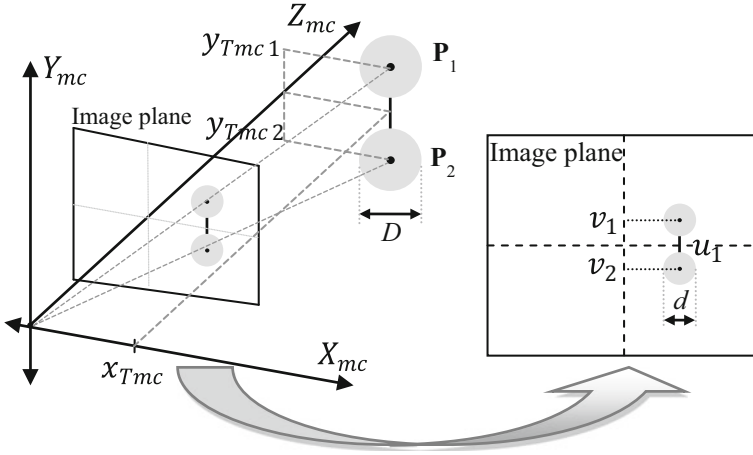


Fig. 12.17 Feature selection for the mobile manipulator

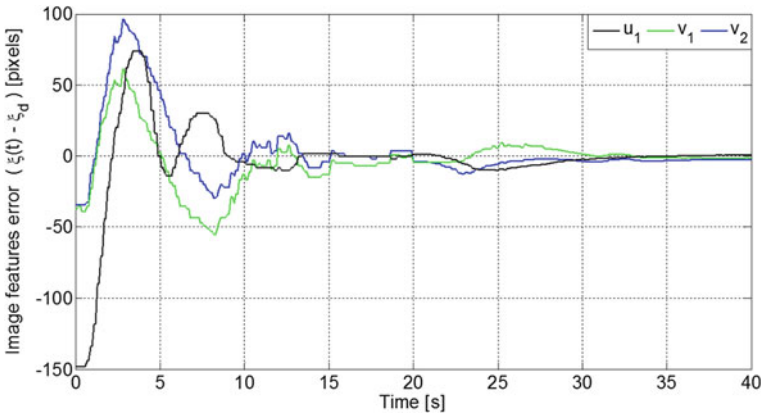


Fig. 12.18 Time evolution of control errors $\tilde{\xi}(t)$

the workspace. Figures 12.22, 12.23, and 12.24 (for $17s < t < 32s$ approximately) show the evolution of involved signals of the control system which make the robot to avoid the obstacle while following the object. In Fig. 12.25 at position labeled as 3, the obstacle avoidance action can also be observed. It is important to notice that when the obstacle is surpassed, maximum manipulability configuration is reached again by the robotic manipulator, fulfilling this specific secondary objective. Figure 12.22 also allows to observe that the image feature errors maintain bounded even during the obstacle avoidance task and in spite of the estimation errors of v_T .

Experiment 12.4.2.3 Numerical simulation is carried out for the system performance evaluation when the object describes trajectories in the 3D space. That is, it also modifies its z-coordinate while performing a movement in the $X - Y$ plane.

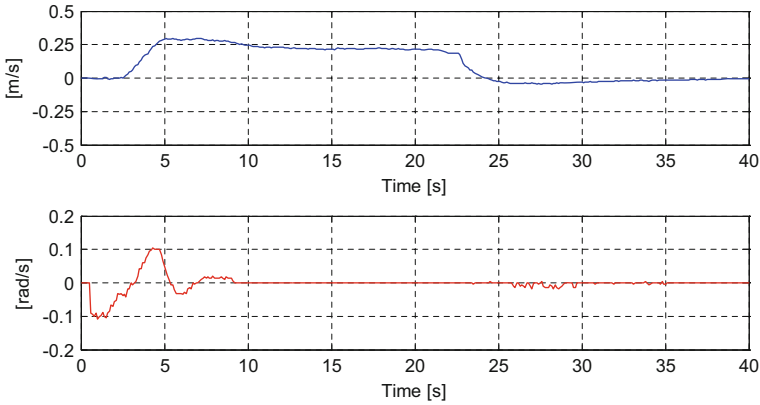


Fig. 12.19 Velocity commands to the mobile platform

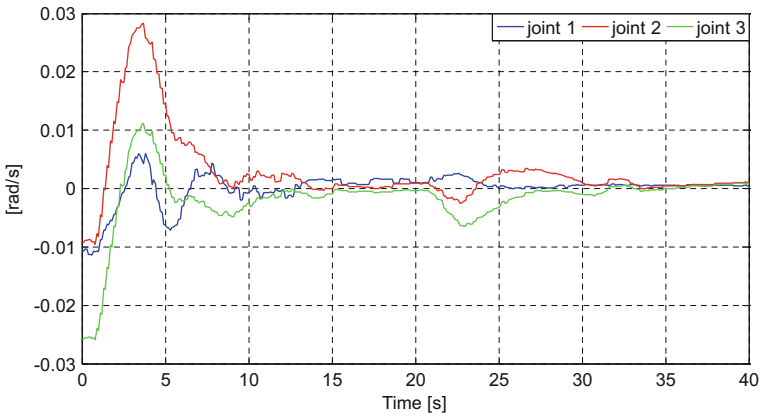


Fig. 12.20 Joint velocity commands to the robotic arm

Simulation considers the robot dynamics and the perfect knowledge of v_T . Initial and desired image feature vectors are $\xi(0) = [212 \ -50 \ 120]^T$ and $\xi_d = [0 \ 50 \ -50]^T$, respectively, both of them expressed in pixels. Results of this experiment are shown in Figs. 12.26, 12.27, 12.28, and 12.29. Behavior of control error $\tilde{\xi}(t)$ is shown in Fig. 12.26. As expected for a simulation, control errors asymptotically converge to zero. Figure 12.27 illustrates trajectories described by both the robot and the target in the 3D workspace. Finally, Figs. 12.28 and 12.29 illustrate the velocity commands for the mobile platform as well as for the robotic arm.

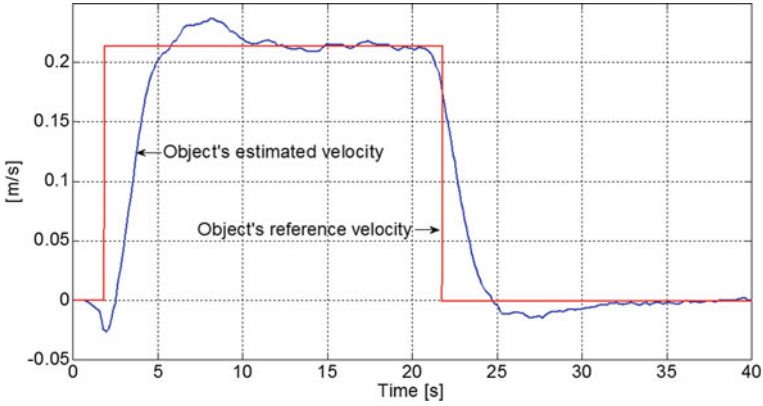


Fig. 12.21 Norm of the estimated object velocity

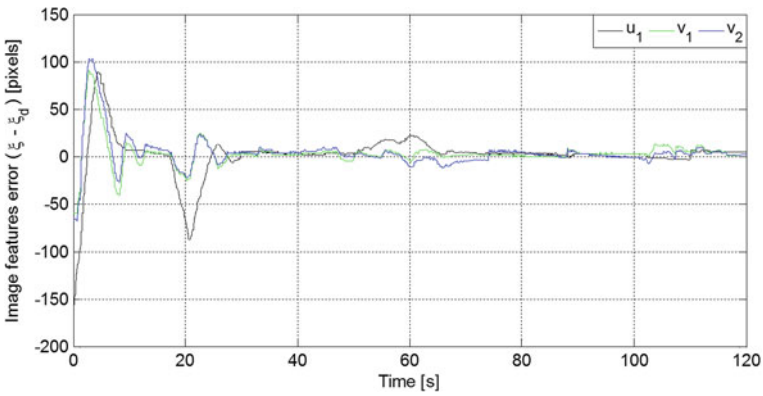


Fig. 12.22 Time evolution of control errors $\tilde{\xi}(t)$

12.4.3 Robotic Manipulator

For the evaluation of the passivity-based controller in the industrial manipulator, simulations considering the dynamic and kinematic models of the Bosch SR-800 robot manipulator (Fig. 12.4c) with an eye-in-hand stereo vision system have been carried out. Additionally, a moving target formed by a single sphere is considered; therefore, the image feature vector is defined as $\xi = [u_l \ u_r \ v_l]^T$ (Fig. 12.30). In all experiments, v_T is unknown for the control system, and its estimation is carried out by an $\alpha - \beta$ filter [43].

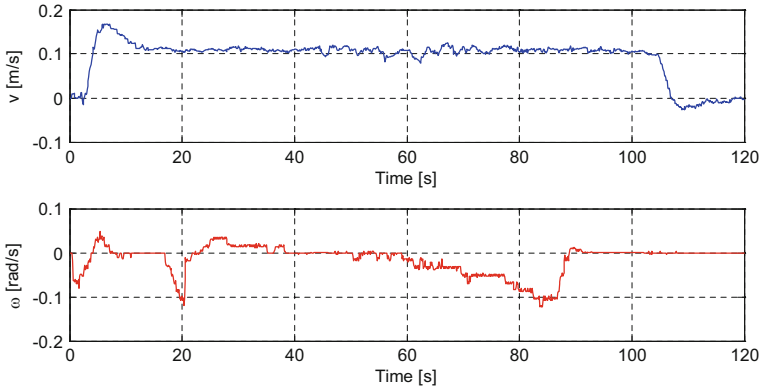


Fig. 12.23 Velocity commands to the mobile platform

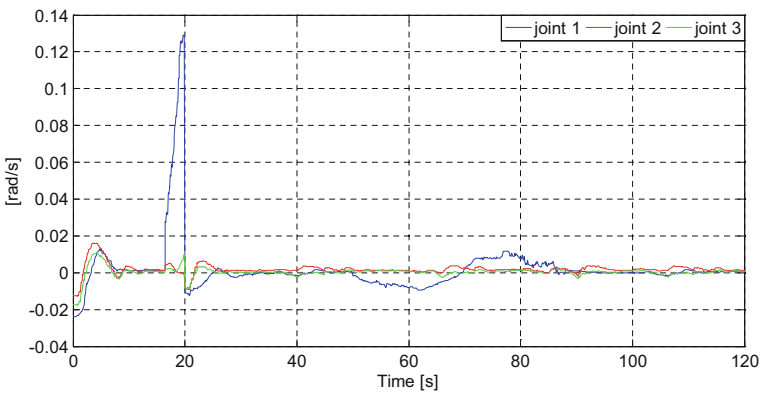


Fig. 12.24 Joint velocity commands to the robotic arm

For the experiments, the initial robot configuration is $\mathbf{q} = \begin{bmatrix} 0.1396\text{rad} \\ 1.117\text{rad} \\ -0.01\text{rad} \end{bmatrix}^T$.

Matrix $\mathbf{K}(\tilde{\boldsymbol{\xi}})$ is defined such that $\mathbf{v}_{\tilde{\boldsymbol{\xi}}} \in L_\infty$,

$$\mathbf{K}(\tilde{\boldsymbol{\xi}}) = \text{diag} \left(\frac{k_i}{a + |\tilde{\xi}_i|} \right); \quad a, k_i > 0; \quad \text{with } i = 1, 2, 3. \quad (12.38)$$

And gain matrices and gain constants of the controller are set as $\mathbf{K}_c = \mathbf{I}$; $a = 100$; $k_1 = 0.16$; $k_2 = 0.16$ and $k_3 = 0.008$.

Experiment 12.4.3.1 The first experiment is performed by considering that the target is moving, describing a straightline trajectory in the 3D space defined as

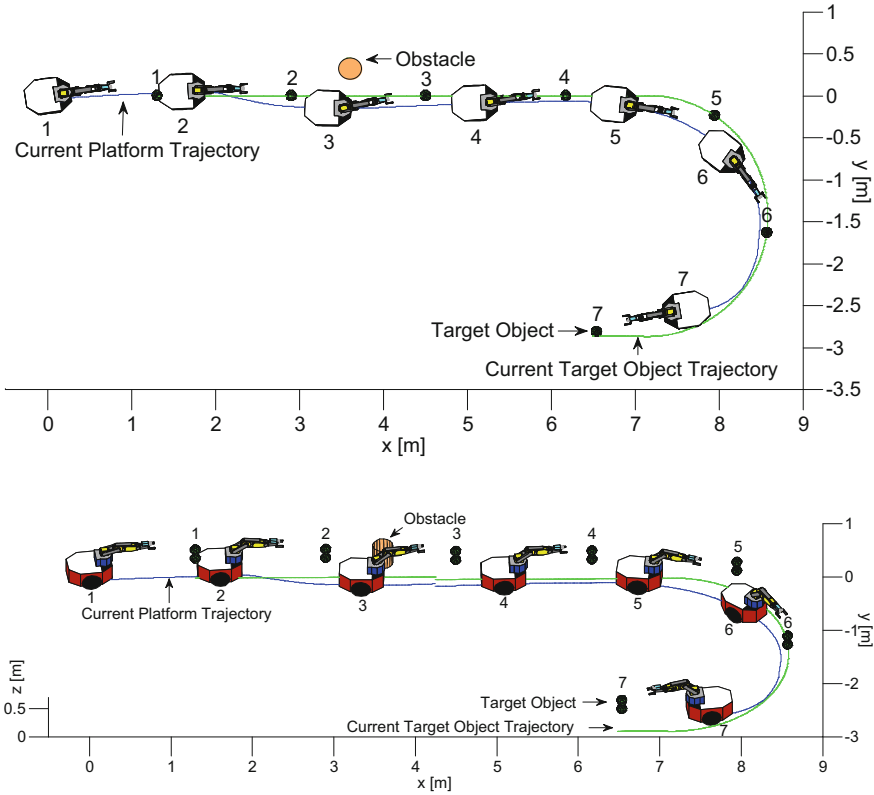


Fig. 12.25 Described trajectories of the mobile manipulator and the target. The positions of both the target and the mobile manipulator at the same time instants are depicted. Seven different time instants are shown

$$\dot{\mathbf{P}} = \begin{bmatrix} \mathbf{P}(0) = [2\mathbf{m} \ 0.2\mathbf{m} \ 0.57\mathbf{m}]^T \\ -0.025\mathbf{m/s} \ -0.035\mathbf{m/s} \ -0.0071\mathbf{m/s} \end{bmatrix} \tag{12.39}$$

expressed in the coordinate system attached to the robot base. Additionally, the desired image features are defined as $\xi_d = [400 \ 240 \ 160]^T$. Figures 12.31, 12.32, and 12.33 show the results obtained for this experiment. Figure 12.31 shows the time evolution of the image feature errors, and Fig. 12.32 illustrates the trajectories in the 3D space described by both the robot’s end effector and the moving target. Finally, Fig. 12.33 shows the evolution of the image features in the image planes of the stereo vision system.

Experiment 12.4.3.2 In this experiment, a complex scenario is considered. The target is moving to describe a helical trajectory in the 3D space defined as

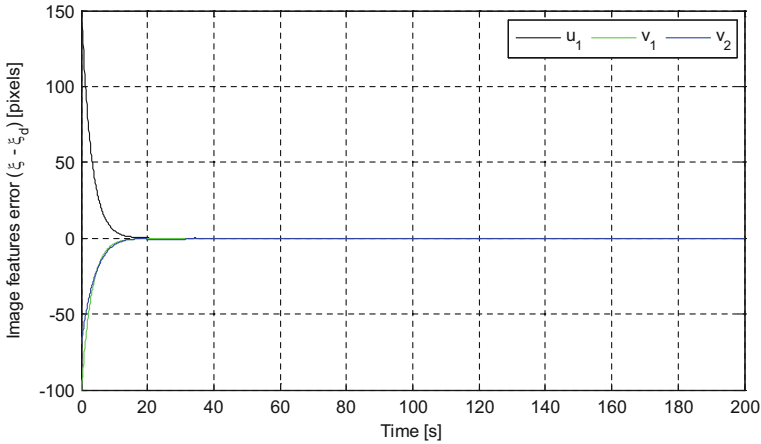


Fig. 12.26 Time evolution of control error $\tilde{\xi}(t)$

$$\begin{aligned} \mathbf{P}(0) &= [2\mathbf{m} \ 0.1\mathbf{m} \ 0.57\mathbf{m}]^T \\ \dot{\mathbf{P}} &= \left[\begin{array}{c} -0.225 \times \sin(1.5t) \ 0.025 \mathbf{m/s} \ 0.225 \times \cos(1.5t) \mathbf{m/s} \ -0.007 \mathbf{m/s} \end{array} \right] \end{aligned} \quad (12.40)$$

expressed in the coordinate system attached to the robot base. Additionally, the desired image features are defined as $\xi_d = [400 \ 240 \ 160]^T$. Figures 12.34, 12.35, and 12.36 show the results obtained for this experiment. Figure 12.34 shows the time evolution of the image feature errors, and Fig. 12.35 illustrates the trajectories in the 3D space described by both the robot's end effector and the moving target. Finally, Fig. 12.36 shows the evolution of the image features in the image planes of the stereo vision system.

Experimental and simulation results demonstrate the viability of implementing the proposed control structure based on the passivity theory, and they also show its good performance not only for two different classes of mobile robots but also for robotic manipulators. Experimental data show that the proposed control system makes the robot capable to achieve and keep the desired posture with respect to a moving object, reaching high performances, even when the velocity of the moving object is nonconstant.

An important issue to remark is that, notwithstanding the unknown velocity of the moving object, the passivity-based control law decreases the image feature errors to values near to zero. These results confirm the theoretical conclusions about not only the asymptotic convergence to zero of the control errors but also the robustness property of the proposed control system against the estimation errors of the object's velocity.

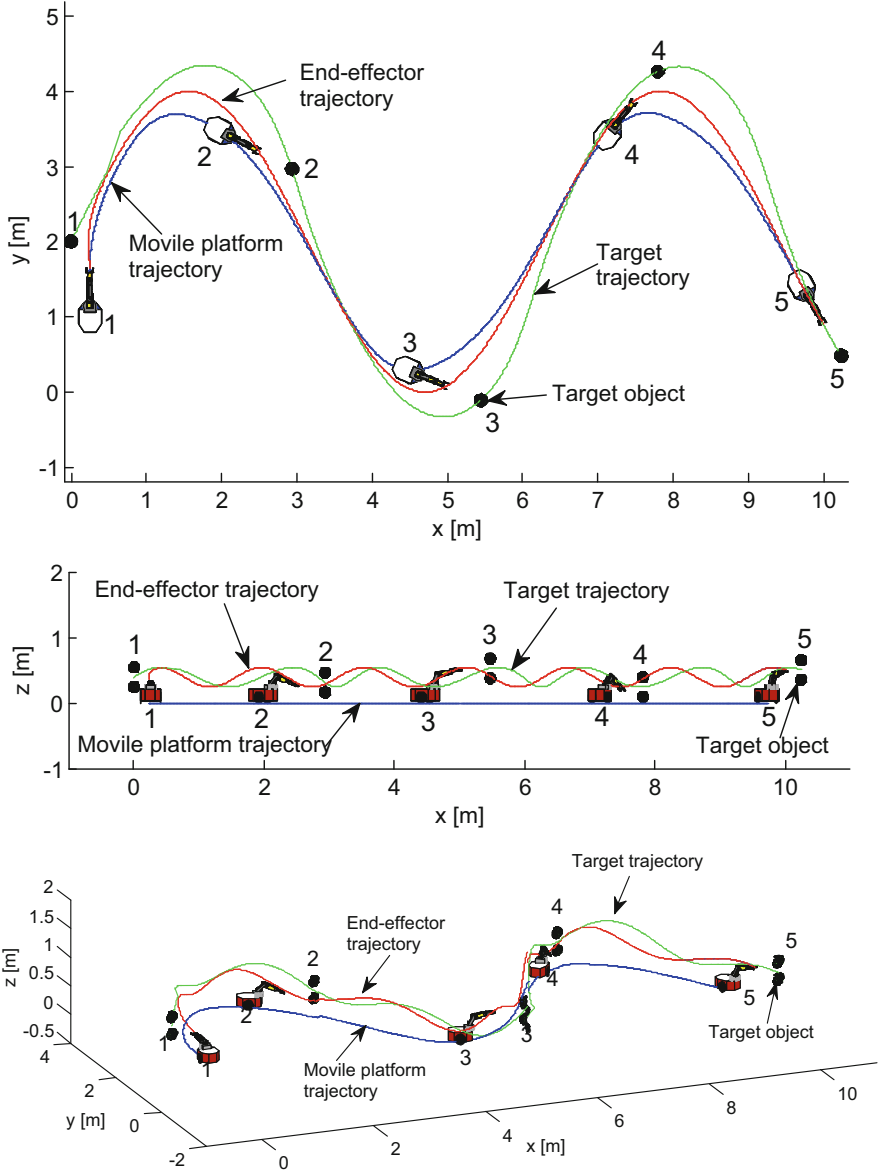


Fig. 12.27 Described trajectories of the mobile manipulator and the target. The position of both the target and the mobile manipulator at the same time instants are depicted. Five different time instants are shown

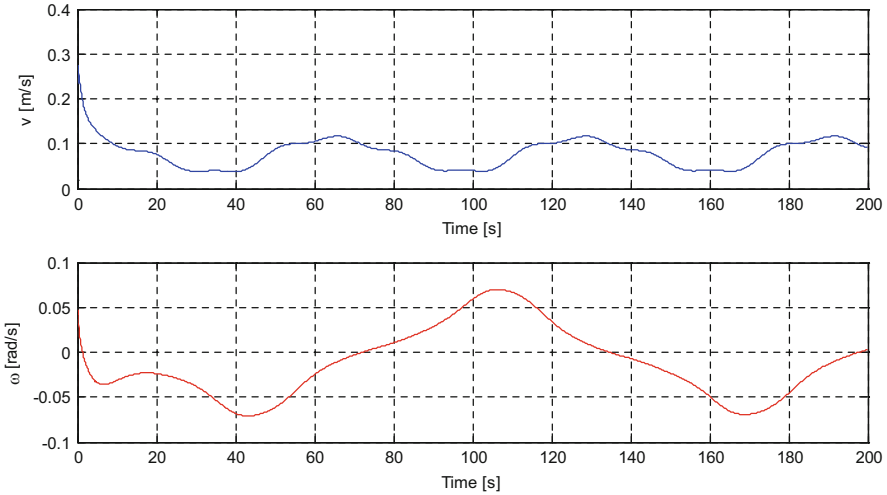


Fig. 12.28 Velocity commands to the mobile platform

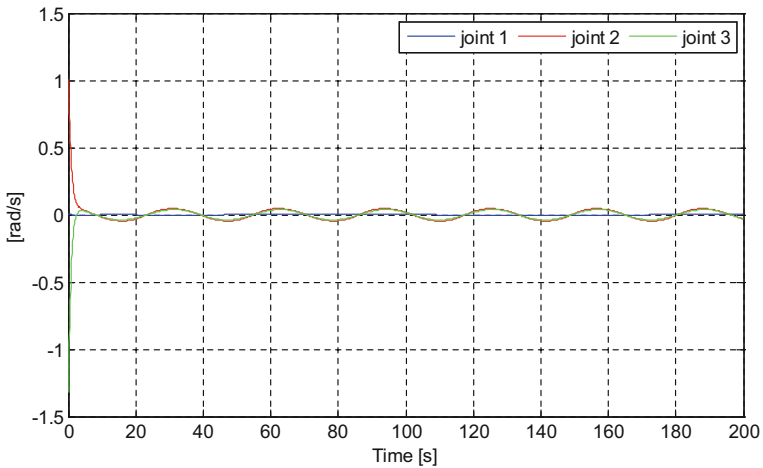


Fig. 12.29 Joint velocity commands to the robotic arm

12.5 Conclusions

This chapter has addressed the problem of moving target tracking for robots with on-board vision system by designing an image-based visual controller in the context of the passivity theory. The entire control system design was based on two cascaded subsystems. Initially, a control law has been proposed by considering only the kinematic model of the robotic system, which means that perfect velocity tracking is considered, proving the asymptotic convergence to zero of the control errors.

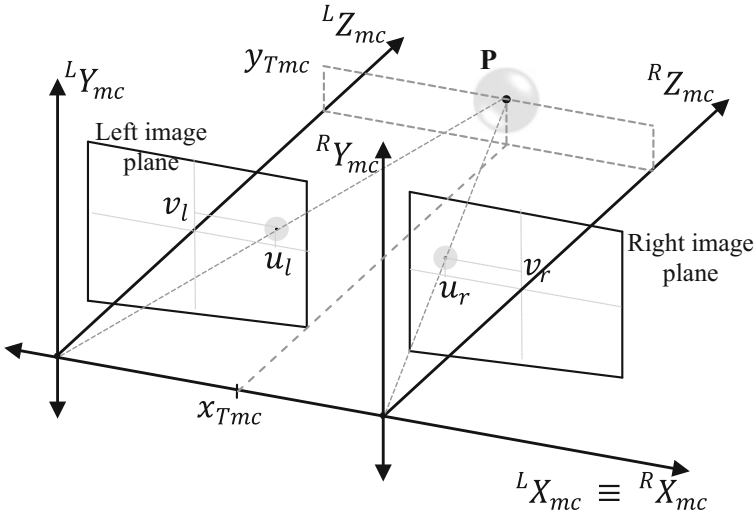


Fig. 12.30 Feature selection for the robotic manipulator

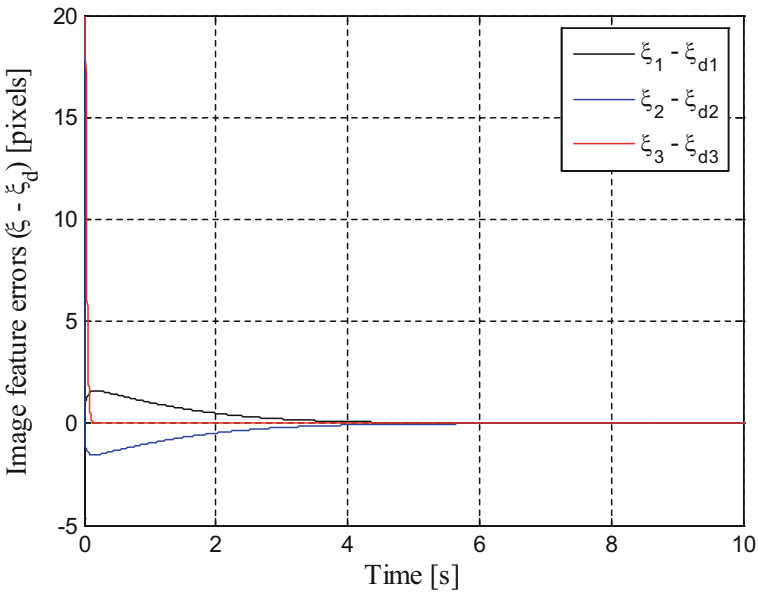


Fig. 12.31 Time evolution of feature errors

Then, it has been considered that dynamic model cannot be disregarded, and the convergence to zero of the control error has been proved again under some conditions to be fulfilled by a dynamic compensation controller. The analysis of the control system was completely performed in the context of the input-output

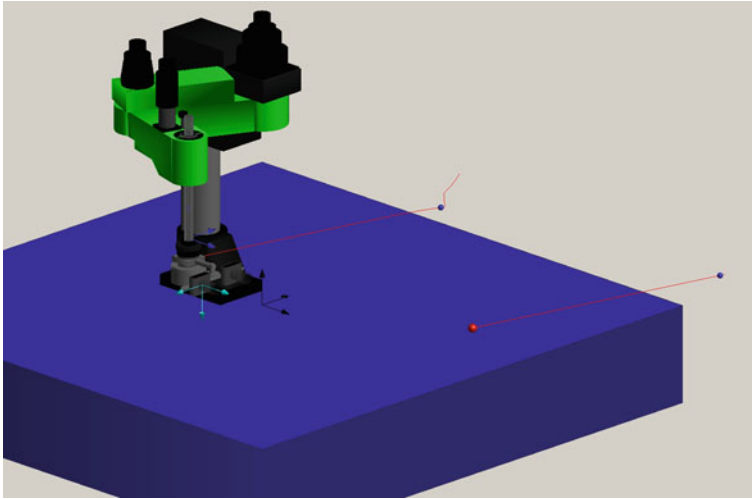


Fig. 12.32 Trajectories described by the robot’s end effector and the target in 3D space

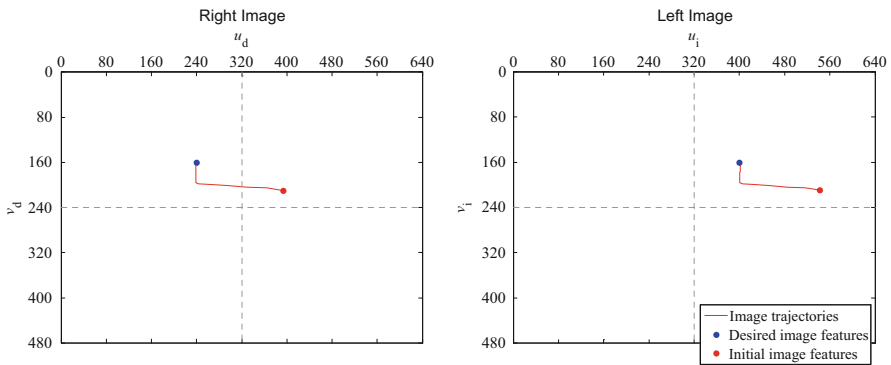


Fig. 12.33 Evolution of image features in the image planes

system theory, specifically by taking advantage of its passivity properties. Finally, since the velocity of the moving object is usually unknown, estimation errors of this velocity have been taken into account on a robustness analysis, concluding that image feature errors are ultimately bounded. Even more, robustness analysis allows to conclude that the control system has L_2 -gain after a finite time by considering the velocity of the target as an external disturbance. Therefore, after a finite time, the proposed control system will be robust to this external disturbance following the L_2 -gain performance criterion. The good performance of the proposed control system is concluded through both real experimental data and numerical simulations with three different classes of robots.

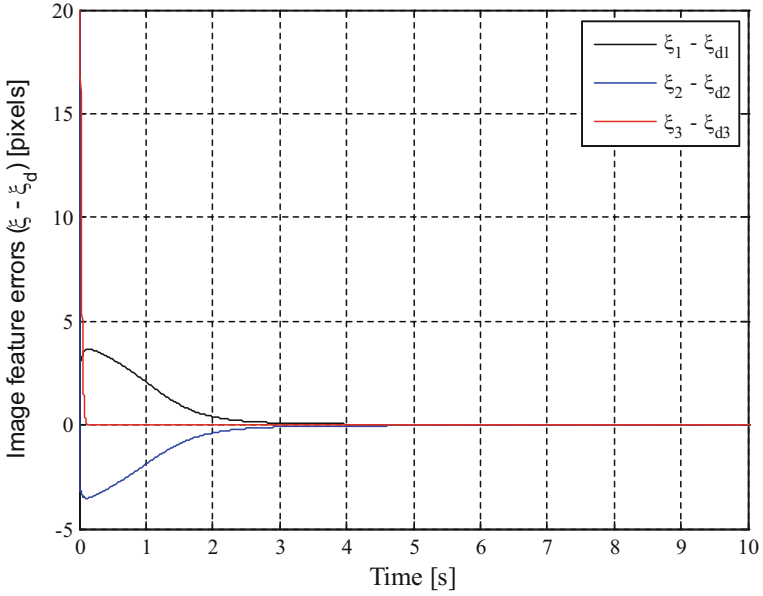


Fig. 12.34 Time evolution of feature errors

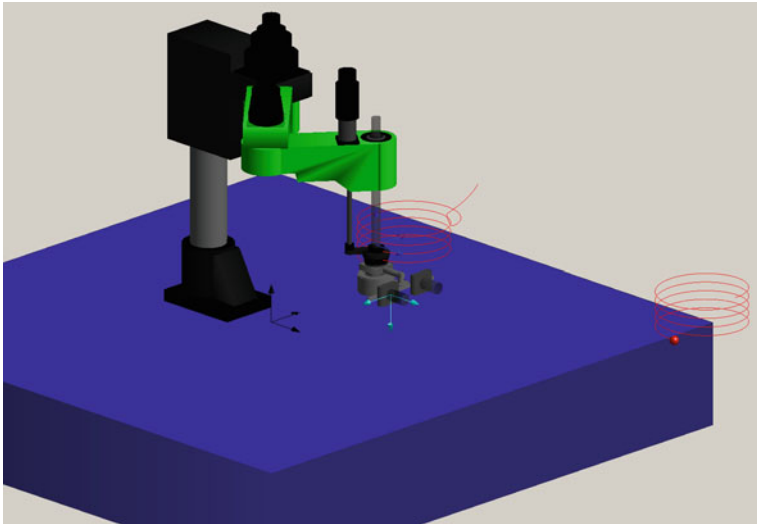


Fig. 12.35 Trajectories described by the robot's end effector and the target in 3D space

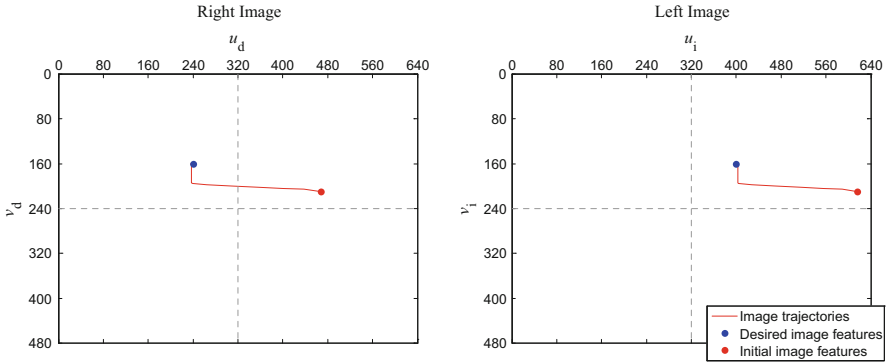
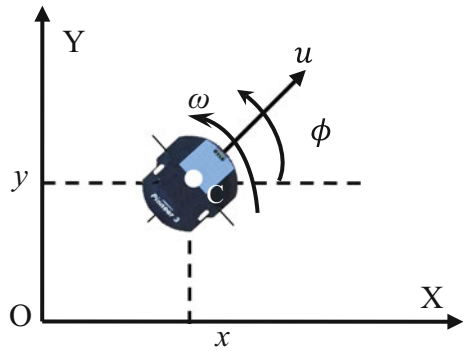


Fig. 12.36 Evolution of image features in the image planes

Fig. 12.37 Geometric description of the mobile robot



A.1 Appendix 1

A.1.1 Mobile Robot Model

This chapter considers a unicycle-like robot, consisting of two self-driven wheels located on the same axle and a castor wheel, as Fig. 12.37 shows. Therefore, if the robot is considered as a concentrated mass placed in the point C at the middle of the wheel’s axle, the kinematic model that represents the pose of the robot in the plane are

$$\begin{aligned}
 \dot{x} &= u \cos \phi \\
 \dot{y} &= u \sin \phi \\
 \dot{\phi} &= \omega
 \end{aligned}
 \tag{12.41}$$

where (x, y) represents the position of the robot, ϕ represents the robot’s orientation, u is the linear velocity of the robot, and ω is the angular velocity of the robot.

This type of robots has a non-holonomic constraint given by

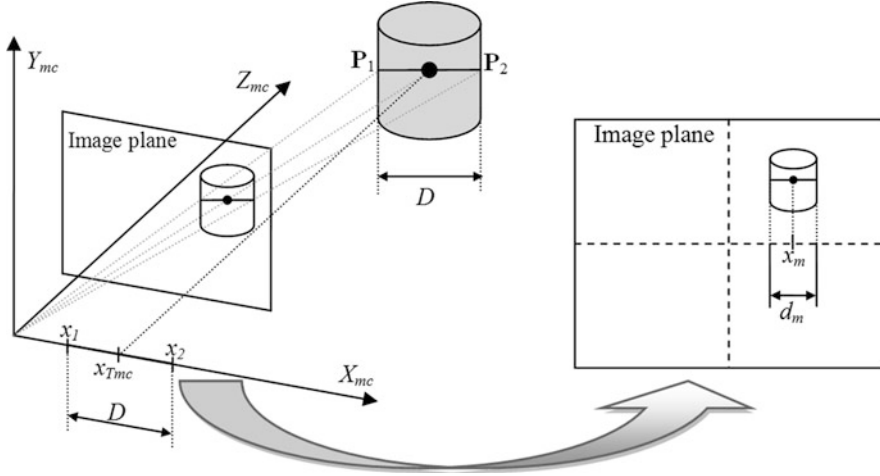


Fig. 12.38 Image features

$$\dot{y} \cos \phi - \dot{x} \sin \phi = 0 \tag{12.42}$$

This restriction states that the robot is not able to move laterally but it is able only to navigate in the perpendicular direction to the wheels' axle.

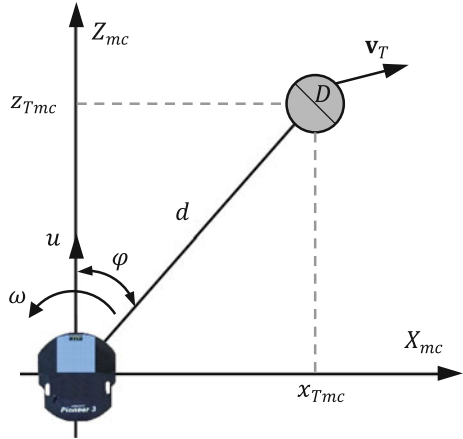
A.1.2 Feature Selection

Without loss of generality for the control laws proposed in this chapter, a cylindrical object is selected, defining the vector of image features as $\xi = [\xi_1 \ \xi_2]^T = [x_m \ d_m]^T$, where x_m is the projection of the cylinder middle point x -coordinate on the image plane; and d_m represents the projection of the actual width D of the cylinder on the image plane [25]. This feature definition is depicted in Fig. 12.38. According to the camera projection model (12.4), the image features are straightforwardly obtained as

$$x_m = f \frac{x_{Tmc}}{z_{Tmc}}; \quad d_m = f \frac{D}{z_{Tmc}} \tag{12.43}$$

Now, the problem consists of obtaining the vision system model. This model has to describe the time variation of the image characteristics ξ as a function of both the robot motion $[u \ \omega]^T$ and the target velocity v_T . For this purpose, the pose of the target on the $X_{mc} - Z_{mc}$ plane with respect to the vision system can be written as a function of the distance d and the angle ϕ , defined as depicted in Fig. 12.39, as follows:

Fig. 12.39 Relative posture between the target and the robot



$$\begin{aligned} x_{Tmc} &= d \sin \varphi \\ z_{Tmc} &= d \cos \varphi \end{aligned} \tag{12.44}$$

Replacing (12.44) with (12.43), the vector of image characteristics is expressed as a function of the relative pose between the target and the camera

$$\xi = \left[f \tan \varphi \quad f \frac{D}{d \cos \varphi} \right]^T \tag{12.45}$$

and the time derivative of (12.45) is

$$\dot{\xi} = \frac{\partial (\xi_1, \xi_2)}{\partial (\varphi, d)} [\dot{\varphi} \quad \dot{d}]^T = \mathbf{J}_1 [\dot{\varphi} \quad \dot{d}]^T \tag{12.46}$$

with

$$\mathbf{J}_1 = \begin{bmatrix} f \sec^2 \varphi & 0 \\ \frac{fD}{d} \sec(\varphi) \tan(\varphi) & -\frac{fD}{d^2} \sec(\varphi) \end{bmatrix} \tag{12.47}$$

The variation in relative position between the robot and the target is due to both the robot motion and the target motion: $[\dot{\varphi} \quad \dot{d}]^T = [\dot{\varphi} \quad \dot{d}]_R^T + [\dot{\varphi} \quad \dot{d}]_T^T$. Now, from the kinematic model of the mobile robot in polar coordinates (see Fig. 12.39) and first considering a static object, the time variation of the relative posture between the target and the robot (time variation of d and φ) is obtained as a function of the linear and angular velocities of the robot $\mu = [u \quad \omega]^T$ as follows:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{d} \end{bmatrix}_R = \begin{bmatrix} \frac{\sin(\varphi)}{d} & 1 \\ -\cos(\varphi) & 0 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = \mathbf{J}_2 \mu \tag{12.48}$$

Now consider a static position for the robot, that is, constant values for x , y , ϕ ; obtaining the time derivative of (12.44), the target velocity $\mathbf{v}_T = [\dot{x}_{Tmc} \ \dot{z}_{Tmc}]^T = \mathbf{A}[\dot{\phi} \ \dot{d}]_T^T$ is expressed as

$$\mathbf{v}_T = \begin{bmatrix} d \cos(\varphi) & \sin(\varphi) \\ -d \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{d} \end{bmatrix}_T \quad (12.49)$$

Then, as \mathbf{A} is invertible, the following expression can be obtained

$$\begin{bmatrix} \dot{\phi} \\ \dot{d} \end{bmatrix}_T = \begin{bmatrix} \frac{1}{d} \cos(\varphi) & -\frac{1}{d} \sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \mathbf{v}_T \quad (12.50)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{d} \end{bmatrix}_T = \mathbf{J}_0 \mathbf{v}_T$$

Finally, by introducing the motions of both the robot (12.48) and the target (12.50) into (12.46), the model of the vision system is obtained

$$\dot{\boldsymbol{\xi}} = \mathbf{J}_1 (\mathbf{J}_2 \boldsymbol{\mu} + \mathbf{J}_0 \mathbf{v}_t) \quad (12.51)$$

Defining

$$\begin{aligned} \mathbf{J} &= \mathbf{J}_1 \mathbf{J}_2 \\ \mathbf{J}_T &= \mathbf{J}_1 \mathbf{J}_0 \end{aligned} \quad (12.52)$$

a compact form for the vision system model is obtained

$$\dot{\boldsymbol{\xi}} = \mathbf{J} \boldsymbol{\mu} + \mathbf{J}_T \mathbf{v}_t \quad (12.53)$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{x_m d_m}{fD} & \frac{f^2 + x_m^2}{f} \\ \frac{d_m^2}{fD} & \frac{x_m d_m}{f} \end{bmatrix} \quad (12.54)$$

$$\mathbf{J}_T = \begin{bmatrix} \frac{d_m}{D} & -\frac{x_m d_m}{fD} \\ 0 & -\frac{d_m^2}{fD} \end{bmatrix} \quad (12.55)$$

B.1 Appendix 2

Formal definitions associated with passivity of operators relating functional spaces, used in this work, follow [44].

Definition 1 L_p signal spaces.

For all $p \in [1, \infty)$, L_p signal spaces are defined as

$$L_p = \left\{ f : \mathbb{R}_+ \rightarrow \mathbb{R} / \int_0^\infty |f(t)|^p dt < \infty \right\}.$$

L_p are the Banach spaces with respect to the norm

$$\|f\|_p = \left(\int_0^\infty |f(t)|^p dt \right)^{\frac{1}{p}}.$$

Definition 2 L_∞ signal spaces.

L_∞ signal spaces are defined as

$$L_\infty = \left\{ f : \mathbb{R}_+ \rightarrow \mathbb{R} / \sup_{t \in \mathbb{R}_+} |f(t)| < \infty \right\}.$$

L_∞ are the Banach spaces with respect to the norm

$$\|f\|_\infty = \sup_{t \in \mathbb{R}_+} |f(t)|.$$

Definition 3 Truncated function.

Let $f : \mathbb{R}_+ \rightarrow \mathbb{R}$. Then, for each $T \in \mathbb{R}_+$, the function $f_T : \mathbb{R}_+ \rightarrow \mathbb{R}$ is defined by

$$f_T(t) = \begin{cases} f(t) & 0 \leq t < T \\ 0 & t \geq T \end{cases}$$

Definition 4 Extended L_p signal spaces.

Extended L_p spaces are defined as

$$L_{pe} = \{ f/f_T \in L_p \quad \forall T < \infty \}.$$

Definition 5 Given $g, h \in L_{2e}$, the inner product and the norm $\|\bullet\|_{2e}$ in the set L_{2e} are defined as

$$\begin{aligned} \langle g, h \rangle_T &= \int_0^T g(t)h(t) dt \quad \forall T \in [0, \infty) \\ \|g\|_{2,T} &= \langle g, g \rangle_T^{\frac{1}{2}} = \left(\int_0^T g(t)g(t) dt \right)^{\frac{1}{2}} \end{aligned}$$

Definition 6 Let $\mathbf{G} : L_{2e} \rightarrow L_{2e}$ be an input-output mapping. Then, \mathbf{G} is passive if there exists some constant β such that

$$\langle \mathbf{G}\mathbf{x}, \mathbf{x} \rangle_T \geq \beta \quad \forall \mathbf{x} \in L_{2e} \quad \forall T \in [0, \infty)$$

Definition 7 Let $\mathbf{G} : L_{2e} \rightarrow L_{2e}$ be an input-output mapping. Then, \mathbf{G} is strictly input passive if there exists some constants $\beta \in \mathbb{R}$ and $\delta > 0$ such that

$$\langle \mathbf{G}\mathbf{x}, \mathbf{x} \rangle_T \geq \beta + \delta \|\mathbf{x}\|_{2,T}^2 \quad \forall \mathbf{x} \in L_{2e}$$

Definition 8 Let $\mathbf{G} : L_{2e} \rightarrow L_{2e}$ be an input-output mapping. Then, \mathbf{G} is strictly output passive if there exists some constants $\beta \in \mathbb{R}$ and $\delta > 0$ such that

$$\langle \mathbf{G}\mathbf{x}, \mathbf{x} \rangle_T \geq \beta + \delta \|\mathbf{G}\mathbf{x}\|_{2,T}^2 \quad \forall \mathbf{x} \in L_{2e}$$

References

1. Basaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., Garcia, X., Tyrsa, V. V., Rivas-Lopez, M., Hernandez-Balbuena, D., Mercorelli, P., Podrygalo, M., Gurko, A., Tabakova, I., & Starostenko, O. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
2. Toibero, J. M., Roberti, F., & Carelli, R. (2009). Stable contour-following control of wheeled mobile robots. *Robotica*, 27(1), 1–12.
3. Toibero, J. M., Roberti, F., Carelli, R., & Fiorini, P. (2011). Switching control approach for stable navigation of mobile robots in unknown environments. *Robotics and Computer Integrated Manufacturing*, 27(3), 558–568.
4. Rodríguez-Quiñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-Lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
5. Toibero, J. M., Soria, C., Roberti, F., Carelli, R., & Fiorini, P. (2009). Switching visual servoing approach for stable corridor navigation. In *Proceedings of the International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009*.
6. Traslosheros, A., Sebastián, J. M., Torrijos, J., Carelli, R., & Roberti, F. (2014). Using a 3DOF parallel robot and a spherical bat to hit a Ping-Pong ball. *International Journal of Advanced Robotic Systems*, 11(5). <https://doi.org/10.5772/58526>.
7. Weiss, L. E., Sanderson, A., & Neuman, P. (1987). Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(9), 404–417.
8. Carelli, R., De la Cruz, C., & Roberti, F. (2006). Centralized formation control of non-holonomic mobile robots. *Latin American Applied Research*, 36(2), 63–69.
9. Gong, Z., Tao, B., Yang, H., Yin, Z., & Ding, H. (2018). An uncalibrated visual servo method based on projective homography. *IEEE Transactions on Automation Science and Engineering*, 15(2), 806–817.
10. López-Nicolás, G., Guerrero, J. J., & Sagüés, C. (2010). Visual control of vehicles using two-view geometry. *Mechatronics*, 20(2), 315–325.
11. Carelli, R., Kelly, R., Nasisi, O. H., Soria, C., & Mut, V. (2006). Control based on perspective lines of a nonholonomic mobile robot with camera-on-board. *International Journal of Control*, 79, 362–371.

12. Wang, H., Guo, D., Xu, H., Chen, W., Liu, T., & Leang, K. K. (2017). Eye-in-hand tracking control of a free-floating space manipulator. *IEEE Transactions on Aerospace and Electronic Systems*, 53(4), 1855–1865.
13. Carelli, R., Santos-Victor, J., Roberti, F., & Tosetti, S. (2006). Direct visual tracking control of remote cellular robots. *Robotics and Autonomous Systems*, 54(10), 805–814.
14. Taryudi, & Wang, M. S. (2017). 3D object pose estimation using stereo vision for object manipulation system. In *Proceedings of the IEEE International Conference on Applied System Innovation, Sapporo, Japan*, 13–17 May 2017.
15. López-Nicolás, G., Guerrero, J. J., & Sagüés, C. (2010). Visual control through the trifocal tensor for nonholonomic robots. *Robotics and Autonomous Systems*, 58(2), 216–226.
16. Andaluz, V., Carelli, R., Salinas, L., Toibero, J. M., & Roberti, F. (2012). Visual control with adaptive dynamical compensation for 3D target tracking by mobile manipulators. *Mechatronics*, 22(4), 491–502.
17. Roberti, F., Toibero, J. M., Soria, C., Vassallo, R., & Carelli, R. (2009). Hybrid collaborative stereo vision system for mobile robots formation. *International Journal of Advanced Robotic Systems*, 6(4), 257–266.
18. Zhang, K., Chen, J., Li, Y., & Gao, Y. (2018). Unified visual servoing tracking and regulation of wheeled mobile robots with an uncalibrated camera. *IEEE/ASME Transactions on Mechatronics*, 23(4), 1728–1739.
19. Fujita, M., Kawai, H., & Spong, M. W. (2007). Passivity-based dynamic visual feedback control for three dimensional target tracking: Stability and L2-gain performance analysis. *IEEE Transactions on Control Systems Technology*, 15(1), 40–52.
20. Kawai, H., Toshiyuki, M., & Fujita, M. (2006). Image-based dynamic visual feedback control via passivity approach. In *Proceedings of the IEEE International Conference on Control Applications, Munich, Germany*, 4–6 October 2006.
21. Murao, T., Kawai, H., & Fujita, M. (2005). Passivity-based dynamic visual feedback control with a movable camera. In *Proceedings of the 44th IEEE International Conference on Decision and Control, Sevilla, Spain*, 12–15 December 2005.
22. Soria, C., Roberti, F., Carelli, R., & Sebastián, J. M. (2008). Control Servo-visual de un robot manipulador planar basado en pasividad. *Revista Iberoamericana de Automática e Informática Industrial*, 5(4), 54–61.
23. Martins, F., Sarcinelli, M., Freire Bastos, T., & Carelli, R. (2008). Dynamic modeling and trajectory tracking control for unicycle-like mobile robots. In *Proceedings of the 3rd International Symposium on Multibody Systems and Mechatronics, San Juan, Argentina*, 8–12 April 2008.
24. Andaluz, V., Roberti, F., Salinas, L., Toibero, J. M., & Carelli, R. (2015). Passivity-based visual feedback control with dynamic compensation of mobile manipulators: Stability and L2-gain performance analysis. *Robotics and Autonomous Systems*, 66, 64–74.
25. Morales, B., Roberti, F., Toibero, J. M., & Carelli, R. (2012). Passivity based visual servoing of mobile robots with dynamics compensation. *Mechatronics*, 22(4), 481–490.
26. El-Hawwary, M. I., & Maggiore, M. (2008). Global path following for the unicycle and other results. In *Proceedings of the American Control Conference, Seattle, Washington*, 11–13 June 2008.
27. Lee, D. (2007). Passivity-based switching control for stabilization of wheeled mobile robots. In *Proceedings of the Robotics: Science and Systems, Atlanta, Georgia*, 27–30 June 2007.
28. Arcak, M. (2007). Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8), 1380–1390.
29. Igarashi, Y., Hatanaka, T., Fujita, M., & Spong, M. W. (2007). Passivity-based 3D attitude coordination: Convergence and connectivity. In *Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA*, 10–11 December 2007.
30. Ihle, I., Arcak, M., & Fossen, T. (2007). Passivity-based designs for synchronized path-following. *Automatica*, 43(9), 1508–1518.
31. Spong, M. W., Holm, J. K., & Lee, D. J. (2007). Passivity-based control of biped locomotion. *IEEE Robotics & Automation Magazine*, 14(2), 30–40.

32. Fujita, M., Hatanaka, T., Kobayashi, N., Ibuki, T., & Spong, M. (2009). Visual motion observer-based pose synchronization: A passivity approach. In *Proceedings of the IEEE International Conference on Decision and Control, Shanghai, China*, 16–18 December 2009.
33. Kawai, H., Murao, T., & Fujita, M. (2011). Passivity-based visual motion observer with panoramic camera for pose control. *Journal of Intelligent & Robotic Systems*, 64(3–4), 561–583.
34. Hu, Y. M., & Guo, B. H. (2004). Modeling and motion planning of a three-link wheeled mobile manipulator. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision, Kunming, China*, 6–9 December 2004.
35. Andaluz, V., Roberti, F., Toibero, J. M., & Carelli, R. (2012). Adaptive unified motion control of mobile manipulators. *Control Engineering Practice*, 20(12), 1337–1352.
36. Hutchinson, S., Hager, G., & Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5), 651–670.
37. Hill, D., & Moylan, P. (1976). Stability results for nonlinear feedback systems. *Automatica*, 13, 373–382.
38. Bynes, C. I., Isidori, A., & Willems, J. C. (1991). Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems. *IEEE Transactions on Automatic Control*, 36(11), 1228–1240.
39. Ortega, R., Loria, A., Nelly, R., & Praly, L. (1995). On passivity based output feedback global stabilization of Euler-Lagrange systems. *International Journal of Robust and Nonlinear Control*, 5, 313–324.
40. Vidyasagar, M. (1979). New passivity-type criteria for large-scale interconnected systems. *IEEE Transactions on Automat. Control*, 24, 575–579.
41. Vidyasagar, M. (1978). *Nonlinear systems analysis*. Englewood Cliffs, NJ: Prentice Hall International Editions.
42. Bayle, B., & Fourquet, J. Y. (2001). Manipulability analysis for mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea*, May 2001.
43. Kalata, P. (1994). The tracking index: A generalized parameter for α - β and α - β - γ target trackers. *IEEE Transactions on Aerospace and Electronic Systems*, 20(2), 174–182.
44. Van der Schaft, A. (2000). *L2 gain and passivity techniques in nonlinear control*. London: Springer.

Chapter 13

Data Exchange and Task of Navigation for Robotic Group



Mikhail Ivanov, Oleg Sergiyenko, Vera Tyrsa, Lars Lindner, Miguel Reyes-García, Julio Cesar Rodríguez-Quiñonez, Wendy Flores-Fuentes, Jesús Elías Miranda-Vega, Moisés Rivas-López, and Daniel Hernández-Balbuena

Acronyms

3D	Three-Dimensional
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FOV	Field of View
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
LED	Light-Emitting Diode
LIDAR	Light Identification Detection and Ranging
MEMS	Microelectromechanical Systems
MIT	Massachusetts Institute of Technology
MRS	Multi Robot Systems
PL	Positioning Laser
SA	Scanning Aperture
SLAM	Simultaneous Localization and Mapping
STP	Spanning Tree Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
ToF	Time-Of-Flight
TVS	Technical Vision System

M. Ivanov · O. Sergiyenko (✉) · L. Lindner · M. Reyes-García · J. E. Miranda-Vega
M. Rivas-Lopez

Universidad Autonoma de Baja California (UABC), Instituto de Ingeniería, Mexicali, BC, Mexico
e-mail: ivanovm@uabc.edu.mx; srgnk@uabc.edu.mx; lindner.lars@uabc.edu.mx;
reyes.miguel73@uabc.edu.mx; elias.miranda@uabc.edu.mx; mrivas@uabc.edu.mx

V. Tyrsa · J. C. Rodríguez-Quiñonez · W. Flores-Fuentes · D. Hernández-Balbuena
Universidad Autonoma de Baja California (UABC), Facultad de Ingeniería, Mexicali, BC, Mexico
e-mail: vtyrsa@uabc.edu.mx; julio.rodriguez81@uabc.edu.mx; flores.wendy@uabc.edu.mx;
dhernan@uabc.edu.mx

UABC Universidad Autonoma de Baja California
UAV Unmanned Aerial Vehicle

13.1 Introduction

Nowadays, the use of swarm intelligence systems can be found in research papers of civil-related tasks, such as autonomous cars, UAVs, etc. Driven by scientific research, swarm intelligence systems are prevalent and specialized for multipurpose tasks that require a group of robots to cover different types of unknown environments (cluttered or rugged terrains, indoor premises, etc.). In articles such robotic groups are referred to as swarm robotics [1–3].

Swarm robotics is a promising technology that can be deeply involved in daily human life. As an example, smart autonomous vehicles can be found by the use of Google [4], Tesla, Uber, etc. Right now they are not so affordable, however in future they will be a huge part of social life from smart cities and campuses with autonomous personal mobility vehicles [5] to simple use in smart buildings as janitors.

In these matters exist two principle tasks—navigation and communication. First is used for obstacle avoidance and moving to the target location, and second is to give the robot a tool to “talk.” Communication helps a swarm to achieve more complete and structured information about the surroundings and improve their navigation as individuals.

Current chapter will consider a solution for such problem and reviewed the influence of data exchange on the path planning in terms of unknown surroundings.

13.2 Swarm Robotics

Development of distributed artificial intelligence [6] (republished [7]) is a subject of many complex researches related to the multi-agent systems [8]. The behavior models used in these systems take their origin in adoption of social animal group activities. In bacteria colonies, fish schooling, animal herding, ants, etc. (Fig. 13.1) individuals have primitive abilities, but while in-group they become a complex organization with improved surrounding interactions, signaling communication, and data transmission [9]. Such natural swarms are based on simple set of rules used by each individual. As the result with common efforts, the homogeneous groups can complete complex tasks. Transferring of this behavior created the principles of swarm intelligence [10, 11].

13.2.1 Nature Swarm Adaption

Primates usually have complex collaboration inside the group, they can have different types of social interactions [12], recognize their relatives [13] and some of the species can use some of the human language aspects.

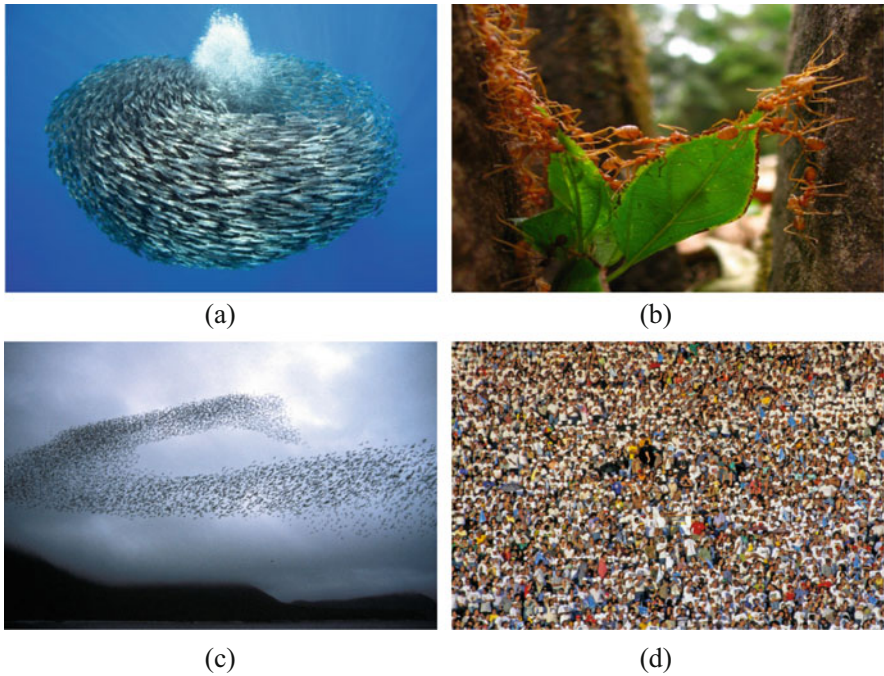


Fig. 13.1 Natural swarms. (a) Fish schooling. (b) Ant colonies. (c) Bird crowds. (d) Human beings

Bacteria and their colonies are usually functioning as biofilms. They have an ability to cell communication [14] and use benefits of task distribution, collective defense, etc. The bacterial colony has higher resistance to antibacterial agents than individuals of the same type of bacteria [15].

Bird Crowds during the migration can locate their destination point according to inner sensing, landmarks, etc. [16].

Ant and Bee Colonies Communication in such type of colonies is based mostly on pheromones [17]. Ants are able to path planning by leaving a pheromone trail, where in case of optimal route, more and more ants are using it [18]. According to suggestion in [19], ants implement role distribution based on their previous performance during foraging.

Locusts while increasing amount of insects in-group convert their type on chaotic movement to align with ability to transfer rapidly directional information [20].

Fish Schools Each fish by analyzing the neighbors' movement can avoid collision while swimming in phalanxes [21]. Fish schools are better in foraging [22] and predator evasion [23].

Human Beings Dyer et al. in [24] have shown that in the group of humans can occur leadership without any verbal communication or other obvious types of signals. Such behaviorism shows the hierarchical structure with role distribution. In case of growing down of the population, collaboration within the group becomes more complex and each role becomes more important. A swarm of individuals in this case can solve complex tasks easily while individual cannot.

According to the brief review of natural swarms can be found a list of tasks that robotic swarm can be capable for. As an example: Aggregation, Flocking, Foraging, Object clustering and sorting, Navigation, Path formation, etc. Next section will describe them more detailed.

13.2.2 Tasks of Swarm Robotics

Swarm robotics has a variety of tasks that can be done. The next list presents some examples according to their recursivity in descending order.

Navigation is a task where robot needs to find location of an object using his sensing capabilities and the help of other robots. In this category, studies consider where object to be reached by individual robot but not a group.

Foraging is another scenario studied in swarm robotics where group needs to find food source. It takes origin in the behavior of ants in colonies. Evolving robots [25] is a perfect example of this behavior implementation.

Multi-Foraging is more complex type of foraging task. Robotic group during the task need to find and collect different types of objects and after bring them to a specific place for this objects type. These task has an application in warehouses, rescue missions, hazardous terrain cleanups, etc.

Odor Source Localization is aimed to solve the problem of odor source search. In [26], was described a project where robots are using binary odor sensor.

Collective Decision-Making is a study of many researches like [27] (authors proposed decision-making based on majority rule model), [28] (charges of preferred labors), [29] (site selection). Collective decision-making applied to swarm robotics during the flocking, path formation, clustering, etc.

Object Clustering and Sorting are similar to foraging task, but in this case the goal is to find objects in environment and to place objects near to each other.

Object Assembly is a task related to construction problems. The task has a relationship with clustering, but in this case it is focused on relationship and physical connections between objects (robots have to create an object of predefined shape). In [30] describes an example of wall constructing by robots, in [31] robots were trying to figure out if the block they use can be attached or not with help of local

communication, in [32] were modeled a process of building using blocks with attached LEDs.

Collaborative Manipulation refers to a task where robotic swarms have to interact with objects. For achieving this task and avoiding centralized coordination (control), robots can use simple behavior rules. Ant colonies can be used as an example.

Self-Assembly is a task of local interaction within the group, where robots need to establish physical connection to each other. The examples can be found in [33] (swarm-bots project), authors of [34] used s-bots with ability to decide who will use its grip to attach to other s-bot. Self-assembly also takes its place during the hole avoidance [35] and navigation in hazardous terrains [36].

Human-Swarm Interaction task is devoted to increase performance of robotic swarms. The task can be described in the next way: if human operation have an information that can help robotic swarm to achieve their goal he can share it, but it cannot be interpreted as a direct controlling. In [37] authors proposed method where by starting direct control over one robot can have an influence on a group by changing robot behavior. In [38] were used a similar method but with leader system implementation. Kolling et al. [39] proposed methods of robots behavior switch and in second method involved manipulation with environment to force robot to change its behavior.

Deployment is a scenario where group of robots must autonomously enter to the environment. Task is useful for the unknown terrain mapping.

Path Formation is a process that refers to a collective movement formation from point to point while minimizing the time to reach the destination. This task usually occurs during the foraging and chain formations.

Coordinated Motion is used during the modular robotic structure to achieve coordinated movement to common direction.

Flocking is a task of swarm to “stay together.” It is based on local interaction within the group. Robots’ sensing systems (vision, laser range finding, sonar, infrared or tactile sensors) and communication abilities use to keep group as a compact formation. Such behaviorism adopted from birds, fish schools, etc.

Morphogenesis exists as an extension of self-assembly. In case of morphogenesis, robots are tasked to create a specific shape. For example [40], s-bots attached to structure used LEDs for communication to tell how other should be attached.

Aggregation is a task of grouping the individual robots in a dedicated place. The behavior model is based according to animal species observation.

Task Allocation is a process of labor division within the group. This ability helps to increase efficiency of work done by a swarm.

Group Size Estimation is a “sub-task” for different applications of swarm robotic, used to coordinate movement, self-assembly, morphogenesis, etc. Authors in [41]

describe method using the propagation of information through the group in case of impossible direct communication within the group.

More detailed review of the mentioned tasks with their solutions can be found in [42].

13.2.3 *Swarm Robotics Projects*

During the last decade, the growth of small and mobile computing devices capabilities increased the interest in swarm robotics research topic. Yet still many of the swarm robotics projects are on the development or even modeling stage [43]. Earlier in the 1980s existed other attempts to create robotic groups like SWARMS [44] and ACTRESS [45]. This section will review some of the existing project and their design.

Pheromone Robotics Project (Fig. 13.2a) was initiated in 2000 [46]. The idea of the project is to use a large number of small robots for different tasks by achieving a swarm behaviorism [47]. In robots was implemented the pheromone idea, by using beacons and sensors mounted on robots.

iRobot Swarm Project (Fig. 13.2b) was made by MIT. Swarm includes more than 100 cooperating robots [48, 49]. The main idea is to create the solution for graceful degradation of the swarm.

E-Puck Education Robots is a group of small robots (Fig. 13.2c) that were made for educational purpose like programming, humane-machine interaction, signal and image processing. They are cheap, have a simple structure, and have possibilities to use extensions [50].

Kobot Project (Fig. 13.2d) consists of mobile robotic platforms equipped with IR range finder system for obstacle detection [51].

Kilobot Project (Fig. 13.2e) is aimed to create the collective behaviorism with hundreds or more individuals in swarm [52]. Robots are easy to assemble and have capabilities for some simple operations like charging, moving, updating programs, etc.

I-Swarm (Fig. 13.2f) came from University of Karlsruhe in Germany [53]. Swarm consists of micro-robots with the abilities of “collective thinking” and to recognize its kin.

Multi Robot Systems (Fig. 13.2g) [54] initially were developed in University of Alberta in Edmonton, USA, studies robots collective behavior. This institution has several robots systems (Multi Robot Systems (MRS)) in development. The project is devoted to problems of collective decision-making.

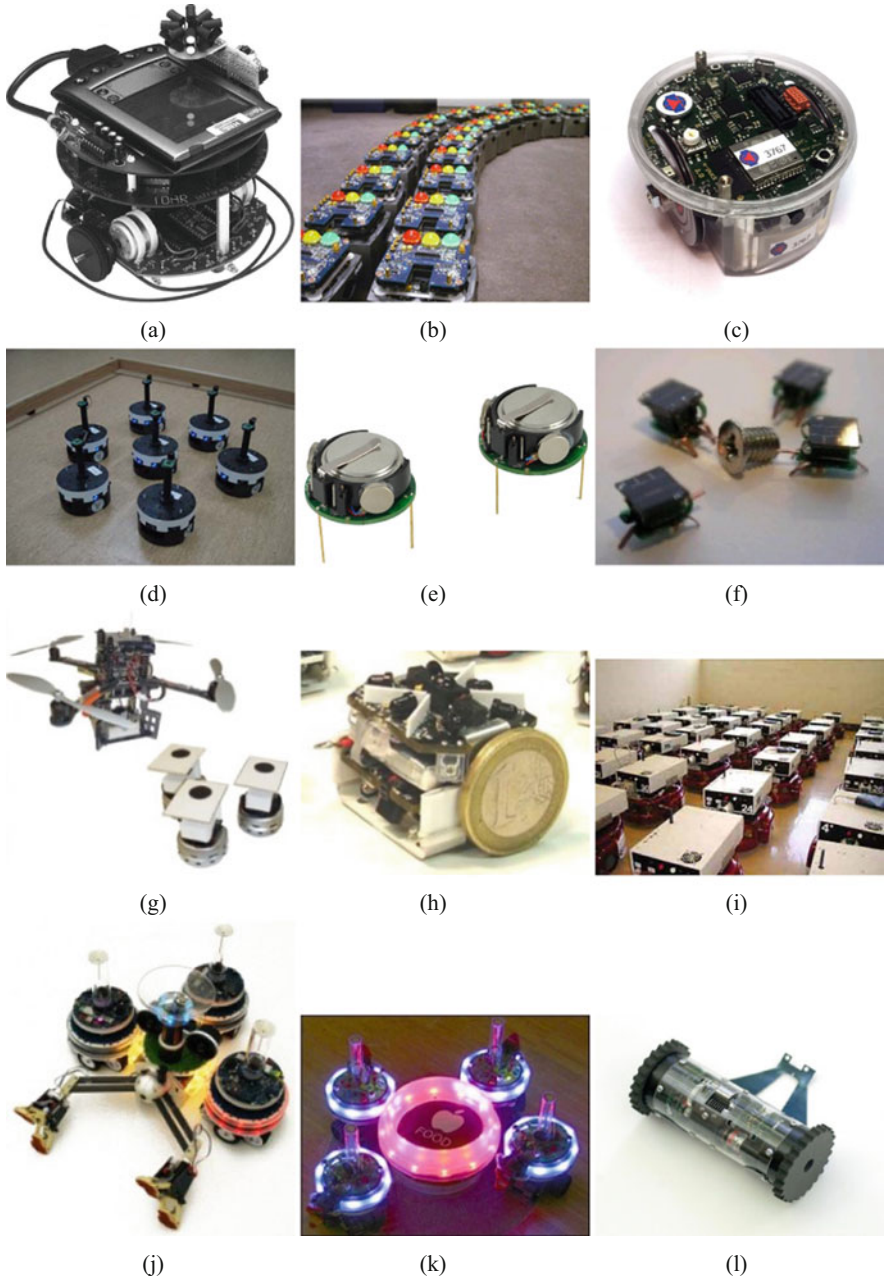


Fig. 13.2 Swarm robotics projects. (a) Pheromone robot. (b) iRobot swarm project. (c) E-puck education robot. (d) Kobot project. (e) Kilobot project. (f) I-Swarm project. (g) Multi robot systems. (h) Project SwarmBot. (i) Project Centibots. (j) Project Swarmanoid. (k) Evolving robots. (l) Robot scout

Project SwarmBot (Fig. 13.2h) was made by iRobot company [55]. Swarm uses small robots that can work together to carry out certain actions. It is expected that the SwarmBot robots can join a group of up to 10,000 and perform tasks such as search for mines, research of unknown territory (including on other planets), detection of harmful substances, etc.

Project Centibots (Fig. 13.2i) uses small robots that can work in-group and autonomously [56], they are not using a centralized management system. Their goal is to map the enclosed space and perform some tasks. Robots are using the roles distribution system based on interaction and depending on the circumstances.

Project Swarmanoid (Fig. 13.2j) studies the behavior of inhomogeneous groups of robots [57]. The considered task was in which a team of wheeled robots, a flying robotic spy, and the handling robot jointly found the object (book) and manipulated it.

Evolving Robots (Fig. 13.2k) were created in Swiss Laboratory of Intelligent Systems (Polytechnic School, Lausanne) during the “evolution” of robots studies [25]. Robots were evolving gene that determines behavior. A group of ten robots competed for food. The challenge was to find robots “food source,” which is a luminous ring on one end of the arena. Robots can “communicate” with each other by light signals. The evolution of robots in experiments sometimes leads that even the robots were taught to deceive opponents, letting the “wrong” light, being near the trough.

Robots Scouts (Fig. 13.2l) were designed for intelligence [58]. Project was developed in the distribution centers of robotics, University of Minnesota, USA, the robot is a very high quality from a technical point of view device. The robot can work in a team. Its design allows you to “shoot” using a device resembling automatic grenade launcher. The robot is designed to help the police and rescue services in carrying out dangerous operations. There is a central control unit, which receives information obtained by robots, and which controls the robot, the basic mode of operation is to control the robot operator.

According to the mentioned projects, [9] and [59] the next advantages of swarm robotics can be described:

- Parallel processing—swarm can perform various tasks simultaneously (each of swarm can perform its scheduled task), this would save time for achieving a common goal.
- Scalable group—including a new individual to a swarm can be performed without any modification of the software or hardware.
- Tasks enlargements—swarm robotics systems can solve the tasks that are impossible for individuals.
- Fault tolerance (graceful degradation)—swarm can continue performing its tasks even what the part of the swarm unable to work. Useful during the tasks in a hazardous environment.

- Distributed sensing and action—distributed on the terrain swarm robotics system can be used as a sensing network for data accumulation and action performing system.

However, can be allocated a number of specific problems that need to be resolved. Among them are:

- Unpredictable environmental dynamics;
- Imperfect and inconsistent knowledge of the environment;
- Variety of options to achieve the goal, the team structures, roles, etc.;
- A complex behavior pattern of teamwork;
- Problems related to the territorial distribution of swarm and its localization;
- Communication problems or data exchange (network architecture, protocols, etc.);
- Data loss and storage redundancy.

Next sections will cover possible solutions for each of the problem and review their implementation according to the machine vision selection, methods of data transferring (communication), and navigation.

13.3 Robotics Vision Systems

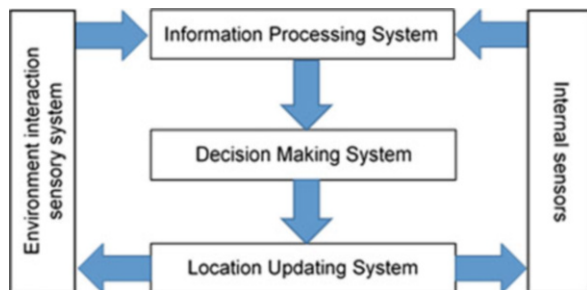
Each mathematical/computing unit in the conventional algorithm of autonomous robot (Fig. 13.3) during its work is basing on data obtained from its sensors.

For the task of navigation are generally used sonars as a basic solution, laser rangefinders as more accurate alternative for sonars, LIDARs (can return the detailed surroundings, but depends on its form factor), camera-based robots [60, 61] and drones [62], or more expensive equipment like ToF (Time-of-Flight) cameras [63]. For some tasks, it is possible to use only inertial navigation system [64].

Plenty of the researches and solutions in the field of obstacle detection and navigation, as mentioned before, are based on cameras and laser systems. As an advances will be revived some works.

In [65] authors presenting a lightweight, inexpensive insect-based stereo-vision system. They used two cameras placed very similar to honey bee eyes and received

Fig. 13.3 Autonomous robotic system



a field-of-view around 280° by 150° . In [66] authors are using camera vision for real-time obstacle avoidance with biped robots. Article [67] describes the obstacle avoidance for the pocket drone based on data from stereo camera. In [68] solution uses Arduino with pixy camera by wheeled robot for the task line tracking and obstacle avoidance. Solution mentioned in [69] can be applied for autonomous cars and shows a method of vehicle detection system design based on stereo vision sensors.

Article [70] describes a benefits of use of long-wavelength infrared stereo vision and 3D-LIDAR combination in case of fire environments. Another publication [71] presenting a MEMS-based LIDAR systems for use in autonomous vehicle. In [72] were proposed method of real-time LIDAR odometry and mapping and its application on drones and cars.

Also such systems can be used for similar tasks in other areas. Authors of [73] used industrial robotic hand with mounted 3D camera-based vision system for object scanning, similar to them in [74] are using industrial robots with camera to track motions of the second robot, in [75] position of the robot is controlled by tracking marker with LEDs and in [76] were proposed human posture tracking and classification using cameras stereo vision and 3D model matching. Also can be mentioned [77] stereo vision-based automation for a bin-picking solution

As can be seen, laser-based systems are more suitable for automobile navigation purpose. It is explained by their surrounding representation principles that help to avoid long post processing like in cameras.

Next section will provide a detailed description of the real-time vision system used for the current solution.

13.3.1 Technical Vision System

13.3.1.1 Historical Background

All the solutions are based on the novel author's technical vision system (TVS) that uses a dynamic triangulation principle [78]. In [79] proposed method for improving resolution of 3D TVS and its implementation for surface recognition. This approach to obstacle recognition was implemented for the single robot navigation in work [80]. The concepts of data transferring within robotic group firstly presented in [81] Further advances of author's 3D TVS found its use and development in works [82] and [83]. Here system received its internal changings and appliance as a machine vision system for UAV. Despite all of researches and results presented in these articles still exist a common problem: all of them are dedicated to one problem at a time. That is why current paper is aimed to present a joint solution taking into consideration problems of machine vision, path planning, and data transferring using the mentioned earlier 3D TVS.

13.3.1.2 Structure and Working Principles

According to current task, robotic group move in low light conditions with a large number of obstacles, the aforementioned vision systems may not always give the correct results during post processing. Therefore, when working in such difficult conditions, authors TVS [78] can satisfy with its accuracy and data representation. 3D TVS (Fig. 13.4a) is able to work in complete darkness and can obtain the real 3D coordinates of any point highlighted by laser ray on real, not imaginable, surfaces. The theory is based on a dynamic triangulation method. The main components of the TVS are the positioning laser (PL) and the scanning aperture (SA) (Fig. 13.4b).

System works in the following way: the laser emits its beam toward a fixed 45° mirror than makes orthogonal redirection of the beam into a rotating 45° mirror driven by a stepper motor. For the guaranteed positioning of the laser direction, PL is driven by a stepper motor. SA receive the reflected laser rays, this indicates that system had detected an obstacle. However, stepper motor has one weak point: on the long distances of scanning dead-zones between two adjacent points of scanning are provoked. Solution of the problem can be found in [84] and [85].

Dynamic triangulation [86, 87] consists of detection of laser instantly highlighted point coordinates based on two detected angles B_{ij} and C_{ij} and fixed distance between projector and receptor. Here ij means the number of horizontal and vertical scanning steps consequently. In such triangle (Fig. 13.4b), if three parameters are known, it makes possible to calculate all others. Angle B_{ij} is calculated as simple ratio of two counters codes: number of clock pulses between two home pulses and in interval “home pulse – spot pulse” (Fig. 13.4c) (Eq. (13.1)).

$$B_{ij} = \frac{2\pi N_A}{N_{2\pi}} \tag{13.1}$$

where N_A is the number of reference pulses when laser rays are detected by the stop sensor and $N_{2\pi}$ is the number of reference pulses when the 45° mirror completes a 360° turn detected by the zero sensor. To calculate x, y, and z coordinates, the next equations are used (Eqs. (13.2)–(13.5)):

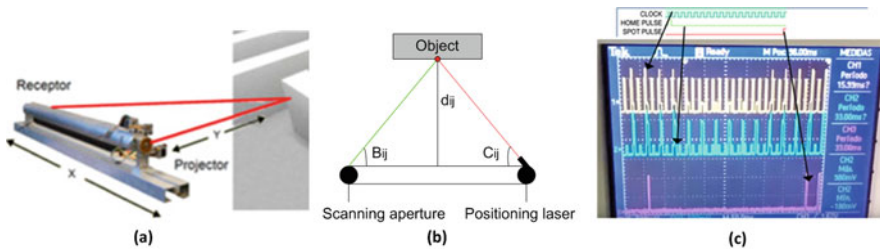


Fig. 13.4 (a) Technical vision system. (b) Dynamic triangulation method. (c) Principle of codes N forming in laser scanning TVS

$$x_{ij} = a \frac{\sin B_{ij} \cdot \sin C_{ij} \cdot \cos \sum_{j=1}^j \beta_j}{\sin[180^\circ - (B_{ij} + C_{ij})]} \quad (13.2)$$

$$y_{ij} = a \left(\frac{1}{2} - \frac{\sin B_{ij} \cdot \sin C_{ij}}{\sin[180^\circ - (B_{ij} + C_{ij})]} \right) \text{ at } B_{ij} \leq 90^\circ \quad (13.3)$$

$$y_{ij} = -a \left(\frac{1}{2} + \frac{\sin B_{ij} \cdot \sin C_{ij}}{\sin[180^\circ - (B_{ij} + C_{ij})]} \right) \text{ at } B_{ij} \geq 90^\circ \quad (13.4)$$

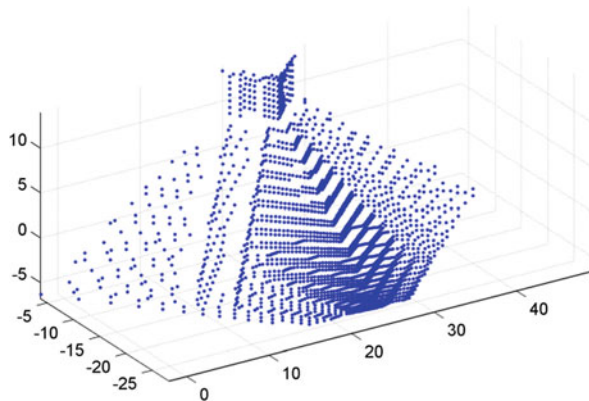
$$z_{ij} = a \frac{\sin B_{ij} \cdot \sin C_{ij} \cdot \cos \sum_{j=1}^j \beta_j}{\sin[180^\circ - (B_{ij} + C_{ij})]} \quad (13.5)$$

13.3.1.3 Data Reduction

According to the specifics of proposed TVS, it returns the scanned surface as a point cloud (Fig. 13.5). On short distances, it gives high detailed object, while on a distance it loses its resolution depending to the opening angle of each step of scanning.

In the memory of a robot, each point obtained by TVS is represented with three variables— x , y , z of Cartesian coordinate system. Each of them stored using double data type that is equal to 64 bit per number, so to store one point of environment 192 bit of memory is used. During the movement and mapping of environment, the data that need to be processed can reach gigabytes. For the navigation, robots need a minimum amount of points to describe an object. That is why it is necessary to implement the method of low density scanning for dead reckoning and to use high density scanning on demand.

Fig. 13.5 Examples of scanned object; Scan of “Mayan pyramid,” on sides the point cloud density is less than a part of a cloud with stairs has. To store this object 75.14 kB of memory is used



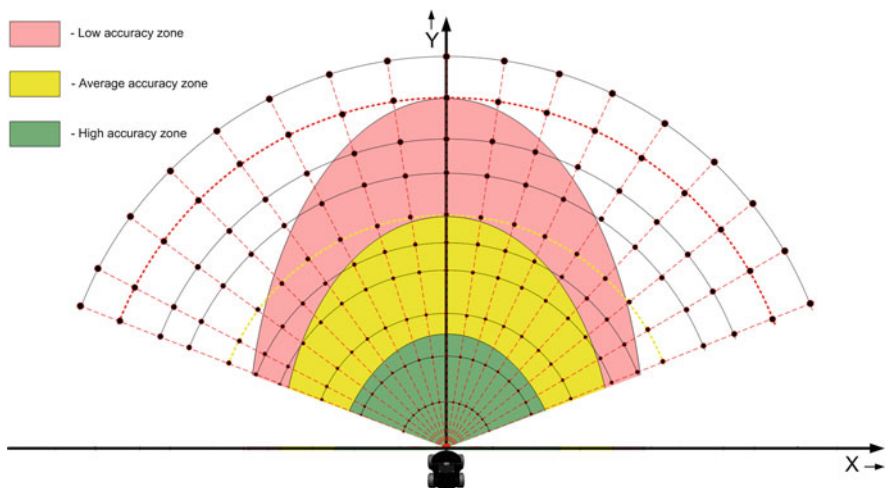


Fig. 13.6 Field of view fragmentation

In previous work [88] were described the method of real-time data reduction during robots’ movement. According to the allocated accuracy zones (Fig. 13.6) were determined an opening angle [89] equivalent to store points on detected obstacles.

As a starting point will be used an arc of one meter according to possible 160° FOV of TVS. Using the research data and type of robots described in work [80], the cloud point density (ρ) of image is 11 points per meter.

$$\rho = \frac{\lambda}{\beta p}, \tag{13.6}$$

where λ —FOV angle, β —opening angle equivalent (14.5636° for initial calculations), p —length of an arc (1 m for initial calculations). In general the length of an arc can be calculated as follows:

$$p = \frac{\pi r \lambda}{180}, \tag{13.7}$$

where r is radius of an arc (striking distance). To prevent the changes in selected resolution, the opening angle will be calculated using Eq. (13.8):

$$\beta = \frac{180}{\rho \pi r}, \tag{13.8}$$

Average opening angle for each of the striking distance zone:

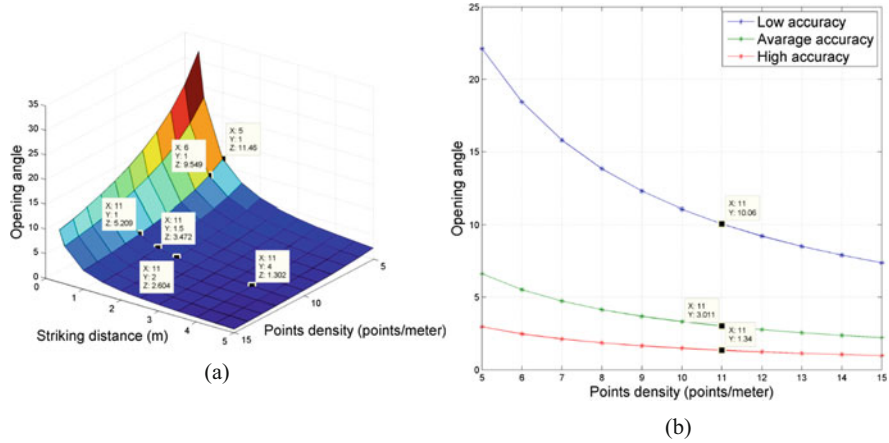


Fig. 13.7 Opening angle equivalents. (a) Dependencies of opening angle, striking distance and point cloud density. (b) Averaged values of point cloud density

$$\beta_i = \frac{\sum_{j=0}^n \beta_{ij}}{n}, \tag{13.9}$$

where β_i is an opening angle for each accuracy zone, β_{ij} is an opening angle for each striking distance in zone i .

According to the calculation, the average angles based on the initial point cloud density (11 points/meter) are 10.059° for “Low accuracy zone,” 3.011° for “Average accuracy zone,” and 1.34° for “High accuracy zone.” The average angle for the “Low accuracy zone” range will give a small resolution equal to 5–6 points per meter. So the low edge value of an opening angle for “Low accuracy zone” was taken. The set of angles changed to 5.209°, 3.011°, 1.34° (Fig. 13.7).

13.4 Path Planning Methods

Motion planning is one of the key tasks in robotics. In mathematics, there are well-developed algorithms for finding the way in an unknown or partially known environment (optimal and heuristic algorithms). For this purpose, discrete mathematics (graph theory) and linear programming are usually used. Tasks of the shortest path search in the graph are known and studied (for example, Dijkstra’s, Floyd-Warshell’s, Prim’s, Kruskal’s, algorithms, etc. [90] and [55]).

Exists many types of researches in the frame of path-planning. For example [91] where authors represented an approach that uses motion primitive libraries. In [92], representing an attempt to implement animal motion for robot behavior, or [93]

suggested an algorithm of collision free trajectory for robots. The articles cover some aspects of certain sub-tasks and widely describe special cases of behavior in a group of robots. However, no one has considered the task from the point of view of the interconnected global approach. It must include the correlation of robots' technical vision systems with communication and navigation rules in defined group of n -agents.

13.4.1 Path Planning Using Technical Vision System

The task of path-planning can be presented the next way:

- Robot is deployed in an unknown environment;
- Its current position marked as a starting point and target location as an end point;
- Robot calculates the heuristic route and starts movement towards the target;
- In case of obstacle detection by vision system, robot updates his navigation map and recalculates the route;
- processes of obstacle detection and path update continue until the target is reached.

Additionally the obtained path should be approximated to obtain a continuous and energy saving trajectory (Fig. 13.8).

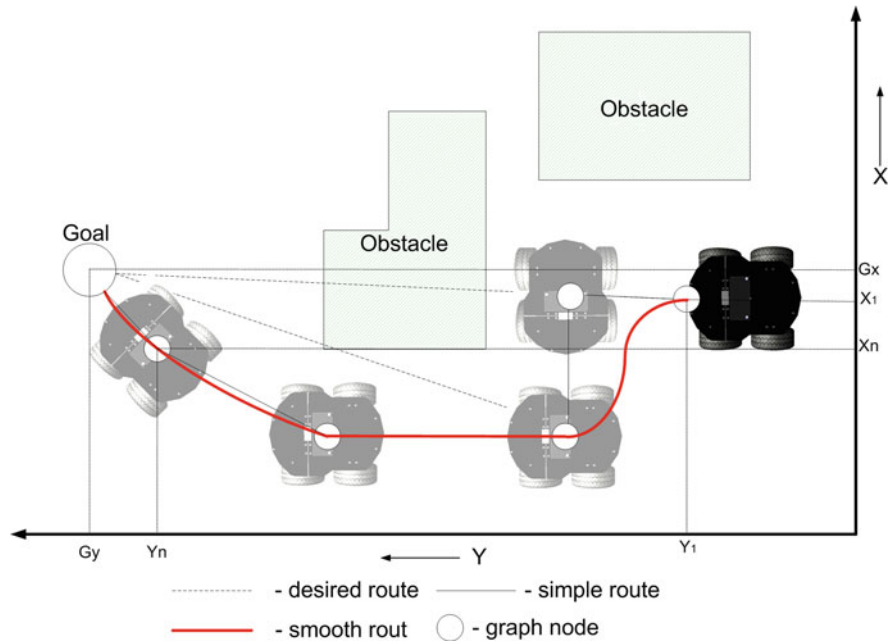


Fig. 13.8 Path planning

As was mentioned, each iteration of the heuristic path is calculated according to the current position of the robot and its surroundings. During the path calculation, robot will place additional shape points (where the direction of movement is changed). To avoid the collision, robot should take into consideration safe distance to an obstacle (Fig. 13.9).

According to the principles of TVS and previous research [88] were decided to use algorithm A* [94] for the tool of obstacle avoidance in this research. Terrain will be represented as matrix where each cell will have a size of robot's half diagonal. Cells are having two states: reachable and unreachable. Initially, all cells in the matrix (map) have the reachable state. After detecting an obstacle, the corresponding cell changes its own state to unreachable. The states of all surrounding cells are also set as unreachable to create a safe zone and avoid collisions during turns Fig. 13.10a.

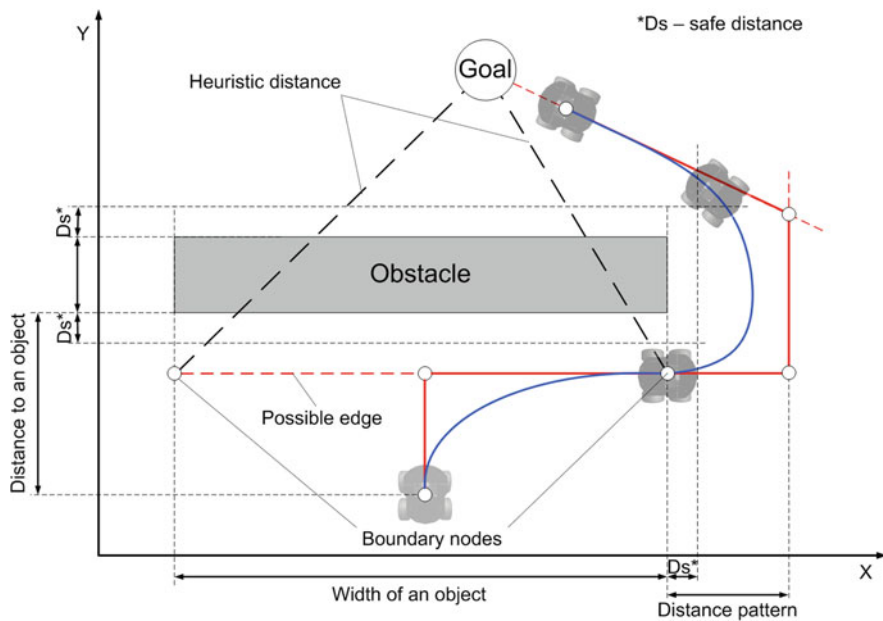


Fig. 13.9 Obstacle avoidance

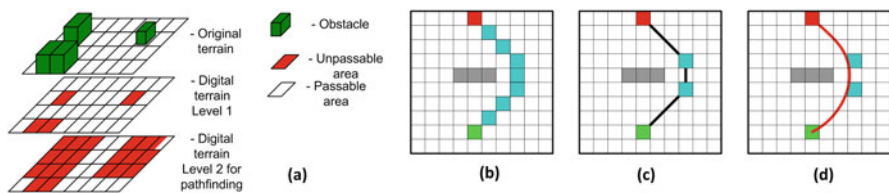


Fig. 13.10 (a) World representation and dead reckoning with two-step post processing. (b) Clear A*. (c) First step of post processing. (d) Second step of post processing

A* returns each cell that has to be visited during the movement Fig. 13.10b. To perform the post processing for receiving continuous trajectories in first step from the trajectory will be removed all unnecessary nodes keeping nodes where direction of movement is changed (Fig. 13.10c). On the second step of post processing should be made path approximation (Fig. 13.10d) for selecting the method of individual trajectory approximation to improve robot movement smoothness. It describes coherence between the decisions interrelation of the navigation system actions and the ability to anticipate and provide feedback to events with sufficient speed.

One of the brief solutions is to achieve approximated trajectory is to use Bezier curve. Such approach is useful and found its application in various tasks of path-planning for autonomous vehicles [95] and [96]. Mathematical parametric representation of a Bezier curve has the form:

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t), \quad 0 \leq t \leq 1 \tag{13.10}$$

where t is a parameter, n is the degree of Bernstein’s polynomial basis, i is the summation index, and B_i represents the i -th vertex of the Bezier polygon.

13.4.2 Secondary Objectives Placement for Surface Mapping

For the task of terrain mapping, it is necessary to locate additional points to visit. Solution can be adopted from surface mapping using UAV [97] or other autonomous surface vehicles [98]. These solutions are based on Dubins car principles [99, 100].

According to the Dubins principles, for the case of terrain mapping with single robot (Fig. 13.11a), territory is covered with pre-calculated trajectory. In place where the trajectory is changing its state (straight to round and round to straight), secondary

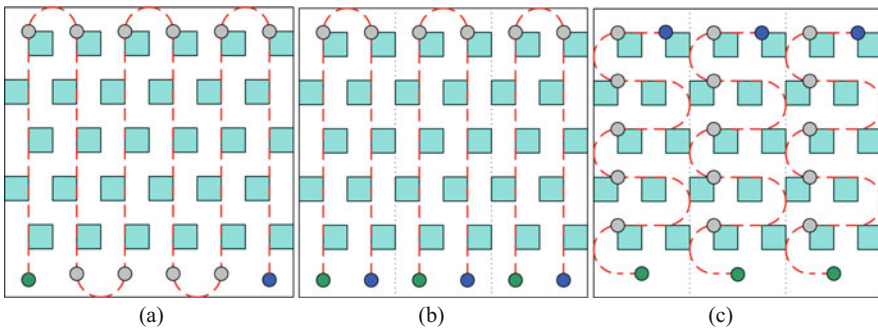


Fig. 13.11 Secondary objectives placement. (a) Single robot. (b) Group vertical movement. (c) Group horizontal movement

points are placed (light gray dots on Fig. 13.11). Terrain mapping using a group of robots can be separated in two types: vertical movement (Fig. 13.11b) and horizontal movements (Fig. 13.11c). In both types, territory is sliced into sectors (amount of sectors depends on amount of robots). Advantages of sectoring the terrain are described in [88].

These methods, in total, solve the task of motion planning for the independent robot in-group. It is obvious that data exchange between the robots in group is a good tool for generalized information obtaining with the aim of more complete implementation of all mentioned above methods.

13.5 Data Transferring Networks and Local Exchange of Information for Robotic Group

Communication within the robotic group is one of the main tasks in swarm robotics. Its implementation helps to expand the possibilities of a swarm by improving task of flocking, foraging, navigation, etc. Exists two types of communication, it can be with global or local interaction. During the global communication, received message contains local information of the sender. In most cases, this information is useless. In swarm or group robotics is used local communication. This type of communication takes its origin in nature (herding is a good example, where the local interaction helps to survive the predators attacks by signaling to their kin with movement or sounds).

In its turn, local communication can be direct and indirect. Direct communication is a real-time data transferring within the group. In this case, robot sends message to the group and they have to process it immediately. For the direct communication can be used WiFi connection, Bluetooth, or more primitive types of communications like light and sound. Indirect communications use different types of medium that can be used for late access information storing (mail services). For example, in swarm robotics it is implemented during the task of SLAM by living NFC card on detected landmark (implementation of pheromones used by ants).

Chapter considers two models of data transferring: information exchange with centralized management (Fig. 13.12a) and strategies of centralized hierarchical control (Fig. 13.12b).

In next sections will be presented solutions for data transferring task in swarm robotics. Solutions are inspired by Spanning Tree Protocol (STP) [101] and Shortest Path Bridging (SPB) [102].

Spanning Tree Protocol is a channel protocol. The main task of STP is to eliminate loops in the topology of an arbitrary Ethernet network, in which there is one or more network bridges connected by redundant connections. STP solves this problem by automatically blocking connections that are currently redundant for the full connectivity of switches.

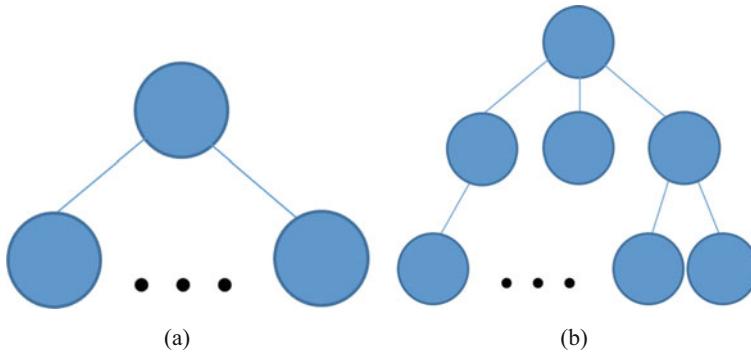


Fig. 13.12 Models of data transferring. (a) Information exchange with centralized management. (b) Information exchange using strategies of centralized hierarchical control

Shortest Path Bridging is a standardized IEEE as 802.1aq is a network technology that simplifies the building and networks configuration while taking advantage of multipath routing.

13.5.1 Spanning Tree Forming for Swarm Robotics

Consider a general case of swarm can be proposed a method of network forming based on creating a spanning tree. Algorithm consists of seven steps and includes the use of classical approaches. Steps of dynamic network forming for robotic swarm are next:

- Build a fully connected network graph;
- Use the Kruskal algorithm to build the minimum spanning tree;
- In the obtained tree, use the Floyd–Warshall algorithm to receive the list of all possible routes in the network;
- Calculate the average route length for each node;
- Select the node with the lowest average length and configure it as a high level node;
- Nodes with “one side” connection configured as low level nodes;
- Other nodes configured as mid-level nodes.

Applying this method, it is possible to obtain both of the networks types automatically (depends on the robots placements, their amount, and signal of the network). In particular cases (Fig. 13.13) in calculations is considered open space without obstacles, so the distances between nodes were used. In scenarios, that are more complicated, distances should be replaced with the wireless network signal levels.

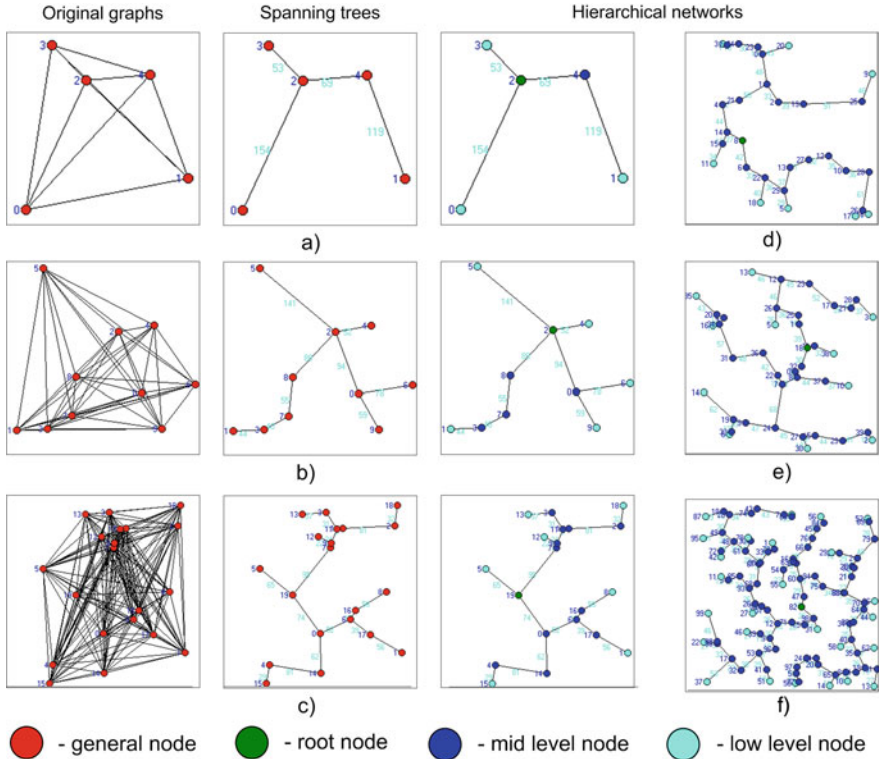


Fig. 13.13 Calculated networks. (a) 5 nodes network. (b) 10 nodes network. (c) 20 nodes network. (d) 30 nodes network. (e) 40 nodes network. (e) 50 nodes network

This method is useful for large swarm, but in case of a small group can be used methods that include more behavior control and solve several tasks simultaneously.

13.5.2 Leader Based Communication

One of the models that describe the locally interacting robots organization is static swarm [91]. It is characterized by the absence of a given control center and is some kind of a fixed network—a set of agents. The basic properties of static swarm are activity, local interactions, and functional heterogeneity. That is why will be reviewed the method of role distribution based on the task of selecting a leader. Under the term “Leader,” we will understand the central node of data exchange (robot for a short period of time will become server to store and merge data). For choosing a leader, robots will be using a voting process. Each robot can be described as a set of parameters:

$$R = (I, L, E, N) \quad (13.11)$$

where I —identifier of robot, L —identifier of voted leader, E —evaluation of the leader L (amount of voices that have to be given for a leader), N —list of connections available for robot (its neighbors).

The voting process on the initial step goes the following way: each robot evaluates his neighbors for the role of leader according to the set of previously defined characteristics; each of these characteristics has their own weight; using the membership function, robot selects the neighbor with the highest value. For the vote, value will be implemented a linguistic variable e = “evaluation of robot.” Its value is based on the scale of M = “very low,” “low,” “medium,” “high,” “very high” or it can have a digit equivalent $M = 1, 2, 3, 4, 5$. After voting process, many alternatives for E will be generated, so it will have next form:

$$E = \{e_1, e_2, \dots, e_n\} \quad (13.12)$$

where e_i —alternative “candidate” at and n is amount of visible neighbors. For robot evaluation are offered following characteristics: (1) surroundings: c_1 = “the number of neighbors evaluable for candidate”; (2) territorial: c_2 = “the distances to the neighbors or levels of signals”; (3) status: c_3 = “the physical state of the robot.” Each of these characteristics is estimated by a voting robot for each of its neighbors:

$$C_i = \{c_{i1}, c_{i2}, \dots, c_{ik}\} \quad (13.13)$$

where c_j —characteristic value of i -th “candidate” at $j = 1 \dots k$. Each of the characteristics has its weight:

$$W = \{w_1, w_2, \dots, w_k\} \quad (13.14)$$

where—the j -th characteristics weighting $\sum w_i = 1$, evaluation of the i -th neighbors uses the following formula:

$$e_i = \sum_{j=1}^k w_j c_{ij} \quad (13.15)$$

To determine the value of linguistic variable, we use three types of membership functions (Eqs. (13.16)–(13.18)), where a general view represented on Fig. 13.14.

$$f_{vl}(e_i) = \left\{ \begin{array}{l} 1, x < vle \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{e_i - vle}{vl - vle}\right), vle \leq x \leq vl \\ 0, x > vl \end{array} \right\} \quad (13.16)$$

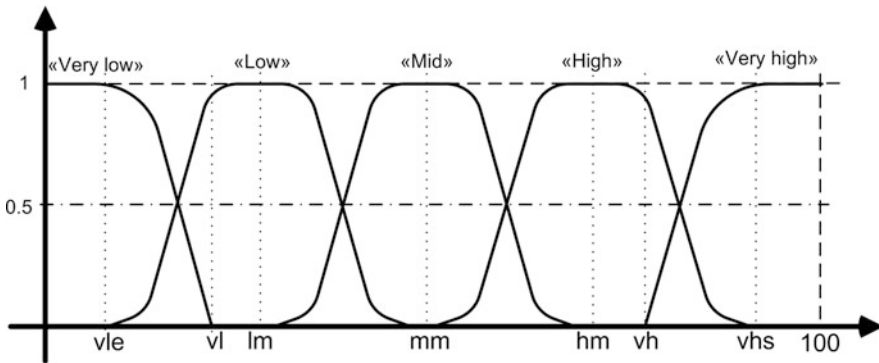


Fig. 13.14 Membership functions

where vle —the threshold to which the membership function is equal “1”, vl —a threshold, after which the membership function is equal “0”.

$$f_{vh}(e_i) = \left\{ \begin{array}{l} 1, e_i > vhs \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{e_i - vhs}{vhs - vh} \pi\right), vh \leq e_i \leq vhs \\ 0, e_i < vh \end{array} \right\} \tag{13.17}$$

where similar to Eq. (13.16) vhs —threshold for “1” and vh —for “0”.

$$f_{gb}(e_i) = \frac{1}{1 + \left|\frac{e_i - c}{a}\right|^{2b}} \tag{13.18}$$

where c —middle part of a membership function, a —value at which $f_{gb}(c + a) = 1$ and $f_{gb}(c - a) = 1$, b —the value of function smooth regulation.

13.5.3 Feedback Implementation

In the case where some of the information need to be transferred between all robots within the group based on that certain of them (robots) have to be notified that transfer is complete. This task is the dissemination of information feedback (PIF—propagation of information with feedback) is formulated as follows: a subset is formed by robots of those which know message M (the same for all robots) which should be spread, that is, all robots must take M . Certain processes must be notified of the transfer is complete, a special event notification must be done, and it can only be done when all processes have already received the M . Alert in PIF-algorithm can be viewed as a return (OK) event.

Any wave algorithm can be used as PIF-algorithm. For example, let A be a wave algorithm. To use A as a PIF-algorithm, we take the robots, initially knowing the M is an initiators of A. Information M is added to each message A. This is possible because, by construction, starters of A know M initially, and the followers do not send messages until they receive at least one message, that is, until they get M. When return (OK) events occur, each process knows the M, and return event (OK) can be considered as the required notify event in PIF-algorithm.

Two models of data transferring were reviewed previously: information exchange with centralized management (Fig. 13.12a) and strategies of centralized hierarchical control (Fig. 13.12b). When using the strategy of centralized management of a robotic group R, every robot r_i ($i = 1, 2, \dots, N$) of group transmits data about its state and information obtained about the environment in the central control device (robot chosen by the estimation process).

Hierarchical strategy of centralized management network between robots can be represented with layers. Layers can be separated into three types: top layer is a single central control device which merges data and initiates backwards propagation; middle layer is a group of control devices for existing to send their data and data from lower levels (layers) to top layer; low layer can communicate only with the elements of middle layer, sending the data and receiving the data after merging.

Leader-changing method can be simplified for the layers distribution inside the group. To define network roles will be implementing linguistic variable $p =$ “pattern of layer.” It uses three levels scale of M = “lower layer,” “middle layer,” “top layer”. Correspondingly, many alternatives of P can be represented in the following form:

To determine the value of linguistic variable (Fig. 13.15), we use three types of membership functions [103], where the extreme values (“Low level” and “Top level”) will determine Z- shaped (31) and the S-shaped (32) functions, the degree of belonging to the “Middle level” value is based on trapeze-like membership function (33) (general formulas are represented).

$$f_{low} = \left\{ \begin{array}{l} 1, e_i \leq a \\ \frac{b-e_i}{b-a}, a < x \leq b \\ 0, x > b \end{array} \right\} \tag{13.19}$$

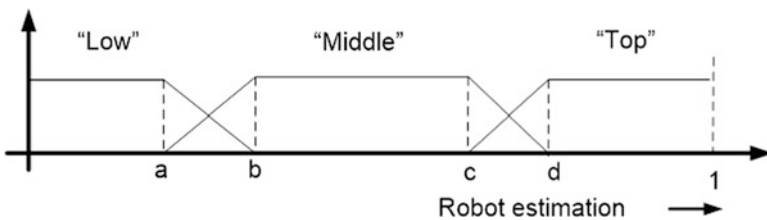


Fig. 13.15 Functions for network layer determination

$$f_{top} = \left\{ \begin{array}{l} 0, e_i \leq c \\ \frac{e_i - a}{b - a}, c < e_i \leq d \\ 1, x > d \end{array} \right\} \tag{13.20}$$

$$f_{mid} = \left\{ \begin{array}{l} 0, e_i \leq a \\ \frac{e_i - a}{b - a}, a < e_i \leq b \\ 1, b < e_i \leq c \\ \frac{d - e_i}{d - c}, c < e_i \leq d \\ 0, e_i > d \end{array} \right\} \tag{13.21}$$

where e_i is evaluation of the i -th robot takes the following form

$$e_i = \sum_{j=1}^k w_j c_{ij} \tag{13.22}$$

In terms of fuzzy logic, it can be described using next IF—THEN rules type:

```

IF robot evaluation IS top level,
    THEN network level EQUALS host
IF robot evaluation IS mid level,
    THEN network level EQUALS level 1
IF robot evaluation IS low level,
    THEN network level EQUALS level 2
    
```

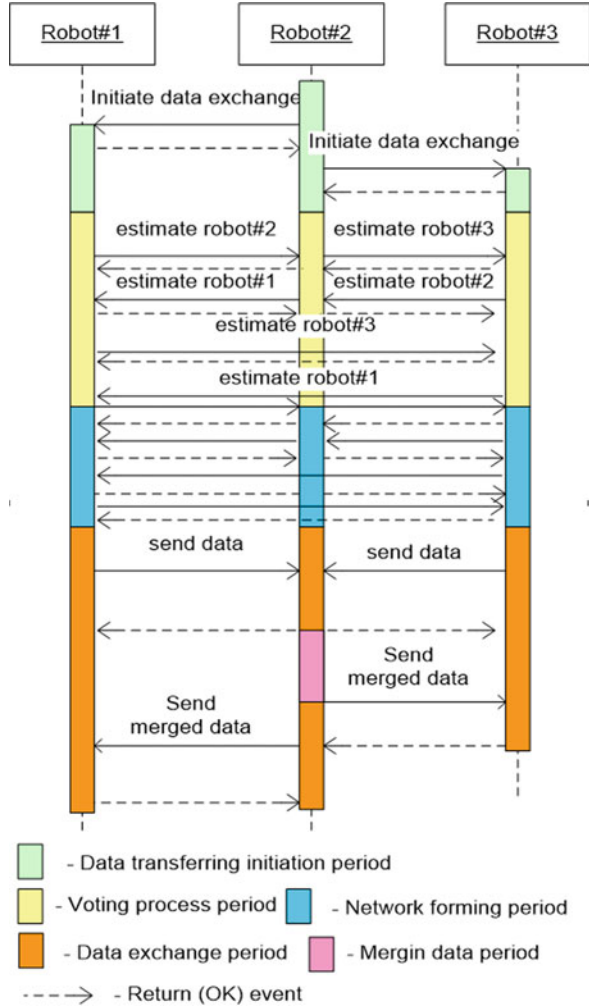
where “NET HOST”—robot becomes a host for data transferring (top level), “NET LVL 1” and “NET LVL 2” for determining the network level for communication (middle and low levels), and n —fixing statement.

Schematic representation of PIF for the current case is in Fig. 13.16.

Proposed dynamic data exchange network forming method extends the potential of our novel TVS. It overlaps an ability of single robot navigation with a cloud-like common knowledge base within the robotic group to improve the efficiency of dead reckoning.

The proposed methods allow the elimination of topological loops in the data network in a group of robots. A fully connected graph of a real network with a high probability leads to endless repetitions of the same messages in a group, while network bandwidth is almost completely occupied by these useless replays. In these conditions, formally the network can continue to operate but in practice its performance becomes so low that may lead to a complete network failure. Therefore, the proposed methods ensure the full propagation of information within the group and help to improve the movement coordination of a robotic group by exchanging information about the missing sectors.

Fig. 13.16 Margin data about environment (sequence diagram). Data transferring initiation period is at a state when one of the robots in group sends messages to others to start data transferring, and occurs when: (1) robot needs additional data for further navigation, or (2) robot has collected enough of a portion of information from TVS that seems to be transferred to others in group. The voting process period is used for evaluation of each robot in group. Compilation of data transferring channels happens at network forming period. The data exchange period is used to interchange the accumulated data according to the topological structure of the network. After this comes the data merge. The last two periods have floating time depending on the amount of data accumulated by each robot



13.6 Surface Mapping

13.6.1 Simulation Frameworks

Before every complex mechanical system practical implementation, it has to go through two stages: theoretical justification and realistic simulation. Create a digital model of the entire system significantly affects the overall efficiency of the project. Process of simulation gives an opportunity to reduce the mistakes during the development, improve the output of the system according to the changes in environmental conditions, and reduce costs of technical issues. Among other benefits of simulation are:

- Reduce the cost of manufacturing robots;
- Resource management and source code diagnostics;
- Simulate different alternatives;
- Before the implementation of the robot, its components can be verified;
- Modeling can be done in stages, for complex projects favor;
- To determine the viability of the system;
- Compatibility with a wide range of programming languages;

However, disadvantages of simulation also can be found. An application can simulate the robotic behavior model only with predefined rules, it will not mimic factors that are not taken into account in the development phase. Real world experience can provide more scenarios than the computational model.

Nowadays, simulation platforms cover a lot of tools and features that can provide close to real life simulations. Most of them use different C/C++ like algorithmic languages, LabVIEW, MATLAB, etc. In this section, several used simulation platforms are summarized.

Player/Stage [104] is a project under which three robotics-related software platforms are currently being developed. It consists of the Player network robotics server, the Stage-2D robot simulation environment, and the Gazebo-3D robot simulation environment. The project was founded in 2000 by Brian Gerkey, Richard Vaughan, and Nathan Koenig at the University of Southern California in Los Angeles and is widely used in research and training within robotics.

The UberSim [105] is a simulator designed for quick testing before uploading program to football robot. UberSim uses the ODE physics engine. Software supports custom robots and sensors.

USARSim [106] is a simulation of urban search and rescue. It is based on Unreal Engine 2.0. USARSim can be interfaced with Player and runs on Windows, Linux, and Mac OS.

Breve [107] is a 3D simulation environment for distributed artificial life systems. Behavioral models are defined using Python. Like UberSim, Breve uses ODE physics engine and OpenGL for 3D graphical representation.

V-REP [108] is a useful 3D simulator for educational process, allows modeling of complex systems, individual sensors, mechanisms, and so on.

Webots [109] is a software product of Swiss company Cyberbotics. It provides supports of different programming languages like C/C++, Java, Python, URBI, and MATLAB. Moreover is compatible with third-party software through TCP/IP.

Gazebo [110] can simulate complex systems and a variety of sensor components. It is helpful in developing robots used in interaction, to lift or grab objects, to push and activities that require recognition and localization in space.

ARGoS [111] is a modular, multi-engine simulator for heterogeneous swarm robotics. System is able to use about 10,000 wheeled robots in real-time during the simulation.

TeamBots [112] is a multi-agent simulation program for robots that allows you to create multi-agent control systems in dynamic environments with visualization. You can develop your control system and implement it in a simulation program, and then test your control system in a real mobile robot.

13.6.2 Modeling System Structure

To prove the theoretical basis of presented questions was used developed software for the simulation and robotic group collaboration. Presented framework has been developed in Unity 5 [59, 113], it is a multiplatform engine provided with different features and tools. Software was developed using programming language C# in MonoDevelop integrated development environment (IDE) for Windows 10. Software (Fig. 13.17) has three operating modes “Without common knowledge,” “Pre-known territory,” “With common knowledge.” First two use only part of decision-making system for path planning and obstacle avoidance. The third one implements full stack of decision-making process. In the end, system returns data about environment and state of each robot in moment of time.

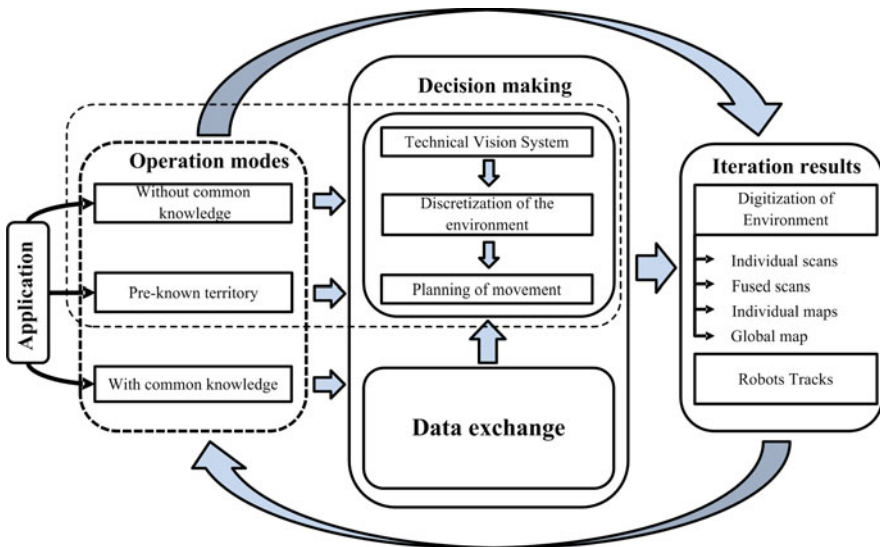


Fig. 13.17 System structure

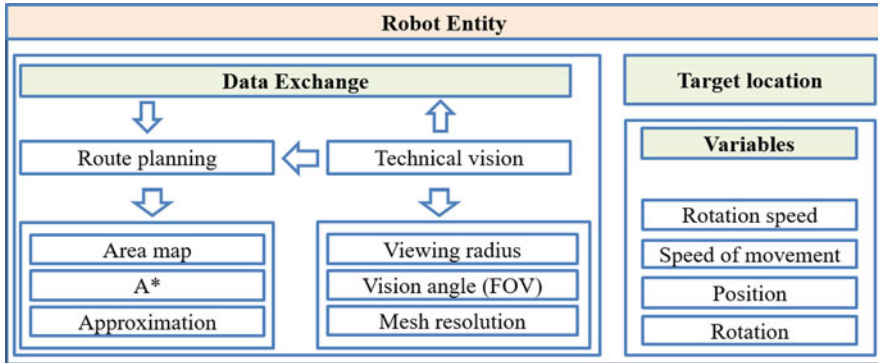


Fig. 13.18 Robot entity

Main object of the system is robot. It can be described with a set of variables (rotation speed, speed of movement, current position, and spatial orientation), goal position, and decision-making system (Fig. 13.18).

In simulation were used four different random scenes presented in Fig. 13.19, the modeling results of with will be presented in next section.

13.6.3 Influence of Data Exchange on Path Planning

To receive the results for each scene were made three modeling scenarios. In first scenario, robots are moving independently from each other (no knowledge sharing and data exchange). In second scenario, three robots are fusing obtained information and are using common knowledge of terrain for path planning (implemented data exchange method). In last one scenario, robot is moving in pre-known terrain. In each case, robots have to reach their personal goals and then get to the common point.

Modeling includes the group of three robots using Pioneer 3-AT and TVS (Fig. 13.20). This robotic platform is reviewed previously in work [80], also this article describes kinematic of the platform.

For all of the scenarios in each scene were made 100 simulations. Results of the modeling and aggregated data are represented in Figs. 13.21, 13.22, 13.23, and 13.24. The first thing that becomes visible on the graphs is that the trajectory lengths deviation of “No common knowledge” and “With common knowledge.”

The decreasing of deviation in case of “With common knowledge” signifies that trajectories forms are strive to the optimal (not taking into account the individual anomalies that have occurred).

The summary of the modeling presented in Fig. 13.25. Comparing averaged distances obtained during the modeling can be seen that use common knowledge base have advantages in all of the scenes. Result shows that robotic group with

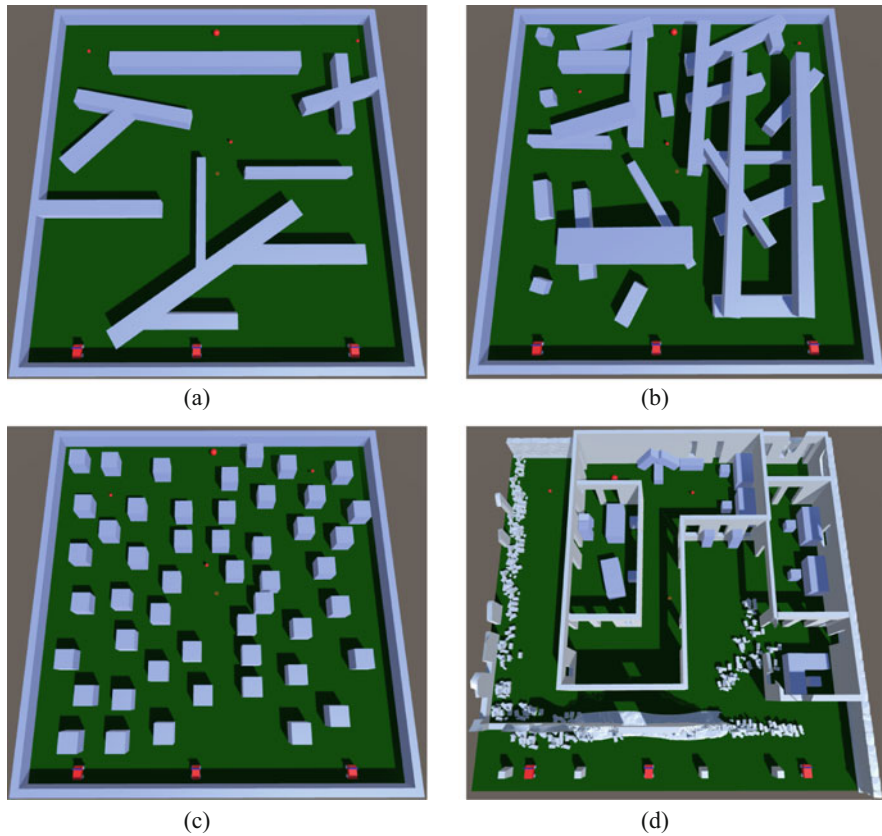


Fig. 13.19 Scenes used for modeling. (a) Scene #1. (b) Scene #2. (c) Scene #3. (d) Scene #4

Fig. 13.20 Pioneer 3-AT mobile robotic platform



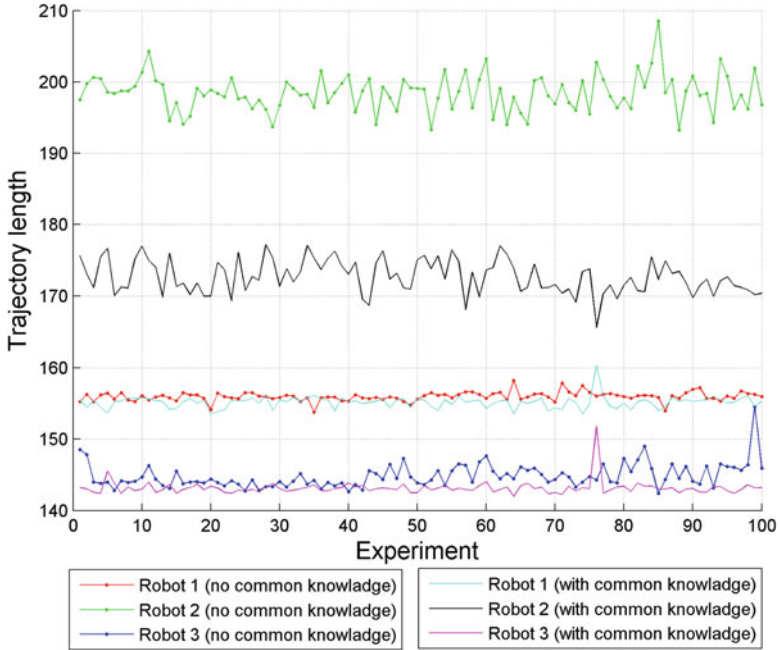


Fig. 13.21 Length of trajectories: Scene #1

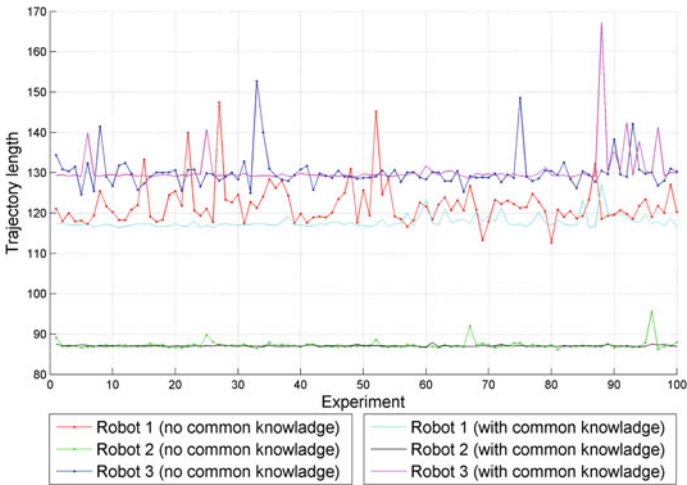


Fig. 13.22 Length of trajectories: Scene #2

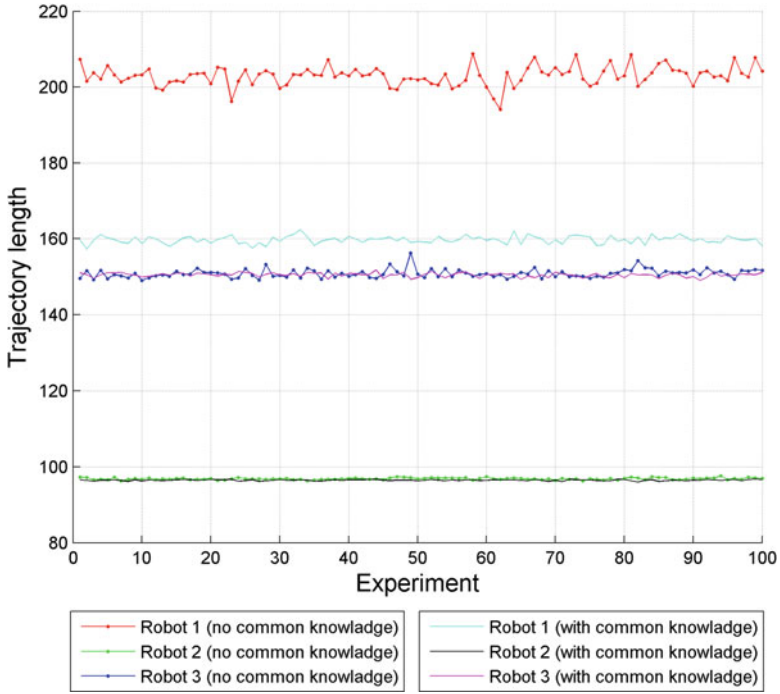


Fig. 13.23 Length of trajectories: Scene #3

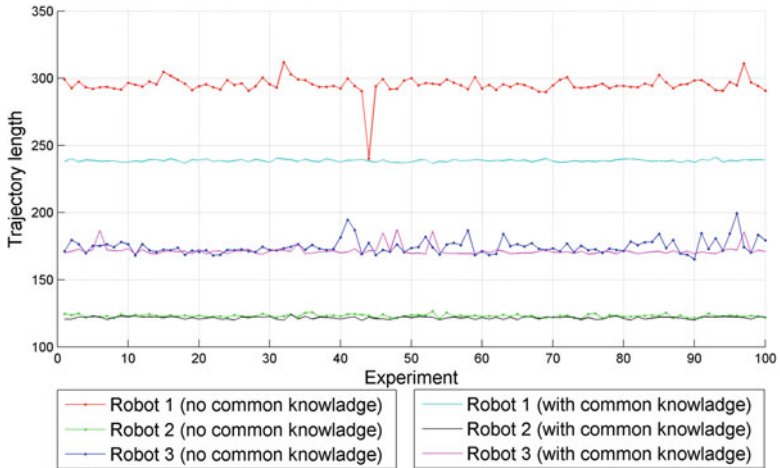


Fig. 13.24 Length of trajectories: Scene #4

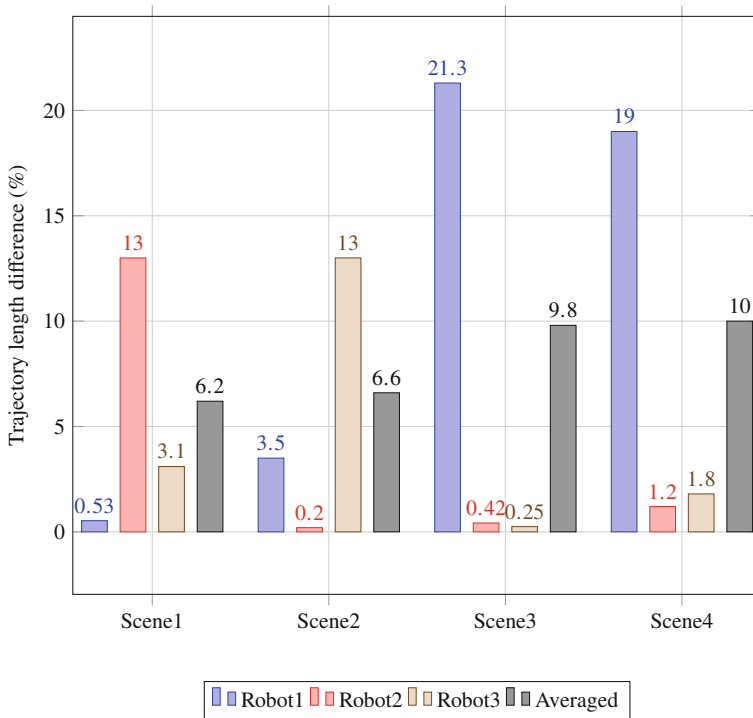


Fig. 13.25 Comparing trajectory lengths for each of the scenes in percent

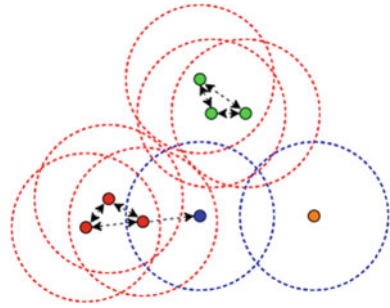
implemented data exchange system has averaged group trajectory length 6.2–10% better comparing to distances of autonomous robotic movement (outside of group). Scaling the results for individual robots in group the improvement of trajectories can reach up to 21.3%.

13.6.4 Objects Extraction

Density-based spatial clustering of applications with noise (DBSCAN) [114] algorithm was proposed by Martin Ester, Hans-Peter Kriegel, and colleagues in 1996 as a solution to the problem of splitting (initially spatial) data into clusters of arbitrary shape. Most algorithms that produce a flat partition create clusters that are close to spherical in shape, since they minimize the distance of the documents to the center of the cluster. DBSCAN authors experimentally showed that their algorithm is able to recognize clusters of various shapes.

The idea of the algorithm is that inside each cluster there is a typical density of points, which is noticeably higher than the density outside the cluster, as well as the density in areas with noise below the density of any of the clusters. For each point

Fig. 13.26 DBSCAN clustering illustration



of the cluster, its neighborhood of a given radius must contain at least some number of points, this number of points is specified by a threshold value (Fig. 13.26).

The algorithm can be presented as next:

1. Given the dataset
2. Label all points as core or non core
3. Until all core points are visited:
 - (a) Add one of non visited core point P to a new cluster
 - (b) Until all points in cluster are visited:
 - For each non visited core point P within the cluster:
 - Add all core points within boundary of P to the cluster
 - Mark P as visited
4. Until all non-core points are visited:
 - (a) If a non-core point P has a core point within its boundary, add it to the cluster corresponding to that core point
 - (b) Else ignore

Example of algorithm implementation presented in Fig. 13.27. Here can be seen a scanned environment using the group of robots and TVS (Fig. 13.27a). After implementation of DBSCAN can be seen clustered objects (Fig. 13.27b).

According to the clustered data set objects can be extracted for further analysis (Fig. 13.28). Obtained data can be used in many applications like object classification and recognition, surface reconstruction, etc.

One of the main sub applications is to use data for structural health monitoring. The scanned surface (Fig. 13.29a) can be analyzed and reconstructed to detect crack (Fig. 13.29b) or other problems that can accrue.

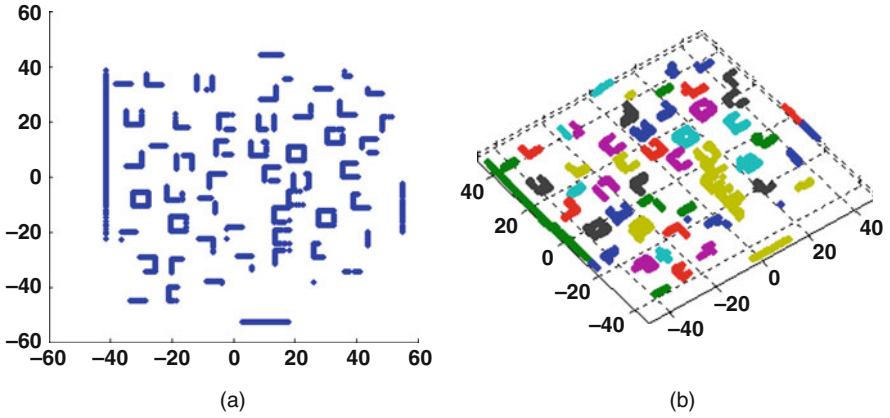


Fig. 13.27 Example of DBSCAN implementation. (a) Original point cloud of scene. (b) Clustered scene

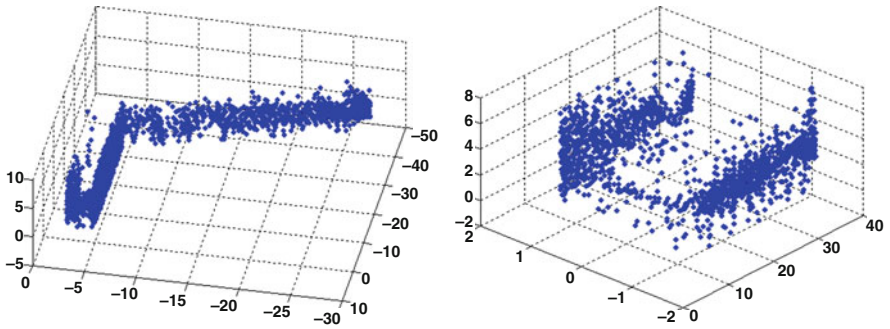


Fig. 13.28 Extracted objects

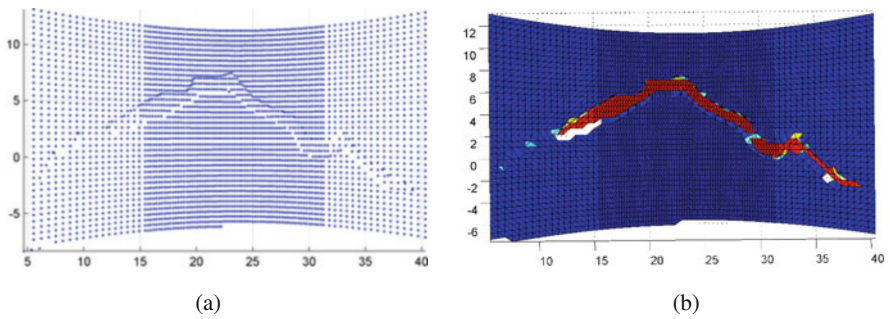


Fig. 13.29 Implementation of approach for structural health monitoring. (a) Scan of cracked surface. (b) Detected crack

13.6.5 Effectiveness of Robotic Group

Let us review effectiveness of robots based on the “Scene #4” (Fig. 13.29). As effectiveness will be understood, the amount of unique data obtained by robot comparing to common data fusion. On Fig. 13.30 are shown four different binary maps that in the end were known to robotic group (three individual maps for each robot and fused map).

Overlapping one individual binary map on other is possible to say that some sectors were detected only by one robot, other by two or three robots (Fig. 13.31).

Besides the overlapped data can be allocated another characteristic—ratio of individual data obtained by robot to total data.

Subtracting overlapped data from each individual scan returns unique data of each robot in group. In this case, ratio equals to 0.58 for Robot #1, 0.486 for Robot #2, and 0.486 for Robot #3. Comparing it to fused data will give the effectiveness

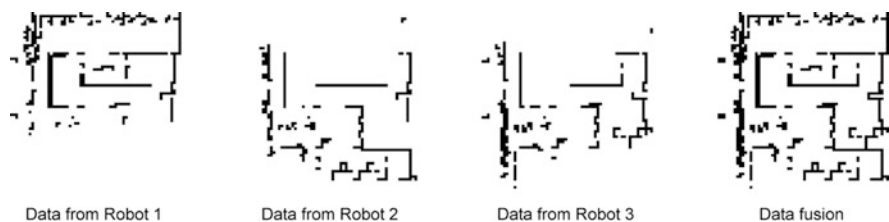


Fig. 13.30 Binary maps of environment

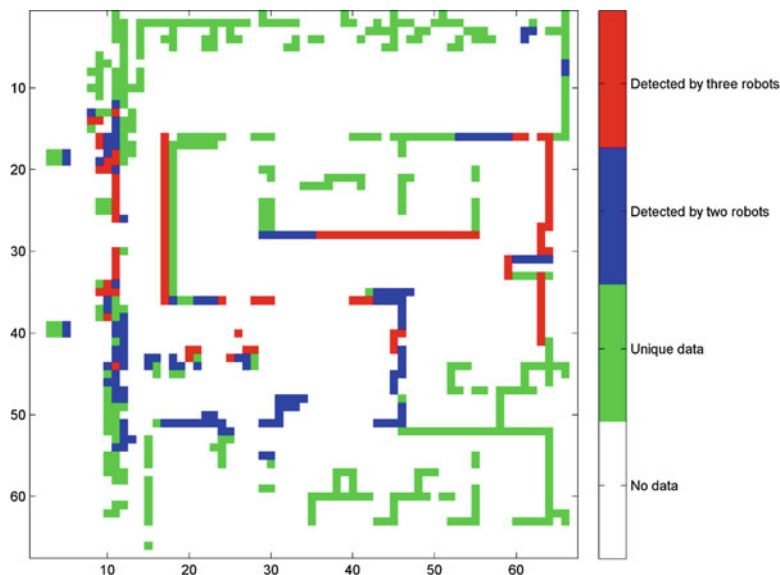


Fig. 13.31 Overlapped individual binary maps

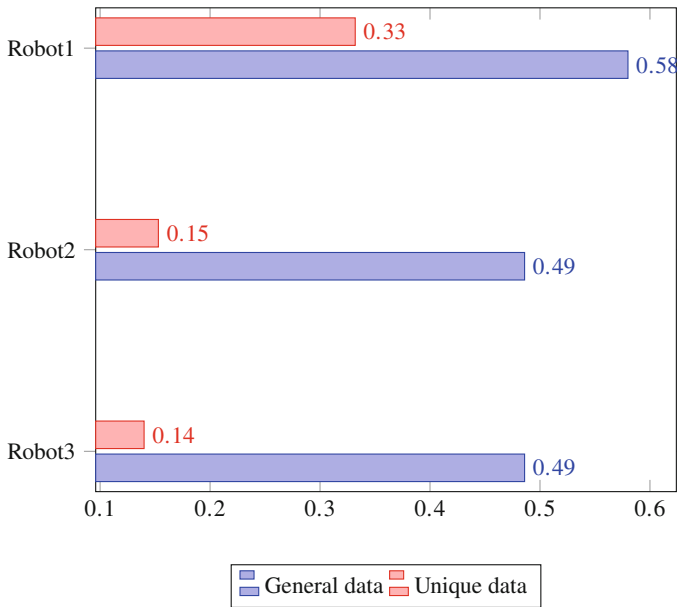


Fig. 13.32 Unique and general data comparison for each robot in group

of each of the robot. Here results are 0.332, 0.153, and 0.14 for first to third robot, respectively (Fig. 13.32).

13.7 Conclusions

Chapter offers an original solution that improves robotic group teamwork. The implementation of communication within the group as the result gave a decreased trajectory length (up to 21.3%) for individual navigation. Other benefit presented in chapter is field of view sectoring. It gives the possibility to free a significant part of memories on individual robot for complementary task solution in real time and use the data only needed for navigation task. Simulations have shown that the use of mentioned improvements applied to behavior of robotic group allows stable functioning of the group at lower energy costs of motion (bending energy) and decreases trajectories length. Presented approach can be applied to the various tasks like surroundings mapping after natural or human-caused disaster, indoor navigation, etc. while the reconstructed 3D image in its turn is possible to use for structural health monitoring.

References

1. Atyabi, A., Phon-Amnuaisuk, S., & Ho, C. K. (2010). Navigating a robotic swarm in an uncharted 2d landscape. *Applied Soft Computing*, 10(1), 149–169.
2. Levi, P., Meister, E., & Schlachter, F. (2014). Reconfigurable swarm robots produce self-assembling and self-repairing organisms. *Robotics and Autonomous Systems*, 62(10), 1371–1376.
3. de Sá, A. O., Nedjah, N., & de Macedo Mourelle, L. (2017). Distributed and resilient localization algorithm for swarm robotic systems. *Applied Soft Computing*, 57, 738–750.
4. Teoh, E. R., & Kidd, D. G. (2017). Rage against the machine? Google's self-driving cars versus human drivers. *Journal of Safety Research*, 63, 57–60.
5. Morales, Y., Watanabe, A., Ferreri, F., Even, J., Shinozawa, K., & Hagita, N. (2018). Passenger discomfort map for autonomous navigation in a robotic wheelchair. *Robotics and Autonomous Systems*, 103, 13–26.
6. Bond, A. H., & Gasser, L. (1992). A subject-indexed bibliography of distributed artificial intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), 1260–1281.
7. Bond, A. H., & Gasser, L. (2014). *Readings in distributed artificial intelligence*. San Mateo, CA: Morgan Kaufmann.
8. Boes, J., & Migeon, F. (2017). Self-organizing multi-agent systems for the control of complex systems. *Journal of Systems and Software*, 134, 12–28.
9. Tan, Y., & Zheng, Z.-y. (2013). Research advance in swarm robotics. *Defence Technology*, 9(1), 18–39.
10. Nebti, S., & Boukerram, A. (2017). Swarm intelligence inspired classifiers for facial recognition. *Swarm and Evolutionary Computation*, 32, 150–166.
11. Mavrovouniotis, M., Li, C., & Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33, 1–17.
12. Parr, L. A., Winslow, J. T., Hopkins, W. D., & de Waal, F. (2000). Recognizing facial cues: Individual discrimination by chimpanzees (pan troglodytes) and rhesus monkeys (*Macaca mulatta*). *Journal of Comparative Psychology*, 114(1), 47.
13. Parr, L. A., & de Waal, F. B. (1999). Visual kin recognition in chimpanzees. *Nature*, 399(6737), 647.
14. Shapiro, J. A. (1998). Thinking about bacterial populations as multicellular organisms. *Annual Reviews in Microbiology*, 52(1), 81–104.
15. Costerton, J. W., Lewandowski, Z., Caldwell, D. E., Korber, D. R., & Lappin-Scott, H. M. (1995). Microbial biofilms. *Annual Reviews in Microbiology*, 49(1), 711–745.
16. Wallraff, H. G., & Wallraff, H. G. (2005). *Avian navigation: Pigeon homing as a paradigm*. New York: Springer.
17. Jackson, D. E., & Ratnieks, F. L. (2006). Communication in ants. *Current Biology*, 16(15), R570–R574.
18. Goss, S., Aron, S., Deneubourg, J.-L., & Pasteels, J. M. (1989). Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12), 579–581.
19. Ravary, F., Lecoutey, E., Kaminski, G., Châline, N., & Jaisson, P. (2007). Individual experience alone can generate lasting division of labor in ants. *Current Biology*, 17(15), 1308–1312.
20. Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., et al. (2006). From disorder to order in marching locusts. *Science*, 312(5778), 1402–1406.
21. Bone, Q., & Moore, R. (2008). *Biology of fishes*. New York: Taylor & Francis.
22. Pitcher, T., Magurran, A., & Winfield, I. (1982). Fish in larger shoals find food faster. *Behavioral Ecology and Sociobiology*, 10(2), 149–151.
23. Moyle, P. B., & Cech, J. J. (2004). *Fishes: an introduction to ichthyology*. No. 597. Upper Saddle River, NJ: Pearson Prentice Hall.
24. Dyer, J. R., Ioannou, C. C., Morrell, L. J., Croft, D. P., Couzin, I. D., Waters, D. A., et al. (2008). Consensus decision making in human crowds. *Animal Behaviour*, 75(2), 461–470.

25. Marocco, D., & Nolfi, S. (2006). Origins of communication in evolving robots. In *International Conference on Simulation of Adaptive Behavior* (pp. 789–803). Heidelberg: Springer.
26. Hayes, A. T., Martinoli, A., & Goodman, R. M. (2003). Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, 21(4), 427–441.
27. de Oca, M. A. M., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., & Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3–4), 305–327.
28. Scheidler, A., Brutschy, A., Ferrante, E., & Dorigo, M. (2016). The k-unanimity rule for self-organized decision-making in swarms of robots. *IEEE Transactions on Cybernetics*, 46(5), 1175–1188.
29. Valentini, G., Hamann, H., & Dorigo, M. (2015). Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, (Richland, SC)* (pp. 1305–1314). International Foundation for Autonomous Agents and Multiagent Systems.
30. Wawerla, J., Sukhatme, G. S., & Mataric, M. J. (2002). Collective construction with multiple robots. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 3, pp. 2696–2701). Piscataway: IEEE.
31. Werfel, J., Bar-Yam, Y., & Nagpal, R. (2005). Building patterned structures with robot swarms. In *Proceedings of the IJCAI* (pp. 1495–1504).
32. Allwright, M., Bhalla, N., El-faham, H., Antoun, A., Pinciroli, C., & Dorigo, M. (2014). Srocs: Leveraging stigmergy on a multi-robot construction platform for unknown environments. In *International Conference on Swarm Intelligence* (pp. 158–169). Berlin: Springer.
33. Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2006). Autonomous self-assembly in a swarm-bot. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)* (pp. 314–322). Berlin: Springer.
34. Tuci, E., Ampatzis, C., Trianni, V., Christensen, A. L., & Dorigo, M. (2008). Self-assembly in physical autonomous robots—the evolutionary robotics approach. In *Proceedings of the ALIFE* (pp. 616–623).
35. Trianni, V., Nolfi, S., & Dorigo, M. Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, 54(2), 97–103 (2006)
36. O’Grady, R., Groß, R., Christensen, A. L., Dorigo, M. (2010). Self-assembly strategies in a group of autonomous mobile robots,” *Autonomous Robots*, 28(4), 439–455.
37. Bashyal, S., & Venayagamoorthy, G. K. (2008, Sept) Human swarm interaction for radiation source search and localization. In *2008 IEEE Swarm Intelligence Symposium* (pp. 1–8).
38. Walker, P., Amraii, S. A., Chakraborty, N., Lewis, M., & Sycara, K. (Sept 2014). Human control of robot swarms with dynamic leaders. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1108–1113).
39. Kolling, A., Sycara, K., Nunnally, S., & Lewis, M. (June 2013). Human-swarm interaction: An experimental study of two types of interaction with foraging swarms. *Journal of Human-Robot Interaction*, 2, 103–129.
40. O’Grady, R., Christensen, A. L., & Dorigo, M. (2009). Swarmorph: Multirobot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, 25(3), 738–743.
41. Brambilla, M., Pinciroli, C., Birattari, M., & Dorigo, M. (2009). A reliable distributed algorithm for group size estimation with minimal communication requirements. In *International Conference on Advanced Robotics, 2009. ICAR 2009.* (pp. 1–6). Piscataway: IEEE.
42. Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172, 292–321.
43. Chen, S., & Fang, H. (2005). Modeling and behavior analysis of large-scale social foraging swarm. *Control and Decision*, 20(12), 1392.
44. Beni, G. (1988). The concept of cellular robotic system. In *IEEE International Symposium on Intelligent Control, 1988. Proceedings* (pp. 57–62). Piscataway: IEEE.
45. Asama, H., Matsumoto, A., & Ishida, Y. (1989). Design of an autonomous and distributed robot system: Actress. In *Proceedings of the IROS* (vol. 89, pp. 283–290).
46. Payton, D., Daily, M., Estowski, R., Howard, M., & Lee, C. (2001). Pheromone robotics. *Autonomous Robots*, 11(3), 319–324.

47. Payton, D., Estkowski, R., & Howard, M. (2003). Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems*, 44(3–4), 229–240.
48. Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International Workshop on Swarm Robotics* (pp. 10–20). Heidelberg: Springer.
49. McLurkin, J., & Smith, J. (2004). Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Citeseer.
50. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., et al. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* (Vol. 1, pp. 59–65). IPCB: Instituto Politécnico de Castelo Branco.
51. Turgut, A. E., Çelikkanat, H., Gökçe, F., & Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2–4), 97–120.
52. Rubenstein, M., Ahler, C., & Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3293–3298). Piscataway: IEEE.
53. Seyfried, J., Szymanski, M., Bender, N., Estaña, R., Thiel, M., & Wörn, H. (2004). The i-swarm project: Intelligent small world autonomous robots for micro-manipulation. In *International Workshop on Swarm Robotics* (pp. 70–83). Berlin: Springer.
54. Beshers, S. N., & Fewell, J. H. (2001). Models of division of labor in social Insects. *Annual Review of Entomology*, 46(1), 413–440.
55. Trianni, V., Tuci, E., Ampatzis, C., & Dorigo, M. (2014). Evolutionary swarm robotics: A theoretical and methodological itinerary from individual neuro-controllers to collective behaviours. *The Horizons of Evolutionary Robotics* (pp. 153–160). New York: ACM .
56. Konolige, K., Fox, D., Ortiz, C., Agno, A., Eriksen, M., Limketkai, B., et al. (2008). Centibots: Very large scale distributed robotic teams. In *Experimental Robotics IX* (pp. 131–140). Springer.
57. Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., et al. (2013). Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4), 60–71.
58. Rybski, P. E., Burt, I., Dahlin, T., Gini, M., Hougen, D. F., Krantz, D. G., et al. (2001). System architecture for versatile autonomous and teleoperated control of multiple miniature robots. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001* (Vol. 3, pp. 2917–2922). Piscataway: IEEE.
59. Suárez, P., Iglesias, A., & Gálvez, A. (2018). Make robots be bats: Specializing robotic swarms to the bat algorithm. *Swarm and Evolutionary Computation*, 44, 113–129.
60. Vilão, C. O., Perico, D. H., Silva, I. J., Homem, T. P., Tonidandel, F., & Bianchi, R. A. (2014). A single camera vision system for a humanoid robot. In *2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol)* (pp. 181–186). Piscataway: IEEE.
61. Gryaznov, N., & Lopota, A. (2015). Computer vision for mobile on-ground robotics *Procedia Engineering*, 100, 1376–1380.
62. Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., et al. (2014). Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics and Automation Magazine*, 21(3), 26–40.
63. Alenyà, G., Foix, S., & Torras, C. (2014). ToF cameras for active vision in robotics. *Sensors and Actuators A: Physical*, 218, 10–22.
64. Fan, Q., Sun, B., Sun, Y., Wu, Y., & Zhuang, X. (2017). Data fusion for indoor mobile robot positioning based on tightly coupled INS/UWB. *The Journal of Navigation*, 70(5), 1079–1097.
65. Sabo, C., Chisholm, R., Petterson, A., & Cope, A. (2017). A lightweight, inexpensive robotic system for insect vision. *Arthropod Structure and Development*, 46(5), 689–702.

66. Wahrmann, D., Hildebrandt, A.-C., Wittmann, R., Sygulla, F., Rixen, D., & Buschmann, T. (2016). Fast object approximation for real-time 3d obstacle avoidance with biped robots. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 38–45). Piscataway: IEEE.
67. McGuire, K., de Croon, G., De Wagter, C., Tuyls, K., & Kappen, H. J. (2017). Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robotics and Automation Letters*, *2*(2), 1070–1076.
68. Li, J.-H., Ho, Y.-S., & Huang, J.-J. (2018). Line tracking with pixy cameras on a wheeled robot prototype. In *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)* (pp. 1–2). Piscataway: IEEE.
69. Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., et al. (2017). Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Robotics Research* (pp. 235–252). Cham: Springer.
70. Starr, J. W., & Lattimer, B. (2017). Evidential sensor fusion of long-wavelength infrared stereo vision and 3D-lidar for rangefinding in fire environments. *Fire Technology*, *53*(6), 1961–1983.
71. Yoo, H. W., Druml, N., Brunner, D., Schwarzl, C., Thurner, T., Hennecke, M., et al. (2018). MEMS-based lidar for autonomous driving. *E & I Elektrotechnik und Informationstechnik* (pp. 1–8).
72. Zhang, J., & Singh, S. (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, *41*(2), 401–416.
73. Kinnell, P., Rymer, T., Hodgson, J., Justham, L., & Jackson, M. (2017). Autonomous metrology for robot mounted 3D vision systems. *CIRP Annals*, *66*(1), 483–486.
74. Šuligoj, F., Šekoranja, B., Švaco, M., & Jerbić, B. (2014). Object tracking with a multiagent robot system and a stereo vision camera. *Procedia Engineering*, *69*, 968–973.
75. Ferreira, M., Costa, P., Rocha, L., & Moreira, A. P. (2016). Stereo-based real-time 6-D of work tool tracking for robot programming by demonstration. *The International Journal of Advanced Manufacturing Technology*, *85*(1–4), 57–69.
76. Pellegrini, S., & Iocchi, L. (2007, Dec) Human posture tracking and classification through stereo vision and 3D model matching. *EURASIP Journal on Image and Video Processing*, *2008*, 476151.
77. Radhakrishnamurthy, H. C., Murugesapandian, P., Ramachandran, N., & Yaacob, S. (2017). Stereo vision system for a bin picking adept robot. *Malaysian Journal of Computer Science*, *20*(1), 91–98.
78. Sergiyenko, O. Y. (2010). Optoelectronic system for mobile robot navigation. *Optoelectronics, Instrumentation and Data Processing*, *46*(5), 414–428.
79. Rodríguez-Quinonez, J. C., Sergiyenko, O., Gonzalez-Navarro, F. F., Basaca-Preciado, L., & Tyrsa, V. (2013). Surface recognition improvement in 3D medical laser scanner using Levenberg–Marquardt method. *Signal Processing*, *93*(2), 378–386.
80. Basaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., Garcia, X., Tyrsa, V. V., Rivas-Lopez, M., et al. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, *54*, 159–169.
81. Sergiyenko, O. Y., Ivanov, M. V., Tyrsa, V., Kartashov, V. M., Rivas-López, M., Hernández-Balbuena, D., et al. (2016). Data transferring model determination in robotic group. *Robotics and Autonomous Systems*, *83*, 251–260.
82. Lindner, L., Sergiyenko, O., Rivas-López, M., Valdez-Salas, B., Rodríguez-Quinonez, J. C., Hernández-Balbuena, D., et al. (2016). Machine vision system for UAV navigation. In *International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)* (pp. 1–6). Piscataway: IEEE.
83. Lindner, L., Sergiyenko, O., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quinonez, J. C., et al. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, *44*(4), 532–541.

84. Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Tyrsa, V., Mercorelli, P., Fuentes, W. F., et al. (2015). Continuous 3D scanning mode using servomotors instead of stepping motors in dynamic laser triangulation. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)* (pp. 944–949). Piscataway: IEEE.
85. Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., et al. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*, *43*(4), 360–369.
86. Sergiyenko, O., Hernandez, W., Tyrsa, V., Cruz, L. F. D., Starostenko, O., & Peña-Cabrera, M. (2009). Remote sensor for spatial measurements by using optical scanning. *Sensors*, *9*(7), 5477–5492.
87. Básaca, L. C., Rodríguez, J., Sergiyenko, O. Y., Tyrsa, V. V., Hernández, W., Hipólito, J. I. N., et al. (2010). Resolution improvement of dynamic triangulation method for 3D vision system in robot navigation task. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society* (pp. 2886–2891). Piscataway: IEEE.
88. Ivanov, M., Lindner, L., Sergiyenko, O., Rodríguez-Quiñonez, J. C., Flores-Fuentes, W., & Rivas-Lopez, M. (2019). Mobile robot path planning using continuous laser scanning. In *Optoelectronics in Machine Vision-Based Theories and Applications* (pp. 338–372). Hershey: IGI Global.
89. Garcia-Cruz, X., Sergiyenko, O. Y., Tyrsa, V., Rivas-Lopez, M., Hernandez-Balbuena, D., Rodríguez-Quiñonez, J., et al. (2014). Optimization of 3D laser scanning speed by use of combined variable step. *Optics and Lasers in Engineering*, *54*, 141–151.
90. Vincent, R., Morisset, B., Agno, A., Eriksen, M., & Ortiz, C. (2008). Centibots: Large-scale autonomous robotic search and rescue experiment. In *2nd International Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems*.
91. Grymin, D. J., Neas, C. B., & Farhood, M. (2014). A hierarchical approach for primitive-based motion planning and control of autonomous vehicles. *Robotics and Autonomous Systems*, *62*(2), 214–228.
92. Kovács, B., Szayer, G., Tajti, F., Burdelis, M., & Korondi, P. (2016). A novel potential field method for path planning of mobile robots by adapting animal motion attributes. *Robotics and Autonomous Systems*, *82*, 24–34.
93. Ali, A. A., Rashid, A. T., Frasca, M., & Fortuna, L. (2016). An algorithm for multi-robot collision-free navigation based on shortest distance. *Robotics and Autonomous Systems*, *75*, 119–128.
94. Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., et al. (2014). Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, *96*, 59–69.
95. Kawabata, K., Ma, L., Xue, J., Zhu, C., & Zheng, N. (2015). A path generation for automated vehicle based on Bézier curve and via-points. *Robotics and Autonomous Systems*, *74*, 243–252.
96. Han, L., Yashiro, H., Nejad, H. T. N., Do, Q. H., & Mita, S. (2010, June). Bézier curve based path planning for autonomous vehicle in urban environment. In *2010 IEEE Intelligent Vehicles Symposium* (pp. 1036–1042).
97. Lugo-Cárdenas, I., Flores, G., Salazar, S., & Lozano, R. (2014). Dubins path generation for a fixed wing UAV. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 339–346). Piscataway: IEEE.
98. Karapetyan, N., Moulton, J., Lewis, J. S., Li, A. Q., O’Kane, J. M., & Rekleitis, I. (2018). Multi-robot Dubins coverage with autonomous surface vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2373–2379). Piscataway: IEEE.
99. Jha, B., Turetsky, V., & Shima, T. (2018). Robust path tracking by a Dubins ground vehicle. *IEEE Transactions on Control Systems Technology*, *99*, 1–8.
100. Wang, Z., Liu, L., Long, T., Yu, C., & Kou, J. (2014). Enhanced sparse a* search for UAV path planning using Dubins path estimation. In *2014 33rd Chinese Control Conference (CCC)* (pp. 738–742). Piscataway: IEEE.

101. Braem, B., Latre, B., Moerman, I., Blondia, C., & Demeester, P. (2006). The wireless autonomous spanning tree protocol for multihop wireless body area networks. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services* (pp. 1–8). Piscataway: IEEE.
102. Fedyk, D., Ashwood-Smith, P., Allan, D., Bragg, A., & Unbehagen, P. (2012). IS-IS extensions supporting IEEE 802.1aq shortest path bridging. Technical Report.
103. Nguyen, H. T., & Walker, E. A. (2005). *A first course in fuzzy logic*. Boca Raton: CRC Press.
104. Duarte, M., Silva, F., Rodrigues, T., Oliveira, S. M., & Christensen, A. L. (2014). Jbotevolver: A versatile simulation platform for evolutionary robotics. In *Proceedings of the 14th International Conference on the Synthesis & Simulation of Living Systems*. MIT Press, Cambridge, MA (pp. 210–211). Citeseer.
105. Browning, B., & Tryzelaar, E. (2003). Übersim: A multi-robot simulator for robot soccer. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 948–949). New York, ACM.
106. Zhibao, S., Haojie, Z., & Sen, Z. (2017). A robotic simulation system combined USARSIM and RCS library. In *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)* (pp. 240–243). New York, IEEE.
107. Klein, J., & Spector, L. (2009). 3D multi-agent simulations in the breve simulation environment. In *Artificial Life Models in Software* (pp. 79–106). New York: Springer.
108. Rohmer, E., Singh, S. P., & Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1321–1326). Piscataway: IEEE.
109. Michel, O. (2004). Cyberbotics Ltd. webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1), 5.
110. Furrer, F., Burri, M., Achtelik, M., & Siegwart, R. (2016). Rotors' a modular gazebo MAV simulator framework. In *Robot operating system (ROS)* (pp. 595–625). Berlin: Springer.
111. Pincirolì, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2011). Argos: A modular, multi-engine simulator for heterogeneous swarm robotics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5027–5034). Piscataway: IEEE.
112. Aşık, O., & Akın, H. L. (2013). Solving multi-agent decision problems modeled as Dec-POMDP: A robot soccer case study. In *RoboCup 2012: Robot Soccer World Cup XVI* (pp. 130–140). Berlin: Springer.
113. Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., & Chen, B. (2010). A new method of virtual reality based on Unity3D. In *2010 18th International Conference on Geoinformatics* (pp. 1–5). IEEE.
114. Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3), 19.

Chapter 14

Real-Time Egocentric Navigation Using 3D Sensing



Justin S. Smith, Shiyu Feng, Fanzhe Lyu, and Patricio A. Vela

Acronyms

AB	Aborted
AD*	Anytime D*
API	Application Programming Interface
BC	Bumper Collision
BM	Block Matching
C-Space	Configuration Space
CVM	Curvature Velocity Method
D*	Dynamic A*
DLP	DWA Local Planner
DWA	Dynamic Window Approach
EB	Elastic Bands
FCL	Flexible Collision Library
FOV	Field of View
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
GVD	Generalized Voronoi Diagram
k-d tree	K-Dimensional Tree
LCM	Lane Curvature Method
LIDAR	Light Detection and Ranging
MAD	Mean of Absolute Differences

J. S. Smith · F. Lyu · P. A. Vela (✉)
Georgia Tech, School of Electrical and Computer Engineering, Atlanta, GA, USA
e-mail: jssmith@gatech.edu; fanzhe@gatech.edu; pvela@gatech.edu

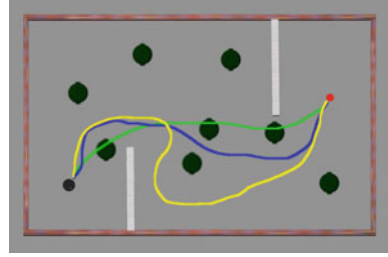
S. Feng
Georgia Tech, School of Mechanical Engineering, Atlanta, GA, USA
e-mail: shiyufeng@gatech.edu

NCC	Normalized Cross-Correlation
ND	Nearness Diagram
PCL	Point Cloud Library
PiPS	Planning in Perception Space
PRM	Probabilistic Road Map
ROS	Robot Operating System
RRT	Rapidly Exploring Random Tree
SAD	Sum of Absolute Differences
SE(2)	Special Euclidean Group for 2-Dimensional Space
SE(3)	Special Euclidean Group for 3-Dimensional Space
SGBM	Semi-Global Block Matching
SSD	Sum of Squared Differences
TEB	Timed-Elastic-Bands
TO	Time-Out
ToF	Time-of-Flight
VFH	Vector Field Histogram
VPH	Vector Polar Histogram

14.1 Introduction

Navigation is an essential computational component of autonomous mobile robots, ensuring that the robot gets from one point in space to another. When deployed, it is commonly implemented as a two time-scale solution involving planning with real-time constraints on decision-making. The longer (or slower) time-scale process aims to find a global path from the current robot location to the desired terminal location and relies on a valid or sufficiently complete map of the world to navigate. The shorter (or faster) time-scale process maneuvers the robot within the world avoiding obstacles while striving to follow the global path. It is common for the global path to be approximately realizable due to uncertainty in the global map. Uncertainty arises from missing map data or modified world structure due to moved, introduced, or removed objects. In dynamic environments, the maneuvering should avoid other moving objects, which would typically be people, animals, cars, or robots, depending on the application. The faster process detects and adjusts the trajectory in response to deviations between locally sensed world geometry and presumed world map geometry, especially when the global trajectory violates the collision-free trajectory constraints based on the local sensor data. Given the domain and source of the map data used, the slower and faster planning processes are often called the *global* and *local* planners. A navigation system modulates and coordinates the information flow between the two planning processes. Figure 14.1 depicts a sample scenario with the robot starting on the left (black dot) and tasked to move to the right (red dot). The global path planner generates a candidate trajectory at the onset (solid green) based on available map information. In this case, only the walls are known. The local planner strives to match the global path, but must detour as unknown obstacles

Fig. 14.1 Depiction of the complementary roles of the global planner (green trajectory) and local planners (blue and yellow trajectories) in the context of navigation



are identified. Depicted are two paths, one of which is realizable by a thinner robot (solid blue), and the other which is realizable by a wider robot (solid yellow). The wider robot would not be aware of the infeasibility of the plan until arriving at the impasse. A global re-plan would then provide a new, potentially feasible path based on the additional map information gained during navigation. This new path would be followed and modified as potential collision points are identified and avoided.

Important considerations associated with a navigation system include the rate at which the global path is recomputed based on locally sensed information or state triggers, what information is shared between the two processes, what world representation is chosen, and what planning strategies are employed for the global and local planners. This chapter summarizes the research and findings related to the global planner and techniques for the local planner. For the latter, the discussion focuses on developments related to the contemporary availability of dense visual measurements of the 3D world structure as compared to the predominantly 2D laser scan measurements used by classical planning algorithms and employed in many local planning methods. The use of image-based measurements that provide depth or range information increases the cardinality of the data to process by two orders of magnitude or more. Classical laser scanning methods do not translate to or do not scale well with the increased sensory data when going from planar to spatial measurements (2D to 3D). This chapter describes alternative data representation, trajectory scoring, and collision detection schemes that improve on the weaknesses of classical methods while striving to be as compatible as possible with them. In doing so, we anticipate that many classical and modern navigation methods can be modified to work with modern dense imaging systems that provide depth or range information.

The chapter is organized to first cover global planning strategies (Sect. 14.2), followed by local planning methods (Sect. 14.3). The review of local planning methods discusses implementation modifications due to advances in robot sensing from laser scanners to dense 3D range-based sensors. Purely monocular, color camera sensing is excluded due to the loss of depth information and the inability of these methods to guarantee collision-avoidance when traveling within the field of view. Findings from neuroscience covered in Sect. 14.4 motivate a mixed method solution, whereby the global world representation and the local world representations differ. In particular, the local representation minimally processes the sensory data and operates in Marr's mid-level visual representation. The result is a perception space

local planner whose design and integration with a global planner is described in Sect. 14.5 and contrasted to existing strategies described in the literature. Prior to implementing and evaluating the PiPS approach, Sect. 14.6 describes a navigation benchmark for evaluating mobile robot navigation algorithms. The experimental outcomes in Sect. 14.7 cover range-based navigation strategies using two mobile robot platforms and multiple environments. Experimental Monte-Carlo runs quantify the success of these methods relative to traditional approaches. Finally, Sect. 14.8 summarizes the chapter contents and provides concluding remarks.

14.2 Global Planning

Planning a robot trajectory, or synthesizing a control signal whose application leads to a robot trajectory, requires a representation of the world within the computer proper [1–3]. Initially, representation strategies may be split into those that attempt to preserve the underlying continuity or those that discretize the world for easier processing. The latter approach is most often taken on account of simplifications afforded by discretization.

14.2.1 Planning in Discrete Spaces

Prevalent discretizations involve conversion of the world into a grid structure or a graph structure. Occupancy maps or occupancy grids are one common grid-based data structure. Ultimately, however, planning on grid-like structures gets reduced to planning on the equivalent graph structure with spatially determined connectivity (e.g., neighbor connectivity). One popular discrete planner is A* [4], a heuristic-informed extension of Dijkstra’s algorithm [5] balancing depth versus breadth based search. If any part of the environment changes, the search must start over. The desire to admit re-planning from a previous invocation led to Dynamic A* (also known as D*) [6], which reduces the cost of replanning by performing local modifications of past searches. D* Lite [7] is a simpler and more efficient version, while Anytime D* (AD*) [8] is both anytime and incremental. Figure 14.2a depicts a grid-based planning scheme with the associated Manhattan world path, which uses only the 4 neighbors (up, down, left, right). Grid-based methods are constrained by the curse of dimensionality, whereby the search space grows exponentially with its dimension. Resolving fine details in a world requires exponentially more memory and results in longer planning times than using a coarser representation. The granularity of the grid implicitly discretizes obstacle and robot dimensions, which may render feasible trajectories infeasible. Adaptive gridding structures and graphical processing unit (GPU) computation overcome the limitations of fixed grid world models and speed up planning time [9].

An alternative strategy to gridded world methods is to employ a sparse graph structure [10]. One such representation is the Voronoi graph of a world. The

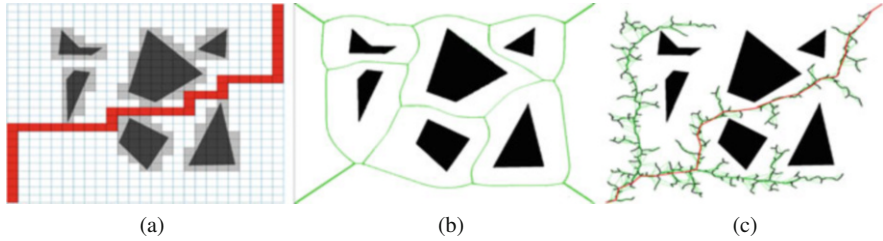


Fig. 14.2 Discretization schemes and plans generated from them. For the grid-based method (left), the plan follows connected discretized volume elements marked as empty. For the Voronoi method (middle), the plan connects to the nearest graph point, traverses the graph, then connects to the terminal point at its nearest graph point. The RRT (right) is like the Voronoi, but with the graph generated through a sampling process. **(a)** Grid world and path. **(b)** Voronoi graph world. **(c)** RRT graph with path

generalized Voronoi diagram (GVD) is a representation of free space consisting of the locus of points equidistant to obstacle boundaries, see Fig. 14.2b. Graph search algorithms and heuristics can be used to find the shortest path through the GVD [11]. Given that the calculation of a GVD may be computationally prohibitive, an alternative graph creation method is probabilistic road maps (PRM). The PRM algorithm constructs a graph of feasible paths during a preprocessing phase and uses this graph to find paths between desired points. It is probabilistically complete with established performance bounds [2, 12]. Lazy PRM decreases the runtime of the algorithm by only performing collision checks when searching for the shortest solution to a query [13], making the planner more suitable for single queries. PRMs have been extended for efficient replanning in dynamic environments by combining concepts from Lazy PRM and AD* [14, 15].

The PRM is an example of a sample-based planner where the sampling is done in advance. Instead of spending time generating the PRM, rapidly exploring random trees (RRT) are likewise probabilistically complete but perform minimal exploration, encouraging fast and efficient planning [16]. These properties are achieved by combining graph creation with plan search by building out the graph in a goal-directed manner through a combination of randomized sampling and heuristic or greedy (go-to-goal) sampling. An RRT planning instance is depicted in Fig. 14.2c. Extensions to the basic RRT method include bi-directional search [17] and continued search to improve optimality [18]. Usually path smoothing is needed after identifying a solution due to the potential jaggedness of paths obtained from random steps. RRTs have been used for the path planning of autonomous vehicles [19]. They have also been extended for more efficient replanning in dynamic environments [20–22]. RRT-X is a recent extension that maintains and updates a single search graph throughout navigation and makes no distinction between local and global planning [23].

Motion planners can also be based on motion primitives. Sample-based planners that incorporate dynamic constraints include kino-dynamic planners, whereby connected nodes are reachable through a feasible control or constraint satisfying trajectory [24–27]. In [28], such a planner permits aircraft to navigate through a

dense forest. It uses one primitive to link any two points in space through constant control inputs and another to perform an aggressive turn. Since the primitives account for the dynamics of the aircraft, the planner only has to find a sequence of points for the aircraft to travel through and then use the appropriate primitives.

14.2.2 Planning in Continuous Spaces

Rather than converting the spatial representation of the world into a graph then inducing the same representation on the sought trajectories, one may seek to preserve the continuous structure of trajectories. If the control dynamics are ignored, then these strategies seek out continuous curves connecting the start and goal points. The simplest of such approaches is the *potential field method*, which employs potential functions to define a gradient-based vector field [29]. Following the potential field gradient as a differential equation yields the trajectory to follow. Since potential field methods have problems with local minima, extensions exist to arrive at formulations whose solutions are true global minima or have no local minima [30–32]. The same conversion applies to fast marching methods on a grid world [33, 34], which provide a more continuous trajectory versus the equivalent Dijkstra implementation over the underlying grid.

More complex approaches seek an actual trajectory by optimizing over the trajectory function space. The infinite-dimensional nature of trajectories requires finite-dimensional, parametric representations of curves to be used [35, 36]. If the actual control signals are also sought, then the problem becomes an optimal control problem, for which there are many solution strategies [37–42]. The main drawback to these methods is the computational cost associated with finding a feasible trajectory, especially when there are many obstacles. For computational efficiency, these methods are often iterative or gradient based in nature. They benefit from pre-conditioning the initial condition via a discrete planning method. If the overall problem has not varied significantly from the previous invocation, then the previous solution serves as a good initialization for iterative solvers [40].

A major issue for all global planners is the speed with which sensed data can be assimilated into the global map. For the slow global planners, global map updates are not the main bottleneck. However, for global planners that assert real-time operation, usually the cost to transfer sensor data to the internal representation and to update any important map data structures or cost functions far exceeds the plan update rate (by 1 to 3 orders of magnitude). Local planners exist to speed up this process by limiting what data is considered in the fast path updates. Following rapidly updated local plans gives the global planner time to generate an update based on the newly integrated information.

14.3 Local Planning

The need to augment global planning with local planning follows from the inability of global planners to re-plan at a rate compatible with the incoming sensory data and the underlying trajectory tracking controller. The purpose of the local planner is to generate real-time updates to the global plan, usually at the control feedback rate or close to it, in order to compensate for map errors, new objects, or moving objects. A common approach is to employ a hierarchical planner, in which the global and local levels use the same world representation though not necessarily the same data structure. Doing so means that a single method may serve two roles, however it requires that the conversion of sensory data to the world model occur faster than the sensory rate.

Early research on navigation primarily emphasized ultrasonic and laser scan methods since these sensors provided direct measurements of the external environment, as opposed to vision-based methods which required costly, at the time, processing to convert the raw visual signals into spatial data regarding the local environment. The dense 1D measurement signal provided from a laser scanner is sufficient for navigating through structured worlds where the scan plane provides correct information regarding collisions. With computing and sensory hardware improvements, navigation with dense 2D range or disparity image signals has started to become the norm. However, there are limitations to literally translating laser scan methods to admit dense imagery. Many of the existing solutions cannot scale with the data cardinality. This section covers the above history and topics.

Reactive Methods

Reactive methods use the locally sensed obstacle space and the current goal point to identify an immediately applied control policy. For ground vehicles, the control consists of a speed and steering command, though some methods assume a constant velocity model and simply steer. The potential field method [29] described earlier, due to its gradient following approach, is implementable as a reactive method. The virtual force field method is one such implementation [43]. Sensor readings are used to update a global certainty grid of obstacles which is then used to calculate the repulsive forces of obstacles.

Rather than use point representations in a local Cartesian grid, the vector field histogram (VFH) [44] method uses a local polar (histogram) representation for local points in the global map relative to the current robot pose. Processing of the polar histogram generates new steering commands, with an additional processing step to establish speed changes. The steering commands aim towards the best free-space option consistent with the current goal point. Improvements to the reactive policy include VFH* [45], which performs look-ahead verification using a short horizon forward time search with A*. It hypothesizes future polar histograms from the global map. Integration into a hierarchical planner for navigation was performed with VFH+ as the local policy [46]. The VFH approach was modified to apply to dense laser scanning sensors by processing directly on the sensor data in the vector

polar histogram approach (VPH) [47, 48]; it classifies travel directions into blocked and unblocked prior to determine the steering direction and the speed update.

Due to specific issues that arise from single policy reactive strategies, the nearness diagram navigation (ND) method [49–51] first classifies the polar data into distinct environmental conditions. Each class is assigned a reactive, or sensor-driven feedback, policy for more robust and consistent navigation. The intent is to avoid the local minima and unstable motion sometimes exhibited by single policy reactive approaches. The ND navigation method factors the width of the robot into its reasoning. A hierarchical navigation implementation with global planning exists [52]. ND navigation is one of the early instances of gap-based navigation. Other gap-based approaches include [53–55] as well as [56] which considers the robot shape and kinematic constraints.

Velocity or Control Space Methods

The approaches considered up to now have not taken into account the vehicle's dynamics. Approaches operating in velocity space, in contrast, are able to accommodate nonholonomic kinematic constraints. For example, rather than look at instantaneous angles, or radial directions relative to the robot, the steer angle field [57] algorithm evaluates forward integrated circular paths, discretized by steering angle. These sampled paths get checked for collisions, with the forward speed modulated by the existence—or lack thereof—of feasible paths. The method was integrated with a hierarchical navigation system [58]. Likewise, [59] performs local reactive obstacle avoidance by considering various circular trajectories and selecting one that maximizes the translational velocity, minimizes the angle to the target point, satisfies the robot's dynamics, and avoids collision for at least 2 s. A later implementation called the dynamic window approach (DWA) [60] samples from a discrete set of relative control or velocity changes from the current control or velocity. It operates on a 2D occupancy grid model of the world and is capable of incorporating planar robot geometry into the collision checking. Like VFH, DWA has been extended to incorporate look-ahead searching [61], best suited for dynamic obstacles. DWA methods exist which integrate with a global planner or into a hierarchical navigation strategy [62, 63].

A similar velocity space approach is curvature velocity method (CVM) [64]. In this approach, obstacle avoidance is solved as a constrained optimization in velocity space. This allows speed and heading to be solved simultaneously and constraints to be added easily. These include robot velocity and acceleration as well as any specific application constraints (safety vs speed, etc.). The lane curvature method (LCM) [65] addresses some shortcomings of CVM. It selects a *lane* based on collision-free distance and width and computes a local heading to guide the robot into the lane. CVM is then used to generate the necessary translational and rotational velocities. Since openings are chosen based on the width of the lane rather than angular width of the opening (as in VFH), the paths generated by LCM may be safer.

Optimal Trajectory Synthesis

Bridging the gap between reactive planners and velocity space planners are continuous trajectory synthesis implementations on the local map. On sufficiently

small domains, they can run in real-time when capable of using the grid-based costmaps for cost and constraint evaluation. The elastic bands (EB) [66] planner generates a free-path (the elastic) in C-Space. This elastic deforms based on external repelling forces generated by obstacles as well as internal contracting forces for path length minimization. The EB approach has extensions for handling kinematic constraints [67]. The timed-elastic-bands (TEB) [68] adds dynamic constraints. TEB has also been extended to maintain and optimize multiple candidate trajectories of distinct topologies [69].

14.3.1 Moving to 3D Environment Representation

A 2D environmental representation was sufficient when the primary sources of sensor data were planar, such as when using sonar, fixed (horizontal) laser scanning sensors, or other laser ranging methods [70, 71]. Dense sources of 3D points provide the opportunity to detect and avoid collisions with obstacles that planar sensors might miss, such as obstacles that protrude from the side or hang down from above. As a result, there is a motivation to utilize dense 3D sensor data when it is available. Common 3D dense data sources include time-of-flight (ToF) cameras, depth cameras, LIDAR, and triangulating scanners [72]. Stereo vision sensors also apply by simply generating a dense depth or disparity map.

One approach for utilizing 3D data with classical planning approaches is to project the data down to a 2D representation and run the planners as normal [63, 73], or performing a column-wise min operation over depth images for planar world information [74]. Generally, points are filtered in order to only consider those within the height of the robot. Though easy to implement and computationally cost efficient, there are some downsides to simplifying the environmental representation in this way. Unless the robot's cross-sectional geometry is constant with height, the simplified representation will be overly conservative; valid configurations may be detected as in collision with the environment.

Another approach is to maintain a 3D world model and use this for planning. However, there are several challenges with doing this. First, not all local planning approaches are readily adaptable in this way. For example, gap aiming (directional) approaches [53] are explicitly designed to process a 1D list of measurements (such as from a laser scan). Indeed, only recently have gap aiming approaches explicitly considered non-point, non-holonomic robots [56]. Sampling-based approaches, on the other hand, have an explicit trajectory scoring step which can be modified to perform collision checking against a 3D environmental representation. Such a tactic is taken with discrete steering directions checked against a filtered point cloud model of the world [75]. Likewise [76] compares a fixed set of sample trajectories against a local point cloud. Using stereo the aerial robot detects points at a given range, and propagates them in time for a richer world model. The range-filtered measurements keep the point cloud model sparse enough for real-time operation on lightweight computing platforms.

Another challenge is selecting the environmental representation to use. One of the simplest is the voxel grid (a 3D cartesian occupancy grid), however memory requirements for voxel grids are high due to the curse of dimensionality. Variable resolution structures can significantly reduce the amount of memory required using data structures such as octrees [77]. For aerial vehicles, the volumetric map can be height filtered for more efficient processing [78]. Rather than employ efficient occupancy data structures, efforts have considered alternative point cloud data structures with efficient query times. Points can be stored in sorted data structures such as kd-trees to allow faster nearest-neighbor queries [79]. The method is very efficient when the sensor data is of low cardinality, the local volume is restricted, and the robot model is a point. There is an added computational cost to build and maintain these data structures.

Even if the planning algorithm itself could operate in real time, the process of updating the world representation introduces latency between sensing and planning. An alternative approach is to avoid the reconstruction process and perform planning in an earlier visual representation. For navigation strategies using stereo camera systems, this involves planning directly in disparity space [80–82] or planning using combinations of optical flow and stereo disparity [83, 84]. After gathering dense data from sensors, these approaches explore C-Space trajectory sampling and collision checking in the perception space. Local path options are mapped into the image space for evaluation [85, 86], where the objective is to follow the path generated by a global planner such as A* while avoiding unmodeled obstacles. Perception space approaches improve time performance by reducing the delays associated with populating and maintaining in-memory data structures.

14.4 Neuroscience Research Related to Navigation

The compartmentalized structure of robotic navigation algorithms is likewise reflected in the human brain. Though different from engineered systems, cognitive and behavioral neuroscientists and researchers have identified distinct processing regions and characteristics regarding navigation via controlled experiments [87–93]. The findings suggest that perception for navigation involves both egocentric (or viewer-centric) models for decision-making and object-centric (or world-centric) models, sometimes also referred to as allocentric. The former corresponds to visual understanding relative to the human's reference frame, whereas the latter is relative to an external reference frame usually belonging to an object receiving attention. Similar structural differences in computation have been proposed by Marr when describing a representational framework for vision [94] and its computational aspects. In particular, from low-level to high-level, he described four primary components as depicted in Fig. 14.3. The first is the raw information associated with the sensed scene, the input image. The second, called the primal sketch, consists of basic signal processing transformations of this image that extract geometric volumetric primitives

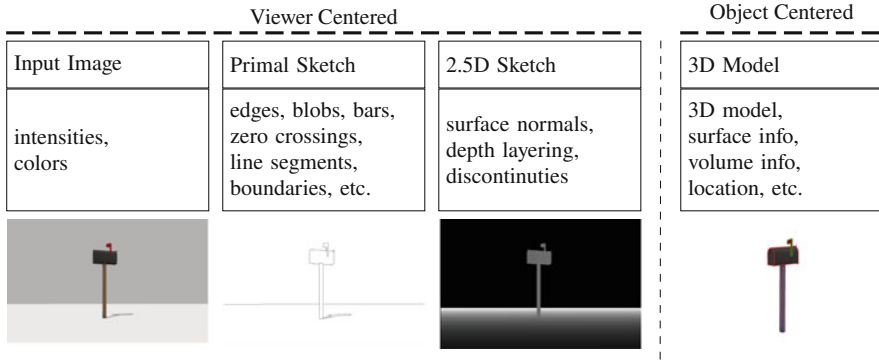


Fig. 14.3 Marr’s representational framework for visual processing. Of the four stages depicted, three are in viewer centered representations, with the fourth in an object centered representation

from the image proper (lines, areas, blobs, etc.). The third, called the 2.5D sketch, reflects higher level interpretation and analysis of the image and its low-level features. The representation of information is still at the image level, however depth data or depth-dependent information exists as a layered representation. The layered images contain information about the distal ordering of the scene relative to the viewer. Represented internally using image-based data structures, these three initial levels are viewer-centric. The last layer reconstructs or generates a richer 3D model of the objects sensed within the scene. At this level the reference frame or viewpoint shifts away from the viewer and to a global reference frame or an object-centered reference frame. In this manner, the reconstructed model is independent of the user’s view and may persist as it changes in response to relative motion between the viewer and the object. Naturally, missing from this description is the agent’s memory of past scenes or locations. However, it is sensible to imagine that both egocentric and allocentric memory and predictive models exist [87, 88].

Though distinct regions and reference frame processing paradigms have been discovered, there is still coupling between them. The research suggests that early computation may primarily rely on egocentric models with some influence from allocentric models [90, 91]. Likewise, information related to egocentric navigation appears to be important for detecting or estimating important allocentric states or collision informing properties [87, 95]. Thus, within the neural processing hierarchy, egocentric processing happens earlier than allocentric processing but can be influenced by earlier allocentric outcomes expected to persist into the near future. With regard to objects, Marr’s framework predicts that object information would primarily be encoded allocentrically. However, since evidence for an object arises from lower level evidence, we should expect that some information regarding objects exists in an egocentric representation. There is indeed evidence that egocentric representations are used for object locations [96] as well as for navigation goal states [97]. There is also support for brain regions that translate between the two representations, such as from egocentric to allocentric representations (ex: what is

my current position on this map given my current view) and vice versa (ex: in order to head west, I need to turn left) [98]. Rat studies have shown that the same stimuli will affect different regions based on their egocentric or allocentric properties [99].

The research suggests that both representational forms for modeling the world should exist within a navigation pipeline. In particular, methods employing only world-centric models at all levels of the navigation hierarchy may be misguided. Yet many navigation frameworks are characterized by this property. Instead, viewer-centric models should be incorporated and prioritized within components requiring fast decision-making and low latency from the sensory input to the controlled response. Additionally, world-centric models should be prioritized for slower decision-making and reasoning that operates beyond the local frame or requires an external frame to simplify processing. Furthermore, the two processes should be coupled at a rate consistent with navigational decision-making.

14.5 PiPS: An Ego-Centric Navigation Framework

This section introduces and expands upon a local planning method employing perception space (PiPS) [100]. In relation to Marr's visual framework, PiPS operates in the mid-level 2.5D representation whereas it is assumed that the global planning method will operate in the world representation. This section first covers the structure of the navigation system, including general details on how the global and local planners integrate. It then covers the collision checking process of PiPS and compares the time costs of various data structures used for collision checking against local 3D environment models. Extensions to PiPS for integration into a hierarchical navigation system are covered in the remaining sections. In particular, the original memoryless PiPS is augmented both with memory and a local cost-map component. These augmented data structures are called the egocylinder and egocircle. They provide means to perform collision checking and trajectory scoring, respectively. The final integrated, PiPS-based navigation system results in a mixed representation navigation scheme with linear scaling properties in the data cardinality and real-time obstacle avoidance properties. Navigation simulations and visualizations are created using Gazebo and RViz.

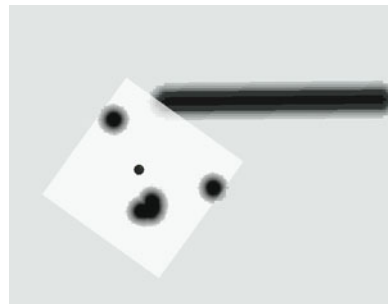
Navigation Using Move Base

The ROS Move Base package provides an API for hierarchical goal directed navigation of planar robots. Its design enables the incorporation of different global and local planners. These are connected by a trajectory synthesis module and several scoring systems that evaluate or influence the final local trajectory chosen.

Move Base, like many other navigation frameworks, operates under the assumption that the global and local planning representations will be the same, e.g., grid-based. As a consequence, the scoring mechanisms fundamentally rely on *costmaps*.

Costmaps are 2D grids populated with travel cost or other scalar value for assigning a cost to each grid cell. These costs inform the Move Base planners. There are several different sources for the costs, with one being the occupancy grid. An occupancy grid keeps track of navigable space and non-navigable space (or occupied space) associated to obstacles. Obstacles are added to the occupancy grid from sensor data such as laser scans or point clouds. Grid points between the sensor location and detected obstacles are marked as unoccupied via ray tracing. To factor in the robot's radius, inflating each occupied cell through dilation techniques generates an inflated occupancy grid (usually by the smaller radius if the robot is not circular). Generating the obstacle distance map involves assigning each unoccupied cell the distance to the nearest occupied grid cell. This obstacle distance map informs the creation of path scoring costmaps. There is a global costmap used for global planning. It covers the entire region of known space within which planning occurs. If a prior model for the environment is available, it can be used to initialize the global costmap. Otherwise, the entire costmap is initialized as unknown and considered traversable for the purposes of global planning. In addition, there is a local occupancy grid that tracks occupancy information in a robot centered local Cartesian grid whose orientation is fixed relative to the initial orientation of the robot. Figure 14.4 depicts such a local map where the initial robot orientation differs from the current robot orientation and the world orientation, hence the rotated projection of the local map onto the world. The local nature of the grid means that costmaps generated from the local occupancy grid are only defined for a fixed Cartesian domain centered on the origin. The domain dimensions are set to be comparable in size to the sensing region of the robot (though they could be set smaller if desired). The local costmap is key to local planning and must update faster than the sensor rate to admit the real-time synthesis of new local paths. Figure 14.5 depicts key costmaps associated with local planning. Low cost regions are red, while high cost regions tend to blue/purple. The obstacle cost penalizes robot trajectories that pass too close to obstacles. The local goal cost penalizes sample robot pose locations based on the minimum travel distance to the local goal point, which is the point of the global trajectory (marked as a black curve in the costmaps) at the boundary of the local map. Basically, it penalizes trajectories for ending away from the local goal. The path cost penalizes robot pose points based on their distance to the nearest global path point. Higher

Fig. 14.4 Mobile robot navigation with visualization of the local map as a square centered on the robot



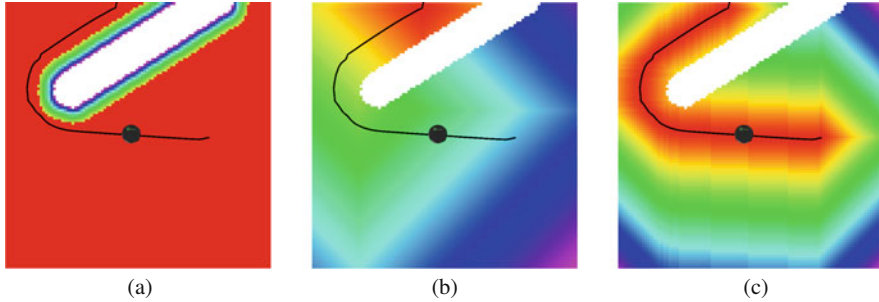


Fig. 14.5 Visualization of the trajectory scoring as cost maps computed on an occupancy grid. Low cost is red and high cost is blue/purple. The black curve is the global path to follow. (a) Obstacle cost. (b) Local goal cost. (c) Path cost

costs are associated to trajectories straying from the global path. The time to populate these costs from the occupancy grid varies from $O(n)$ to $O(n \log n)$ where n is the grid area [33, 101].

The global planner in Move Base uses a variant of Dijkstra’s algorithm to find a path from the robot’s current pose to the specified goal, while the local planner generates velocity commands to direct the robot along this path. The default local planner provided with Move Base is the DWA local planner (DLP), which invokes DWA [60] to sample velocity commands. Sampled velocities are forward simulated to create trajectories. Each trajectory is scored based on a weighted sum of costs for the described cost functions: the obstacle cost, the goal cost, and the path cost. The local and global planners run at specified frequencies. If the local planner fails to find a valid velocity command, global replanning is triggered. If the local or global planner fails for longer than its specified time limit, a recovery behavior rotates the robot in an effort to clear obstacles from the local costmap. If all recovery behaviors have run and the planner is still unsuccessful, navigation aborts.

PiPS Modifications to Move Base

The principal components of Move Base are depicted in the data flow diagram of Fig. 14.6. Not depicted is the robot pose information generated by the odometry or localization process, which is required to propagate forward in time past measurements. This estimated pose update will be denoted g_{move} . The traditional DLP implementation involves the blue blocks, the black blocks, and the dashed block. We will modify the local planner to employ an egocentric representation for the world based on distal information of obstacles and global path points and maintained in a 2D or 1D array, thereby operating in perception space rather than in world space. These are the red blocks labeled Ego-Circle and Ego-Cylinder, which correspond to the propagation of perceptual memory regarding the local egocentric representation of occupied space. Additional important modules include collision checking and path scoring, which are both parts of the local planner. Employing a

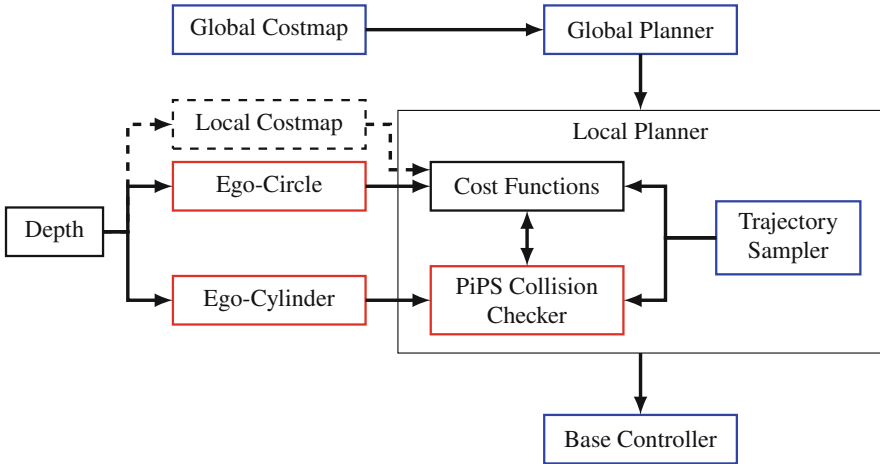


Fig. 14.6 Block diagram depicting processing and data flow for the PiPS-based navigation strategy. Design considerations for the critical components are covered in this section. The blue components are unchanged, the red components are PiPS enhancements. The dashed component is only used by the traditional Move Base pipeline. The “Cost Functions” block is also modified for PiPS

mixed representation requires extensive modification of the existing scoring methods, especially those linking the local path to the global path.

14.5.1 Collision Checking in Perception Space

Rather than map the sensor data regarding the local world into a world centric representation, PiPS keeps the sensor data in a viewer centric representation. Collision checking requires mapping the robot model into the same viewer centric representation and comparing the robot model’s distal properties to those of the surrounding environment. Thus, rather than evaluating collisions using 2D or 3D world maps, we evaluate collisions using 2D images. Collisions occur when the robot model maps to depth layers that lie further away from sensed world depth layers in a given region of space. In the opposite case, if the robot geometry of a test pose maps to image regions whose obstacle depth layers are further away than the robot depth layers, then the test pose is considered safe. In this manner, PiPS switches the main collision-checking computations from relying on the transformation of sensory data to world data for collision checking of the robot in the world, to relying on the transformation of the robot world model to the sensor representation for collision checking in image-space.

Mapping of the robot to the sensor representation requires modifying the graphics z -buffer rendering pipeline. To render an object model, the traditional approach to graphics considers all object points that project to the image then chooses at each

pixel only the closest point projecting to it as the measurement source. The remaining points that project to the camera at that pixel are ignored. This is a closest point model. Given closed mesh models of all objects in the world, the set of projection points that may realistically project to the camera are those whose outward normal to the mesh points toward the camera. For collision checking we wish to synthesize not the closest point to the camera, but the furthest point from the camera (with a normal vector pointing in agreement with the camera optical axis vector).

Given a surface mesh model of the robot, synthesis of the collision depth image of a hallucinated robot in an empty world will select the furthest points. Let the set $\{(q^i, \hat{n}^i)\}$ of robot points with associated outward surface normals be those that project to the camera at pixel r . Then the chosen point to be used for the synthesized depth image is the one with the index

$$i^* = \arg \max_i D(q^i) \quad \text{subject to} \quad \begin{bmatrix} \hat{n}^i \\ 0 \end{bmatrix} \cdot q^i > 0 \wedge z > 0, \quad (14.1)$$

where $D : \mathbb{R}^3 \rightarrow \mathbb{R}^+$ maps points to depths. The depth of point q^{i^*} is placed at pixel r . All points in the image of the hallucinated robot that do not have a projected point are simply set to 0.

As the hallucinated robot pose $g \in SE(3)$ changes, the collection of surface points and normals will change under g , leading to different depth images. This robot pose is in collision with actual objects in the real world if the sensor image D_m has a value less than that of the hallucinated robot image $D_H = D \circ g(\mathcal{M}^R)$, where \mathcal{M}^R is the mesh model of the robot (simplified to a set of points and outward normals). Simply put, a hallucinated robot pose is collision-free if

$$D_m(r) > D_H(r) \quad \forall r \in \mathcal{I}, \quad (14.2)$$

where \mathcal{I} is the image coordinate domain. A collision-free, hallucinated robot pose is called a *safe* pose.

Figure 14.7 depicts the process just described. Viewing from left to right, the top row consists of the world view for a pose testing scenario. The real robot (black) contemplates itself at a future trajectory pose (red) that is not safe. Collision checking is not performed with the more complex true model, but rather with a cylindrical simplification (red cylinder in second column). For collision purposes in depth space, only the far sides of the robot matter (cyan surface in third column). A portion of the robot collides with walls in the real world (yellow surface in fourth column). The blue volume depicts the projection cone associated to the tested robot pose. Image depth values indicating world point within this cone indicate a robot collision at the hallucinated robot pose. PiPS does not perform the collision checking from the world (or third person perspective). Rather it performs the collision checking in the viewer-centric perception space (or from the first person perspective) as depicted in the second row. From left to right, the first person views are visualized. The right-most column actually involves the overlay of the two depth images, as can be seen

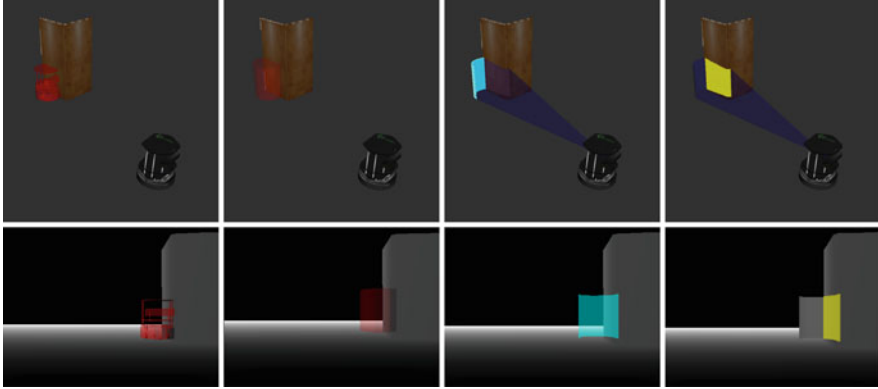


Fig. 14.7 Each column illustrates a conceptual step of PiPS. The top image depicts the scene from a third person perspective, while the bottom image depicts it from the robot’s first person perspective. From left to right: hallucinate the robot at a pose (red); Replace the robot with a simplified geometric representation (red); Find the far surface of the robot (light blue); Detect collisions (yellow)

by the gray depth values corresponding to the robots far cylindrical surface patch. The yellow region contains robot depth values that are greater than the measurement depth image values.

Efficiency of Perception-Space Collision Checking

The PiPS local planner in [100] sampled from a pre-determined set of trajectories that remained in the field of view relative to the robot’s current perspective, then selected the longest non-colliding trajectory. It was capable of real-time operation on embedded processors with the computational power of circa 2014 cell phones, thereby demonstrating favorable processing properties. It can run in real-time on today’s embedded system-on-a-chip processors, such as those using the latest Arm Cortex chips. Here, we explore more deeply the computational cost of PiPS. The asserted value of a perception space approach to collision checking is that conversion of the sensory data to alternative world representations incurs a time cost that determines the minimal response latency of the local navigation policy. This section performs a collision check time cost comparison for several world model data structures and collision query strategies. The data structures chosen for comparison are depth image (PiPS), point cloud, octree, and k-d tree. With reference to Fig. 14.6, the latter data structures would require swapping out the “PiPS Collision Checker” block with the appropriate collision checking implementation, and would involve programming alternatives to the Ego-Circle and Ego-Cylinder blocks. The first evaluation metric of interest is the time cost of initializing the data structure from a new sensor input. This value is significant because it represents the minimum latency between receiving environmental information and being able to act on it. The second evaluation metric is the time cost of performing a collision check with a given robot pose. Together, these impact the time cost of collision-checking a single trajectory or a set of trajectories. The timing experiments were performed on an

Intel Xeon E5-2640 @ 2.50 GHz processor with single-thread Passmark score of 1468 and multi-threaded score of 9512. All reported timings are for single-threaded collision checks.

The investigated data structures require different amounts of processing to initialize from sensor data. For the PiPS depth image process, the only processing performed is transposition of the incoming depth image for more efficient pixel-wise comparisons during collision checking. The point cloud approach requires converting the incoming point cloud sensor message to a point cloud library (PCL) [102] data structure. The k-d tree and octree approaches also require the input depth image to be converted to a point cloud for populating the data structures. The k-d tree implementation is from the Point Cloud Library (PCL) [102]. The implementation used for the octree is octomap [77]. An octree with 5 cm resolution is created from a point cloud using code derived from the Octomap Server ROS package [103].

The time required to perform the preparatory conversions is listed in Table 14.1. Conversions are performed using nodelets from the *depthimage_to_laserscan*, *depth_image_proc*, and *image_proc* packages. Depth images from a ROS/Gazebo simulated environment pass through the Decimation nodelet and on to the PointCloud and LaserScan nodelets. Custom timing nodelets record the time required for conversion and are enabled for only one conversion nodelet at a time. Measurements are collected until the average value is stable to at least two significant digits.

Across the board, the depth image decimation time remains lower than the other conversion strategies. The non-zero cost of the 640×480 decimation approach represents the overhead of passing an image on without any processing.

We also measured the average time required to populate each data structure from new sensor inputs of several different resolutions and report the results in Table 14.2. The per-frame and per-pose computation cost is calculated using a collection of 791 depth images captured while the robot wandered through a ROS/Gazebo simulated environment. A set of 200 poses, representing a set of trajectories, was also generated. In order to evaluate how computational requirements scale with the size of sensor data, each set of tests is repeated with the depth image decimated to the following sizes 640×480 , 640×240 , 320×240 , 320×120 , 160×120 . All of the collision

Table 14.1 Average times (in ms) for data preparation/conversion to necessary input format

Resolution	640×480	320×240	160×120
Depth image to PointCloud	2.2	0.61	0.21
Depth image to LaserScan	0.66	0.35	0.22
Decimate	0.084	0.31	0.20

Table 14.2 Average times for initializing data structures from an input (in ms)

Resolution	640×480	320×240	160×120
Depth image	0.64	0.14	0.048
Point cloud	6.67	2.74	0.59
K-d tree	34.0	7.8	3.7
Octree	245	88	42.7

checkers are tested using the exact same series of sensor data and candidate poses. The collision checkers are evaluated offline one after the other, each testing all poses on all images.

The table demonstrates that the data structures have poor scaling properties when increasing the amount of data to process. Real-time operation requires significant decimation of the input data, thereby losing important structural knowledge of the local environment. Though the decimation enables real-time processing for small compute platforms as would be deployed on weight restricted robots, like quadcopters, it does not scale well for mobile robots without these restrictions. At full resolution, the time cost of conversion can exceed the data arrival times for sensors operating at typical frame rates (presumed to be around 30 Hz).

The next performance metric is the average time required to collision check a single candidate pose of a robot. All else being equal, a smaller value will allow more poses to be tested in a given time frame. For the purposes of these tests, we assume a cylindrical robot. Depth image collision checking uses the approach described in Sect. 14.5.1. The point cloud approach naively iterates through the points in the point cloud, checking if any lie inside the specified cylindrical region. The k-d tree approach first queries the tree for any points lying within a sphere bounding the candidate robot pose cylinder and then checks if any of these are within the cylinder. The octree approach uses the flexible collision library (FCL) [104] to check for collisions between the populated octomap and a cylinder. We measure the average times required to collision check a candidate robot pose for several candidate resolutions and report the results in Table 14.3. The table demonstrates why some researchers choose to employ specialized data structures, as their collision checking time cost can be quite low and nearly constant versus image resolution. Once the data structure has been populated, collision checking is the lowest cost and practically negligible in comparison with the preparation time. The PiPS depth image approach, on the other hand, has a larger resolution dependent time cost. The value of a PiPS approach lies in the total cost.

The results in the tables indicate that using the depth image approach can result in significant time savings, so long as the number of collisions tested can be kept low enough. While the per-collision check time of the depth image approach is not as low as that of the k-d tree, the initialization time of the depth image data structure is much lower than that of the k-d tree. For example, with 640×480 sensor data, a depth image

Table 14.3 Average times for collision checking a pose (μs)

Resolution	640×480	320×240	160×120
Depth image	55.0	31.6	19.6
Point cloud	882	450	118
K-d tree	8.9	7.4	6.9
Octree	11.4	11.6	11.6

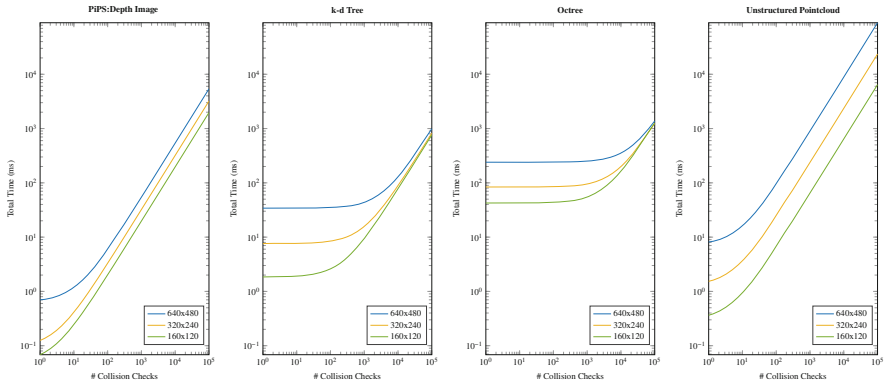


Fig. 14.8 Semilog plots of the total time to initialize data structures with one sensor input and to perform N collision checks. From left to right, the methods tested are depth image (PiPS), kd-tree, octree, and point cloud

approach can initialize its data structure and then perform over 600 collision checks in the time it takes the k-d tree to just initialize. The graphs in Fig. 14.8 compare the total time for initializing the data structure and performing a given number of collision checks across the different approaches. Both axes have log spacing. The value of the k-d tree and octree approaches lies in the near flat curves for less than 100 collision checks, and the relatively low slopes after that. In contrast, the PiPS depth image approach has a mostly linear curve, just like the point cloud approach, albeit with an improved base cost. When the expected number of collision checks is relatively low, the depth image approach is significantly faster. Identifying where the depth image curves cross a comparison strategy indicates how many collision checks should be performed to achieve an equal compute time. Similar findings should hold for robots with different geometries, however the slopes may differ.

Importantly, during actual deployment, the robot will not be evaluating new trajectories at every frame. Rather, the first step in the process is to test the current local path for feasibility. If it is feasible, then the navigation system continues to drive along the local path. New path sampling, scoring, and testing happen when an obstacle is detected along the current local path or the robot nears the terminal point of the given path. Evaluating the time cost to collision check the current path, it is clear that the PiPS approach has a significant advantage as it can perform this check and determine the feasibility of the current path before any of the other methods can even initialize, even if the comparison is between a full resolution PiPS implementation and most decimation levels for the alternative implementations. The time cost for PiPS to perform this check is 6.2, 3.6, and 1.3 ms across the resolutions evaluated, for the case that the number of robot poses to test in the current trajectory is 100. This time will decrease as the robot moves along the current trajectory.

14.5.2 Egocylindrical Perception Space for Enhanced Awareness

Though navigating with PiPS using trajectories that map into the current sensing image domain is collision free [100], the published approach employs simple straight line trajectories. Except for the floor space immediately in front of the robot, the sampled trajectories map the robot into the depth image. However, visual sensors have a limited field of view when compared to laser scanners (usually around 60–90° horizontally versus 270–330° horizontally), thus restricting trajectories to the field-of-view is quite limiting. DWA and many other local planners sample a richer trajectory space whereby some trajectories leave the visual sensor’s FOV. Not factoring in these trajectory segments for trajectory scoring or collision checking leads to unsafe navigation. Much like a local cost-map accumulates and propagates occupied points, the local PiPS module requires the ability to accumulate, propagate, and retain previously seen perceptual information lying outside of the current FOV. This will be done using the egocylindrical image space [105], another 2.5D image space representation whose theoretical domain extents surround the robot. Due to the nature of camera projection equations, which require positive z -values in the camera frame, traditional depth images cannot retain information of world geometry behind the robot. Furthermore, the homographic projection involved in traditional pinhole models requires an infinite image region to map the forward-facing half-plane to an image. The egocylindrical perception-space representation avoids these modeling degeneracies. World points from sensors are projected onto a virtual cylinder surrounding the robot and are propagated as the robot moves. The surface of the cylinder is discretized into a 2D grid.

Whereas the egocylindrical image in [105] stores stereo disparity values, the egocylindrical image here stores the ranges corresponding to each point on the virtual cylinder. In relation to the previous section, Sect. 14.5.1, the only modification required is on the image domain and the projection equations for rendering the depth image measurement (now as an egocylindrical image). The left side of Fig. 14.9 visualizes the egocylindrical (image or perceptual) representation. The color coding indicates distance of world points from the sensor’s optical origin. The right side provides a third person view of the scenario, simulated using ROS/Gazebo. The simulated sensor in this scenario has a forward facing field of view of 60°.

When the sensor is a depth sensor, then the pixel depth data is based on the ray projecting out from that pixel. Mapping the depth value to a range value requires factoring in the ray information. Using the homogeneous image ray representation with unit z -coordinate, the egocylindrical range is

$$\rho = D_m(r_{im}) \left\| \begin{bmatrix} x_{ray}(r_{im}) \\ y_{ray}(r_{im}) \\ 1 \end{bmatrix} \right\| = D_m(r_{im}) \rho_{ray}(r_{im}), \quad (14.3)$$

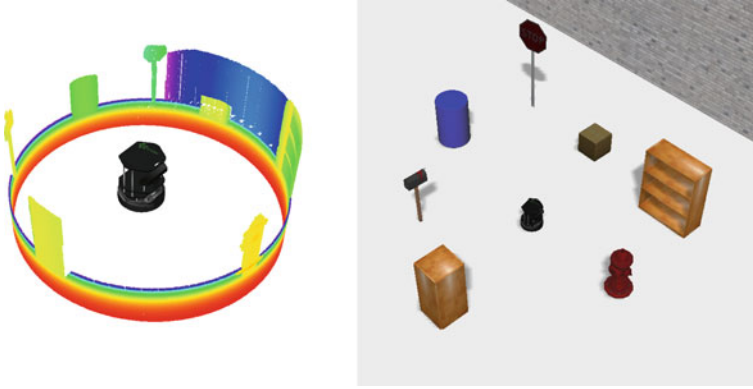


Fig. 14.9 Gazebo/Rviz visualization of virtual egocylinder (left) and the environment it represents. Range colormapped from near to far goes from red to blue/purple

where (x_{ray}, y_{ray}) are the important ray coordinates obtained from the image pixel coordinates r_{im} , and ρ_{ray} is the corresponding length of the ray when treated as a vector. For efficiency purposes, the function values $\rho_{ray} : \mathcal{I} \mapsto \mathbb{R}^+$ should be precomputed and stored in an image whose dimension is the same as the depth image for direct lookup. Note that the camera convention is for the z -coordinate to point along the optical axis and for the x -coordinate to be horizontal with the y -coordinate pointing downwards.

The mapping of the depth value to the egocylinder coordinates involves computing the angle coordinate θ and the height value z_{cyl} , as per

$$\begin{aligned} \theta &= \text{Arg}(x_{ray}(r_{im}) + j) = \theta_{im}(r_{im}) \\ z_{cyl} &= D_m(r_{im})y_{ray}(r_{im}) \end{aligned} \quad (14.4)$$

where the constant imaginary term j is due to the unit z -coordinate in the ray representation. As above, for computational efficiency, the $\theta_{im} : \mathcal{I} \mapsto \mathbb{R}$ and $y_{im} : \mathcal{I} \mapsto \mathbb{R}$ functions should be precomputed over the image domain. If the sensor is a range sensor, the ray will have unit length rather than a unit z -coordinate. The appropriate modifications of the above equations will be needed.

These points then get mapped to egocylinder image coordinates $r_{cyl} \in \mathcal{I}_{cyl}$ using the homogeneous egocylinder projection matrix K_{cyl} ,

$$r_{cyl} = K_{cyl} \begin{bmatrix} \theta \\ z_{cyl} \\ 1 \end{bmatrix} \quad \text{where} \quad K_{cyl} = \begin{bmatrix} f_h & 0 & h_c \\ 0 & f_v & v_c \end{bmatrix}, \quad (14.5)$$

for $f_h = f_v = \cot(2\pi/n_{cols})$, $h_c = n_{cols}/2$, and $v_c = n_{rows}/2$, where $n_{rows} \times n_{cols}$ are the egocylinder image dimensions (note that v_c can be shifted if the domain is biased upwards or downwards). To identify the appropriate bin to map the point

into, the decimal coordinates r_{cyl} should be discretized to whole numbers. The Cartesian coordinates of the point are stored in the bin $\mathcal{B} = \mathcal{B}_{ego}(r_{cyl})$ contained at the discretized coordinate location $\lfloor r_{cyl} \rfloor$, where \mathcal{B}_{ego} consists of all bins in the egocylindrical map representation indexed by discretized coordinate locations.

To generate a range map using the egocylindrical representation, it suffices to compute the range of the point in each bin. For shorthand, we write

$$D_m(r_{cyl}) = \rho(\mathcal{B}_{ego}(r_{cyl})) \quad (14.6)$$

which then renders an egocylindrical range image of all points stored in memory. When updating the egocylindrical representation with recently sensed depth or range information, the new data overwrites the stored data.

Synthesis of a hallucinated egocylindrical image employs the egocylindrical projection equations instead of the standard pinhole projection equations. Collision checks involve the same conceptual procedure described in Sect. 14.5.1, but with egocylindrical range values instead of depth values. The time cost to perform collision checking with egocylindrical images is close to that of traditional depth images, thus the egocylindrical collision time cost curves resemble those of Fig. 14.8.

Egocylinder Propagation

Propagation of the stored points involves transforming them under the motion induced pose change $g_{move} \in SE(2) \subset SE(3)$ from one time point to the next, where g_{move} gives the coordinate frame of the old robot pose relative to the new robot pose. Define the egocylindrical coordinate vector $p_{cyl} = (\rho, \theta, z_{cyl})^T$ and the Cartesian coordinate vector $p = (x, y, z)^T$. Consider the mapping from egocylindrical coordinates to Cartesian coordinates T_{e2c} and vice versa T_{c2e} ,

$$p = T_{e2c}(p_{cyl}) = \begin{bmatrix} \rho \cos(\theta) \\ \rho \sin(\theta) \\ z_{cyl} \end{bmatrix} \quad \text{and} \quad p_{cyl} = T_{c2e}(p) = \begin{bmatrix} \sqrt{x^2 + z^2} \\ \text{Arg}(x + jz) \\ y \end{bmatrix}, \quad (14.7)$$

both with reference to the viewer/camera frame. The new egocylindrical coordinates p'_{cyl} of a stored point p_{cyl} are:

$$p'_{cyl} = T_{c2e} \circ g_{move} \circ T_{e2c}(p_{cyl}). \quad (14.8)$$

Identifying the new bin that the point should be moved to involves applying the projection matrix K_{cyl} to the last two coordinates in p_{cyl} mapped to homogeneous form, then discretizing the resulting coordinate outputs. In the event that multiple points map to the same bin, only the point with the lowest range is kept. To speed up calculations, our implementation only keeps track of p , using p_{cyl} solely to determine in which bin to store p .

With propagated and depth image updated egocylindrical data, synthesis of the egocylindrical range image contains historical knowledge regarding the local environment, thereby mitigating the FOV issues of depth images. The egocylindrical

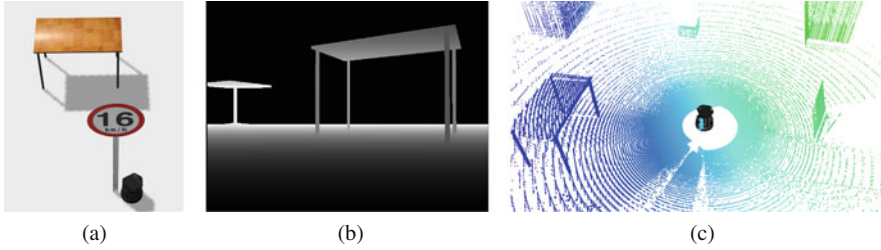


Fig. 14.10 Visualization of an out-of-FOV obstacle scenario. The robot needs to turn right to avoid colliding with the signpost, which is currently outside the FOV of the depth sensor. Since the signpost was visible to the depth sensor previously and is in the egocylinder, it can be avoided. (a) Gazebo world. (b) Depth camera. (c) Egocylinder points

image enhances collision-checking and collision-free navigation when performing tight cornering and maneuvering around obstacles. Figure 14.10 depicts a scenario whereby the mobile robot is close to a signpost and with the signpost outside of the field of view, as noted by the lack of a signpost in the depth camera image. However, the egocylindrical data structure does contain points from the signpost's pole. It is the small blue point cloud near the robot in Fig. 14.10c. The point cloud was generated from the egocylindrical data (using T_{e2c}). The egocylindrical representation is not guaranteed to be correct, as world geometry that was never seen does not exist in the model and is not propagated. Thus it is possible to conclude that a trajectory is safe though it may not be. Forward navigating mobile robots usually exhibit this problem at the beginning of a global trajectory and less so later on due to the fact that forward travel will propagate the seen world out into the unseen portions of the egocylindrical image. The main danger lies when performing high angle turning into unsensed world regions. The role of the global path and the local planner scoring functions is to prevent these situations from happening by giving preference to safer trajectories.

14.5.3 Egocircular Representation and Trajectory Scoring

The egocylindrical perception space representation provides an efficient viewer-centric means to collision check based on current and previous perception space measurements (e.g., depth, range, or disparity). However, it is not an efficient means to score trajectories for collision assessment purposes. The relatively high slope of the PiPS collision-checking cost means that only a small set of trajectories should be tested for collision checking. Thus, the typically large set of trajectories sampled in sample-based methods should be rank ordered with only the top few being evaluated. Additionally, the PiPS approach can only give an indication of safe or unsafe. It cannot score based on proximity to obstacles or other pertinent geometry or goal information. In traditional local planners, the transformation of sensor-based obstacle

geometry to occupancy grids occurs because of the ease with which distance or proximity information can be generated (though there is a significant time cost if the occupancy grid is 3D). The distance information is essential to scoring and rank ordering the sampled trajectories.

For fast trajectory scoring, a more compact and efficient representation of the local collision space is necessary. For that we employ an *egocircle*, which can be thought of as a flattening of the egocylinder image model to a 1D space or laser scanner type of space; the true calculations will be different but the conceptual idea is correct. The egocircle is an egocentric polar obstacle data structure reminiscent of the data structure used in polar based methods [44, 45, 47, 48]. Its purpose is to populate, propagate, and store the local environmental history necessary for approximately and efficiently scoring candidate trajectories relative to obstacle proximity, goal point proximity, and global path following. These scores provide ranked orderings of the sampled trajectories.

The local planner block depicted in Fig. 14.6 initially samples a rich set of trajectories, scores them according to predetermined criteria using the egocircle data, then collision checks them according to their score ranking using the egocylindrical representation and PiPS collision-checking. The first sample to pass the collision-check module is the trajectory to follow for the next local planning period. Because collision checking occurs using the egocylindrical representation, the egocircle scoring does not need to be a strict or conservative scoring method. Rather it can be liberal and admit collision inducing trajectories. Its design is meant to provide efficient scoring, data storage, and propagation implementations.

Egocircle Measurements, Storage, and Propagation

Since the egocircle collapses the 3D information down to 2D information (angle and range), the data format of the egocircle measurement module is compatible with a laser scan. The laser scan information populates the egocircle data structure, whose contents get propagated and updated as the robot maneuvers. Similar to a laser scanner, the egocircle evenly divides the angular space into n_{circ} cells or buckets, with each containing a list of 2D points that fall within the cell's angular range. Generating an egocircular map from the stored data entails performing a min operation over all egocircle buckets individually. For shorthand, we write

$$L_m(r_{cir}) = \min(\rho(\mathcal{L}_{ego}(r_{cir}))), \quad (14.9)$$

where r_{cir} is the coordinate indexing into the egocircle structure, and \mathcal{L}_{ego} is the collection of buckets. The process above renders a 1D measurement “image” L_m from all points stored in memory. It is equivalent to a 360° laser scan sensor reading whose angular resolution is $n_{circ}/(2\pi)$.

Following Sect. 14.5.2, propagation of the stored points involves transforming them under the motion induced pose change $g_{move} \in SE(2)$ from one time point to the next, where g_{move} gives the coordinate frame of the old robot pose relative to the new robot pose. Define the egocircular coordinate vector $p_{cir} = (\rho, \theta)^T$ and the Cartesian coordinate vector $p = (x, y)^T$. Consider the mapping from egocircular coordinates to planar Cartesian coordinates T_{l2p} and vice versa T_{p2l} ,

$$p = T_{l2p}(p_{cir}) = \begin{bmatrix} \rho \cos(\theta) \\ \rho \sin(\theta) \end{bmatrix} \quad \text{and} \quad p_{circ} = T_{p2l}(p) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \text{Arg}(x + jy) \end{bmatrix}, \tag{14.10}$$

both with reference to the viewer/camera frame modeled as an $SE(2)$ frame. The new egocircular coordinates p'_{cir} of a stored point p_{cir} are:

$$p'_{cir} = T_{l2p} \circ g_{move} \circ T_{p2l}(p_{cir}). \tag{14.11}$$

Identifying the new bucket that the point should be moved to involves partitioning the angular values according to the egocircle’s angular resolution. To speed up calculations, our implementation stores points using Cartesian coordinates and keeps track of the minimum range per cell. Points are removed once they lie outside the radius ρ_{max} associated with the local egocircular map. When incorporating new measurements from the most recent depth image, range-based clearing occurs with the stored egocircle data.

This design allows the egocircle to track multiple points in each direction and to quickly return the distances to the nearest obstacle in each direction. With propagated and depth image updated egocircle data, synthesis of the 1D range image per Eq. (14.9) contains historical knowledge regarding the local environment. Figure 14.11 depicts a navigation scenario for a simulated Turtlebot mobile robot.

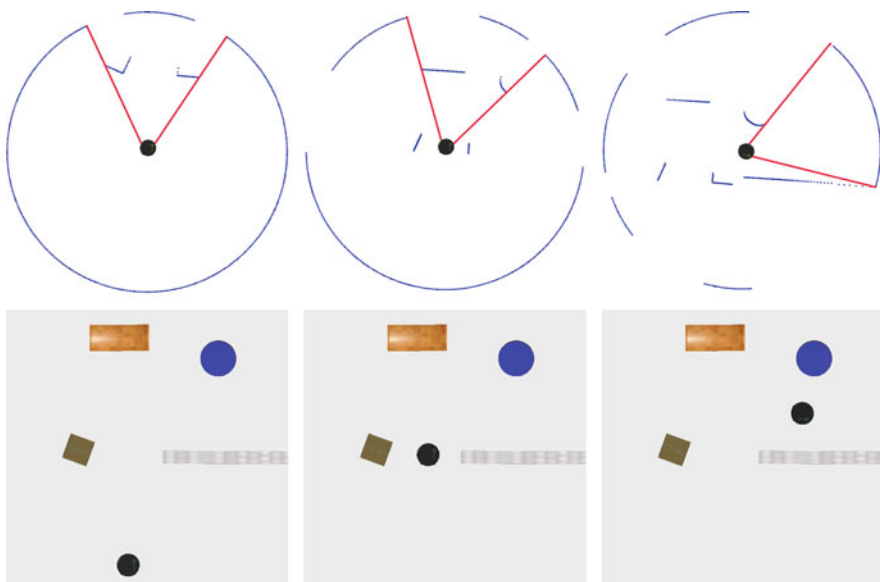


Fig. 14.11 Visualizations of the egocircle measurement predictions based on the stored data. The top row depicts the egocircle measurements where only those points that would be visible to a laser scanner are plotted (occluded points are not). The origin corresponds with the camera origin, but the orientation of the egocircle measurement has been adjusted to roughly align with the global overhead views of the bottom row. The red lines are FOV limits

The robot travels upwards relative to the overhead views in the bottom row, turns right, then proceeds rightwards. Visualizations of egocircle generated measurements are in the top row (but with a coordinate system orientation roughly matching that of the world). The red lines delineate the FOV of the mobile robot. Historical data sensed by the robot is contained in the egocircle measurements outside of the FOV. Note that, in the rightmost egocircle measurement, a section of the upper surface of the wall (just below and to the left of the robot) does not exist in the egocircle data structure. As the robot turns towards the right, FOV limitations mean that this small portion of the wall never gets sensed, hence the missing data. By maintaining a local, approximate cost map in planar space, the egocircle provides a means to rapidly score candidate robot trajectories. The following subsections describe the different cost functions evaluated and contributing to the total score of a candidate trajectory.

14.5.3.1 Egocircle Trajectory-Based Cost Functions

The purpose of the egocircle representation is to replace the grid-based cost functions utilized by DLP and many other non-perception space methods. Figure 14.12 depicts a navigation scenario with the global trajectory (in color green), the actual navigated trajectory (in color red), and a candidate future trajectory (in color yellow). This candidate trajectory should be scored in order to identify an appropriate fitness relative to the constraints of following the established trajectory and avoiding

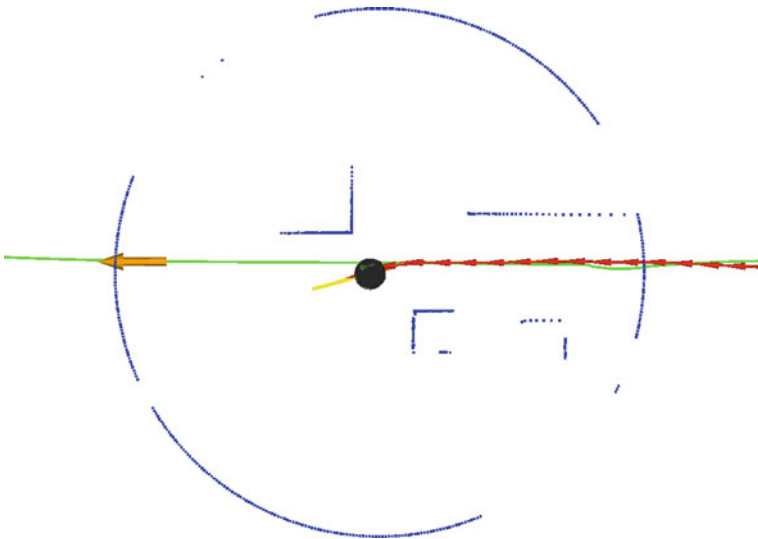


Fig. 14.12 Local egocircular representation with global path (green), sampled local path (yellow), local goal (orange arrow), and odometry (red arrows). The robot moves leftwards

collisions. The local egocircle map contains only the world sensed information within the depicted radius. It generates a local egocircular range scan from the information (color blue), which is used to calculate some of the trajectory costs. The particular costs needed for a functional Move Base implementation include the obstacle cost function, the go-to-goal cost function, and the path comparison cost function. These costs were depicted earlier in Fig. 14.5 based on a local occupancy grid. A description of these costs and the analogous egocircle implementation representation will be given below.

The trajectory scoring is meant to provide a rank ordering of the sampled trajectories from best to worst for prioritizing collision checking and trajectory safety assessment. It does not need to involve a precise representation of the robot model nor scoring functions, as long as the model is approximately correct and the scoring functions are monotonically correct over large swaths of the local area around the mobile robot. Therefore several simplifications are made to improve computational runtime.

An important simplification is using an inflated egocircle range scan in order to treat the robot as a point for certain tasks. Conceptually, a circle of radius r_{ins} is placed at the location of each point represented by L_m and a new egocircle scan is generated based on the ranges to these inflated points (see Fig. 14.13). The inscribed radius r_{ins} of the robot is used to ensure that the result is liberal. This requires first finding the subset of points in the egocircular map within a specified distance d_s of a given pose $p_s = (\rho_s, \theta_s)^T$. We approximate this as:

$$\beta(p_s, d_s) = \left\{ p_b \in L_m \mid (\theta_s - \theta_d) \leq \theta_b \leq (\theta_s + \theta_d) \right\} \quad (14.12)$$

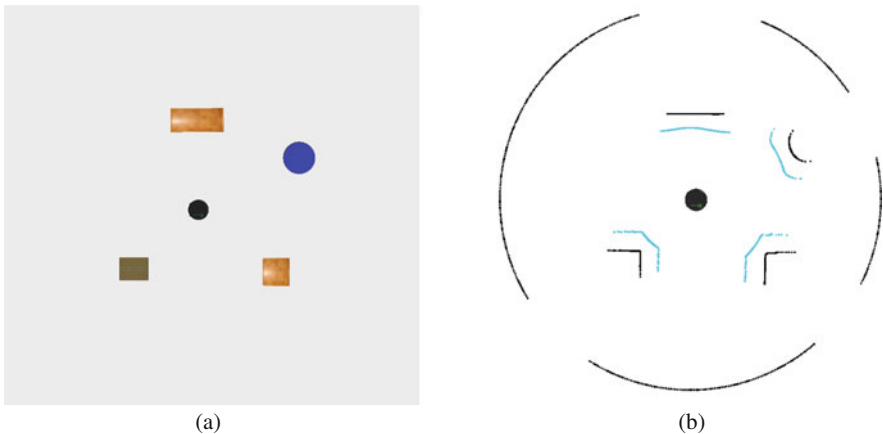


Fig. 14.13 Overhead views showing the simulated environment (left) and the egocircle representation (right). The original egocircle is shown in black while the inflated egocircle is shown in cyan. (a) Simulated environment. (b) Original and inflated egocircles

where $\theta_d = d_s / \rho_s$.

Let $T_{m2p}(L, i)$ map the i th element of egocircle range scan L to egocircular coordinates. The inflated egocircle can then be expressed as follows:

$$L_{infl}(i) = \min_{j \in B(i)} L_m(j) - r_{ins}, \text{ where}$$

$$B(i) = \left\{ j \in n_{circ} \mid T_{m2p}(L_m, i) \in \beta(T_{m2p}(L_m, j), r_{ins}) \right\} \quad (14.13)$$

Obstacle Cost Function

The obstacle cost function reflects the cost of traveling close to obstacles. The obstacle cost of a trajectory is a function of the obstacle costs of the poses in the trajectory. The obstacle cost of query point p is a function of the distance to the nearest obstacle point, represented as $d_{\min}(p)$. The obstacle cost c_{obs} is

$$c_{obs}(p) = c_{obs} \circ d_{\min}(p)$$

$$= \begin{cases} -1, & \text{if } d < r_{ins} \\ \bar{c}_{obs} \exp^{-w(d-r_{ins})}, & \text{if } r_{ins} \leq d < r_{max} \\ 0, & \text{otherwise} \end{cases} \quad (14.14)$$

where $d = d_{\min}(p)$, and \bar{c}_{obs} is a predetermined constant cost. The values r_{ins} and r_{max} represent the nearest permissible distance and the distance beyond which an obstacle has no cost. An obstacle cost of -1 means that a pose definitely collides. Depending on the geometry of the robot (i.e., if it is elongated or otherwise not a circle), it is possible for colliding poses to receive nonfatal obstacle costs. If any pose in a trajectory collides, the trajectory is assigned the fatal cost of -1 . Otherwise, the trajectory is assigned the obstacle cost of the last pose in the trajectory.

The standard costmap-based obstacle cost function uses a distance map to implement d_{\min} . The distance map is computed such that each cell contains the Euclidean distance to the nearest occupied cell in the occupancy grid. The egocircular representation does not admit such a calculation since it is a boundary based polar model of 3D space (it does not measure space according to discretized area or volume). Instead, d_{\min} is brute force computed between the query pose and a local subset of the egocircle as follows:

$$d_{min}(p) = \min_{b \in B} \text{dist}_p(p, b) \quad (14.15)$$

where $B = \beta(p, r_{max})$ and dist_p returns the distance between polar points p and b using the law of cosines.

To understand this distance based cost function, it is best to examine the first column of costmap images in Fig. 14.14. The red regions correspond to 0 values as those locations are far from the obstacles, which are the black regions in the image. The coloring trend goes to blue/purple as the distance from the query location to the

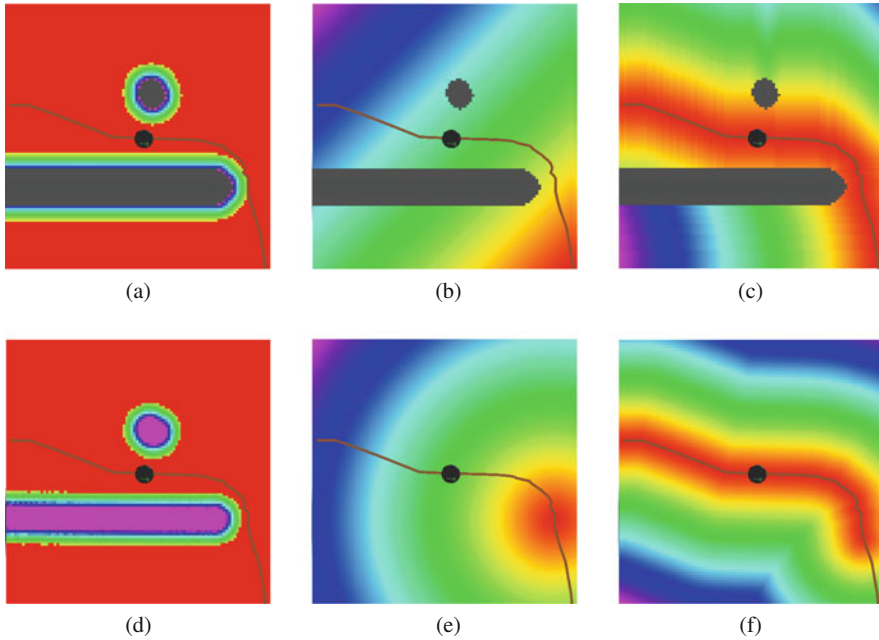


Fig. 14.14 Visualization of the trajectory scoring as cost maps computed on an occupancy grid (top row) and from egocircle (bottom row). The grid points are converted to polar representation and scored according to the described scoring functions. Low cost is red and high cost is blue/purple. Points lethally close to obstacles are black in the occupancy grid and purple in the egocircle. The brown curve is the global path to follow. (a), (d) Obstacle cost. (b), (e) Local goal cost. (c), (f) Path cost

nearest obstacle point lowers to be within the interval $[r_{ins}, r_{max}]$. The black obstacle regions would give -1 values. These should be very large or infinite costs, however the Move Base implementation checks for—and rejects trajectories with—negative scores. The top image, Fig. 14.14a, is the grid-based costmap while the bottom image, Fig. 14.14d, is the egocircle costmap evaluated using Eq. (14.14) over the same grid as the costmap, where each point in the 2D grid is converted to polar representation. The two functions have similar scores outside of the occupied regions. The pairwise computations for the egocircle implementation, per pose tested, are quadratic but based on two low cardinality point sets. The grid-based distance computing scheme is linear in the local occupancy grid area [101], thus it too is quadratic in time cost but with larger base values. Once computed, grid-based trajectory costs are constant per robot pose tested. As with collision checking, the data preparation time cost is near zero for the perception space approach but has a high slope in per pose scoring time. The world centric model has a non-negligible time cost to build the scoring maps and a small constant cost to score per pose. The perception space method will be faster than the grid-based approach until a large enough quantity of poses is sampled. The number of poses to score grows linearly with the number of trajectories to score.

Goal Point Cost Function

The purpose of this cost is to reward candidate trajectories whose terminal points are close to the local goal point. In a grid-based scoring strategy, the local goal is selected as the first pose on the global path that exits the local costmap. The distance map for the go-to-goal cost warps around obstacles and reflects the true cost-to-go if the occupancy map is correct (see Fig. 14.14b). Under an egocircle representation, two deficiencies occur: (1) the true cost-to-go requires more computation when there is an obstacle between the evaluation point $p_{cir,i}$ and the local goal p_{loc}^* , and (2) the true cost-to-go cannot be ascertained when there is an obstacle between the robot's camera (located at the origin) and the local goal. For the first case, we simply compute the distance as though there were no obstacles. We avoid the second case by selecting local goal as the last unobstructed pose on the global path that lies within the egocircle. Poses are classified as obstructed or unobstructed by using the inflated egocircle $r_{inflated}$.

Let the index set J be defined as follows:

$$J = \left\{ j \in \mathbb{N} \mid g_j^* = g^*(t_j) \text{ for } t_j \in \mathbb{R}^+ \right\} \quad (14.16)$$

where $g^*(t)$ is the global path and t_j indexes into it to create a set of global path waypoints.

Pose p_{loc} is unobstructed if $\rho_{loc} < r_{inflated}(\text{Angle}(p_{loc}))$. Poses that are less than $2 * r_{ins}$ behind an inflated point are definitely in collision, but poses further behind may simple be occluded. Consequently, we classify each pose as either safe, colliding, or occluded. If g^*t_j is safe and g^*t_{j+1} is colliding, the global path leads to collision and replanning is triggered. Otherwise, the local goal is selected as follows:

$$p_{loc}^* = g^*(t_j) \max_{j \in J} \left| \text{dist}(g_{robot}, g^*(t_j)) < \rho_{circ} \right.$$

$$\text{and } L_m \circ L_{infl} \left(\text{Angle}(g_{robot}^{-1} g^*(t_j)) \right) > \text{dist}(g_{robot}, g^*(t_j)) \quad (14.17)$$

where g^* is the global path and g_{robot} is the $SE(2)$ pose of the robot in the world frame.

Figure 14.15 summarizes the different cases determining the selection of the local goal. If there are no occluding obstacles, the local goal is the last pose on the global path that lies within the egocircle radius (Fig. 14.15a). Otherwise, the local goal is the last unobstructed pose on the global path (Fig. 14.15b). However, if the global path definitely collides (Fig. 14.15c), local planning is aborted and global replanning occurs.

This cost function is adapted to the star-like free space region associated to the current robot pose and based on its local environment, as captured by L_m . It employs the line-of-sight visibility property, from the robot pose (the origin of the egocircle) to the goal point, to establish the scoring.

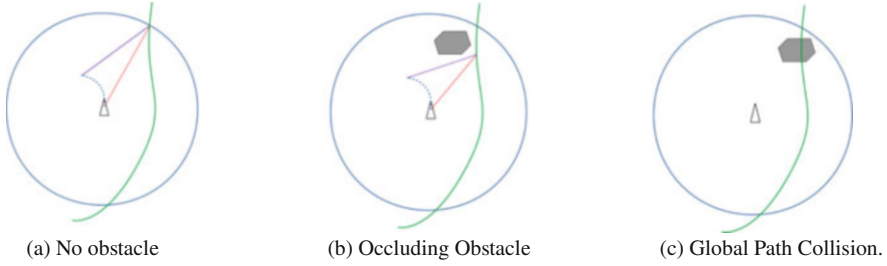


Fig. 14.15 Selection of local goal for goal point cost function without (a) and with (b) an occluding obstacle. The right-most plot (c) depicts a scenario that would trigger global replanning.

Visualization of the go-to-goal cost function is given in the second column of Fig. 14.14 for the grid-based method (top) and the proposed heuristic (bottom). The grid point values would be the scores associated to the point $p_{cir,i}$ if it were located at those grid points. Note that the location of the goal point p^* differs between the two methods. The important characteristic is that the proposed heuristic approximately matches the monotonic increase in the go-to-goal cost of the equivalent occupancy grid cost (see Fig. 14.5b versus Fig. 14.5e). The difference in the shape of the cost level sets is due to the grid-based method using Manhattan distance rather than Euclidean distance.

Path Cost Function

In addition to the go-to-goal cost, there is another cost associated to the global path. This one scores the sample trajectory's terminal point against the global path to identify how close it is to a point on said path. Relying again on the star-like properties of the free-space region described by the polar egocircle estimate L_m , any point on the global path that is obscured by a point on the inflated egocircle is not considered as reachable from the terminal point of the candidate trajectory. With the index set J defined as in Eq. (14.16), the local set of visible points on the global path is the subset:

$$J_{local} = \left\{ j \in J \mid \begin{aligned} &\text{dist}(g_{robot}, g^*(t_j)) < \rho_{circ} \\ &\text{and } L_m \circ r_{inflated} \left(\text{Angle}(g_{robot}^{-1} g^*(t_j)) \right) > \text{dist}(g_{robot}, g^*(t_j)) \end{aligned} \right\} \quad (14.18)$$

where g_{robot} is the $SE(2)$ pose of the robot in the world frame. The set consists of indices to unobstructed global path poses within the egocircle domain. Pose obstruction is evaluated as in the goal cost function. The path cost is then

$$C_{path}(p, J_{local}) = \min_{j \in J_{local}} \text{dist}(p, \text{Pos}(g_{robot}^{-1} g^*(t_j))), \quad (14.19)$$

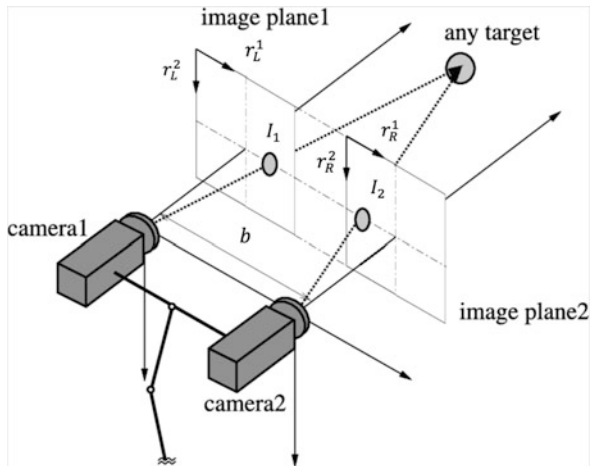
where $\text{Pos}(g)$ grabs the translational coordinates of the $SE(2)$ element g . Again, the distance calculation in the above equation uses the law of cosines to generate the distance using polar representation for point positions. The cost is basically the straight path length from the trajectory terminal point given by p to the path waypoint in the robot's frame given by $\text{Pos}(g_{robot}^{-1}g^*(t_j))$. Global path waypoints obscured by an obstacle do not factor into the cost. The third column of Fig. 14.14 depicts the traditional grid-based path cost (top) and the heuristic polar path cost (bottom). For the latter, regions occluded by obstacles and lacking line of sight have a cost based on the nearest visible path point. As with the previous cost comparison, the overall trend matches between the two implementations in the free-space regions, with the level sets once again having different shapes due to the different distance metrics utilized.

14.5.4 Working with Stereo Cameras

With additional processing, stereoscopic camera configurations provide similar structural information as depth or range sensors (as can multi-camera setups involving more than two cameras with overlapping fields of view). The process for estimating depth, triangulation, requires matching pixels on the left camera to those of the right camera (the two pixels should represent the same world point, usually on an object surface). A popular stereo configuration for robotic systems offsets two cameras horizontally to provide image pairs similar to human binocular vision (see Fig. 14.16).

In the parallel stereo configuration, epipolar lines are horizontal lines in the image plane (e.g., dash dotted line in the figure). For each pixel in the left image, the corresponding pixel in the right image is on the epipolar line. Limiting the search to

Fig. 14.16 Stereo camera model [106]. (r_L^1, r_L^2) and (r_R^1, r_R^2) are coordinate systems of two image planes. b is the baseline of stereo camera model. I_1 and I_2 are pixel values of the same target point in the left and right image planes



a vicinity of the epipolar line constrains the pixel matching search space for more efficient stereo matching. Image rectification can be applied first to transform them into one common image plane so that all epipolar lines are horizontal. When rectified, matching pixels are displaced horizontally between the left and right rectified images. The disparity measures the pixel horizontal distance, usually in the r^1 coordinate, between two corresponding points in the left (L) and right (R) images, $\delta = r_L^1 - r_R^1$. The depth value z is a function of the disparity and camera model parameters.

$$z = \frac{b f^1}{\delta} \quad (14.20)$$

where b is the baseline representing the distance between two camera centers, f^1 is the focal length of the horizontal coordinate, and δ is the disparity.

Following the taxonomy of stereo matching algorithms [107], approaches are divided into local and global methods. The traditional block matching algorithm is one of the local methods. This algorithm first extracts a small patch around the pixel in the left image, and horizontally shifts the patch along the epipolar line by candidate distance values (in a predefined disparity range) in the right image. The estimated disparity is the one that minimizes the difference between the patch in the left image and the shifted patch region in the right image. This difference can be represented by cost function. One traditional cost is the sum of squared differences (SSD).

$$\sum_{(\zeta^1, \zeta^2) \in \mathcal{N}(r^1, r^2)} (I_1(\zeta^1, \zeta^2) - I_2(\zeta^1 + \delta, \zeta^2))^2 \quad (14.21)$$

where $I_1(\zeta^1, \zeta^2)$ is a pixel value in the left image and $I_2(\zeta^1 + \delta, \zeta^2)$ is a pixel value in the right image. Several alternative cost functions exist for the matching optimization, such as normalized cross-correlation (NCC), sum of absolute differences (SAD), mean of absolute differences (MAD), etc. [107]. Hirschmuller [108] also introduces the semi-global block matching algorithm that integrates local pixelwise matching and global smoothness constraints. This approach has better performance when dealing with varied illuminations, occlusions, and low texture surfaces. Moreover, in order to reduce computational complexity and obtain smoother disparity, some optimization and refinement techniques are integrated [107]. Real-time disparity estimation implementations exist based on specialized hardware approaches, such as via onboard FPGA [109, 110] or graphical processing units [111]. Once the disparity image is estimated, generation of the depth image using Eq. (14.20) is immediate and can be used within the PiPS local planner. Though [105] employs disparity space for navigation, we advocate its conversion to depth space due to the fact that calculation of the equivalent Cartesian point representation is more efficient for propagation of the points in the egocylindrical and egocircular representations (and easier to write as a set of operations).

14.6 Benchmarking Navigation Methods

To aid in the evaluation and comparison of navigation frameworks and strategies, this section describes a set of ROS/Gazebo based environments and associated initial/terminal point synthesis methods for generating repeatable navigation scenarios. While it is preferable to deploy in real-world scenarios, doing so is more difficult due to the need for the other components of a robotic navigation system to be working perfectly, the setup and real-estate costs of creating and maintaining the environment, the lack of configuration flexibility [112]. Plus more universal evaluation by other researchers in these actual environments would be difficult. The value of ROS/Gazebo is that highly repeatable experiments are possible in a diverse array of worlds. The same experiments can be performed by anyone with access to a system configured with ROS/Gazebo and to our publicly available benchmark worlds and testing configurations [113]. Furthermore, in our experience developing the original local PiPS algorithm [100], we found little difference in performance between deploying in a well-designed Gazebo world and in our actual office environment.

The testing protocol for the benchmark scenarios includes Monte Carlo runs employing multiple point-to-point navigation instances that generate statistical outcome models for comparison purposes. Important metrics include completion rate, path length, and travel time. Though Gazebo simulations are not perfectly deterministic, the outcomes should be close enough that the final Monte Carlo statistics will have low variation (i.e., inter-experiment variance is low).

14.6.1 World Synthesis

Because navigation is a generic capability expected of mobile robots, the environments where robots may be deployed will vary in scale, structure, and obstacle density. The proposed benchmark consists of several synthetic worlds modeled after environments observed to exist here on our university campus. These synthetic worlds are called *sector world*, *campus world*, and *office world*.

1. **Sector World** (Fig. 14.17a). The sector world consists of a single large room, partially divided in the middle by a wall running from left to right. It is intended to represent locations that are essentially large open areas dotted with a wide assortment of obstacles. The lack of known obstacles means that the global planner will generate simple piecewise linear paths from a start pose to a target goal pose.
2. **Campus World** (Fig. 14.17b). The campus world is intended to model the outdoor free space of a university campus, where adjacent structures and landscaping force one to follow specific paths, and open areas admit more free form movement across them. Consequently, it consists of several relatively large open areas connected by narrower corridors. The narrow corridors are well defined and generally clear

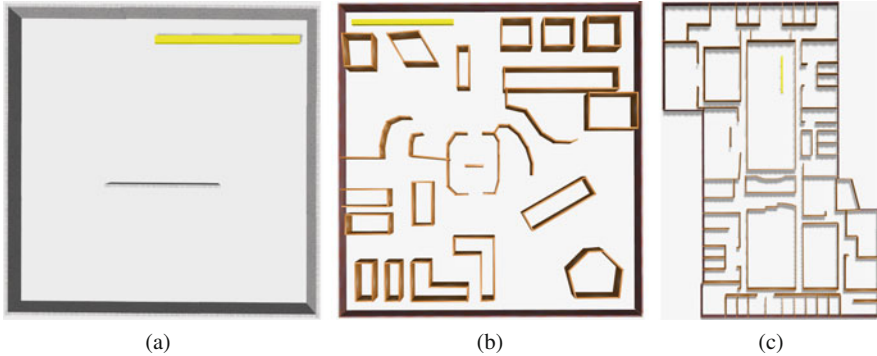


Fig. 14.17 Overhead views of the benchmark worlds. The 10m yellow bar provides a means to assess the scaling of the three worlds. The office world is rotated. (a) Sector world. (b) Campus world. (c) Office world

of miscellaneous obstacles. The open areas (e.g., open quads) are presumed to have multiple purposes and therefore will contain a randomized assortment of obstacles. When considering particular global planning or world modeling strategies, some areas of this world are best simplified through a graph-based or topological model of the space (e.g., corridors linking open spaces), whereas other areas benefit from spatial world models.

3. **Office World** (Fig. 14.17c). The office world is based on digital architectural floor plans of the fourth-floor of the building containing our lab, provided by the interior designers of the building after it was remodeled. It serves as an example of a real office indoor scenario, with long hallways connecting open cubicle areas, enclosed offices, and larger conference rooms and laboratory spaces. While halls are narrow enough to be easily blocked, there are generally alternate routes available. Doors that are usually closed have been replaced with walls.

For each of these worlds, there is an associated global map containing only the permanent structural elements (the walls). Global paths generated from the map will typically be infeasible due to unmodeled obstacles. Some paths may be blocked for the narrower passages, should there be objects (randomly) placed within them. The next section describes how specific navigation scenarios are configured and instantiated.

14.6.2 Scenario Configuration

Monte Carlo runs provide statistical insight on the performance outcomes from a large sample of data. In each Monte Carlo run, one or more characteristics of the experiment are determined by a provided random seed. The seed ensures that experimental conditions are variable and repeatable. An important aspect of a



Fig. 14.18 The left area depicts laser safe obstacles while the right area depicts laser unsafe obstacles. The top-right obstacle is a custom box with very short height that cannot be detected by the laser scan.

scenario configuration is the type and placement of obstacles. We distinguish between two general categories of obstacles: laser safe and laser unsafe. Laser safe obstacles have vertical sides or an invariant occupancy profile from the floor up to the height of the robot. This property ensures constant cross-sectional geometry and satisfies the assumptions of laser-scan based planning approaches regarding world geometry for successful collision avoidance. Laser scan approaches should be able to safely avoid laser safe obstacles but may not be able to avoid laser unsafe obstacles. Figure 14.18 depicts some of the safe and unsafe obstacles available for the scenarios, as available through the Gazebo model database. The specific randomized configurations for each world are as follows:

1. **Sector World** (Fig. 14.19a). Starting poses are sampled from a line (depicted in red) running inside and parallel to the north wall of the world and place the robot facing inwards. The coordinates for this region are $x \in [-9, 9]$ and $y = -9$. Goal poses are sampled from a parallel line just inside the south wall (depicted as the green line), whose coordinates are $x \in [-0, 9]$ and $y = 9$. All navigation tasks require traveling from one side of the world to the other. The area between the start and goal lines is populated by laser safe obstacles at fixed locations in the world. Low and medium density configurations exist. Manual placement of the obstacles was done with the aim of creating an approximately uniform distribution with moderate clearance between obstacles. If desired, laser unsafe obstacles can also be randomly placed throughout the same area of the environment.
2. **Campus World** (Fig. 14.19b). For the campus world, there is one start pose (the red dot) and seven candidate goal poses. A given scenario will randomly select one of these predefined goal poses. A specified number of obstacles are uniformly distributed among the primary open areas of the world (random position and orientation). The obstacle type is randomly chosen as either a blue cylinder (laser

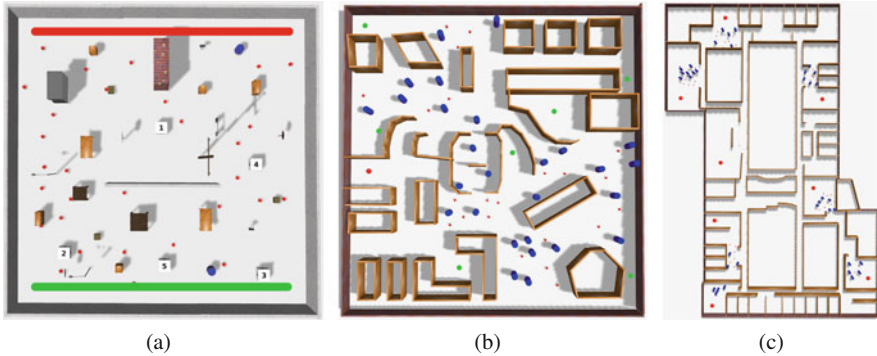


Fig. 14.19 Worlds annotated with start (red) and goal (green) points or regions. For sector world, the start and final points are selected from regions. The campus world has a single start point (larger red circle) and multiple goal points (green circles). The office world start and goal points are randomly chosen from the red circles in the map. The worlds also show examples of random positions populated with obstacles. The blue objects are randomly placed laser safe obstacles. The smaller red dots are laser unsafe obstacles. **(a)** Sector world. **(b)** Campus world. **(c)** Office world

safe) or a small red box (laser unsafe). As obstacles are placed, a minimum obstacle-spacing threshold ensures that the navigation task remains feasible.

3. **Office World** (Fig. 14.19c). In the office world, there is a fixed set of locations (the red points). Start and goal poses are randomly selected from a list of locations around office. In order to reduce the duration of experiments in this significantly larger world, the goal pose is randomly selected from only the three poses nearest to the start pose. Obstacles are randomly placed in designated regions (free space between start/goal poses and corridors) using the same approach as in Campus World.

14.6.3 Benchmarking

Benchmarking a particular navigation strategy involves multiple runs with a pre-determined set of random seeds. Comparison with another navigation method requires using the same random seed set. An experiment instance proceeds through four stages:

1. **World Setup:** The test Gazebo world is loaded and a given robot model is placed within in it.
2. **Task Setup:** The robot is moved to the starting pose and all obstacles are added to the environment. The specified robot navigation method is then initialized.
3. **Navigation:** The goal pose is sent to the robot controller and a timer is started. The experiment remains in this stage until one of the following conditions is met:

- a. *Succeed*: the robot reaches the goal without collisions;
- b. *Bumper Collision*: the robot bumps into an obstacle en route to the goal;
- c. *Aborted*: the controller reports that planning has failed;
- d. *Timed out*: the timer reaches 10 min.

During execution of the navigation scenario, pertinent scoring metrics are maintained or accumulated as needed.

4. End: The value of the timer is saved as “path time.” The final path length is also saved, along with the condition that ended Navigation. The controller is shutdown.

To compare performance, several metrics are calculated and stored for each experimental instance. Popular metrics for evaluating robot navigation frameworks include success rate, defined as the number of successful runs divided by the total number of runs; path distance, defined as the robot path length travelled from start to goal; and path time, defined as the time required for robot to reach the goal. The latter two statistics are computed from the subset of successful runs. These metrics measure the performance of planners from different perspective including robustness, efficiency, and optimality. A good planner should perform well in these metrics. Potential navigation parameters to consider or configure include the replanning rate, the recovery behaviors, and the local map radius.

14.7 Navigation Experiments

Evaluation of the described perception space approach to navigation (PiPS DLP) will consist of Monte Carlo testing on the benchmark navigation scenarios. Comparison will be made with the standard Move Base implementation (baseline DLP), designed for use with a laser scanner. The first scenario will test in the sector world, with both laser-scan friendly and laser-scan unfriendly sector world instances. No laser unsafe obstacles are added to friendly instances, while 30 are added to unfriendly instances. The purpose of the experiment is to show that the perception-space approach performance matches the classical methods under environmental conditions appropriate for laser scanners and outperforms the classical methods under more general conditions. The second set of scenarios will test the perception space and classical navigation algorithms on the other two benchmark worlds, campus world and office world.

An additional experimental variable will be the mobile robot type, where the robot geometry will vary. The two robots will be the Turtlebot, Fig. 14.20a, and the Pioneer mobile robot, Fig. 14.20b. The Turtlebot is a two-wheel differential drive robot platform with cylindrical robot configuration and a circular base. For collision checking purposes, it is modeled as a cylinder with 0.2 m radius. Depth images on the Turtlebot are captured by a Kinect camera. The Pioneer is a four-wheel skid-steer drive robot platform with a non-circular base, and non-cylindrical configuration. The robot is modeled as a 0.56 m \times 0.5 m rectangular box and configured to have a Realsense R200 depth camera. Since we are only evaluating the ability of the

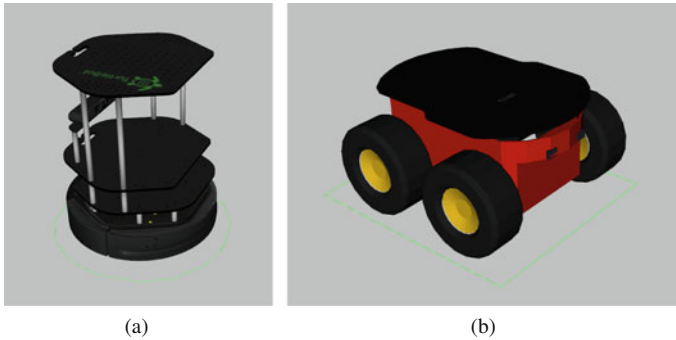


Fig. 14.20 Robot models used in the experiments. Green boundaries are footprints of robots. (a) Turtlebot. (b) Pioneer

Table 14.4 Common parameters

Goal position tolerance	Goal orientation tolerance	Global replanning frequency	Controller patience
1 m	2π	1	3
Planner patience	Local planning frequency	# v samples	# ω samples
5	5	6	20

controllers to handle the changed geometry, the Pioneer is simulated as a Turtlebot base with the Pioneer’s chassis geometry on top. Since neither of these robots is equipped with a laser scanner, we use the Depth Image to Laser Scan ROS package [114] to create virtual scans based on the 10 rows of pixels nearest the optical center of the depth camera. Each scenario involves 50 Monte Carlo experimental runs for each local planner and each robot model.

Global replanning and recovery behaviors (see section “Navigation Using Move Base”) are both enabled in all experiments. The time threshold of recovery behavior is defined as *controller patience*. The purpose of the recovery behavior for baseline DLP is to clear space in the local costmap. Since PiPS DLP does not use a local costmap, its recovery behavior is altered to rotate the robot in 90° increments in an attempt to point the robot away from whatever was obstructing it. Recovery behaviors are also used if *planner patience* elapses without a valid global plan being found. The size of the local region considered during local planning is also an important parameter. A $5\text{ m} \times 5\text{ m}$ square costmap with 5 cm resolution is used in baseline DLP, while an egocircle with 512 cells and a radius of 3 m is used in PiPS DLP. Common parameter values are given in Table 14.4. The forward sim time values are different: 1 s for baseline DLP, 2 s for PiPS DLP. Though PiPS DLP can be run with a local planning frequency of nearly 30 Hz, here we use the default frequency value of baseline DLP (5 Hz) for all experiments. The global costmaps of both approaches are populated using the virtual laser scans described above. The global costmap of

PiPS DLP is also populated with the locations of collisions detected by the collision checker. The values for all customized parameters are available in the configuration files at [113].

14.7.1 Sector World with Laser Safe and Unsafe Obstacles

In the medium density sector world with laser safe obstacles, both baseline laser scanner DLP and Ego-Centric perception-space planning have nearly 100% success rates (see Table 14.5), which shows that our approach has similar performance under normal environments. The failure case abbreviations are bumper collision (BC), aborted (AB), and time-out (TO). The Pioneer and Turtlebot robots each only had one AB out of 50 runs, with the rest being successful. The path lengths taken by all of the robots were within 2% of each other, indicating that they all found comparable paths. The completion times of the PiPS approach were close to but a few seconds more than the laser scan baseline.

After randomly adding 30 laser unsafe obstacles with 1 and 1.2 m minimum distance between each other for the Turtlebot and Pioneer robots, respectively, the success rate of the baseline laser scanner DLP drops to 40% and 24%. The egocentric PiPS approach still has good performance with a success rate of 94% and 84%, see Table 14.6. Again, the average path lengths and completion times of both methods are similar, except for PiPS approach with Pioneer robot. The additional maneuvers necessary for keeping clear of laser unsafe obstacles led to reduced forward speeds for PiPS, yielding completion times 30% larger than the baseline. The success rate of Pioneer PiPS DLP is lower than that of Turtlebot. Most of the additional failure cases are caused by bumper collision. Due to the geometry of Pioneer and the programming of the recovery behavior, it can collide with obstacles while executing the recovery behavior.

Table 14.5 Results for sector world with laser safe obstacles

Sector world with laser safe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	100%	40.98 s	19.62 m	0%/0%/0%
PiPS DLP	98%	42.96 s	19.47 m	0%/2%/0%
Pioneer				
Baseline DLP	100%	43.42 s	20.03 m	0%/0%/0%
PiPS DLP	98%	46.36 s	19.74 m	0%/2%/0%

Table 14.6 Results for sector world with laser safe and randomly placed laser unsafe obstacles

Sector world with laser unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	40%	43.62 s	19.68 m	58%/2%/0%
PiPS DLP	94%	46.37 s	19.68 m	0%/6%/0%
Pioneer				
Baseline DLP	24%	43.45 s	19.76 m	76%/0%/0%
PiPS DLP	84%	56.64 s	20.51 m	8%/8%/0%

Table 14.7 Results of campus world with randomly placed laser safe and unsafe obstacles

Campus world with randomly selected laser safe and unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	68%	38.82 s	18.44 m	30%/2%/0%
PiPS DLP	80%	50.03 s	20.37 m	0%/20%/0%
Pioneer				
Baseline DLP	50%	40.17 s	19.26 m	48%/2%/0%
PiPS DLP	88%	49.27 s	21.43 m	8%/4%/0%

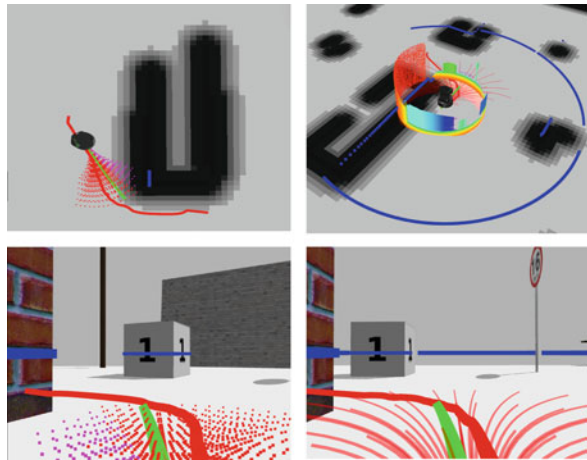
14.7.2 Campus World and Office World

In campus world and office world, 50 obstacles are randomly selected from laser safe and unsafe obstacles with the same minimum distance offsets as in sector world. In all cases, the PiPS modification has improved success rates versus the equivalent baseline implementation (Tables 14.7 and 14.8). Furthermore, the success rate of the baseline navigation scheme decreases when switching from the cylindrical robot to the rectangular box robot while the success rate of PiPS DLP does not. PiPS completion times continue to be a few seconds longer, as seen in the Sector world cases, while the path lengths remain similar to those of the baseline cases. Having roughly comparable path lengths indicates that the scoring system of the egocircle representation is capable of providing a ranked ordering of the sampled trajectories similar to that of the baseline local planner. In campus world, the completion rates of Pioneer PiPS DLP are larger than those of Turtlebot. One reason could be the different minimum distances between obstacles (1.2 m for Pioneer, 1 m for Turtlebot). Though the distances are designed to give similar clearance to both robots, the greater spacing may disproportionately reduce the problem of local minima for the Pioneer (see Sect. 14.7.3). It should also be noted that the majority of failures are due to aborted navigation as opposed to collisions. The PiPS method is successful at avoiding collisions. The abort outcomes reveal failure modes of the hierarchical planner.

Table 14.8 Experiment results of office world with randomly placed laser safe and unsafe obstacles

Office world with randomly selected laser safe and unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	72%	100.93 s	51.03 m	28%/0%/0%
PiPS DLP	92%	103.54 s	48.70 m	0%/8%/0%
Pioneer				
Baseline DLP	66%	98.18 s	64.25 m	34%/0%/0%
PiPS DLP	96%	104.96 s	61.21 m	2%/2%/0%

Fig. 14.21 Visualizations of the navigation process of baseline DLP (left) and perception-space DLP (right). The top row has visualizations in the 3D world space from an external reference frame. The bottom row has visualizations of the same information overlaid on the robot’s camera view



14.7.3 Review of Outcomes

Here, we look a little more closely at the two different hierarchical navigation implementations and also review the causes of failure. First, we review how the baseline and PiPS navigation methods compare. The navigation information associated to the two implementations is shown in Fig. 14.21. The top row provides an overhead view of the mobile robot during navigation past and around a wall, with the local segment of the global path also plotted. Due to the expansion of the occupancy grid by the robot radius, the occupied regions are slightly thicker than the true occupancy. The blue curve segments on the plots are from the sensor information (laser scan or egocircle) and indicate the true locations of the obstacle surfaces. The depicted occupancy grids communicate the information that the baseline DLP system would have during planning, while the red and magenta points are sampled and scored robot poses. The magenta poses correspond to trajectories with collisions, while the red poses correspond to safe trajectories. The thick green curve is the chosen local path to follow. It is the trajectory with the lowest weighted sum of obstacle cost, local

go-to-goal cost, and path cost. In the case of the PiPS approach, it was also deemed to be collision free based on the egocylindrical model. The baseline DLP system is depicted by the left column images, and the PiPS DLP system is depicted by the right column images. For the PiPS approach, the visualization is augmented with the egocylindrical image, which contains memory of historical depth information to reduce the effect of the sensor's limited field of view. The wall data points to the left of the robot are colored red, indicating close proximity to the robot. The other obstacles in front of, to the right of, and behind the robot are colored green to blue, indicating larger distances from the robot. Also depicted is the egocircle (colored blue) which shows that the navigation system has good knowledge of the local surroundings for informing the path scoring component of the local planner.

Failure Cases

Next, we explore the failure cases for the PiPS DLP navigation method. The two failure case types experienced were bumper collisions and aborts.

Bumper collisions occur much more frequently with the Pioneer than with the Turtlebot. The main source of bumper collisions with the Pioneer is rotating in place near an obstacle since unlike with the Turtlebot this can cause the Pioneer to go from a noncolliding state to a colliding state. This happens most frequently if a laser unsafe obstacle is directly on the global path. Such obstacles are represented in the egocylindrical model but not in the egocircle. As a result, evaluation of the cost functions prioritizes trajectories that stick close to the global path (see Fig. 14.22a). The trajectory accepted by collision checking will therefore be the one that gets the robot as close to the obstacle as possible the obstacle. Similar behavior is exhibited by the Turtlebot, but the Turtlebot is able to safely rotate in place to follow a replanned global path while the Pioneer cannot (see Fig. 14.22b). Even if the local planner

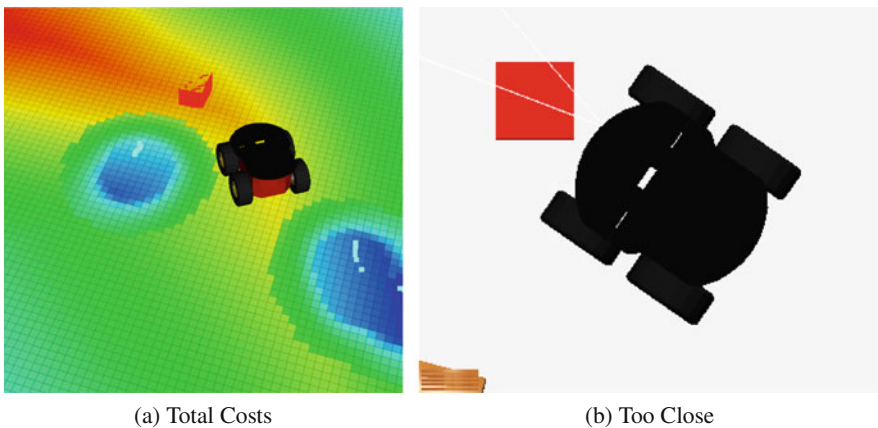
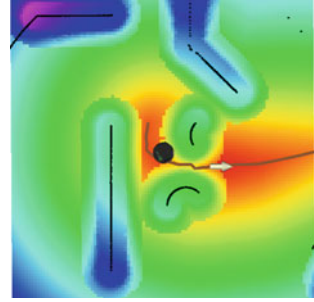


Fig. 14.22 (a) Visualization of approximate total costs associated with trajectories ending at each point on the grid (red = low, blue = high). A laser unsafe obstacle can be seen in front of the robot (colored red). (b) Visualization of robot's state a short time later; attempts to turn will result in collision

Fig. 14.23 Approximate total cost values of a scenario with a local minimum (red = low, blue = high). Also visualized: local goal (tan arrow), global path (brown curve), egocircle (black points)



correctly concludes that it cannot safely turn, this may result in the execution of a recovery behavior and still result in a collision. Incorporating collision checking information into the egocircle should help to prevent the planner from taking the robot so close to obstacles. Another option may be to permit the robot to drive backwards if it is unable to go forwards or turn in place.

Local planning can also fail by getting stuck in a local minimum. Figure 14.23 depicts a scenario where the Turtlebot must pass between two obstacles in order to follow the global path (brown). As shown by the color coded total cost values (red = low, blue = high), there is a local minimum on the near side of the obstacles that prevents the robot from traveling into the gap. Since the gap is sufficiently wide for the Turtlebot to safely enter it, global replanning does not provide an alternative path and navigation ultimately aborts. Tuning cost function parameters is only a partial solution since different situations may require different sets of parameters to achieve the desired behavior [61]. Incorporating concepts from gap-based approaches may help to prevent problems related to local minima.

14.7.4 Implementation Using Stereo Camera

We now explore the performance of PiPS DLP when using a stereo camera. As mentioned in Sect. 14.5.4, the depth images required by the PiPS system can be produced by stereo matching algorithms. For these experiments, a stereo camera is attached 20 cm in front of the Kinect on the Turtlebot and at the same position as the Realsense on the Pioneer. A stereo camera is simulated using *gazeborosmulticamera* Gazebo plugin with 7 cm baseline, 60° field of view, 640 × 480 resolution, and 30 Hz frame rate.

The ROS stereo image processing package [115] generates disparity maps from stereo image pairs. Depth images can be easily computed from these disparity maps per Eq. (14.20). Traditional block matching (BM) and semi-global block matching (SGBM) are both implemented in the ROS package. Figure 14.24 shows the results of these methods on a simulated scene after parameter tuning. The first column (Fig. 14.24a, d) depicts the simulated scene. In order to improve the performance

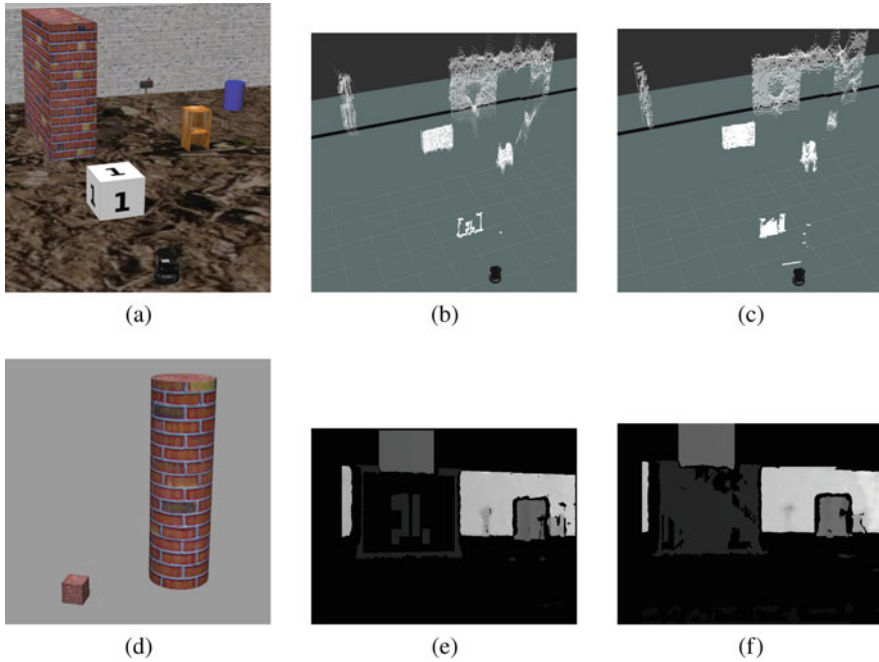


Fig. 14.24 Visualization of stereo implementation. (a) A simulated world with textured ground plane and Turtlebot; (d) The textured box and cylinder obstacles used in experiments; (b) and (e) The pointclouds and depth image generated with the traditional block matching (BM) method; (c) and (f) The pointclouds and depth image generated with the semi-global block matching (SGBM) approach

of stereo matching, texture has been added to the ground plane as well as to the randomly placed obstacles used in the experiments. The middle column (Fig. 14.24b, e) shows the stereo matching results from BM for the scene in Fig. 14.24a. The right column (Fig. 14.24c, f) displays the results from SGBM for the same scene. BM and SGBM represent different trade-offs between speed and accuracy in stereo processing. BM is significantly faster than SGBM (approximately 25 ms vs 120 ms on the test machine), potentially enabling faster planning rates. However, the additional processing performed by SGBM results in smoother, higher quality depth estimates. The difference is especially apparent when looking at each algorithm's depth estimate of the white cube in Fig. 14.24a: BM was only able to estimate depth along the edges of the cube and the number '1' printed on it while SGBM was able to generate an accurate estimate for most of the face of the cube. Since the local planning rate used in the previous experiments was only 5 Hz, the superior quality of SGBM's depth images outweighs its longer processing time.

Apart from the depth image (and consequently the virtual laser scan) being derived from stereo matching rather than a depth camera, the stereo implementation experiments are identical to the previous experiments. The experiment results are

Table 14.9 Results for sector world with laser safe obstacles using stereo implementation

Sector world with laser safe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	100%	41.90 s	19.63 m	0%/0%/0%
PiPS DLP	90%	44.85 s	19.53 m	0%/10%/0%
Pioneer				
Baseline DLP	100%	44.04 s	20.00 m	0%/0%/0%
PiPS DLP	82%	46.49 s	19.74 m	4%/14%/0%

Table 14.10 Results for sector world with laser safe and randomly placed laser unsafe obstacles using stereo implementation

Sector world with laser unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	34%	41.90 s	19.63 m	66%/0%/0%
PiPS DLP	72%	51.58 s	19.83 m	6%/22%/0%
Pioneer				
Baseline DLP	22%	443.45 s	20.44 m	76%/2%/0%
PiPS DLP	74%	56.69 s	20.61 m	8%/18%/0%

Table 14.11 Results of campus world with randomly placed laser safe and unsafe obstacles using stereo implementation

Campus world with randomly selected laser safe and unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	64%	38.58 s	18.43 m	34%/2%/0%
PiPS DLP	84%	49.83 s	20.67 m	8%/8%/0%
Pioneer				
Baseline DLP	46%	41.09 s	19.50 m	50%/4%/0%
PiPS DLP	86%	48.10 s	20.88 m	6%/8%/0%

shown in Tables 14.9, 14.10, 14.11, and 14.12. In all cases, the success rate of the stereo implementation is equal or slightly lower than the success rate of the corresponding depth image implementation. The stereo implementation is vulnerable to all of the failure cases described in Sect. 14.7.3 in addition to stereo-related sources of failure (noise, occlusion, featureless surfaces, etc.).

Table 14.12 Results of office world with randomly placed laser safe and unsafe obstacles using stereo implementation

Office world with randomly selected laser safe and unsafe obstacles				
Approach	Success rate	Completion time	Path length	Failures (BC/AB/TO)
Turtlebot				
Baseline DLP	72%	101.09 s	60.50 m	28%/0%/0%
PiPS DLP	88%	115.52 s	48.38 m	4%/8%/0%
Pioneer				
Baseline DLP	66%	100.55 s	75.94 m	34%/0%/0%
PiPS DLP	90%	104.33 s	70.63 m	4%/4%/2%

14.8 Conclusion

Modern hierarchical navigation methods mostly rely on laser scan sensor measurements due to the computational cost of processing the depth or range imagery signals generated from contemporary sensors. Approaches geared towards resolving this problem rely on data structures that are efficient for low resolution imagery but do not scale well for higher resolution imagery. Modifying the internal world representation of the local planner to a viewer-centric or perception-space world representation avoids the cost of mapping the data to data structures with poor scaling and grants linear scaling properties as a function of the image resolution. A local planning pipeline for trajectory scoring and collision checking using perception space has the potential to replace the existing laser scan inspired strategies while preserving real-time operation. This chapter described a set of modifications employing perception space for a classical hierarchical navigation system. When evaluated on a described navigation benchmark, the perception space navigation system had comparable or better performance to the original laser scan implementation. Deficiencies of the system were found to be a result of the local planner scoring system rather than of the perception space modifications. Analysis and improvement of the scoring functions and the recovery behaviors should resolve the identified issues. Alternatively, exploring perception-space implementations of other local planning strategies might lead to improved performance. Future work aims to do so.

Acknowledgement This work was supported in part by NSF Awards #1400256 and #1849333.

References

1. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., et al. (2005). *Principles of robot motion: Theory, algorithms, and implementation*. Cambridge: MIT Press.
2. LaValle, S. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.
3. Ivanov, M., Lindner, L., Sergiyenko, O., Rodríguez-Quíñonez, J. C., Flores-Fuentes, W., & Rivas-Lopez, M. (2019). *Mobile robot path planning using continuous laser scanning* (pp. 338–372). Hershey: IGI Global.

4. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968, July). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107.
5. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. Available: <http://dx.doi.org/10.1007/BF01386390>
6. Stentz, A. T. (1994, May). Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)* (Vol. 4, pp. 3310–3317).
7. Koenig, S., & Likhachev, M. (2005, June). Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3), 354–363.
8. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., & Thrun, S. (2008). Anytime search in dynamic graphs. *Artificial Intelligence*, 172(14), 1613–1643. Available: <http://www.sciencedirect.com/science/article/pii/S000437020800060X>
9. García, F. M., Kapadia, M., & Badler, N. I. (2014, May). Gpu-based dynamic search on adaptive resolution grids. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1631–1638).
10. Tsardoulias, E. G., Iliakopoulou, A., Kargakos, A., & Petrou, L. (2016, December). A review of global path planning methods for occupancy grid maps regardless of obstacle density. *Journal of Intelligent & Robotic Systems*, 84(1), 829–858.
11. Takahashi, O., & Schilling, R. J. (1989, April). Motion planning in a plane using generalized Voronoi diagrams. *IEEE Transactions on Robotics and Automation*, 5(2), 143–150.
12. Kavraki, L. E., Kolountzakis, M. N., & Latombe, J. (1998, February). Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1), 166–171.
13. Bohlin, R., & Kavraki, L. E. (2000, April). Path planning using lazy PRM. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)* (Vol. 1, pp. 521–528).
14. Belghith, K., Kabanza, F., Hartman, L., & Nkambou, R. (2006, May). Anytime dynamic path-planning with flexible probabilistic roadmaps. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (pp. 2372–2377).
15. van den Berg, J., Ferguson, D., & Kuffner, J. (2006, May). Anytime path planning and replanning in dynamic environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (pp. 2366–2371).
16. Frazzoli, E., Dahleh, M. A., & Feron, E. (2002). Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1), 116–129. <http://arc.aiaa.org/doi/abs/10.2514/2.4856>
17. Kuffner, J. J., & LaValle, S. M. (2000, April). RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)* (Vol. 2, pp. 995–1001).
18. Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894. Available: <http://ijr.sagepub.com/content/30/7/846.abstract>
19. Kuwata, Y., Fiore, G. A., Teo, J., Frazzoli, E., & How, J. P. (2008, September). Motion planning for urban driving using RRT. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1681–1686).
20. Ferguson, D., Kalra, N., & Stentz, A. (2006, May). Replanning with RRTs. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (pp. 1243–1248).
21. Zucker, M., Kuffner, J., & Branicky, M. (2007, April). Multipartite RRTs for rapid replanning in dynamic environments. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 1603–1609).
22. Bruce, J., & Veloso, M. (2002, September). Real-time randomized path planning for robot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 3, pp. 2383–2388).

23. Otte, M., & Frazzoli, E. (2015). *RRT X: Real-time motion planning/replanning for environments with unpredictable obstacles* (pp. 461–478). Cham: Springer International Publishing. Available: http://dx.doi.org/10.1007/978-3-319-16595-0_27
24. Pivtoraiko, M., & Kelly, A. (2011). Kinodynamic motion planning with state lattice motion primitives. In *Proceedings of the IEEE International Conference on Intelligent Robotic and Systems*.
25. Hauser, K., Bretl, T., Harada, K., & Latombe, J.-C. (2008). Using motion primitives in probabilistic sample-based planning for humanoid robots. In S. Akella, N. Amato, W. Huang, & B. Mishra (Eds.), *Algorithmic foundation of robotics VII. Springer Tracts in Advanced Robotics* (Vol. 47, pp. 507–522). Berlin: Springer.
26. Frazzoli, E., Dahleh, M., & Feron, E. (2005). Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics*, 21(6), 1077–1091.
27. Şucan, I. A., Moll, M., & Kavraki, L. (2012, December). The open motion planning library. *IEEE Robotics & Automation Magazine*, 19, 72–82. <http://ompl.kavrakilab.org>.
28. Paranjape, A. A., Meier, K. C., Shi, X., Chung, S., & Hutchinson, S. (2013, November). Motion primitives and 3-d path planning for fast flight through a forest. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2940–2947).
29. Khatib, O. (1985, March). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 500–505).
30. Rimon, E., & Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5), 501–518.
31. Arslan, O., & Koditschek, D. (2016). Exact robot navigation using power diagrams. In *IEEE International Conference on Robotics and Automation* (pp. 1–8).
32. Hyun, N. P., Verriest, E. I., & Vela, P. A. (2015). Optimal obstacle avoidance trajectory generation using the root locus principle. In *IEEE Conference on Decision and Control* (pp. 626–631).
33. Sethian, J. (1999). *Level sets methods and fast marching methods*. Cambridge: Cambridge University Press.
34. Osher, S., & Fedkiw, R. (2003). *Level set methods and dynamic implicit surfaces*. Berlin: Springer.
35. Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4), 849–904.
36. Ross, I. M., & Karpenko, M. (2012). A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2), 182–197.
37. Andersson, J., Gillis, J., Horn, G., Rawlings, J., & Diehl, M. (2018, July). CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.
38. Deits, R., & Tedrake, R. (2015). Efficient mixed-integer planning for UAVs in cluttered environments. In *IEEE International Conference on Robotics and Automation* (pp. 42–49).
39. Hyun, N., Vela, P., & Verriest, E. (2017). A new framework for optimal path planning of rectangular robots using a weighted l_p norm. *IEEE Robotics and Automation Letters*, 2(3), 1460–1465.
40. Mukadam, M., Dong, J., Yan, X., Dellaert, F., & Boots, B. (2018). Continuous-time Gaussian process motion planning via probabilistic inference. *The International Journal of Robotics Research*, 37(11), 1319–1340.
41. Pham, Q. (2014). A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 30(6), 1533–1540.
42. Schulman, J., Ho, J., Lee, A. X., Abwal, I., Bradlow, H., & Abbeel, P. (2013). Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems*, 9(1), 1–10 (Citeseer).
43. Borenstein, J., & Koren, Y. (1989, September). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5), 1179–1187.

44. Borenstein, J., & Koren, Y. (1991, June). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3), 278–288.
45. Ulrich, I., & Borenstein, J. (2000, April). Vfh/sup */: Local obstacle avoidance with look-ahead verification. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)* (Vol. 3, pp. 2505–2511).
46. Nepal, K., Fine, A., Imam, N., Pietrocola, D., Robertson, N., & Ahlgren, D. J. (2009). Combining a modified vector field histogram algorithm and real-time image processing for unknown environment navigation. In *Intelligent Robots and Computer Vision XXVI: Algorithms and Techniques* (Vol. 7252, p. 72520G). Bellingham: International Society for Optics and Photonics.
47. An, D., & Wang, H. (2004, June). VPH: A new laser radar based obstacle avoidance method for intelligent mobile robots. In *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)* (Vol. 5, pp. 4681–4685).
48. Gong, J., Duan, Y., Man, Y., & Xiong, G. (2007, August). VPH+: An enhanced vector polar histogram method for mobile robot obstacle avoidance. In *2007 International Conference on Mechatronics and Automation* (pp. 2784–2788).
49. Minguez, J., & Montano, L. (2000, October). Nearness diagram navigation (ND): A new real time collision avoidance approach. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No. 00CH37113)* (Vol. 3, pp. 2094–2100).
50. Minguez, J., & Montano, L. (2004, February). Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1), 45–59.
51. Durham, J. W., & Bullo, F. (2008, September). Smooth nearness-diagram navigation. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 690–695).
52. Minguez, J., Montano, L., Simeon, T., & Alami, R. (2001, May). Global nearness diagram navigation (GND). In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)* (Vol. 1, pp. 33–39).
53. Mujahad, M., Fischer, D., Mertsching, B., & Jaddu, H. (2010, October). Closest Gap based (CG) reactive obstacle avoidance Navigation for highly cluttered environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1805–1812).
54. Mujahad, M., Fischer, D., & Mertsching, B. (2013, September). Safe Gap based (SG) reactive navigation for mobile robots. In *2013 European Conference on Mobile Robots (ECMR)* (pp. 325–330).
55. Sezer, V., & Gokasan, M. (2012). A novel obstacle avoidance algorithm: “Follow the Gap Method”. *Robotics and Autonomous Systems*, 60(9), 1123–1134. Available: <http://www.sciencedirect.com/science/article/pii/S0921889012000838>
56. Mujahad, M., & Mertsching, B. (2017, May). The admissible gap (AG) method for reactive collision avoidance. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1916–1921).
57. Bauer, R., Feiten, W., & Lawitzky, G. (1994). Steer angle fields: An approach to robust manoeuvring in cluttered, unknown environments. *Robotics and Autonomous Systems*, 12(3), 209–212.
58. Feiten, W., Bauer, R., & Lawitzky, G. (1994, May). Robust obstacle avoidance in unknown and cramped environments. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (Vol.3, pp. 2412–2417).
59. Buhmann, J., Burgard, W., Cremers, A., Fox, D., Hofmann, T., Schneider, F., et al. (1995). The Mobile Robot RHINO. *AI Magazine*, 16(1), 31–38.
60. Fox, D., Burgard, W., & Thrun, S. (1997, March). The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1), 23–33.
61. Stachniss, C., & Burgard, W. (2002). An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 1, pp. 508–513).

62. Brock, O., & Khatib, O. (1999). High-speed navigation using the global dynamic window approach. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)* (Vol. 1, pp. 341–346).
63. Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., & Konolige, K. (2010). The Office Marathon: Robust navigation in an indoor office environment. In *IEEE International Conference on Robotics and Automation* (pp. 300–307).
64. Simmons, R. (1996, April). The curvature-velocity method for local obstacle avoidance. In *Proceedings of IEEE International Conference on Robotics and Automation* (Vol. 4, pp. 3375–3382).
65. Ko, N. Y., & Simmons, R. (1998, October). The lane-curvature method for local obstacle avoidance. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)* (Vol. 3, pp. 1615–1621).
66. Quinlan, S., & Khatib, O. (1993, May). Elastic bands: Connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 802–807).
67. Khatib, M. (1996). *Sensor-based motion control for mobile robots*. Toulouse: LAAS-CNRS.
68. Roesmann, C., Feiten, W., Woesch, T., Hoffmann, F., & Bertram, T. (2012, May). Trajectory modification considering dynamic constraints of autonomous robots. In *ROBOTIK 2012; 7th German Conference on Robotics* (pp. 1–6).
69. Rösmann, C., Hoffmann, F., & Bertram, T. (2017). Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88, 142–153. <http://www.sciencedirect.com/science/article/pii/S0921889016300495>.
70. Lindner, L., Sergiyenko, O., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., et al. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: The International Journal of Robotics Research and Application*, 44(4), 532–541.
71. Sergiyenko, O., Ivanov, M., Tyrsa, V., Kartashov, V., Rivas-López, M., Hernández-Balbuena, D., et al. (2016). Data transferring model determination in robotic group. *Robotics and Autonomous Systems*, 83, 251–260.
72. Ivanov, M., Sergiyenko, O., Tyrsa, V., Mercorelli, P., Kartashov, V., Perdomo, W., et al. (2018, October). Individual scans fusion in virtual knowledge base for navigation of mobile robotic group with 3D TVS. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society* (pp. 3187–3192).
73. Maier, D., Hornung, A., & Bennewitz, M. (2012, November). Real-time navigation in 3D environments based on depth camera data. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)* (pp. 692–697).
74. Murray, D., & Jennings, C. (1997, May). Stereo vision based mapping and navigation for mobile robots. In *IEEE International Conference on Robotics Automation* (Vol. 2, pp. 1694–1699).
75. Biswas, J., & Veloso, M. (2012, May). Depth camera based indoor mobile robot localization and navigation. In *2012 IEEE International Conference on Robotics and Automation* (pp. 1697–1702).
76. Barry, A. J., Florence, P. R., & Tedrake, R. (2017). High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics*, 35(1), 52–68. <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21741>.
77. Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the ICRA 2010 workshop*.
78. Schmid, K., Tomic, T., Ruess, F., Hirschmuller, H., & Suppa, M. (2013). Stereo vision based indoor/outdoor navigation for flying robots. In *IEEE International Conference on Intelligent Robots and Systems* (pp. 3955–3962).
79. Lopez, B. T., & How, J. P. (2017, May). Aggressive 3-D collision avoidance for high-speed navigation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5759–5765).

80. Matthies, L., Brockers, R., Kuwata, Y., & Weiss, S. (2014, May). Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3242–3249).
81. Cao, T., Xiang, Z., & Liu, J. (2015, October). Perception in disparity: An efficient navigation framework for autonomous vehicles with stereo cameras. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2935–2948.
82. Matthies, L., Brockers, R., Kuwata, Y., & Weiss, S. (2014, May). Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3242–3249).
83. McGuire, K., de Croon, G., De Wagter, C., Tuyls, K., & Kappen, H. (2017, April). Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robotics and Automation Letters*, *2*(2), 1070–1076.
84. Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., & Roberts, J. (2005, August). Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3309–3316).
85. Cao, T., Xiang, Z., & Liu, J. (2015, October). Perception in disparity: An efficient navigation framework for autonomous vehicles with stereo cameras. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2935–2948.
86. Otte, M. W., Richardson, S. G., Mulligan, J., & Grudic, G. (2007, October). Local path planning in image space for autonomous robot navigation in unstructured environments. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2819–2826).
87. Cutting, J., Vishon, P., & Braren, P. (1995, October). How we avoid collisions with stationary and moving obstacles. *Psychological Review*, *102*(4), 627–651.
88. Fajen, B. R. (2013, July). Guiding locomotion in complex, dynamic environments. *Frontiers in Behavioral Neuroscience*, *7*, 85.
89. Vallar, G., Lobel, E., Galati, G., Berthoz, A., Pizzamiglio, L., & Le Bihan, D. (1999, January). A fronto-parietal system for computing the egocentric spatial frame of reference in humans. *Experimental Brain Research*, *124*(3), 281–286. <https://doi.org/10.1007/s002210050624>.
90. Dillon, M. R., Persichetti, A. S., Spelke, E. S., & Dilks, D. D. (2018). Places in the brain: Bridging layout and object geometry in scene-selective cortex. *Cerebral Cortex*, *28*(7), 2365–2374.
91. Dilks, D. D., Julian, J. B., Paunov, A. M., & Kanwisher, N. (2013). The occipital place area is causally and selectively involved in scene perception. *Journal of Neuroscience*, *33*(4), 1331–1336.
92. Greene, M. R., & Oliva, A. (2009). Recognition of natural scenes from global properties: Seeing the forest without representing the trees. *Cognitive Psychology*, *58*(2), 137–176.
93. Bonner, M., & Epstein, R. A. (2017). Coding of navigational affordances in the human visual system. *Proceedings of the National Academy of Sciences*, *114*(18), 4793–4798.
94. Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. Cambridge: MIT Press.
95. Galati, G., Lobel, E., Vallar, G., Berthoz, A., Pizzamiglio, L., & Le Bihan, D. (2000, July). The neural basis of egocentric and allocentric coding of space in humans: A functional magnetic resonance study. *Experimental Brain Research*, *133*(2), 156–164. <https://doi.org/10.1007/s002210000375>.
96. Wang, R. F., & Spelke, E. S. (2000). Updating egocentric representations in human navigation. *Cognition*, *77*(3), 215–250. <http://www.sciencedirect.com/science/article/pii/S0010027700001050>.
97. Spiers, H. J., & Maguire, E. A. (2007). A navigational guidance system in the human brain. *Hippocampus*, *17*(8), 618–626. <https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20298>.
98. Epstein, R. A. (2008). Parahippocampal and retrosplenial contributions to human spatial navigation. *Trends in Cognitive Sciences*, *12*(10), 388–396. <http://www.sciencedirect.com/science/article/pii/S136466130800199X>.

99. Wilber, A. A., Clark, B. J., Forster, T. C., Tatsuno, M., & McNaughton, B. L. (2014). Interaction of egocentric and world-centered reference frames in the rat posterior parietal cortex. *Journal of Neuroscience*, 34(16), 5431–5446. <http://www.jneurosci.org/content/34/16/5431>.
100. Smith, J., & Vela, P. (2017). Planning in perception space. In *IEEE International Conference on Robotics and Automation* (pp. 6204–6209).
101. Felzenszwalb, P., & Huttenlocher, D. (2012). Distance transforms of sampled functions. *Theory of Computing*, 8(19), 415–428.
102. Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 9–13 May 2011.
103. OctoMap, Github - octomap/octomap_mapping, Oct 2017. https://github.com/OctoMap/octomap_mapping.
104. Pan, J., Chitta, S., & Manocha, D. (2012, May). FCL: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation* (pp. 3859–3866).
105. Brockers, R., Fragoso, A., Matthies, L. (2016). Stereo vision-based obstacle avoidance for micro air vehicles using an egocylindrical image space representation. In *Micro- and Nanotechnology Sensors, Systems, and Applications VIII* (Vol. 9836). <http://dx.doi.org/10.1117/12.2224695>.
106. Asada, M., Tanaka, T., & Hosoda, K. (2016, April). Adaptive binocular visual servoing for independently moving target tracking. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)* (Vol. 3, pp. 2076–2081). <http://ieeexplore.ieee.org/document/846335/>.
107. Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1), 7–42. <http://dx.doi.org/10.1023/A:1014573219977>.
108. Hirschmuller, H. (2005, June). Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 807–814).
109. Jin, S., Cho, J., Pham, X. D., Lee, K. M., Park, S. Kim, M., et al. (2010, January). FPGA design and implementation of a real-time stereo vision system. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(1), 15–26.
110. Li, Y., Yang, C., Zhong, W., Li, Z., & Chen, S. (2017, January). High throughput hardware architecture for accurate semi-global matching. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)* (pp. 641–646).
111. Hernandez-Juarez, D., Chacón, A., Espinosa, A., Vázquez, D., Moure, J. C., & López, A. M. (2016). Embedded real-time stereo estimation via semi-global matching on the GPU. *Procedia Computer Science*, 80, 143–153.
112. Sprunk, C., Röwekämper, J., Parent, G., Spinello, L., Tipaldi, G. D., Burgard, W., et al. (2016). An experimental protocol for benchmarking robotic indoor navigation. In *Experimental Robotics* (pp. 487–504). Berlin: Springer.
113. Smith, J., Hwang, J., & Vela, P. (2018). Benchmark worlds for testing autonomous navigation algorithms. [Repository]. <http://github.com/ivalab/NavBench>.
114. Rockey, C. (2014). depthimage_to_laserscan. [Repository]. https://github.com/ros-perception/depthimage_to_laserscan.
115. Mihelich, P., Konolige, K., & Leibs, J. Github - ros-perception/image_pipeline/stereo_image_proc. https://github.com/ros-perception/image_pipeline.git.

Chapter 15

Autonomous Mobile Vehicle System Overview for Wheeled Ground Applications



**Luis Carlos Básaca-Preciado, Néstor Aarón Orozco-García,
Oscar A. Rosete-Beas, Miguel A. Ponce-Camacho, Kevin B. Ruiz-López,
Verónica A. Rojas-Mendizabal, Cristobal Capiz-Gómez,
Julio Francisco Hurtado-Campa, and Juan Manuel Terrazas-Gaynor**

Acronyms

2D	Two dimensional
3D	Three dimensional
AS/RS	Automated storage and retrieval system
BDS	BeiDou Navigation Satellite System
CML	Concurrent mapping and localization
CPR	Cycles per revolution
DGPS	Differential global positioning system
DoF	Degree of freedom
FMCW	Frequency-modulated continuous wave
FPGA	Field programmable gate array
GNSS	Global navigation satellite system
GPS	Global positioning system
IR	Infrared radiation
IRNSS	Indian Regional Navigation Satellite System
IT	Information technologies
IMU	Inertial measurement unit
LiDAR	Light detection and ranging
MAV	Micro aerial vehicle
MEO	Medium earth orbit

L. C. Básaca-Preciado (✉) · N. A. Orozco-García · O. A. Rosete-Beas · M. A. Ponce-Camacho
K. B. Ruiz-López · V. A. Rojas-Mendizabal · C. Capiz-Gómez · J. F. Hurtado-Campa
J. M. Terrazas-Gaynor
CETYS Universidad, Mexicali, Mexico
e-mail: luis.basaca@cetys.mx; nestor.orozco@cetys.mx; oscar.rosete@cetys.mx;
miguel.ponce@cetys.mx; kevinbennett.ruiz@cetys.edu.mx; veronica.rojas@cetys.mx;
cristobal.capiz@cetys.mx; juan.terrazas@cetys.mx

MUTCD	Manual on uniform traffic control devices
OD	Obstacle detection
PPR	Pulses per revolution
RADAR	Radio detection and ranging
ROV	Remotely operated vehicle
SAE	Society of Automotive Engineers
SLAM	Simultaneous localization and mapping
SoC	System on a chip
S/R	Storage and retrieval
ToF	Time of flight
TSR	Traffic sign recognition
UL	Unit load
VO	Visual odometry

15.1 Introduction

We are in the era of liquid modernity [1], where the fragility of markets lays on the power of the will of new generations. Post-centennial generation who will be born around 2020 or even already were born will grow up with the speed of 5G, Internet of things, and augmented reality, between other disruptive technologies yet unknown. In the era of innovation, products will be mostly services [2, 3] based on value propositions [4] that fit with the profile of a market that will use disruptive technologies. The use of autonomous electric cars depends on the evolution of business models based on open services. Clearly, with the advent of 5G technology, disruptive technologies will define the future market trends. One of those disruptive technologies will be autonomous electric cars.

Why electric autonomous cars? First off because autonomous cars or self-driving cars have all the features of business trends. Those businesses will be ruled by IT platforms and consumer profiles of the markets. Innovation is changing the future of successful business models. The next generations, after Z generation, will choose to buy or rent products and services online. Business based on IT platforms will ease the use of autonomous electric cars, in an era when big and expensive cars will be obsolete. Perhaps an icon like Amazon which has no factories can teach us the generational path to the new generation practices of individual transport. We understand why people prefer buying online. It is easier, cheaper, faster, and even functional. A single place to pick up from a variety of choices.

Individual transport as a service will change in the next 15 years, because of the natural timeline of future drivers. According to H. Chesbrough, the next generations of entrepreneurs will embrace new paradigms to develop businesses. So the commitment to the future of cars is a combination of two generations, centennials and post-centennials, and the use of business models based on IT platforms. The innovative business committed to fit customer needs that use to buy or rent online but with a power one thousand times faster than their previous generation because of 5G technologies. In the near future, innovation on products as a service will be a practice on a regular basis, so autonomous electric cars are in the center of the

trend. Products as a service are a trend in the business model of a lot of companies. Nowadays more and more, most of the revenues for enterprises come from services instead of products. Of course, technology is important, but the way the value proposition is delivered is more important.

So, at the end of the day, owning a product in the era of liquid modernity will not be relevant, but the service delivery will be. Perhaps disruptive business models in the near future will change the way people transport or get things transported on each day basis, leaving behind the madness of gasoline cars as a heavy, huge, and polluting artifact for museums. That day is not way long far, as well as future drivers will be born around 2020, or even were already born and right now are preparing to ask for a lift from an autonomous electric car through a 5G-IoT graphene device.

Therefore, the following sections present an introduction to what is considered an autonomous vehicle, its main components and applications. Afterward, more in-depth topics are addressed, such as perception (sensors), self-localization, environment mapping, trajectories planning; along with a case study of an electric autonomous vehicle which is currently in development at CETYS University Mexicali Campus in México, for transportation of people inside the campus. And in the end, a perspective on how innovative business models will change the future of cars and transport, along with our conclusions, is provided.

15.2 Fundamentals of Autonomous Mobile Vehicles

15.2.1 *Levels of Automation*

The current taxonomy and definitions of the automation of an on-road motor vehicle are governed by the J3016 standard, originally published by the Society of Automotive Engineers (SAE International) in 2014 and revised in 2016 and 2018 [5]; they are organized in a classification system consisting of six different levels (see Fig. 15.1):

- **Level 0**, No Automation: In this level of automation, the entire performance of the motor vehicle is under the authority of a human driver, this including all the tasks related to driving, such as the execution of steering, acceleration/deceleration maneuvers, monitoring and sensing the driving environment, and the fallback performance of the dynamic driving task, which is the operational and tactical aspects of driving a car, but not the strategic ones, such as choosing a destination. Most commercial motorized vehicles fall under this category of being driven fully by humans.
- **Level 1**, Driver Assistance: in this level of automation, the entire performance of the motor vehicle still falls under the authority of a human driver, but under certain circumstances, and using information of the driving environment, an integrated assistance system can aid the human driver in certain tasks such as steering, or accelerating/decelerating, all under the assumption the driver will take care of the remaining aspects of the drive. It must be noted that a motor vehicle may have



SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
What do these features do?	These are driver support features			These are automated driving features		
	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none"> • automatic emergency braking • blind spot warning • lane departure warning 	<ul style="list-style-type: none"> • lane centering OR • adaptive cruise control 	<ul style="list-style-type: none"> • lane centering AND • adaptive cruise control at the same time 	<ul style="list-style-type: none"> • traffic jam chauffeur 	<ul style="list-style-type: none"> • local driverless taxi • pedals/steering wheel may or may not be installed 	<ul style="list-style-type: none"> • same as level 4, but feature can drive everywhere in all conditions

Fig. 15.1 Levels of driving automation by SAE [5]

one or more tasks aided by said systems, depending on the number of system capabilities, or driving modes.

- **Level 2, Partial Automation:** In this level of automation, a driving mode executes a specific task of the steering, and acceleration/deceleration ADAS capabilities. Certain parts of the driving experience can be automated in this level, with systems such as self-parking, following a car at a certain distance and staying in lane. The driver is always in control of the car.
- **Level 3, Conditional Automation:** In this level of automation, there is an autonomous driving mode which uses data of the driving environment obtained through an array of advanced sensors, typically ultrasonic, RADARs, LiDARs, and machine vision. These cars can make decisions and perform the same and more advanced tasks than the previous level. Nevertheless, this level is still not fail-safe, and drivers must be ready to take control of the vehicle at all times if the situation requires it.
- **Level 4, High Automation:** In this level of automation, all capabilities of the previous level are available, the difference is that these systems possess more advanced technology, that is, redundant systems which allows the vehicle to handle any situation or system failure on its own. Passengers are not required to supervise but may take control of the vehicle if desired or preferred. These features are limited only to certain areas or conditions and if the required conditions are not met, autonomous features will be limited or will not engage.
- **Level 5, Full Automation:** In this level of automation (which is the highest), similarly to level 4, passengers do not need to supervise and the vehicle can

perform full autonomous driving in any conditions while adapting to any driving situation and guaranteeing the safety of passengers, pedestrians, and other drivers. These features are not limited to specific areas or conditions. These vehicles can drive everywhere while offering a more responsive and refined experience for the passenger on board.

Currently, there is no autonomous vehicle that is level 5 but companies such as Google Waymo and Tesla (among others) are working hard to reach this level and it is only a matter of time before they do. See Sect. 15.2.3 for a commercial autonomous vehicle application, the Google Waymo's self-driving taxi.

15.2.2 Main Components

The components of an autonomous vehicle vary depending on the application of the vehicle; however, there are certain main components that are indispensable inside a vehicle to be able to achieve a level of autonomy.

Central or main computer: This computer can be from a microcontroller or a Field Programmable Gate Array (FPGA) to a complete computer or a mini computer. It should be mentioned that depending on the design architecture implemented in the vehicle, it is possible to find more than one computer connected and working for the central computer. Additionally, it is responsible for communications between electronic devices, sensor reading, data processing, trajectory calculation, control of vehicle actuators, among other functions.

Some designs of autonomous low-end and low-cost vehicles that are in development can get to use mini or micro computers such as the raspberry pi [6] or the zini [7]. However most of the designs of autonomous vehicles for industry or commercial use full computers on board, with multicore processors and dedicated graphics cards.

The sensors are some of the indispensable and most important components of an autonomous vehicle because the sensors perform the function of measuring physical variables or desired parameters such as angular position, speed, temperature, acceleration, position, distance, among others. The measurement of these parameters and the analysis of the information they provide is called perception and is basically the ability of a vehicle to be aware of itself and what is happening around it, this being the first step to move toward the autonomy.

The most important sensors found in autonomous vehicles are the following:

Odometry sensors that can be optical or magnetic encoders to measure the angular position and speed of the vehicle's wheels, in order to have the ability to calculate the displacement of the vehicle in real time.

Inertial measurement unit (IMU) is a device that can hold up to three sensors of three dimensions each simultaneously, commonly to each dimension of measurement known as degree of freedom (DoF). The sensors contained in an IMU are a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer. The signals

provided by these sensors are used to calculate the angular position and orientation of the vehicle.

Light Detection and Ranging or LiDAR is an advanced sensor used to create digital maps of the surrounding environment of the vehicle, these sensors are complex optoelectronic systems that use moving mechanical parts to control the rotation and speed of 1–128 lasers that are inside the device. The signals of the lasers are emitted and reflected by obstacle surfaces or objects in the field of view of the device; afterward, the reflection is detected and through the technique time of flight or ToF, the distance from the emitting device to the laser reflexion point (on the object surface) is calculated, allowing effectively the use of this information to create digital maps or point clouds of the environment surrounding the LiDAR (see Fig. 15.2). Later, in the perception section of this chapter, the use, advantages, most common models of LiDAR employed in the autonomous vehicle industry, and its characteristics are studied.

The video cameras, similar to the LiDAR, are used to perceive the environment surrounding the vehicle; however, due to the nature of how a video camera operates, the information that can be extracted from the data of the same is very different, and although there are techniques to calculate the depth of some feature of an image, such as stereovision [8], the LiDAR is the most appropriate and fastest to perform the task of creating digital maps of the environment. The main function of a camera is to detect features and patterns within the images captured in video and the identification, location, and classification of objects or figures of interest such as people, other vehicles, and dynamic objects, the lines that divide the lanes in a street or highway, the traffic signals, or landmarks used by the vehicle for the correction of its position, among others.

Ultrasonic sensors are mainly used to measure distances or detect the presence of objects within a range of vision. These sensors use a transmitter and an ultrasonic wave receiver and the ToF principle to calculate the distance of the detected object(s). Its main advantage is its wide field of view, which makes this type of sensor able to detect obstacles in close proximity to the vehicle.

Infrared sensors have various applications, but their main function is to detect the presence of objects at a predefined distance using an emitter or transmitter and the detection of the reflection of light as a signal. In Fig. 15.3 the basic principle of operation is shown.

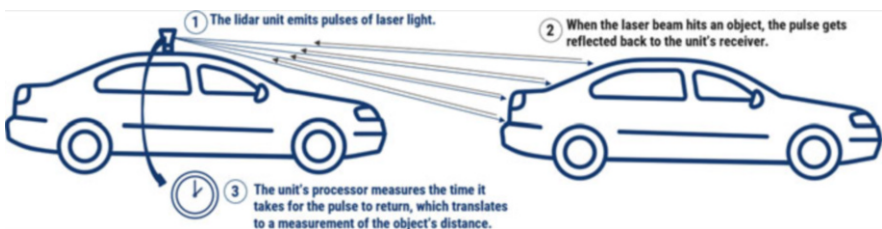


Fig. 15.2 Simplified diagram of how automotive lidar works

Fig. 15.3 Infrared sensor basic operation principle

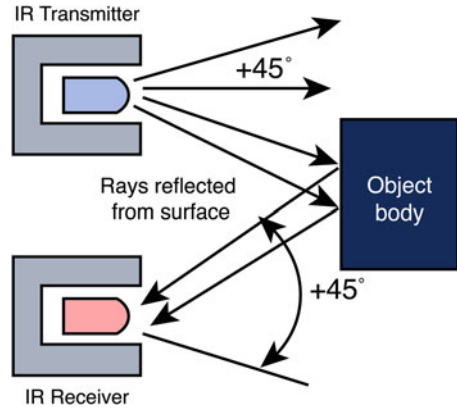


Fig. 15.4 Satellite GPS III launched in 2018 [9]

As a disadvantage, these sensors have by nature a limited range of vision, unlike ultrasonic sensors. However, they also have a great advantage, and this is their high response speed compared to other sensors such as those based on a principle of resistive transduction. Because of this, these types of sensors are a viable option to quickly detect obstacles in a limited range of vision.

The Global Positioning System or GPS is fundamental in an autonomous vehicle because this system, as its name indicates, is able to calculate its current position in any part of the world, using satellites as a reference (see Fig. 15.4).

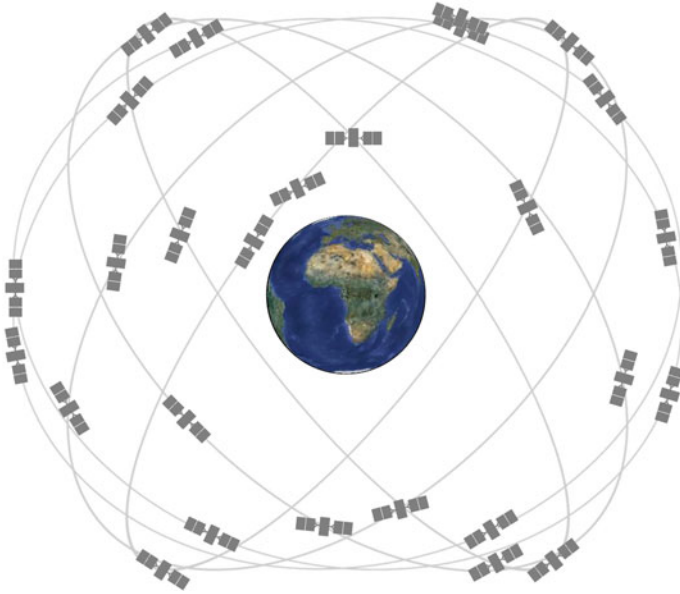


Fig. 15.5 Twenty-four-slot satellite constellation, as defined in the GPS Performance Standard [9]

Devices equipped with GPS receive signals from a constellation of satellites which contains at least 24 operational satellites that are flying in medium earth orbit (MEO), at approximately 20,000 km altitude (see Fig. 15.5).

As of January 9, 2019, there were a total of 31 operational satellites in the GPS constellation. The GPS satellite expansion gave as a result improved coverage in most parts of the world [9].

The term GPS is owned by the United States; however, there are other satellite navigation systems which use their own satellite constellation such as BeiDou Navigation Satellite System (BDS) from China [10], the Indian Regional Navigation Satellite System (IRNSS) [11], Galileo from the European Union [12], among others.

Radio Detection and Ranging (RADAR or radar) is a well-known and studied technology developed originally for military applications during the Second World War. In autonomous vehicles, radars are active radio frequency devices used to determine the location, speed, and direction of obstacles or targets of interest within its coverage area [13].

Radars work similar to the LiDAR sensors, the main difference between these systems is that radars use radio waves instead of laser; however, there are advantages and disadvantages between them. LiDARs can detect smaller objects and can build an exact 3D image of an object, while RADARs cannot. On the other hand, RADARs can operate in cloudy weather conditions and possess a longer operating distance, whereas LiDARs have greater difficulty in these conditions [14]. In Fig. 15.6, a dual RADAR configuration for an autonomous vehicle is shown; this configuration combines two RADARs in one, which generates some advantages such as a broader

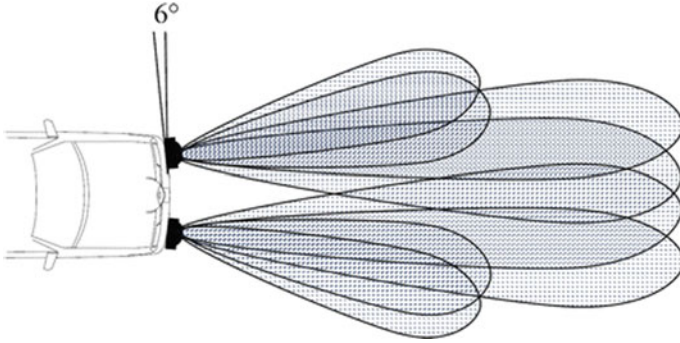


Fig. 15.6 Dual RADAR configuration for autonomous vehicle [15]

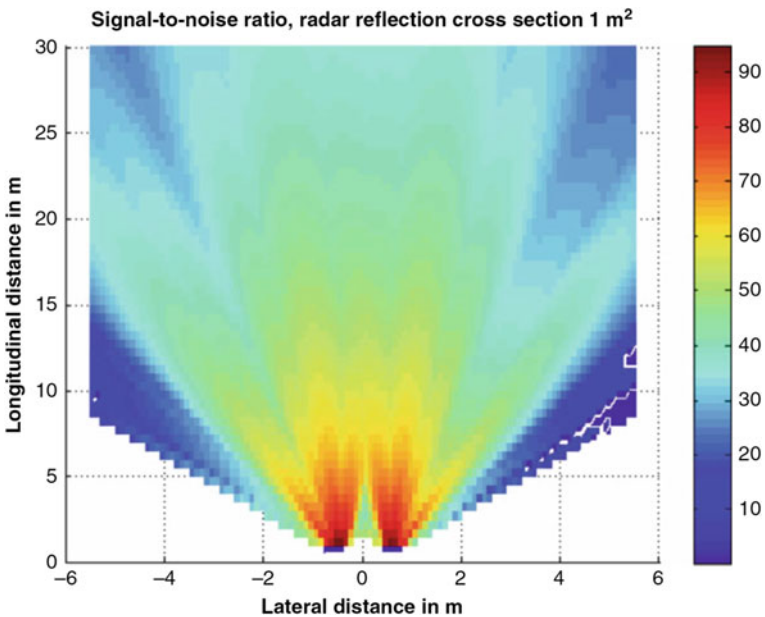


Fig. 15.7 Detection coverage of the dual RADAR configuration [15]

coverage, fault detections, and an improved signal processing due to the overlap of both sensors [15] (see Fig. 15.7).

The components mentioned above are the most important or indispensable in any autonomous vehicle development that wants to achieve a high level of autonomy; however, they are not the only ones and new solutions, and technologies for autonomous vehicles are constantly being developed. In the next sections of this chapter, you will find more detailed and in-depth information on the functioning of these components.

15.2.3 Applications

Due to the fact that autonomous vehicles are wheeled mobile platforms, they share some of the applications that conventional land vehicles, such as automobiles, tuggers, cranes, and carts have, with the added benefits that automation brings in specific scenarios, such as higher precision, endurance, worker safety, and efficiency, among others. This section will explore common modern autonomous vehicle applications across different industries.

15.2.3.1 Automated Storage and Retrieval System (AS/RS)

Unit load (UL) and storage and retrieval (S/R) systems are a critical link for supply chains in global markets. The introduction of crane and tugger-based technologies allowed for automated vehicle systems to be introduced in UL S/R system configurations, such as the one presented in Fig. 15.8, finding wide implementation in modern storage facilities, built or renovated after 1994, year in which AS/RS systems had a significant increase in implementation in distribution environments in the United States [17]. The main benefits of AS/RS over conventional UL R/S systems are:

- Considerable savings in labor costs.
- Higher efficiency of floor space and storage layout configuration.
- Increased working reliability.
- Reduced error rates and product waste.

Nevertheless, it must be noted that since AS/RS has a higher level of technology integration than conventional UL R/S systems, they tend to be significantly costlier, as a high investment is needed in order to establish the desired level of automation, and a proper closed loop control system; similarly, because of the systematic approach

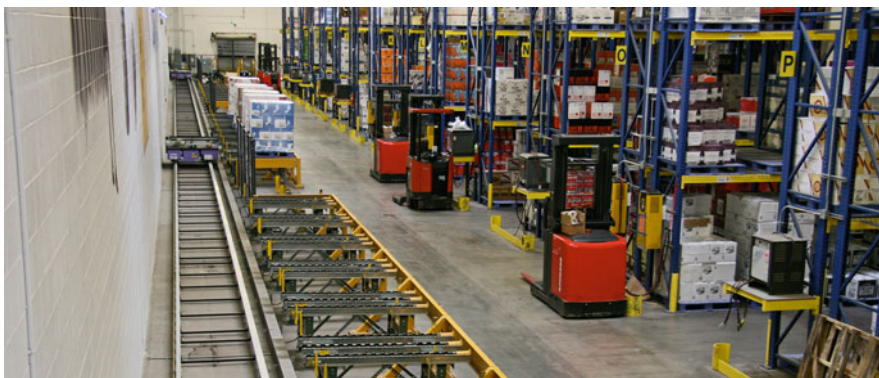


Fig. 15.8 Distribution center with an implemented AS/RS [16]

that automation entails, it tends to be significantly less flexible than conventional, non-automated systems [17, 18].

15.2.3.2 Mobile Industrial Robots

Mobile industrial robots are cooperative robots that are specifically programmed to work in an industrial setting (see Fig. 15.9). Their main functionalities are focused on increasing efficiency of industrial processes and low-value logistics tasks, increasing the speed of internal transportation through the use of automation and deep learning, and providing a cost-efficient solution for improving the overall reliability of the industrial environment by using a set of very powerful and precise sensors.

15.2.3.3 Commercial Autonomous Vehicles

One of the most common applications of autonomous vehicles, “self-driving cars” are modern solutions that are commercially available with the implementation of automation up to a certain degree. Most commercially available self-driving vehicles fall between the second and third level of automation, with some vehicles in the fourth level currently undergoing trial sessions [19] (see Fig. 15.10). Due to the sensibility of the working environment of these vehicles, namely the fact that human lives are

Fig. 15.9 MiR500 Mobile Robot for industrial applications [16]



Fig. 15.10 Google Waymo’s self-driving taxi is in service in Arizona in December 2018 [19]



at stake, the trial period of these systems are considerably more rigorous than for other autonomous land vehicle applications, this with the intention of having greater passenger security, and reducing the amount of disengage incidents, when the driver has to take control of the autonomous system. Due to current level of automation of commercially available self-driving vehicles, it must be noted that, for safety reasons, the driver must be alert during the entire driving task, in case a disengage incident occurs.

15.2.3.4 Autonomous Vacuum Cleaners

Robotic vacuum cleaners, such as the Roomba i7 from iRobot (see Fig. 15.11), have gone from simple closed loop systems to full-fledged deep learning autonomous systems. These autonomous robotic systems use an integrated array of specialized sensors that allow them to navigate through cleaning environments, focusing on avoiding obstacles, such as furniture, while maximizing the efficiency of surface-cleaning through the use of proprietary algorithms. Modern iterations of autonomous vacuum cleaning robots include smart mapping features that automatically generate a layout of the cleaning environment, and labels it accordingly for future cleaning tasks; these systems typically have some degree of IoT integration in the form of an application or through communication with dedicated personal home assistants such as Google Home or Amazon's Alexa.

It must be noted that the applications explored in this section were chosen based on relevance, and notability among the general public. As more levels of automation are achieved, more relevant applications of autonomous systems will emerge; as such, there are other relevant applications currently in development for autonomous land vehicles in sectors such as agriculture, logistics, entertainment, and healthcare.

Fig. 15.11 iRobot Roomba i7, autonomous robotic vacuum [20]



15.3 Perception

15.3.1 Environment Sensing

In a general perspective, sensors can be grouped according to the functions they provide. Internal vehicle state sensors provide information about the current operation and state of the vehicle, including lower-level functions such as engine operations and higher-level states such as vehicle motion and position. External environment sensors provide information about the world outside the vehicle, potentially including road and lane information, the location and motion of other vehicles, and stationary physical objects in the world. Finally, driver state and intention sensors provide information about the state or intentions of the driver. These sensors can include seat occupancy and passenger weight (pressure or infrared sensors), audio sensors, internal cameras, eye trackers, breath alcohol sensors, and haptic transducers [21].

There are a number of issues when designing the external environment sensing suite for an autonomous vehicle. The system objectives and requirements are obviously a direct influence in the design process. For example, highway driving is a much more structured environment than off-road driving, and that structure can be exploited in sensor and control system design. On the other hand, the urban environment, which may consist of irregular and changing road networks and vehicles and pedestrians behaving unpredictably, is much less structured than highway driving and may require significantly more sensing capability. Robustness and safe performance of hardware and software are obviously required for both off-road driving and on-road production automobiles. Redundancy in sensing modalities is also a highly desired feature, especially in a less structured, more uncertain environment, but the cost of many of the sensors used on research vehicles would be prohibitive for a commercial passenger vehicle application. The level of autonomy with respect to the human driver is also a significant design issue in a passenger vehicle system. The degree to which the driver is to be part of the sensing and control loop is a design decision driven by both technical and nontechnical (i.e., marketing and legal) considerations.

A number of different sensors have been developed for sensing the external environment of an autonomous vehicle. Many have been developed initially for safety warning or safety augmentation systems that are now being deployed on some high-end vehicles. Among the array of different sensors available for establishing an environment sensing system, the following are worth mentioning:

15.3.1.1 LiDAR Sensor

In principle, a LiDAR consists of a transmitter and a receiver. Short light pulses with lengths of a few to several hundred nanoseconds and specific spectral properties are generated by the laser. Many systems apply a beam expander within the transmitter

unit to reduce the divergence of the light beam before it is sent out into the atmosphere. At the receiver end, a telescope collects the photons backscattered from the atmosphere. It is usually followed by an optical analyzing system which, depending on the application, selects specific wavelengths or polarization states out of the collected light. The selected radiation is directed onto a detector, where the received optical signal is converted into an electrical signal. The intensity of this signal in its dependence on the time elapsed after the transmission of the laser pulse is determined electronically and stored in a computer [22].

LiDARs carry out sensing by using their own emitted light, and therefore, they are not sensitive to environmental illumination. Road detection systems that rely on this type of sensor can then, in principle, provide the same level of accuracy across the full spectrum of light conditions experienced in daily driving; therefore, they are particularly suitable for achieving higher levels of driving automation [23].

Among the existing commercial LiDAR, the following devices are worth mentioning:

- **Velodyne VLP-16 LiDAR**

Capable of delivering the most accurate real-time 3D data that consists of the creation of a full 360° environmental point cloud view (Fig. 15.12).

- **Velodyne HDL-64E**

The HDL-64E LiDAR sensor includes 64 channels with an accuracy of ± 2 cm and a 100–120 m range, designed for obstacle detection and navigation of autonomous ground vehicles and marine vessels (Fig. 15.13).

Recently a new LiDAR technology arrived, solid state LiDARs, which have the advantage that they are smaller and, importantly, they eliminate moving parts involved in the optical mechanisms as seen in Fig. 15.14, which may enable their mass manufacture thereby bringing manufacturing costs down [27].

One of the most remarkable solid-state LiDARs InnovizOne with a detection range of 250 m, a horizontal field of view of 120° and a vertical field of view of 25° which comes at a fraction of the cost of the equivalent electromechanical solution (Fig. 15.15).

Fig. 15.12 Velodyne VLP-16 LiDAR [24]



Fig. 15.13 Velodyne HDL-64E LiDAR [25]

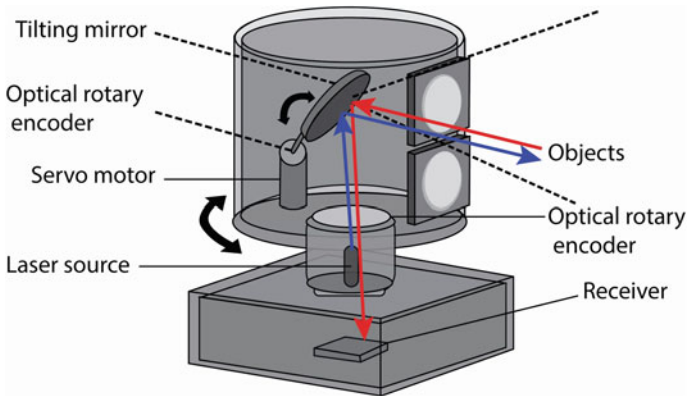


Fig. 15.14 Diagram of the LiDAR optics and encoders [26]

Fig. 15.15 InnovizOne solid-state LiDAR [28]



15.3.1.2 Kinect Sensor

The Kinect device is a horizontal bar composed of multiple sensors connected to a base with a motorized pivot. Looking at the Kinect sensor from the front, from the outside, it is possible to identify the infrared radiation (IR) projector, the RGB camera,

Fig. 15.16 Kinect sensor
[29]



and the depth camera. An array of four microphones, a three axis accelerometer, and the tilt motor are arranged inside the plastic case (Fig. 15.16).

The device is connected to a PC through a USB 2.0 cable. It needs an external power supply in order to work because USB ports do not provide enough power. The IR projector is the device that Kinect uses for projecting the IR rays that are used for computing the depth data. The IR projector, which from the outside looks like a common camera, is a laser emitter that constantly projects a pattern of structured IR dots at a wavelength around 830 nm [30].

15.3.2 *Obstacle Detection and Tracking*

Obstacle detection (OD) is one of the main control system components in autonomous vehicles, since a reliable perception of the real world is a key-feature of any obstacle detection system for dynamic environments. In recent years, most of the historical approaches in literature have been readjusted in the framework of stereo vision and other 3D perception technologies (e.g., LiDAR), and important results have been provided by several experiments on autonomous ground vehicles.

In order to achieve a good performance, most of the OD algorithms take some assumptions about the ground or about the approximated free space on it. Algorithms based on stereo vision and other 2D/3D sensors. Each obstacle detection system is focused on a specific tessellation or clustering strategy; hence, they have been categorized into four main models [31].

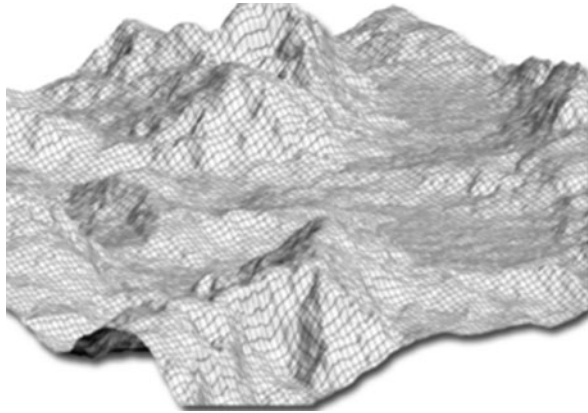
15.3.2.1 **Probabilistic Occupancy Map**

Occupancy grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, under the assumption that the robot pose is known. The basic idea of the occupancy grids is to represent the map as a field of random variables arranged in an evenly spaced grid [32]. As seen in Fig. 15.17, each random variable is binary and corresponds to the occupancy of the location it covers.

Fig. 15.17 Occupancy grid map obtained from ultrasound data [33]



Fig. 15.18 Landscape digital elevation map [35]



15.3.2.2 Digital Elevation Map

The concept of digital terrain model was defined as a digital (numerical) representation of the terrain. Other alternatives to this term have been used like digital elevation models, digital height models, digital ground models, as well as digital terrain elevation models [34]. Figure 15.18 shows the case of specific digital elevation models which refers to a digital representation of the terrain on a height above a given level, especially that of the sea.

15.3.2.3 Scene Flow Segmentation

Scene flow estimation is the challenging problem of calculating geometry and motion at the same time. By considering images from one view point, scene flow estimation is underdetermined. Also, a moving camera still creates ambiguities between camera and scene motion. Only after introducing additional cameras, this situation can be resolved and increase the robustness of the system [36]. Figure 15.19 shows the application of scene segmentation techniques on vehicle detection.

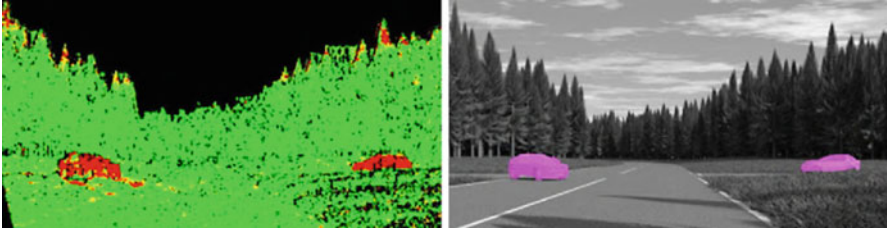


Fig. 15.19 Scene flow segmentation obstacle detection [37]



Fig. 15.20 Epipolar geometry-based clustering applied to an urban traffic scene [39]

15.3.2.4 Geometry-Based Clusters

These algorithms are based on a search method that clusters the sensor measurements using a specific geometry model like the double-cone model [38]. As seen in Fig. 15.20, geometry-based clustering techniques can be applied to cities to detect obstacles.

15.3.3 Traffic Signs

Traffic signs are markers placed along roads to inform drivers about either road conditions and restrictions or which direction to go. They communicate a wealth of information but are designed to do so efficiently and at a glance. This also means that they are often designed to stand out from their surroundings, making the detection task fairly well defined. The designs of traffic signs are standardized through laws but differ across the world. In Europe, many signs are standardized via the Vienna Convention on Road Signs and Signals. In the United States, traffic signs are regulated by the Manual on Uniform Traffic Control Devices (MUTCD).

While signs are well defined through laws and designed to be easy to spot, there are still plenty of challenges for Traffic Sign Recognition (TSR) systems [40].

1. Signs are similar within or across categories.
2. Signs may have faded or are dirty so they are no longer their original color.
3. The sign post may be bent, and therefore, the sign is no longer orthogonal to the road.
4. Lighting conditions may make color detection unreliable.
5. Low contrast may make shape detection hard.
6. In cluttered urban environments, other objects may look very similar to signs.
7. There may be varying weather conditions.

Among the challenges that must be considered for the adequate detection of traffic signs, lighting, viewpoints, and weather conditions are present. These considerations suggest that relying solely on color is problematic; therefore, shape information is useful for sign detection. An example of the use of shape information is seen in [41].

Traffic sign detection methods include the following [40]:

- Learning-based traffic sign detectors
- Segmentation methods
- Hough transform derivatives

15.3.4 Landmarks

Landmarks are features of an environment or scenery that can be easily recognized and thus can enable someone to locate his or her spatial position within a given location. Landmarks in an urban environment tend to be human-made features, including but not limited to:

- Street name signs
- House number plates
- Sidewalk symbology
- Traffic signs
- Billboards
- Buildings
- Monuments
- Other easily recognizable urban features

This is due to the fact that, in urban environments, human-made features tend to be more permanently established and recognizable than nature-based features [42].

Autonomous vehicles often include landmark-based navigation systems. Landmark-based navigation systems serve to complement the data obtained from the main perception and environment sensing system to determine the vehicle positioning within a known environment or to determine the most efficient route to a specific landmark; this can be done by establishing spatial relationships, with their corresponding constraints. Constraints can be established through the measurement of the spatial relationship between a known landmark and the vehicle itself or through a geometric relationship such as collinearity or coplanarity [42, 43].

Vision systems are often implemented to recognize landmarks in an urban environment; the algorithms used for this purpose execute feature extractions using resources such as the bag-of-words framework. Data obtained from the vision system can be then compared to images in a cloud-based database server to recognize a specific landmark. The reliability of this method and the integrity of the data reported to the server can be further improved through the integration of other sensor systems; an example would be the validation of the outlines provided by the vision system through correction using LiDAR point clouds [43, 44].

It must be noted that the main limitation of autonomous navigation using a landmark-based system is the lack of a comprehensive and readily available database of every single landmark in a city and the fact that its query accuracy relies solely on the validity and actualization of entries in the database. While companies such as Google and Waze resort to user-fed databases that are continuously being updated, it is important to note that they can be immediately accurate only up to a certain extent because of factors such as human error, deliberated entry of false data due to malicious intent, landmark update time, or simply the sheer amount of landmarks in a city [44].

15.4 Localization and Map Building

In an autonomous vehicle, perception is the process to extract different features or elements from the sensors that can describe the environment so we can make a model (map). But what if we have additional information from a knowledge base that can be related to the map of where the robot is located, we can use all this information from the sensors, and the map to determine robot real location. Thus we can describe Mobile Robot Localization as a process used to figure out the pose of a robot; the pose of a robot is the position and orientation, with respect to a model of the environment, and this process is also known as position estimation or position tracking [28].

Mobile Robots have sensors which are usually affected by noise, that is why the robot must be capable of processing uncertainty affected data, and errors detected in sensor readings due to noise are not statistically independent. Actually these errors increase their appearance as long as the number of measurements increases, and the correspondence problem and data association problem raise the question whether the taken measurements are from the same place or object. Correspondence problem can be produced by similarities in the environment or sensor range limits [45].

The concept of robot state is used to describe its pose and also some other magnitudes like robot velocity, sensor biases, and calibration. The map is a description of important features like landmark positions and different physical constraints like obstacles in the environment in which the robot is moving [46].

The process of localization starts by assigning a coordinate system to a map which is called a global frame, another coordinate system that is assigned to a mobile robot whose position and orientation moves along with the robot's pose;

this is called local frame. We have to establish a relationship between the local and the global frames, and this is expressed as a transformation matrix. If we know this coordinate transformation, we can locate position of different objects like landmarks and obstacles with respect to a robot local frame or a global frame, which is a necessary task for robot navigation.

This transformation matrix can be easily obtained if we know the mobile robot’s pose with respect to the global frame as can be shown at Eq. (15.1), as a function of coordinates x , y and orientation θ .

$$M_L^G = (x, y, \theta)^T \tag{15.1}$$

The problem is that robot’s pose cannot be obtained directly from sensors, and it has to be estimated from data measurements taken from the environment [28]. This is because all measurements are affected by sensors noise, aliasing, and also the actuators that carry some noise and inaccuracies.

A general schematic for the mobile robot localization is presented in Fig. 15.21. Position estimation is carried out by probabilistic algorithms, and these types of processes have received a great research attention over the last years [47].

Localization is enforced with a collection of map representations; some of these are [28]:

- (a) Feature-based maps. Specify the shape of the environment at the specific locations, this means the location of the objects contained in the maps.
- (b) Location-based maps. These are volumetrics, they offer a label for any location in the environment, also have information about objects and absences of objects contained in the map.

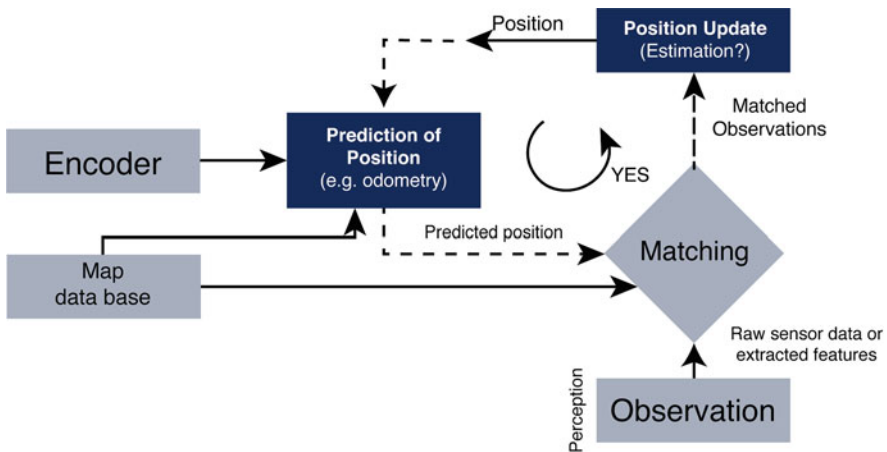


Fig. 15.21 Schematic for mobile robot localization process [47]

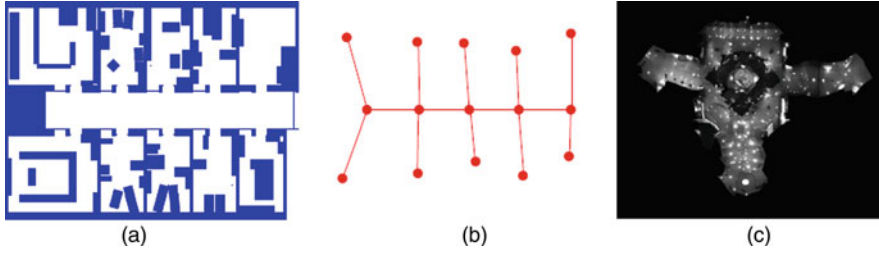


Fig. 15.22 Examples of maps used for robot localization process: (a) 2D metric map, (b) graph topological map, and (c) mosaic from ceiling [28]

(c) Occupancy grid maps. This is a type of location-based map, they assign a binary value to a x-y coordinate grid associated in order to know if it is occupied with an object. Some instances used by this representation are shown at Fig. 15.22: a 2D metric map at (a), Graph-like topological map at (b), and Image Mosaic from ceiling at (c).

There are many types of maps used for localization and navigation algorithms. The localization problem assumes that an accurate map is available.

Localization problem challenge can be classified by the nature of the environment and the initial knowledge that a robot possesses about its localization. So we present four different criteria considered to solve the location problem [28].

- (a) Local and global localization. Localization problems can be classified by the type of information that is available initially or during navigation.
- Local localization assumes that the initial pose of robot is known so the difference between real robot pose and calculated is small.
 - Global localization. The initial pose of the robot is unknown. The robot is placed somewhere in its environment but does not know where.
- (b) Static and dynamic environments. If the robot is the only object moving on its environment, then it is called static; hence, dynamic environments consist of multiple objects moving during robot navigation.
- (c) Passive and active localization approaches.
- Passive localization approach consists of a robot navigation which is not aimed to improve localization algorithm.
 - Active localization approach controls the robot navigation in order to minimize localization error, but in practice these algorithms combine designated tasks fulfillment with localization goals.
- (d) Single and multi-robot localization.

15.4.1 Mapping Sensors

15.4.1.1 LiDAR Sensor

As seen in Sect. 15.3.1, a LiDAR (also known as optical radar) is also known as a ToF sensor that consists of a transmitter that illuminates a target with a laser beam, and there is a receiver capable of detecting the returning component of light coaxial with the transmitted light [47]. The transmitter produces a pulse of light aimed at the environment; if there is a target, a component of the pulse returns, a mechanical mirror system is implemented to sweep the light beams and scan a required space which can be planar or three dimensional. The range between source and target can be measured by different methods:

- (a) ToF, this is sending the pulse beam and detecting the time it takes to be reflected and returned to the receiver. The range R is calculated by the following expression at Eq. (15.2) [48].

$$R = \frac{1}{2}c\Delta t \quad (15.2)$$

where c is the speed of light and Δt is the ToF of the pulse

- (b) Measure the beat frequency between a Frequency Modulated Continuous Wave (FMCW) and its received reflection [47].
 (c) Measure the phase shift of the reflected light. Phase shift laser scanners are more accurate, but range is shorter [48].

The phase shift measurement sensor transmits an amplitude modulated of light at a known frequency, the wavelength of the modulating signal poses the next relationship with the speed of light expressed at Eq. (15.3) [47].

$$c = f \cdot \lambda \quad (15.3)$$

where c = speed of light

λ = wavelength of the modulating signal

f = modulating frequency

The total distance covered by the emitted light is given by Eq. (15.4):

$$D' = L + 2D = L + \frac{\theta}{2\pi}\lambda \quad (15.4)$$

where D' = total distance covered by the emitted light

L = distance between transmitter and phase measurement sensor

D = distance from beam splitter and the target

θ = electronically measured phase difference

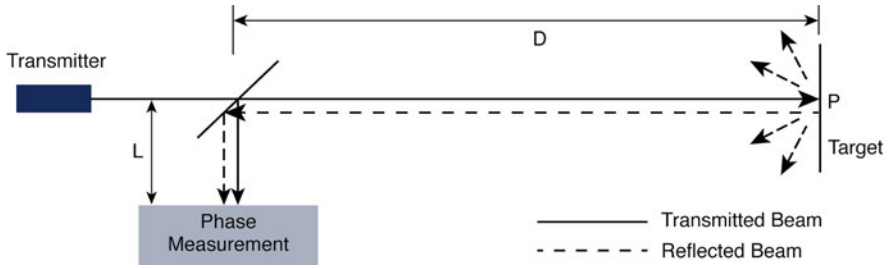


Fig. 15.23 Phase shift measurement [47]

The required distance beam between the beam splitter and the target is given by Eq. (15.5):

$$D = \frac{\lambda}{4\pi} \theta \quad (15.5)$$

Figure 15.23 shows a schematic of a laser range measurement system using the method of phase shift; a light beam is collimated and transmitted from the transmitter, and hits the target point P in the environment and is reflected back to a receiver.

Multi-beam laser scanners are used in robot mobile applications; these systems use an array of sensor applying the method of ToF measurement, and this array of laser beams scan and measure simultaneously and generate in parallel a wide amount of coordinate points which is called point cloud and represent the environment with high fidelity. As a consequence multi-beam laser scanners are used to capture wide regions and landscapes and are used for mapping applications. The laser pulse repetition rate (RRR), in combination with the scanning mirror deflecting pattern, determines the LiDAR data collection rate [48].

15.4.1.2 Kinect Sensor

Laser scanners have obtained great relevance, making 3D models of the environment. Some parameters of laser scanners are very important to differentiate one system or another, parameters such as measurement rate, range, and accuracy are considered for this task. Autonomous mobile vehicles use laser scanners to obtain a model of their environment. Since this task is performed in real time, they need a high scanning speed and also require a medium range (30 m maximum) and low accuracy (3–5 cm) capabilities. Video Game industry has made a contribution to the autonomous robots, since they have developed the so-called gaming sensors, which are devices that have a high measurement speed and an intermediate range (between 1 and 5 m) [49].

Asus and Microsoft companies have developed Xtion and Kinect systems, respectively; both use the triangulation method for measuring targets and have become very popular due to the large number of developers who have done projects with these systems and have extended the potential of their applications to other fields

different from entertainment, such as facial recognition, virtual learning, forensic science, and of course navigation of autonomous vehicles [49].

The “range sensors” provide a way to capture measurements of space and objects from mobile robot’s environment. Some important requirements must be considered to select this type of sensors, that is, size, weight, resolution, refresh rate, field of view, and robustness to external light conditions.

Kinect is a high-speed sensor for 3D measurement applications and has a fairly accurate resolution and many applications in robotics and navigation. The first generation of Kinect uses triangulation technologies to make the measurement; however, the fundamental structure of this technology is not suitable in sunlight conditions, and this limits its scope only for indoor application [50].

A second generation of these sensors arose with the appearance of the Kinect v2, which is based on the ToF measurement principles (like LiDAR), which allows it to be used outdoors and in sunlight. It offers higher resolution and a larger field of view compared to that of the first generation.

The Kinect v2 depth sensor has a strobe IR light which illuminates the scene and is reflected by obstacles, and the ToF for each pixel is registered by an IR camera. Figure 15.24 shows the whole system, which includes illumination, sensor

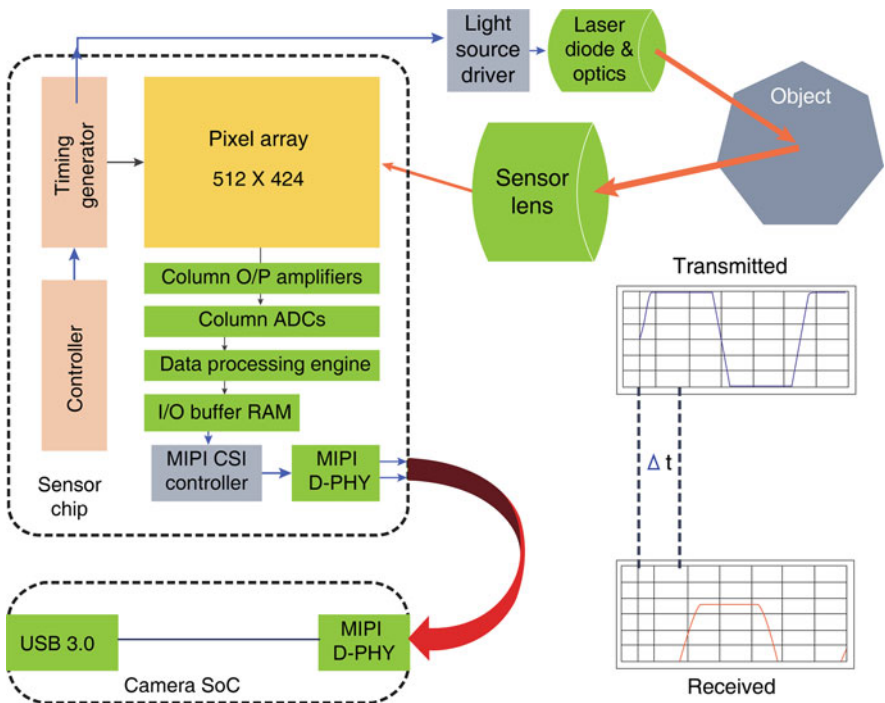


Fig. 15.24 3D image perception system [51]. The system comprises the sensor chip, camera SoC, illumination, and sensor optics

optics, sensor chip, and a camera with System on a Chip (SoC) technology. The SoC function is to communicate with the computer or Xbox for gaming applications. The ToF measurement system sends a modulated square wave signal to the light source driver in order to measure the time it takes to travel from diode optics to the target and returns back to the sensor optics, and this is used to calculate distance. The timing generator creates a modulation square signal, which is used to synchronize light source driver transmitter generation and the pixel array receiver detection. Light hits the object and comes back to the sensor optics in time Δt , the system calculates Δt estimating phase shift between received light signal at each pixel and the known synchronization signal (indirect ToF) [51]. Due to the intensity of the signal emitted to determine the ToF, the sensor can be used outdoors.

A mathematical model of noise can be obtained, which can be used in post-processing tasks in the depth image filtering stages for any application. Different measured characteristic parameters can be analyzed to determine the qualities of Kinect v2 in robot navigation, for example, systematic and non-systematic error, measurement capabilities in short ranges, and the influence of ambient light for indoor, cloudy, and direct sunlight applications. The noise model can be a function of the distance of measurement and the angle of the observed figure; however, the model should include conditions of extreme solar luminosity, overcast, and interiors [50].

In order to integrate Kinect v2 to a robotic system, it is necessary to associate a coordinate frame system with the depth measurements in terms of a global coordinate frame which describes the environment or a local coordinate frame associated with robot pose. Measurements collected are processed and can be rendered as robust maps which have a lot of data which describes the environment and objects around.

15.4.2 Localization Sensors

15.4.2.1 Wheel Encoders

Wheel/motor sensor has a wide history being used to measure internal state and dynamics of motor powered systems. These types of sensors offer an excellent resolution at a very low cost. These sensors are usually located at the shaft of the motor drive or at a steering mechanism attached to the motor.

Optical encoders can be classified due to the way it presents the data in:

- (a) Absolute encoders. Represent a position with a unique digital code, this means that if the encoder has a resolution up to 16 bits, has a span of 2^{16} different codes to represent a physical position.
- (b) Incremental encoder. Generates an output signal every time the mechanism moves, the number of signals per revolution or in a span of distance defines the resolution of the device.

In mobile robotics, incremental encoders are the most common device, used to measure speed and position within a motor drive or at the shaft of a wheel, steering mechanism, or another joint mechanism. The sensors are proprioceptive so their estimation of position has the best performance in the reference frame of the robot and need corrections when applied to mobile robot localization.

An optical encoder produces a number of pulses sine or square for each shaft revolution, and it has the next elements.

- Illumination source
- A fixed grating that masks the light
- A rotary disc with a fine optical grid that rotates with the shaft
- A fixed optical detector

As the motor shaft moves, the amount of light varies based on the alignment of fixed and moving gratings. In robotics, the resulting sine wave is transformed into a square wave.

As the motor shaft moves, the amount of light that reaches the optical detectors varies based on the alignment of the fixed and moving grids. In the robotics, the resulting sine wave is transformed into a square wave using a threshold value to differentiate the values that represent the light and dark states [47].

Resolution is measured in pulses per revolution (PPR) or cycles per revolution (CPR). A typical encoder in mobile robotic applications may have 2000 PPR, while the optical encoder industry can reach up to 10,000 PPR sensors. Also, the sample rate needed for detecting shaft spin's speed can easily be achieved by commercial encoders, so there is no bandwidth limitation for mobile robot applications.

Quadrature encoders are another version of these sensors modified in order to obtain more data from an incremental encoder. This device has two pairs of emitter-detector instead of one, both detectors are 90° shifted with respect to each other in terms of the rotary disc grid. The resulting output signal is a pair of square wave shifted the same 90° as can be seen in Fig. 15.25, both signals receive the name channel A and channel B. A third pair emitter-detector is included in order to detect one unique position called index, which is used as a reference to count revolutions or finding home position. As it can be seen in Fig. 15.25, this index sensor pair has only one slot clear position in the outer track, and thus, it is only one pulse in the whole

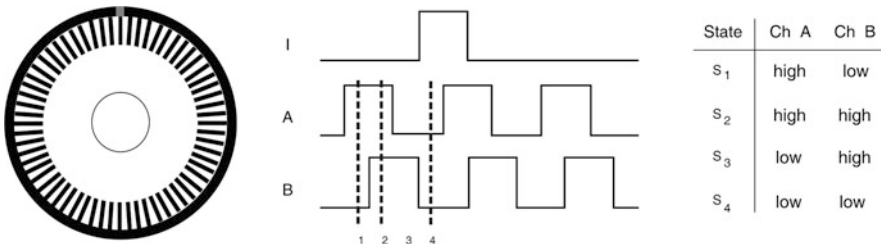


Fig. 15.25 Quadrature optical incremental encoder [47]

turn. Channel A and channel B signals provide more information such as direction of rotation (if we detect which signal rises first). In addition, a different combination of channel states causes different detectability states which increase resolution by four. This means that a sensor with 2000 PPR hence in quadrature yields 8000 PPR.

The systematic errors and the cross sensitivity inherent to encoders can be eliminated using engineering tools such as filters and probabilistic methods. The accuracy of optical encoders is usually considered 100% for mobile robot applications, and any error due to any optical encoder failure is minimized by errors caused by motor shaft or actuator.

15.4.2.2 Global Positioning System (GPS)

Historically humans have used beacons, like stars, mountains, guide posts, and lighthouses for navigation, a modern approach involves the use of emitters located far away from a receiver which is situated at the mobile vehicle. This sensory setup makes available accurate outdoor localization, with some limitations due to the nature of the technology used.

GPS is a modern beacon system instrument used nowadays for outdoor navigation and localization of air and land vehicles.

Twenty-four satellites are available for free GPS service, and they orbit every 12 h. Four satellites are located in each of six inclined planes, 55° with respect to the plane of the equator [47] (see Fig. 15.26).

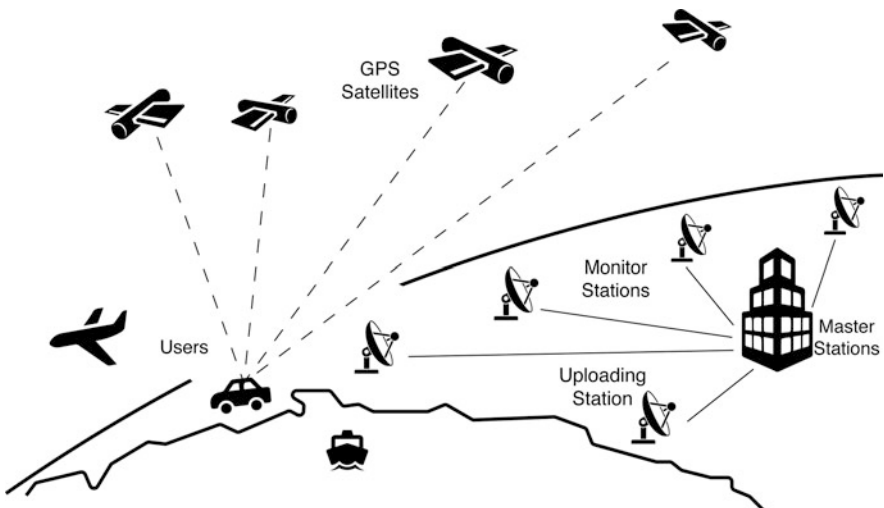


Fig. 15.26 Calculation of position and direction based on GPS usually requires more than three satellites [47]

GPS are passive and exteroceptive sensors. Satellites with GPS service are synchronized, so their transmission signals are sent at the same time. A GPS receiver reads the transmission of two or more satellites, the time difference between these transmissions can be used as an indicator to the receiver about its relative distance to each satellite. By combining the information of the position of each satellite and the delay of arrival times, the receiver can calculate its own position.

This operation requires only three satellites, but as long as there are more transmitting to the receiver, position calculus becomes more accurate. The arrival time delay ranges are of the order of nanoseconds; that is why signals from satellites must be perfectly synchronized. In order to achieve this, satellite timing register is refreshed by ground stations routinely and also an atomic clock is implemented on satellite's circuitry [47].

GPS receiver clock timing is Quartz crystal based, that is why often needs more than three satellites to have an acceptable location reading, that is the main reason for GPS limitations on cramped spaces like those surrounded by tall buildings, big trees, or mountains and places where reading more than three signals at the same time becomes impossible, and position calculus is inaccurate, that is also why indoor applications for GPS fail to provide a reliable position sensing.

GPS performance is also affected by other different factors like the orbital path which is not a straight line, but a curve, and therefore, resolution is not uniform on all earth regions, having a lot of variations and uncertainties away from the terrestrial equator.

GPS satellite information can be obtained by implementing different techniques in order to get better resolution. Some of these techniques are pseudo-range which usually has a resolution of 15 m; another technique is the so-called differential GPS (DGPS), which uses a second receiver located at a reference place to correct position and can have resolution of less than 1 m; and one last technique is used measuring phase of the carrier signal from satellites transmission, this GPS receiver can achieve 1 cm resolution for point position [47].

One last consideration must be taken into account when using GNSS receiver with automated mobile robots, which is update rates. Usually 5Hz GNSS updates rates are obtained, that is why GNSS are used with other types of sensor and navigation algorithms.

15.4.3 Navigation Control

15.4.3.1 Simultaneous Localization and Mapping (SLAM)

On a day-to-day basis, we make use of navigation control when we drive our cars or when we navigate inside of schools or buildings. But unless we have a map, we must always do an exploration previously. Nowadays we have applications to guide us anywhere in the world (such as Google Maps and Waze). Accordingly, autonomous

vehicles need these requirements too; they need to have a predetermined map of their work area or they need to create one while exploring their environments.

The process of simultaneous localization and mapping (SLAM) is sometimes called the concurrent mapping and localization (CML) process. This is the continuous construction of a map and the computation of the state of the robot within its environment [46].

The maps give a human operator an intuitive perception of the environment, but also it is essential to create location maps for the navigation control of autonomous robots, several types of maps can be generated (e.g., feature based or location based) [28]. These can be used for route program and the correction of the calculation error produced by estimating the state of the robot. When the hardware and software are more complex, robots can even be programmed to make their own decisions in real time in order to avoid any collision or path planning. If we do not have a map, the navigation estimation would lose its path and the vehicle would drift. But if we have a map, the robot can correct its location error by returning to the same areas; this process is also called loop closure [46].

SLAM can be used in applications in which a map is not available and requires to be created, that is, in SLAM the trajectory of the vehicle and the location of all landmarks are estimated online without the need for any a priori knowledge of location [52]. Sometimes SLAM may not be required, like in applications in which an accurate prebuilt map describing fixed physical constraints of the environment is provided, that is, an autonomous vehicle operating in an indoor facility with a manually built map and also in applications in which the robot has access to a GPS (the GPS satellites can be considered as moving beacons at known locations) [46].

In most of the projects, different types of sensors were used to cover the widest range of visibility and scope possible, from ultrasonic sensors to RGB-D cameras, LiDAR, and IMU such as accelerometers, encoders, and GPS. All of these are necessary to be able to map and make decisions for the robot in navigation control, in addition to having algorithms to represent the 3D and 6D maps for vehicles.

The autonomous navigation of a robot is a complicated task which leads to several problems that need to be solved. Nowadays, the articles written on projects or advances on this subject are more concurrent specific, and with more diversity on how to apply the hardware to the software in order to achieve a great improvement in performance. Hence, we will see some case projects related to an area as important as is the control of navigation for autonomous robots.

In most of the investigations about autonomous robots' navigation, the SLAM used is made by LiDAR (Fig. 15.27). Its use can be divided into two categories: first, to make an initial map sample of the area where the robots will be used and, second, also [53] can be used on top of a robot to continue building maps while the robot interacts with its environment and, thus, be able to have a better response to certain situations. Other projects generate different types of mappings with different resolutions for certain specific tasks that they want to solve. For example, in [53], to maintain high performance and low memory consumption, they used multi-resolution maps, such as maps based on octrees (Octomap) which are a structure of information for multiple storage resolutions in 3D, how you can see at Fig. 15.28.

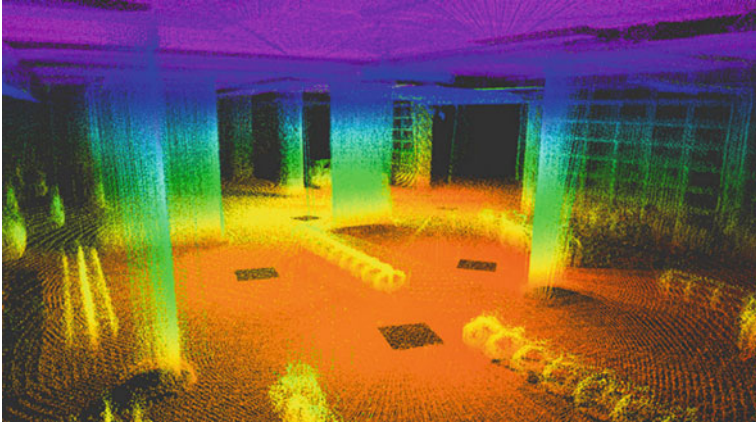


Fig. 15.27 Aligned scans from parking garage [53]

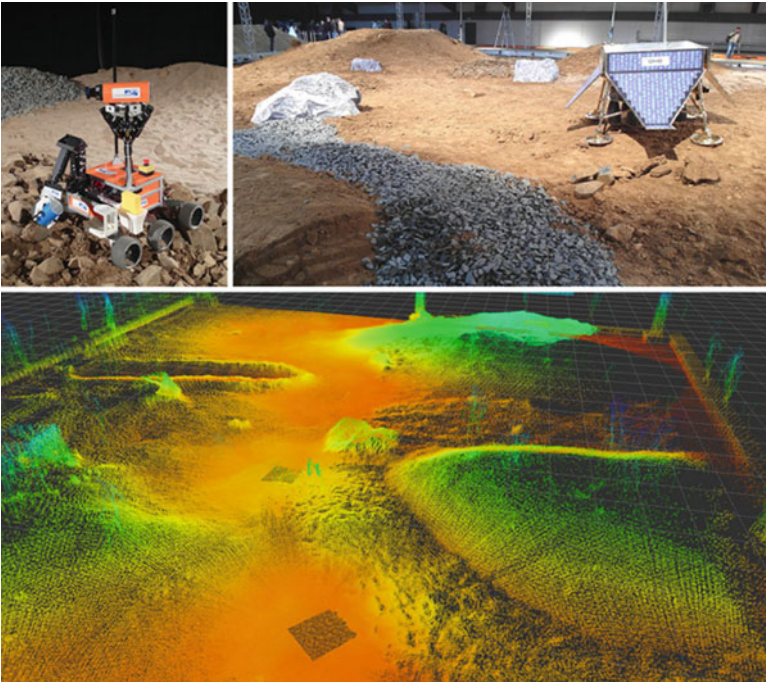


Fig. 15.28 (a) Explorer, Schadler team robot for DLR SpaceBot Cup (upper left image). (b) SpaceBot Cup Arena (upper right image). (c) 3D scans of the arena taken by Schadler (lower image) [53]

In other autonomous robots, we can find [54] cameras and IMUs (barometers, gyroscopes, accelerometer, and Global Navigation Satellite System (GNSS)) in addition to having a LiDAR. The cameras can rotate to have an omnidirectional vision, or stereo configurations as can be seen in Fig. 15.29, depending on their locations in the vehicle. All of this to be able to obtain 6D maps based on maps in SLAMs. In environments where it is not possible to use GNSS, all of the other components can give estimated location coordinates to be able to general mapping in 6D (Fig. 15.30).

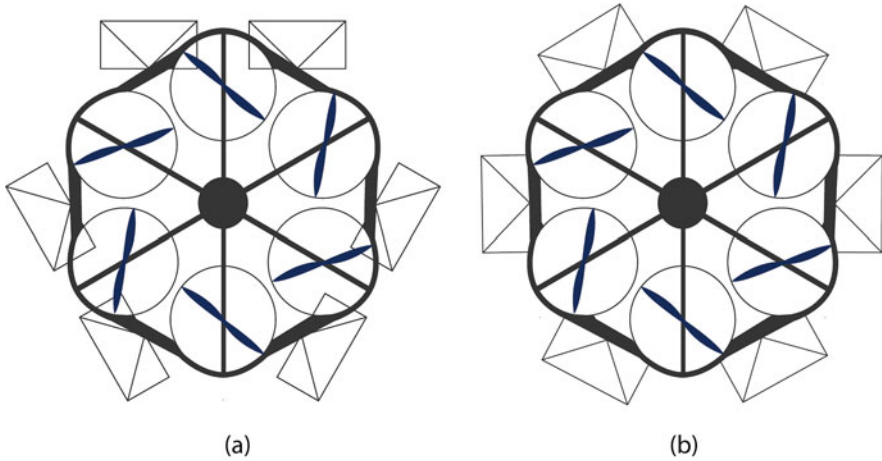


Fig. 15.29 Mounting of the cameras in micro aerial vehicles (MAV). (a) Triple stereo configuration. (b) Omnidirectional configuration [54]

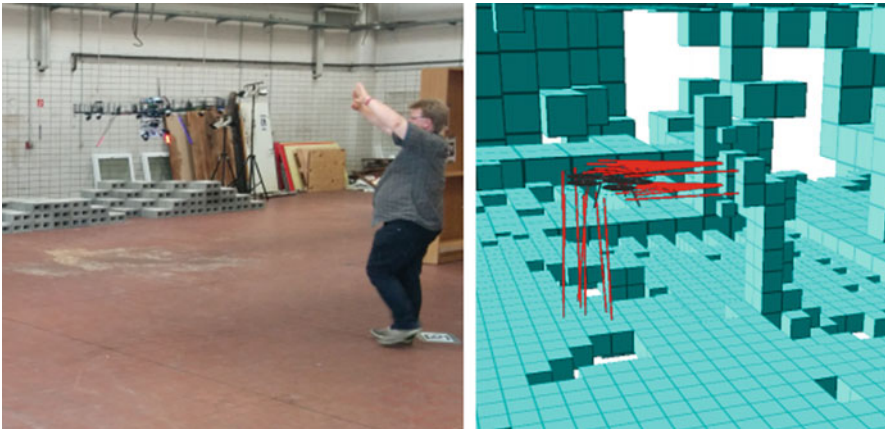


Fig. 15.30 (a) Picture taken when testing the MAV with a person. (b) Scan from the MAV, where the MAV is pushed away from approaching a person [54]

Once having the mapping of the area where the tests and activities are to be carried out, in addition to having SLAM as well, a big problem that presents itself in an autonomous robot will arise since the pre-established map will be compared with the route in real time. Therefore, obstacles or uncertain situations can be presented, and it is here when it is of the utmost importance for the robot to have a navigation control, so that it will be able to dodge the aforementioned obstacles and to select other routes with better times or that are clearer. For decision-making, various algorithms and logics accompanied by hardware in order to be able to enter data for feedback are incorporated. And so, all this can be used in various ways, such as in 2D maps, 3D, and 6D being the main modes of mapping. This means that the equations that are designed to be used in robots are different. The design of their flow diagrams also has an effect consequently; these two are going to determine all of their model decisions. Currently, there is a project [55] in which the images captured by the cameras of the robots pass through filters which can see each step of the robot. With the application of equations, the free areas that lie ahead can be predicted. In addition, a 2D map is generated in which their layers are completely connected. For this reason, their decision-making is based on networks of value interactions that work, in turn, to a neural network (Fig. 15.31).

In another case [55], they generate SLAM to see their environment in real time, thereby knowing if there are variations, such as temporary or permanent obstacles, changes in their work environment, with their predetermined mapping or with previous interactions with their journey. In addition, they make cells (grid cell) in 2D and group them to evaluate the drivable areas, all of this to avoid complicated areas where it is not optimal for the robot to move (Fig. 15.32). In this last case [54],

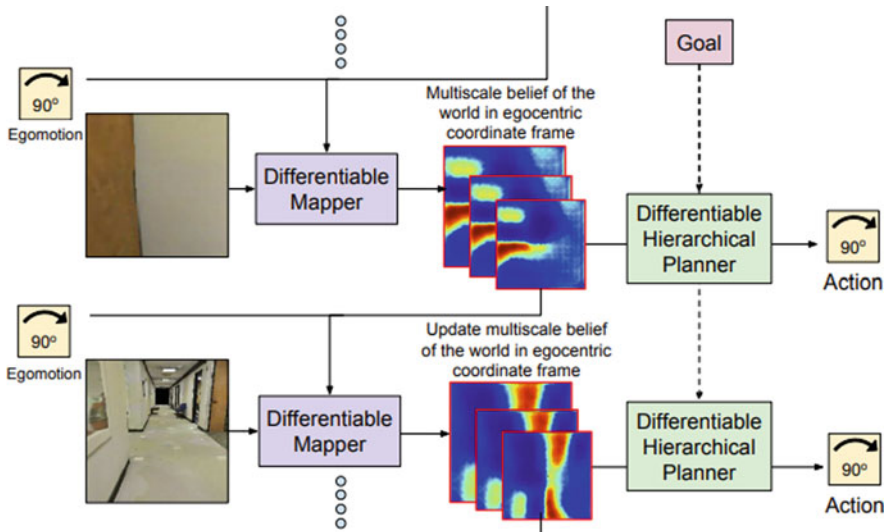


Fig. 15.31 Overall network architecture from S. Gupta [55]

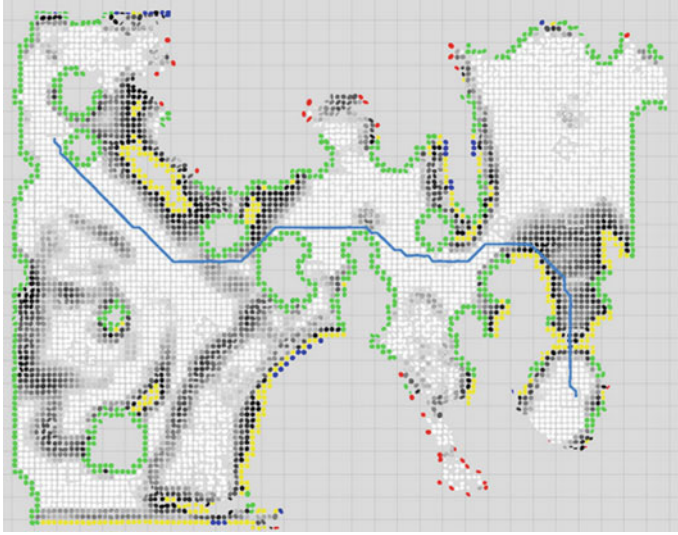


Fig. 15.32 Navigation planning [53]

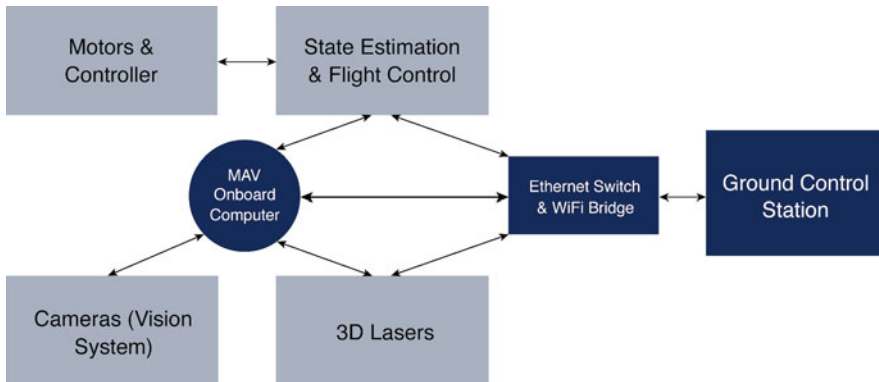


Fig. 15.33 Overall schematic for MAV [54]

an autonomous MAV uses cameras to classify objects but also uses a SLAM (3D lasers) to create 3D maps. Also, with both technologies, they can create environments in 6D, which allows them to identify the obstacles to redo a new route to reach the destination. Apart from this, GNSS is used for outdoor environments, but for interior use, a visual odometry (VO) is used as well as components that integrate an IMU in order to be able to know their location in the 3D and 6D planes (state estimation and flight control). All of this information passes through usb hub 3.0 and 2.0 for it to be processed in a computer center inside the vehicle and to send information to another computer via WiFi for monitoring (Fig. 15.33), which shows an overall schematic.

15.5 Path Planning

When we address any topic related to autonomous transportation, often many doubts arise, that is because there are several factors that must be taken into consideration in order to successfully reach our desired destination; being the motion or path planning, one of those factors, and that is what this section is oriented to.

As humans, most of the time, we do not even need to worry about getting from point A to point B, as a result of our capabilities in identifying the answers to the questions: where am I? Where do I want to go? How do I get there? And what are the obstacles or dangers in the path to my objective?

In order for a robot to answer the questions shown previously, a certain kind of intelligence might be integrated into it; in consequence, many kinds of research have been done in subjects such as mapping, localization, and path planning. The objective of this last subject is the minimization of the distance the vehicle needs to travel, and to accomplish that, an efficient path planning algorithm is necessary.

Some years ago the computational power available could not handle the requirements for online path planning algorithms, and researches opted for rule-based algorithms that gave good results in indoor controlled environments [56].

15.5.1 Algorithms

15.5.1.1 A* Algorithm

The A* algorithm belongs to the graph search group, and it shares several similarities with the Dijkstra's algorithm. A* is one of the most efficient algorithms in solving the problem of path planning; however, in scenarios with a high density of obstacles, it might not be the ideal algorithm to implement due to its high time consumption when it is dealing with a hard to process situation.

$$f(n) = g(n) + h(n) \quad (15.6)$$

A* is based on the evaluation function presented in Fig. 15.34, where $g(n)$ is the distance from the current node to the node n . On the other hand, $h(n)$ is the distance from our goal position to the node n . The distance can often be called "cost" or "heuristic."

The way A* works is simple yet brilliant. The algorithm starts by identifying the nodes that already have a direct path to the starting position, following the example in Fig. 15.34. Those nodes would be F , E , B , and D , then the algorithm evaluates the already mentioned nodes with Eq. (15.6) (see Table 15.1).

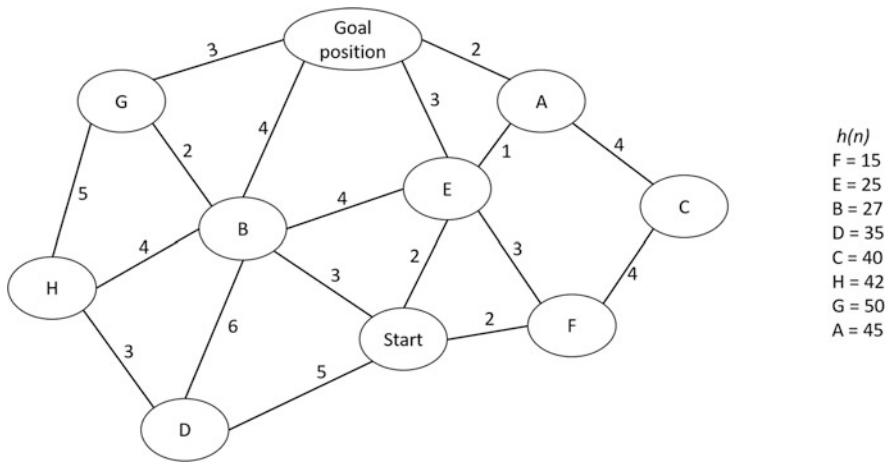


Fig. 15.34 A* algorithm example

Table 15.1 A* algorithm node evaluation example

$f(F) = 2 + 15 = 17$
$f(E) = 2 + 25 = 27$
$f(B) = 3 + 27 = 30$
$f(D) = 5 + 35 = 40$

15.5.1.2 Field D* Algorithm

In mobile robot navigation, a representation of the environment is needed, generally divided in cells. Depending on the environment, the representation could be binary, obstacle, or free, or it could have an associated cost for each cell, $g(s)$. Most algorithms like Dijkstra, A*, or D* are limited by the small set of possible transitions; they plan from center of cell to center of cell resulting in paths that are not optimal and difficult to follow in practice [57].

Traditional algorithms compute the path cost assuming that the only possible transition from one cell to another is a straight line from the cell to one of its neighbors. If this restriction is relaxed and the path can go from the cell to any point in the border of the neighbor cell, a minimal cost trajectory can be calculated. Unfortunately, there are an infinite number of points, and calculation of all the possible trajectories is not possible.

It is possible to provide an approximation for each boundary point by using interpolation. In order to do this, it is necessary to assign nodes to the corner of each cell, with this the cost of traversing two equal length segments of an edge will be the same, this is shown in Fig. 15.35. This solves the cost problem when a segment crosses two cells, each with different cost. The nodes in the graph are used as sample points of a continuous cost field. The optimal path from a node must pass through an edge connecting two consecutive neighbors.

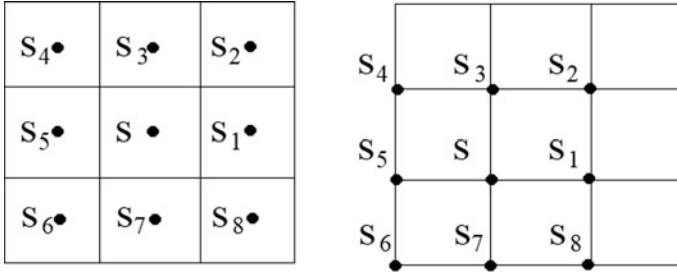


Fig. 15.35 Solution of the cost problem

To calculate the cost efficiently, assume the path cost of any point s_y residing on the edge between nodes s_1 and s_2 , which is a linear combination of their cost.

$$g(s_y) = yg(s_1) + (1 - y)g(s_2) \tag{15.7}$$

Equation (15.7) is not perfect, and the path cost may not be a linear combination of $g(s_1)$ and $g(s_2)$, but it is sufficiently exact to be practical.

The path cost, given s_1 and s_2 and cell costs b and c , can be calculated by Eq. (15.8):

$$\min \left[bx + c \sqrt{(1 - x)^2 + y^2} + yg(s_2) + (1 - y)g(s_1) \right] \tag{15.8}$$

where x is the distance traveled along the bottom of cell s before turning the center cell to reach the right side at a distance y .

Because a linear interpolation is used, it is less expensive to cut through the cell than following the boundaries. If there is a component of the cheapest solution that goes through the center of the cell, then it will be as large as possible forcing $x = 0$ and $y = 1$. If there is no component through the center, then $y = 0$.

Assuming that the optimal path is as shown in Fig. 15.36a, it travels along x axis to a certain point and then crosses diagonally to a point y and finally to the upper right node. Clearly this path is shorter than going horizontal all the way to lower right node and then up to upper right node. Scaling the resulting triangle so that the upper vertex is now located in the upper right node, maintaining the slope will be shorter than the previous path as seen in Fig. 15.36b.

Thus, the path will travel a distance over the x axes and then go straight to the upper right node, or it can go directly from the origin node diagonally to some point on the right edge and then up to the upper right node depending on the costs b and c .

Once the cost of the path from the initial position to the goal is calculated, the path is determined by interactively computing the cell boundary point to the next. Because of the interpolation technique, it is possible to calculate the cost of any point inside a cell, not just the corners.

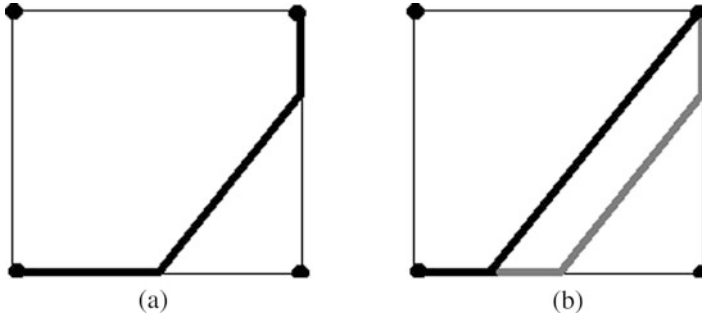


Fig. 15.36 (a) Optimal shortest path solution. (b) Simplified path with same slope solution

15.6 Case Study: Intelligent Transportation Scheme for Autonomous Vehicles in Smart Campus

Nowadays, diverse autonomous vehicle designs are used to assist in various daily tasks, for example, in smart farming, smart industry, among others. The self-driving vehicles positively impact on economy and environment, as well as the completion of tasks efficiently. Similarly, the use of autonomous vehicles on Smart Campus could increase student and worker satisfaction at universities as it is the case of the intelligent transportations scheme for autonomous vehicles in CETYS University [58].

To introduce this case study, it is essential to understand the context. CETYS University Mexicali Campus is located at the capital city of the Mexican State of Baja California, which is a US–Mexico border city. Mexicali city has a warm and arid climate; in consequence, walking in summer could be very dangerous. According to the National Institute of Ecology, the fatality rates caused by heat waves in Mexico were presented in Mexicali [58]. Likewise, the Mexicali’s General Hospital reported 72 patients with heat waves symptoms, and 23 of them died during the period from 2006 to 2010 [59].

Consequently, autonomous vehicles in the campus will possibly be a solution to reduce the risk that heat waves affect the students, professors, and administrative personnel during their movement inside the university. Additionally, above all population inside the campus, a significant number are elderly people, and studies have shown that elderly population is more vulnerable to heat waves and elevated temperatures [60, 64], given the previously mentioned statistics.

Moreover, seeking the well-being of people with mobility or visual impairment in the campus, self-driving vehicles benefit their transportation inside the university. As the parking and buildings inside the campus are separated by long distances, the possibility to have a qualified transportation system improves the life quality and inclusivity inside the campus.

The previous context description stands for the motivation to develop an intelligent transportation system implemented in an automobile structure capable to transport passengers between buildings inside the campus and from the parking

to a building. The design of the vehicle includes the artfulness to avoid obstacles (i.e., people walking through the path). Hence, the design of the vehicles included the development of a 3D vision system [61] capable of detecting the road, potential static, and dynamic obstacles (or dangerous conditions), among other variables. Then, design an autonomous navigation system capable of safely traversing a path from point A to B using the 3D vision system data and re-route the vehicle if needed to achieve the desired location. Also, build a sustainable electrical system for the vehicle, with a battery capable of recharging itself through solar energy. Finally, design a network infrastructure to suffice the whole communication scheme and fulfill the requirements of a Smart Campus.

The term Smart Campus refers to a university campus capable to keep all the data infrastructure and offer different services to the students and to all people inside the campus. Hence, to develop a smart agent inside a smart network, some technologies should be adopted as a standard, for example, the sample rate for sensors, the data communication scheme, how systems interact with each other, how all the microcontrollers behave as master and slave depending on the network itself, etc. One of the agents interacting in this Smart Campus would be the autonomous vehicle. To achieve an effective interaction inside the system, the vehicle must be capable to respond to user's requests, to navigate avoiding obstacles and leave the passengers at their selected endpoint. The steps taken for the design of the autonomous vehicle are described in overall terms.

15.6.1 Applied Simultaneous Localization and Mapping (SLAM)

In order to map the surroundings accurately and to properly move along the different route traced by the system, an efficient mapping algorithm known as SLAM (simultaneous localization and mapping) must be implemented (see Fig. 15.37). This algorithm is implemented to provide autonomy to the vehicle.

As Fig. 15.37 shows, the design includes some sensor inputs such as a LIDAR and Kinect Frame, both described in previous sections. Odometry readings change according to the vehicle movement. The uncertainty from these lectures is treated by the odometry update algorithm. The re-observation process is essential to map construction. All the elements improve the accuracy of the vehicle movement.

Figure 15.38 shows the control flow process for the autonomous vehicle. The vehicle starts the movement after getting current location and the calculus of the initial trajectory. The sensors are implemented to detect obstacles. If an obstacle is detected, the vehicle reduces its speed or stops. The map construction process intended for scanning the environment and analyzing data is performed during all the trajectory in order to control correct navigation through map corrections based on the obstacles' position. A GPS is available to give information about vehicle's location and to correct initial trajectory or re-route if necessary. The sensors described

Fig. 15.37 Diagram of a SLAM algorithm [58]

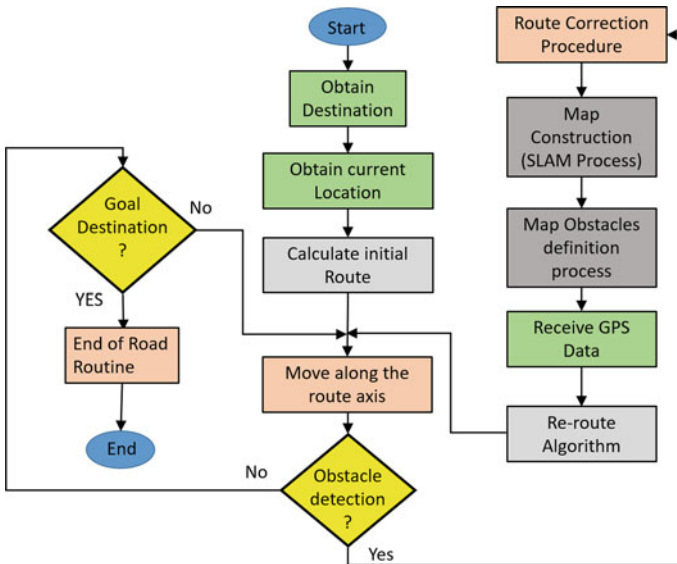
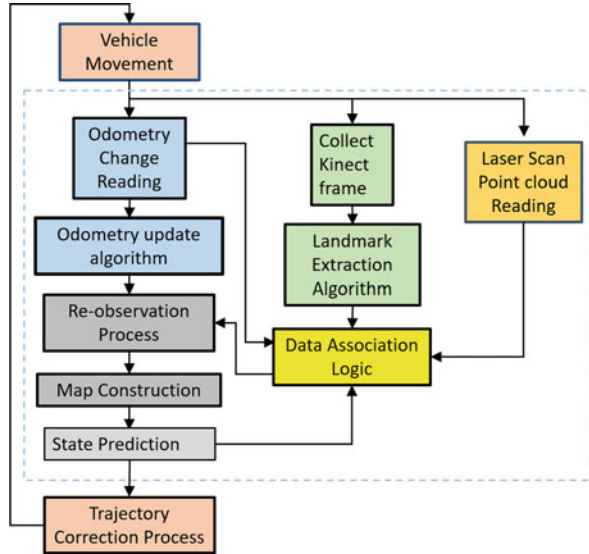


Fig. 15.38 Control flow process for the autonomous vehicle [58]

below should behave synchronously in order to maintain consistency in the whole scheme and for the system to choose the best route to take, and how to re-route correctly taking into account all the obstacles in the vehicle’s line of vision.

15.6.2 Mechanical Design and Kinematic Model

Our mechanical design takes into account the vehicle's chassis, gearboxes, motors, wheels, sensors, batteries, computer, Ethernet switch, cable management, and airflow. The whole autonomous system (vision and control) is mounted on an aluminum chassis with 4// Omni directional wheels in order to satisfy the requirements of forces applied to the base of the vehicle, so all forces can be dissipated and damped. This type of drive allows linear and rotary motion through space which provides the specific advantage of maneuvering in all directions without steering the wheels, which allow for more accurate control and precise movements (see Fig. 15.39).

The proposed architecture scheme is shown in Fig. 15.40. The design achieves the automation goals. The communication between all the elements is established, closing the loop for a complete automation and control scheme.

This case study is an example of how technology contributes to improving the well-being of students, professors, and administrative personnel in universities. Autonomous vehicle must respond needs in Mexicali where the weather conditions affect the health of several people.



Fig. 15.39 Autonomous vehicle scale prototype [58]

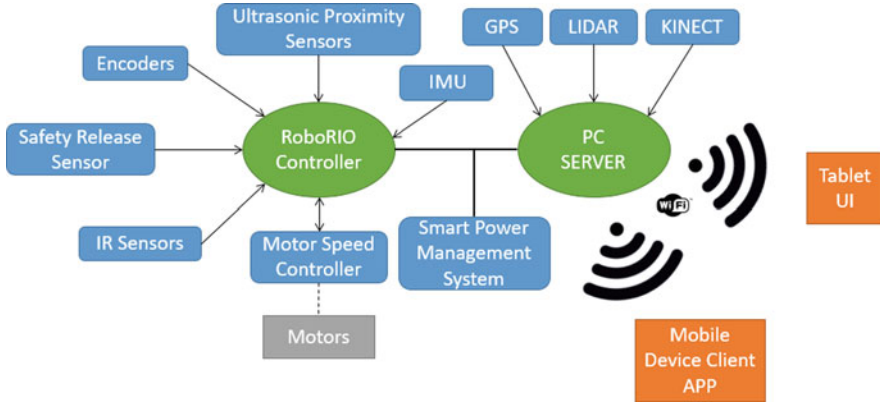


Fig. 15.40 General interconnection scheme for the autonomous vehicle system [58]

15.7 How Innovation in Business Models Will Change the Future of Cars

Electric cars are the future. We all perceive that. It is a topic that resonates in our daily life. But electric cars do not have the same features as gasoline cars, which are more powerful, bigger, heavier and more expensive. Generations in the last one hundred years have seen the evolution of cars, from de T model of Ford to the latest Tesla model. But from the silence generation passing through the baby boomers, generation X, millennials, and finally generation Z, things have changed mainly in the business models, those businesses that are ruled by the consumer profiles of the biggest market segments. Nowadays innovation is changing the rules of the business models. People have the chance to choose between different services online, Amazon, Google, Airbnb, and HostaPet. New business is based on IT platforms where innovation allows the idea to do services online between people around the world. Our generations are in the era of open innovation [3, 62]. This is the time when Amazon has no factories, Airbnb has no hotels, and Uber has no cars. But people can get better services because Internet and smartphones make easier trading.

In the era of open innovation [3, 62], the future of the automobile lays over the profile trends of generation Z. While businesses like Kodak took one hundred years to experience a change in their business model, and lost the battle [62] of the digital camera in the early 2000s, other enterprises like Procter & Gamble have understood market trends to change their business models in order to sustain competitiveness. In that sense, Henry Chesbrough refers to the use of the automobile as an opportunity.

15.7.1 *The Misuse of Expensive Vehicles*

Think about the number of hours your vehicle takes to take you anywhere at any time you want during a regular day. Let us say, it is about 2.5 h of use each day. That means you are using your car a little more than 10% of time, but you paid for 100% of the time you really need that good. Assume an average cost of a car as around \$15,000. This means that you paid ten times more of the use of your vehicle, because you only need it 10% of the time. From a business point of view, there is a chance to make business to take advantage of that extra cost between 10% of usage and \$15,000.00.

15.7.2 *Generation Z Consumer Profile and the Future of Vehicle*

The business model canvas of Alexander Osterwalder and Yves Pigneur [4] spins around the value proposition for a market segment. Watching consumer profile for generation Z, it can be seen that generation Z or centennials are also considered a better version of millennials. They are digital natives. They do not know a world without Internet. Generation Z will be around 30% of the world population with a very important percentage of purchasing power for 2020 (see Table 15.2). Generations are really fuzzy to define, but this generation locates between 1995 and 2010. For generation Z, value is the number one need as a consumer. This means, as they have many options to choose from, they are not people of one brand. So, for them value is the most important thing about a product or a service. Centennials trust youtubers more than a celebrity, because youtubers are people like them without makeup. It means that they trust authenticity more than appearance. They use to watch at reviews on social media and in general generation Z communicate with their friends, families, and also with brands.

Analyzing the three core concepts of the consumer profile defined by Alexander Osterwalder, it can be expected from the consumer profile of centennials generation that they expect products or services to fulfill its purpose, they appreciate value. So

Table 15.2 Population resident in the USA in 2017 [63]

Generation	Population (millions)	Percentage	Time range
Greatest	2.57	1	Before 1928
Silent	25.68	8	1928–1945
Baby boomers	73.47	23	1946–1964
X	65.71	20	1965–1980
Millennials	71.86	22	1981–1996
Centennials	86.43	27	1997–2016
	325.72	100	

as they are not loyal to a brand and appreciate authenticity, their consumer profile is practical. So when they spend money, they commit with those products and services that are good about achieving functional jobs over the ones that achieve emotional or social needs. That means, although also as millennials, property is not their first concern, function is the most important thing for centennials as well as for millennials when they buy a car. So in this stage, two of the relevant concepts from the definition of value proposition canvas from Alexander Ostewalder for the consumer profile pointed as pain relievers and gain creators for the generation Z are each day more relevant. From all these generation behavior and trends from a business model point of view, it can be expected that eventually as well as the use of electric cars, new and innovative business models around the consumer profile drivers will be met in the next years.

15.7.3 Business Model Canvas for Car to Go

In 2008 Daimler Chrysler launched their program Car2Go (www.car2go.com). Their value proposition (see Table 15.3) for the market segment is as follows: (a) Use by the minute. With car2go, driver is in control of every mile and every minute. Unlike other car sharing services, drivers pay only for how long they actually using the car and (b) one-way trips. Start at point A and end at point B without returning the car to point A. Pick up from the street and get going, (c) on demand. Drivers can get going without prior reservations. Drivers can use their smartphones to find and drive on the spot or reserve for up to 30 min (for free) before beginning.

Table 15.3 Value proposition canvas [4]

Value proposition	Customer segment
Gain creators Easy localization of an electric vehicle Quick instructions to use vehicle Vehicles registered on Google maps	Gains Quick finding of vehicle Want to pay only by minute Easy finding of the vehicle Smartphone localization
Pain relievers Use of Google map to find easy and quick all the vehicles of the company Battery charged Electric charge stations	Pains Cannot find a car close to the area Easy use of the vehicle Low-battery electric car Sanitized vehicle
Products and services Electric car easy to use rated by minute and registered on Google maps	Customer Jobs Use of a car only for the time the customer needs

15.7.4 Autonomous Car as an Innovative Business Model

Market trends in recent years formed by the consumer profiles of millennials and centennials open the possibilities for the use of autonomous cars. Evolution of business models based on open services innovation in the era of 5G technology may in the near future set the bases for the use of autonomous electric cars. Clearly, technology has reached a new level with the advent of 5G technology. And innovation on business models, mainly in open services are becoming a very important part of the economies. On this stage, it could happen that new generations will use to buy services online through their 5G technology smartphones, including autonomous electric cars.

Analyzing the value proposition of the business model Car2Go, it could be a natural evolution to scale for autonomous cars. In a first approach, the business model of an autonomous electric car meets all tasks for the profile of the customer. And more, business model of the autonomous car exceeds the expectations defined as gain creators for a value proposition of the Car2Go model. An autonomous car eliminates the need for an easy localization of the vehicle. The autonomous car can go to meet the customer. Being autonomous also discards the need for understanding how to use the vehicle. And of course, autonomous vehicle needs only a button to know where to go. Of course, 5G technology in a smartphone makes easier for customer to command tasks to vehicle. It can be found that autonomous cars can also program electric charge timing.

Finally, it can be concluded that latest technologies and evolution of business models on open services meet profiles of customer segments for new generations.

15.8 Conclusions

In this work, we have reviewed the fundamentals of autonomous vehicles, from automation concepts going through the standard for on-road vehicles defined by the Society of Automotive Engineers, to novel sensors and image processing techniques, including some other commercial products, that is, mobile robots for industry and other small devices as a mobile vacuum. Modernity offers plenty of technology, and beyond an autonomous electric car itself, the disruption line in the advent of autonomous electric cars to be used on a regular basis depends more from the future drivers or even users of autonomous electric cars as a service.

New generations have left the typical profile of a consumer. So, the next step in the business of individual transport is more about the needs of the future driver or user. It seems that at least for high-density population cities, there are still new innovative ways to get from one location to another. Post-centennial generations will use faster technologies and will define market trends. The use of autonomous electric cars in the near future will be a reality when disruptive technologies meet disruptive models for individual transport committed to relief the needs of the customer. Autonomous

electric cars will be in the center of business trends ruled by IT platforms, which will catalyze autonomous electric cars as regular transport. In the near future, innovation on services will take the use of autonomous electric cars to the next level.

The future driver or user profile of centennials will appreciate the value proposition of the autonomous electric car. There will not be need of finding parking, neither need to pay for it. An autonomous car will meet the user in the place and time that the user needs. It is estimated that as well as millennials, for centennials and next generations property is not in the first row of their needs, but the work to be done by the service is the most important thing when younger generations buy or rent a service. So, it seems that for future generations of drivers, those autonomous electric cars paid as a service, by minute, will be a reality.

Acknowledgments The authors would like to thank Center of Innovation and Design (CEID) of CETYS University Mexicali Campus for all facilities to perform the research and for providing the necessary resources to develop this project. Also, special thanks to the image illustrators Luis Esquivel, Alexa Macías, and Valeria Muñoz.

References

1. Bauman, Z. (2000). *Liquid modernity*. Cambridge, UK: Polity Press.
2. Chesbrough, H. W. (2003). *Open innovation: The new imperative for creating and profiting from technology*. Boston: Harvard Business School Press.
3. Chesbrough, H. W. (2003). The era of open innovation. *MIT Sloan Management Review*, 44(3), 35–41.
4. Alexander, O., & Pigneur, Y. (2010). *Business model generation*. Hoboken, NJ: Wiley.
5. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles J3016_201806. (2018). Retrieved February 5, 2019, from https://www.sae.org/standards/content/j3016_201806/
6. Raspberry Pi. (2019). Retrieved February 5, 2019, from <https://www.raspberrypi.org/>
7. Zini 1880. (2019). Retrieved February 5, 2019, from <https://zareason.com/zini-1880.html>
8. Al-Muteb, K., Faisal, M., Emaduddin, M., et al. (2016). An autonomous stereovision-based navigation system (ASNS) for mobile robots. *Intelligent Service Robotics*, 9, 187. <https://doi.org/10.1007/s11370-016-0194-5>.
9. What Is GPS? (2019). Retrieved February 5, 2019, from <https://www.gps.gov/systems/gps/>
10. BeiDou Navigation Satellite System. (2019). Retrieved February 6, 2019, from <http://en.beidou.gov.cn/>
11. Indian Regional Navigation Satellite System (IRNSS). (2019). Retrieved February 6, 2019, from <https://www.isro.gov.in/irnss-programme>
12. European Global Satellite-Based Navigation System. (2019). Retrieved February 6, 2019, from <https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system>
13. Graham, A. (2010). *Communications, radar and electronic warfare*. Hoboken: Wiley. Available from: ProQuest Ebook Central. [7 February 2019].
14. LIDAR vs RADAR Comparison. Which System is Better for Automotive? (2018). Retrieved February 7, 2019, from <https://www.archer-soft.com/en/blog/lidar-vs-radar-comparison-which-system-better-automotive>
15. Winner, H. (2016). Automotive RADAR. In H. Winner, S. Hakuli, F. Lotz, & C. Singer (Eds.), *Handbook of driver assistance systems*. Cham: Springer.

16. Mobile Industrial Robots. (2019). Retrieved February 11, 2019, from <http://www.jacobsonconstruction.com/projects/dabc-asrs-expansion-warehouse-remodel/>
17. Ekren, B. Y., & Heragu, S. S. (2012). A new technology for unit-load automated storage system: Autonomous vehicle storage and retrieval system. In R. Manzini (Ed.), *Warehousing in the global supply chain*. London: Springer. https://doi.org/10.1007/978-1-4471-2274-6_12.
18. Kuo, P.-H., et al. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, 31(10), 2332–2346. <https://doi.org/10.1016/j.apm.2006.09.011>.
19. Waymo unveils self-driving taxi service in Arizona for paying customers. (2018). Retrieved February 11, 2019, from <https://www.reuters.com/article/us-waymo-selfdriving-focus/waymo-unveils-self-driving-taxi-service-in-arizona-for-paying-customers-idUSKBN1O41M2>
20. iRobot. (2019). Retrieved February 11, 2019, from <https://store.irobot.com/default/home>
21. Özgüner, U., Acarman, T., & Redmill, K. (2011). *Autonomous ground vehicles* (pp. 69–106). Boston: Artech House.
22. Weitkamp, C. (2005). *Lidar* (pp. 3–4). New York, NY: Springer.
23. Caltagirone, L., Scheidegger, S., Svensson, L., & Wahde, M. (2017). Fast LIDAR-based road detection using fully convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*.
24. Velodyne VLP-16. (2019). Retrieved February 27, 2019, from <https://velodynelidar.com/vlp-16.html/>
25. Velodyne HDL-64E. (2019). Retrieved February 27, 2019, from <https://velodynelidar.com/hdl-64e.html/>
26. Renishaw plc. Optical Encoders and LiDAR Scanning. (2019). Retrieved February 27, 2019, from <https://www.renishaw.it/it/optical-encoders-and-lidar-scanning%2D%2D39244/>
27. YeeFen Lim, H. (2018). *Autonomous vehicles and the law: Technology, algorithms and ethics* (p. 28). Edward Elgar Publishing.
28. InnovizOne. (2019). Retrieved February 27, 2019, from <https://innoviz.tech/innovizone/>
29. Kinect Sensor. (2019). Retrieved February 27, 2019, from Amir, S., Waqar, A., Siddiqui, M. A., et al. (2017). Kinect controlled UGV. *Wireless Personal Communications* 95, 631. <https://doi.org/10.1007/s11277-016-3915-3>.
30. Giori, C., & Fascinari, M. (2013). *Kinect in motion* (pp. 9–10). Birmingham, UK: Packt Pub..
31. Bernini, N., Bertozzi, M., Castangia, L., Patander, M., & Sabbatelli, M. (2014). Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.
32. Thrun, S., Burgard, W., & Fox, D. (2006). *Probabilistic robotics* (p. 221). Cambridge, MA: MIT Press.
33. Siciliano, B., & Khatib, O. (2008). *Springer handbook of robotics* (p. 857). Berlin: Springer.
34. Li, Z., Zhu, Q., & Gold, C. (2004). *Digital terrain modeling* (p. 7). Boca Raton: CRC Press.
35. Mach, R., & Petschek, P. (2007). *Visualization of digital terrain and landscape data* (p. 38). Berlin: Springer.
36. Hernandez-Aceituno, J., Arnay, R., Toledo, J., & Acosta, L. (2016). Using Kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sensors Journal*, 16(10), 3603–3610.
37. Wedel, A., & Cremers, D. (2011). *Stereo scene flow for 3D motion analysis* (p. 89). Springer.
38. Plemenos, D., & Miaoulis, G. (2013). *Intelligent computer graphics 2012* (pp. 243–263). Berlin: Springer.
39. Schaub, A. (2017). *Robust perception from optical sensors for reactive behaviors in autonomous robotic vehicles* (p. 161). Springer.
40. Jensen, M., Philipsen, M., Mogellose, A., Moeslund, T., & Trivedi, M. (2016). Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17(7), 1800–1815.

41. Mogelmoose, A., Trivedi, M., & Moeslund, T. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4), 1484–1497.
42. Trepagnier, P., Nagel, J., & McVay Kinney, P. Navigation and control system for autonomous vehicles. US Patent 8,050,863 B2.
43. Cox, I., & Wilfong, G. (1990). *Autonomous robot vehicles*. New York, NY: Springer.
44. Jiang, X., Hornegger, J., & Koch, R. (2014). *Pattern recognition* (p. 4). Cham: Springer.
45. Dhiman, N. K., Deodhare, D., & Khemani, D. (2015). Where am I? Creating Spatial awareness in unmanned ground robots using SLAM: A survey. *Sadhana Academy Proceedings in Engineering Sciences*, 40(5), 1385–1433. <https://doi.org/10.1007/s12046-015-0402-6>.
46. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332. Retrieved from <https://ieeexplore.ieee.org/document/7747236>.
47. Siegwart, R., & Nourbakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. Cambridge, MA: MIT Press.
48. Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement*, 46(7), 2127–2145. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0263224113000730>.
49. Gonzalez-Jorge, H., Rodríguez-González, P., Martínez-Sánchez, J., González-Aguilera, D., Arias, P., Gesto, M., & Díaz-Vilariño, L. (2015). Metrological comparison between Kinect I and Kinect II sensors. *Measurement*, 70, 21–26.
50. Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., & Siegwart, R. (2015, July). Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 International Conference on Advanced Robotics (ICAR), Istanbul*, pp. 388–394. Retrieved from <https://ieeexplore.ieee.org/document/7251485>
51. Sell, J., & O'Connor, P. (2014). The Xbox one system on a chip and Kinect sensor. *IEEE Micro*, 34(2), 44–53. Retrieved from <https://ieeexplore.ieee.org/document/6756701>.
52. Durrant-Whyte, H., & Bailey, T. (2016). Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *IEEE Robotics and Automation Magazine*, 13(2), 99–110. Retrieved from <https://ieeexplore.ieee.org/document/1638022>.
53. Schadler, M., Stückler, J., & Behnke, S. (2014). Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner. *Künstliche Intelligenz*, 28(2), 93–99. <https://doi.org/10.1007/s13218-014-0301-8>.
54. Beul, M., Krombach, N., Zhong, Y., Droeschel, D., Nieuwenhuisen, M., & Behnke, S. (2015, July). A high-performance MAV for autonomous navigation in complex 3D environments. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO*. <https://ieeexplore.ieee.org/document/7152417>
55. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., & Malik, J. (2017, November). Cognitive mapping and planning for visual navigation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*. Retrieved from <https://ieeexplore.ieee.org/document/8100252>
56. Lacaze, A., Moscovitz, Y., DeClaris, N., & Murphy, K. Path planning for autonomous vehicles driving over rough terrain. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell.*
57. Ferguson, D., & Stentz, A. *The Field D* algorithm for improved path planning and replanning in uniform and non-uniform cost environments*. Technical Report CMU-TR-RI-05-19, Carnegie Mellon University.
58. Básaca-Preciado, L. C., Orozco-García, N. A., & Terrazas-Gaynor, J. M., et al. (2018). *Intelligent transportation scheme for autonomous vehicle in smart campus*. IEEE, pp. 3193–3199.
59. Martínez-Austria, P. F., Bandala, E. R., & Patiño-Gómez, C. (2016). Temperature and heat wave trends in northwest Mexico. *Physics and Chemistry of the Earth, Parts A/B/C*, 91, 20–26.

60. Åström, D. O., Bertil, F., & Joacim, R. (2011). Heat wave impact on morbidity and mortality in the elderly population: A review of recent studies. *Maturitas*, *69*, 99–105.
61. Básaca-Preciado, L. C., et al. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Laser in Engineering*, *54*, 159–169. <https://doi.org/10.1016/j.optlaseng.2013.08.005>.
62. Lucas, H. C., Jr., et al. (2009). Disruptive technology: How Kodak missed the digital photography revolution. *Journal of Strategic Information Systems*, *18*, 46–55.
63. Resident population in the United States in 2017, Statista. (2018). *The Statistics Portal*. Retrieved from January 25, 2019, from <https://www.statista.com/statistics/797321/us-population-by-generation/>
64. Díaz Caravantes, R. E., Castro Luque, A. L., & Aranda Gallegos, P. (2014). Mortality by excessive natural heat in Northwest Mexico: Social conditions associated with this cause of death. *Front Norte*, *26*, 155–177.

Part IV
Aerial Imagery Processing

Chapter 16

Methods for Ensuring the Accuracy of Radiometric and Optoelectronic Navigation Systems of Flying Robots in a Developed Infrastructure



Oleksandr Sotnikov, Vladimir G. Kartashov, Oleksandr Tymochko, Oleg Sergiyenko, Vera Tyrsa, Paolo Mercorelli, and Wendy Flores-Fuentes

Abbreviations

ACS	Automated control systems
ADD	Antenna directivity diagram
CAF	Correlation analysis field
CCC	Coefficient of cross correlation
CENS	Channel extreme correlation navigation systems
CENS-I	CENS in which information is currently removed at a point
CENS-II	CENS in which information is currently removed from a line
CENS-III	CENS in which information is currently removed from an area (frame)

O. Sotnikov
Scientific Center of Air Forces, Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

V. G. Kartashov (✉)
Kharkiv National University of Radioelectronics, Kharkiv, Ukraine
e-mail: volodymyr.kartashov@nure.ua

O. Tymochko
Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

O. Sergiyenko · V. Tyrsa
Universidad Autónoma de Baja California, Mexicali, Mexico
e-mail: srgnk@uabc.edu.mx; vtyrsa@uabc.edu.mx

P. Mercorelli
Leuphana University of Lueneburg, Lueneburg, Germany
e-mail: mercorelli@uni.leuphana.de

W. Flores-Fuentes
Facultad de Ingeniería Mexicali, Universidad Autónoma de Baja California, Mexicali, Baja California, Mexico
e-mail: flores.wendy@uabc.edu.mx

CI	Current image
CS	Control systems
DF	Decision function
EMR	Electromagnetic radiation
FO	False object
FR	Flying robots
FW-UAV	Fixed wings unmanned aerial vehicle
IF	Informational field
INS	Inertial navigation system
LPF	Low-pass filter
NS	Navigation system
OB	Object of binding
PM	Propagation medium
RI	Reference image
RM	Radiometric
RMI	Radiometric imaging
RW-UAV	Rotary wings unmanned aerial vehicle
SD	Standard deviation
SDPN	Sensors of different physical nature
SI	Source image
SS	Sighting surface
TNM	Technical navigation means
UAV	Unmanned aerial vehicle

16.1 Introduction

Recently, robotics is one of the most promising and extensively developing areas worldwide. The advances in robotics theory and practice are the evident mainstream in national priority development program of the main industrial countries. On the other hand, in robotics one of the newest and complex high-tech branches is the flying robot theory. Many different types of flying robots [1] like RW-UAV (rotary wings unmanned aerial vehicle) [2], Bioinspired [3], and FW-UAV [4] (fixed wings UAV) exist. These apparatus are used in very versatile practical applications, such as industrial and agricultural inspections [5, 6], search and rescue [7], environment monitoring [7, 8], security surveillance [7, 9, 10], and automated mapping [11–13].

The main difference of FR over other robots subject to automatic navigation is the significantly higher speed of change of environmental situation and, as a consequence, the enhanced requirements to the efficiency and fastness of processing algorithms.

16.1.1 Autonomous and Noise-Cancel FR Navigation Systems

Modern FR control systems are designed to control complex multifunctional objects that operate in a complex environment. FR flight control can be carried out by using autonomous, automated, and nonautonomous navigation systems. The purpose of autonomous navigation systems is reduced to the most effective detection of certain location objects, their classification (identification) within the established classes, and the issuance of appropriate commands to the executive control system [14, 15].

In FR automated navigation systems, control tasks are solved by a human operator according to the information of the processing and integrating system [15].

The FR autonomous navigation systems operate on the basis of determining the coordinates of the system's signals that accumulate errors over time (inertial, course-Doppler, etc.), with subsequent correction of the FR flight path by signals from radio navigation systems, airborne radar stations, and sight target devices [15].

Some of the core problems for the single flying robot are navigation and obstacle avoidance. Novel methods can be found in articles of authors J. Keller et al. [16] where the method of coordinated path planning for fixed-wing unmanned aerial systems was used, in [17] using computer simulations confirmed their theory of wireless network system use for collision avoidance by flying robot, and article [18] considered some principal problems on precise attitude determination of the free-flying robots and their maneuvering.

In most of cases, the main navigation system of the aircraft autonomous control system is the inertial navigation system (INS). The current position of the navigation object in the INS is established indirectly based on its initial positioning and the double integration result of the acceleration vector or single integration of the velocity vector.

Systems that implement an overview and comparative method of navigation—extreme correlation navigation systems—carry out the correction of INS errors at selected points of the movement trajectory [19].

The NS possesses the highest autonomy level, which operates without the participation of the operator at a large “over-the-horizon” range. The SN data is divided into direction finding systems that navigate small, high-contrast objects and CENS, which navigate the aircraft in two-dimensional and three-dimensional object images.

The main task in navigation of a moving object, the FR in particular, is to ensure its accurate movement along a given trajectory and accurate output to a given point at the appointed time in the most favorable way for given conditions [1, 19]. Each stage of the navigation process corresponds to the navigation mode, which is understood to be the preservation/maintenance of the direction, speed, and altitude of the FR flight. The navigation mode is determined by the combination of a large navigation element number, which represents geometric or mechanical quantities and characterizes the object movement and position.

The definition of the specified navigation elements is made on the basis of measurements using various technical navigation means (TNM), based on various

principles and intended to measure navigation parameters, which are a geometric value or one of its derivatives. The navigation parameter either coincides with the navigation element, or is associated with it by a simple relation.

Recently, autonomous systems have been increasingly used for FR navigation, the main principle of which is based on determining the navigation parameters of an object by examining information on natural geophysical fields, such as the field of relief, the field of optical, radar, and radiometric contrast, the anomalous magnetic field of the earth, the gravitational field and others [20], and using measured navigation parameters to generate correction signals for a coarse navigation system.

The navigation parameters are determined by comparing the current image (CI), which is the distribution of the physical field in the system view area, with one or several copies of pre-stored images, called reference image (RI), based on a certain statistical criterion, at which most often the reciprocal (mutual) correlation function serves. Such corrective systems are called correlation-extreme navigation systems, even in cases where an image matching criterion different from the correlation function is used.

According to the volume and nature of information removed from the physical field at each time point, CENS is classified into systems in which information is currently removed at a point (CENS-I), from a line (CENS-II), and from an area (frame) (CENS-III) [19]. Systems of the first type can use both *surface* fields (relief field, optical, radio location and radiometric contrast fields; such systems belong to the TNM type 2) and *spatial* fields (anomalous magnetic and gravitational fields of the TNM type 1). CENS-II and CENS-III can work only on surface fields, as usually FRs are small in comparison to the correlation radius of the spatial field. These systems closely adjoin the image recognition systems and image combining systems. Forming an image line in CENS-II or a frame in CENS-III is performed by scanning the field sensor or by using the horizontal component of the aircraft speed.

Further division of CENS sub-classes can be made according to the storing and processing method of a priori and working information. From this point of view, they are divided into *analog* (continuous), *digital*, and *analog-digital* (mixed).

According to the method for determining the deviation from the extreme, the criterion for comparing images of CENS are divided into *search*, *search-free*, and *combined*.

Search-free CENSes are tracking systems. For their operation, at each correction session, one copy of RI is required, and it is necessary that the mismatch (shift) of CI relative to RI does not exceed the correlation interval of the surface field [15]. Correlation processing in such systems is carried out under the condition of homogeneity and ergodicity of the surface field, and the dimensions of the field of view of sensor must significantly exceed the field correlation interval. Search-free CENSes are suitable to use when these conditions are met in the capture mode of a ground object—the navigation guideline. It is assumed that in this mode, the probability of missing a navigation landmark is small, and the main characteristic of the quality of the system is the accuracy of the controlled object (FR navigation) on a given trajectory.

In the search ACS [19], the extreme position of the image matching criterion is carried out by searching for it, which, unlike tracking systems, requires performing test motions necessary to determine the direction of movement to the extreme. This is achieved by shifting the field of view of the sensor. In CENS, it is more convenient to use different fragments from the RI set that differ from each other in shear when comparing test motions. As a result, the RI fragment, which corresponds to the greatest degree of CI, is recognized, and the value of the shift vector is determined at the same time as the image sampling interval on the ground, which is determined by the impulse spatial characteristic of the field sensor. The main characteristic of the quality of the system at this stage is the probability of correct recognition of the ground object, and the accuracy of determining the navigation parameters plays a secondary role.

The combined (recurrent-search) [19] are automated control systems (ACS) that combine the principles of Kalman filtering with the theory of statistical solutions, more precisely, its branch testing statistical hypothesis. The instance of RI choice acts as a hypothesis, which differs by a certain shift. As a comparison criterion of CI and RI, the conditional relative to observation joint probability density of the shift parameter and the hypothesis appears.

Depending on the type of physical field used, CENSEs are subdivided into *radar*, *optoelectronic*, *radiometric*, *relief-based*, etc.

Depending on the used physical field formation method, CENS can be *active*, *passive*, and *semi-active* (*passive-active*).

16.1.2 A Formalized Basic Description Model for the Functioning Process of the FR with Correlation-Extreme Navigation Systems and Radiometric and Optoelectronic Sensors

The basic model of the combined CENSEs functioning process, we will understand as some idealized model of the system, describes the conversion signal patterns arising from the solvable tasks, but does not take into account the interference characteristic specifics of radiometric reception or reception by optoelectronic sensors. It is assumed that the only interference of receiving signals is the internal noise of the receiver, and various destabilizing factors distorting the CI structure (changing weather conditions, underlying surface conditions, etc.) are absent.

16.1.2.1 The Main Objectives and Model of Signal Processing in the RM Channel CENS

In accordance with the theory of thermal radiation in the radiometric (RM) channel, the selection of information about the object of binding (OB) is based

on the difference in emissivity of materials and underlying surfaces. Therefore, the recorded (informative) signal parameter used for detection, identification, recognition, mapping, etc., is the intensity of the received signal.

The main problem solved by the RM channel at each session of the binding trajectory is to estimate the coordinates of the OB characteristic point (X_{OB}, Y_{OB}) based on the processing of the heat thermal relief $T(x, y)$ in the view field $S \subset R^2$ and using a priori information in the form of RI about one of the view field parts, that is, display formation $S \rightarrow (X_{OB}, Y_{OB})$. This task is divided into a number of sub-tasks:

1. Review of the underlying surface in the OB area
2. Formation in a certain subject plane associated with the FR coordinate system, a radio-thermal relief of the area within the field of view, that is, radiometric imaging (RMI)
3. RMI analysis in order to identify anomalies due to the presence of the OB, based on its comparison with the RI, determining the coordinates of the selected anomaly characteristic point, and issuing target indications of the CS.

On the basis of these data and information about the height at the time of the frame pickup received from the altimeter, the coordinates for the FR mass center and the mismatch angles of the FR velocity vector with the OB direction are calculated. This information, in the form of target indications, enters the control system and is used to correct the FR trajectory.

Let in the coordinate system (x, y) on the earth surface be given the field of the heat (radio) thermal relief $T(x, y)$. A multi-path antenna converts radiothermal radiation with an intensity $T(x, y)$ into a combination of processes $\{u_{s_{ij}}(t)\}_{i=1, N_2}^{N_1, N_2}$ with two-sided spectral power densities

$$S_{s_{ij}}(f) = kT_{s_{ij}}/2, \quad (16.1)$$

and the antenna temperature $T_{s_{ij}}$ for each channel is determined by the expression

$$T_{s_{ij}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(x, y) G(x_{ij} - x, y_{ij} - y) dx dy, \quad i \in \overline{1, N_1}, \quad j \in \overline{1, N_2}, \quad (16.2)$$

where $G(x_{ij} - x, y_{ij} - y)$ is the function describing the ij -th partial ADD, converted to a coordinate system on the surface of the earth, in which the axis intersects the earth surface at a point (x_{ij}, y_{ij}) .

Thus, the heat radiothermal relief distribution undergoes the information loss due to image discretization following the finiteness of the processing channel number and the image “blurring” of the final width of each partial ADD, which leads to a decrease in the accuracy and reliability of the positioning.

Let further signals be processed by a multichannel matrix radiometer, which uses independent radiometric channels. In each channel, the signal is amplified in the amplification path to the level necessary for normal operation of the quadratic detector. Before amplification, the signal is summed with interference due to the antenna noise, the input circuits, and the intrinsic noise of the amplification path. Assuming that the frequency path response is rectangular with a bandwidth Δf and center frequency f_0 , the signal at its output can be represented as

$$u_{ij}(t) = u_{sij}(t) + u_{nij}(t), \quad (16.3)$$

where $u_{sij}(t)$, $u_{nij}(t)$ are the band Gaussian random processes with spectral power densities

$$\begin{aligned} S_{sij}(f) &= kT_{sij} \left[\text{rect}(f + f_0, \Delta f) + \text{rect}(f - f_0, \Delta f) \right] / 2; \\ S_{nij}(f) &= kT_{nij} \left[\text{rect}(f + f_0, \Delta f) + \text{rect}(f - f_0, \Delta f) \right] / 2, \end{aligned} \quad (16.4)$$

where $k = 1.38 \cdot 10^{-23} \text{ J.K}^{-1}$ is the Boltzmann constant,

$$\text{rect}(f, \Delta f) = \begin{cases} 1, & |f| \leq \Delta f/2, \\ 0, & |f| > \Delta f/2, \end{cases} \quad (16.5)$$

T_{nij} is the equivalent temperature of internal noise.

The signal processing algorithm, for example, in an ideal compensation matrix radiometer is as follows:

$$\left\{ u_{ij}(t) \mapsto \hat{T}_{sij} \right\}; \quad \hat{T}_{sij} = \frac{1}{k\Delta f\tau} \int_0^\tau u_{ij}^2(t) dt - T_{nij}; \quad i \in \overline{1, N_1}, \quad j \in \overline{1, N_2}, \quad (16.6)$$

where \hat{T}_{sij} is the antenna temperature estimate for the ij -th channel.

In the signal optimal reception theory with fluctuation noise, it is customary to use the square root of the mean square deviation of the measured value (estimate) of the antenna temperature \hat{T}_{sij} from its true value as an indicator of noise immunity T_{sij} , that is,

$$\delta T_{ij} = \left[\mathbf{M} \left(\hat{T}_{sij} - T_{sij} \right)^2 \right]^{1/2}. \quad (16.7)$$

By a direct averaging of expression (16.6), we can verify that the estimate \hat{T}_{sij} is unbiased, that is, $\mathbf{M}\hat{T}_{sij} = T_{sij}$. Then the noise immunity index (16.7) coincides with the standard deviation \hat{T}_{sij}

$$\delta T_{ij} = \left(\mathbf{D} \hat{T}_{s_{ij}} \right)^{1/2}. \quad (16.8)$$

Using (16.6), direct calculations can show that

$$\delta T_{ij} = \frac{T_{s_{ij}} + T_{n_{ij}}}{\sqrt{\Delta f \tau}}. \quad (16.9)$$

Thus, the signal at the output of the radiometric channel can be represented as

$$\hat{T}_{s_{ij}} = T_{s_{ij}} + n_{ij}, \quad (16.10)$$

where n_{ij} is the fluctuation component of the output signal. Since the integrator in (16.6) is a low-pass filter, by virtue of the well-known theorem on the random process normalization at the output of a narrow-band filter, n_{ij} is a Gaussian random variable with zero mean value and standard deviation (16.9).

The set of output signals of radiometric channels forms $N_1 \times N_2$ CI matrix $\left\{ \hat{T}_{s_{ij}} \right\}$, on the basis of comparing fragments with the RI matrix $\mathbf{E} = [e_{ij}]$, $i \in \overline{1, M_1}$, $j \in \overline{1, M_2}$, $M_1 < N_1$, $M_2 < N_2$, in the secondary processing device using an appropriate algorithm, and it is decided to localize the OB by searching for the CI fragment most relevant to RI.

Fragment by the author is called the fragment CI, which actually corresponds to RI. For this fragment, the relation (16.10) takes the form

$$\hat{T}_{s_{ij}}^{kl} = e_{ij} + n_{ij}^{kl}, \quad (16.11)$$

where (k, l) are the coordinates of the author fragment.

The coordinates of the author fragment it is a value that determines the position of the current image quote on the anchoring area plane on the surface of the earth. The coordinates are known in advance as far as the reference image of the anchoring area (object) is forming beforehand.

Thus, the secondary processing system of the RM channel forms the mapping $\left\{ \hat{T}_{s_{ij}} \right\} \mapsto \left(\hat{k}, \hat{l} \right)$ of the CI matrix into the coordinate estimate of the author fragment.

As the indicator of a secondary processing device noise immunity, we will use the probability of the OB correct localization P_{CL} . Let us reveal the meaning of this concept.

On the i -th session of the trajectory binding ($i \in \overline{1, N_t}, N_t$ the total number of binding sessions (frames)), the RI estimated position \hat{k}_i, \hat{l}_i is obtained relative to CI, and the true values are equal to k_i, l_i , ($k_i \in \overline{1, N_1 - M_1 + 1}$, $l_i \in \overline{1, N_2 - M_2 + 1}$). Let us denote by the A_i event that the i -th binding session for the absolute error ($\Delta k_i = \hat{k}_i - k_i$, $\Delta l_i = \hat{l}_i - l_i$) satisfies the condition

$$|\Delta k_i| < 1, \quad |\Delta l_i| < 1. \quad (16.12)$$

Then this event will be called the correct localization of the OB at the i -th session of the trajectory binding. When this event is performed, the accuracy of the trajectory reference will be no worse than the resolution element on the ground, determined by the width of the beam, the height of the FR at the time of the shot, and the angle of OB sight.

For the first binding session, we have $P_{CL_1} = P(A_1)$. In the second session, the correct localization probability of the correct OB is the probability of combining events A_1 and A_2 , which is determined by the multiplication rule of probabilities

$$P_{CL_2} = P(A_1 \cap A_2) = P(A_1) P(A_2 | A_1).$$

Similarly, at the i -th session

$$P_{CL_i} = P\left(\bigcap_{j=1}^i A_j\right) = P(A_1) P(A_2 | A_1) \cdots P\left(A_i \left| \bigcap_{j=1}^{i-1} A_j\right.\right).$$

Then the final probability of the correct OB localization will be determined by the expression:

$$P_{CL} = P_{CL_{N_t}} = P\left(\bigcap_{j=1}^{N_t} A_j\right) = P(A_1) P(A_2 | A_1) \cdots P\left(A_{N_t} \left| \bigcap_{j=1}^{N_t-1} A_j\right.\right). \quad (16.13)$$

Let us estimate the signal influence and interference components in the image (16.10) on the probability of the correct OB localization. For simplicity, we will consider events A_i mutually independent. Then (16.23) takes the form

$$P_{CL} = \prod_{i=1}^{N_t} P(A_i). \quad (16.14)$$

We will accept next suppositions and assumptions:

1. RI is a sample of a one-dimensional Gaussian ergodic process with zero mean and variance σ_e^2 .
2. An additive model of CI interaction with the kind of noise (16.11) is applied

$$z_i = e_i + n_i;$$

3. The noise n_i has the same characteristics as RI, except for dispersion, which is equal to σ_n^2 and it is the same for all channels.
4. The processes z and n are independent.
5. To compare the fragments of RI with CI, the mean absolute difference algorithm is used, whose criterion function is

$$D_j = \frac{1}{N} \sum_{i=1}^N |e_{i+j} - z_i|, \quad (16.15)$$

where N is the number of elements CI.

For these suppositions and assumptions the probability of correct OB localization during single act of object binding will be determined by the following expression:

$$P_{CL} = \frac{1}{2^M \sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2} \left[1 + \operatorname{erf} \left(\frac{x + \sqrt{\frac{N}{\pi-2}} (\sqrt{2q+1} - 1)}{\sqrt{2q+1}} \right) \right]^M dx, \quad (16.16)$$

where M is the number of independent samples of the criterion function

N is the number of independent images samples

$q = \sigma_e^2 / \sigma_n^2$ is the signal-to-noise ratio

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

From expression (16.16), it follows that

1. The probability of the correct localization of the OB increases with increasing signal-to-noise ratio.
2. The probability of correct localization of OB increases with increasing N and decreases with increasing M .

However, the expression (16.16) does not take into account a number of factors that will lead to a decrease in the probability of correct OB localization.

16.1.3 Analysis of Factors That Lead to Distortions in a Decisive Function Formed by a Correlation-Extreme Navigation System

The task solution of high-precision FR autonomous positioning that uses combined CENS should be carried out primarily in the informative parameter bundle of the system “three-dimensional shape of objects on CI—sensors of different physical nature (SDPN)—geometric position of FR, taking into account its random change” under the influence of

- Various types of CI interference and distortion, which may be natural or artificial
- Propagation medium (PM) of radio waves and the interference effect on the state of the navigation system (NS)

The CENS FR efficiency functioning is determined by the decisive function (the term for the correction of the spatial FR position) and is estimated by the accuracy parameters and positioning probability [21].

The FR locating probability using the CENS in the k -th session of the binding is determined by the expression:

$$P_{m_i} = P_{CL_i} \cdot P_{C_i}, \quad (16.17)$$

where P_{CL_i} is the probability of the correct OB localization on the CI.

$P_{C_i} = P_W \cdot P_{IP} \cdot P_{CS}$ is the performing probability of the FR flight path correction at the k -th session of the binding, which is determined by the following parameters: P_W is the influence probability of weather conditions on the control system (CS) LR. P_{IP} is the interference possibility on the FR (CS) functioning. P_{CS} is the failure-free operation probability of the FR CS.

The accuracy index (positioning error) of the CENS is characterized by the standard deviation (SD) of the real FR coordinates after performing the k -th correction with respect to the given one [19, 21].

Accuracy index CENS FR can be represented by the expression:

$$\sigma_k = \sqrt{\sigma_{CL_i}^2 + \sigma_{C_k}^2 + \sigma_{CS_k}^2}, \quad (16.18)$$

where $\sigma_{CL_i} = f(\sigma_{CI}, \sigma_{x_i y_i z_i})$ is the SD of the OB localization on the CI at the k -th binding session, which depends on the RI manufacturing accuracy σ_{RI} and the errors in determining the spatial position of the FR, arising under the influence of random factors $\sigma_{x_i y_i z_i}$.

σ_{C_k} is the deviation of the coordinates after performing the correction in the k session of the reference CENS.

σ_{CS_k} is the SD of testing control signals after the correction FR flight path.

The random factor influence that leads to errors in determining the FR spatial position requires the appropriate method development on the formation of the CENS DF and the FR positioning.

Probability σ_{CL_i} is determined by many factors. The probability is determined by many factors, such as three-dimensional form of objects of SS, the influence of FR spatial position instability, that determine the quality of SI, and is formed by the sensors of CENS [22].

The need to take into account the three-dimensional shape of SS objects, especially in conditions of the sight geometry changes, is caused by the object-saturated images of SS. Such images are very characteristic for the developed infrastructure, shadows and blurring contours. These factors, as well as the rapid change in the spatial position of the FR due to the wind gusts, air holes and air flows, can also contribute to the distortion of the DF, which in its turn leads to a decrease of the CENS operation efficiency.

16.1.4 The Changes Impact Analysis in the FR Spatial Position on the CI Formation

Preserving the reasoning integrity intact, we will refine the CI model, which is formed by the channels of the combined CENS using the example of an RM channel with an informative parameter—a radiobrightness temperature, followed by the generalization to an optoelectronic channel [23].

The CI model building will be carried out taking into account the following conditions:

1. The FR flies evenly and straightforwardly at speed V at an angle φ to the vertical.
2. The ADD is approximated by a Gaussian surface.
3. The CI frame formation is carried out by a multi-path (matrix) system.
4. The CI is formed in accordance with (16.11) in each channel under the influence of additive noise.
5. The effect of changing the FR spatial position on the CI is carried out by the yaw angle $\psi' = \psi \pm \Delta\psi$.
6. The angles of pitch and roll do not change.

The first condition of the current image modeling indicates that the flying robot does not make the maneuver while forming the image frame. That is why there will be no immediate changes in roll and pitch angles. In addition, the orienting system that provides constant roll and pitch angles is a part of the flying robots' strapdown inertial reference systems. That is why roll and pitch angles do not affect the image frame forming. So the sixth condition about consistency of roll and pitch angles reflects reality.

Suppose that the FR moves in the xz plane of the x, y, z coordinate system associated with the SS (Fig. 16.1a). The position of each partial ADD is characterized by angles β and α . The aperture angle of the ADD at the half power level is θ_x in the plane along the elevation angle and θ_y in the azimuth plane. For the ADD Gaussian approximation, its section by the yx plane is an ellipse.

The CI is formed as a matrix of M rows and N columns. The inclination of the plane in which the ADD are located along the axes is given by the angle $\beta_{i,j} \in \overline{1, M}$ relative to the velocity vector V , and the position of the axis of each ADD in the row is characterized by the angle α_{ij} .

The distortions of the CI frame can be found from the motion equation of the centers and sizes of the principal semi-axes of half-power ellipses for each beam of the ADD.

In accordance with the motion direction (Fig. 16.1), the FR spatial position at the moment of time t is represented as:

$$\begin{cases} x(t) = x_0 + V \cdot (t - t_0) \sin \phi; \\ y(t) = 0; \\ z(t) = z_0 - V \cdot (t - t_0) \sin \phi. \end{cases} \quad (16.19)$$

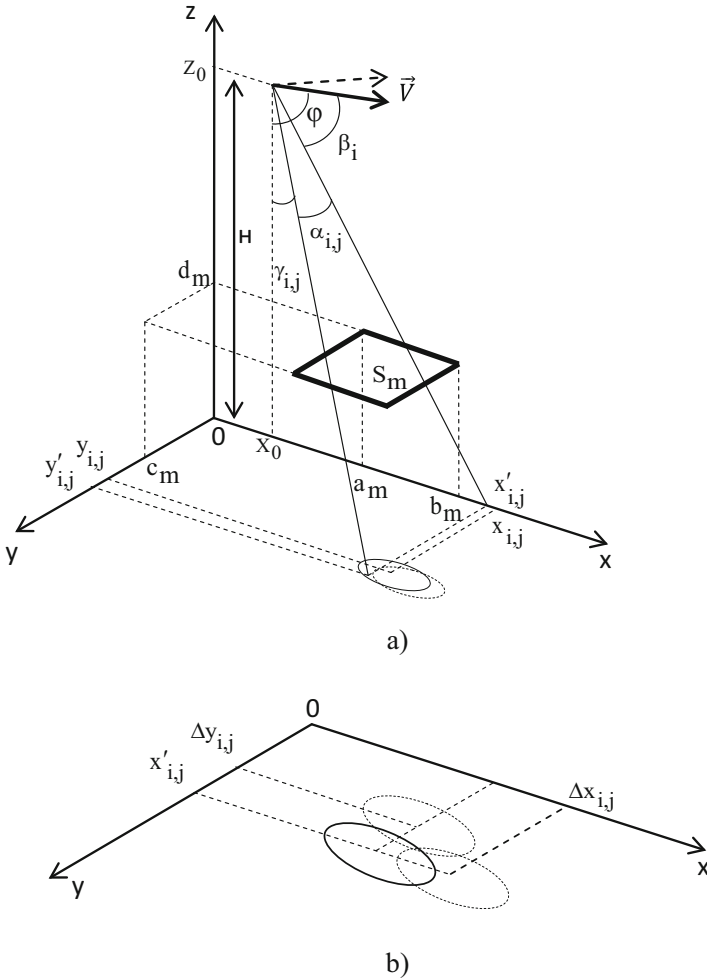


Fig. 16.1 Geometric conditions for the formation of CENS CI

In view of (16.19), the motion equation of the ellipse center for the ij -th partial ADD is written as

$$\begin{cases} x_{ij}(t) = k_{ij}x'_{ij}(t); \\ y_{ij}(t) = k_{ij}y'_{ij}(t), \end{cases} \quad (16.20)$$

where $x'_{ij}(t) = x_0 + z_0 t g(\phi - \beta_i) + V(t - t_0) \sin \beta_i \sec(\phi - \beta_i)$

$$y'_{ij}(t) = z(t) t g(\alpha_{ij}) \sec(\phi - \beta_i); [0, 1]$$

$$k_{ij} = \left\{ 1 - \left[\frac{\sin(\theta_{x_{ij}}/2)}{\cos \alpha_{ij} \cos(\phi - \beta_i)} \right]^2 \right\}^{-1}$$

The dimensions of the main ellipse semi-axes are determined by the relations:

1. In a plane that passes through points $(x_0, 0, z_0)$, $(x_0, 0, 0)$, $(x_{ij}, y_{ij}, 0)$

$$\Delta x_{i,j}(t) = z(t) (k_{i,j}(t) - 1) \operatorname{ctg}(\theta_{x_{i,j}}/2); \tag{16.21}$$

2. In the orthogonal plane

$$\Delta y_{ij}(t) = z(t) \operatorname{tg} \frac{\theta_{y_{ij}}}{2} \sqrt{(k_{ij} - 1) \left(k_{ij} \operatorname{ctg}^2 \frac{\theta_{x_{ij}}}{2} - 1 \right)}. \tag{16.22}$$

Assume that the principal axes of the half-power ellipses are parallel to the x, y axes. For these conditions, the normalized ADD at a point $(x_0, 0, z_0)$, in which the axis intersects the xz plane at a point (x_{ij}^0, y_{ij}^0) , can be represented as follows [19]:

$$G(x, y, x_{ij}^0, y_{ij}^0) = \frac{1}{2\pi \delta_{x_{ij}} \delta_{y_{ij}}} \exp \left\{ - \left[\frac{(x - x_{ij}^0)^2}{2\delta_{x_{ij}}^2} + \frac{(y - y_{ij}^0)^2}{2\delta_{y_{ij}}^2} \right] \right\}. \tag{16.23}$$

We select the parameters $\delta_{x_{ij}}, \delta_{y_{ij}}$, so that the main ellipse semi-axes dimensions of the ADD half-power coincide with the dimensions that are determined by the formulas (16.21) and (16.22):

$$\frac{(x - x_{ij}^0)^2}{2 \ln 2 \delta_{x_{ij}}^2} + \frac{(y - y_{ij}^0)^2}{2 \ln 2 \delta_{y_{ij}}^2} = 1. \tag{16.24}$$

As a result, we get

$$\delta_{x_{ij}} = \frac{\Delta x_{ij}}{\sqrt{2 \ln 2}}; \quad \delta_{y_{ij}} = \frac{\Delta y_{ij}}{\sqrt{2 \ln 2}}. \tag{16.25}$$

Let us find the dependence of the brightness temperature in time at the output of a separate RM channel. To do this, suppose there are K zones with brightness temperatures T_m , on the CI that are distributed on a uniform background with the temperature T_B . Then, in the xy plane, the following brightness distribution temperatures will occur:

$$T_{Br}(x, y) = \begin{cases} T_m, & x, y \in S_m, \quad m \in \overline{1, K} \\ T_B, & x, y \notin S_m = \bigcup_{m=1}^K S_m \end{cases}, \tag{16.26}$$

where $S_m \cap S_n = \emptyset, \quad m \neq n$.

Let us take into account that the radiometer low-pass filter (LPF) has an impulse response $h_{ij}(t) = \frac{1}{\tau_{ij}} \exp(-t/\tau_{ij})$. Then the signal at the output of the RM channel is represented in the form:

$$T_{S_{i,j}}^r = e^{-(t-t_0)/\tau_{i,j}} \left[T_{S_{eff}}(t_0) + \frac{1}{\tau_{ij}} \int_{t_0}^t e^{\eta/\tau_{ij}} T_{S_{eff}}(ij)(\eta - t_0) d\eta \right], \tag{16.27}$$

where $T_{S_{eff}ij}$ is the antenna temperature at the ij -th input

τ_{ij} is the time constant of the ij -th channel

Taking into account that the antenna system parameters $T_{S_{ij}}$ can be represented as follows [19]:

$$T_{S_{ij}}(t) = \int_{R^2} T_{S_{eff}}(x, y) G(x, y, x_{i,j}(t), y_{i,j}(t)) dx dy. \tag{16.28}$$

Taking into account (16.26), we write (16.28) as follows:

$$T_{S_{ij}}(t) = T_{\phi} + \sum_{m=1}^R (T_m - T_{\phi}) \int_{S_m} G(x, y, x_{i,j}(t), y_{i,j}(t)) dx dy. \tag{16.29}$$

Taking into account (16.23), (16.27), and (16.28), after integration in the range from $t_0 = t - 3\tau$ to t , we obtain:

$$T_{S_{ij}}^r(t) = T_{\phi} + \sum_{m=1}^K (T_m - T_{\phi}) \Phi \left(\frac{x - y_{ij}(t_0)}{dy} \right) \Big|_{x=c_m}^{d_m} \left[\Phi \left(\frac{x - x_{ij}(t)}{dx} \right) - B_{ij}(t, x) \right] \Big|_{x=a_m}^{b_m}, \tag{16.30}$$

where $B_{ij}(t, x) = \exp \left(\frac{r_{ij}^2}{2} + \frac{x - x_{ij}(t_0)}{dx} r_{ij} - \frac{t - t_0}{\tau_{ij}} \right) \left[\Phi \left(\frac{x - x_{ij}(h)}{dx} + r_{ij} \right) \right]_{h=t_0}^t$

$$r_{ij} = \frac{\delta_{x_{ij}}}{V_{x_{ij}} \tau_{ij}}$$

$$\Phi(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} e^{-\theta^2/2} d\theta$$

$$i \in \overline{1, M}; \quad j \in \overline{1, N}$$

Expression (16.30) is a signal model that is formed by a separate RM channel depending on the spatial position of the FR and its orientation, without taking into account the channel noise.

The expression that describes the CI model taking into account the noise in the radiometric channel can be represented as [23]:

$$S_{RM} = \left\| S_{RM_{i,j}} \right\|_{\substack{i = \overline{1 \dots M} \\ j = \overline{1 \dots N}}}, \tag{16.31}$$

where

$$S_{RM}(i, j) = T_b + \sum_{m=1}^K (T_m - T_b) \Phi\left(\frac{\xi - y_{ij}(t_0)}{\delta y}\right) \left| \begin{matrix} d_m \\ \xi = c_m \end{matrix} \left[\Phi\left(\frac{\xi - x_{ij}(t)}{\delta x}\right) - B_{ij}(t, \xi) \right] \right|_{\xi = a_m} b_m + n(t).$$

The CI model in the optoelectronic channel in accordance with (16.31) can be represented as follows:

$$S_{OE} = \left\| S_{OE_{i,j}} \right\|_{\substack{i = \overline{1 \dots M} \\ j = \overline{1 \dots N}}}, \tag{16.32}$$

$$S_{OE}(i, j) = B_b + \sum_{m=1}^K (B_{Br_m} - B_{Br_b}) \Phi\left(\frac{\xi - y_{i,j}(t_0)}{\delta y}\right)$$

where

$$\left| \begin{matrix} d_m \\ \xi = c_m \end{matrix} \left[\Phi\left(\frac{\xi - x_{i,j}(t_0)}{\delta x}\right) - B_{i,j}(t, \xi) \right] \right|_{\xi = a_m} b_m + n(t)$$

The signal brightness $B_{\text{Br}}(i, j, t, \varepsilon, \mu, \varpi)$ which is received by the CENS OE sensor from the SS element at a point (i, j) in time t is described by the expression [6]

$$B_{\text{Br}}(i, j, t, \varepsilon, \mu, \varpi) = E(i, j, t, \varepsilon, \mu, \varpi) r_{\text{Br}}(i, j, t, \varepsilon, \mu, \varpi), \quad (16.33)$$

where $E(i, j, t, \varepsilon, \mu, \varpi)$ is the spectral illumination field, which is created by the image element (i, j)

$r_{\text{Br}}(i, j, t, \varepsilon, \mu, \varpi)$ is the spectral brightness coefficient
 ϖ is the observation and illumination condition vector

$$\varpi = \|\phi \ \varphi \ \omega \ \psi \ E_{\text{dir}}/E_{\text{dif}}\| \quad (16.34)$$

ϕ and φ are the angles of the SS element observation

ω and ψ are the angles of the SS element illumination

E_{dir} and E_{dif} are random illumination fields, which are created by direct and diffuse radiation

Taking into account (16.33), the model of the image of the SS formed by the OE sensor can be represented as:

$$S_{\text{SSOE}}(i, j) = B_{\text{Br}}(i, j, t, \varepsilon, \mu, \phi, \varphi, \omega, \psi, E_{\text{dir}}, E_{\text{dif}}), \quad (16.35)$$

where ε, μ are the dielectric and magnetic permeability of covers and SS materials.

The PM distortion effect leads to the modulation of the informational field (IF) parameters and can be described by the functional [20].

$$\mathbf{S}_{\text{PM}_k}(t) = \mathbf{B}_{\text{PM}}(\mathbf{S}_{\text{SS}_k}(t)), \quad (16.36)$$

where $\mathbf{B}_{\text{PM}_k}(\mathbf{S}_{\text{SS}_k}(t))$ is the image conversion operator by the EMW propagation medium.

The FR can be exposed to the natural noises, intentional interference, and high-power electromagnetic radiation (EMR).

As a result, the input k -th channel signal of a sensor of different physical nature (SDPN) $\mathbf{S}_{\text{SDPN}_k}$ can be represented in the form:

$$\mathbf{S}_{\text{SDPN}_k}(t) = \mathbf{S}_{\text{PM}_k}(t) + \mathbf{N}_{\text{FR}_k}(t), \quad (16.37)$$

where

$$\mathbf{N}_{\text{FR}_k}(t) = \mathbf{n}_k(t) + \mathbf{N}_{\text{IN}_k}(t) + \mathbf{N}_{\text{EMR}_k}(t) \quad (16.38)$$

$\mathbf{N}_{FR_k}(t)$ is the FR additive effect matrix on the k -th reception channel

$\mathbf{n}_k(t)$ is the receiver's own noise matrix and artificial noise in the k -th reception channel

$\mathbf{N}_{IN_k}(t)$ is the active and passive interference affecting the receiver's path in the k -th channel of the SDPN

$\mathbf{N}_{EMR_k}(t)$ is the EMR impact on the channel

The result of the primary information processing in the CENS, which consists of CI set formation, noise filtering, as well as noise compensation, in general terms can be represented by the relation:

$$\mathbf{S}_{CI_k} = \mathbf{F}_{pre_k} (\mathbf{S}_{SDPN_k}(t)), \tag{16.39}$$

where \mathbf{F}_{pre_k} is an operator that describes the process of preprocessing informative fields in the k -th SDPN.

Taking into account the RI presence setup in advance for each SDPN, that is $\mathbf{S}_{RI_1}, \mathbf{S}_{RI_2}, \dots, \mathbf{S}_{RI_G} \in \mathbf{S}_{RI}$, the DF CENS in a general form can be represented as the result of an RI correlation comparison \mathbf{S}_{RI_g} from the formed aggregate \mathbf{S}_{RI} with the CI which are formed by the SDPN $(\mathbf{S}'_{RI_1}, \mathbf{S}'_{RI_2}, \dots, \mathbf{S}'_{RI_K})$.

$$\mathbf{R}_{DF}(t, \mathbf{r}) = \mathbf{F}_{sp} (\mathbf{S}'_{RI_1}(t), \mathbf{S}'_{RI_2}(t), \dots, \mathbf{S}'_{RI_K}(t); \mathbf{S}_{RI}), \tag{16.40}$$

where \mathbf{F}_{sp} is the secondary processing system operator.

Because of the mentioned above, the block diagram of the model of DF formation process in the operator form can be presented as shown in Fig. 16.2.

With the sequential use of CENS sensors, DF is determined by one of the sensors operating at a given point in time. This means that the DF corresponds to one of the private DFs that are formed by the separate CI channels.

$$\mathbf{R}_{DF}(t, \mathbf{r}) = \mathbf{F}_{sp_k} (\mathbf{S}'_{CI_k}(t), \mathbf{S}_{RI_k}). \tag{16.41}$$

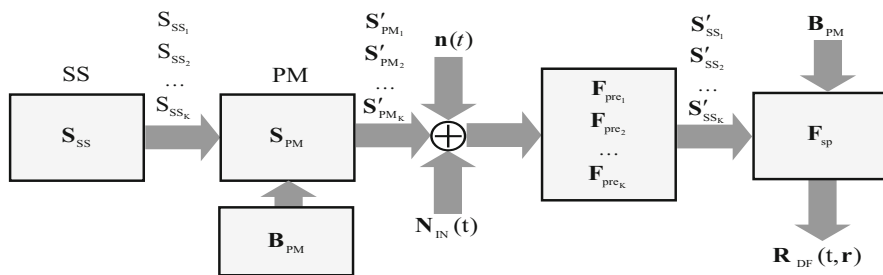


Fig. 16.2 Block diagram of a generalized model of the formation of the DF

16.2 Formation Features of a Crucial Function by a Radiometric CENS Channel When Navigating Low-Flying FR in Terms of Three-Dimensional Object Binding

The autonomous navigation solution of low-flying flying robots in the conditions of a developed infrastructure using the CENS makes it necessary to clarify the description of both the RI set and the CI three-dimensional objects. This is due to the fact that the solid OB angle can significantly exceed the size of the partial ADD. As a result, depending on the sighting geometry within the OB, various differences in the brightness temperature can occur, which in turn will lead to the formation of a nonstationary CI. Thus, a structural discrepancy arises between the RI and CI, which necessitates the use of other additional invariant OB image features. In this regard, ensuring the required accuracy of determining the OB coordinates requires the development of appropriate methods for the RI formation. It should be considered that the RI constructing principles by the informative features must comply with the CI.

The CI quality can also be affected by various interference, seasonal factors and adverse weather conditions, which lead to a change in the image structure.

16.2.1 Formation of Reference Images of Three-Dimensional Form Object Binding

The task of development of the method and algorithm for the formation of reference images of three-dimensional objects of sight will be solved with the following assumptions [24–26]:

1. The effect of distortion factors on the sighting surface objects is absent.
2. The size of the source image (SI) of the sighting surface: $M_1 \times M_2$, the size of the slide window $S_{SW} \in S_{SI} - N_1 \times N_2$ with coordinates of the upper corner (i, j) .
3. Each i -th, j -th element of the CI is a normally distributed magnitude with a dispersion σ^2_{ij} and an average radiobrightness temperature $S(i, j)$.
4. Noise of the CENS channels is not taken into account.
5. Comparison of SI of SS S_{SI} with the formed image fragment will be carried out at a maximum value of the coefficient of cross correlation (CCC), $K_{\max}(i, j)$.

CCC of SI and the formed image fragment for all $i = 1 \dots M_1 - N_1$ and $j = 1 \dots M_2 - N_2$ forms a correlation analysis field (CAF). The formed CAF characterizes the degree of similarity of informative fields of the SS image fragment with the IF of other image fragments.

Let us analyze the possibility of using radio brightness contrasts between objects in a CI as an informative attribute used to form zonal structure of the image in conditions when spatial angle of the object of sight exceeds the size of partial ADD. Taking into account influence of radiobrightness temperature of atmospheric column, T_{atm} , reflected by visible parts of the three-dimensional object, radiobrightness temperature of the object of sighting, T_m (16.26), will be determined by temperatures of its individual visible surfaces. Proceeding from this, the expression for determining radiobrightness temperature of the object of a complex three-dimensional form can be represented as [23]:

$$T_m = \frac{T_0 \sum \chi_i S_{i(\chi)} + T_{\text{atm}} \sum k_j S_{j(k)}}{S_0}, \quad (16.42)$$

where $S_0 = \sum_{i=1}^n S_{i(c)} + \sum_{j=1}^m S_{j(k)}$ is the area of visible sections of surface of the object of sighting which is characterized by emissivity and reflectivity

k_j is the coefficient of reflection

T_0 is the thermodynamic temperature of the object

Relation (16.42) makes it possible to calculate the value of radiobrightness temperature in individual elements of the object of sighting taking into account its configuration.

For this purpose, perform simulation of distribution of radiobrightness temperature between the object elements depending on sighting angles for the three-dimensional object shown in Fig. 16.3.

Simulation terms are [23]:

1. Sighting altitude: 500 m
2. Radiobrightness temperature of the sky: 50 K
3. Thermodynamic temperature: 300 K
4. Angles of opening of radiometer ADD: $30^\circ \times 40^\circ$
5. Width of partial ADD: 1°
6. Working wavelength: 3.2 mm (frequency 94 GHz)
7. Sighting corners: $90^\circ, 60^\circ, 45^\circ$
8. Pixel dimensions in the image: 8×8 m

Parameters of the object of sighting:

1. Three-dimensional object of a complex form (Fig. 16.1)
2. Dimensions of the object of sighting (in nadir) 10×30 m
3. The area of individual object elements varies according to the sighting angles
4. Emissivity of the object elements:
 - Horizontal platforms (concrete): 0.76
 - Vertical platforms (brick): 0.82

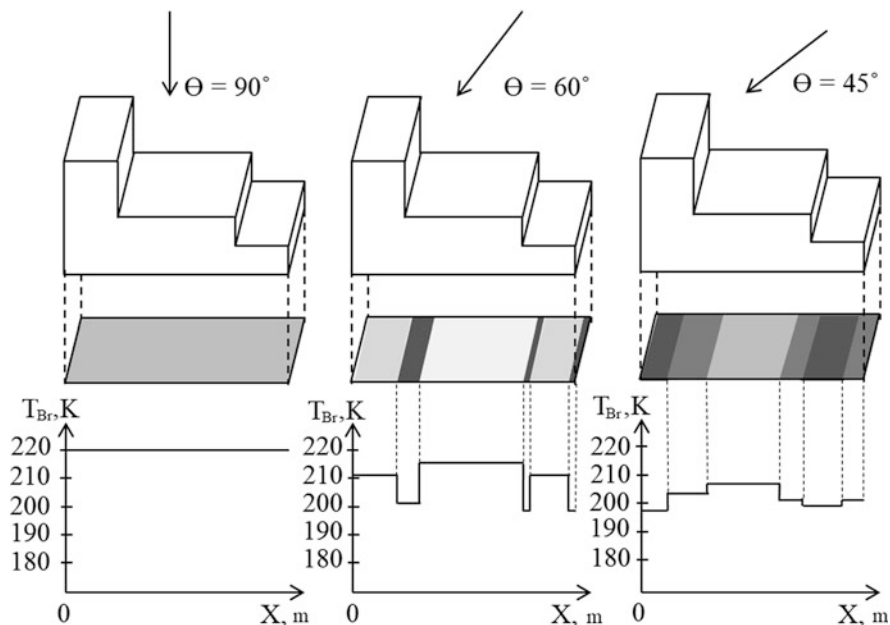


Fig. 16.3 Radiobrightness temperature distribution among the object elements depending on the sighting angle

Background parameters:

1. Material: asphalt
2. Emissivity: 0.85

According to the results of simulation, it was found that different radiobrightness temperature gradients can be observed within one object at different sighting angles. These gradients lead to disappearance of existing and emergence of new radiobrightness contrasts and, accordingly, boundaries and contours in the object images. Thus, formation of RI with the use of boundaries and contours as invariants is inappropriate in conditions of developed infrastructure. Therefore, a necessity of search for additional informative attributes for the formation of RI of SS arises.

It is suggested to use geometric attributes of the set of the most bright stationary objects of SS as such attributes. To this end, it is necessary to introduce the concept of geometrically connected objects through their contouring and subsequent definition of an average radiobrightness temperature for such equivalent object.

This approach makes it possible to refuse from transformation of similarity in the reference space for a large number of shifted and turned RIs to select an RI which corresponds most closely to the compared fragment when comparing with CI fragment.

Figure 16.4 shows SI fragment and the objects that are defined for the formation of an equivalent OR based on three bright areas of the terrain.

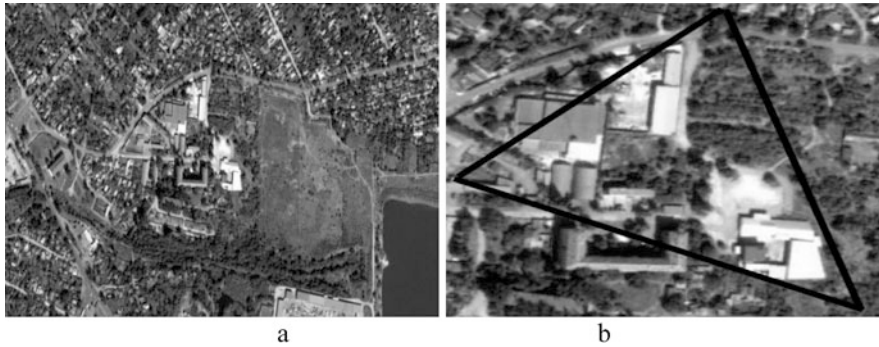


Fig. 16.4 Source image: full-size image (a); a fragment of SI and the objects that are defined for the formation of an equivalent OR (b)

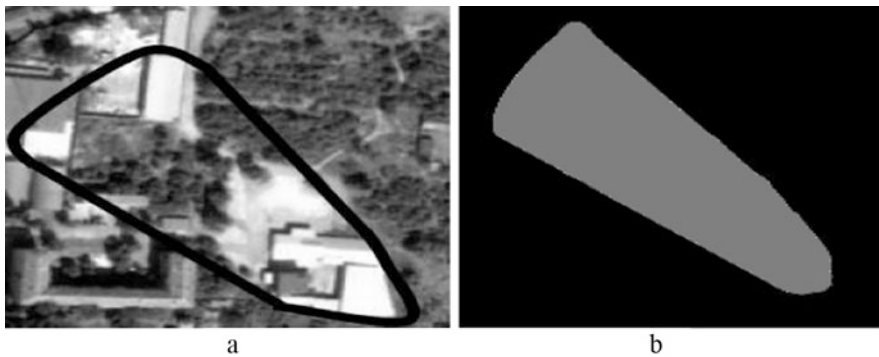


Fig. 16.5 Image of SI fragment: with contoured objects of OR (a); with average radiobrightness temperature within the contour (b)

Let us use average radiobrightness temperature for the radiometric channel as an informative attribute of the OR introduced in this way. The contoured object and its selective image are shown in Fig. 16.5.

Perform formation of CAF of the contoured object and SI at radiobrightness temperature $T_{Br}(i, j)$.

Calculate the maximum value of CCC, $K_{max}(i, j)$, corresponding to the selected images in accordance with the classical correlation algorithm for each (k, l) by the expressions [26]:

$$K_{i,j}(k, l) = \frac{1}{N_1 N_2} \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} S_{CO_{ij}}(m, n) \times S_{OI}(m + k - 1, n + l - 1), \tag{16.43}$$

where

$$\mathbf{K}_{ij} = \|K_{ij}(k, l)\| \quad \text{when } k = 1 \dots M_1 - N_1, \quad l = 1 \dots M_2 - N_2. \tag{16.44}$$

The maximum value of each resulting \mathbf{K}_{ij} matrix which is ensured at a complete coincidence of $\mathbf{S}_{CO_{ij}}$ and \mathbf{S}_{OI} is determined:

$$K_{\max}(i, j) = \max_{kl} \|K_{ij}(k, l)\|, \tag{16.45}$$

where $i = 1 \dots M_1 - N_1, j = 1 \dots M_2 - N_2$

$S_{CO_{i,j}}(m, n) \in \mathbf{S}_{PM}$ and $S_{OI}(m + k - 1, n + l - 1) \in \mathbf{S}_{OI}$.

The matrix with dimensions $(M_1 - N_1) \times (M_2 - N_2)$ which characterizes distribution of $K_{\max}(i, j)$ is CAF with brightness (CAF_{Br}).

The CCC resulting from comparison of the source image and the image fragment with an equivalent OR is shown in Fig. 16.6 [4].

Comparison of RI with the source image was carried out in accordance with the classical correlation algorithm. During simulation of SI, a radiometric image was taken. It was obtained by radiometric channel from altitude of 1000 m at sighting angle of 60° .

It has been established that in the case of the use of an equivalent OR, the DF is unimodal. At the same time, complete coincidence of RI fragment with SI ensures that there is no impact of scale and perspective distortions of the SS objects on the

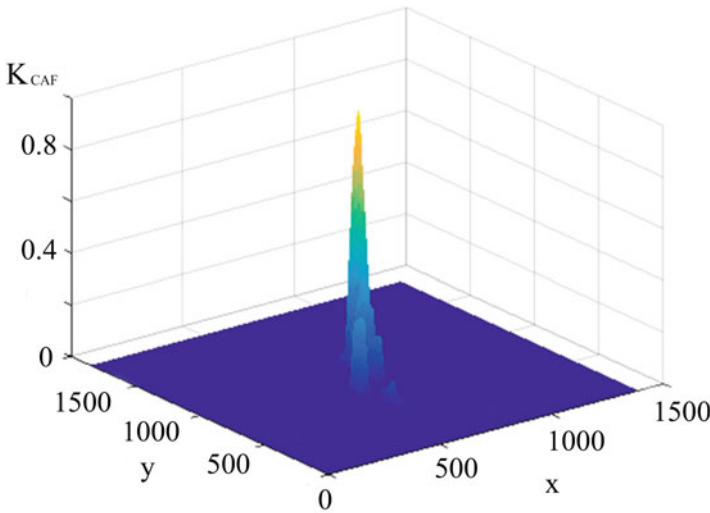


Fig. 16.6 CCC of the formed equivalent OR and SI

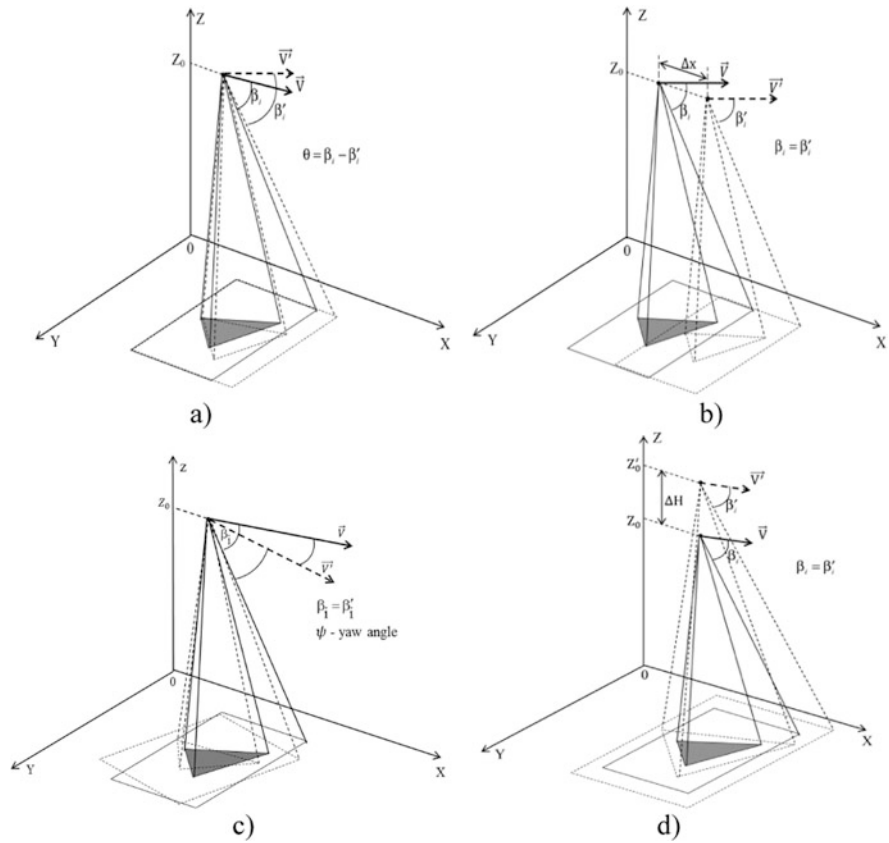


Fig. 16.7 Factors influencing accuracy of superposition: perspective distortion (a); CI bias relative to RI (b); turn of CI relative to RI (c); scale changes of CI (d)

comparison results. This is confirmed by the geometric construction of shift, scale change, perspective distortions, and SI turn with an equivalent OB relative to RIs shown in Fig. 16.7.

16.2.2 Formation of unimodal decision function of the radiometric CENS

Formation of a unimodal DF requires CI preprocessing in order to reduce the latter to a form close to RI.

According to radiometric CI, this means that its processing results in a necessity of determining a set of objects in the image with the highest value of radiobrightness

temperature $T_{Br}(i, j)_{max}$. On the basis of the set, an equivalent OB will be formed later as shown above and accordingly a new CI.

To do this, determine the average value of radiobrightness temperature of the background part of the image, $T_{Br\ av}$, and quantize radiobrightness temperatures of the objects in the current image taking into account radiometer sensitivity, ΔT , and dynamic range $T_{Br}(i, j)_{max} - T_{Br}(i, j)_{min}$. When quantizing, the values of the selected gray levels will be determined by the number of selected intervals, k .

The quantization of radiobrightening object temperature on the image (radiance temperature) has no influence on the resolution and operating speed of the radiometric channel. The quantization is determined by the radiometer's sensitivity. It allows to arrange the objects by radiobrightening temperature points and relatively to the background.

Let us quantize radiobrightness temperatures of CI by breaking the temperature range $T_{Br}(i, j)_{max} - T_{Br}(i, j)_{min}$ in even intervals, ΔT_{Br} :

$$\Delta T_{Br} = \frac{T_{Br}(i, j)_{max} - T_{B_{av}}}{k}, \quad (16.46)$$

where $k = 10-20$ is the number of quantization levels; $\Delta T_{Br} > \Delta T$.

In accordance with the defined maximum values of radiobrightness of the objects, form the current image of $S_{CI}(M_1, M_2)$ which will be considered the source image. Calculate the average value of radiobrightness temperature for a set of bright objects and represent geometrically related objects in the image in the form of an equivalent OB with a value of brightness averaged over its plane. It is the completion of preliminary processing of CI.

Next, transform CI of $S_{CI}(M_1, M_2)$ into a binary image, \mathbf{H} , according to the rule [20, 23]:

$$H_i = \begin{cases} 1, & S_i \in S_{max}; \\ 0, & S_i \leq S_{\rho}; \end{cases} \quad i \in \overline{1, F_0}, \quad (16.47)$$

where i is the number of CI fragment occupied by OB; ρ is the number of CI fragment occupied by background; F_0 is the size of the sample that forms two classes of ω_i that do not intersect each other and correspond to the signals of the object of reference, ω_u , and background, ω_{ρ} .

Solve the problem of OB selection in a binary image as follows. According to the selected threshold of values of radiobrightness temperature, compare fragments of the layered current image, $\mathbf{H}^i \subset \mathbf{H}$, with the reference image and find the fragment of CI which will have the greatest number of coincidences.

The decision rule that defines the DF consists of the following:

$$R_j = \sup_{i \in \overline{0, M}} R_i \tag{16.48}$$

As a result, a unimodal DF $R(x, y, t)$ of radiometric CENS will be formed.

The index i takes as many values of M as possible fragments are shifted one relative to the other with the set configuration in frame \mathbf{H} . If the rule (16.48) corresponds to several fragments, then decision on OR localization is not made.

Probability of correct OB localization will be determined as follows when using rule (16.48) [20, 27]:

$$P_{CL} = \sum_{j=1}^{F_u} C_{F_u}^j (1 - \alpha)^j \alpha^{F_u - j} \left[\sum_{m=0}^{j-1} C_{F_u}^m \beta^m (1 - \beta)^{F_u - m} \right]^M \tag{16.49}$$

where α, β are errors of the first and second kind which are determined by the value of the signal-to-noise ratio; F_u is the sample size corresponding to OB; $m \in \overline{0, M}$.

The results of estimation of probability of correct OB localization using an equivalent OB with brightness averaging in accordance with a set of geometrically related objects are shown in Fig. 16.8. The shown dependences were constructed for OB with area of 5–50% of the total image area [23].

In order to the verify effectiveness of the proposed approach, a statistical test of the DF formation algorithm of radiometric CENS with the use of the RI built using a set of geometrically related objects was performed. The algorithm of DF formation of radiometric CENS is shown in Fig. 16.9 [23].

The following assumptions were taken in simulation:

1. The SS scene is photographed in nadir.
2. Quantization of gradations of gray levels of OB and background is completed.
3. There are no mutual turns, geometric and scale distortions of CI and RI.

Fig. 16.8 Dependence of probability of correct OB localization on the signal-to-noise ratio for equivalent OB of various sizes

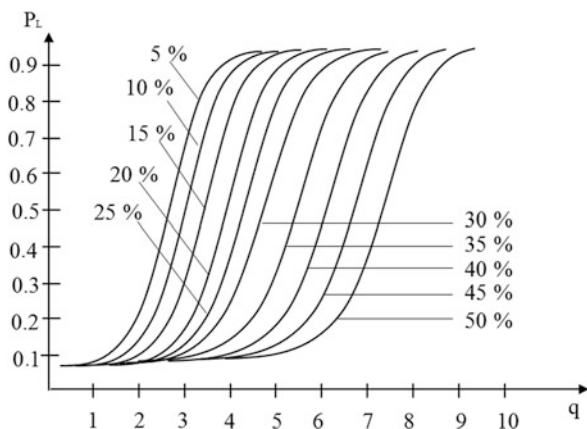
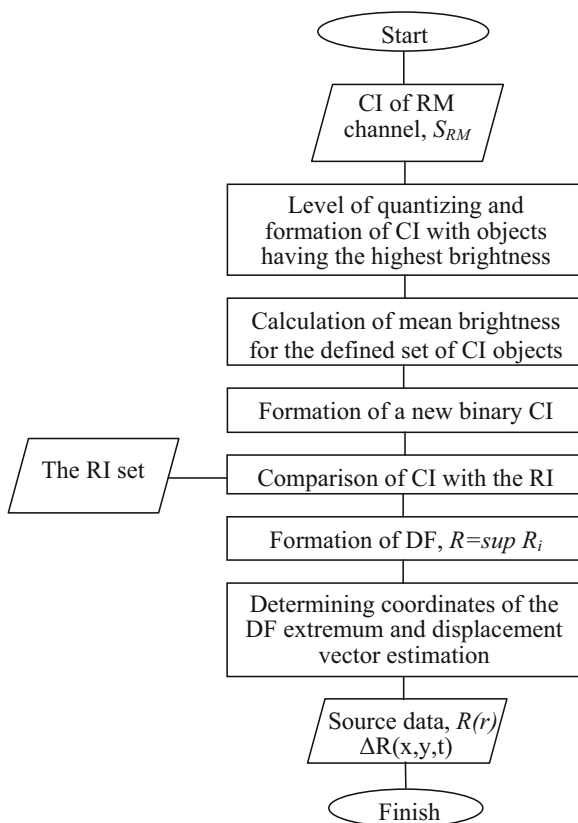


Fig. 16.9 Block diagram of the algorithm of formation of DF of radiometric (RM) CENS



4. The nodes of CI and RI matrices coincide.
5. OB is located in the center of RI.

Source data of the simulation algorithm:

Matrix of RI:

1. Size: 8×8 elements
2. Informational content: binary image, OB corresponds to “1” and background to “0”
3. OB size: 3×3 elements
4. OB form: conventional square

Matrix of CI:

1. Size: 16×16 elements
2. The number of levels of gray quantization: OB: 5 levels, background: 0–7 levels
3. OB size: 9 elements
4. OB form: inscribed in a conventional square with the size of 3×3

The simulation results are presented in Figs. 16.10 and 16.11 [23].

Fig. 16.10 CCC of CI and RI
 (a) ratio of signal/noise = 10;
 (b) ratio of signal/noise = 5

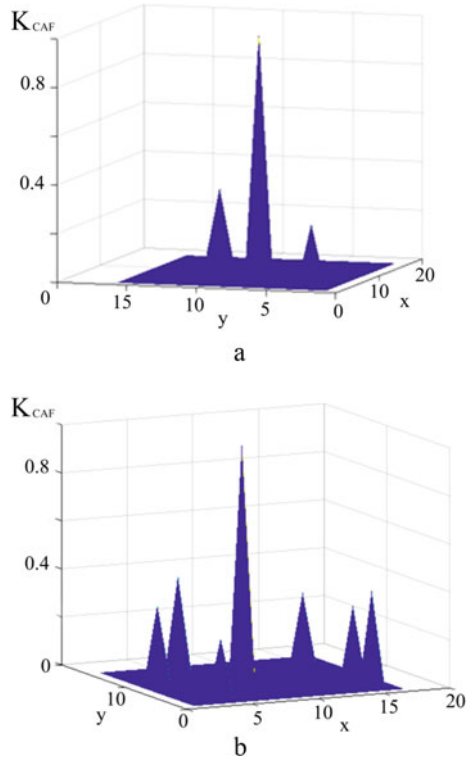
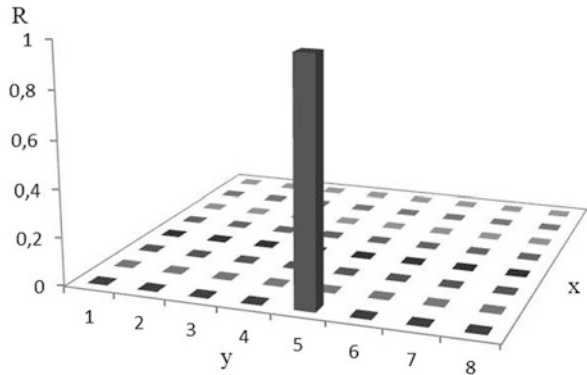


Fig. 16.11 The result of DF formation at signal/noise ratio of 5–10



Thus, it was established that the use of an auxiliary geometric attribute in the CI makes it possible to ensure probability of OB localization close to unity at a signal-to-noise ratio of 3–4. In this case, the area of OB must not exceed 30% of the entire CI area.

16.3 Features of the Formation of the CENS Decisive Function When Navigating Flying Robots in the Case of the Appearance of False Objects in the Current Image

When solving navigation tasks for flying robots equipped with CESN with reference to terrain sections with high object saturation, when other bright objects that are close to the binding object occur and are close in parameters (geometric dimensions and brightness), false objects may appear on CI depending on the sighting geometry. As a result, the efficiency of CENS may be insufficient, due to the complexity of the formation of a unimodal decisive function.

In many practical cases, other researchers are recommending the use of dynamic filtering in image processing. Solutions of dynamic filtering can be found in [28] where authors presented solutions of principal component analysis use for speeding up image filtering, in [29] where the authors used algorithm based on block coordinate descent, and article [30, 31] proposes an SD (for static/dynamic) filter.

However, all these methods are more proper for standardized/converted computer format images. We are still in search of continued idea of natural image on camera output processing.

16.3.1 Models of Current and Reference Images: Statement of the Task of Developing a Method for Localizing an Object Binding on Image

The CI Model \mathbf{S}_{CI} . To describe an SS, we will adopt a model in which the undistorted initial image \mathbf{S}_{II} is described by the brightness values of the corresponding objects and SS backgrounds in the resolution elements [20, 25]:

$$\mathbf{S}_{CI} = \mathbf{S}_{OI} = \|S(i, j)\|, \quad (16.50)$$

$$\text{where } S(i, j) = \begin{cases} S_v(i, j), & \text{at } S(i, j) \in \mathbf{S}_v \\ S_w(i, j), & \text{at } S(i, j) \in \mathbf{S}_w \end{cases}$$

$S_v(i, j)$ is the brightness of the image element of the v -th object of \mathbf{S}_v

$S_w(i, j)$ is the brightness of the image element of the w -th background of \mathbf{S}_w

V and W are, respectively, the numbers of objects and backgrounds of different brightness and shape in the SI

In accordance with (16.50), let us make the following assumptions for the CI model of the viewing surface:

- The current and initial images have the same size of $N_1 \times N_2$ pixels.
- The SS objects have significant brightness values relative to the background. The OB of the CENS has the greatest brightness.

- The OB and the background within the resolution element are uniforms in brightness.
- Each i,j -th element of the CI is a normally distributed value with the variance σ_{ij}^2 and the average luminance value $S(i, j)$. In the absence of interference, $S(i, j)$ can take one of the two values: $S_v(i, j)$ or $S_w(i, j)$. The contrast of the OB relative to the ambient background is defined as $\Delta S = S_v(i, j) - S_w(i, j)$.
- The dispersion of noise in the receiving channels of the CENS is the same, that is,

$$\sigma_{ij}^2 = \sigma^2, i \in \overline{1, N_1}, j \in \overline{1, N_2}$$

- For the number of the background elements belonging to the set of \mathbf{S}_w , and the objects belonging to the set of \mathbf{S}_v , the valid relation is $V \ll W$.

Taking into account the assumptions made, the density distributions of the brightness S of the background and object elements are determined by the expressions [20]:

$$w_w(S) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-(S - S_w)^2/2\sigma^2\right], \quad (16.51)$$

$$w_v(S) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-(S - S_v)^2/2\sigma^2\right]. \quad (16.52)$$

Concerning the signals of other objects of \mathbf{S}_ρ , close in brightness and commensurable with the reference object, hereinafter referred to as false objects (FOs), we make the following assumptions:

- The maximum size of \mathbf{S}_ρ does not exceed the diameter D_e of the resolution element on the terrain; otherwise, such an FO can be considered stable and as usable as a reference object.
- The equivalent diameters of \mathbf{S}_ρ are distributed according to an exponential law.

The latter assumption makes it possible to take into account only one distribution parameter—the average diameter $\mathbf{S}_\rho D_0$, in the formulation of the problem and also to assume that for a known mean value, the maximum entropy is revealed by the following exponential distribution [20]:

$$w(D) = \begin{cases} \frac{1}{D_0} \exp(-D/D_0), & D \leq D_e; \\ 0, & D > D_e. \end{cases} \quad (16.53)$$

The signal from S_ρ with the equivalent diameter D_e taking into account the fill factor of the image resolution element has a luminance value determined in accordance with the expression:

$$S(i, j) = S_\rho \frac{D}{D_e} + S_w(i, j) \left(1 - \frac{D}{D_e}\right) = S_w(i, j) - \frac{D}{D_e} (S_w(i, j) - S_\rho),$$

where S_ρ is the average brightness of an FO.

The density of the distribution of the probability of a signal from an FO with allowance for distribution (16.53) has the form:

$$\omega_\rho(S) = \begin{cases} \lambda e^{\lambda(S_w - S_\rho)} & S \leq S_\rho; \\ 0, & S > S_\rho, \end{cases} \quad (16.54)$$

where $\lambda = \frac{D_e}{D_0(S_w - S_\rho)}$.

Let us assume that the signals from an FO in the area of the CI frame are randomly distributed and represented by a Poisson flow, possessing the property of stationarity and ordinarity without any aftereffect.

Description of the RI. Because of the instability of both the absolute values of the brightness of the individual elements of the SS and the contrast of objects and the background, we will assume that the RI is given by the sign of the contrast and the geometric shape of the object. That is, we represent the RI as a binary image. The elements of the object have the value of 1, and the background elements have the value of 0.

Statement of the problem. To consider the CI model with several objects that are close in brightness and geometric form with the OB, it is necessary to solve the problem of localizing the reference object.

Let us denote by F_ρ the number of cells in the frame with signals from the F_0 , so [20]:

$$F_\rho + F_v + F_w = F_0, \quad (16.55)$$

where F_0 is the total number of frame elements that hit the camera's field of view F_v and F_w are the numbers of frame elements taken by the reference object and the background. Using the assumptions that $\sigma \ll \Delta S$, $F_v/F_0 > 0.5$ makes it possible to split the solution of the problem of localizing the OB in the CI into several stages. The first stage is to detect the object, whereas the second consists of its preliminary selection against the background of the FO. The third stage consists of finding the maximum DF value of R_j from the aggregate $R_j = \sum_{\zeta=1}^L R_i(i, j)$, which is determined by a layer-by-layer analysis of the number of DF cross sections ζ and the search for its single value corresponding to the maximum.

16.3.2 The Solution of the Detection Problem and Multi-Threshold Selection of the OB in a Current Image with Bright False Objects

The current image in the line-by-line expansion in accordance with (16.55) represents a vector of the dimension F_0 . As a result, we have a sample of the volume F_0 , which forms three disjoint classes of ω_i , corresponding to signals from the background ω_w , FO ω_ρ and OB ω_v , and the density of the sample distribution is given by the expression [20]:

$$w(S) = \sum_{i=1}^3 p_i w_i(S), \quad (16.56)$$

where $p_i = \frac{F_i}{F_0}$, $i = 1, 2, 3$ are a priori probabilities of the classes $w_i(S) = w(S|\omega_i)$ means the conditional probability densities of a random variable S under the condition that S belongs to a class of ω_i , defined by expressions (16.51), (16.52), (16.54)

In order to isolate the signals of the OB from the background signals, we split the sample consisting of the elements of the three classes into two classes with respect to the quality index, which is defined further.

Let us set quantization threshold l of the sample to two classes, according to which we assign the signals of the OB to one of them and the signals of the FO to the second class. In this case, the probabilities of errors of the first and second kinds are determined by the expressions [20]:

$$\alpha = \int_{S_w-l}^{\infty} w_v(S) dS, \quad (16.57)$$

$$\beta = \frac{1}{1+K} \int_{-\infty}^{S_w-l} [w_\rho(S) + K w_w(S)] dS, \quad (16.58)$$

where $K = \frac{F_\rho}{F_w}$.

By the probabilities α and β in the second stage, we can determine the probability of the correct localization of the OB, which, in accordance with (16.57) and (16.58), can be considered a function of threshold l and can be maximized by choosing the corresponding threshold $l = l_{\text{opt}}$. Since the distribution parameters of mixture (16.57) are unknown, the first step is to estimate the unknown parameters, which include ΔS , S_w , p_w . The parameter λ is uniquely determined by expression (16.56). The number of the OB elements F_v is known; hence, the probability is known as $p_v = \frac{F_0}{F_v}$, $p_\rho = 1 - p_w - p_v$.

By applying a nonlinear transformation to the original sample, we can construct a histogram of the distribution of the transformed random variable. The nonlinear transformation best emphasizes the “center” of the background distribution $w_w(S)$. By comparing the central parts of the histogram and the theoretical probability density, we determine the mean value of the background \widehat{S}_w .

During the transformations, the dynamic range of ΔS was divided into intervals of the σ length, and the center of the interval with the largest number of sample values was taken as a rough estimate of the mean value of S_w .

By performing the appropriate transformations and substituting the parameter estimates in expressions (16.57) and (16.58), we obtain estimates of the error probabilities $\widehat{\alpha}$, $\widehat{\beta}$ [20]:

$$\widehat{\alpha} = \int_{\widehat{S}_w-l}^{\infty} w_v \left(S \middle| \widehat{S}_w \right) dS, \tag{16.59}$$

$$\widehat{\beta} = \frac{1}{1 + \widehat{K}} \int_{-\infty}^{\widehat{S}_w-l} \left(w_w \left(S \middle| \widehat{S}_w \right) + \widehat{K} w_\rho \left(S \middle| \widehat{S}_w \right) \right) dS, \tag{16.60}$$

where $\widehat{K} = \widehat{p}_\rho / \widehat{p}_w$, $\widehat{p}_v = 1 - \widehat{p}_w - p_\rho$.

For the given threshold l , we transform the initial \mathbf{S}_{CI} of the CI into a binary image \mathbf{H} according to the rule (16.47).

The quantization threshold determines the probability of occurrence of errors of the first α and the second β kinds. In turn, the values of α and β determine the minimum value of the signal-to-noise parameter $q = q_{\min}$, at which the required probability of the correct localization of the OB is reached:

$$q_{\min} = \Phi^{-1} (1 - \alpha) + \Phi^{-1} (1 - \beta),$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt$ the probability integral.

Now it is necessary to solve the task of selecting an object in a binary CI against the background of the MO, using a priori information in the form of a binary RI.

The algorithm for processing the binary CI for the purpose of solving the problem of selecting the OB is as follows. For each fragment $\mathbf{H}^i \subset \mathbf{H}$, of the CI, having a certain configuration and size of the object, a comparison is made with the RI, which consists entirely of single units. The operation of comparing the binary images consists of adding, “according to module two,” image elements and forming the DF by using the formula [20]:

$$R_i = \sum_{k=1}^{F_\rho} \left(S_{RI_m} \oplus_{\text{mod } 2} H_m^i \right), \quad (16.61)$$

where H_m^i is the m -th element of the i -th fragment of the CI

S_{RI_m} is the m -th element of the reference image S_{RI}

The decision rule is that the fragment $\mathbf{H}^i \subset \mathbf{H}$, for which [20]:

$$R_j = \inf_i R_i, \quad (16.62)$$

is declared to coincide with the RI. The index i takes as many values of M , as all possible fragments are shifted with respect to each other with a given configuration in the frame \mathbf{H} . If the property of (16.62) is true for several fragments, then the decision to localize the OB is not accepted.

To compare \mathbf{H}^i with the RI consisting of units, it is more convenient to operate with numbers:

$$z_i = F_\rho - s_i, \quad i \in \overline{0, M},$$

each of them represents the number of units in the \mathbf{H}^i fragment. Then the decision rule is that the fragment $\mathbf{H}^i \subset \mathbf{H}$, for which [20]:

$$z_j = \sup_{i \in \overline{0, M}} z_i, \quad (16.63)$$

is declared to coincide with the RI.

To estimate the probability of the correct localization of an object, we proceed as follows. Let the size of the object be $T_1 \times T_2$ cells. We divide the CI matrix into rectangular $T_1 \times T_2$ sub-matrices.

If sub-matrix amount is not an integer, it will be necessary to increase the frame size, thus an integer number of sub-matrices (mark it $M + 1$) fit in it.

In this case, the estimates of the probability of the correct localization of the OB will be obtained as underestimated due to the increase in the frame. Suppose that the true position of the object falls into one of the sub-matrices, then we denote the fragment of the CI corresponding to the RI by $\mathbf{H}^0 \subset \mathbf{H}$ and we shall denote the fragments placed in the remaining sub-matrices by $\mathbf{H}^i, i \in \overline{1, M}$.

Let the probability of occurrence of 1 in \mathbf{H}^i be equal to r_i . Then the numbers of z_i are distributed according to the binomial law [19, 20]:

$$P(z_i) = C_{F_v}^{z_i} r_i^{z_i} (1 - r_i)^{F_v - z_i}, \quad z_i \in \overline{0, F_v}. \quad (16.64)$$

The probability of the correct localization of the OB by using decision rule (16.64) is equal to the probability that the number of units in z_0 , corresponding to the coincidence of the RI and the object, will exceed all other numbers of z_i , $i \in \overline{1, M}$.

We denote an event consisting of the appearance of $z_0 = j \in \overline{1, F_v}$ units in \mathbf{H}^0 by A_0^j , and the event consisting of the fact that the number of units in z_i does not exceed $j - 1$ by A_i^j , $i \in \overline{1, M}$. The events A_i^j , $i \in \overline{1, M}$ are independent in aggregate because $\mathbf{H}^i \cap \mathbf{H}^m = \emptyset \forall i, m \in \overline{0, M}$. In accordance with formula (16.64), the probabilities of the events A_i^j are determined by the expressions:

$$P(A_i^j) = \begin{cases} C_{F_v}^j r_0^j (1 - r_0)^{F_v - j}, & i = 0; \\ \sum_{m=1}^{j-1} C_{F_v}^m r_i^m (1 - r_i)^{F_v - m}, & i \in \overline{1, M}. \end{cases} \quad (16.65)$$

Then, by the probability multiplication theorem, the probability of the event $L_j = \bigcap_{i=0}^M A_i^j$ is equal to

$$P(L_j) = \prod_{i=0}^M P(A_i^j), \quad j \in \overline{1, F_v}.$$

Since the events in the aggregate are incompatible, the probability that the number of units in \mathbf{H}^0 exceeds the number of units in all other fragments of \mathbf{H}^i is determined by the expression for the probability of the correct localization [2]:

$$P_{CL} = \sum_{j=1}^{F_v} P(L_j) = \sum_{j=1}^{F_v} \prod_{i=0}^M P(A_i^j), \quad (16.66)$$

where the probabilities are determined by formula (16.66).

Considering that

$$r_i = \begin{cases} 1 - \alpha, & i = 0; \\ \beta, & i \in \overline{1, M}, \end{cases}$$

(the probabilities α and β are given by relations (16.57) and (16.58)), for the probability of the correct localization of the object, we obtain the final expression [2]:

$$P_{CL} = \sum_{j=1}^{F_v} C_{F_v}^j (1 - \alpha)^j \alpha^{F_v - j} \left[\sum_{m=0}^{j-1} C_{F_v}^m \beta^m (1 - \beta)^{F_v - m} \right]^M. \quad (16.67)$$

In order to ensure unambiguous decisions, it is necessary to develop an algorithm that performs an iterative process of processing with a varying threshold before obtaining a single solution.

One of the possible variants of such an algorithm is as follows. After calculating the average value estimate \widehat{S}_w , the initial value of the threshold $l^0 = \alpha\sigma$ is set (the algorithm tests showed that it is expedient to choose $\alpha \in 1.8 \dots 2.2$), with respect to which the CI:

$$S_{CI} = \|S(i, j)\|,$$

is transformed into a binary image, which we denote by H^0 . By comparing this image with the RI in accordance with criterion (16.63), the matrix of the decision function $\|z_{ij}^0\|$ is calculated, and the set is found as follows:

$$M^0 = \left\{ (k, l) \in \overline{1, N_1} \times \overline{1, N_2} \mid z_{kl} = \max_{i,j} z_{ij} \right\}$$

Moreover, the maximum of the decision function z_{\max}^0 is not necessarily equal to F_v , but it is possible that $z_{\max}^0 < F_v$. If the set M^0 consists of one element, that is, $M^0 = \{1(m, l)\}$, then the decision is made that the coordinates of the reference element of the object relative to the CI are m, l .

16.3.3 Solution to the Problem of Forming a Unimodal Decision Function

Expression (16.67) for the probability of the correct localization of the object is suitable for evaluating the effectiveness of applying the CENS in the areas of the SS with the reference object by a uniquely determined system. In this case, the system forms a unimodal DF. If reference is made to an SS with several objects that are comparable in parameters to the OB, it becomes necessary to refine the result of the reference. For this, in the third stage, the search is performed for the largest DF value, corresponding to the complete coincidence of the CI with the RI.

The essence of the method is to form a set of matrices G_i of the DF with the subsequent determination of the largest number of units in the summary representation of the DF as $\sum_{i=1}^U G_i$.

The decision rule is that the fragment $G_j \subset G$, for which [2]:

$$G_j = \sup_{i \in \overline{0, U}} G_i, \tag{16.68}$$

is declared to coincide with the RI.

The index i assumes as many values as there are cuts U of the fragments $\mathbf{G}_j \subset \mathbf{G}$ by the time of determining the cut with the greatest number of units.

As a criterion for localizing the RO, we choose an integral indicator of relative brightness, the values of which are formed as independent samples of Q in the elements of the DF matrices \mathbf{G}_i . All the resulting matrices \mathbf{G}_i are summed together element-wise. The resulting matrix \mathbf{G}_i consists of the elements $\mathbf{G}_j(i, j)$, the values of the independent samples in the form of integrated luminance indices. The matrix \mathbf{G}_i , with the largest number of units written as $(\sum_{i=1}^U G_i = \max)$ is taken as the result of localizing the required reference object.

The probability of the correct localization of the RO in accordance with the described algorithm is determined by the expression [2, 4]:

$$P_{CL} = 1 - \left(1 - \sum_{j=1}^{F_v} C_{F_v}^j (1 - \alpha)^j \alpha^{F_v-j} \left[\sum_{m=0}^{j-1} C_{F_v}^m \beta^m (1 - \beta)^{F_v-m} \right]^Q \right)^U \tag{16.69}$$

The results of estimating the probability of the correct localization and the DF formation for two values of the signal-to-noise ratio are shown in Figs. 16.12, 16.13, 16.14, and 16.15.

The analysis of the results of estimating the probability of the correct localization of the RO (Figs. 16.12 and 16.14) and the formation of the DF (Figs. 16.13 and 16.15) with the use of the SS with FOs has revealed that the application of the detection procedure and the multi-threshold selection of the RO in the image allows ensuring the probability of the correct localization of the object that is close to 1. At the same time, the presence of false objects in the image of the SS, comparable in parameters with the RO, does not affect the formation of the unimodal DF. Thus, the algorithm for implementing the developed method is distortion-proof, and it can be used in the CENS to ensure efficient functioning in the conditions of a complex background-object situation.

Fig. 16.12 The result of estimating the probability of localizing the RO in the CI with a signal-to-noise ratio $q \approx 10$

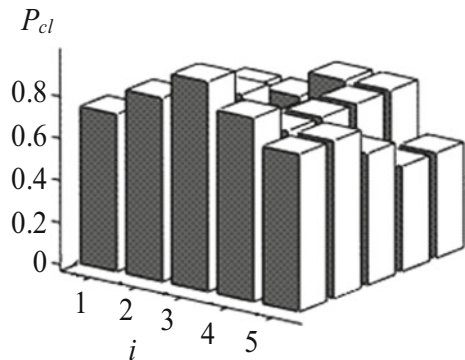


Fig. 16.13 The result of the DF formation with a signal-to-noise ratio $q \approx 10$

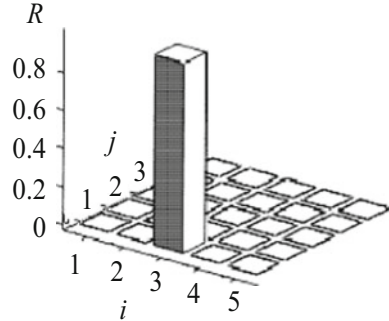


Fig. 16.14 The result of estimating the probability of localizing the RO in the CI with a signal-to-noise ratio $q \approx 20$

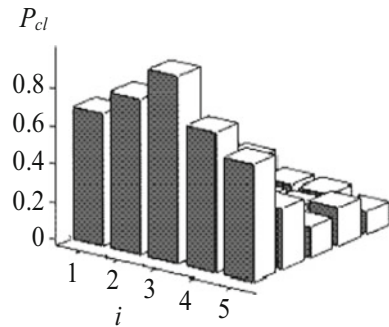
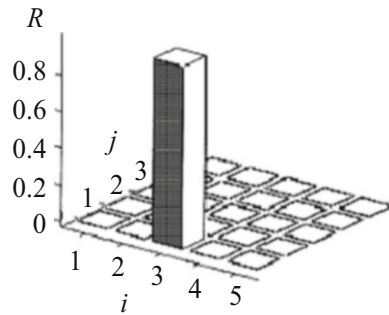


Fig. 16.15 The result of the DF formation with a signal-to-noise ratio $q \approx 20$



16.4 Conclusions

Assuming all points mentioned above, we could conclude that the offered approach of passive combined correlation-extreme systems implementing the survey-comparative method for recognition and analysis of images obtained from machine vision of a FR, significantly improves the correct localization of the objects in the image. This process has the potential to improve the quality of the control decisions during the navigation of the FR. It is important to note that this approach with only small corrections implemented through additional constraints of the bearing surface to the framework could target a variety of novel applications for terrestrial mobile

robots and platforms [5, 32–48]. The results from the performed simulations show that they can be compatible with recognized robot simulation frameworks [47–49] after slight modification and assimilation.

References

1. Leutenegger, S., et al. (2016). Flying robots. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics. Springer handbooks* (pp. 623–670). Cham: Springer.
2. Basset, P. M., Tremolet, A., & Lefebvre, T. (2014). Rotary wing UAV pre-sizing: Past and present methodological approaches at onera. *AerospaceLab*, 1–12.
3. Taha, H., Kiani, M., & Navarro, J. (2018). Experimental demonstration of the vibrational stabilization phenomenon in bio-inspired flying robots. *IEEE Robotics and Automation Letters*, 3(2), 643–647.
4. Beard, R. W., Ferrin, J., & Humpherys, J. (2014). Fixed wing UAV path following in wind with input constraints. *IEEE Transactions on Control Systems Technology*, 22(6), 2103–2117.
5. Lindner, L., Sergiyenko, O., Rodríguez-Quiriones, J., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Murrieta-Rico, F. N., & Tyrsa, V. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot*, 43(4), 360–369.
6. Lindner, L., Sergiyenko, O., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quiriones, J., Murrieta-Rico, F., Ivanov, M., Tyrsa, V., & Basaca, L. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 44(4), 532–541.
7. Scaramuzza, D., et al. (2014). Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics & Automation Magazine*, 21(3), 26–40.
8. Dunbabin, M., & Marques, L. (2012). Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1), 24–39.
9. Song, G., Yin, K., Zhou, Y., & Cheng, X. (2009). A surveillance robot with hopping capabilities for home security. *IEEE Transactions on Consumer Electronics*, 55(4), 2034–2039.
10. Finn, R. L., & Wright, D. (2012). Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications. *Computer Law & Security Review*, 28(2), 184–194.
11. Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206.
12. Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: A review. *Applied Geomatics*, 6(1), 1–15.
13. Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., & Scaramuzza, D. (2016). Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, 33(4), 431–450.
14. Sotnikov, A. M., & Tarshin, V. A. (2013). Problemy i perspektivy razvitiya navigatsionnoy podderzhki vozдушnykh sudov. *Zbirnyk naukovykh prats' Kharkivs'koho universytetu Povitryanykh Syl*, 3(36), 57–63. [In Russian].
15. Sotnikov, O. M., Tarshin, V. A., & Otkryto, P. V. (2013). Problemy i pryamoye razvitiye yadrospetsificheskikh ekstremal'nykh sistem, nalozhennykh kerovanami apparatury. *Sovremennyye informatsionnyye tekhnologii v sfere oborony bez oborony*, 3(18), 93–96. [In Russian].
16. Keller, J., Thakur, D., Likhachev, M., Gallier, J., & Kumar, V. (2017). Coordinated path planning for fixed-wing UAS conducting persistent surveillance missions. *IEEE Transactions on Automation Science and Engineering*, 14(1), 17–24.

17. Li, H., & Savkin, A. V. (2018). Wireless sensor network based navigation of micro flying robots in the industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8), 3524–3533.
18. Somov, Y., Butyrin, S., Somov, S., & Somova, T. (2017). In-flight calibration, alignment and verification of an astroinertial attitude determination system for free-flying robots and land-survey satellites. In *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), Padua*, pp. 474–478.
19. Sotnikov, A. M., Antyufeyev, V. I., Bykov, V. N., Grichanyuk, A. M. et al. (2014). Matrichnyye radiometricheskiye korrelyatsionno-ekstremal'nyye sistemy navigatsii letatel'nykh apparatov. Book: Ministerstvo obrazovaniya i nauki Ukrainy. KHNU imeni V.N. Karazina. 372p. [In Russian].
20. Sotnikov, A., Tarshyn, V., Yeromina, N., Petrov, S., & Antonenko, N. (2017). A method for localizing a reference object in a current image with several bright objects. *Eastern-European Journal of Enterprise Technologies*, 3(87), 68–74.
21. Sotnikov, A. M., & Tarshin, V. A. (2012). Obosnovaniye printsipov postroyeniya i razvitiya modeli korrelyatsionno-ekstremal'noy sistemy navedeniya kombinirovannogo tipa. *Sistema Upravleniya Navigatsiyey ta Zvozdku. K.*, 4(24), 7–11. [In Russian].
22. Sotnikov, A. M., Tantsiura, A. B., & Lavrov, O. Y. (2018). Calculating method of error calculations of the object coordinating platform free inertial navigation systems of unmanned aerial vehicle. *Advanced Information Systems*, 2(1), 32–41.
23. Sotnikov, O. M., Vorobey, O. M., & Tantsyura, O. B. (2018). Modeli potochnykh zobrazhen', shcho formuyut'sya kanalami kombinovanoi korrelyatsiyno-yekstremal'noii sistemi navigatsiï bezpilotnogo lital'nogo aparatu. *Suchasni informatsiyni Tekhnologii u Sferi Bezpeki ta Oboroni*, 2(32), 29–38. [In Ukrainian].
24. Tarshin, V. A., Sotnikov, A. M., Sidorenko, R. G., & Mezentsev, A. V. (2015). Metodologiya otsenki informativnosti iskhodnykh izobrazheniy dlya vysokotochnykh korrelyatsionno-ekstremal'nykh navigatsionnykh system. *Sistemy obrabotki informatsii*, 10, 60–63. [In Russian].
25. Tarshin, V. A., Sotnikov, A. M., Sidorenko, R. G., & Megel'bey, V. V. (2015). Podgotovka etalonnykh izobrazheniy dlya vysokotochnykh korrelyatsionno-ekstremal'nykh sistem navigatsii na osnove formirovaniya polya fraktal'nykh razmernostey. *Sistemi ozbroënnyia i viys'kova tekhnika*, 2, 142–144. [In Russian].
26. Tarshin, V. A., Sotnikov, A. M., & Sidorenko, R. G. (2015). Podgotovka etalonnykh izobrazheniy dlya vysokotochnykh korrelyatsionno-ekstremal'nykh sistem navigatsii na osnove ispol'zovaniya pryamogo korrelyatsionnogo analiza. *Nauka i tekhnika povitryanikh sil zbroynikh sil ukraïni*, 2, 69–73. [In Russian].
27. Smelyakov, K. S., Ruban, I. V., Smelyakov, S. V., & Tymochko, O. I. (2005). Segmentation of small-sized irregular images. In *Proceeding of IEEE East-West Design & Test Workshop (EWDTW'05), Odessa*, pp. 235–241.
28. Hernandez, W., & Mendez, A. (2018). Application of principal component analysis to image compression. In *Statistics-Growing Data Sets and Growing Demand for Statistics*. IntechOpen.
29. Shen, Z., & Song, E. (2018). Dynamic filtering of sparse signals via L1 minimization with variant parameters. In *2018 37th Chinese Control Conference (CCC), Wuhan*, pp. 4409–4414.
30. Ham, B., Cho, M., & Ponce, J. (2018). Robust guided image filtering using nonconvex potentials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1), 192–207.
31. Tymochko, O. I., & Podorozhnyak, A. O. (2007). Lokalizatsiya ob'ektu poshuku na potochnomu zobrazheni navigatsiynoy systemy z raiometrychnymy datchykamy. *Zbirnyk naukovykh prats Kharkivskoho universytetu Povitryanykh Syl*, 1(13), 47–50. [In Ukrainian].
32. Sergiyenko, O., Hernandez, W., Tyrsa, V., Devia Cruz, L., Starostenko, O., & Pena-Cabrera, M. (2009). Remote sensor for spatial measurements by using optical scanning. *Sensors*, 9(7), 5477–5492.
33. Sergiyenko, O. Y. (2010). Optoelectronic system for mobile robot navigation. *Optoelectronics, Instrumentation and Data Processing*, 46(5), 414–428.
34. Garcia-Cruz, X. M., Sergiyenko, O. Y., Tyrsa, V., Rivas-Lopez, M., Hernandez-Balbuena, D., Rodriguez-Quiñonez, J. C., Basaca-Preciado, L. C., & Mercorelli, P. (2014). Optimization of

- 3D laser scanning speed by use of combined variable step. *Optics and Lasers in Engineering*, 54, 141–151.
35. Básaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., García, X., Tyrsa, V., Rivas-Lopez, M., Hernandez-Balbuena, D., Mercorelli, P., Podrygalo, M., Gurko, A., Tabakova, I., & Starostenko, O. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
 36. Cañas, N., Hernandez, W., González, G., & Sergiyenko, O. (2014). Controladores multivariables para un vehículo autónomo terrestre: Comparación basada en la fiabilidad de software. *RIAI-Revista Iberoamericana de Automática e Informática Industrial*, 11(2), 179–190. [In Spanish].
 37. Straßberger, D., Mercorelli, P., Sergiyenko, O., & Decoupled, A. (2015). MPC for motion control in robotino using a geometric approach. *Journal of Physics: Conference Series*, 659, 1–10.
 38. Sergiyenko, O. Y., Ivanov, M. V., Tyrsa, V. V., Kartashov, V. M., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quinonez, J. C., Nieto- Hipólito, J. I., Hernandez, W., & Tchernykh, A. (2016). Data transferring model determination in robotic group. *Robotics and Autonomous Systems*, 83, 251–260.
 39. Rodríguez-Quinonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
 40. Sergiyenko, O., Tyrsa, V., Flores-Fuentes, W., Rodríguez-Quinonez, J., & Mercorelli, P. (2018). Machine vision sensors. *Journal of Sensors*, 2018, 3202761.
 41. Sergiyenko, O., Flores-Fuentes, W., & Tyrsa, V. (2017). *Methods to improve resolution of 3D laser scanning* (p. 132). LAP LAMBERT Academic Publishing.
 42. Sergiyenko, O., & Rodríguez-Quinonez, J. C. (Eds.). (2016). *Developing and applying optoelectronics in machine vision* (p. 341). Hershey, PA: IGI Global.
 43. Ivanov, M., Lindner, L., Sergiyenko, O., Rodríguez-Quinonez, J. C., Flores-Fuentes, W., & Rivas-López, M. (2019). Mobile robot path planning using continuous laser scanning. In *Optoelectronics in machine vision-based theories and applications* (pp. 338–372). IGI Global. <https://doi.org/10.4018/978-1-5225-5751-7.ch012>.
 44. Sergiyenko, O. Y., Tyrsa, V. V., Devia, L. F., Hernandez, W., Starostenko, O., & Rivas Lopez, M. (2009). Dynamic laser scanning method for mobile robot navigation. In *Proceedings of ICCAS-SICE 2009, ICROS-SICE International Joint Conference, Fukuoka, Japan*, 18–21 August 2009, pp. 4884–4889.
 45. Rivas, M., Sergiyenko, O., Aguirre, M., Devia, L., Tyrsa, V., & Rendón, I. Spatial data acquisition by laser scanning for robot or SHM task. In *IEEE-IES Proceedings "International Symposium on Industrial Electronics" (ISIE-2008)*, Cambridge, UK, 30 of June–2 of July, 2008, pp. 1458–1463.
 46. Básaca, L. C., Rodríguez, J. C., Sergiyenko, O., Tyrsa, V. V., Hernández, W., Nieto Hipólito, J. I., & Starostenko, O. (2010). Resolution improvement of dynamic triangulation method for 3D vision system in robot navigation task. In *Proceedings of IEEE-36th Annual Conference of IEEE Industrial Electronics (IECON-2010)*, Glendale-Phoenix, AZ, 7–10 November 2010, pp. 2880–2885.
 47. Rohmer, E., Singh, S. P., & Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 1321–1326.
 48. Michel, O. (2004). Cyberbotics Ltd—WebotsTM: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1), 5, pp. 39–42.
 49. Furrer, F., Burri, M., Achtelik, M., & Siegwart, R. (2016). RotorS—A modular gazebo MAV simulator framework. *Robot Operating System (ROS)*, 595–625.

Chapter 17

Stabilization of Airborne Video Using Sensor Exterior Orientation with Analytical Homography Modeling



Hadi Aliakbarpour, Kannappan Palaniappan, and Guna Seetharaman

Acronyms

GPS	The Global Positioning System
IMU	Inertial measurement unit
RANSAC	RANdom SAMple Consensus
VIRAT	DARPA Video and Image Retrieval and Analysis Tool
WAMI	Wide Area Motion Imagery
WAAS	wide area aerial surveillance
WAPS	wide-area persistent surveillance
WFOV	wide field-of-view
UAV	Unmanned Aerial Vehicles
GSD	Ground Sampling Distance
SfM	Structure-from-Motion
BA	Bundle Adjustment
NEU	North-East-Up

17.1 Introduction

Wide area motion imagery (WAMI), also known as, wide area aerial surveillance (WAAS), wide-area persistent surveillance (WAPS), or wide field-of-view (WFOV) imaging is an evolving imaging capability that enables persistent coverage of large

H. Aliakbarpour (✉) · K. Palaniappan
Computational Imaging and VisAnalysis (CIVA) Lab, EECS, University of Missouri,
Columbia, MO, USA
e-mail: akbarpour@missouri.edu

G. Seetharaman
Advanced Computing Concepts, U.S. Naval Research Laboratory, Washington, DC, USA

geographical regions on the order of a few to tens of square miles [1] at tens of centimeter resolution, or very small areas such as bridges and construction projects at very high resolution from closer range using the same sensor package. It has become even more popular due to performance advances in sensor technologies, computing hardware, battery performance, and reduction in size, weight, and cost of these components. WAMI sensors can be placed on many types of airborne platforms including fixed wing or multi-rotor unmanned aerial vehicles (UAVs)—both fixed wing and multi-rotor drones, small (manned) aircraft and helicopters [2]. Depending on the imaging sensor characteristics and aircraft altitude, these systems can cover a small city-sized area with an approximate ground sampling distance (GSD) of 10–30 cm per pixel, tens to hundreds of megapixels at the focal plane using single or multiple optical systems (e.g. 6600 × 4400 RGB color) with a frame rate of 1–10 Hz.

Detection of small and distant moving objects, e.g. cars or pedestrians, in a scene which is observed by a camera that by itself undergoes motions and jitters is extremely challenging. This can be even more challenging considering that small objects like cars may appear as 10–25 pixels in their length. To improve detection and tracking in aerial imagery [3–5] in which videos are captured on a moving platform, the images are stabilized (registered) to maintain the relative movement between the moving platform and the scene fixed. An accurate image stabilization in such scenarios can be important for both higher level video analytics and visualization. Traditionally, aerial image registration methods are performed through applying 2D homography transformations in the *image space* [6–8]. Aerial image registration is challenging for urban scenes where there are large 3D structures (tall buildings) causing high amount of occlusion and parallax. In such situations, the presence of parallax can lead to significant error when inter-image 2D registration approaches are used [9].

In this paper, a method to register aerial images is proposed which utilizes available (noisy or approximate) GPS and IMU measurements from the airborne platform and robustly stabilizes images by optimizing camera 3D poses using a homography-based cost function. Unlike most existing methods, the homography transformations in our approach are not estimated using any image-to-image estimation techniques, but directly derived as a closed-form analytic expression from the 3D camera poses. In our previous work, we leveraged our fast structure-from-motion (SfM) technique (BA4S [10, 11]) to derive a novel georegistration approach that did not need to estimate local patch-based homographies and used an analytical model that was both accurate and fast [12]. Although that approach was fast and globally accurate, its cost function is defined over the full 3D space in order to optimize the *retinal plane reprojection pixel error* over the full 3D scene as required by most SfM downstream applications (e.g. dense 3D reconstruction [13–16]). However, as an alternative to full SfM-based georegistration, we propose to stabilize an image sequence or remove jitter, with the objective of deriving a smooth motion trajectory over the sequence of images such that the dominant ground plane is stabilized minimizing a 2D metric distance-based error function. Therefore, in this paper, we propose an alternative approach for the parameter optimization with an emphasize on stabilizing the geoprojected aerial imagery by defining a cost function over a *single dominant 2D Euclidean world plane*. The points that do not lie on

the dominant are automatically marginalized during the optimization process. In the experiments, we will show that the method proposed in this paper is more robust in situations where the available camera sensor pose measurements are extremely inaccurate.

17.1.1 Related Work

The majority of approaches for image stabilization use pairwise and combinatorial matching and warping transformation for stabilizing the ground plane prior to moving object detection [6–8, 17–26]. Aerial image registration is challenging for urban scenes where there are large 3D structure and tall buildings causing high amount of occlusion and parallax [9, 12, 27]. An aerial image registration method was proposed in [6, 28] which uses a multi-layer (coarse to fine) homography estimation approach to deal with parallax and occlusions. Although using a hierarchical homography estimation helped to reduce false registration, their approach still suffer from the presence of strong parallax, as it was not able to seamlessly register all images within a dataset altogether. By observing Table-I in their paper [6], one can see that each dataset was broken into several segments in the registration process, due to an inability to faithfully handle strong parallax. Molina and Zhu [17] proposed a method to register nadir aerial images in which a pyramid block-based correlation method was used to estimate inter-frame affine parameters. They stated [17] not being able to use available GPS/INS measurements [1, 29–31] and just relying on the imagery itself: *the measures made by GPS/INS devices come with errors due to hardware, and [if] used directly will produce panoramas with large apparent errors and discontinuities*. Their approach requires persistent (multiple) cyclic video data collections to work. Moreover their approach has been only tested on nadir imagery with negligible parallax issues, while in oblique imagery (WAMI) the parallax is significantly stronger.

Direct georeferencing of high resolution unmanned aerial vehicles (UAV) imagery was discussed in [32] while performances of different SfM softwares (Photoscan [33], Pix4D [34] and Bundler [35, 36]) were evaluated. Pritt [37] proposed a fast orthorectification method for registration of thousands of aerial images (acquired from small UAVs). The author argued that BA is not able to handle hundreds of aerial images and therefore it is not scalable. Notice that the results presented in [37] appear to be tested over relatively flat terrain with negligible parallax. In [38], IMU was used to register laser range measurements to the images captured from a stereo camera. Crispell et al. introduced an image registration technique to deal with parallax, assuming to have a dense 3D reconstructed model of the scene [9]. In [39], GPS and IMU were used to perform an initial (coarse) orthorectification and georeferencing of each image in an aerial video. Then a RANSAC-based method was used to find optimal affine transformations in 2D image space. A method for registering and mosaicking multi-camera images was proposed in [7]. In the proposed method, registration is achieved using control points and projective image-to-image

transformations (using a variation of RANSAC). Among the corresponding control points found in overlapping images, those that best satisfy the projective constraint are used to register the images. A technique for optimal feature matching was recently proposed in [40], called locally linear transforming (LLT), which tries to tackle outliers using the consistency of local features within a neighborhood. A local geometrical constraint was developed that can preserve local structures among neighboring feature points which is also robust to a large number of outliers. It has a relatively high complexity and also uses exhaustive iterative methods; two drawbacks in the feature matching stages of a SfM pipeline which we are avoiding in our proposed approach. A similar algorithm, called restricted spatial order constraints (RSOC), was proposed in [41] to deal with outliers for registering aerial images. Both local structure and global information were considered in RSOC. It assumes that neighbor spatial order is preserved after rigid and affine transformation and based on that an affine invariant descriptor was defined. However, such assumption for oblique aerial imagery of urban scenes is not held, due to existence of high parallax. Recently, some image-based methods for robust registration (mosaicking) of long aerial video sequences have been introduced in [42–44].

17.2 Feature Track Building

In persistent aerial imagery, images are sequentially acquired meaning that one knows that what frame is adjacent to which one. By leveraging the temporal consistency of the images and using them as a prior information, the time complexity of matching can be reduced to $O(n)$. Interest points are extracted from each image using a proper feature extraction method. Starting from the first frame, for each two successive image frames, the descriptors of their interest points are compared. While successively matching them along the sequence, a set of feature *tracks* are generated. A track basically indicates that a potentially unique 3D point in the scene has been observed in a set of image frames.

17.3 Imaging Model

Figure 17.1 shows a world coordinate system W and a dominant ground plane π spanning through its X and Y axes. The scene is observed by n airborne cameras $C_1, C_2 \dots C_n$. To make the notations succinct, we will omit the camera indices from now on unless otherwise stated. The image homogeneous coordinate of a 3D point $\mathbf{X} = [x \ y \ z]^T$ from the world reference system W projected on the image plane of camera C is obtained as

$$\tilde{\mathbf{x}} = \mathbf{K}(\mathbf{R}\mathbf{X} + \mathbf{t}) \quad (17.1)$$

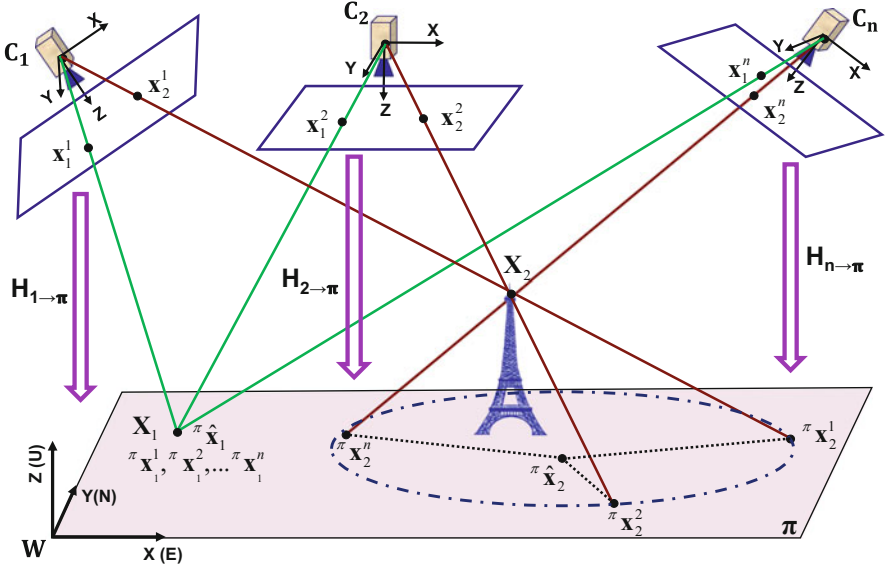


Fig. 17.1 A scene and its dominant ground plane π are observed by n airborne cameras. For an on-the-plane 3D point such as X_1 , its homographic transformations from the image plane of every single camera onto π , all merge together and converge to the same identical 3D point X_1 , whereas for an off-the-plane 3D point such as X_2 , its homographic transformations are spread out (diverged)

where \mathbf{K} is the calibration matrix (intrinsic), \mathbf{R} and \mathbf{t} are, respectively, the rotation matrix and translation vector from W to C . For a 3D point \mathbf{X} lying on π , its Z component is zero:

$$\tilde{\mathbf{x}} = \mathbf{K} \left([\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3] \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \mathbf{t} \right) \tag{17.2}$$

\mathbf{r}_1 , \mathbf{r}_2 , and \mathbf{r}_3 being the first, second, and third columns of \mathbf{R} , respectively. After simplification, we have

$$\tilde{\mathbf{x}} = \mathbf{K} [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \pi \tilde{\mathbf{x}} \tag{17.3}$$

where $\pi \tilde{\mathbf{x}} = [x \ y \ 1]^T$ represent the 2D homogeneous coordinates of the 3D point \mathbf{X} on π . One can consider the term $\mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$ as a 3×3 homography transformation matrix which maps any 2D point from π onto the camera image plane as:

$$\tilde{\mathbf{x}} = \mathbf{H}_{\pi \rightarrow c} \pi \tilde{\mathbf{x}}. \tag{17.4}$$

Likewise, a homogeneous image point $\tilde{\mathbf{x}}$ can be mapped on π as:

$${}^\pi \tilde{\mathbf{x}} = \mathbf{H}_{c \rightarrow \pi} \tilde{\mathbf{x}} \tag{17.5}$$

where $\mathbf{H}_{c \rightarrow \pi}$ is the inverse of $\mathbf{H}_{\pi \rightarrow c}$ and is equal to:

$$\mathbf{H}_{c \rightarrow \pi} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]^{-1} \mathbf{K}^{-1}. \tag{17.6}$$

Assuming $\mathbf{T} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$, f as the focal length in pixel, and (u, v) as the camera image principal point, (17.6) can be expressed as:

$$\mathbf{H}_{c \rightarrow \pi} = \mathbf{T}^{-1} \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix}^{-1} \tag{17.7}$$

$$\mathbf{H}_{c \rightarrow \pi} = \frac{1}{\lambda} \begin{bmatrix} m_{11} & -m_{21} & [-m_{11} & m_{21} & m_{31}] \mathbf{v} \\ -m_{12} & m_{22} & [m_{12} & -m_{22} & -m_{32}] \mathbf{v} \\ r_{13} & r_{23} & -\mathbf{r}_3^T \mathbf{v} \end{bmatrix} \tag{17.8}$$

where $\mathbf{v} = [u \ v \ f]^T$ and λ is a scalar defined as

$$\lambda = f \mathbf{r}_3^T \mathbf{t}, \tag{17.9}$$

and m_{ij} is the *minor*(i, j) of matrix \mathbf{T} . One can omit λ in (17.8) as a homography matrix is defined up-to-scale, yielding:

$$\mathbf{H}_{c \rightarrow \pi} = \begin{bmatrix} m_{11} & -m_{21} & [-m_{11} & m_{21} & m_{31}] \mathbf{v} \\ -m_{12} & m_{22} & [m_{12} & -m_{22} & -m_{32}] \mathbf{v} \\ r_{13} & r_{23} & -\mathbf{r}_3^T \mathbf{v} \end{bmatrix} \tag{17.10}$$

17.4 Optimization

Suppose our global reference system W in Fig. 17.1 is aligned with NEU (North-East-Up). Reminding that π is the dominant ground plane in the scene, and there are n cameras (or one camera in n different poses) observing the scene. The pose of each camera C_i is defined by a rotation matrix \mathbf{R}_i and \mathbf{t}_i which are defined from the global coordinate system to the camera local coordinate system. Also suppose to have m feature tracks in the scene. A feature track is basically a sequence of feature points which are matched across the sequence of image frames. All features within a track are the observations corresponding to a hypothetically identical 3D point in the scene. The homogeneous image coordinates of a 3D point \mathbf{X}_j on the image plane of camera C_i are expressed as $\tilde{\mathbf{x}}_j^i$, and it can be mapped from image plane to the Euclidean plane π as

$${}^{\pi}\tilde{\mathbf{x}}_j^i = \mathbf{H}_{i \rightarrow \pi} \tilde{\mathbf{x}}_j^i. \quad (17.11)$$

Ideally, if 3D point \mathbf{X}_j lies on the plane π , then mapping of all its corresponding image observations ($\tilde{\mathbf{x}}_j^1, \tilde{\mathbf{x}}_j^2, \dots, \tilde{\mathbf{x}}_j^n$) onto the plane, using (17.11), have to merge to an identical 2D point on π , which also coincides on the 3D point \mathbf{X}_j itself (see Fig. 17.1):

$${}^{\pi}\tilde{\mathbf{x}}_j^1 = {}^{\pi}\tilde{\mathbf{x}}_j^2 = \dots = {}^{\pi}\tilde{\mathbf{x}}_j^n \simeq \mathbf{X}_j \quad (17.12)$$

However, it is not the case in real scenarios due to different source of errors such as inaccuracy in the measured camera poses (e.g. from GPS/IMU). Therefore, the set of mapped 2D points, $\left\{{}^{\pi}\tilde{\mathbf{x}}_j^i \mid i = 1 \dots n\right\}$, corresponding to 3D point \mathbf{X}_j , will be dispersed around the actual point ${}^{\pi}\tilde{\mathbf{x}}_j$ on π . One can consider, ${}^{\pi}\hat{\mathbf{x}}_j$, the centroid of the distribution of 2D projected points, as an estimate for the actual point:

$${}^{\pi}\hat{\mathbf{x}}_j = \frac{1}{n} \sum_{i=1}^n {}^{\pi}\tilde{\mathbf{x}}_j^i \quad (17.13)$$

The Euclidean distance between each mapped point ${}^{\pi}\tilde{\mathbf{x}}_j^i$ and the estimated centroid is considered as an error metric:

$$e_j = \sum_{i=1}^n \|\mathcal{F}(\mathbf{H}_{i \rightarrow \pi} \tilde{\mathbf{x}}_j^i) - \mathcal{F}({}^{\pi}\hat{\mathbf{x}}_j)\|^2 \quad (17.14)$$

Overall error for all points and cameras can be used as a cost function to optimize \mathbf{R}_i , \mathbf{t}_i , and ${}^{\pi}\hat{\mathbf{x}}_j$ (see Fig. 17.2):

$$E = \min_{\mathbf{R}_i, \mathbf{t}_i, {}^{\pi}\hat{\mathbf{x}}_j} \sum_{i=1}^n \sum_{j=1}^m \|\mathcal{F}(\mathbf{H}_{i \rightarrow \pi} \tilde{\mathbf{x}}_j^i) - \mathcal{F}({}^{\pi}\hat{\mathbf{x}}_j)\|^2 \quad (17.15)$$

where $\mathcal{F}(\cdot)$ designates a function that returns the Euclidean coordinates from 2D homogeneous ones. Such a minimization can be done through using various iterative optimization techniques among which Levenberg–Marquardt methods are well known and popular in the literature [45]. Here, total number of parameters to be optimized is $6n + 2m$, where n is the number of views and m is number of tracks. Basically, each view i has six parameters including three for the rotation and three for the translation components. Likewise, each track j is represented by the 2D mean position vector, ${}^{\pi}\hat{\mathbf{x}}_j$, as expressed by (17.13). Total number of parameters in the observation space is $\leq 2 \times n \times m$. Note that the length of each track is $\leq n$.

The introduced mathematical model for image registration is held only if all 3D points to be imaged lie on the reference ground plane π (assuming to have perfect features correspondences). However, in WAMI and particularly in urban scenarios,

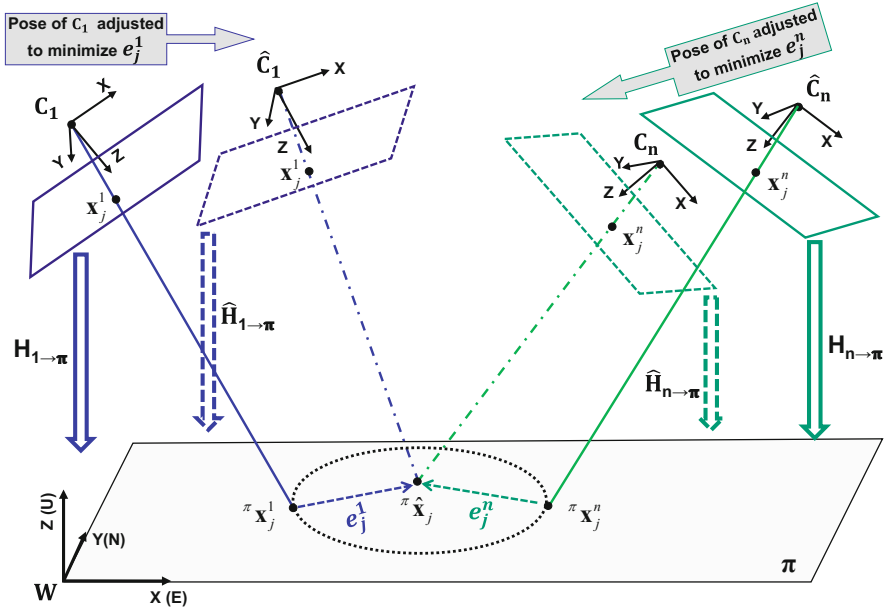


Fig. 17.2 The optimization scheme: two matched image points in a features track j , x_j^1 from C_1 and x_j^n from C_n , are available as the observations corresponding to hypothetically an identical 3D point X_j in the scene. The features points are projected on π using the analytical homographies defined in (17.10), where π is the dominant ground plane of the scene. If the 3D point X_j lies on π , its corresponding mapped homography points should be all close by each other. Indeed, in an ideal case where the camera poses are accurate, all such mapped points on π have to merge and coincide to a single point, however it is not often the case due to different source of noise in IMU and GPS measurements. Here, we use the mean of the homographic transformed points on π as an estimate to initialize the optimization. e_j^1 and e_j^n are the Euclidean distances between each projected point and the mean. The optimization defined in (17.15) aims to minimize these distance errors by adjusting the camera poses. Notice that if 3D point X_j does not lie on π , its error values are automatically marginalized during the optimization process, thanks to the used robust functions

the presence of 3D structures/buildings is highly expected. The observed 3D points from such structures once imaged and mapped onto plane π , their corresponding 2D points would not coincide on π and will be dispersed. This phenomenon is known as *parallax*, and its magnitude gets stringer as the 3D point get farther from the plane (π) which induces the homography. For example, in Fig. 17.1, consider X_2 as a 3D point which is off-the-plane. It is imaged as x_2^1 , x_2^2 , and x_2^n on the image planes of cameras C_1 , C_2 , and C_n . Mapping them on π using homography transformations will result πx_2^1 , πx_2^2 , and πx_2^n . As illustrated in Fig. 17.1, these mapped points are all spread out on π , and the radius of distribution is proportional to the magnitude of parallax.

There is another type of noise which is likely to exist in the tracks of feature correspondences along the image sequence. The source of such noise can be from the precision of the feature extraction algorithm or errors in the feature matching

algorithm which could lead to many mismatches or outliers. In real scenarios, one can expect to have a considerable percentage of outliers. To deal with outliers, mostly RANSAC (or its variations) is used in the literature. In this context, a RANSAC-based approach tries to (jointly) estimate a homography model and at the same time to eliminate the outliers, by looping through a hundreds of iterations. In each iteration of RANSAC, a subset of correspondence candidates is randomly chosen, a homography model is estimated for the chosen population, and then the fitness of the whole population of the correspondences is measured using the estimated model. In this randomly exhaustive process, a model that provides the most consensus result would be chosen and at the same time, the feature matches which do not obey the estimated model within a threshold will be identified as outliers. Notice that in our work, the homographies are analytically derived and no RANSAC estimation is used and instead the inaccurate sensor measurements from the platform are directly incorporated. Not using RANSAC gives the advantage of avoiding any adverse random behavior in the model estimation. However, as a consequence of eliminating RANSAC, the existing outliers cannot be explicitly identified. In order to address this issue, we propose to use a robust error function in an appropriate formulation of the problem.

Robust functions also known as M-estimators are popular in robust statistics and reduce the influence of outliers in estimation problems. We have observed that not every choice of a robust function works well [46] and a proper robust function is critical for achieving a robust minimization of the reprojection error when the initial parameters are too noisy and outliers are not explicitly eliminated beforehand. Two commonly used robust statistics functions are the *Cauchy* (or *Lorentzian*) and *Huber* [45] measures:

- Cauchy or Lorentzian cost function

$$\rho(s) = b^2 \log(1 + s^2/b^2) \quad (17.16)$$

- Huber cost function

$$\rho(s) = \begin{cases} s^2 & \text{if } |s| < b \\ 2b|s| - b^2 & \text{otherwise} \end{cases} \quad (17.17)$$

where s is the residual (i.e. reprojection error) in (17.15) and b is usually one or a fixed user defined value. We have chosen Cauchy robust function since it down-weights the residuals more rigidly [47]. This characteristic of Cauchy is appropriate for our purpose especially because there expect to be enormous number of large residuals due to potential parallaxes in the scene. One can consider using other types of robust functions such as a generalization of the *Cauchy/Lorentzian*, *Geman-McClure*, *Welsch*, and *generalized Charbonnier* loss functions [48].

The proposed optimization method is presented in a pseudo code form in Algorithm 1. This method is an alternative approach for the parameter optimization with an emphasize on stabilizing the geoprojected aerial imagery by defining a

Algorithm 1 Analytical airborne video stabilization

Input : A set of camera parameters acquired from inaccurate platform sensors, e.g. IMU and GPS: $(\mathbf{R}_i, \mathbf{t}_i, f)$, $i = 1 \dots n$, n being number of cameras/images.
 m sets of tracked features along the sequence.

Output : Optimized homography matrices to robustly stabilize the imagery

- 1: $\mathbf{v} \leftarrow [u \ v \ f]^T$
- 2: **for** $i = 1$ **to** n **do**
- 3: $\mathbf{T}_i \leftarrow [\mathbf{r}_{1,i} \ \mathbf{r}_{2,i} \ \mathbf{t}_i]$
- 4: Assign $m_{bc,i}$ as the minor(b,c) of matrix \mathbf{T}_i
- 5: $\mathbf{H}_{i \rightarrow \pi} \leftarrow \begin{bmatrix} m_{11,i} & -m_{21,i} & [-m_{11,i} & m_{21,i} & m_{31,i}] \mathbf{v} \\ -m_{12,i} & m_{22,i} & [m_{12,i} & -m_{22,i} & -m_{32,i}] \mathbf{v} \\ r_{13,i} & r_{23,i} & -\mathbf{r}_{3,i}^T \mathbf{v} \end{bmatrix}$
- 6: **end for**
- 7: **for** $j = 1$ **to** m **do**
- 8: **for** $i = 1$ **to** n **do**
- 9: ${}^\pi \tilde{\mathbf{x}}_j^i \leftarrow \mathbf{H}_{i \rightarrow \pi} \tilde{\mathbf{x}}_j^i$
- 10: **end for**
- 11: ${}^\pi \hat{\mathbf{x}}_j \leftarrow \frac{1}{n} \sum_{i=1}^n {}^\pi \tilde{\mathbf{x}}_j^i$
- 12: **end for**
- 13: $E \leftarrow \sum_{i=1}^n \sum_{j=1}^m \|\mathcal{F}(\mathbf{H}_{i \rightarrow \pi} \tilde{\mathbf{x}}_j^i) - \mathcal{F}({}^\pi \hat{\mathbf{x}}_j)\|^2$
- 14: Optimize \mathbf{R}_i , \mathbf{t}_i and ${}^\pi \hat{\mathbf{x}}_j$ to minimize the cost function E
- 15: **for** $i = 1$ **to** n **do**
- 16: $\hat{\mathbf{T}}_i \leftarrow [\hat{\mathbf{r}}_{1,i} \ \hat{\mathbf{r}}_{2,i} \ \hat{\mathbf{t}}_i]$
- 17: Assign $\hat{m}_{bc,i}$ as the minor(b,c) of matrix $\hat{\mathbf{T}}_i$
- 18: $\hat{\mathbf{H}}_{i \rightarrow \pi} \leftarrow \begin{bmatrix} \hat{m}_{11,i} & -\hat{m}_{21,i} & [-\hat{m}_{11,i} & \hat{m}_{21,i} & \hat{m}_{31,i}] \mathbf{v} \\ -\hat{m}_{12,i} & \hat{m}_{22,i} & [\hat{m}_{12,i} & -\hat{m}_{22,i} & -\hat{m}_{32,i}] \mathbf{v} \\ \hat{r}_{13,i} & \hat{r}_{23,i} & -\hat{\mathbf{r}}_{3,i}^T \mathbf{v} \end{bmatrix}$
- 19: **end for**
- 20: **return** optimized homography matrices $\hat{\mathbf{H}}_{i \rightarrow \pi}$, $i = 1 \dots n$

cost function over a *single dominant 2D Euclidean world plane*. The points that do not lie on the dominant ground plane are *automatically marginalized* during the optimization process, thanks to the used robust functions, instead of using a RANSAC-based outlier elimination approach.

17.5 Experiments

The proposed method was applied to the DARPA Video and Image Retrieval and Analysis Tool (VIRAT) dataset and several WAMI datasets provided by Transparent-Sky (<http://www.transparensky.net>). The introduced sequential feature tracking method was used to track the identified SIFT features over each video sequence, followed by applying the proposed optimization method. The top images in Fig. 17.3 are some sample frames from a shot in the sequence “flight2Tape1_2” of VIRAT dataset which contains 2400 images. The metadata that comes with the images is

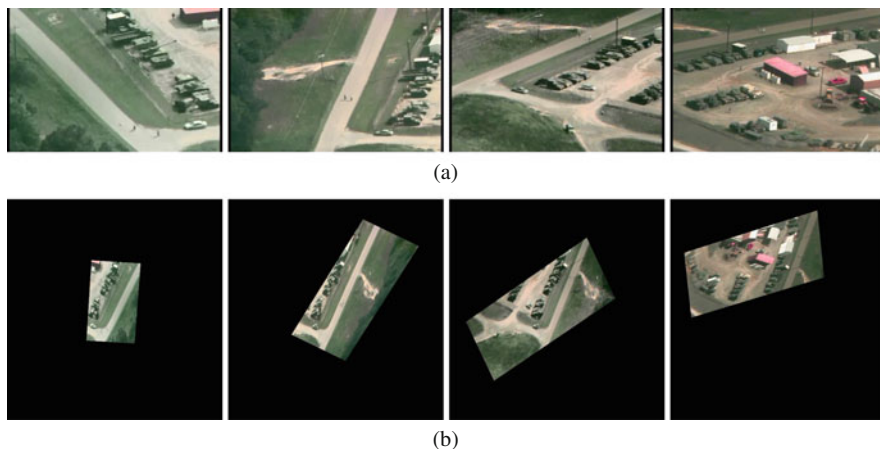


Fig. 17.3 Stabilized sequence of VIRAT dataset using the proposed method. We used a full shot from “flight2Tape1_2” which contains 2400 frames of 720×480 pixels. The camera metadata in all VIRAT datasets is extremely inaccurate. Our approach managed to perform the georegistration and stabilization on this long sequence without any jump or jitter in the result. **(a)** Original images (frame numbers: 3615, 4610, 5351, and 5901). **(b)** Stabilized and geoprojected images (frame numbers: 3615, 4610, 5351, and 5901)

extremely inaccurate. To the best of our knowledge, these metadata have not been of use in any SfM, stabilization, and georeprojection project. However, our approach managed to seamlessly register the full video shot, smoothly with no jitter or jump. The results corresponding to the frames of the first row are shown in Fig. 17.3 bottom.

Figure 17.4 shows the result of running our method on a WAMI aerial imagery. The metadata and images were provided by Transparent-Sky (<http://www.transparentsky.net>) via flying a fixed wing airplane over the downtown of Berkeley in California. Two exemplary images, with about 45° difference in their viewing angle along (200 frames apart along the sequence), are shown in Fig. 17.4-top. Their corresponding georegistered frames are plotted in Fig. 17.4-middle. The bounding boxes of the regions of the interest from the two frames are zoomed and shown in Fig. 17.4-Bottom. The rectified epipolar line (yellow dotted line) demonstrates the alignments for an exemplary pair of corresponding points (in red) in the two frames after stabilization. A similar evaluation is demonstrated for ABQ (Albuquerque downtown area) WAMI dataset, in Fig. 17.5. Figure 17.6 depicts the original and stabilized images from another WAMI dataset, LA downtown area. As one can see, everything from the dominant ground plane is well aligned between the two registered views, and just the building and off the ground objects were wobbled which is due to the existence of parallax. Despite the presence of strong parallax, the method succeeded to seamlessly stabilize the images without any jitter. It is worth to remind that no RANSAC or any other random-based method has been used in the proposed.

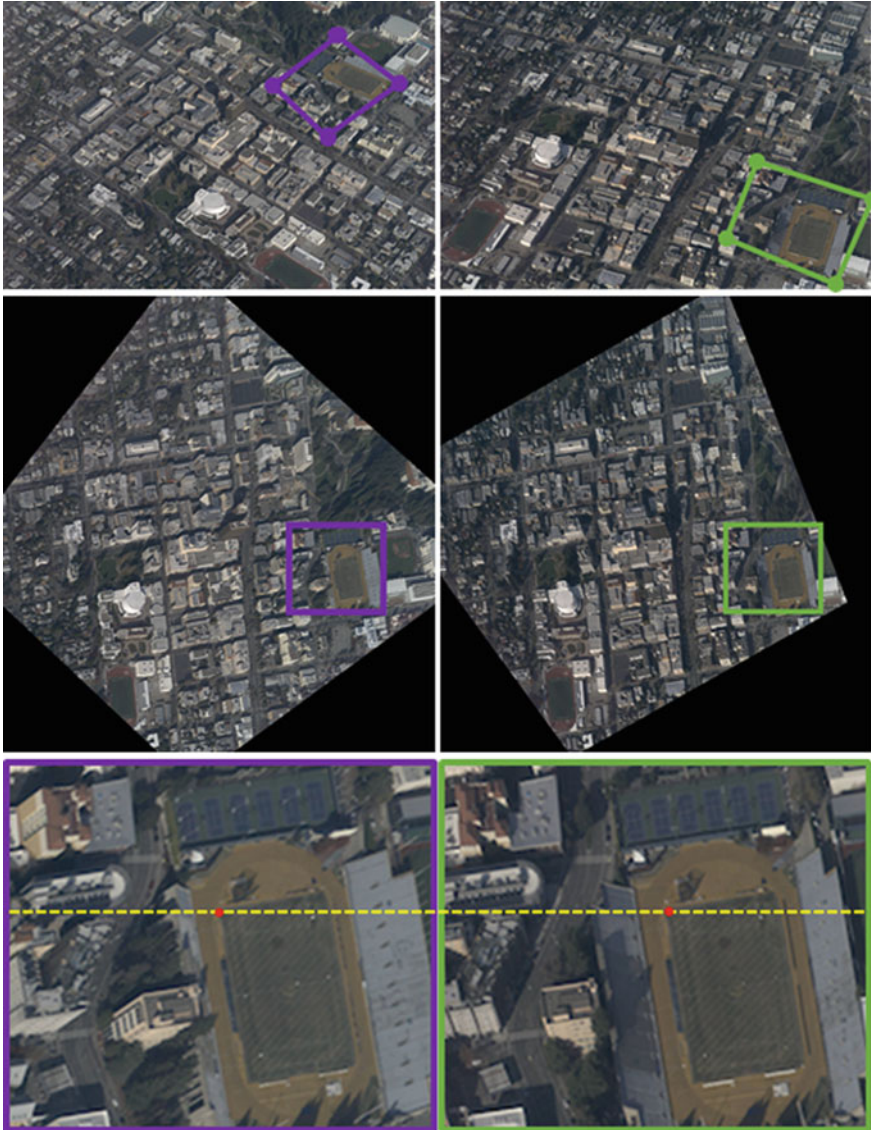


Fig. 17.4 Stabilization result of Berkeley dataset. **Top:** two raw WAMI images, with size of 6600×4400 pixels (frame #0 at left, frame #200 at right). **Middle:** geoprojection of the raw frames after stabilization using the proposed approach. **Bottom:** Zoomed-in versions of the middle row corresponding to the areas which are marked by *purple* and *green* bounding boxes. The rectified epipolar line (yellow dotted line) depicts the alignment for a pair of corresponding points (in red) after stabilization

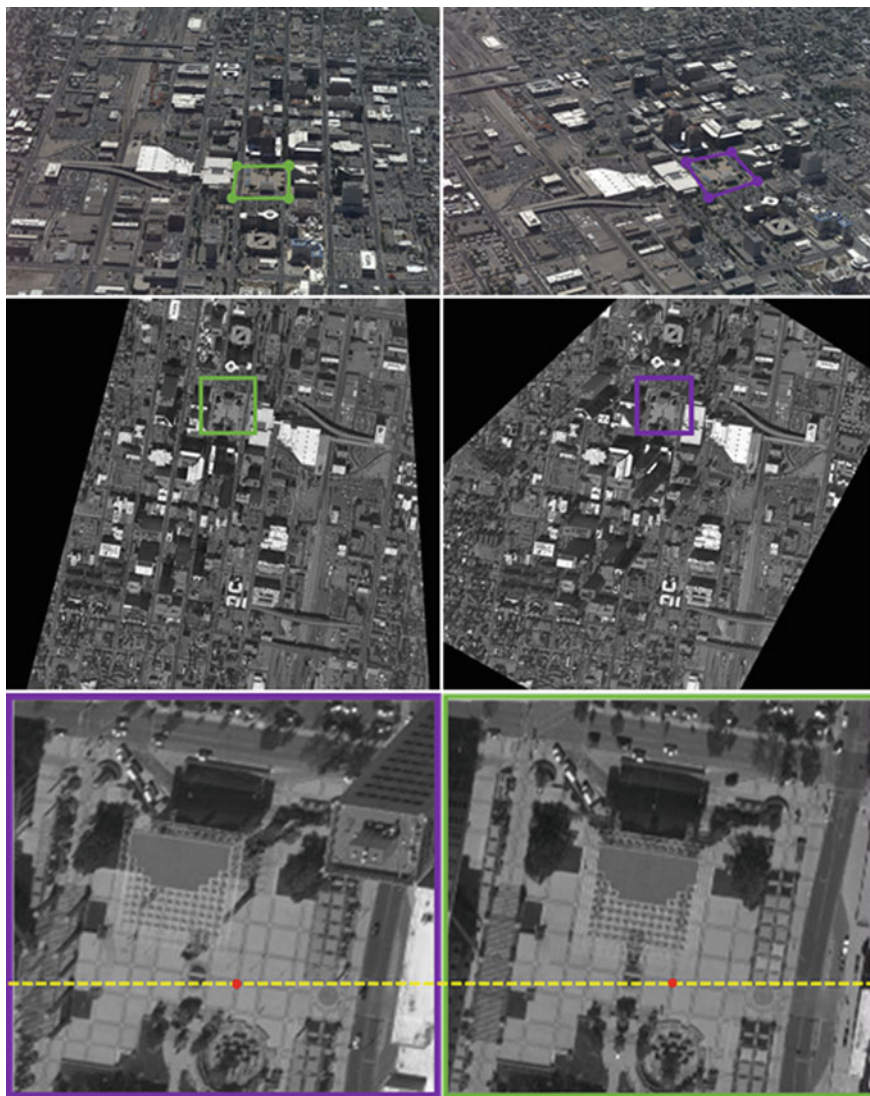


Fig. 17.5 Stabilization result of Albuquerque dataset. **Top:** two raw WAMI images, with size of 6600×4400 pixels (frame #0 at left, frame #100 at right). **Middle:** geoprojection of the raw frames after stabilization using the proposed approach. **Bottom:** Zoomed-in versions of the middle row corresponding to the areas which are marked by *purple* and *green* bounding boxes. The rectified epipolar line (yellow dotted line) depicts the alignment for a pair of corresponding points (in red) after stabilization

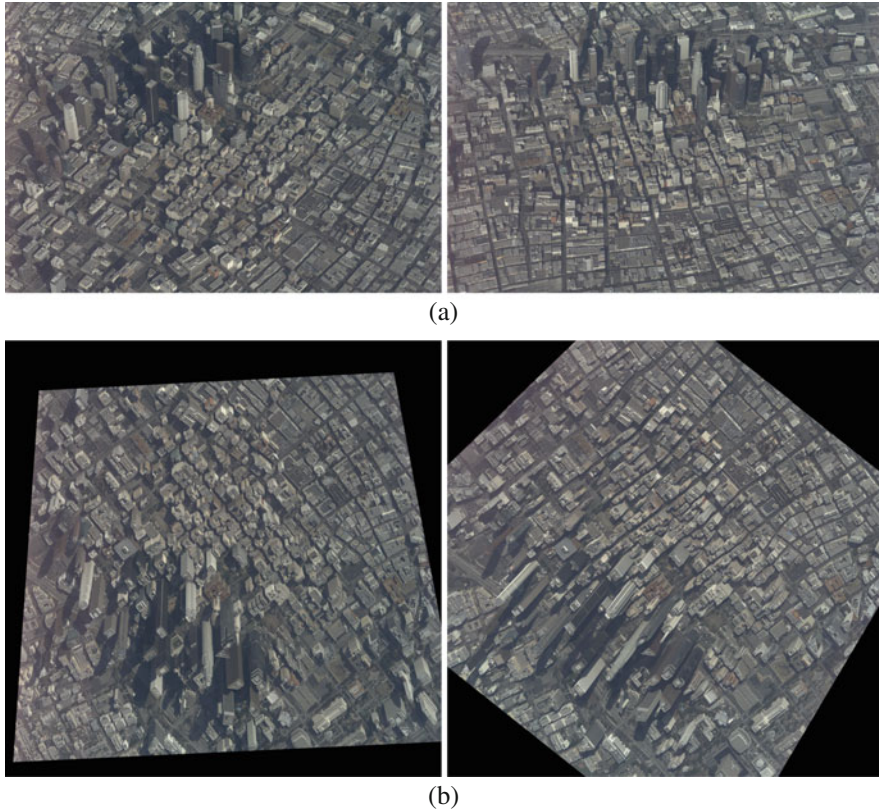


Fig. 17.6 Stabilized sequence of Los Angeles (California) dataset using the proposed method. The high resolution WAMI imagery (with the image size of 6600×4400) along with initial metadata were provided by Transparent-Sky (<http://www.transparensky.net>). Despite presence of strong parallax induced by the tall buildings, our method managed to smoothly stabilize the WAMI images. **(a)** Original images (frames #0 and #100). **(b)** Stabilized and geoprojected images (frames #0 and #100)

17.6 Conclusions

We proposed a stabilization and geoprojection method which is able to use available sensor metadata (i.e. GPS and IMU) to register airborne video in a robust and seamless manner. This became possible by deriving a set of analytical homography transformations and defining a *metric* cost function over a dominant 2D *Euclidean* ground plane in the scene. The solution has been formulated such that no RANSAC (any random-based iterative techniques) is used, in contrary to most existing approaches. The robustness in our work is achieved by defining an appropriate (robust) cost function which allows to implicitly marginalize the outliers automatically within the optimization process. Our approach has been

tested over a very challenging dataset of DARPA, known as VIRAT. Unlike the imagery component of this dataset is very rich and has been frequently used in different algorithms by several well-known research groups, its metadata component is extremely challenging. We know no group or research work which could have relied on the metadata in this dataset and used it in a SfM or stabilization method, as the available sensor measurements are highly inaccurate. Nevertheless, our approach has been tested on this dataset where the challenging metadata was directly utilized to perform a smooth and seamless stabilization on the video sequence. In addition to VIRAT dataset, two high resolution WAMI datasets corresponding to the downtown areas of Berkeley and Los Angeles were successfully tested and stabilized in our experiments.

Acknowledgements This research was partially sponsored by the Army Research Laboratory and Air Force Research Laboratory under Cooperative Agreements W911NF-18-2-0285 and FA8750-19-2-0001, respectively. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Palaniappan, K., Rao, R., & Seetharaman, G. (2011). Wide-area persistent airborne video: Architecture and challenges. In *Distributed video sensor networks: Research challenges and future directions* (pp. 349–371). London: Springer.
2. Porter, R., Fraser, A. M., & Hush, D. (2010). Wide-area motion imagery. *IEEE Signal Processing Magazine*, 27(5), 56–65.
3. Poostchi, M., Aliakbarpour, H., Viguier, R., Bunyak, F., Palaniappan, K., & Seetharaman, G. (2016). Semantic depth map fusion for moving vehicle detection in aerial video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1575–1583).
4. Palaniappan, K., Poostchi, M., Aliakbarpour, H., Viguier, R., Fraser, J., Bunyak, F., et al. (2016). Moving object detection for vehicle tracking in wide area motion imagery using 4D filtering. In *IEEE International Conference on Pattern Recognition (ICPR)*.
5. Nilosek, D. R., Walvoord, D. J., & Salvaggio, C. (2014). Assessing geoaccuracy of structure from motion point clouds from long-range image collections. *Optical Engineering*, 53(11), 1–10.
6. Linger, M. E., & Goshtasby, A. (2015). Aerial image registration for tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4), 2137–2145.
7. Holtkamp, D. J., & Goshtasby, A. A. (2009). Precision registration and mosaicking of multicamera images. *IEEE Transactions on Geoscience and Remote Sensing*, 47, 3446–3455.
8. Lee, J., Cai, X., Schonlieb, C.-B., & Coomes, D. A. (2015). Nonparametric image registration of airborne LiDAR, hyperspectral and photographic imagery of wooded landscapes. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–12.
9. Crispell, D., Mundy, J. L., & Taubin, G. (2008). Parallax-free registration of aerial video. In *Proceedings of the British Machine Vision Conference* (pp. 1–4).

10. Aliakbarpour, H., Palaniappan, K., & Seetharaman, G. (2015). Fast structure from motion for sequential and wide area motion imagery. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.
11. Aliakbarpour, H., Palaniappan, K., & Seetharaman, G. (2015). Robust camera pose refinement and rapid SfM for multiview aerial imagery - without RANSAC. *IEEE Geoscience and Remote Sensing Letters*, 12(11), 2203–2207.
12. Aliakbarpour, H., Palaniappan, K., & Seetharaman, G. (2017). Parallax-tolerant aerial image georegistration and efficient camera pose refinement- without piecewise homographies. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8), 4618–4637.
13. Aliakbarpour, H., Prasath, V. B. S., Palaniappan, K., Seetharaman, G., & Dias, J. (2016). Heterogeneous multi-view information fusion: Review of 3-D reconstruction methods and a new registration with uncertainty modeling. *IEEE Access*, 4, 8264–8285.
14. Aliakbarpour, H., & Dias, J. (2012). Three-dimensional reconstruction based on multiple virtual planes by using fusion-based camera network. *IET Computer Vision*, 6(4), 355.
15. Aliakbarpour, H., Almeida, L., Menezes, P., & Dias, J. (2011). Multi-sensor 3D volumetric reconstruction using CUDA. *3D Research*, 2(4), 1–14.
16. Aliakbarpour, H., & Dias, J. (2011). PhD forum: Volumetric 3D reconstruction without planar ground assumption. In *Fifth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* (pp. 1–2). Piscataway: IEEE.
17. Molina, E., & Zhu, Z. (2014). Persistent aerial video registration and fast multi-view mosaicing. *IEEE Transactions on Image Processing*, 23(5), 2184–2192.
18. Saleemi, I., & Shah, M. (2013). Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos. *International Journal of Computer Vision*, 104(2), 198–219.
19. Medioni, G., Cohen, I., Bremond, F., Hongeng, S. & Nevatia, R. (2001). Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), 873–889.
20. Lin, Y., Yu, Q., & Medioni, G. (2010). Efficient detection and tracking of moving objects in geo-coordinates. *Machine Vision and Applications*, 22, 505–520.
21. Chellappa, V. M., & Govindu, R. (2011). Feature-based image to image registration. In *Image Registration for Remote Sensing* (pp. 215–239). Cambridge: Cambridge University Press.
22. Taylor, C. N. (2013). Improved evaluation of geo-registration algorithms for airborne EO/IR imagery. In *SPIE, Geospatial Infofusion III* (Vol. 8747, p. 874709).
23. Vasquez, J., Hytla, P., Asari, V., Jackovitz, K., & Balster, E. (2012). Registration of region of interest for object tracking applications in wide area motion imagery. In *The IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1–8).
24. Stone, H. S., Orchard, M. T., Chang, E. C., & Martucci, S. A. (2001). A fast direct Fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(10), 2235–2243.
25. Hafiane, A., Palaniappan, K., & Seetharaman, G. (2008). UAV-video registration using block-based features. In *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium* (Vol. 2, no. 1, pp. II-1104–II-1107).
26. Seetharaman, G., Gasperas, G., & Palaniappan, K. (2000). A piecewise affine model for image registration in nonrigid motion analysis. In *Proceedings of the International Conference on Image Processing, 2000* (pp. 561–564).
27. Zhu, Z., Hanson, A. R., & Riseman, E. M. (2004). Generalized parallel-perspective stereo mosaics from airborne video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), 226–237.
28. Jackson, B. P., & Goshtasby, A. (2014). Adaptive registration of very large images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 351–356).

29. Castro-Toscano, M. J., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., Lindner, L., Sergiyenko, O., Rivas-Lopez, M., & Flores-Fuentes, W. (June 2017). A methodological use of inertial navigation systems for strapdown navigation task. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)* (pp. 1589–1595).
30. Lindner, L., Sergiyenko, O., Rivas-López, O., Ivanov, M., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (June 2017). Machine vision system errors for unmanned aerial vehicle navigation. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)* (pp. 1615–1620).
31. Lindner, L., Sergiyenko, O., Rivas-López, M., Hernández-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., et al. (2017). Exact laser beam positioning for measurement of vegetation vitality. In *Industrial Robot: the International Journal of Robotics Research and Application*, 44(4), 532–541.
32. Turner, D., Lucieer, A., & Wallace, L. (2014). Direct georeferencing of ultrahigh-resolution UAV imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5), 2738–2745.
33. Agisoft, *agisoft photoscan professional*. <http://www.agisoft.com>
34. Pix4D. <http://pix4d.com>
35. Snavely, N. *Bundler: Structure from motion (SfM) for unordered image collections*. <http://phototour.cs.washington.edu/bundler>
36. Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80, 189–210.
37. Pritt, M. D. (2014). Fast orthorectified mosaics of thousands of aerial photographs from small UAVs. In *Applied Imagery Pattern Recognition Workshop (AIPR)*. Piscataway: IEEE.
38. Aliakbarpour, H., Nuez, H. P., Prado, J., Khoshhal, K., & Dias, J. (2009). An efficient algorithm for extrinsic calibration between a 3D laser range finder and a stereo camera for surveillance. In *2009 International Conference on Advanced Robotics*.
39. Redmill, K. A., Martin, J. I., & Ozguner, U. (2009). Aerial image registration incorporating GPS/IMU data. *Proceedings of SPIE*, 7347(1), 73470H–73470H–15.
40. Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., & Tian, J. (2015). Robust feature matching for remote sensing image registration via locally linear transforming. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–13.
41. Liu, Z., An, J., & Jing, Y. (2012). A simple and robust feature point matching algorithm based on restricted spatial order constraints for aerial image registration. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2), 514–527.
42. Aktar, R., Aliakbarpour, H., Bunyak, F., Kazic, T., Seetharaman, G., & Palaniappan, K. (2018). Geospatial content summarization of UAV aerial imagery using mosaicking. In *Proceedings of SPIE - The International Society for Optical Engineering* (Vol. 10645).
43. Viguier, R., Lin, C. C., AliAkbarpour, H., Bunyak, F., Pankanti, S., Seetharaman, G., et al. (2015). Automatic video content summarization using geospatial mosaics of aerial imagery. In *2015 IEEE International Symposium on Multimedia (ISM)* (pp. 249–253).
44. Viguier, R., Lin, C. C., Swaminathan, K., Vega, A., Buyuktosunoglu, A., Pankanti, S., et al. (2015). Resilient mobile cognition: Algorithms, innovations, and architectures. In *Proceedings of the 33rd IEEE International Conference on Computer Design, ICCD 2015* (pp. 728–731).
45. Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment : A modern synthesis. *Vision algorithms: theory and practice*. S, 34099, 298–372.
46. Aliakbarpour, H., Palaniappan, K., & Seetharaman, G. (2015). Robust camera pose refinement and rapid SfM for multi-view aerial imagery without RANSAC. *IEEE Journal of Geoscience and Remote Sensing Letters*, 12(11), 2203–2207.
47. Johan, A. A., Suykens, A. K., & Signoretto, M. (2014). *Regularization, optimization, kernels, and support vector machines*. London/Boca Raton: Chapman and Hall/CRC Press.
48. Barron, J. T. (2017). A more general robust loss function. *Arxiv*, 1(5), 2–5. <https://arxiv.org/abs/1701.03077>

Chapter 18

Visual Servo Controllers for an UAV Tracking Vegetal Paths



Jorge A. Sarapura, Flavio Roberti, Juan Marcos Toibero,
José María Sebastián, and Ricardo Carelli

Acronyms

ExG-ExR	Excess green minus excess red vegetation index
GPS	Global positioning system
IMU	Inertial measurement unit
RMSE	Root mean square error
TLS	Total least squares algorithm
UAV	Unmanned aerial vehicles

18.1 Introduction

In recent years, due to advances in computer technology and robotics, autonomous aerial vehicles (UAVs) have found numerous applications, such as search and rescue, forest fire monitoring, surveillance, remote inspection, and precision agriculture. This last, is an upcoming technology with a great value to modernize the agriculture applications and their purpose is not only to apply robotic technologies in the field of agriculture, but also to develop new techniques and systems to adapt to contemporary agricultural challenges. As most agricultural environments have large semi-structured open spaces, autonomous vehicles have experienced great

J. A. Sarapura (✉) · F. Roberti · J. M. Toibero · R. Carelli
Instituto de Automática, UNSJ-CONICET, San Juan, Argentina
e-mail: jsarapura@inaut.unsj.edu.ar; froberti@inaut.unsj.edu.ar; mtoibero@inaut.unsj.edu.ar;
rcarelli@inaut.unsj.edu.ar

J. M. Sebastián
Centro de Automática y Robótica, Universidad Politécnica de Madrid, Madrid, Spain
e-mail: jsebas@etsii.upm.es

interest since the advancement of autonomous robotic technologies has promoted the possibility of sending unmanned vehicles to operate in these large areas [2].

In general, the aerial robots use a global positioning system (GPS) or an inertial measurement unit (IMU) as sensors to measure the position and speed of the vehicle, required by the internal controllers and for the planning of the trajectories during an autonomous flight. However, the GPS cannot be used in some environments such as indoor environments or near buildings and trees, as it has large errors due to different sources such as obstructed signals, signals with multiple routes in crowded environments, and jammed or denied receptions. On the other hand, an IMU used as position sensor for navigation employs accelerometers and gyros to find the linear and angular movement from the starting point; but their measurements suffer from accumulated error over time and even with a small drift error it will accumulate up to a large value over time [1, 21].

In the real-world scenario of agriculture when the UAV will be flying close to vegetation, the use of sensors and techniques based on artificial vision allows to overcome the above-mentioned problems, improving the estimation of the vehicle position during navigation in such environments due to its greater capacity to capture the environment information [4, 6]. Numerous works have proposed control systems using only visual information for the control of autonomous aerial vehicles [3, 18].

Other works use camera-based artificial vision systems as complementary devices. In [10, 11] are presented laser-based scanning systems that allow precise measurements of the distance to the objects under observation or the vitality of the agricultural vegetation, respectively. The complementary use of the vision system with the laser allows quick measurements to be obtained when working with a non-redundant amount of information.

A main topic for the visual servoing is the image processing for the automatic detection of crops [5, 7, 8, 13, 14]. In this work a vegetation index was used because it accentuates a particular color such as the greenness of the plants which is intuitive for human comparison, and is less sensitive to variations in lighting as well as to different residues backgrounds [12]. Based on the image obtained with the vegetation index, a simple and fast algorithm was developed capable of detecting a crop line and providing references to the quadcopter for its autonomous navigation.

Among the agricultural tasks addressed by precision agriculture is the inspection and data collection in structured rows of crop plantations, navigating at low-altitude above the objects to be inspected. In this area, works based on artificial vision are found such as [18] with a visual control system of an UAV to navigate along rows of a crop field using the concept of oriented textures; or the one in [3] where a kinematic servo-visual controller is presented for tracking paths with references extracted by means of an artificial vision algorithm.

In order to achieve controllers with high performance, their design should consider the dynamics of the UAV and be theoretically validated through stability proofs. In [17] the authors have presented the first version of a visual servoing controller for following straight paths based on passivity properties of the visual and controller systems.

In addition to the robustness and stability properties, the controller should satisfy with a good performance regarding the response speed and the magnitude of the control actions. To this we propose in this work three visual servo controllers based on different design principles (including the one based on passivity properties), and compare their performance when applied to an UAV following vegetal lines.

The main contribution of this work is the development of the image based controller using passivity properties of the system, the stability and robustness analysis, and the comparative performance with the other controllers. This study is of practical value when designing UAVs visual servoing for a specific application.

18.2 UAV Models

The kinematic and simplified dynamic models of a quadcopter type UAV, Fig. 18.1, is now presented. These models have been used for the controller design.

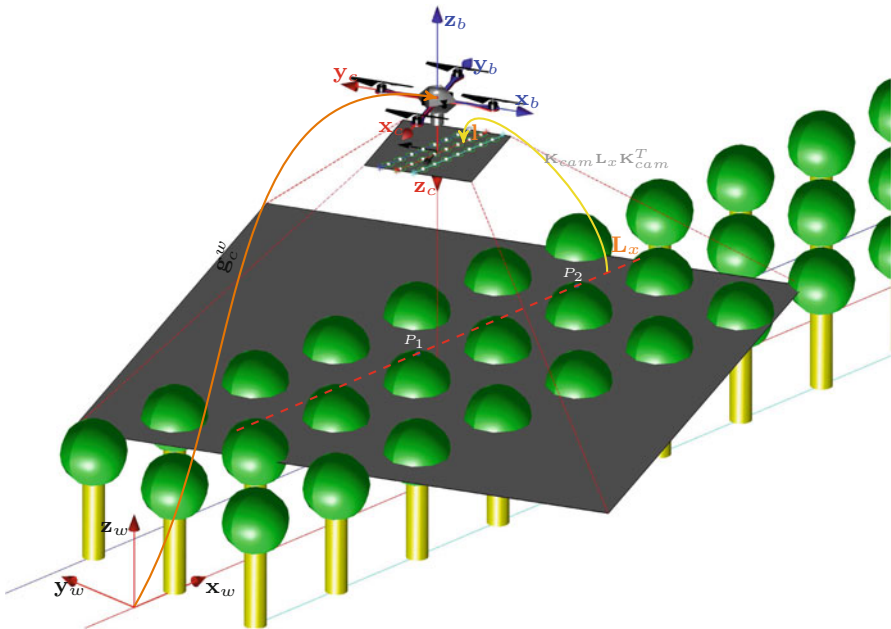


Fig. 18.1 System reference frames

18.2.1 Kinematic Model

This model is expressed by four speeds $\dot{\mathbf{x}}^b = [\dot{x}^b, \dot{y}^b, \dot{z}^b, \dot{\psi}^b]^T$ expressed in the frame $\langle b \rangle$ of the vehicle, transformed in the inertial frame $\langle w \rangle$ through a matrix $\mathbf{F}(\psi)$ [16]. Each linear velocity (\dot{x}^b , \dot{y}^b and \dot{z}^b) produces a displacement on an axis of $\langle b \rangle$ and the angular velocity $\dot{\psi}^b$ a rotation around the axis z_b . Therefore the UAV movements can be described by

$$\dot{\mathbf{x}}^w = \mathbf{F}\dot{\mathbf{x}}^b, \quad (18.1)$$

where $\dot{\mathbf{x}}^w = [\dot{x}, \dot{y}, \dot{z}, \dot{\psi}]^T$ are the velocities in $\langle w \rangle$ and the $\mathbf{F}(\psi)$ matrix is given by

$$\mathbf{F}(\psi) = \begin{bmatrix} \mathbf{R}(\mathbf{z}^w, \psi) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_\psi & -s_\psi & 0 & 0 \\ s_\psi & c_\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (18.2)$$

where $\mathbf{R}(\mathbf{z}^w, \psi)$ and ψ represents a rotation matrix and yaw angle around the axis z_w , respectively, \mathbf{z}^w is the z_w axis unit vector and $c_* \doteq \cos(*)$, $s_* \doteq \sin(*)$. In this transformation, the angles of rotation around the x and y axes are neglected and considered small for a smooth navigation condition.

18.2.2 Dynamic Model

The simplified dynamic model here considered is [15]

$$\mathbf{u}^b = \mathbf{A}\ddot{\mathbf{x}}^w + \mathbf{B}\dot{\mathbf{x}}^w, \quad (18.3)$$

where $\dot{\mathbf{x}}^w = [\dot{x}, \dot{y}, \dot{z}, \dot{\psi}]^T$ y $\ddot{\mathbf{x}}^w = [\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\psi}]^T$ are the speeds and accelerations of the UAV in $\langle w \rangle$; and $\mathbf{u}^b = [u_x, u_y, u_z, u_\psi]^T$ the control actions in $\langle b \rangle$. Matrices \mathbf{A} and \mathbf{B} are given by

$$\mathbf{A} = (\mathbf{F}\mathbf{K}_u)^{-1}, \quad (18.4)$$

$$\mathbf{B} = \mathbf{A}\mathbf{K}_v. \quad (18.5)$$

For the ArDrone quadcopter, the positive definite diagonal matrices containing the dynamic parameters have been identified as given by

$$\mathbf{K}_u = \begin{bmatrix} 4.72 & 0 & 0 & 0 \\ 0 & 6.23 & 0 & 0 \\ 0 & 0 & 2.65 & 0 \\ 0 & 0 & 0 & 2.38 \end{bmatrix}, \quad (18.6)$$

$$\mathbf{K}_v = \begin{bmatrix} 0.28 & 0 & 0 & 0 \\ 0 & 0.53 & 0 & 0 \\ 0 & 0 & 2.58 & 0 \\ 0 & 0 & 0 & 1.52 \end{bmatrix}. \quad (18.7)$$

The model (18.3) can be rewritten in $\langle b \rangle$ as

$$\mathbf{u}^b = \mathbf{H}\ddot{\mathbf{x}}^b + \mathbf{C}\dot{\mathbf{x}}^b, \quad (18.8)$$

where $\mathbf{H} = \mathbf{K}_u^{-1}$ and $\mathbf{C} = \mathbf{K}_u^{-1}\mathbf{K}_v$ are positive definite symmetric matrices and $\dot{\mathbf{x}}^b = [\dot{x}^b, \dot{y}^b, \dot{z}^b, \dot{\psi}^b]^T$ y $\ddot{\mathbf{x}}^b = [\ddot{x}^b, \ddot{y}^b, \ddot{z}^b, \ddot{\psi}^b]^T$ are the speeds and accelerations of the UAV in $\langle b \rangle$.

18.3 Vision System

The modelling of the vision system provides the equations that relate the line to follow in the 3-D space by the UAV and its corresponding projection in the image plane. It is considered that the camera, with associated frame $\langle c \rangle$, is mounted on the quadcopter base on a gimbal type stabilizer device so that it takes images always parallel to the ground plane, as shown in Fig. 18.1.

18.3.1 Image Processing

The vision system was implemented using a single board computer *Raspberry Pi II* with a *Raspicam* camera. The camera captures color images in *RGB* format of 640×480 px at a rate of 30 fps. The processing of the images acquired by the vision system was done with the OpenCV image processing library and can be described in five stages, as shown in Fig. 18.2, which includes the processing times.

In the first stage an *RGB* color image of the crop row is captured, Fig. 18.3a. Then, a segmentation of the image is made for discriminating by color the plants from the rest of the soil, by selecting a color space and an optimal vegetation index [12]. The result of this stage is an almost binary gray image, Fig. 18.3b.

In the third stage, the gray image obtained is thresholded with a simple zero value due to the choice of the improved excess green minus excess red vegetation index (*ExG-ExR*). In the next stage, a filtering of the binary image is performed using

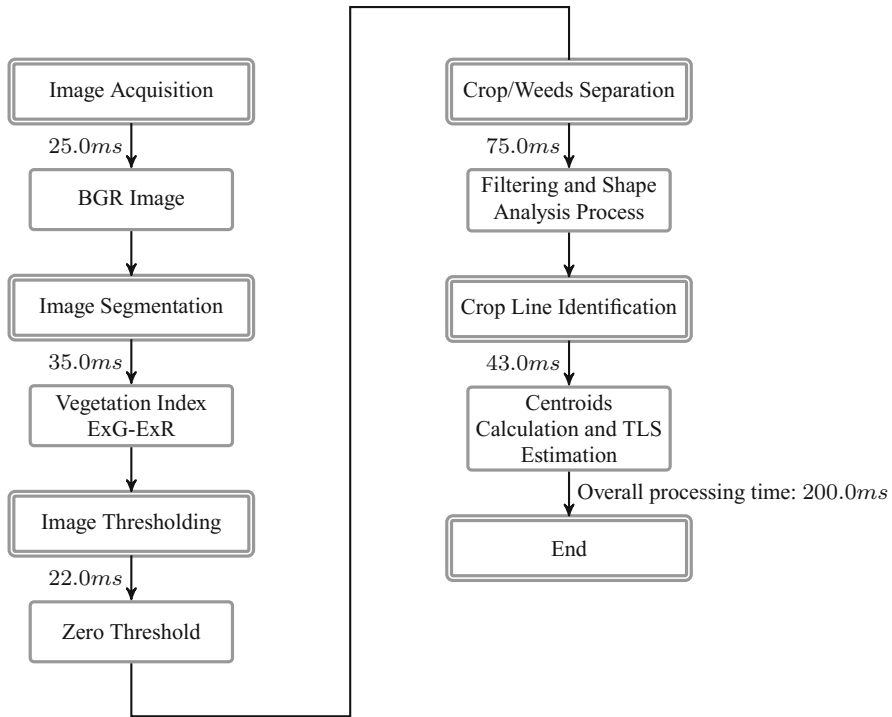


Fig. 18.2 Image processing algorithm

several morphological transformations and by means of techniques of shape analysis, it is completed the discrimination between the treetops and the soil, Fig. 18.3c.

Finally, in the last stage, the centroids of the detected trees are found and, using a total least squares algorithm, the best line passing through them is estimated as the line that describes the row of trees, Fig. 18.3d. In addition, in this estimate continuity constraints are imposed to discriminate treetops in contiguous rows.

18.3.2 Kinematics of the Vision System

The way forward for the UAV, without loss of generality, is a straight line L_x coinciding with the x^w axis (line $y_w = 0$ in $\langle w \rangle$) of the 3-D space defined by a skew-symmetric matrix Plücker's [9] given by

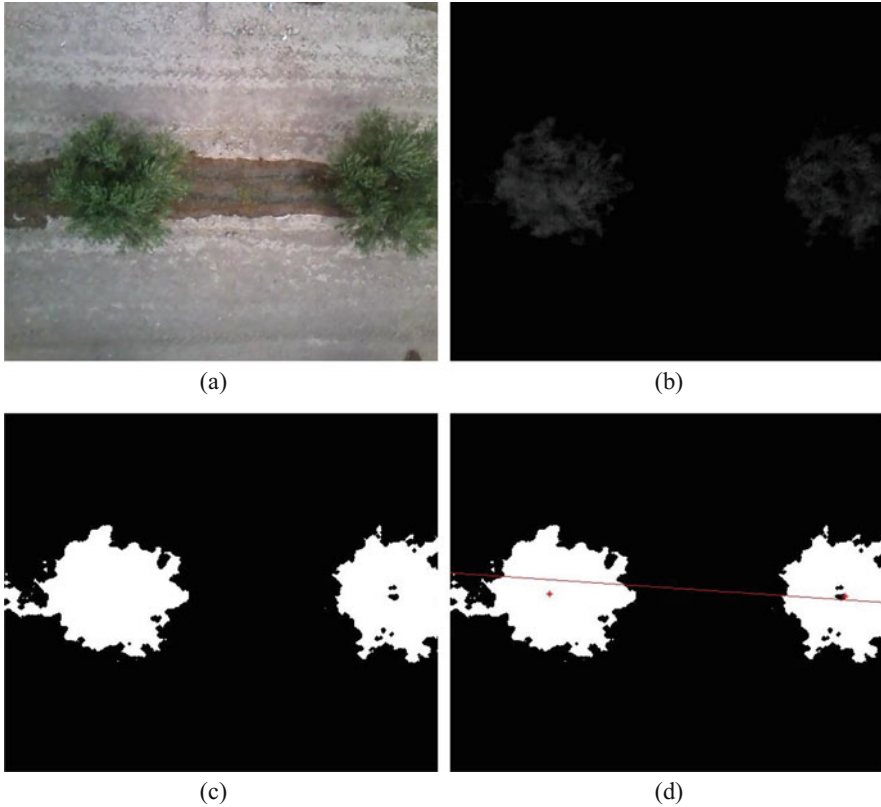


Fig. 18.3 Results of the stages of image processing. (a) Real image. (b) Almost binary image. (c) Binary image. (d) Centroids and estimated line

$$\mathbf{L}_x = \mathbf{P}_1\mathbf{P}_2^T - \mathbf{P}_2\mathbf{P}_1^T = (x_1 - x_2) \begin{bmatrix} 0 & y & 0 & 1 \\ -y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & y & 0 & 1 \end{bmatrix}, \quad (18.9)$$

where $\mathbf{P}_1 = [x_1, y, 0, 1]^T$ y $\mathbf{P}_2 = [x_2, y, 0, 1]^T$ are the homogeneous coordinates of any two points on the line L_x , and the coordinate $y = 0$ to consider the line $y^w = 0$. Whereas the center of the frame $\langle c \rangle$ matches the center of the UAV frame, since a kinematic calibration has been carried out between both frames, and negligible values have been obtained in relation to the height at which the UAV navigates. The camera matrix $\mathbf{K}_{\text{cam}} = \mathbf{K}\mathbf{H}_0\mathbf{g}_w^c$, with normalized image coordinates in the unknown depth and then in metric coordinates, is given by the product of the matrices of intrinsic parameters of the camera \mathbf{K} , of perspective projection \mathbf{H}_0 and homogeneous transformation \mathbf{g}_w^c of the frame $\langle w \rangle$ to the frame $\langle c \rangle$ as

$$\mathbf{K}_{\text{cam}} = f \begin{bmatrix} s_\psi & -c_\psi & 0 & c_\psi y_{ob} - s_\psi x_{ob} \\ -c_\psi & -s_\psi & 0 & c_\psi x_{ob} + s_\psi y_{ob} \\ 0 & 0 & -\frac{1}{f} & \frac{z_{ob}}{f} \end{bmatrix}, \tag{18.10}$$

where f represents the focal length of the camera and x_{ob} , y_{ob} , and z_{ob} represent the $x - y - z$ coordinates of the frame center $\langle b \rangle$ expressed in the frame $\langle w \rangle$.

Using Eq. (18.10), the line L_x is mapped to the image plane on the line

$$\mathbf{l} = \mathbf{K}_{\text{cam}} \mathbf{L}_x \mathbf{K}_{\text{cam}}^T. \tag{18.11}$$

The Cartesian expression ($\mathbf{l}: ax + by + c = 0$) of the line \mathbf{l} in the image plane is given by the 3×1 matrix $\mathbf{l}_{\text{cartesian}} = [a, b, c]^T$

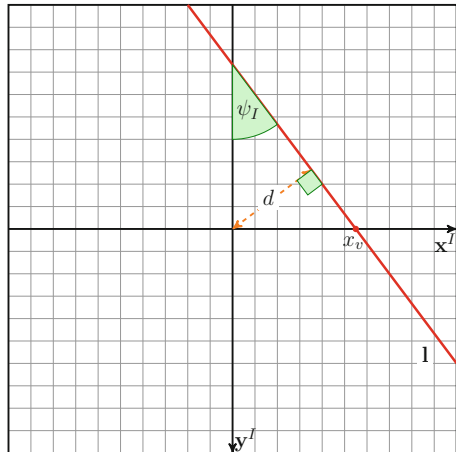
$$\mathbf{l}_{\text{cartesian}} = f(x_1 - x_2) \left[c_\psi z_{ob}, s_\psi z_{ob}, -f(y_{ob} - y) \right]^T. \tag{18.12}$$

Using the expression (18.12) and the transformations between the coordinate frames, the following vector of image characteristics is proposed

$$\mathbf{x}^I = \begin{bmatrix} d \\ \psi_I \end{bmatrix} = \begin{bmatrix} x_v c_\psi \\ \psi \end{bmatrix} = \begin{bmatrix} \frac{f y_{ob}}{z_{ob}} \\ \psi \end{bmatrix}, \tag{18.13}$$

where d is the distance from the line \mathbf{l} to the origin of the image plane and ψ_I is the angle between the line and the \mathbf{y}^I axis of the plane. Note that $\psi_I = \psi$ and d is a signed distance when it is calculated as $d = x_v c_\psi$ to determine which side of the real line in 3D space is the quadcopter, where x_v is the intersection of the line \mathbf{l} with the \mathbf{x}^I axis of the image plane, see Fig. 18.4.

Fig. 18.4 Line and features in the image plane



Considering that the height (z_{ob}) of the UAV is kept constant due to the action of an independent controller ($\dot{z}_{ob} \cong 0$), the temporal derivative of the vector \mathbf{x}^I is given by

$$\dot{\mathbf{x}}^I = \begin{bmatrix} \dot{d} \\ \dot{\psi}_I \end{bmatrix} = \begin{bmatrix} \frac{f}{z_{ob}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y}_{ob} \\ \dot{\psi} \end{bmatrix} = \mathbf{J}\dot{\mathbf{x}}_T^w, \tag{18.14}$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{f}{z_{ob}} & 0 \\ 0 & 1 \end{bmatrix}, \tag{18.15}$$

represents the Jacobian image matrix and $\dot{\mathbf{x}}_T^w = [\dot{y}_{ob}, \dot{\psi}]^T$ represents the truncated speed vector of the UAV in $\langle w \rangle$.

18.4 Kinematic Visual Servoing Controllers

This section presents three different visual servoing controllers, first one is a position based controller while the other two are image based controllers. It is important to remark that the design of the last proposal is based on the passivity properties of the system [17].

18.4.1 Position Based Controller

For the design of this controller, control errors are defined as: $\tilde{\mathbf{x}}^w = \mathbf{x}_d^w - \mathbf{x}^w = [\tilde{x}^w, \tilde{y}^w, \tilde{z}^w, \tilde{\psi}^w]^T = [x_d^w - x^w, y_d^w - y^w, z_d^w - z^w, \psi_d^w - \psi^w]^T$, where the component \tilde{x}^w of the vector $\tilde{\mathbf{x}}^w$ is zero because it is a path controller, see Fig. 18.5.

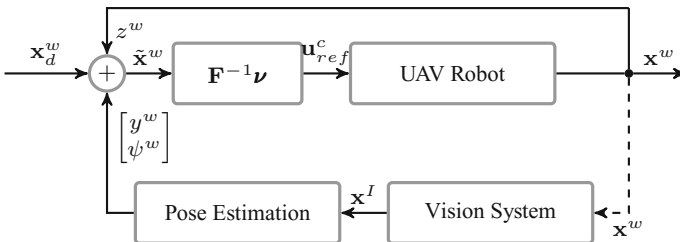


Fig. 18.5 Position based visual servoing control

To satisfy the control objective $\tilde{\mathbf{x}}^w(t) = [0, \tilde{y}^w, \tilde{z}^w, \tilde{\psi}^w]^T \rightarrow 0$, when $t \rightarrow \infty$; the following servo-visual position based control law is proposed

$$\mathbf{u}_{\text{ref}}^c = \mathbf{F}^{-1}[\dot{\mathbf{x}}_d^w + f(\tilde{\mathbf{x}}^w)] = \mathbf{F}^{-1}\mathbf{v}, \quad (18.16)$$

where $\dot{\mathbf{x}}_d^w = [\dot{x}_{\text{max}}^w c_{\tilde{\psi}^w} / (1 + |\tilde{y}^w|), 0, 0, 0]^T$ is the desired speed to follow the path bounded by the control errors to the maximum value \dot{x}_{max}^w along the line, $c_{\tilde{\psi}^w} \doteq \cos(\tilde{\psi}^w)$ and $f(*)$ is a function of the $\tilde{\mathbf{x}}^w$ control errors. Then, the auxiliary control function \mathbf{v} has the form

$$\mathbf{v} = \begin{bmatrix} \frac{\dot{x}_{\text{max}}^w c_{\tilde{\psi}^w}}{1 + |\tilde{y}^w|} \\ K_y \tanh(\tilde{y}^w) \\ K_z \tanh(\tilde{z}^w) \\ K_\psi \tanh(\tilde{\psi}^w) \end{bmatrix}, \quad (18.17)$$

where $K_* > 0$ are the gains of the controller.

18.4.1.1 Controller Analysis

Considering a perfect tracking of speed, that is $\dot{\mathbf{x}}^w = \mathbf{F}\mathbf{u}_{\text{ref}}^c$, the closed loop equation of the system results

$$\dot{\mathbf{x}}^w = \mathbf{F}(\mathbf{F}^{-1}\mathbf{v}) = \mathbf{v}, \quad (18.18)$$

i.e.,

$$\dot{\tilde{\mathbf{x}}}^w - f(\tilde{\mathbf{x}}^w) = \mathbf{0}. \quad (18.19)$$

Considering a perfect tracking of speed, that is $\tilde{\mathbf{x}}_t^w = [\tilde{y}^w, \tilde{z}^w, \tilde{\psi}^w]^T$, the candidate function Lyapunov's is defined

$$V = \frac{1}{2} \tilde{\mathbf{x}}_t^w T \tilde{\mathbf{x}}_t^w = \frac{1}{2} (\tilde{y}^w{}^2 + \tilde{z}^w{}^2 + \tilde{\psi}^w{}^2). \quad (18.20)$$

whose time derivative is given by

$$\begin{aligned} \dot{V} &= -\tilde{y}^w K_y \tanh(\tilde{y}^w) - \tilde{z}^w K_z \tanh(\tilde{z}^w) - \\ &\quad - \tilde{\psi}^w K_\psi \tanh(\tilde{\psi}^w). \end{aligned} \quad (18.21)$$

The function \dot{V} is negative definite because it is a sum of terms less than or equal to zero. It is then concluded that $\tilde{\mathbf{x}}_t^w(t) \rightarrow \mathbf{0}$ when $t \rightarrow \infty$; therefore, the equilibrium point $\tilde{\mathbf{x}}_t^w = \mathbf{0}$ is asymptotically stable.

18.4.2 Image Based Controller

For the design of the controller, the image feature vector \mathbf{x}^I defined in (18.13) is used and the control errors are defined as $\tilde{\mathbf{x}}^I = \mathbf{x}_d^I - \mathbf{x}^I = [\tilde{d}, \tilde{\psi}_I]^T = [-d, -\psi_I]^T$, the reference vector of image characteristics $\mathbf{x}_d^I = \mathbf{0}$ since the task defined for the UAV is to navigate on the line with the same orientation of it, see Fig. 18.6.

The following control law is then proposed

$$\dot{\mathbf{x}}_{T_{\text{ref}}}^w = \mathbf{J}^{-1}[\dot{\mathbf{x}}_d^I + \mathbf{K}_I \tilde{\mathbf{x}}^I] = \mathbf{J}^{-1} \mathbf{K}_I \tilde{\mathbf{x}}^I, \tag{18.22}$$

where $\mathbf{K}_I = \text{diag}(k_d, k_\psi)$ is a positive definite gain matrix, $\dot{\mathbf{x}}_d^I = \mathbf{0}$ is the temporal derivative of the vector \mathbf{x}_d^I , and \mathbf{J} is the Jacobian image matrix defined in (18.15).

The control actions of the x and z axes of the UAV are proposed as:

$$\dot{x}_{\text{oref}}^w = \frac{\dot{x}_{\text{max}}^w c \tilde{\psi}_I}{1 + |\tilde{x}_v|} + K_y \tilde{y}_I, \tag{18.23}$$

$$\dot{z}_{\text{oref}}^w = K_z \tanh(\tilde{z}^w); \tag{18.24}$$

where $\tilde{x}_v = 0 - x_v$ is the error of intersection of the line \mathbf{l} with the \mathbf{x}^I axis of the image plane, $\tilde{z}^w = z_d^w - z^w$ is a height error of the UAV and the term $K_y \tilde{y}_I = 0$ since it is a path controller.

Grouping Eqs. (18.22) and (18.23) the final control action sent to the UAV is given by

$$\mathbf{u}_{\text{ref}}^c = \mathbf{F}^{-1} \mathbf{v}, \tag{18.25}$$

where the auxiliary control function \mathbf{v} has the form

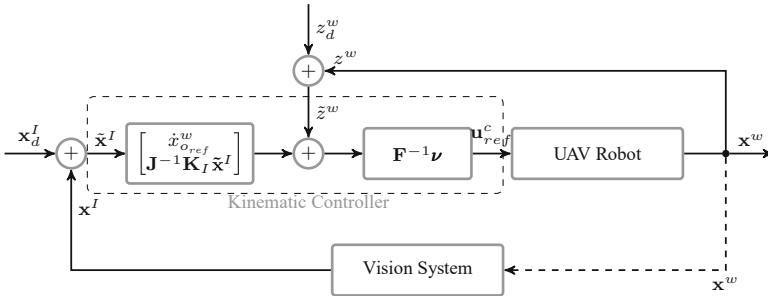


Fig. 18.6 Image based visual servoing control

$$\mathbf{v} = \begin{bmatrix} \frac{\dot{x}_{\max}^w c_{\tilde{\psi}^I}}{1+|\tilde{x}_v|} \\ \frac{z_{ob}^w}{f} k_d \tilde{d} \\ K_z \tanh(\tilde{z}^w) \\ k_{\psi} \tilde{\psi}^I \end{bmatrix}. \tag{18.26}$$

Following the procedure similar to that performed with the position-based controller, it can be proved that the image errors converge to zero, and consequently the position errors of the UAV.

18.4.3 Passivity Based Controller

In this section, the control system of Fig. 18.7 is proposed to consider the problem of regulation in the image plane, where the control errors are defined in the same way as for the image based controller of Sect. 18.4.2, i.e. $\tilde{\mathbf{x}}^I = \mathbf{x}_d^I - \mathbf{x}^I$ with $\mathbf{x}_d^I = \mathbf{0}$.

By defining $\mathbf{v}_{\tilde{\mathbf{x}}^I} = \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I) \tilde{\mathbf{x}}^I$ it can be shown that the mapping $\dot{\mathbf{x}}_T^w \rightarrow \mathbf{v}_{\tilde{\mathbf{x}}^I}$ is passive by designing the positive definite gain matrix $\mathbf{K}(\tilde{\mathbf{x}}^I)$ such that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_\infty$ for any value of the image characteristics, see Appendix.

In this way, the following control law is proposed to follow a straight path

$$\dot{\mathbf{x}}_{\text{ref}}^w = -\mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} = -\mathbf{K}_c \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I) \tilde{\mathbf{x}}^I, \tag{18.27}$$

with $\mathbf{K}_c > \mathbf{0}$.

In addition, assuming for the moment a perfect speed tracking ($\dot{\mathbf{x}}_T^w = \dot{\mathbf{x}}_{\text{ref}}^w$), it can be shown that the mapping $\mathbf{v}_{\tilde{\mathbf{x}}^I} \rightarrow -\dot{\mathbf{x}}_T^w$ (the controller) is strictly input passive, see Appendix.

Then, the proposed control system is formed by the interconnection of passive systems as shown in Fig. 18.7.

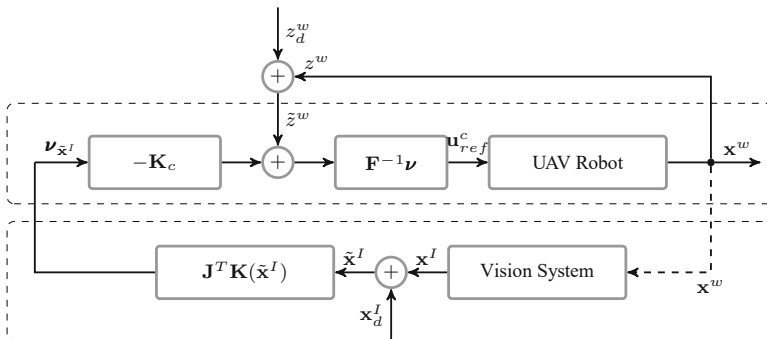


Fig. 18.7 Passivity based visual servoing control

18.4.3.1 Controller Analysis

Adding Eqs. (18.53) and (18.55) of Appendix

$$\mathbf{0} \geq -V(0) + \lambda_{\min}(\mathbf{K}_c) \int_0^T \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|^2 dt, \quad (18.28)$$

i.e.,

$$\int_0^T \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|^2 dt \leq \frac{V(0)}{\lambda_{\min}(\mathbf{K}_c)}. \quad (18.29)$$

which implies that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_{2e}$. Remembering also that $\mathbf{K}(\tilde{\mathbf{x}}^I)$ is designed such that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_\infty$, and that $\dot{\mathbf{v}}_{\tilde{\mathbf{x}}^I} \in L_\infty$ because the speeds of the robot are bounded (by the definition of the controller), it can be concluded by Barbalat's Lemma [20] that

$$\mathbf{v}_{\tilde{\mathbf{x}}^I}(t) \rightarrow \mathbf{0} \quad \text{with} \quad t \rightarrow \infty. \quad (18.30)$$

Now, assuming that the line describing the path is outside any singularity and remembering that $\mathbf{v}_{\tilde{\mathbf{x}}^I} = \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I) \tilde{\mathbf{x}}^I$, the previous condition implies that

$$\tilde{\mathbf{x}}^I(t) \rightarrow \mathbf{0} \quad \text{with} \quad t \rightarrow \infty. \quad (18.31)$$

In this way, the objective of control in the plane of the image is achieved.

18.5 Compensation of UAV Dynamics

Ignoring the assumption of perfect speed tracking, a speed controller was designed considering the simplified dynamics of the UAV and it was included in each of the proposed control systems, see Fig. 18.8. This dynamic controller causes the aerial robot to reach the reference speed calculated by the kinematic controller with good performance, which is important to improve the whole performance of the control system.

The proposed control law is given by

$$\mathbf{u}_r^b = \mathbf{H}(\ddot{\mathbf{x}}_{\text{ref}}^c + \mathbf{K}_d \dot{\tilde{\mathbf{x}}}) + \mathbf{C} \dot{\tilde{\mathbf{x}}}, \quad (18.32)$$

where $\ddot{\mathbf{x}}_{\text{ref}}^c = \dot{\mathbf{u}}_{\text{ref}}^c$, $\dot{\tilde{\mathbf{x}}} = \mathbf{u}_{\text{ref}}^c - \dot{\mathbf{x}}^b$ is the speed error, $\mathbf{u}_{\text{ref}}^c$ is the control action generated by any of the kinematic controllers of the previous sections, and $\dot{\mathbf{u}}_{\text{ref}}^c$ their temporal derivative; \mathbf{H} and \mathbf{C} are the matrices of the dynamic model defined in (18.8) and \mathbf{K}_d is a gain positive definite symmetric matrix.

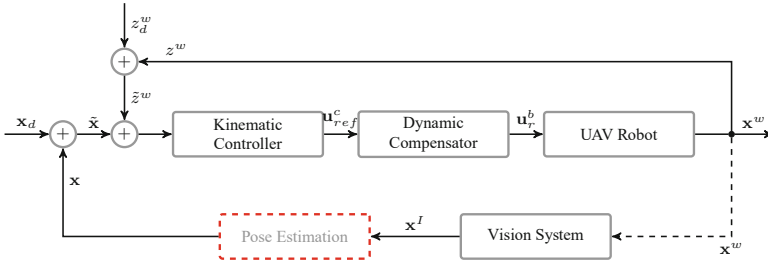


Fig. 18.8 Visual servoing control with dynamic compensation

18.5.1 Controller Analysis

Replacing u_r^b in the dynamic model yields the closed loop equation of the system

$$\ddot{\tilde{\mathbf{x}}} + \mathbf{K}_d \dot{\tilde{\mathbf{x}}} = \mathbf{0}, \tag{18.33}$$

Since \mathbf{K}_d is positive definite symmetric, it is concluded that

$$\dot{\tilde{\mathbf{x}}}(t) \rightarrow \mathbf{0} \quad \text{with} \quad t \rightarrow \infty. \tag{18.34}$$

Now, consider the following positive definite function

$$V = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \mathbf{H} \dot{\tilde{\mathbf{x}}}, \tag{18.35}$$

and its time derivative in the trajectories of the system

$$\dot{V} = \dot{\tilde{\mathbf{x}}}^T \mathbf{H} \ddot{\tilde{\mathbf{x}}} = -\dot{\tilde{\mathbf{x}}}^T \mathbf{H} \mathbf{K}_d \dot{\tilde{\mathbf{x}}} = -\dot{\tilde{\mathbf{x}}}^T \mathbf{K}'_d \dot{\tilde{\mathbf{x}}}, \tag{18.36}$$

the term $-\dot{\tilde{\mathbf{x}}}^T \mathbf{K}'_d \dot{\tilde{\mathbf{x}}}$ is negative definite, therefore $\dot{V} < 0$. Then, it is concluded that $\dot{\tilde{\mathbf{x}}} \in L_\infty$.

On the other hand, integrating \dot{V} over $[0, T]$

$$V(T) - V(0) = - \int_0^T \dot{\tilde{\mathbf{x}}}^T \mathbf{K}'_d \dot{\tilde{\mathbf{x}}} dt, \tag{18.37}$$

i.e.,

$$- V(0) \leq -\lambda_{\min}(\mathbf{K}'_d) \int_0^T \dot{\tilde{\mathbf{x}}}^T \dot{\tilde{\mathbf{x}}} dt, \tag{18.38}$$

Or

$$\int_0^T \|\dot{\tilde{\mathbf{x}}}\|^2 dt \leq \frac{V(0)}{\lambda_{\min}(\mathbf{K}'_d)} \quad \forall T \in [0, \infty). \quad (18.39)$$

Then, it is also concluded that $\dot{\tilde{\mathbf{x}}} \in L_2$. After proving that $\dot{\tilde{\mathbf{x}}} \in (L_2 \cap L_\infty)$, it can be noticed that $\dot{\tilde{\mathbf{x}}}^w = \dot{\mathbf{x}}_{\text{ref}}^w - \dot{\mathbf{x}}_T^w = \mathbf{F}\dot{\tilde{\mathbf{x}}}$ also $\in (L_2 \cap L_\infty)$ because \mathbf{F} is a homogenous transformation matrix.

Now, eliminating the hypothesis of perfect speed tracking, that is, considering a speed error different from zero $\dot{\tilde{\mathbf{x}}} = \mathbf{u}_{\text{ref}}^c - \dot{\mathbf{x}}^b$, the control error $\tilde{\mathbf{x}}^I(t)$ must be analyzed. This speed error and the kinematic controller are introduced in

$$\begin{aligned} -\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}_T^w dt &= -\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T (\dot{\mathbf{x}}_{\text{ref}}^w - \dot{\tilde{\mathbf{x}}}^w) dt = \\ &= -\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\mathbf{x}}_{\text{ref}}^w dt + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}^w dt = \\ &= -\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} dt + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}^w dt, \end{aligned} \quad (18.40)$$

adding the previous expression with

$$\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}_T^w dt \geq -V(0), \quad (18.41)$$

is obtained

$$\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{v}_{\tilde{\mathbf{x}}^I} dt \leq V(0) + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}^w dt, \quad (18.42)$$

Or

$$\lambda_{\min}(\mathbf{K}_c) \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T}^2 \leq V(0) + \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T} \|\dot{\tilde{\mathbf{x}}}^w\|_{2,T} \quad \forall T \in [0, \infty). \quad (18.43)$$

Then, remembering that $\dot{\tilde{\mathbf{x}}}^w \in L_2$, the previous inequality only remains if $\|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T} < \infty$, which implies that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_2$. This conclusion implies that the property $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_2$ remains valid after including the dynamic speed controller when the assumption of a perfect tracking speed is eliminated. Recalling also that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_\infty$ and $\dot{\tilde{\mathbf{x}}}^w \in L_\infty$, then the control objective is reached, that is,

$$\tilde{\mathbf{x}}^I(t) \rightarrow \mathbf{0} \quad \text{with} \quad t \rightarrow \infty. \quad (18.44)$$

18.6 Simulation Results

Different simulation experiments were carried out to evaluate the performance of the proposed controllers. The control schemes based on position, image, and passivity are represented in the different figures with blue, red, and black curves, respectively. The path following task starts with the UAV at rest at certain pose and it is desired to follow the crop line with a maximum speed $\dot{x}_{max}^w = 2.5$ m/s. The gains of the kinematic controllers were adjusted experimentally in such a way that the response trajectories of the UAV during the task are as fast as possible without generating saturations of the real UAV speeds (Table 18.1).

Figure 18.9 shows the trajectory of the UAV during the path tracking when a kinematic controller is used without compensating the dynamics; Figs. 18.10, 18.11, 18.12 show the controlled pose variables.

When the dynamics of the UAV are not compensated, large oscillations are generated and the task of following the line suffers large delays due to these errors, and as can be seen in Figs. 18.9, 18.10, 18.11 and 18.12, the controller based on the position is the slower of the three. These effects are accentuated by generating

Table 18.1 Gain of kinematic controllers without compensation

Position	Image	Passivity
$K_y = 0.76$	$K_I = \begin{bmatrix} 0.245 & 0 \\ 0 & 2.3 \end{bmatrix}$	$K(\bar{x}^I) = \begin{bmatrix} \frac{1.0}{0.1+ d } & 0 \\ 0 & \frac{1.0}{0.1+ \psi } \end{bmatrix}$
$K_z = 0.6$	$K_z = 0.6$	$K_c = \begin{bmatrix} 4.0 & 0 \\ 0 & 1.8 \end{bmatrix}$
$K_\psi = 2.8$		$K_z = 0.6$

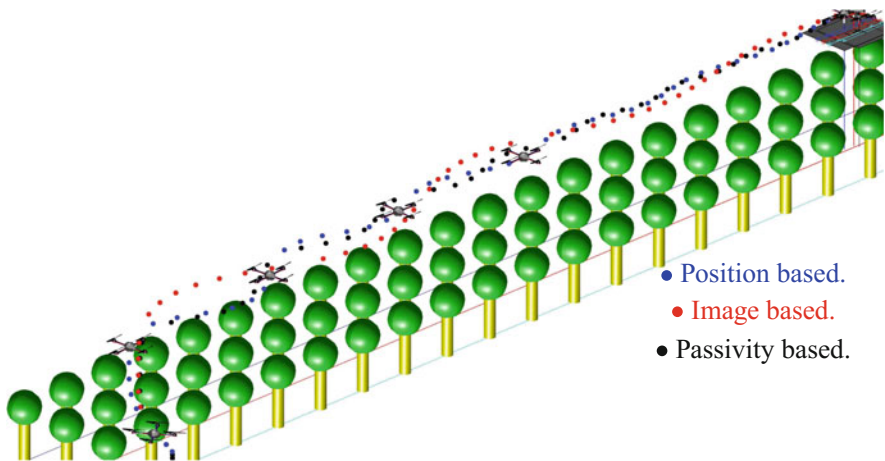


Fig. 18.9 UAV trajectory without dynamic compensation

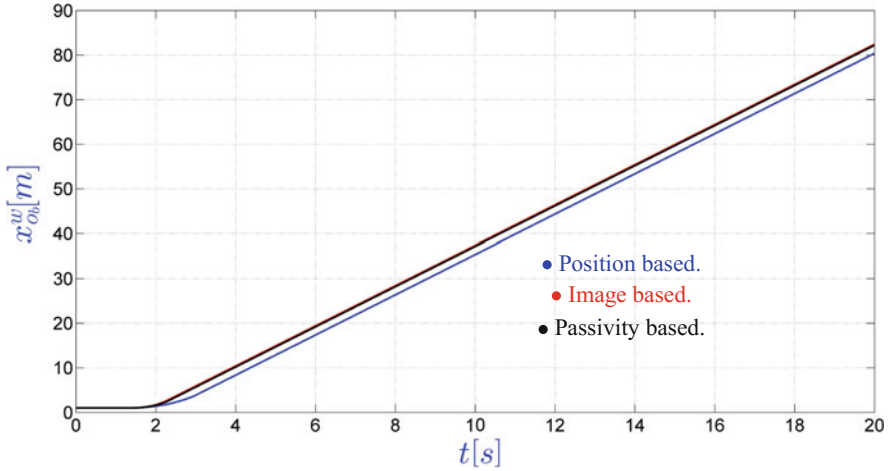


Fig. 18.10 Position x of the UAV without dynamic compensation

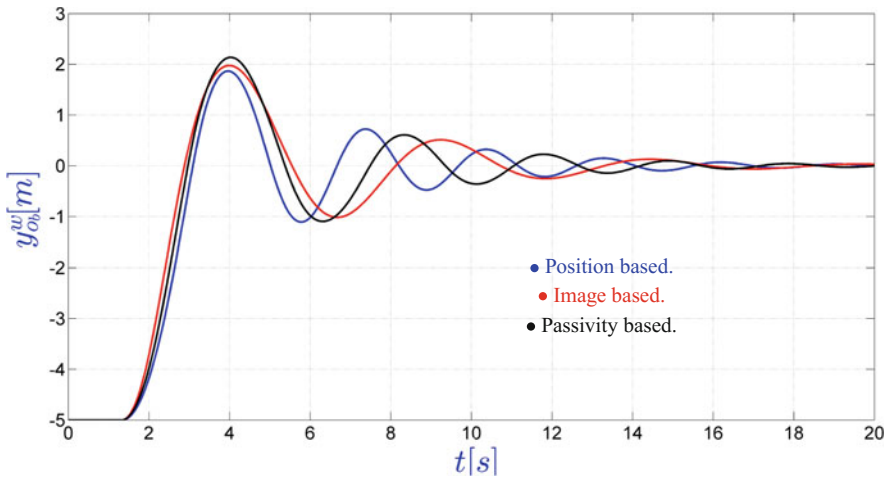


Fig. 18.11 Position y of the UAV without dynamic compensation

large control actions to try to correct the oscillations in the control errors as shown in Figs. 18.13, 18.14, and 18.15. In these figures, it should be noted that the image-based controller generates the largest control actions and the passive produces actions comparable to those of the position-based controller despite its greater response speed in the correction of the UAV pose. It is also important to emphasize that the three controllers (designed considering only the aerial robot kinematics) finally manage to reduce the control errors to values very close to zero in a real robot; showing a great robustness to satisfy, for practical purposes, the control objectives despite being used in a robot whose dynamic model differs from that used for its design.

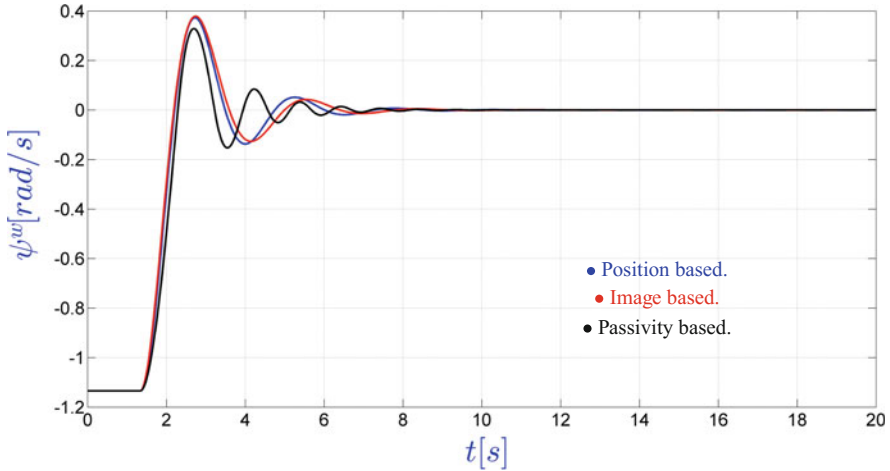


Fig. 18.12 Orientation ψ of the UAV without dynamic compensation

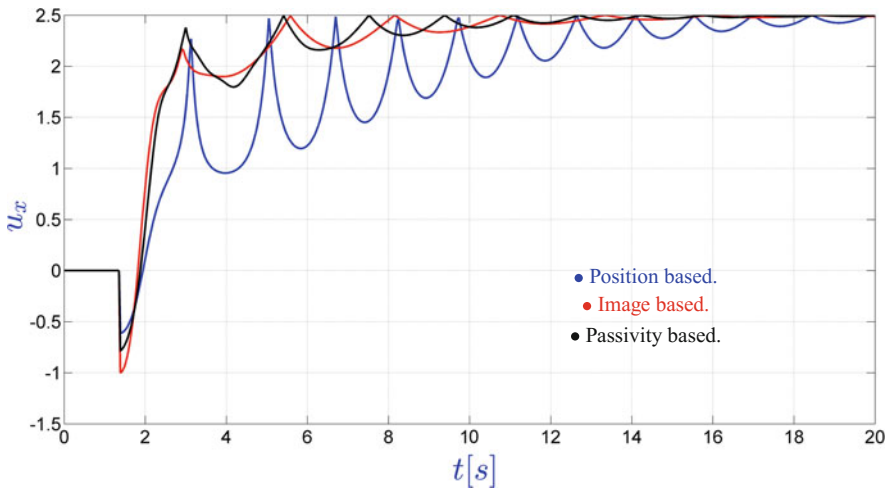


Fig. 18.13 UAV pitch command without dynamic compensation

On the other hand, when inserting the dynamic compensator in cascade with any of the kinematic controllers, faster trajectories are obtained and without oscillations as shown in Fig. 18.16, where the gains of the kinematic controllers were adjusted with the same criterion previously used and the gains of the dynamic compensators were adjusted to the value of $\mathbf{K}_d = 10\text{diag}(1, 1, 1, 1)$ (Table 18.2).

Figures 18.17, 18.18, and 18.19 show the pose evolution when the vehicle dynamics are compensated. In these figures it can be seen how the exact compensation of the dynamics of the UAV considerably improves the performance of the kinematic controllers allowing them to generate the correct control actions to achieve the control

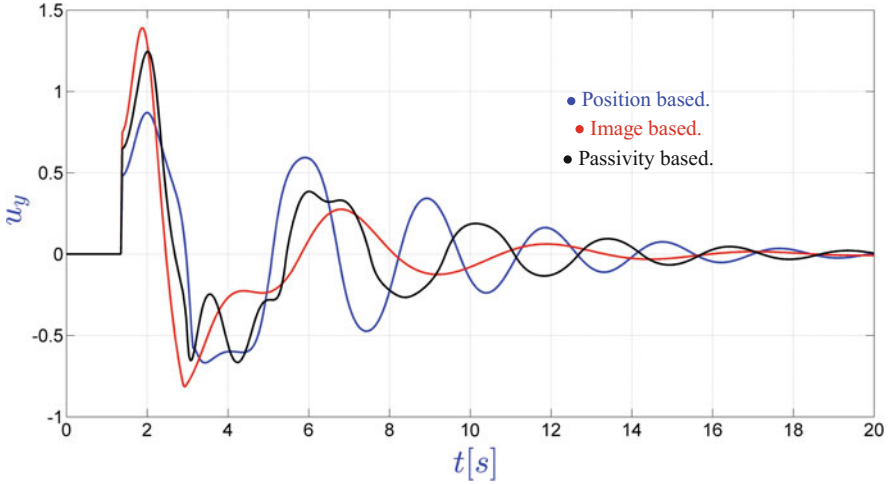


Fig. 18.14 UAV roll command without dynamic compensation

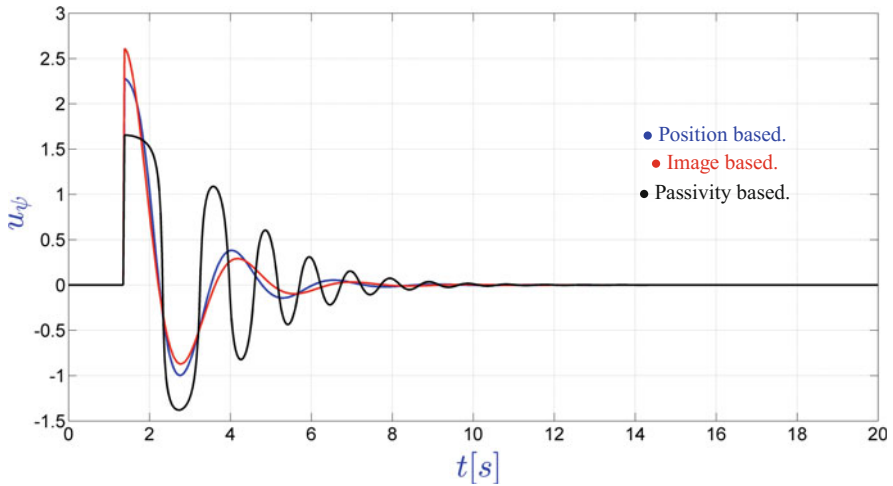


Fig. 18.15 UAV yaw command without dynamic compensation

objectives, since the system formed by the dynamic compensator and the dynamic model of the UAVs behave in front of the kinematic controller as an aerial robot with a kinematic model and its dynamics compensated.

It can further be seen that the image-based controller exhibits the slowest response in the correction of robot pose variables, while the controller designed with passivity techniques produces a higher response speed with similar control actions as compared with the ones of the position based controller, which exhibits the smallest control actions, see Figs. 18.20, 18.21, and 18.22.

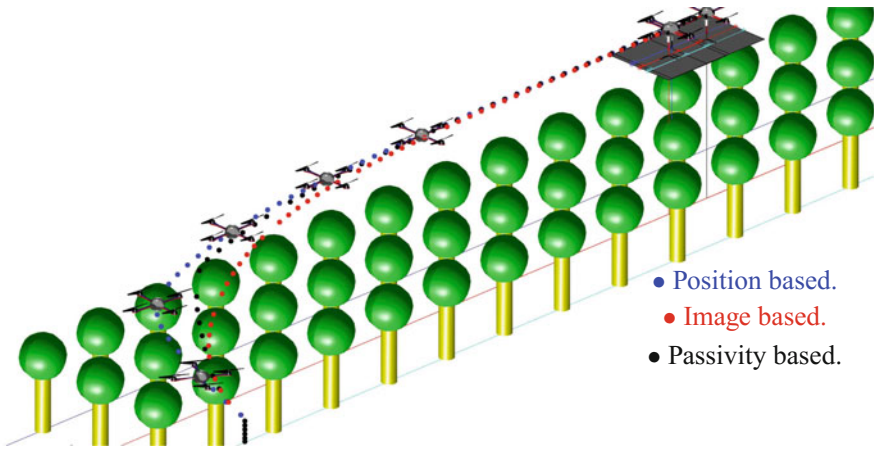


Fig. 18.16 UAV trajectory with dynamic compensation

Table 18.2 Gain of kinematic controllers with compensation

Position	Image	Passivity
$K_y = 3.35$	$K_I = \begin{bmatrix} 0.67 & 0 \\ 0 & 1.15 \end{bmatrix}$	$K(\tilde{x}^I) = \begin{bmatrix} \frac{1.27}{0.01+ \tilde{d} } & 0 \\ 0 & \frac{0.135}{0.05+ \tilde{\psi} } \end{bmatrix}$
$K_z = 1.0$	$K_z = 1.0$	$K_c = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$
$K_\psi = 1.59$		$K_z = 1.0$

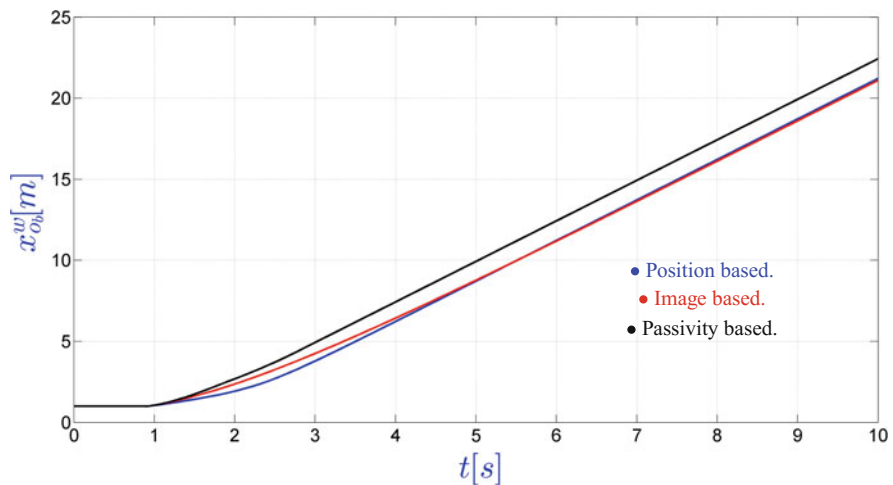


Fig. 18.17 Position x of the UAV with dynamic compensation

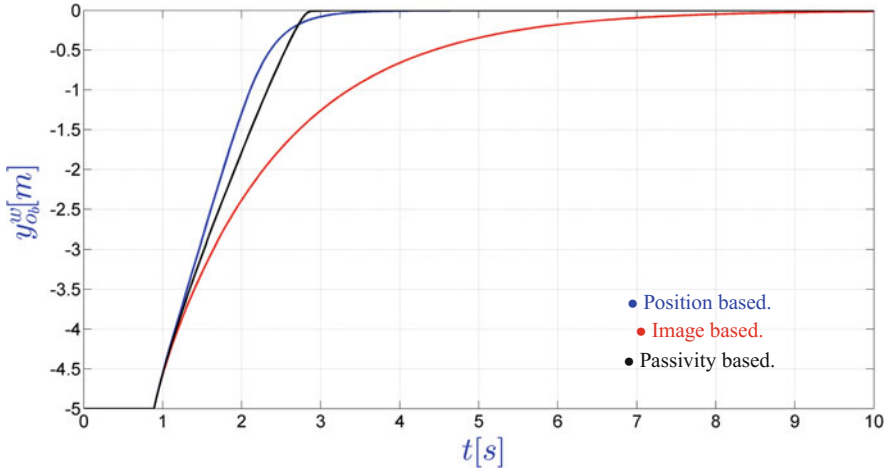


Fig. 18.18 Position y of the UAV with dynamic compensation

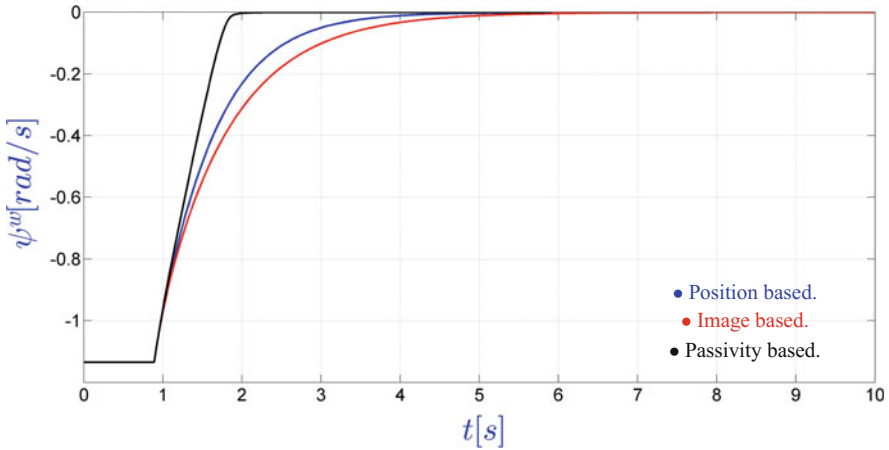


Fig. 18.19 Orientation ψ of the UAV with dynamic compensation

The observed results allow to conclude that the controller based on passivity presents a better performance during the task of tracking tree lines than the other controllers (even when the dynamics of the UAV are not compensated) when a fast correction of the vehicle pose is desired both to position and navigate crops from the moment of takeoff. Also when correcting the position with respect to plant lines in the face of environmental disturbances that directly affect the robot such as the wind or the vision system under lighting changes that can cause errors in the detection and discrimination of trees with respect to the rest of the soil. This last passivity-based driver characteristic is justified with the robustness analysis, see Appendix. Another notable feature of this design is that it presents moderate control actions,

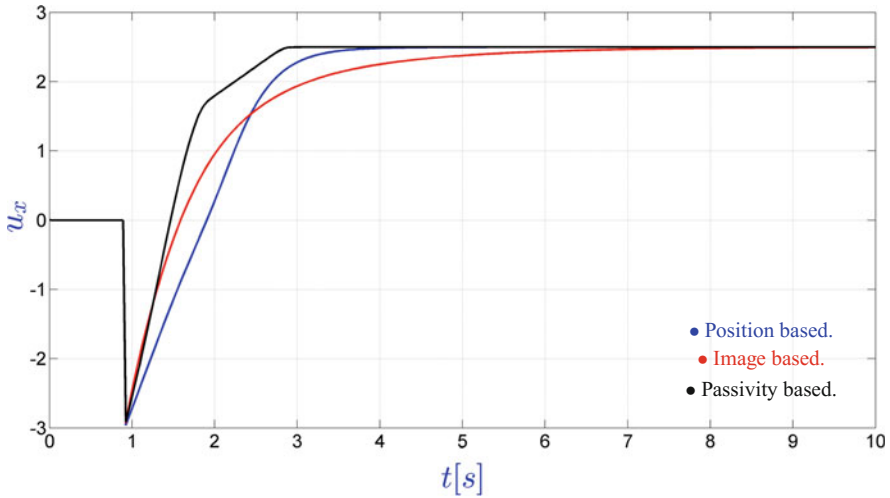


Fig. 18.20 UAV pitch command with dynamic compensation

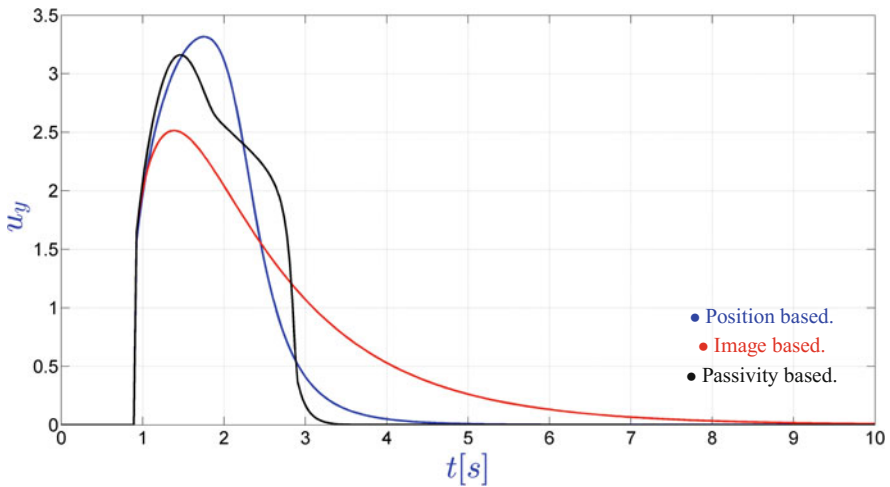


Fig. 18.21 UAV roll command with dynamic compensation

very desirable in the field of mobile robots to reduce the energy consumed by the robot during the task and achieve greater flight autonomy.

Table 18.3 shows the root mean square error (RMSE) of the pose variables and the energy ($\int_0^T u^2(t)dt$) of the control actions in the systems with compensation and without dynamic compensation. The RMSE shows that in steady state, the performance of all the schemes is very similar. However, the compensation of the dynamics decreases the energy consumption, being lower in the image based controller.

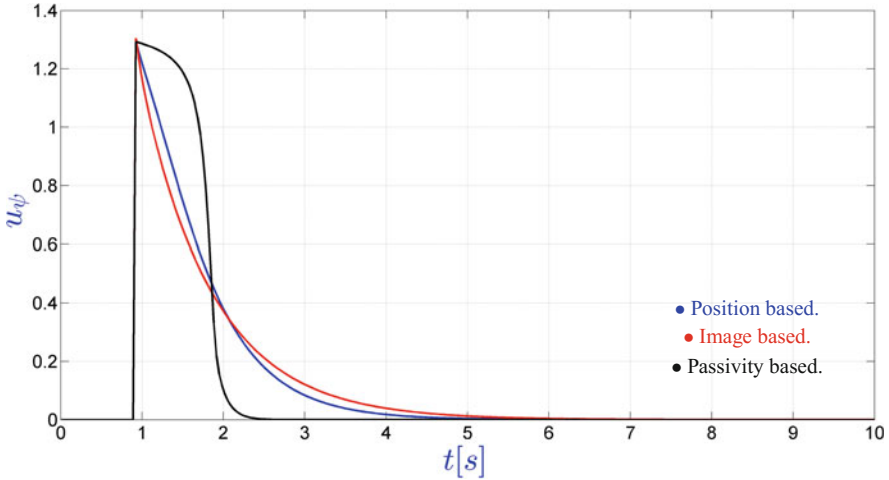


Fig. 18.22 UAV yaw command with dynamic compensation

Table 18.3 RMSE and energy

Type	Position based	Image based	Passivity based
Without compensation	RMSE _y = 1.75	RMSE _y = 1.72	RMSE _y = 1.75
	RMSE _z = 0.95	RMSE _z = 0.95	RMSE _z = 0.95
	RMSE _ψ = 0.34	RMSE _ψ = 0.34	RMSE _ψ = 0.34
	E _{u_x} = 74.89	E _{u_x} = 99.26	E _{u_x} = 99.35
	E _{u_y} = 1.99	E _{u_y} = 1.77	E _{u_y} = 1.83
	E _{u_z} = 1.42	E _{u_z} = 1.42	E _{u_z} = 1.42
With compensation	E _{u_ψ} = 3.17	E _{u_ψ} = 3.13	E _{u_ψ} = 4.47
	RMSE _y = 1.86	RMSE _y = 2.03	RMSE _y = 1.90
	RMSE _z = 1.13	RMSE _z = 1.13	RMSE _z = 1.13
	RMSE _ψ = 0.40	RMSE _ψ = 0.41	RMSE _ψ = 0.39
	E _{u_x} = 49.02	E _{u_x} = 44.65	E _{u_x} = 51.37
	E _{u_y} = 12.51	E _{u_y} = 9.03	E _{u_y} = 12.57
E _{u_z} = 2.98	E _{u_z} = 2.98	E _{u_z} = 2.98	
E _{u_ψ} = 0.85	E _{u_ψ} = 0.73	E _{u_ψ} = 1.30	

18.7 Conclusions

In this work we have presented the design of three servo-visual kinematic controllers oriented to the task of the precision agriculture field in following vegetable lines, one based on position and two based on image. The main contribution of this development was focused on the design of one of the image-based controllers, for which passivity theory concepts were used, and the demonstration of their best performance against other designs. A dynamic compensator was also designed to improve the performance of the kinematic controllers when they are used to control a real aerial robot where

the effects of its dynamics cannot be ignored. In the design of all the controllers, a stability analysis was included, showing that the control objectives are achieved. Also, a robustness analysis for the passivity-based controller is confirmed with a better performance in comparison with the other controllers when attenuating the effects of disturbances, non-modeling dynamics, and estimation errors. In addition, a description of the operation of the vision system used, both as a pose sensor and to generate the references of the designed controllers, was presented during the task of following tree lines.

The performance of the three kinematic visual servo controllers has been compared for the agricultural task of navigating and following a trajectory along plant rows compensating and without compensating the dynamics of the vehicle. The results of this comparison show clearly the superior performance of the system when the dynamic effects of the aerial robot are compensated. In addition, when the nature of the task gives priority to a fast response speed to correct the UAV pose while maintaining moderate control actions to reduce energy consumption, the results showed that the controller based on passivity has a higher performance and robustness compared to other designs based on more conventional techniques, even when the dynamics of the robot are not compensated, but with a slightly higher energy consumption. Therefore, the results developed in this work can be of interest when selecting the most appropriate controller for a specific task in the field of precision agriculture such as following vegetable lines.

Appendix

This appendix shows some passivity properties of the systems and a robustness analysis of some of the controllers.

Passive Property of the UAV Dynamic Model

Considering the following positive definite function

$$V = \frac{1}{2} \dot{\mathbf{x}}^{bT} \mathbf{H} \dot{\mathbf{x}}^b, \quad (18.45)$$

and its time derivative

$$\dot{V} = \dot{\mathbf{x}}^{bT} \mathbf{H} \dot{\mathbf{x}}^b = \dot{\mathbf{x}}^{bT} \mathbf{u}^b - \dot{\mathbf{x}}^{bT} \mathbf{C} \dot{\mathbf{x}}^b. \quad (18.46)$$

Integrating \dot{V} over the range $[0, T]$

$$V(T) - V(0) = \int_0^T \dot{\mathbf{x}}^{bT} \mathbf{u}^b dt - \int_0^T \dot{\mathbf{x}}^{bT} \mathbf{C} \dot{\mathbf{x}}^b dt. \quad (18.47)$$

Neglecting the term $V(T)$

$$-V(0) \leq \int_0^T \dot{\mathbf{x}}^{bT} \mathbf{u}^b dt - \lambda_{\min}(\mathbf{C}) \int_0^T \|\dot{\mathbf{x}}^b\|^2 dt, \quad (18.48)$$

i.e.,

$$\int_0^T \dot{\mathbf{x}}^{bT} \mathbf{u}^b dt = \langle \dot{\mathbf{x}}^b, \mathbf{u}^b \rangle \geq -V(0) + \lambda_{\min}(\mathbf{C}) \|\dot{\mathbf{x}}^b\|_{2T}^2, \quad (18.49)$$

where $\lambda_{\min}(\mathbf{C})$ represents the minimum eigenvalue of \mathbf{C} .

Then, it can be concluded that the mapping $\mathbf{u}^b \rightarrow \dot{\mathbf{x}}^b$ is strictly output passive [19].

Passive Property of the Vision System

It is shown that the passivity property of the vision system is maintained considering the positive definite function

$$V = \int_0^{\mathbf{x}^{I^T}} \boldsymbol{\eta}^T \mathbf{K}(\boldsymbol{\eta}) d\boldsymbol{\eta}, \quad (18.50)$$

where $\mathbf{K}(\tilde{\mathbf{x}}^I)$ is a positive definite symmetric gain matrix, defined to avoid saturations. Then the time derivative of the V function is

$$\dot{V} = \tilde{\mathbf{x}}^{I^T} \mathbf{K}(\tilde{\mathbf{x}}^I) \dot{\tilde{\mathbf{x}}}^I = \tilde{\mathbf{x}}^{I^T} \mathbf{K}(\tilde{\mathbf{x}}^I) (\mathbf{J} \dot{\mathbf{x}}_T^w). \quad (18.51)$$

Integrating \dot{V} over the range $[0, T]$

$$\int_0^T \dot{V} dt = \int_0^T \tilde{\mathbf{x}}^{I^T} \mathbf{K}(\tilde{\mathbf{x}}^I) (\mathbf{J} \dot{\mathbf{x}}_T^w) dt, \quad (18.52)$$

and defining $\mathbf{v}_{\tilde{\mathbf{x}}^I} = \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I) \tilde{\mathbf{x}}^I$, the following expression is obtained

$$\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\mathbf{x}}_T^w dt \geq -V(0). \quad (18.53)$$

It is then concluded that the $\dot{\mathbf{x}}_T^w \rightarrow \mathbf{v}_{\tilde{\mathbf{x}}^I}$ mapping is passive. The matrix $\mathbf{K}(\tilde{\mathbf{x}}^I)$ should be designed such that $\mathbf{v}_{\tilde{\mathbf{x}}^I} \in L_\infty$ for any value of the image characteristics.

Passive Property of the Kinematic Passivity Based Controller

Considering a perfect tracking of speed, that is $\dot{\mathbf{x}}_T^w = \dot{\mathbf{x}}_{\text{ref}}^w$ (controller based only on the vehicle's kinematics)

$$\begin{aligned} \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\mathbf{x}}_T^w dt &= - \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} dt \\ &\leq -\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{v}_{\tilde{\mathbf{x}}^I} dt, \end{aligned} \quad (18.54)$$

or,

$$\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T (-\dot{\mathbf{x}}_T^w) dt \geq \lambda_{\min}(\mathbf{K}_c) \int_0^T \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|^2 dt. \quad (18.55)$$

Then, it can be concluded that the mapping $\mathbf{v}_{\tilde{\mathbf{x}}^I} \rightarrow -\dot{\mathbf{x}}_T^w$, that is, the controller is strictly input passive [19].

Robustness Analysis of the Passivity Based Controller

The controller calculation assumes a perfect knowledge of the speed of the robot, but in practice this speed will be measured or estimated by sensors. From this estimation, the problem of analyzing the effect on the control errors immediately arises.

In this analysis, the profit performance L_2 -criterion will be used. The velocity error $\dot{\mathbf{x}}^w$ will be considered as part of an external disturbance \mathbf{w} and it will be proved that the mapping $\mathbf{w} \rightarrow \tilde{\mathbf{x}}^I$ has a finite L_2 -gain, i.e.,

$$\int_0^T \|\tilde{\mathbf{x}}^I\|^2 dt < \gamma^2 \int_0^T \|\mathbf{w}\|^2 dt \quad \forall T > 0. \quad (18.56)$$

Consider the external disturbance as $\mathbf{w} = \dot{\mathbf{x}}^w$ and assume that it is bounded. Then, considering that the speed is not perfectly known, the expression of the controller is modified as

$$\begin{aligned} \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\mathbf{x}}_T^w dt &= \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T (\dot{\mathbf{x}}_{\text{ref}}^w + \dot{\mathbf{x}}^w) dt \\ &= \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\mathbf{x}}^w dt - \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} dt, \end{aligned} \quad (18.57)$$

i.e.,

$$\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}_T^w dt = - \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} dt + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{w} dt. \quad (18.58)$$

Subtracting the previous expression from $\int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \dot{\tilde{\mathbf{x}}}_T^w dt \geq -V(0)$

$$\mathbf{0} \geq -V(0) + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{K}_c \mathbf{v}_{\tilde{\mathbf{x}}^I} dt - \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{w} dt. \quad (18.59)$$

Then,

$$\lambda_{\min}(\mathbf{K}_c) \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{v}_{\tilde{\mathbf{x}}^I} \leq V(0) + \int_0^T \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \mathbf{w} dt, \quad (18.60)$$

or defining $\epsilon = \lambda_{\min}(\mathbf{K}_c)$ and remembering the definition of product in L_{2e} ,

$$\epsilon \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T}^2 \leq \langle \mathbf{v}_{\tilde{\mathbf{x}}^I}, \mathbf{w} \rangle_T + V(0). \quad (18.61)$$

Adding to the second term of the previous expression, the positive term $\frac{1}{2} \left\langle \frac{1}{\sqrt{\epsilon}} \mathbf{w} - \sqrt{\epsilon} \mathbf{v}_{\tilde{\mathbf{x}}^I}^T, \frac{1}{\sqrt{\epsilon}} \mathbf{w} - \sqrt{\epsilon} \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \right\rangle_T$. After some algebraic operations

$$\begin{aligned} \epsilon \|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T}^2 &\leq \left\langle \mathbf{v}_{\tilde{\mathbf{x}}^I}^T, \mathbf{w} \right\rangle_T + \frac{1}{2\epsilon} \langle \mathbf{w}, \mathbf{w} \rangle_T + \frac{\epsilon}{2} \left\langle \mathbf{v}_{\tilde{\mathbf{x}}^I}^T, \mathbf{v}_{\tilde{\mathbf{x}}^I}^T \right\rangle_T \\ &\quad - \left\langle \mathbf{v}_{\tilde{\mathbf{x}}^I}^T, \mathbf{w} \right\rangle_T + V(0), \end{aligned} \quad (18.62)$$

$$\|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T}^2 \leq \frac{1}{\epsilon^2} \|\mathbf{w}\|_{2,T}^2 + V(0). \quad (18.63)$$

Now, for $\|\mathbf{w}\|_{2,T}^2$ such that $\|\mathbf{v}_{\tilde{\mathbf{x}}^I}\|_{2,T}^2$ is bounded by its saturation value; and replacing $\mathbf{v}_{\tilde{\mathbf{x}}^I} = \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I) \tilde{\mathbf{x}}^I$ in the previous expression

$$\|\tilde{\mathbf{x}}^I\|_{2,T}^2 \leq \frac{1}{\lambda_{\min}(\mathbf{M})\epsilon^2} \|\mathbf{w}\|_{2,T}^2 + V(0), \quad (18.64)$$

where $\mathbf{M} = \mathbf{K}^T(\tilde{\mathbf{x}}^I) \mathbf{J} \mathbf{J}^T \mathbf{K}(\tilde{\mathbf{x}}^I)$.

Integrating the previous expression over the interval $[0, T]$; it can be concluded that the mapping $\mathbf{w} \rightarrow \tilde{\mathbf{x}}^I$ has finite L_2 -gain smaller or equal to $\gamma = \frac{1}{\lambda_{\min}(\mathbf{M})\epsilon^2}$. That is to say, the proposed control system is robust to \mathbf{w} according to the performance L_2 -criterion (attenuation of disturbance in L_2 -gain standard or energy attenuation). In this context, the parameter γ can be considered as an indicator of the performance of the system in the presence of estimation errors.

References

1. Balamurugan, G., Valarmathi, J., & Naidu, V. P. S. (2016). Survey on UAV navigation in GPS denied environments. In: *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE5)* (pp. 198–204). Piscataway: IEEE.
2. Billingsley, J., Oetomo, D., & Reid, J. (2009). Agricultural robotics [TC spotlight]. *IEEE Robotics Automation Magazine*, 16(4), 16–16, 19. pISBN: 1070-9932
3. Brandão, A. S., Martins, F. N., & Soneguetti, H. B. (2015). A vision-based line following strategy for an autonomous UAV. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)* (Vol. 2, pp. 314–319). Piscataway: IEEE. eISBN: 978-9-8975-8149-6
4. Brandao, A. S., Sarapura, J. A., Eliete, M. D. O., Sarcinelli-Filho, M., & Carelli, R. (2010). Decentralized control of a formation involving a miniature helicopter and a team of ground robots based on artificial vision. In *2010 Latin American Robotics Symposium and Intelligent Robotic Meeting (LARS)* (pp. 126–131). Piscataway: IEEE. pISBN: 978-1-4244-8639-7
5. Burgos-Artizzu, X. P., Ribeiro, A., Tellaeche, A., Pajares, G., & Fernández-Quintanilla, C. (2010). Analysis of natural images processing for the extraction of agricultural elements. *Image and Vision Computing*, 28(1), 138–149.
6. Carelli, R., Kelly, R., Nasisi, O. H., Soria, C., & Mut, V. (2006). Control based on perspective lines of a non-holonomic mobile robot with camera-on-board. *International Journal of Control*, 79(4), 362–371.
7. Guijarro, M., Pajares, G., Riomoros, I., Herrera, P. J., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1), 75–83.
8. Hamuda, E., Glavin, M., & Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125, 184–199.
9. Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press. ISBN: 0521540518
10. Lindner, L., Sergiyenko, O., Rivas-López, M., Valdez-Salas, B., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (2016). Machine vision system for UAV navigation. In *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)* (pp. 1–6). Piscataway: IEEE.
11. Lindner, L., Sergiyenko, O., Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., et al. (2017) Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 44(4), 532–541.
12. Meyer, G. E., & Camargo Neto, J. (2008). Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture*, 63(2), 282–293.
13. Montalvo, M., Pajares, G., Guerrero, J. M., Romeo, J., Guijarro, M., Ribeiro, A., et al. (2012). Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Systems with Applications*, 39(15), 11889–11897.
14. Moorthy, S., Boigelot, B., & Mercatoris, B. C. N. (2015). Effective segmentation of green vegetation for resource-constrained real-time applications. In *Precision agriculture* (pp. 93–98). Wageningen: Wageningen Academic Publishers.
15. Santana, L. V., Brandão, A. S., Sarcinelli-Filho, M., & Carelli, R. (2014). A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 756–767). Piscataway: IEEE. eISBN: 978-1-4799-2376-2
16. Santos, M., Rosales, C., Sarapura, J., Sarcinelli-Filho, M., & Carelli, R. (2018). Adaptive dynamic control for trajectory tracking with a quadrotor. *Journal of Intelligent and Robotic Systems*, 1(10846), 1–12. ISSN: 0921-0296.

17. Sarapura JA, Roberti F, Rosales C, & Carelli R (2017) Control servovisual basado en pasividad de un uav para el seguimiento de líneas de cultivo. In *XVII Reunión de trabajo en Procesamiento de la Información y Control (RPIC), 2017* (pp. 1–6).
18. Sotomayor, J. F., Gómez, A. P., & Castillo, A. (2014). Visual control of an autonomous aerial vehicle for crop inspection. *Revista Politécnica*, 33(1), 1–8. ISSN: 1390-0129
19. Van Der Schaft, A. (2000) *L₂-Gain and passivity techniques in nonlinear control* (2nd ed.). Berlin: Springer London. ISBN: 1852330732
20. Vidyasagar, M. (1978). *Nonlinear systems analysis* (1st ed.). Englewood Cliffs, NJ: Prentice Hall. ISBN: 1852330732
21. Wu, A., Johnson, E., Kaess, M., Dellaert, F., & Chowdhary, G. (2013). Autonomous flight in GPS-denied environments using monocular vision and inertial sensors. *Journal of Aerospace Information Systems*, 10(4), 172–186. <https://doi.org/10.2514/1.I010023>

Part V
Machine Vision for Scientific, Industrial
and Civil Applications

Chapter 19

Advances in Image and Video Compression Using Wavelet Transforms and Fovea Centralis



Juan C. Galan-Hernandez, Vicente Alarcon-Aquino, Oleg Starostenko,
Juan Manuel Ramirez-Cortes, and Pilar Gomez-Gil

Acronyms

ABAC	Adaptive binary arithmetic coding
AFV-SPECK	Adaptive fovea centralis set partitioned embedded block codec
AVC	Advance video coding
AWFV-Codec	Adaptive wavelet/fovea centralis-based codec
BGR	Blue green red color space
bpp	Bits per pixel
CDF9/7	Cohen–Daubechies–Feauveau wavelet
CIF	Common intermediate format
CMYK	Cyan magenta yellow black color space
CWT	Continuous wavelet transform
dB	Decibel
DCT	Discrete cosine transform
DWT	Discrete wavelet transform
FWT	Fast wavelet transform
FVHT	Fovea centralis hierarchical trees
GIF	Graphics interchange format
HEVC	High efficiency video coding
HVS	Human visual system

J. C. Galan-Hernandez · V. Alarcon-Aquino (✉) · O. Starostenko
Department of Computing, Electronics and Mechatronics, Universidad de las Americas Puebla,
Cholula, Puebla, Mexico
e-mail: juan.galan@udlap.mx; vicente.alarcon@udlap.mx; oleg.starostenko@udlap.mx

J. M. Ramirez-Cortes · P. Gomez-Gil
Department of Electronics and Computer Science, Instituto Nacional de Astrofisica, Tonantzintla,
Puebla, Mexico
e-mail: jmram@inaoep.mx; pgomez@inaoep.mx

iDCT	Integer discrete cosine transform
iSPECK	Inverse SPECK
iLWT	Inverse LWT
JBIG	Joint bi-level image group
JPEG	Joint photographic experts group
JPEG2000	Joint photographic experts group 2000
LCL	Lossless compression limit
LIP	List of insignificant pixels
LIS	List of insignificant sets
LSP	List of significant pixels
LWT	Lifting wavelet transform
MPEG	Moving picture experts group
MSE	Mean squared error
PCX	Personal computer exchange
pixel	Picture element
PNG	Portable network graphics
ppi	Pixels per inch
PSNR	Peak signal to noise ratio
RAR	Roshal archive file format
RGB	Red green blue color space
RLE	Run length encoding
ROI	Region of interest
SPECK	Set partitioned embedded block codec
SPIHT	Set partitioning in hierarchical tree
sRGB	Standard red green blue color space
SSIM	Structural similarity index
WebP	WebP
WT	Wavelet transform
Y'CBCR	Luma chrominance color space
ZIP	.ZIP file format

19.1 Introduction

The problem of storing images appeared along with the devices that allowed to capture and represent data in the form of visual information. Devices like image scanners (1950) and graphic processing units (1984) along with graphic manipulation software made possible to capture, create, and display images as digital images on a computer. A digital image is a numeric representation of a captured or software created image. This numeric representation is a discretization value made from a digital scanner device. The digital image can be represented as a two-dimensional numeric matrix. Each element of the matrix represents a small picture element (pixel). Such images are also known as raster images. Computer software such

as Adobe Photoshop¹ or Gimp² allow to create raster images. Also, most capturing devices like cameras and image scanners capture the image as a raster image [38, 39].

Compression algorithms are encapsulated with its decompression counterpart on digital image formats. A digital image format or standard specifies the following: a compression algorithm, a decompression algorithm, which color space is used for representing the image, how the data is stored inside a binary file and headers with metadata for image software [28]. Digital image formats that use only one two-dimensional matrix are used for storing black and white images or gray scale images. Digital color images on the other hand require more than one matrix in order to represent color. Usually, the number of matrices used are three for color spaces such as Red Green Blue color space (RGB) [20], Luma Chrominance color space ($Y' C_B C_R$) [33], and derivatives or four matrix for spaces such as Cyan Magenta Yellow Black color space (CMYK) [47]. Each matrix is known as a color channel. A common practice when using integer representation is to use one matrix of elements of 32 bits. The bits of each 32-bit element are split into four sets of 8 bits. Each 8-bit set is related to one color channel. When using a three channel color space, usually the four most significant bits set is either discarded as in RGB (or Blue Green Red color space (BGR) representation [62]) or used as a *transparency* information as in the Standard Red Green Blue color space (sRGB) format [20]. There are other digital image representations of an image such as vector images [48, 57]. However, this chapter will be focused only on raster type images. From now on the term image will be used to refer to digital raster images unless otherwise stated. Usually, the quality of an image grows as the amount of pixels taken per inch grows. This is known as pixels per inch (ppi).

Lossless compression is the best way to reduce the space needed to store a high quality image. Examples of such lossless compression algorithms are the Personal Computer Exchange (PCX) file format and the Graphics Interchange Format (GIF) file format. Nevertheless, it has been shown that the upper limit for an ideal lossless compression algorithm is around 30% [28]. Therefore, image file formats based on lossless compression algorithms are less convenient as the image increases in size. In consequence, new file formats were designed that take advantage of lossy compression algorithms. Lossy compression algorithms take into account the sensibility of the human visual system (HVS) in order to drop some of the details of the image when compressing. As a result, the reconstructed image is not the original image but a close representation of it. The aim of lossy compression is to build an algorithm that when reconstructing the image using the compressed stream, the reconstructed image will look almost the same for the user. Several lossy compression algorithms for images have been proposed; however, most of them are based on mathematical transformations that take the image matrix of color intensities and map it into a different space. The most common space used is the frequency space also known as frequency domain. When using the frequency domain, a matrix

¹<http://www.adobe.com>.

²<http://www.gimp.org>.

representing the intensity of each pixel of an image is considered to be in a spatial domain.

The Joint Photographic Experts Group (JPEG) and the JPEG2000 standards are examples of image lossy compression using a transform function [1, 50, 59]. Hybrid codecs based on the discrete cosine transform (DCT) are designed to attain higher compression ratios by combining loss and lossless compression algorithms. A modified version of the DCT is used in H.264/AVC (Advanced Video Coding) and H.265/HEVC (High Efficiency Video Coding) standards [6, 60, 63]. Algorithms based on the DCT are the one used in the JPEG file format [59] and the one used in the lossy definition of the WebP file format.³ The DCT is widely used because of its low computational complexity and its high quality when used for lossy compression. However, the wavelet transform (WT) shows better image reconstruction quality when used for lossy compression [5, 30]. By using the HVS based on fovea centralis, coding the quality of the reconstruction may be improved [12, 24, 40, 41]. Nowadays, few image formats use the WT for image compression. An example of these formats is the JPEG2000 file format [1].

There are several proposals for improving classic algorithms for current wavelet-based image compression methods such as the ones proposed in [13, 25]. However, there is no ideal algorithm that produces the best image reconstruction quality for any kind of image in any given application [3]. The reason is that when doing lossy compression, the algorithm must choose which details must be dropped in order to reach a given compression ratio. The main problem lies in which details to drop. For video compression, the problem of storage increases because a digital video is a set of several images, called frames, that represent a state of a taken video at a specific instant. Also, video file formats must store the information of sound, increasing the need of efficient lossy compression algorithms for images even if the sound is also compressed. Because sound compression is a related but different problem to image compression, from now on the rest of this chapter the term video compression will be used to refer only to the compression of the visual information or frames.

19.2 Data Compression

Lossless compression algorithms exploit the statistical behavior of the data to be compressed. The original data can be recovered perfectly after decompression. Lossless compression is used for generic data compression regardless of what the data represents images, sound, text, and so on. Current formats for data file compression like .Zip file format (ZIP) [44] and Roshal Archive file format (RAR) [45] use lossless compression algorithms. There are two main classifications of lossless compression: dictionary and statistical methods. These methods require data information. Information is an intuitive concept that deals with the acquisition

³<https://developers.google.com/speed/webp/?csw=1>.

of knowledge. This acquisition can be done in several ways such as through study, experience of past events, or in the form of a collation of data [3]. Thereof, an important aspect to take into account is how to measure data. Quantifying information is based on the observation of the content of a given message and evaluating how much it is learned from a previous pool of knowledge. The value of how much information is gained depends on the context. Claude Shannon, the precursor of information theory [18], proposed a way to measure how much information is transmitted or gain after transmitting a string of characters. Given an alphabet Σ , the amount of information H (entropy) from a string s is expressed in terms of the probability of each symbol where each symbol can be seen as a value of a random variable. The amount of information indicates how easily a set of data, in this case the given string, can be compressed. The entropy is expressed as

$$H(\mathbf{s}) = - \sum_{i=1}^n P_i \log_2 P_i \quad (19.1)$$

where n is the amount of symbols in the alphabet of s calculated as $n = |\Sigma|$ and P_i is the probability of the i -th symbol.

Equation (19.1) can be interpreted as the amount of information gained from a string. This is known as the data entropy. The term entropy was coined by Claude Shannon [45]. The name entropy was chosen because the same term is used in thermodynamics to indicate the amount of disorder in a physical system. The meaning in the thermodynamics field can be related to the information theory field by expressing the information gained from a string s as the different frequencies each symbol of the alphabet appears on the string s . Using Eq. (19.1), the redundancy of R in the data is defined by the difference between the largest entropy of a symbol set and its actual entropy [44] defined by

$$R(\mathbf{s}) = \log_2 n + H(\mathbf{s}). \quad (19.2)$$

How much a data stream can be compressed is defined in terms of its redundancy R . If the stream has a redundancy $R = 0$, the data cannot be further compressed. Thus, the aim of a lossless compression algorithm is, from a given data stream with $R > 0$, to create a compressed data stream where its redundancy $R = 0$. The main theorem of Shannon of source coding states that [45] a stream of data cannot be compressed further to a limit without lossless. Such limit defined in this chapter by lossless compression limit (LCL) denoted by ρ is defined using Eq. (19.1) as follows.

$$\rho(\mathbf{s}) = mH(\mathbf{s}) \quad (19.3)$$

where m is the amount of different symbols that appear on string s .

Dictionary and statistical coding algorithms use different approach to reduce the redundancy of a data stream. Dictionary methods encode the data by choosing strings and encoding them with a *token*. Each token is stored in a dictionary and is

associated with a particular string. An example of this is to use a numerical index for each word on a dictionary. For a dictionary of length of N , it will be needed an index with a size close to $\lceil \log_2 N \rceil$ bits. Dictionary methods perform better as the size of the data stream to be compressed tends to infinity [46]. There are popular methods of dictionary source coding such as LZ77, LZ78, and LZW [43]. A common implementation of LZ77 is the DEFLATE algorithm used by the Unix operating system. The statistical methods for compression use a statistical model of the data to be compressed. It assigns a code to each of the possible symbols of the stream. The performance of the algorithms is determined by how such codes are assigned. These codes are variable in size, and usually the shortest one is assigned to the symbol with the higher frequency on the data stream. There are different variable size codes that allow to assign codes to each symbol without ambiguity. One of the most popular methods is the Huffman code. The Huffman code uses the statistical model of the data in order to assign a unique variable size code to each symbol. Huffman code is used in current standards such as JPEG and Portable Network Graphics (PNG). However, Huffman code only produces ideal size codes when the probabilities of each symbol are a negative power of two [45]. Arithmetic encoding on the other hand is known for its better performance against the Huffman codes [44].

The main disadvantage of lossless coding is that it is bounded by Shannon's theorem (see Eq. (19.3)). However, a consequence of the Shannon's theorem is that if a data stream is compressed beyond the LCL, the new data stream begins to lose information and a reconstruction of the original data cannot be made [46]. As a result, lossy algorithms must be designed in order to select which data will be lost in the compression and how to get a close representation of the original data using the compressed data stream. The data selected to be discarded is usually the one that contains the fewer information possible about the data stream. Thereof, lossy coding algorithms are designed for specific data sets in order to be able to select which data is significant and which data will be discarded. There are several ways to design lossy compression algorithms. There are lossy algorithms that operate over the original mathematical domain of the given stream such as the run-length encoding (RLE) for images. However, the best algorithms known are those that its output is calculated when using a mathematical transform.

A mathematical transform is a function that maps a set into either another set or itself. Mathematical transforms used in lossy compression, specifically on sound and image compression, are projections from one space to another. The inverse of the chosen transform must be invertible in order to reconstruct a close approximation of the original data. The use of mathematical transform for compression is also known as transform coding. Transform coding is widely used in multimedia compression and is known as perceptual coding. The preferred transformations for perceptual coding are the ones that present graceful degradation [3]. This property allows to discard some of the data on the projected space while the inverse of the transform can reconstruct a close approximation of the original data. The most common functions for perceptual coding are the ones related to the Fourier transform. When using a Fourier-related transform, it is said that the transform translates the original data from the spatial

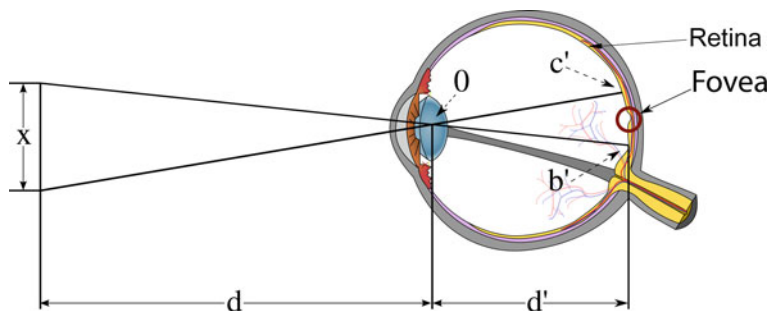


Fig. 19.1 Structure of the human eye

or time domain into the frequency domain. Being the data on the frequency domain which allows to discard some of the frequencies that are imperceptible to human perception. Hence the name of perceptual coding. Also, Fourier-related transforms degrade gracefully when decimal precision of some coefficients is lost. This allows to reduce the arithmetic precision of certain coefficients, thus reducing the number of bits required for its representation while retaining the most of the information of the original data. This process is known as *quantization*. The quantization method depends on how the data is represented by the transform on the frequency domain. There are several quantization algorithms for a given transform. The performance of the lossy compression algorithm depends on its transform and the quantization method. Several transforms have been proposed for multimedia coding such as the previously discussed DCT or the discrete WT (DWT).

Another common method used in lossy compression is the selection of regions of interest (ROIs) at different compression ratios. This feature mitigates such loss by conserving the details over a specific area. The ratio between the size of the compressed stream and the uncompressed stream is known as compression ratio [44]. In current standards such as MPEG4 and JPEG2000, ROIs can be defined [1, 19]. ROI-based algorithms are commonly used on image and video compression, and their main purpose is to assign more screen resources to a specific area [10]. ROIs are areas defined over an image selected by a given characteristic. ROI compression is when areas of the image are isolated using different desired final quality [19].

19.2.1 Fovea Centralis

The structure of the human eye (see Fig. 19.1) can also be exploited for compression. In applications wherever ROI is isolated, a selected part of the human eye called fovea centralis is utilized to increase the image quality for the human eye around ROI areas [49]. There are two main bodies on the tissue layer, particularly cones and rods. The number of cones in every eye varies between half a dozen and seven million. They are placed primarily within the central portion of the tissue layer, referred to as the fovea centralis, and are highly sensitive to color. The number of rods is

much larger, some seventy five to one hundred fifty million are distributed over the retinal surface. In Fig. 19.1, the circle between the points b' and c' marks wherever the cones reside, such area is termed fovea centralis. The larger area of distribution and the fact that several rods are connected to a single nerve reduce the amount of detail discernible by these receptors. The distance x in Fig. 19.1 is the area where the perception of an user would be the most acute, where the size of x is determined by the distance d between the observer and the image, and the distance d' between the retina and the back of the eye where the rods and cones reside. Anything outside of such area will be perceived with fewer details. This aliasing is exploited in fovea centralis compression. Fovea centralis compression can be applied over images with ROI; the use of fovea centralis around defined ROI improves the image quality for the human eye [15, 16, 24].

19.3 Wavelet Transforms

Fourier analysis is an useful tool for signal analysis. Fourier analysis is the study of general functions represented by using the Fourier transform [34]. The analysis is done by representing any periodic function as series of harmonically related sinusoids. It is useful in numerous fields, however it has some limitations [14]. Many of these limitations come from the fact that the Fourier basis elements are not localized in space. It is said that the basis of a transform is localized in space when its energy is concentrated around a given point. Accordingly, elements of the basis beyond certain radius will be 0 valued or close to 0. A basis that is not localized does not give information about how the frequency changes in relation to its position in time or space. There are refined tools that extend the capabilities of the Fourier transform in order to cover its weakness such as the windowed Fourier transform [27]. One mathematical tool that is able to analyze a signal and the structure of a signal at different sizes, thus yielding into information about the changes of frequency related to its position in time or space is the wavelet transform [2]. Time-frequency atoms are mathematical constructions that help to analyze a signal over multiple sizes. Time-frequency atoms are waveforms that are concentrated in time and frequency. The set of time-frequency atoms used for analyzing a signal is known as dictionary of atoms denoted by \mathfrak{D} . The wavelet transform builds this dictionary from a function $\psi(t) \in \mathbf{L}^2(\mathbb{R})$, where \mathbf{L}^2 is the Lebesgue space at power of 2, \mathbb{R} is the set of real numbers, and $\psi(t)$ denotes a wavelet function. ψ has several properties, it has zero average [27]

$$\int_{-\infty}^{\infty} \psi(t)dt = 0. \quad (19.4)$$

It is normalized $\|\psi\| = 1$ and centered in the neighborhood of $t = 0$. ψ is known as the *mother wavelet*. In order to create a dictionary \mathfrak{D} , ψ is scaled by ℓ and translated by u , namely [27]

$$\mathfrak{D} = \left\{ \psi_{\ell,u}(t) = \frac{1}{\sqrt{\ell}} \psi \left(\frac{t-u}{\ell} \right) \right\}_{u \in \mathbb{R}, \ell > 0} \tag{19.5}$$

The atoms remain normalized $\|\psi_{\ell,u}\| = 1$. The constant $\frac{1}{\sqrt{\ell}}$ is for energy normalization. The continuous wavelet transform (CWT) ω of $f(t) \in \mathbf{L}^2(\mathbb{R})$ at time u and scale ℓ is

$$\omega_{\ell,u}(f) = \langle f, \psi_{\ell,u} \rangle = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{\ell}} \psi^* \left(\frac{t-u}{\ell} \right) dt \tag{19.6}$$

where ψ^* is the complex conjugate of the mother wavelet ψ and $\langle \cdot, \cdot \rangle$ denotes an inner product.

Because images are two-dimensional signals, a two-dimensional wavelet transform is needed. Let $\bar{\psi}_{\ell,u}$ be

$$\bar{\psi}_{\ell,u}(t) = \frac{1}{\sqrt{\ell}} \psi^* \left(\frac{t-u}{\ell} \right), \tag{19.7}$$

extending Eq. (19.6) to two dimensions, the wavelet transform at parameters u_v, ℓ_v, u_h, ℓ_h of $f(t, x) \in \mathbf{L}^2(\mathbb{R}^2)$ yields into

$$\begin{aligned} \omega_{\ell_v, u_v, \ell_h, u_h}^2(f) &= \langle \langle f, \bar{\psi}_{\ell_v, u_v} \rangle, \bar{\psi}_{\ell_h, u_h} \rangle \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, x) \bar{\psi}_{\ell_v, u_v} \bar{\psi}_{\ell_h, u_h} dt dx. \end{aligned} \tag{19.8}$$

where ω is the wavelet operator. Also, because digital images are stored as a discrete finite signal, a discrete version of the CWT is needed. Let $f[n]$ be a discrete signal obtained from a continuous function f defined on the interval $[0, 1]$ by a low-pass filtering and uniform sampling at intervals N^{-1} . The DWT can only be calculated at scales $N^{-1} < \ell < 1$. Also, let $\psi(n)$ be a wavelet with a support included in $[-K/2, K/2]$. For $1 \leq \ell = a^j \leq NK^{-1}$, a discrete wavelet scaled by a^j is defined by [27]

$$\psi_j[n] = \frac{1}{\sqrt{a^j}} \psi \left(\frac{n}{a^j} \right). \tag{19.9}$$

The DWT is defined by a circular convolution with $\bar{\psi}_j[n]$ defined as $\bar{\psi}_j[n] = \psi_j^*[-n]$ with DWT described as

$$\omega_{a^j} f[n] = \sum_{m=0}^{N-1} f[m] \psi_j^*[m-n] = f * \bar{\psi}_j[n] \tag{19.10}$$

where $*$ is the convolution operator. Also, signal f is assumed to be periodic of length N in order to avoid border problems.

In order to speed up the computation of the wavelet coefficients, a second approach that simplifies the DWT is referred to as the *lifting scheme*. The lifting scheme [32, 51] is another way of looking at the DWT, where all the operations are performed in the time domain [1]. Computing the wavelet transform using lifting steps consists of several stages. The idea is to compute a trivial wavelet transform (the lazy wavelet) and then improve its properties by alternating the dual lifting or prediction step and the primal lifting or updating step [44]. The lazy wavelet only splits the signal into its even and odd indexed samples, namely

$$(even[n - 1], odd[n - 1]) = Split(f[n]). \quad (19.11)$$

where $f[n]$ is a given discrete signal, *even* and *odd* are the even and odd signals of the lazy wavelet, and *Split* is the split function. A dual lifting step consists of applying a filter to the even samples and subtracting the result from the odd ones. This is based on the fact that each value $f[n]_{2\ell+1}$ of the next decomposition level in the odd set is adjacent to the corresponding value $f[n]_{2\ell}$ in the even set, where ℓ is the decomposition level. Thus, the two values are correlated and any can be used to predict the other. The prediction step is given by

$$d[n - 1] = odd[n - 1] - P(even[n - 1]). \quad (19.12)$$

where d is the difference signal of the *odd* part of the lifting wavelet and the result of the prediction P operator applied to the *even* part of the lazy wavelet. A primal lifting step does the opposite: applying a filter to the odd samples and adding the result to the even samples. The update operation U follows the prediction step. It calculates the $2[n - 1]$ averages $s[n - 1]_\ell$ as the sum

$$s[n - 1] = even[n - 1] + U(d[n - 1]). \quad (19.13)$$

where U is defined by

$$s[n - 1]_\ell = f[n]_{2\ell} + \frac{d[n - 1]_\ell}{2}. \quad (19.14)$$

The process of applying the prediction and update operators is repeated as many times as needed. Each wavelet filter bank is categorized by its own U operator and the amount of rounds of the process. The calculation process of U is described in [26]. This scheme often requires far fewer computations compared to the convolution-based DWT, and its computational complexity can be reduced up to 50% [1, 11, 53]. As a result, this lifting approach has been recommended for implementation of the DWT in the JPEG2000 standard.⁴

⁴JPEG2000 draft at <http://www.jpeg.org/public/fcd15444-1.pdf>.

19.4 Image Compression

Image data compression is concerned with coding of data to minimize the number of bits used to represent an image. Current image compression standards use a combination of lossless and lossy algorithms. These can be used over the same data set because both algorithms exploit different properties of the image. On the one hand, lossless-based compression exploits data redundancy, but on the other hand, lossy-based compression exploits its transform properties and quantization. The simplest quantization equation used in image coding is defined as [4, 17]

$$C^q = \left\lfloor \frac{1}{\Delta q} C \right\rfloor \tag{19.15}$$

where $\lfloor \cdot \rfloor$ is the floor operation, $\Delta q > 1$ is known as the *quantization delta*, C is the matrix of coefficients obtained from applying a transform to the given image, and C^q is the matrix of quantized coefficients. Spatial redundancy takes a variety of different forms in an image. For example, it includes strongly correlated repeated patterns in the background of the image and correlated repeated base shapes, colors, and patterns across an image. The combination of lossy and lossless compression allows achieving lower compression ratios. Figure 19.2 shows a block diagram of the classic lossy/lossless image coding scheme [4, 17].

In Fig. 19.2, the image is interpreted as a matrix I , then the coefficients matrix C of the chosen transform is calculated. Subsequently, the quantized coefficient matrix C^q is calculated and the final lossless compressed stream S is calculated on the entropy coding block. The color space used for image compression is often the $Y' C_B C_R$ color space. This color space is chosen because it has been found that the human eye is more sensitive to changes on the luma channel (Y') than on the color channels ($C_B C_R$) [33]. This allows compressing at lower ratios the color channels than the luma channel. As a result, compression algorithms are evaluated over the luma channel only. Thereof, all the analyses of the algorithms presented in this chapter are evaluated on the luma channel. The equation used for calculating $Y' C_B C_R$ from other common color space RGB suggested in [17] is the following:

$$\begin{bmatrix} Y' \\ C_R \\ C_B \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & 0.515 & 0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{19.16}$$

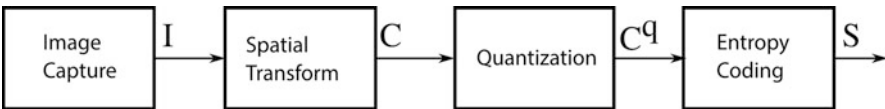


Fig. 19.2 Block diagram of the classic image coding scheme [17]

where R , G , B are the values for each channel on the RGB color space of a given pixel. From Eq. (19.16), the luma is calculated as

$$Y' = 0.299R + 0.587G + 0.114B. \quad (19.17)$$

One of the foremost image compression algorithms is the JPEG image coding standard (see Fig. 19.3). Outlined in [59], JPEG framework defines a lossy compression and a lossless compression algorithm used in tandem. The lossy compression algorithm of JPEG uses the DCT. However, in order to reduce the complexity of the algorithm [4], the image is split into non overlapping blocks of 8×8 pixels. Each block is referred to as *macroblock*. Processing macroblocks requires less computation and allows the algorithm to optimize transmission by sending the data of processed macroblocks while processing the rest of the images [19].

In Fig. 19.3, the RGB image is transformed into the $Y' C_B C_R$ color space. Then, the image is split into macroblocks. Then, the DCT macroblock applies the transform to each macroblock individually. After the transformation of a macroblock is calculated, the coefficients are quantized by a fixed ratio. JPEG standard defines a quantization matrix. Because each coefficient has a different significance on the reconstruction of the image, the quantization matrix stores a quantization ratio for each of the coefficient of a macroblock. The standard provides with values for the quantization matrix. However, some manufacturers defined its own quantization matrices in order to improve the quality of the algorithm. After quantization, the next step is resorting each macroblock into zigzag order. This allows to exploit the entropy of the lower diagonal of the macroblocks [1]. The last block of the algorithm applies lossless compression to the quantized sorted coefficients. Early versions of the algorithm define RLE and Huffman coding as the lossless algorithms for JPEG. However, the last version of JPEG [45] also includes arithmetic coding in order to reduce the compression ratio. The final overall quality of JPEG is mostly given by the

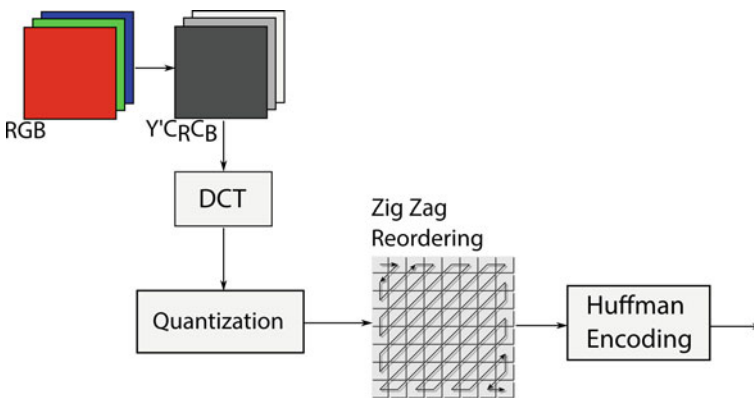


Fig. 19.3 Block diagram of the JPEG2000 standard [59]

quantization matrix, however it is not possible to precalculate the final compression ratio [43].

It is well-known that in compression applications, wavelet-based approaches outperform the block-DCT methods [22, 35]. This is due to the fact that wavelet-based approaches can reduce the blocking artifacts, provide better energy compaction because of the multi-resolution feature of wavelet basis, and have better correspondence with the HVS system [58]. Therefore, wavelet-based compression algorithms have been recommended for the JPEG2000 standard [17, 45].

19.4.1 Foveated Images

Images with a non-uniform resolution that have been used in image and video compression are known as foveated images. Equation (19.18) shows a representation of a foveated image [9].

$$I_x^0 = \int I_x c^{-1}(x) s \left(\frac{t-x}{\omega(x)} \right) dt \tag{19.18}$$

where $c(x) = \left\| s \left(\frac{-x}{\omega(x)} \right) \right\|$, I_x is the pixel at position x of a given image, $\omega(x)$ is a weight function, and I_x^0 is the foveated image. The function s is known as the weighted translation of s by x [24]. A variation of the fast wavelet transform (FWT) is reported in [7] that operates over the wavelet transform. For an image I , its foveation is given by

$$I^0 = \langle I, \Phi_{\ell_0,0,0} \rangle + \sum_{u_v, \ell_v, u_h, \ell_h} c_j^k[\ell_v, u_h] \langle I, \Psi_{\ell_v, u_h, \ell_h}^{u_v} \rangle \Psi_{\ell_v, u_h, \ell_h}^{u_v} \tag{19.19}$$

where $\Phi_{\ell_0,0,0}$ is the father wavelet, and $\Psi_{\ell_v, u_h, \ell_h}^{u_v}$ is the mother wavelet scaled and translated with $u_v = \{h, v, d\}$ and the operator $\langle \cdot, \cdot \rangle$ is the convolution operator. $c_j^k[\ell_v, u_h]$ is defined as

$$\begin{aligned} c_j^k[\ell_v, u_h] &= \langle T \Psi_{0, \ell_v, u_h}^{u_v}, \Psi_{0, \ell_v, u_h}^{u_v} \rangle \\ &= \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx \Psi_{0, \ell_v, u_h}^{u_v}(x, y) \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} ds \Psi_{0, \ell_v, u_h}^{u_v}(s, t) g_{\omega(x,y)}(s, t) \end{aligned} \tag{19.20}$$

where T is the fovea centralization operator and $g_{\omega(x,y)}(s, t)$ is the smoothing function defined as

$$g_{\omega(x,y)}(s, t) := \frac{1}{\omega(x, y)^2} g \left(\frac{s-x}{\omega(x, y)}, \frac{t-y}{\omega(x, y)} \right). \tag{19.21}$$

where the weight function $\omega(x, y)$ is defined by

$$\omega(x, y) = \alpha \| (x, y) - (\gamma_1, \gamma_2) \|_2 + \beta \tag{19.22}$$

where α is the rate, $\gamma = (\gamma_1, \gamma_2)$ is the fovea centralis, and β is the fovea centralis resolution [7].

19.5 Video Compression

Because video is just a sequence of several images called frames, video coding algorithms or video codecs use image compression extensively. To achieve high compression ratios is suitable to combine lossy and lossless compression algorithms. Classic video coding frameworks have three main algorithms (see Fig. 19.4), namely intra-frame coding (spatial transform and inverse spatial transform), inter-frame coding (motion estimation and compensation), and variable length coding (variable length coder).

In intra-frame coding, which uses the information of previous or future frames, a frame of a video stream is normally compressed using lossy algorithms. The encoder should work out the variations (prediction error) between the expected frame and the original frame. The first step in the motion compensated video coder is to create a motion compensated prediction error of the macroblocks. This calculation requires only a single frame to be stored in the receiver. Notice that for color images, motion compensation is performed only for the luma component of the image. The decimated motion vectors obtained for luma are then exploited to form motion compensated chroma components. The resulting error signal for each of the components is transformed using DCT, quantized by an adaptive quantizer, entropy

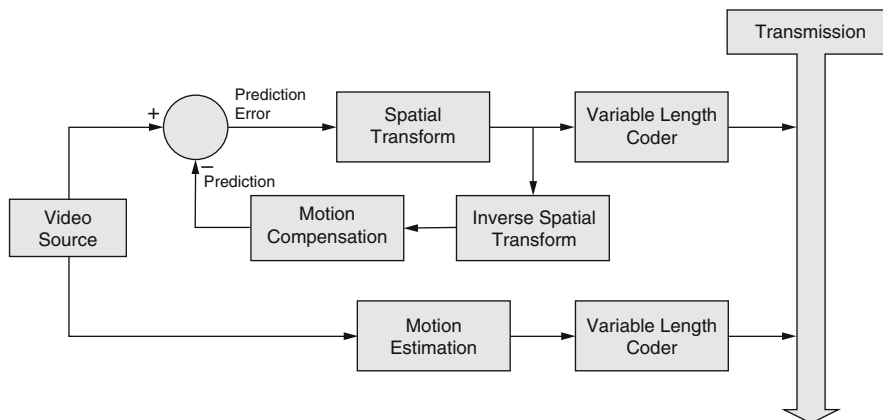


Fig. 19.4 Block diagram of the classic video coding framework [6]

encoded using a variable length coder, and buffered for transmission over a fixed-rate channel. The main problem of the block matching motion compensation is its high computational complexity.

Most video coding standards such as the H.264 [36] or the newest proposed standard H.265/HEVC codec [52] rely on the DCT for lossy intra-frame coding applied to macroblocks of a dimension of 4×4 . The smaller macroblock allows reducing artifacts on the reconstructed image [37]. However, in order to improve the speed of the algorithm, the transform used is the integer discrete cosine transform (IDCT) [8]. The IDCT is an approximation of the DCT used in JPEG standard. Instead of calculating a convolution, two different matrices are defined that are an approximation of the base of the DCT.

19.6 An Approach to Image Compression Based on ROI and Fovea Centralis

Image compression within the frequency domain based on real-valued coefficients is carried out through coefficient quantization. In this process of quantization, these coefficients become integer-valued for further compression employing either a RLE or an arithmetic encoding algorithms, which are known as variable quantization algorithms. The variable quantization algorithm exploits the fovea centralis result of the HVS based on a fovea centralis window, which is focused at a given fixation point to see a way to quantize each wavelet coefficient [15]. A modified version of the set partitioning in hierarchical tree (SPIHT) algorithm is utilized to quantize and compress these coefficients.

Figure 19.5 shows the block diagram of the compression approach based on ROIs and fovea centralis called here fovea centralis hierarchical tree (FVHT) algorithm. Assuming a video stream with frames F_i , the applied blocks can be described as follows [15]. In the *Motion estimation* block, the fovea centralis points are estimated using video frames F_i and F_{i-1} . The *ROI estimation* block outputs an array of fovea centralis points as ROI_i , where each pixel different of 0 is taken as a fovea centralis. The fovea centralis cutoff window is described in [15]. The *Lifting Wavelet Transform (LWT)* block generates the coefficients denoted as $C(\cdot)_i$ (see Sect. 19.3). The *Quantization* block maps to integers the coefficients $C(\cdot)_i$ into $C(\cdot)_i^q$ using a fixed quantization for compression. Finally, the *FVHT* block outputs a compressed stream of the quantized coefficients $C(\cdot)_i^q$ using the information of the estimated fovea centralis points ROI_i .

Note that the fovea centralis points ROI_i are input parameters to the FVHT rather than using the motion estimation block. The window parameters and the cutoff window are calculated as long as there is a fixation point [15, 24]. The reported method permits defining ROIs of variable size around the fixation point that retains the best quality. Further details on the approach described here can be found in [15].

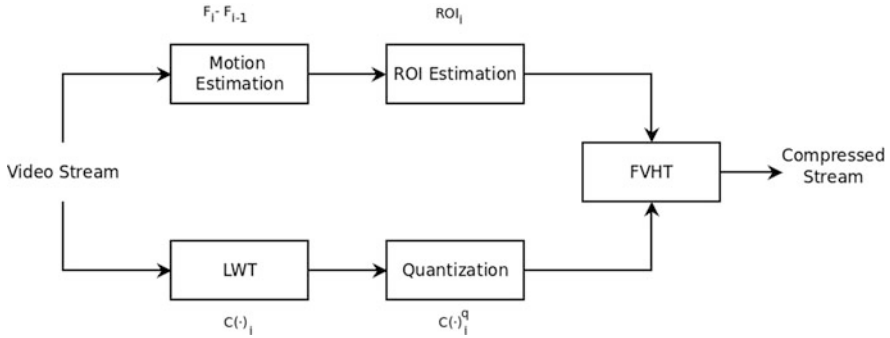


Fig. 19.5 Block diagram of compression approach based on ROI and fovea centralis [15]

19.6.1 FVHT Algorithm

The compression bit rate can be computed by assessing the decaying window function on each algorithm pass at each coefficient coordinate as it is proposed in the FVHT algorithm [15]. First, the coefficient is encoded whether the current bit rate is lower than wavelet subband, otherwise it is discarded. The sorting pass is modified in order to classify the coefficients according to its distance to the scaled fovea centralis and the cutoff window. Each time an attempt to add a coefficient to the list of significant pixels (*LSP*) is done, the assigned bits per pixel (bpp) is calculated, and the coefficient is classified. However, it should be noted that on the significance pass, the positions of the coefficients are discarded from the list of insignificant pixels (*LIP*) and on the refinement pass, they are discarded from the *LSP*. The list of insignificant sets (*LIS*) will remain the same as in the SPIHT algorithm [15, 42]. The execution time of the algorithm was analyzed using *Big O* notation, concluding that the complexity of the algorithm is linear ($\mathcal{O}(n)$) [15]. The memory usage was also analyzed, yielding a size of $\frac{71}{64}n$. The FVHT is memory intensive when compared with classic methods based on the DCT transform that can be computed using no extra storage.

19.6.2 Simulation Results

The FVHT algorithm is assessed using standard non-compressed 512×512 images. The fovea centralis is defined at the center pixel (256,256) with two parameters, namely a radius of the ROI and the power law function (the ramp function), which are defined in [15]. As stated in the JPEG2000 standard and for a fair comparison, the biorthogonal Cohen–Daubechies–Feauveau (CDF) 9/7 is considered using four levels of decomposition [1]. The reported results are compared against the SPIHT algorithm. Figures 19.6 and 19.7 show the reconstructed wavelet coefficients of the

Fig. 19.6 Reconstructed image (“cameraman”) using SPIHT algorithm at 1 bpp compression ratio [15]

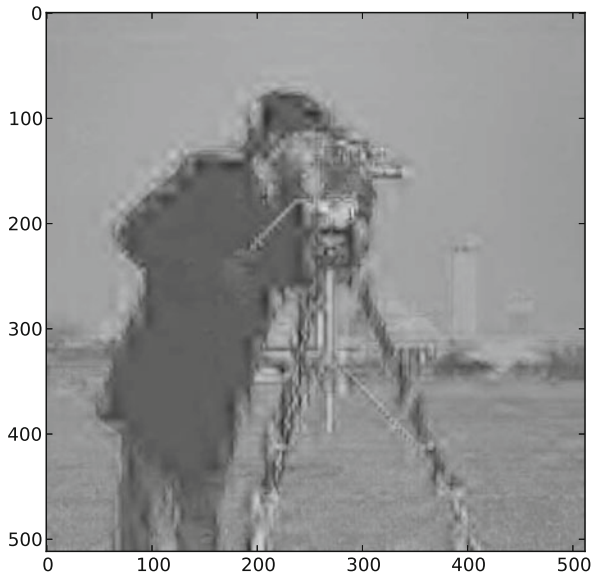
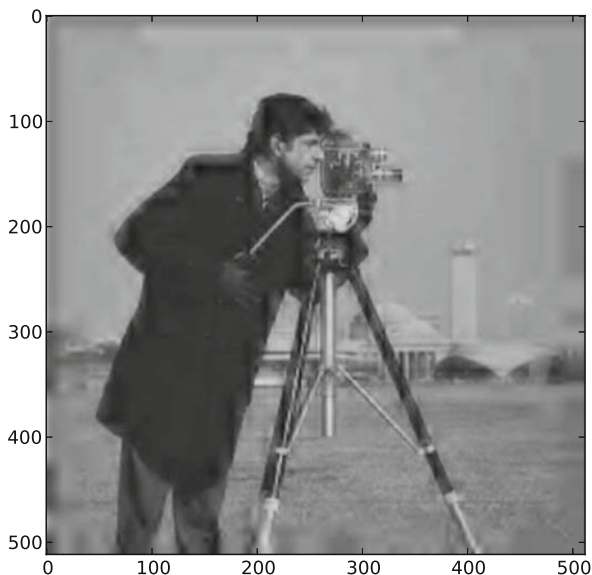


Fig. 19.7 Reconstructed image (“cameraman”) using FVHT compression algorithm at 0.06–1 bpp compression ratio [15]. Fovea centralis at (256,256)



cameraman image at 1 bit per pixel (bpp) with both SPIHT and FVHT, respectively. The same reconstructed wavelet coefficients at 1 bpp as its higher compression ratio and 0.06 bpp as its lower compression ratio are shown in Fig. 19.7. It is observed that the FVHT algorithm has better performance than SPIHT algorithm particularly over small areas around the fovea centralis or those closer to the fixation point. Further details on this approach can be found in [15].

19.7 Wavelet-Based Coding Approaches: SPECK-Based Codec and Adaptive Wavelet/Fovea Centralis-Based Codec

Two wavelet-based coding approaches based on the LWT [27] are described in this section [16]. The first called Set Partitioned Embedded bloCK (SPECK)-based codec (SP-Codec) is shown in Fig. 19.8 [31]. In the *Z-order block*, all coefficients position are organized and mapped from 2D to 1D using the Z-transform. The quantization step is carried out on *LWT and SPECK* blocks. The *adaptive binary arithmetic coding (ABAC)* block, which is a lossless compression algorithm, allows compressing a data stream while at the same time computes the statistical model (see Sect. 19.7.1) [31]. The *inverse LWT (iLWT) and inverse SPECK (iSPECK)* are applied to the compressed stream generated in the SPECK block, and finally the *motion compensation and estimation* blocks compute the motion vectors based on the block matching algorithm for each inter-frame.

The second proposal referred to as adaptive wavelet/fovea centralis-based codec (AWFV-Codec) reported in [16] aims to further increase the quality of the decoded frames (see Fig. 19.9). The reported adaptive fovea centralis-SPECK (AFV-SPECK) algorithm defines a center, a ROI area radius, and a decaying window [15, 16] and as a result various compression ratios may be considered. An external subsystem is assumed that computes the fovea centralis point of one observer, which is later provided to the AFV-SPECK coding algorithm.

19.7.1 Adaptive Binary Arithmetic Coding

The adaptive binary arithmetic coding (ABAC) is a version of the arithmetic coding algorithm applied to an alphabet with only two elements $\Sigma = 0, 1$ [64]. This application is commonly used for bi tonal images [23]. Also, it does not require a previously calculated statistical model. Each time a symbol is read, the statistical

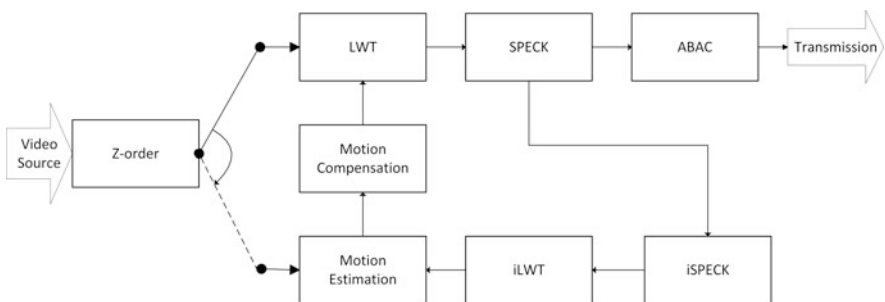


Fig. 19.8 Video coding framework SPECK-based codec (SP-Codec) [16]

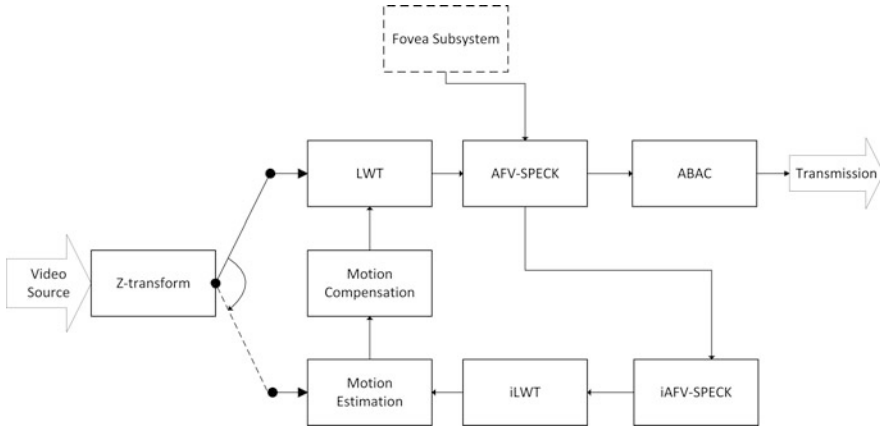


Fig. 19.9 Video coding framework AWFV-Codec [16]

model is updated. The adaptive part of the algorithm decreases its performance when compared against a static approach. However, the main advantage is that the input data is not preprocessed. As a result, the efficiency of the transmission of the compressed stream increases because there is no wait time for the calculation of the statistical model. There are several applications for ABAC as in JPEG and Joint Bi-level Image Group (JBIG)⁵ when dealing with black and white images. However, because SPECK encodes per bit, it makes ABAC suitable to compress the output of SPECK. In order to increase the computing time performance of the proposed framework, ABAC is included as its variable length encoder. Listing 19.1 shows the pseudocode for adaptive binary arithmetic coding.

Listing 19.1 ABAC algorithm

```

function ArithmeticCoding ( s )
    fq ← 1
    r ← 0
    l ← 0
    u ← 1
    for all s ∈ s do
        r ← r + 1
        if s = 0 then
            l' ← 0
            u' ← P/r
            fq ← fq + 1
        else
            l' ← fq/r
            u' ← 1
    
```

⁵<http://jpeg.org/jbig/index.html>.

```

        end if
        d = u - l
        u = l + d · u'
        l = l + d · l'
    end for
    return l
end function

```

In classic arithmetic coding, the interval used for arithmetic compression is $[0, 1)$. The function receives a string s to be compressed. The variable fq will store the frequency of the symbol 0. Because there are only two symbols on the alphabet, it is only needed to store one of the frequencies and compute the other by

$$P_1 = 1 - P_0 \quad (19.23)$$

where P_i is the probability of the symbol i . The probability of the symbol 0 is given by

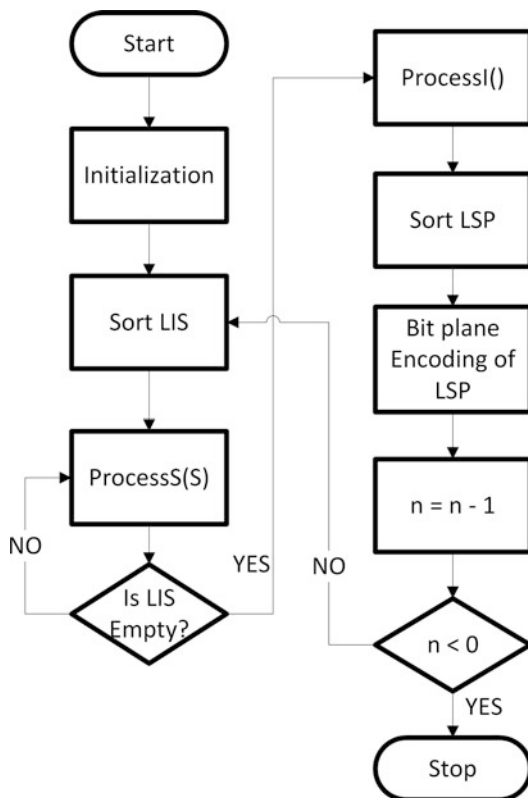
$$P_0 = \frac{fq}{r} \quad (19.24)$$

where r is the amount of read symbols. The algorithm stores the lower bound of the main interval on l and the upper bound on u . Each time a symbol is read, the counter r is increased on 1 and the interval for the input symbol is updated by using the frequency of the symbol 0 stored on fq . If a 0 is read, the frequency fq is increased in 1. After updating the statistical model, the new main interval is computed and the next symbol is read. The process stops when there are no more symbols to read on s and the statistical model P . P is a set that contains the probabilities of all different symbols s of the alphabet of s .

19.7.2 AFV-SPECK Algorithm

In the AFV-SPECK algorithm, every time a new coefficient is categorized as significant it will also be tested for its individual compression ratio using the cutoff window for each wavelet decomposition subband [16] (see Fig. 19.10). Note that the main loop remains the same as with SPECK. The input is the set of quantized coefficients, while the output is stored on S (assessed for significance by the function *ProcessS*), and the sorting of the LSP set is also added. If S is significant and only has one element (x, y) , the sign of quantized coefficient is stored on S and the set is removed from LIS. The function *ProcessI* evaluates I for its significance. As with FVHT, the computational complexity of AFV-SPECK will be expressed in terms of the *Big O* notation. The AFV-SPECK algorithm has a computational complexity of $\mathcal{O}(n)$. The analysis of the memory usage yielded that AFV-SPECK uses more memory when implemented as proposed in [31]. Further details can be found in [16].

Fig. 19.10 Flowchart of the main AFV-SPECK algorithm loop [16]



19.7.3 Simulation Results

To assess the reviewed video coding frameworks, SP-Codec and AWFV-Codec standard test images and video sequences were used⁶ [16]. For intra-frame coding, H.265 standard based on the IDCT using a 4×4 pixel block size is compared against SPECK and AFV-SPECK algorithms. Both binary streams were further compressed using the ABAC algorithm. The delta used for quantization was set to $\Delta = 40$, see e.g., [52]. Note that the chosen quantization delta and other parameters were used as input for SPECK and AFV-SPECK algorithms [16]. This is due to the fact that the compression ratio of the H.265 cannot be specified beforehand.

It is well-known that there is no analytic method to represent the exact perception of the HVS [56]. As a result, there are different metrics for image quality metrics [55]. In this work, the peak signal-to-noise ratio (PSNR) is used as performance metric [37]. The PSNR is defined in terms of the mean squared error (MSE) given by the equation

⁶<https://media.xiph.org/video/derf/>.

$$\text{MSE}(I, K) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_{i,j} - K_{i,j}]^2 \quad (19.25)$$

where m denotes the rows and n the columns of original image matrix, I is the matrix of the original image, and K represents the reconstructed image matrix. Using Eq. (19.25), the PSNR is given by

$$\text{PSNR}(I, K) = 10 \log_{10} \left[\frac{\text{MAX}_I^2}{\text{MSE}(I, K)} \right] \quad (19.26)$$

where MAX_I^2 is the square of the maximum value that a pixel of the image I can take. Such value depends on the amount of bits used per channel. Commonly, an image of 8 bits per channel has $\text{MAX}_I^2 = 255^2$. PSNR is measured in decibels (dB). Usually, it is considered that a reconstructed image with a PSNR of 40 dB or higher is of good quality for an average user [44]. However, trained users should require higher PSNR values. The 40 dB threshold is only a convention and has not been proved. Expected values of good reconstructions are between 20 dB and 50 dB [44].

As stated in the standard JPEG2000 and for a fair comparison, we use the biorthogonal CDF9/7 with four levels of decomposition [1]. Two metrics are used to assess the performance of the reported algorithms, namely PSNR and structural similarity index (SSIM) [54, 61]. This metric indicates that a reconstructed image with high quality should give a SSIM index closer to 1. Table 19.1 depicts comparisons in images for various video sequences using H.265, SPECK and AFV-SPECK algorithms, where CIF stands for common intermediate format. This table shows that the SPECK algorithm has a high PSNR (see e.g., [29]). It also observed that since the reported AFV-SPECK algorithm is based on ROIs and fovea centralis, it is expected that the result of these metrics to be equal or lower than the SPECK algorithm. Further details on these comparisons and other sequences are reported in [16].

Table 19.1 Comparisons between SPECK, AFV-SPECK and H.265 (see [16])

Name	BPP	H.265		SPECK		AFV-SPECK	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena gray 512	1.32	29.37	0.83	37.07	0.96	35.00	0.95
Lake	1.33	29.24	0.83	32.93	0.93	31.56	0.91
Peppers gray	1.27	29.31	0.81	34.04	0.92	32.77	0.90
Cameraman	1.28	28.93	0.75	40.07	0.97	34.59	0.94
Akiyo cif	1.19	28.96	0.78	35.42	0.94	33.12	0.92
Paris cif	1.28	28.79	0.76	30.45	0.85	29.53	0.83
Soccer cif	1.37	29.33	0.74	34.89	0.93	32.23	0.92

19.8 Conclusions

In this chapter, two wavelet-based algorithms were reviewed, namely FVHT and AFV-SPECK. Such algorithms exploit the HVS in order to increase the quality of the reconstructed image for an observer. The algorithms were assessed against classic compression algorithms such as the JPEG base algorithm and the algorithm used on the H.265 standard. Simple wavelet compression shows better performance when compressing images allowing to reach compression ratios of 0.06 bpp while retaining a good visual quality. The reported algorithms show similar behavior while increasing the quality of the compressed image over designed areas. However, when evaluated for overall quality, the reported algorithms show less performance than its non-fovea-based counterparts. This makes necessary an external subsystem that calculates the fixation point of the observers. Additionally, two wavelet-based video coding frameworks were surveyed, namely SP-Codec and AWFV-Codec [16]. The revised video frameworks increase the key frame reconstruction using wavelet-based compression that is also applied to motion compensation reconstruction. Fovea centralis coding also increases the quality of the reconstructed video as in AWFV-Codec, and in some cases, increases the quality of the reconstructed frames against non-fovea-based frameworks like SP-Codec. The reported AWFV-Codec is a viable choice for fast video streaming but it also reduces the utility of the stream when recorded. This is due to the fact that the video would be recorded without possibility of recovering the information discarded outside the fovea centralis. However, when stream recording is needed SP-Codec yields into better reconstruction quality than classic methods such as the H.265/HEVC video coding frameworks [15, 16]. The reported image compression algorithms FVHT and spatial transform AFV-SPECK require extra storage besides the wavelet coefficients. Methods will be investigated for in place computation for quantization in order to decrease the memory usage of both reported algorithms and for automatic foveation such as in [21].

Acknowledgement The authors gratefully acknowledge the financial support from CONACYT, Mexico.

References

1. Acharya, T., & Tsai, P. S. (2004). *JPEG2000 standard for image compression*. Hoboken, NJ: Wiley.
2. Alarcon-Aquino, V., & Barria, J. A. (2006). Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(2), 208–220. <https://doi.org/10.1109/TSMCC.2004.843217>
3. Bocharova, I. (2010). *Compression for multimedia*. Cambridge: Cambridge University Press. <http://books.google.com/books?id=9UXBxPT5vuUC&pgis=1>
4. Böck, A. (2009). *Video compression systems: From first principles to concatenated codecs*. IET telecommunication series. Stevenage: Institution of Engineering and Technology. <http://books.google.com.mx/books?id=zJyOx08p421C>

5. Boopathi, G., & Arockiasamy, S. (2012). Image compression: Wavelet transform using radial basis function (RBF) neural network. In: *2012 Annual IEEE India Conference (INDICON)* (pp. 340–344). Piscataway: IEEE. <https://doi.org/10.1109/INDICON.2012.6420640>
6. Bovik, A. C. (2009). *The essential guide to video processing* (1st ed.). London: Academic Press.
7. Chang, E., Mallat, S., & Yap, C. (2000). Wavelet foveation. *Applied and Computational Harmonic Analysis*, 9(3), 312–335.
8. Cintra, R., Bayer, F., & Tablada, C. (2014). Low-complexity 8-point DCT approximations based on integer functions. *Signal Processing*, 99, 201–214. <https://doi.org/10.1016/j.sigpro.2013.12.027>. <http://www.sciencedirect.com/science/article/pii/S0165168413005161>
9. Ciocoiu, I. B. (2009). ECG signal compression using 2D wavelet foveation. In *Proceedings of the 2009 International Conference on Hybrid Information Technology - ICHIT '09* (Vol. 13, pp. 576–580)
10. Ciubotaru, B., Ghinea, G., & Muntean, G. M. (2014). Subjective assessment of region of interest-aware adaptive multimedia streaming quality. *IEEE Transactions on Broadcasting*, 60(1), 50–60. <https://doi.org/10.1109/TBC.2013.2290238>. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6755558>
11. Daubechies, I., & Sweldens, W. (1998). Factoring wavelet transforms into lifting steps. *The Journal of Fourier Analysis and Applications*, 4(3), 247–269. <https://doi.org/10.1007/BF02476026>. <http://link.springer.com/10.1007/BF02476026>
12. Dempsey, P. (2016). The teardown: HTC vive VR headset. *Engineering Technology*, 11(7–8), 80–81. <https://doi.org/10.1049/et.2016.0731>
13. Ding, J. J., Chen, H. H., & Wei, W. Y. (2013). Adaptive Golomb code for joint geometrically distributed data and its application in image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4), 661–670. <https://doi.org/10.1109/TCSVT.2012.2211952>. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6261530>
14. Frazier, M. (1999). *An introduction to wavelets through linear algebra*. Berlin: Springer. <http://books.google.com/books?id=IIRdY9nUTZgC&pgis=1>
15. Galan-Hernandez, J., Alarcon-Aquino, V., Ramirez-Cortes, J., & Starostenko, O. (2013). Region-of-interest coding based on fovea and hierarchical tress. *Information Technology and Control*, 42, 127–352. <http://dx.doi.org/10.5755/j01.itc.42.4.3076>. <http://www.itc.ktu.lt/index.php/ITC/article/view/3076>
16. Galan-Hernandez, J., Alarcon-Aquino, V., Starostenko, O., Ramirez-Cortes, J., & Gomez-Gil, P. (2018). Wavelet-based frame video coding algorithms using fovea and speck. *Engineering Applications of Artificial Intelligence*, 69, 127–136. <https://doi.org/10.1016/j.engappai.2017.12.008>. <http://www.sciencedirect.com/science/article/pii/S0952197617303032>
17. Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing* (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.
18. Gray, R. M. (2011). *Entropy and information theory* (Google eBook). Berlin: Springer. <http://books.google.com/books?id=wdSQgVbdRcC&pgis=1>
19. Hanzo, L., Cherriman, P. J., & Streit, J. (2007). *Video compression and communications*. Chichester, UK: Wiley.
20. Homann, J. P. (2008). *Digital color management: Principles and strategies for the standardized print production* (Google eBook). Berlin: Springer. <http://books.google.com/books?id=LatEFg5VBZ4C&pgis=1>
21. Itti, L. (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10), 1304–1318. <http://dx.doi.org/10.1109/TIP.2004.834657>
22. Kondo, H., & Oishi, Y. (2000). Digital image compression using directional sub-block DCT. In *WCC 2000 - ICCT 2000. 2000 International Conference on Communication Technology Proceedings (Cat. No.00EX420)* (Vol. 1, pp. 985–992). Piscataway: IEEE. <http://dx.doi.org/10.1109/ICCT.2000.889357>. <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=889357>

23. Lakhani, G. (2013). Modifying JPEG binary arithmetic codec for exploiting inter/intra-block and DCT coefficient sign redundancies. *IEEE transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, 22(4), 1326–39. <http://dx.doi.org/10.1109/TIP.2012.2228492>. <http://www.ncbi.nlm.nih.gov/pubmed/23192556>
24. Lee, S., & Bovik, A. C. (2003). Fast algorithms for foveated video processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(2), 149–162. <http://dx.doi.org/10.1109/TCSVT.2002.808441>
25. Li, J. (2013). An improved wavelet image lossless compression algorithm. *International Journal for Light and Electron Optics*, 124(11), 1041–1044. <http://dx.doi.org/10.1109/10.1016/j.ijleo.2013.01.012>. <http://www.sciencedirect.com/science/article/pii/S0030402613001447>
26. Liu, L. (2008). On filter bank and transform design with the lifting scheme. Baltimore, MD: Johns Hopkins University. <http://books.google.com/books?id=f0IxpHYF0pAC&pgis=1>
27. Mallat, S. (2008). *A wavelet tour of signal processing, third edition: The sparse way* (3rd ed.). New York: Academic Press.
28. Miano, J. (1999). *Compressed image file formats: JPEG, PNG, GIF, XBM, BMP* (Vol. 757). Reading, MA: Addison-Wesley. http://books.google.com/books?id=_nJL.vY757dQC&pgis=1
29. Mohanty, B., & Mohanty, M. N. (2013). A novel speck algorithm for faster image compression. In *2013 International Conference on Machine Intelligence and Research Advancement* (pp. 479–482). <http://dx.doi.org/10.1109/ICMIRA.2013.101>
30. Ozenli, D. (2016). Dirac video codec and its performance analysis in different wavelet bases. In *24th Signal Processing and Communication Application Conference (SIU)* (pp. 1565–1568). <http://dx.doi.org/10.1109/SIU.2016.7496052>
31. Pearlman, W., Islam, A., Nagaraj, N., & Said, A. (2004) Efficient, low-complexity image coding with a set-partitioning embedded block coder. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(11), 1219–1235. <http://dx.doi.org/10.1109/TCSVT.2004.835150>
32. Peter, S., & Win, S. (2000). *Wavelets in the geosciences*. Lecture Notes in Earth Sciences (Vol. 90). Berlin: Springer. <http://dx.doi.org/10.1007/BFb0011093>. <http://www.springerlink.com/index/10.1007/BFb0011091>, <http://link.springer.com/10.1007/BFb0011091>
33. Poynton, C. (2012). *Digital video and HD: Algorithms and interfaces* (Google eBook). Amsterdam: Elsevier. <http://books.google.com/books?id=dSCEGFt47NkC&pgis=1>
34. Rao, K. R., Kim, D. N., & Hwang, J. J. (2011). *Fast Fourier transform—algorithms and applications: Algorithms and applications* (Google eBook). Berlin: Springer. <http://books.google.com/books?id=48rQQ8v2rKEC&pgis=1>
35. Rehna, V. (2012). Wavelet based image coding schemes: A recent survey. *International Journal on Soft Computing*, 3(3), 101–118. <http://dx.doi.org/10.5121/ijsc.2012.3308>. <http://www.airccse.org/journal/ijsc/papers/3312ijsc08.pdf>
36. Richardson, I. E. (2004). *H.264 and MPEG-4 video compression: Video coding for next-generation multimedia* (Google eBook). London: Wiley. <http://books.google.com/books?id=n9YVhx2zgZ4C&pgis=1>
37. Richardson, I. E. G. (2002). Video codec design. Chichester, UK: Wiley. <http://dx.doi.org/10.1002/0470847832>, <http://doi.wiley.com/10.1002/0470847832>
38. Rivas-Lopez, M., Sergiyenko, O., & Tyrsa, V. (2008). Machine vision: Approaches and limitations. In: Zhihui, X. (ed.) *Chapter 22: Computer vision*. Rijeka: IntechOpen. <https://doi.org/10.5772/6156>
39. Rivas-Lopez, M., Sergiyenko, O., Flores-Fuentes, W., & Rodriguez-Quinonez, J. C. (2019). *Optoelectronics in machine vision-based theories and applications* (Vol. 4018). Hershey, PA: IGI Global. ISBN: 978-1-5225-5751-7.
40. Ross, D., & Lenton, D. (2016). The graphic: Oculus rift. *Engineering Technology*, 11(1), 16–16. <http://dx.doi.org/10.1049/et.2016.0119>
41. Sacha, D., Zhang, L., Sedlmair, M., Lee, J. A., Peltonen, J., Weiskopf, D., et al. (2017). Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 241–250. <http://dx.doi.org/10.1109/TVCG.2016.2598495>

42. Said, A., & Pearlman, W. (1996). A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3), 243–250.
43. Salomon, D. (2006). *Coding for data and computer communications (Google eBook)*. Berlin: Springer. <http://books.google.com/books?id=Zr9bjEpXKnIC&pgis=1>
44. Salomon, D. (2006). *Data compression: The complete reference*. New York, NY: Springer.
45. Salomon, D., Bryant, D., & Motta, G. (2010). *Handbook of data compression (Google eBook)*. Berlin: Springer. <http://books.google.com/books?id=LHCY4VbiFqAC&pgis=1>
46. Sayood, K. (2012). *Introduction to data compression*. Amsterdam: Elsevier. <http://dx.doi.org/10.1016/B978-0-12-415796-5.00003-X>. <http://www.sciencedirect.com/science/article/pii/B978012415796500003X>
47. Schanda, J. (2007). *Colorimetry: Understanding the CIE system (Google eBook)*. London: Wiley. <http://books.google.com/books?id=uZadszSGe9MC&pgis=1>
48. Sergiyenko, O., & Rodriguez-Quinonez, J. C. (2017). *Developing and applying optoelectronics in machine vision* (Vol. 4018). Hershey, PA: IGI Global. ISBN: 978-1-5225-0632-4.
49. Silverstein, L. D. (2008). Foundations of vision. *Color Research & Application*, 21(2), 142–144.
50. Song, E. C., Cuff, P., & Poor, H. V. (2016). The likelihood encoder for lossy compression. *IEEE Transactions on Information Theory*, 62(4), 1836–1849. <http://dx.doi.org/10.1109/TIT.2016.2529657>
51. Stollnitz, E., DeRose, A., & Salesin, D. (1995). Wavelets for computer graphics: A primer. I. *IEEE Computer Graphics and Applications*, 15(3), 76–84. <http://dx.doi.org/10.1109/38.376616>. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=376616>
52. Sullivan, G. J., Ohm, J. R., Han, W. J., & Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1649–1668. <http://dx.doi.org/10.1109/TCSVT.2012.2221191>
53. Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2), 186–200. <http://dx.doi.org/10.1006/acha.1996.0015>. <http://www.sciencedirect.com/science/article/pii/S1063520396900159>
54. Tan, T. K., Weerakkody, R., Mrak, M., Ramzan, N., Baroncini, V., Ohm, J. R., et al. (2016). Video quality evaluation methodology and verification testing of HEVC compression performance. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1), 76–90. <http://dx.doi.org/10.1109/TCSVT.2015.2477916>
55. Tanchenko, A. (2014). Visual-PSNR measure of image quality. *Journal of Visual Communication and Image Representation*, 25(5), 874–878. <http://dx.doi.org/10.1016/j.jvcir.2014.01.008>. <http://www.sciencedirect.com/science/article/pii/S1047320314000091>
56. Theodoridis, S. (2013). *Academic press library in signal processing: Image, video processing and analysis, hardware, audio, acoustic and speech processing (Google eBook)*. London: Academic Press. <http://books.google.com/books?id=QJ3HqmLG8glC&pgis=1>
57. Viction Workshop L. (2011). *Vectorism: Vector graphics today*. Victionary. <http://books.google.com/books?id=dHaeZwEACAAJ&pgis=1>
58. Wallace, G. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), xviii–xxxiv. <http://dx.doi.org/10.1109/30.125072>. <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=125072>
59. Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, 34(4), 30–44. <http://dx.doi.org/10.1145/103085.103089>. <http://dl.acm.org/citation.cfm?id=103085.103089>
60. Walls, F. G., & MacInnis, A. S. (2016). VESA display stream compression for television and cinema applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(4), 460–470. <http://dx.doi.org/10.1109/JETCAS.2016.2602009>
61. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <http://dx.doi.org/10.1109/TIP.2003.819861>

62. Werner, J. S., & Backhaus, W. G. K. (1998). *Color vision: Perspectives from different disciplines*. New York, NY: Walter de Gruyter. <http://books.google.com/books?id=gN0UaSUTbnUC&pgis=1>
63. Wien, M. (2015). *High efficiency video coding— coding tools and specification*. Berlin: Springer.
64. Zhang, L., Wang, D. &, Zheng, D. (2012). Segmentation of source symbols for adaptive arithmetic coding. *IEEE Transactions on Broadcasting*, 58(2), 228–235. <http://dx.doi.org/10.1109/TBC.2012.2186728>. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6166502>

Chapter 20

Stairway Detection Based on Single Camera by Motion Stereo for the Blind and Visually Impaired



Javier E. Sanchez-Galan, Kang-Hyun Jo, and Danilo Cáceres-Hernández

Acronyms

BLS	Bottommost line segments
CCD	Charge-coupled device
CNN	Convolutional neural networks
HH	Horizontal histogram
ILSVRC2012	Large Scale Visual Recognition Challenge 2012
KNN	K-nearest neighbors
MDPG	Maximum distance of plane ground
PA	Candidate area
PE	Number of areas
PL	Number of lines
PP	Number of pixels
SVM	Support vector machines

J. E. Sanchez-Galan

Grupo de Investigación en Biotecnología, Bioinformática y Biología de Sistemas, Centro de Producción e Investigaciones Agroindustriales (CEPIA), Universidad Tecnológica de Panamá, Panama, Republic of Panama

Institute of Advanced Scientific Research and High Technology, Panama, Republic of Panama
e-mail: javier.sanchezgalan@utp.ac.pa

K.-H. Jo

Intelligent Systems Laboratory, Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea
e-mail: acejo@ulsan.ac.kr

D. Cáceres-Hernández (✉)

Grupo de Sistemas Inteligentes, Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá, Panamá, Republic of Panama
e-mail: danilo.caceres@utp.ac.pa

vAOV	Vertical angle of view
VH	Vertical histogram

20.1 Introduction

Stairway detection has been widely studied in the field of image processing. In this regard, this set of research has solved different kind of problems such as ascending or descending stairways to solve for example the case of disaster situations, as well as personal and assistive applications. Examples of this development are unmanned ground vehicle for exploration and surveillance and implementation of passive alarm systems intended for blind and visually impaired people.

The development of stair localization and recognition application has a great importance to solve problems faced by blind or visually impaired persons. To solve the current problem, researchers propose partly solving the problem by separating outdoor and indoor cases. In addition, in most of the cases, the region where the stairs appear has been usually evaluated in predefined region of interest [1–8].

Authors in [1] use geometrical properties and vertical vanishing point to solve the detection problem. To test the algorithm, the system should follow some criteria such as the camera coordinate position, distance between the camera and the stairs. In that sense, the algorithm will encounter difficulties at the moment to deal with outdoor environments. In [2], the authors develop the idea of using the stereo vision. Hence, the authors presented a lightweight idea to update the stair model. Although the method shows good performance, the algorithm will encounter problems with stairs located far from the camera. Researchers in [3] proposed a method based on depth information using supervised learning models.

In recent years, convolutional neural networks (CNN) has been proposed as suitable method for object detection in the exploration of indoor environments for robots and for visually impaired people using different approaches from image segmentation with CNN, to faster R-CNN and finally to pretrained deep convolutional neural network (CNN) [9–12]. One example of the latter is documented in Bashiri et al. which introduced the MCIIndoor dataset which contains a large-scale fully labeled image dataset for doors, stairs, and hospital signs and experimented with a pretrained deep convolutional neural network (CNN) model AlexNet and fine-tuned to their image datasets, and furthermore, compared their results with classical techniques such as support vector machines (SVM), k-nearest neighbors (KNN) and naïve Bayesian classifiers [13, 14].

In this chapter, the initial results from the ongoing research belonging to stair recognition for both indoor and outdoor stairs are presented. This is an important task due to the fact that there are many different types of stairs with problems of illumination, shade problems, and problems given by the shapes and designs. Based on these considerations, it has been proposed to develop a system using deep learning for stair image analysis to classify within the indoor, outdoor, not stair, or structure classes. Taking into account that the presented idea is an ongoing research, the next step deals with the stair features given from the stairway edge.

20.2 Algorithm Description

In this section, the proposed algorithm for determining the cognitive properties of stairways has two main stages: (1) convolutional neural network model description and (2) feature extraction of stairways (see Fig. 20.1).

20.2.1 Convolutional Neural Network Model Description

Images coming from the indoor, outdoor, and structures datasets (see Fig. 20.2) were queried against a state-of-the-art deep learning model NASNet-Large [15]. NasNET is an Auto-ML CNN model developed by Google comprising 1041 convolutional layers distinguished into Normal and Reduction layers and 88.9 millions of parameters pretrained on the Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) image classification Imagenet dataset [16].

The Imagenet dataset is composed of 1.2 million images representing 1000 classes of images or labels. Regarding the classification challenge in hand, Imagenet does not provide a high-level descriptor for stairs, ladder or staircases; however, it provides one label “n02788148” corresponding a prominent features in stairs representing “bannister, banister, balustrade, balusters, handrail.” So the prediction task involves, passing the image through the convolutional layers of the pre-trained network and then assessing if among the resulting predicted class labels, the class banister is among the top-n results.

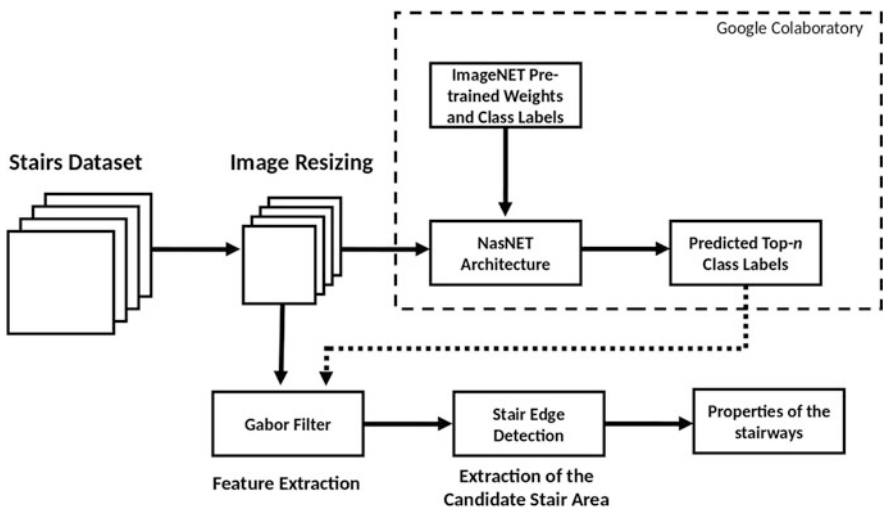


Fig. 20.1 Flowchart of the implemented algorithms for stairway recognition. On the top right part of the image, the convolutional neural network approach and, on the bottom part, the feature extraction methodology are shown. A dashed arrow connecting both methods describes the ensemble method that will be considered in future works

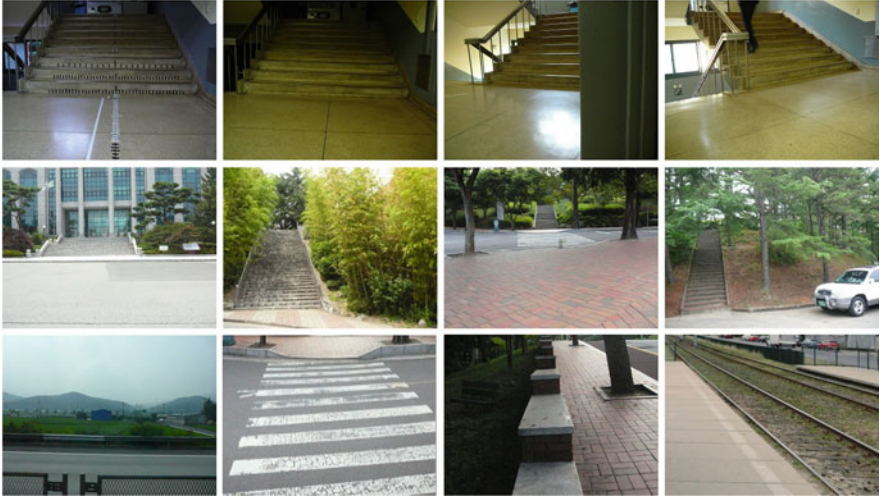


Fig. 20.2 Image dataset. The first row shows the indoor stair categories. The second row shows the outdoor categories. The third row shows other structure dataset

20.2.2 Stairway Detections

Since our efforts focus on helping blind or visually impaired people with reduced mobility, the stairways are previously detected within a defined region. This stairways detection algorithm can be summarized as follows.

First, from a given image [17, 18], the true horizon or maximum distance of plain ground (MDPG) is detected. This is done by finding the pixel position that can discard the area located in the horizon vision within the image. The horizon is computed according to the following equations:

$$d = \frac{h}{\tan \delta + \alpha} \quad (20.1)$$

$$\alpha = \tan^{-1} \left(\frac{y - y_c}{f} \right) \quad (20.2)$$

$$f_p = \frac{f_{mm} - \text{Img}_p}{\text{CCD}_{mm}} \quad (20.3)$$

where d is the distance between the camera and the target object, h is the height of the camera above the ground, and δ is the angle between the optical axis of the camera and the horizontal projection, α is the angle formed by scanning the image from top to bottom with respect to the center point of the image, y is the pixel position of the target on the y axis, y_c is the pixel position on the image center, f_p is the focal length



Fig. 20.3 Testing image sets for finding the MDGP with different camera height values, from up to down the height value were 1 m, 0.75 m, and 0.50 m above the ground plane. (a) Input images, the red line indicated where the MDGP was localized. (b) Output image after removing the information above the horizon line

in pixels, f_{mm} is the focal length in millimeters, Img_p is the image width in pixels, and CCD_{mm} is the sensor width in millimeters.

Figure 20.3 shows the results of the MDGP stage in our testing image sets. Those images were taken with the camera set at three different heights above the ground plane. The heights were 1.00 m, 0.75 m, and 0.50 m, respectively. The red line in the input images is the localization of the discontinuity after scanning the images from top to bottom. Finally, this step is done in order to get a more efficient system and,

consequently, reduce the time required for all the computational process, which is nowadays of paramount importance in real-time applications which consist mainly of extracting the information surrounding the ground plane.

Second, by using Gabor filter [19–22], the information related to the stairs is estimated. This is done using the equation:

$$G_{(x,y,\lambda,\theta,\phi,\sigma,\gamma)} = \exp\left(\frac{x'^2+y'^2+y'^2}{2\sigma}\right) \cos 2\pi \frac{x'}{y} + \phi \quad (20.4)$$

$$x' = x \cos \theta + y \sin \theta \quad (20.5)$$

$$y' = -x \sin \theta + y \cos \theta \quad (20.6)$$

where x and y specify the position intensity value along the image, λ represents the wavelength of the sinusoidal factor, q represents the orientation, θ is the phase shift, and ϕ is the standard deviation of the Gaussian envelope along the x and y axes. In addition, γ is the spatial aspect ratio and specifies the ellipticity of the support of the Gabor function. Figure 20.4 shows the result of the Gabor filter in the synthesized image sets.

Third, the line segments within the image are extracted by using a three-chain rule, represented as

$$T = f(x+1, y \pm d\phi), g(x, y) = \begin{cases} 0, & f(x, y) = T \\ 1, & f(x, y) \neq T \end{cases} \quad (20.7)$$

where x and y specify the position intensity value along the image f , T is the pixel value of the pixel which is located in the actual position, $d\phi$ represents the pixel orientation, and $g(x, y)$ represents the new image based on the neighborhood information. The set of candidates are extracted by applying a horizontal and vertical histogram of the edge image.

The horizontal histogram (HH) determines the numbers of candidates. The vertical histogram (VH) determines the number of lines within the candidates extracted from HH. The algorithm is then able to determine the number of lines and pixels belonging to each line segment. Second, based on the probability analysis, the proposed method evaluated the candidate area (PA) as well as the number of lines (PL) and number of pixels (PP) in each respective area. The probability of the estimates is defined as follows:

$$P_E = \frac{N_E}{N} \quad (20.8)$$

where N_E is the number of areas, lines segments, and pixels in E, and N is the total number of area, line segments, and pixels in the image. Figure 20.5 shows the result

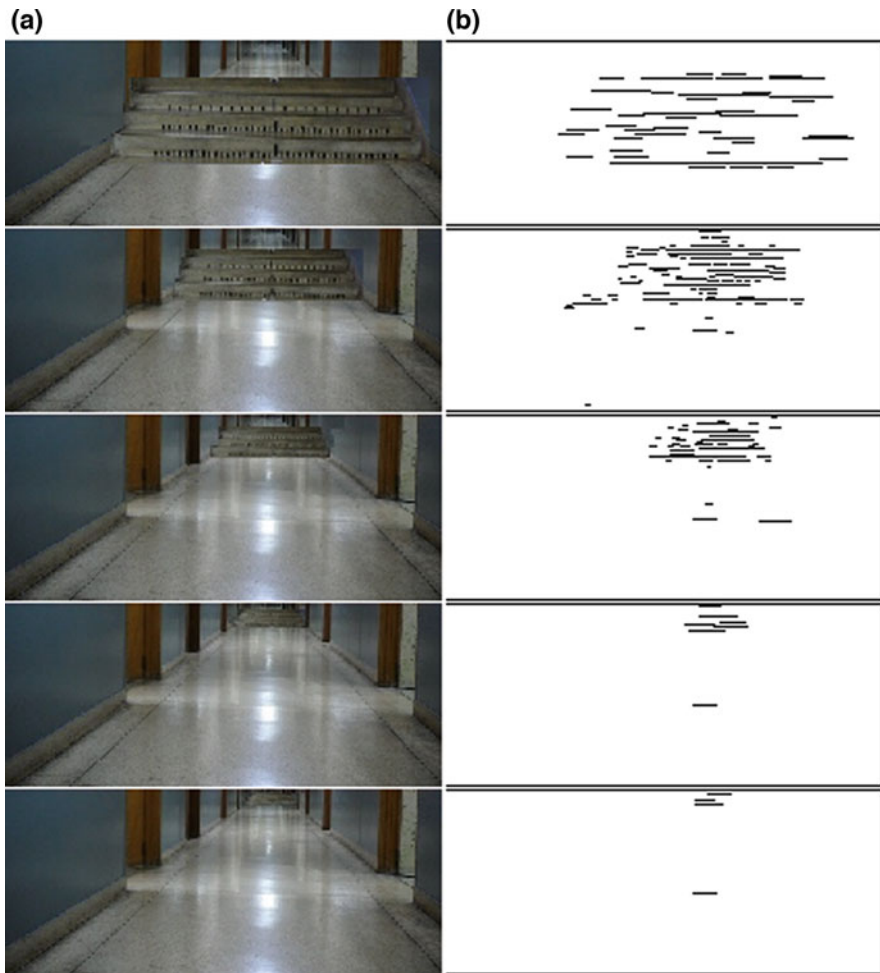


Fig. 20.4 Result of the Gabor filter in our synthesized image sets with different distance value between the camera position and the stairways, from up to down the approximate distance values were 4.25 m, 6 m, 9 m, 13 m, and 15 m. **(a)** Source image. **(b)** Result of the Gabor filter after binarization. Note that most of the line segments were extracted from the area where the stairway was located

of this step applied in a set of synthesized images, where the stair is located at a different distance from the camera by applying a statistical analysis in horizontal and vertical positions.

Fourth, the correspondence between two consecutive images is computed by using the normalized cross-correlation, expressed as follows:

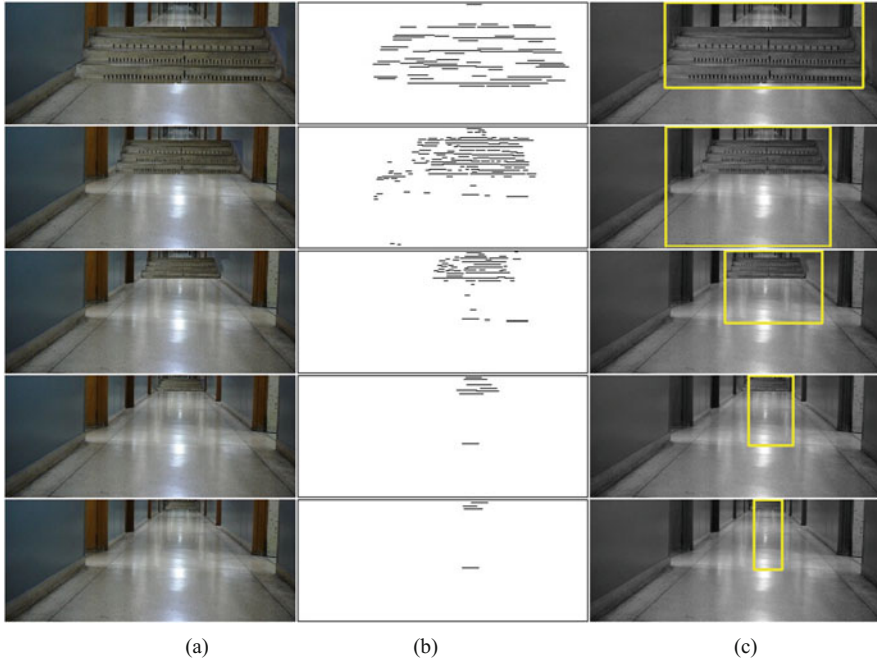


Fig. 20.5 Synthesized image stair set. The stair is located at different distances from the camera. (a) Input images. (b) Gabor filter results. (c) Stair candidate area

$$r_s = \frac{\sum_i (x_i - m_x)(y_i - m_y)}{\sum_i (x_i - m_x)^2 \sum_i (y_i - m_y)^2} \quad (20.9)$$

where r_s is the cross-correlation coefficient, x_i is the intensity of the i -th pixel in image within the frame t , y_i is the intensity of the i -th pixel in image in the frame $t + 1$, and m_x and m_y are the means of intensity of the corresponding images. A correlation coefficient matrix is computed for every displacement between the regions. After the stairway candidate region is obtained by proposed algorithm, proceed with estimated trajectory to infer the actual localization of the stairway. The aim of this stage is to estimate the localization of the stairways into the 3D plane. The process established under the previous step so far is recursively repeated until the extraction of the set of targets. In order to calculate the bottommost line segment (BLS) distance according to the configuration of the camera system, consider the measured limit as 1.30 m from the center of the camera. On the image plane, this distance is found on the y -axis at the bottommost pixel position, which was approximately equal to 240. Based on the above information, the process will stop when the bottommost line segments are located at the 230 pixel on the y -axis. From the last step, one is able to define the stairway properties.

The proposed algorithm needs to calculate the vertical angle of view (vAOV) as follows:

$$\text{vAOV} = 2 \tan^{-1} \left(\frac{\text{CCD}_{\text{mm}}}{2f_{\text{mm}}} \right) \quad (20.10)$$

$$\text{BSL} = \tan^{-1} \left(\frac{h}{\delta + \text{vAOV}} \right) \quad (20.11)$$

where vAOV is the vertical angle of view of the camera which describes the angular extent of a given scene that is imaged by a camera, f_{mm} is the focal length in millimeters, h is the height of the camera above the ground, CCD_{mm} is the sensor width in millimeters, and BSL is the blind spot distance according to the camera coordinate system.

20.3 Experimental Results

For this first experiment of convolutional neural network model description, the Google Colaboratory cloud was used. Keras and Tensorflow Models written in the Python programming language were directly run in the cloud instance on a NVIDIA[®] Tesla[®] K80 GPU card with 13 Gb of GDDR5 VRAM. The indoor, outdoor, and structure image sets were queried against NasNET obtaining the average class labels described in Tables 20.1, 20.2, and 20.3, respectively. As the expected results for the indoor dataset were found to be the most accurate with the network successfully classifying the image into the “banister” class in most cases, inferred with a probability of at least 70%.

Table 20.1 Localization of the stairway candidate results for indoor dataset

D	NIS	L	LF	AP
Group 1	47	Banister	47	0.81
Group 2	23	Banister	23	0.70
Group 3	57	Banister	57	0.79
Group 4	50	Banister	47	0.73
		Prison	3	0.46

Note: D stands for dataset, NIS stands for number of image in the set, L stands for label (top-1), LF stands for label frequency, and AP stands for average probability. Dataset from group 1 belongs to image taken during the daylight with the stair located in front of the camera. Image in group 2 belongs to image taken during nightlight with the stair located in front of the camera. Group 3 belongs to images where the image is located at the right part of the camera, and group 4 belongs to images where the stair is located at the left part of the camera

Table 20.2 Localization of the stairway candidate results for outdoor dataset

D	NI	L	LF	AV
Outdoor	62	Banister	18	0.70
		Library	10	0.77
		Bell cote	7	0.81
		Patio	7	0.47
		Suspension bridge	4	0.86
		Maze	4	0.62
		Planetarium	3	0.88
		Triumphal arch	3	0.33
		Park bench	3	0.30
		Lakeside	3	0.42
		Palace	3	0.31
		Bookshop	2	0.59
		Obelisk	2	0.49
		Brass	2	0.27
		Unicycle	2	0.37
		Organ	2	0.38
Limousine	2	0.11		
Motor scooter	2	0.42		
Stupa	2	0.13		

Note: *D* stands for dataset, *NI*S stands for number of image in the set, *L* stands for label (top-1), *LF* stands for label frequency, and *AP* stands for average probability

Table 20.3 Localization of the stairway candidate results for structure datasets

Dataset	NI	L	LF	AP
Structure dataset	5	Banister	1	0.18
		Maze	2	0.58
		Limousine	1	0.17
		Horizontal bar	1	0.15

Notable exceptions, although with low probability, were in the cases that the algorithm inferred the “prison” class, most likely related to the angle-of-view of the image and the similarity of the metal banister and the prison bars. Also, an interesting result was when the algorithm inferred the “scabbard” class, for a close-up of the stairs, mostly suggesting or taking into account the high-level features of the image such as roughness or surface of the stairs in the likeness of a sword sheath.

For the outdoor dataset results were mixed, inferring the banister class on 29% of the examples of the class with a probability of 70%. Other labels appear but mostly alluding the fact that other objects appear in the image including buildings (“planetarium,” “palace,” “arch”) or even vehicles (“unicycle,” “limousine,” “motor scooter”).

For the cases of Structures, banister is inferred with marginal probability of 0.18. Also, among the predictions other unrelated labels classes of structural elements (horizontal bar and maze) and vehicles (limousine) are found. Taken together the results for outdoor and structures datasets suggest that the approach of passing the image through the convolutional layers of the NASnet pre-trained network and comparing the resulting predicted labels, is working modestly well. More importantly for the task in hand, on average, the probabilities for predicting the label “bannister” for the indoor dataset, are well over 70%.

In order to fully evaluate the performance of the proposed features extracting stairway algorithm, four different image datasets were created. This data consisted of 75 color images with a resolution of 320×240 . This database is formed by photographing one stairway with both day- and nightlighting conditions. Different properties, such as orientation with respect to the camera position, and illumination conditions were observable. Below are some example results for the stairway algorithm for the three first loops (Figs. 20.6, 20.7, 20.8, and 20.9). Table 20.4 shows the respective results using said algorithm.

The results of the proposed method supported in Table 20.4 suggest that the approach presented in this dissertation has a good performance of finding the stairway localization into the image plane. However, the difficulties with the distance estimation stage are due to the constraint in forward motion. Consequently, improvements to the performance were done by adding information about the camera rotation in the 3D world (such information is important because nonplanar surfaces affect the result).

20.4 Conclusions

This approach offers the following contributions. First, the research helps to define an approach for stairway detection. This information is important and necessary in order to localize stairways in unknown environments. It is also a fundamental step for the implementation of autonomous stair climbing navigation and passive alarm systems intended for blind and visually impaired people. Second, our results using the CNN approach are well over the 70% of identification, suggesting the possibility of assessing high-level features, by inferring the class label “banister.” This approach can be used in robust real-time identification of high-level features or can be used in an ensemble learning approach along with traditional methods. Third, these findings can support a case study in order to evaluate the applicability of this approach in real-time implementation.

Further changes can be made to the implementation by fine-tuning the last layers of network with training examples of stairs, which often results in higher probability accuracy. Also, other directions can be explored, such as the development of Multi-task Cascaded Convolutional Networks as has been proposed for face-like features [23], but for the detection of specific high-level feature of stairs.

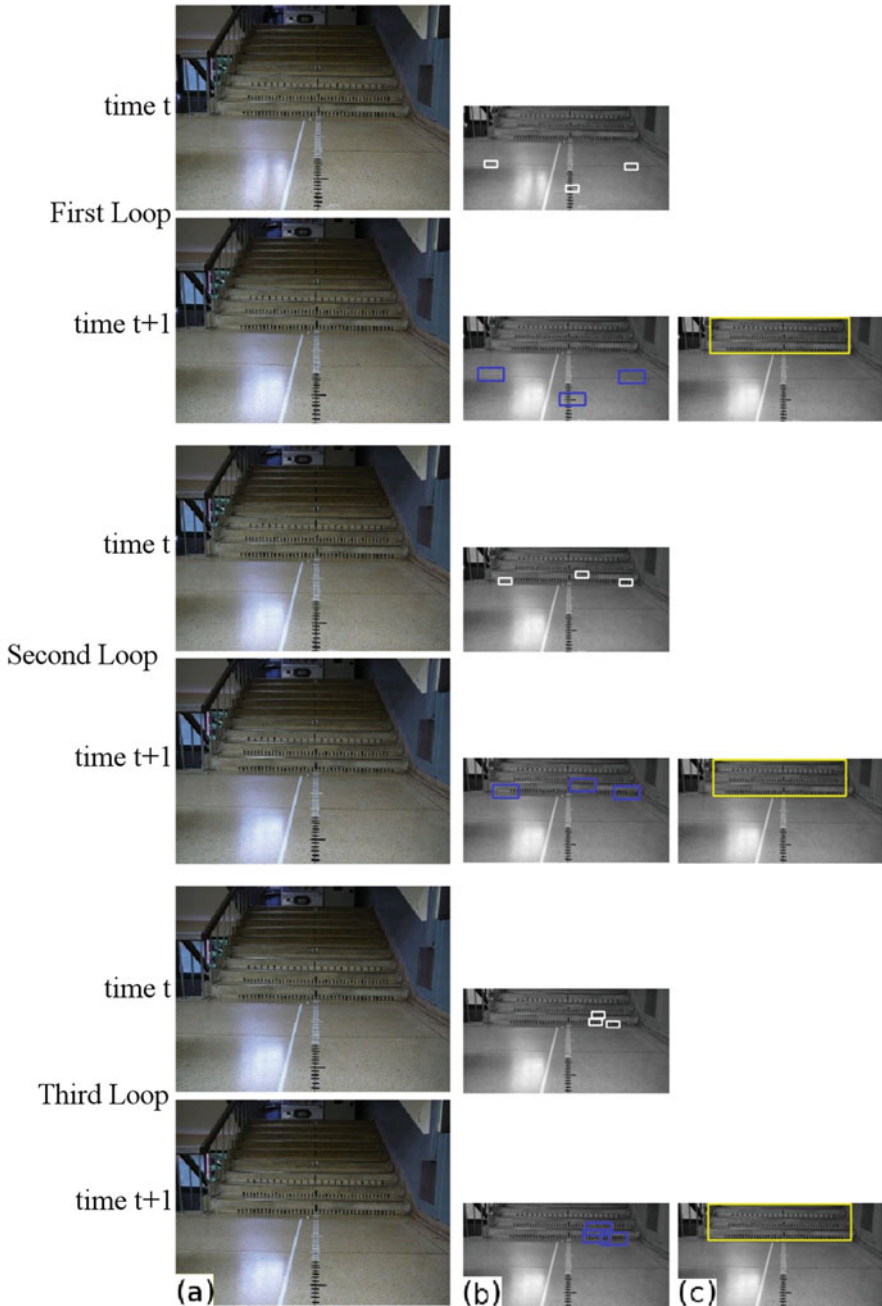


Fig. 20.6 Frontal image dataset at daylight. (a) Source image at time t and $t + 1$. (b) Extracting the area of interest for ROI_1 and ROI_2. The white block represents the ROI_1, using a 20×10 pixel size for each block extracted at time t ; the numbers show the order of extraction. The blue block represents the ROI_2, using a 40×20 pixel size for each block extracted at time $t + 1$. (c) The yellow block shows the stairway candidate area

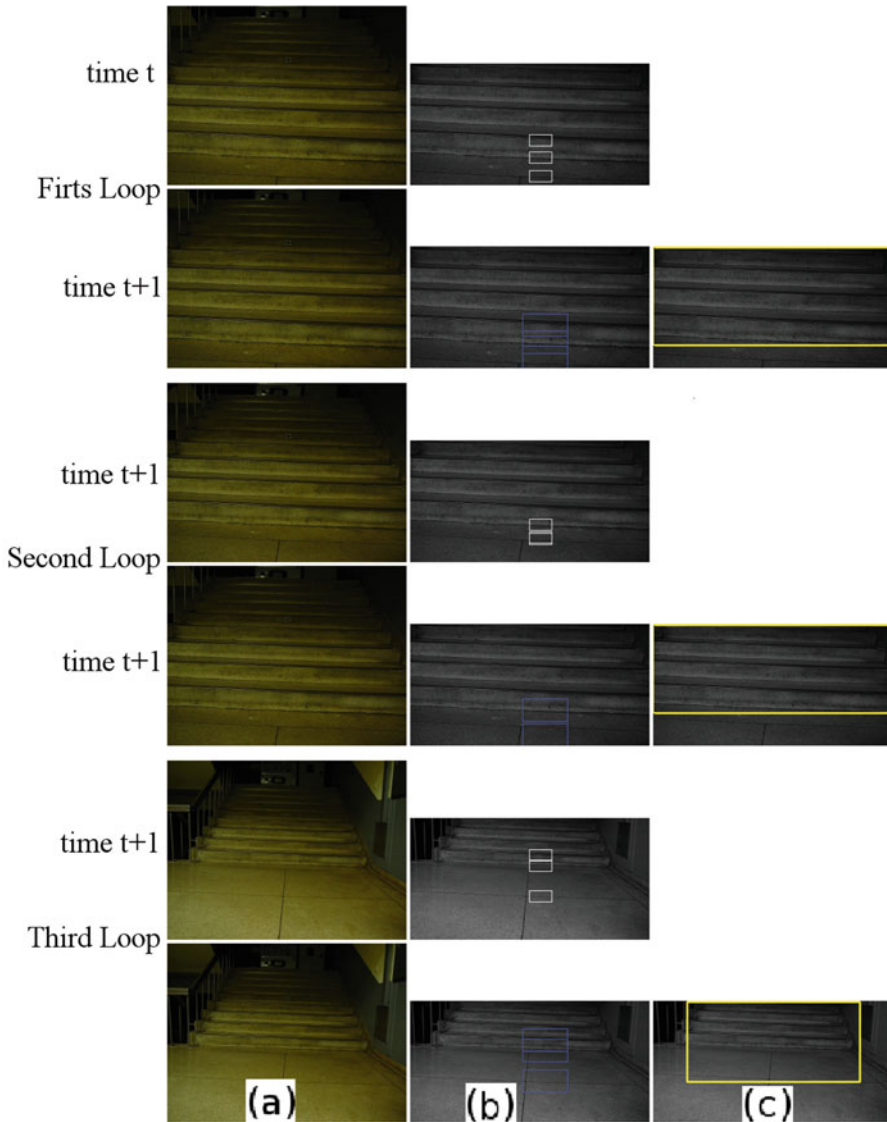


Fig. 20.7 Frontal image dataset at nightlight. (a) Source image at time t and $t + 1$. (b) Extracting the area of interest for ROI_1 and ROI_2. The white block represents the ROI_1, using a 20×10 pixel size for each block extracted at time t ; the numbers show the order of extraction. The blue block represents the ROI_2, using a 40×20 pixel size for each block extracted at time $t + 1$. (c) The yellow block shows the stairway candidate area

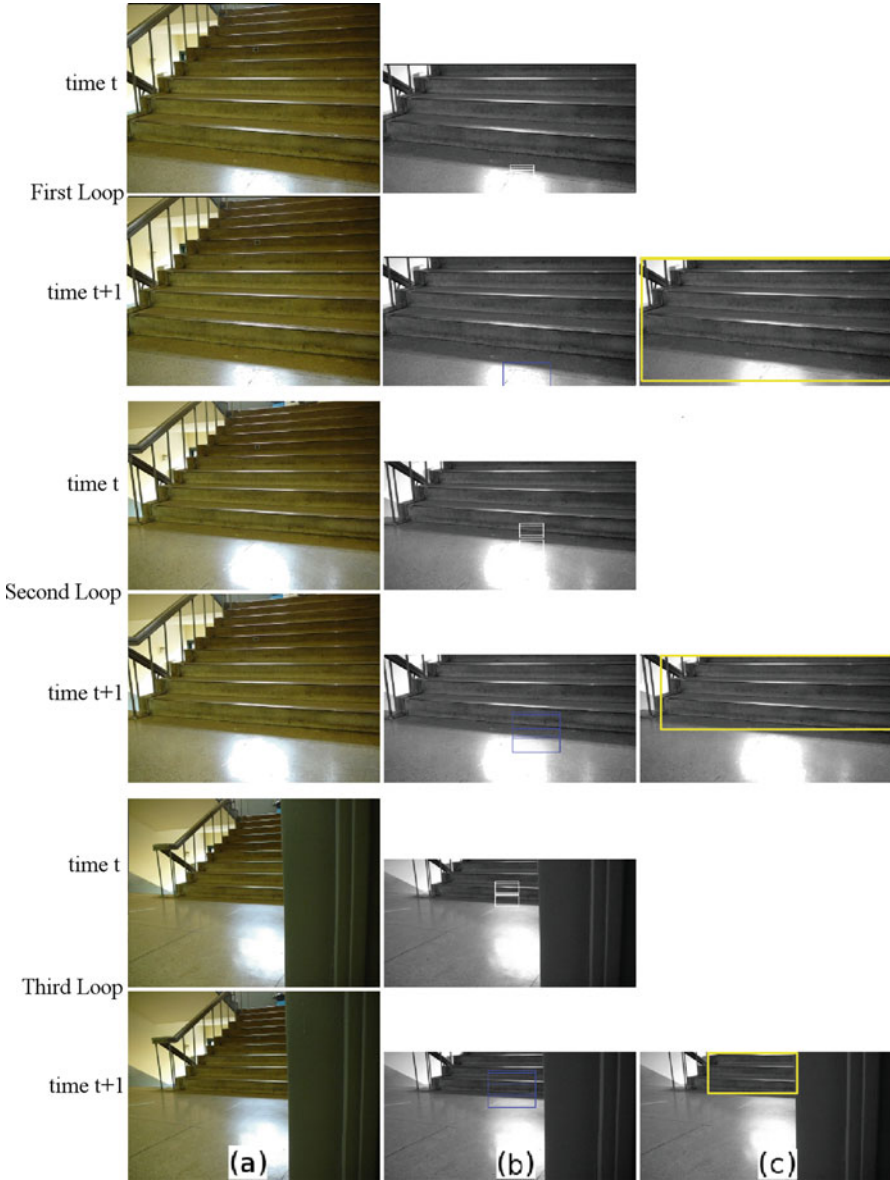


Fig. 20.8 Right image dataset. (a) Source image at time t and $t + 1$. (b) Extracting the area of interest for ROI_1 and ROI_2. The white block represents the ROI_1, using a 20×10 pixel size for each block extracted at time t ; the numbers show the order of extraction. The blue block represents the ROI_2, using a 40×20 pixel size for each block extracted at time $t + 1$. (c) The yellow block shows the stairway candidate area

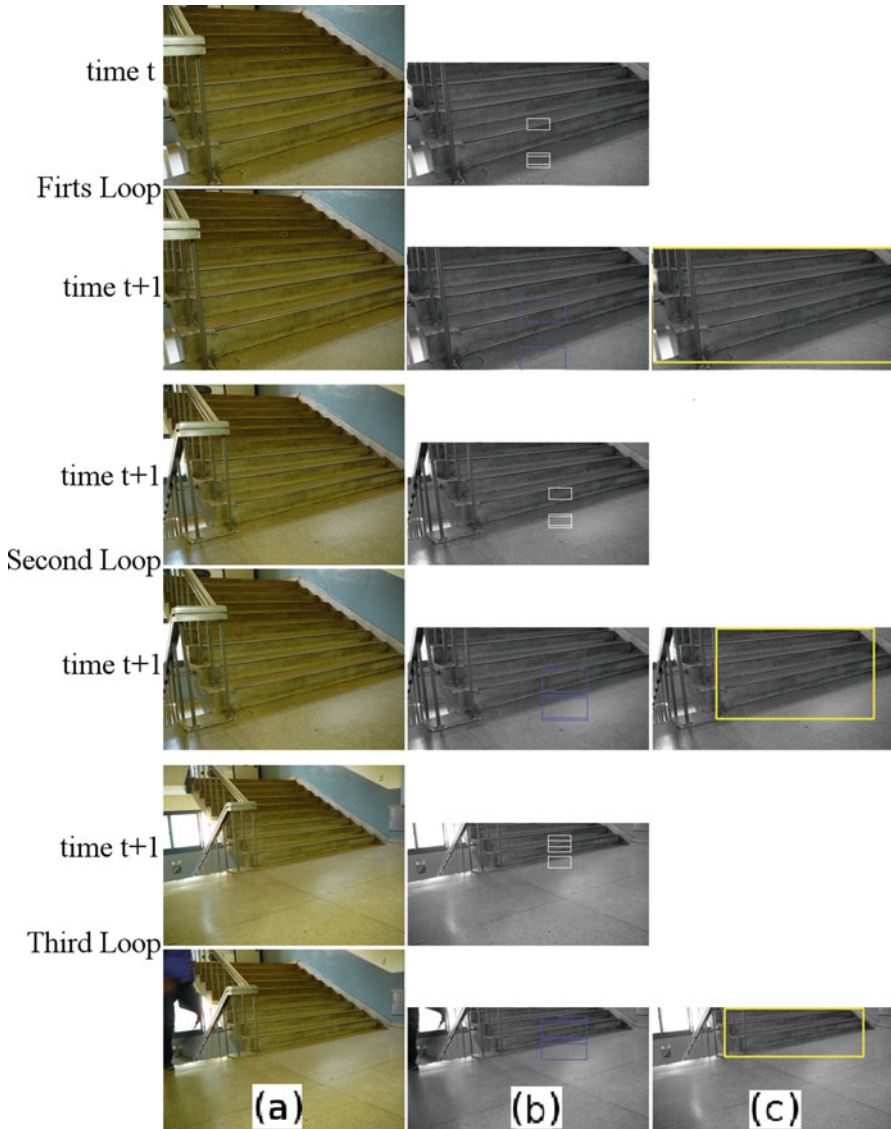


Fig. 20.9 Left image dataset. (a) Source image at time t and $t + 1$. (b) Extracting the area of interest for ROI_1 and ROI_2. The white block represents the ROI_1, using a 20×10 pixel size for each block extracted at time t ; the numbers show the order of extraction. The blue block represents the ROI_2, using a 40×20 pixel size for each block extracted at time $t + 1$. (c) The yellow block shows the stairway candidate area

Table 20.4 Localization of the stairway candidate used in Table 20.1

Data	Loop	RD (m)	DE (m)	RE (%)	<i>T</i> (ms)
Group 1	1	3.80	3.68	3.16	30
	2	3.75	3.74	0.27	30
	3	3.70	3.55	4.05	30
Group 2	1	2.20	1.94	11.95	30
	2	2.15	1.92	10.74	40
	3	2.10	1.86	11.43	30
Group 3	1	4.30	4.26	0.93	30
	2	4.25	3.95	7.06	30
	3	4.20	3.88	7.64	30
Group 4	1	4.30	3.67	14.65	20
	2	4.15	3.80	8.43	20
	3	4.20	3.74	-10.95	20

Note: This information is obtained by using the coordinate system of the camera. *rd* is the real distance in meter, *de* is the distance estimation, *re* is the relative error, and *t* is the computational time

Furthermore, the ensemble method of both convolutional neural network model description and feature extraction of stairways will be considered for future works.

Finally, the main contribution of this chapter is the fact that the proposed algorithm gave an estimate of information about the stairways, such as localization with respect to the coordinate system of the camera. This information is imperative and absolutely necessary in order to localize stairways in unknown environments.

Acknowledgments The authors would like to thank the National Bureau of Science, Technology and Innovation of Panama (SENACYT), the Sistema Nacional de Investigación (SNI) of Panama (SNI Contract 168-2017 and SNI Contract 129-2018), and the Universidad Tecnológica de Panamá for their administrative support and contribution.

References

1. Khalinluzzaman, M., & Deb, K. (2018). Stairways detection based on approach evaluation and vertical vanishing point. *International Journal of Computational Vision and Robotics*, 8(2), 168–189.
2. Schwarse, T., & Zhong, Z. (2015). Stair detection and tracking from egocentric stereo vision. In *IEEE International Conference on Image Processing (ICIP)*.
3. Wang, S., Pan, H., Zhang, C., & Tian, Y. (2014). RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs. *Journal of Visual Communication and Image Representation*, 25(2), 263–272.
4. Yang, C., Li, X., Liu, J., & Tang, Y. (2008). A stairway detection algorithm based on vision for UGV stair climbing. In *2008 IEEE International Conference on Networking, Sensing and Control*.
5. Lu, X., & Manduchi, R. (2005). Detection and localization of curbs and stairways using stereo vision. In *International Conference on Robots and Automation*.

6. Gutmann, J.-S., Fucuchi, M., & Fujita, M. (2004). Stair climbing for humanoid robots using stereo vision. In *International Conference on Intelligent Robots and System*.
7. Se, S., & Brady, M. (2000). Vision-based detection of stair-cases. In *Fourth Asian Conference on Computer Vision ACCV 2000*, Vol. 1, pp. 535–540.
8. Ferraz, J., & Ventura, R. (2009). Robust autonomous stair climbing by a tracked robot using accelerometer sensors. In *Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*.
9. Contreras, S., & De La Rosa, F. (2016). Using deep learning for exploration and recognition of objects based on images. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. <https://doi.org/10.1109/lars-sbr.2016.8>.
10. Poggi, M., & Mattocchia, S. (2016). A wearable mobility aid for the visually impaired based on embedded 3D vision and deep learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. <https://doi.org/10.1109/iscc.2016.7543741>.
11. Lin, B.-S., Lee, C.-C., & Chiang, P.-Y. (2017). Simple smartphone-based guiding system for visually impaired people. *Sensors*, 17(6), 1371. <https://doi.org/10.3390/s17061371>.
12. Yang, K., Wang, K., Bergasa, L., Romera, E., Hu, W., Sun, D., et al. (2018). Unifying terrain awareness for the visually impaired through real-time semantic segmentation. *Sensors*, 18(5), 1506. <https://doi.org/10.3390/s18051506>.
13. Bashiri, F. S., LaRose, E., Peissig, P., & Tafti, A. P. (2018). MCIndoor20000: A fully-labeled image dataset to advance indoor objects detection. *Data in Brief*, 17, 71–75. <https://doi.org/10.1016/j.dib.2017.12.047>.
14. Bashiri, F. S., LaRose, E., Badger, J. C., D'Souza, R. M., Yu, Z., & Peissig, P. (2018). Object detection to assist visually impaired people: A deep neural network adventure. In: Bebis G. et al. (eds) *Advances in Visual Computing. ISVC 2018. Lecture notes in computer science*, vol 11241. Springer, Cham, 500–510. doi:https://doi.org/10.1007/978-3-030-03801-4_44.
15. B. Zoph, V. Vasudevan, J. Shlens, and Q. Le. *Learning transferable architectures for scalable image recognition*. arXiv.org. Retrieved from <https://arxiv.org/abs/1707.07012.2019>
16. Krizhevsky, A., Sutskever, I. S., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>.
17. Hernández, D. C., & Jo, K.-H. (2010). Outdoor stairway segmentation using vertical vanishing point and directional filter. In *The 5th International Forum on Strategic Technology*.
18. Hernández, D. C., Kim, T., & Jo, K.-H. Stairway Detection Based on Single Camera by Motion Stereo. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Vol. 2, p. 11.
19. Barnard, S. T. (1983). Interpreting perspective images. *Artificial Intelligence*, 21(4), 435–462.
20. Weldon, T. P., Higgins, W. E., & Dunn, D. F. (1996). Efficient Gabor filter design for texture segmentation. *Pattern Recognition*, 29, 2005–2015.
21. Lee, T. S. (1996). Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10).
22. Basca, C. A., Brad, R., & Blaga, L. (2007). Texture segmentation. Gabor filter bank optimization using genetic algorithms. In *The International Conference on Computer Tool*.
23. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. <https://doi.org/10.1109/lsp.2016.2603342>.

Chapter 21

Advanced Phase Triangulation Methods for 3D Shape Measurements in Scientific and Industrial Applications



Sergey Vladimirovich Dvoynishnikov, Ivan Konstantinovich Kabardin,
and Vladimir Genrievich Meledin

Abbreviations

3D	Three dimensional
IT SB RAS	Institute of Thermophysics Siberian Branch of Russian academy of Science
RFBR	Russian Fund of Basic Research
RMS	Root mean square

21.1 Introduction

Methods of complex 3D object geometry measurement via the triangulation principle using phase triangulation and structured illumination are actively developed and improved [1]. A wide range of practical application of methods in such areas as mechanical engineering, medicine, biology, archeology, and modeling [2–8] is due to the low cost of the optical-electronic components of the system and its high reliability.

At the same time, modern development of phase triangulation methods focuses on different areas, including reducing measurement time [9, 10] to enable measurement of the geometry of moving objects [11, 12], developing fast and convenient calibration methods [13–15], and increasing the measurement accuracy using various methods and approaches [16].

Nevertheless, there are a number of problems associated with the complexity of using the existing phase triangulation methods for high-precision measurements

S. V. Dvoynishnikov (✉) · I. K. Kabardin · V. G. Meledin
Kutateladze Institute of Thermophysics SB RAS, Novosibirsk, Russia
e-mail: kabardin@itp.nsc.ru; meledin@itp.nsc.ru

under varying ambient light, narrow dynamic range of sources and receivers of optical radiation, limited depth of field image of a photodetector, and arbitrary light scattering of the measured object surface properties. A review of new methods of phase triangulation is presented. These methods allow measuring three-dimensional geometry under conditions of arbitrary measured object surface light scattering properties, varying measurement setting external illumination, and optical elements limited depth of field of the optical radiation source and receiver. There are five sections in the chapter. The first section presents the steady method for decoding phase images via arbitrary phase shifts. The second section describes the method for the optical radiation source–receiver path nonlinearity compensation in 3D measurements based on phase triangulation. The third section contains methods comparison for structured images decoding under conditions of nonlinearity of the optical radiation source–receiver path. The fourth section includes methods for expanding the dynamic range of phase triangulation measurements. The fifth section describes the method for estimating the spatial modulation optimal frequency in phase triangulation measurements.

21.2 The Steady Method for Decoding Phase Images with Arbitrary Phase Shifts

The intensity of the observed image at implementing the phase triangulation method may be described by the following expression:

$$I(x, y) = A(x, y) (1 + V(x, y) \cos \varphi(x, y)), \quad (21.1)$$

where $I(x, y)$ is the phase image intensity distribution; $A(x, y)$ is the distribution of background intensity; $V(x, y)$ is the average visibility; and $\varphi(x, y)$ is the desired distribution of the wave fronts phase difference. The illumination intensity in each point of the structured image is a function of three unknown parameters: background intensity $A(x, y)$, average visibility $V(x, y)$, and difference in phase between wave fronts $\varphi(x, y)$.

To decode phase images with arbitrary incremental shifts, there are several known approaches, based on solving a system of transcendental equations [17, 18]. Expression (21.1) in vector form is as follows:

$$I = AR + (AV \cos \phi) C + (AV \sin \phi) S, \quad (21.2)$$

where $R = (1, \dots, 1)^T$, $C = (\cos \delta_0, \dots, \cos \delta_{N-1})^T$, $S = (\sin \delta_0, \dots, \sin \delta_{N-1})^T$, and the vector dimensions can be determined by the quantity of phase shifts. It can be shown that

$$AV \sin \phi = \frac{I \cdot C^\perp}{S \cdot C^\perp}, \quad (21.3)$$

$$AV \cos \phi = \frac{I \cdot S^\perp}{C \cdot S^\perp}, \tag{21.4}$$

where S^\perp and C^\perp are the vectors orthogonal to vectors S, R and C, R , respectively. Given the properties of the scalar product, we obtain $S \cdot C^\perp = C \cdot S^\perp$. Then

$$\phi = \arcsin \frac{I \cdot C^\perp}{I \cdot S^\perp} \tag{21.5}$$

or

$$\phi = \arcsin \frac{I^\perp \cdot C}{I^\perp \cdot S}. \tag{21.6}$$

In the latter case, the vector I^\perp is only needed. The matrix operator $I^\perp = M \cdot I$ is an appropriate use in this case. The transformation matrix M must meet the following requirements: $(M \cdot I)I = 0$ and $M \cdot R = 0$.

For example, with three phase shifts, the skew symmetric matrix satisfies these conditions:

$$M = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}. \tag{21.7}$$

Then from (21.6), we obtain the following decoding algorithm:

$$\phi = \arcsin \frac{(MI) \cdot C}{(MI) \cdot S} = \arcsin \frac{(I_1 - I_2) c_0 + (I_2 - I_0) c_1 + (I_0 - I_1) c_2}{(I_1 - I_2) s_0 + (I_2 - I_0) s_1 + (I_0 - I_1) s_2}, \tag{21.8}$$

where $c_i = \cos \delta_i, s_i = \sin \delta_i$ are the corresponding components of vectors C and S .

The matrix M is obtained by symmetrically continuing the matrix (21.8) for an odd number of phase shifts larger than 3:

$$M = \begin{bmatrix} 0 & 1 & -1 & \vdots & 1 & -1 & 1 \\ -1 & 0 & 1 & \vdots & -1 & 1 & -1 \\ 1 & -1 & 0 & \vdots & 1 & -1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -1 & 1 & -1 & \vdots & 0 & 1 & -1 \\ 1 & -1 & 1 & \vdots & -1 & 0 & 1 \\ -1 & 1 & -1 & \vdots & 1 & -1 & 0 \end{bmatrix}. \tag{21.9}$$

With an even number of phase shifts, the matrix M may be represented as:

$$M = \begin{bmatrix} 0 & B \\ -B & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & 1 & \vdots & -1 & 1 \\ 1 & -1 & \vdots & 1 & -1 \\ \dots & \dots & \dots & \dots & \dots \\ -1 & 1 & \vdots & -1 & 1 \\ 1 & -1 & \vdots & 1 & -1 \end{bmatrix}. \quad (21.10)$$

With four phase shifts, we obtain the following algorithm:

$$\phi = \arctan \frac{(I_2 - I_3)(c_1 - c_0) + (I_1 - I_0)(c_2 - c_3)}{(I_2 - I_3)(s_1 - s_0) + (I_1 - I_0)(s_2 - s_3)}. \quad (21.11)$$

The described algorithm for decoding phase images with a step-by-step shift provides interpretation of phase images at arbitrary phase shifts. However, this method does not fully take into account the additive and multiplicative noise in phase patterns. Therefore, this method cannot minimize the error of phase determination in the presence of noise in the analyzed images. For practical application of this method, the authors, as a rule, use preliminary filtering of the initial phase images, or this method is used for a limited class of objects.

To solve the scientific and technical problem of measuring three-dimensional geometry of large-sized objects by triangulation methods with structured lighting, it is necessary to develop robust approaches for processing and decoding structured images. The authors propose a new method for decoding phase images that minimizes inaccuracy in phase calculation in structured images.

Expression (21.1) can be written in the form:

$$I_i = A + B \cdot \sin(\delta_i) + C \cdot \cos(\delta_i), \quad (21.12)$$

$$\varphi = -\arctan\left(\frac{B}{C}\right), \quad (21.13)$$

$$V = \frac{\sqrt{B^2 + C^2}}{A}. \quad (21.14)$$

The coefficients A , B , and C can be calculated from finding the functional minimum of the discrepancy between the experimental and theoretical data $S(A, B, C)$:

$$S(A, B, C) = \sum_{i=1}^N (I_i - A - B \cdot \sin(\delta_i) - C \cdot \cos(\delta_i))^2. \quad (21.15)$$

The minimization condition for $S(A, B, C)$ is the equality of all partial derivatives to zero:

$$\frac{\partial S}{\partial A} = 0, \frac{\partial S}{\partial B} = 0, \frac{\partial S}{\partial C} = 0. \quad (21.16)$$

As a result, we have the system of three linear equations:

$$\begin{cases} k_1 \cdot A + k_2 \cdot B + k_3 \cdot C = k_7 \\ k_2 \cdot A + k_4 \cdot B + k_5 \cdot C = k_8 \\ k_3 \cdot A + k_5 \cdot B + k_6 \cdot C = k_9 \end{cases}, \quad (21.17)$$

where $k_1 \dots k_9$ can be determined from the following equations:

$$\begin{aligned} k_1 &= N; \\ k_2 &= \sum_{i=1}^N \cos(\delta_i); \\ k_3 &= \sum_{i=1}^N \sin(\delta_i); \\ k_4 &= \sum_{i=1}^N \cos^2(\delta_i); \\ k_5 &= \sum_{i=1}^N \cos(\delta_i) \cdot \sin(\delta_i); \\ k_6 &= \sum_{i=1}^N \sin^2(\delta_i); \end{aligned} \quad (21.18)$$

$$\begin{aligned} k_7 &= \sum_{i=1}^N I_i \\ k_8 &= \sum_{i=1}^N I_i \cdot \cos(\delta_i); \\ k_9 &= \sum_{i=1}^N I_i \cdot \sin(\delta_i); \end{aligned}$$

Solving the system of linear Eq. (21.17), we obtain the following expressions for A , B , and C :

$$A = -\frac{k_5^2 \cdot k_7 - k_4 \cdot k_6 \cdot k_7 - k_3 \cdot k_5 \cdot k_8 + k_2 \cdot k_6 \cdot k_8 + k_3 \cdot k_4 \cdot k_9 - k_2 \cdot k_5 \cdot k_9}{-k_3^2 \cdot k_4 + 2 \cdot k_2 \cdot k_3 \cdot k_5 - k_1 \cdot k_5^2 - k_2^2 \cdot k_6 + k_1 \cdot k_4 \cdot k_6}, \quad (21.19)$$

$$B = -\frac{k_3 \cdot k_5 \cdot k_7 - k_2 \cdot k_6 \cdot k_7 - k_3^2 \cdot k_8 + k_1 \cdot k_6 \cdot k_8 + k_2 \cdot k_3 \cdot k_9 - k_1 \cdot k_5 \cdot k_9}{k_3^2 \cdot k_4 - 2 \cdot k_2 \cdot k_3 \cdot k_5 + k_1 \cdot k_5^2 + k_2^2 \cdot k_6 - k_1 \cdot k_4 \cdot k_6}, \quad (21.20)$$

$$C = -\frac{-k_3 \cdot k_4 \cdot k_7 - k_2 \cdot k_5 \cdot k_7 - k_2 \cdot k_3 \cdot k_8 + k_1 \cdot k_5 \cdot k_8 + k_2^2 \cdot k_9 - k_1 \cdot k_4 \cdot k_9}{k_3^2 \cdot k_4 - 2 \cdot k_2 \cdot k_3 \cdot k_5 + k_1 \cdot k_5^2 + k_2^2 \cdot k_6 - k_1 \cdot k_4 \cdot k_6}, \quad (21.21)$$

where φ is calculated from expression (21.13). Standard deviation of the measured intensity $S(A, B, C)$ and that of the phase $\sigma(A, B, C)$ can be estimated by the following

equations:

$$S(A, B, C) = \frac{1}{N} \sqrt{\sum_{i=1}^N (I_i - A - B \cdot \sin(\delta_i) - C \cdot \cos(\delta_i))^2}, \quad (21.22)$$

$$\sigma(A, B, C) = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\arccos\left(\frac{I_i - A}{\sqrt{A^2 + B^2}}\right) + \arctan\left(\frac{B}{C}\right) - \delta_i \right)^2}. \quad (21.23)$$

The method of determining the phase shift is based on the classical harmonic regression. The harmonic regression is a variation of the least squares method. The least squares method is based on minimizing the functional of the discrepancy between experimental and theoretical data by varying the desired parameters in a limited range of their possible values. Therefore, the proposed method for determining the phase shift gives a stable solution of the system of Eq. (21.3). That is, it guarantees the minimization of the phase calculation error even in the presence of noise, having zero expectation M and constant variance D in the sample corresponding to certain coordinates (x, y) :

$$M(x, y) = M(\{I_i(x, y)\}, i \in 1 \dots N) = 0 \quad (21.24)$$

$$D(x, y) = D(\{I_i(x, y)\}, i \in 1 \dots N) = \text{const} \quad (21.25)$$

From expression (21.13), it follows that phase φ does not depend on the distribution of background intensity A . That is why the proposed method minimizes the determination error φ in the presence of noise with constant expectation and variance.

A necessary and sufficient condition for determining the phase φ is the non-degeneracy of the system of linear Eq. (21.17) is:

$$\text{rank} \begin{pmatrix} N & \sum_{i=1}^N \cos(\delta_i) & \sum_{i=1}^N \sin(\delta_i) \\ \sum_{i=1}^N \cos(\delta_i) & \sum_{i=1}^N \cos^2(\delta_i) & \sum_{i=1}^N \cos(\delta_i) \cdot \sin(\delta_i) \\ \sum_{i=1}^N \sin(\delta_i) & \sum_{i=1}^N \cos(\delta_i) \cdot \sin(\delta_i) & \sum_{i=1}^N \sin^2(\delta_i) \end{pmatrix} = 3. \quad (21.26)$$

Since expression (21.26) depends only on δ_i and N , it determines the necessary and sufficient conditions for finding the solution of the system of Eq. (21.3) by the proposed method depending on the values of the introduced shifts δ_i and their number N .

The algorithmic complexity of the proposed method for determining the phase shift depending on the parameter N is the sum of the complexity of all the steps in

determining the phase. Given that there is a linear dependence on N in the formulas for coefficients, the complexity of the whole method can be estimated as $O(N)$.

Next, we verify the proposed method. To do this, we compare the result of the presented method for determining the phase shift based on harmonic regression and the generalized algorithm for decoding phase images based on the vector representation of expression (21.3) [19]. The result of the work based on the deviation of the phase measured value from the known initial value can be calculated as:

$$\varepsilon = \psi - \varphi, \tag{21.27}$$

where ε is the discrepancy between the measured phase and the original one; ψ is the phase value obtained using the corresponding method; and φ is the initial phase. The initial phase is set by simulating the typical intensity distribution in phase patterns. The shifts in the generation of phase patterns are calculated like a set of random variables with a uniform distribution in the interval $[0, 2\pi)$. Phase patterns are maiden with background intensity $A = 10$ and visibility $V = 0.5$. The phase pattern intensity distribution is set according to Eq. (21.2) with the addition of noise (Figs. 21.1 and 21.2). Noise has Gaussian distribution. The noise level is presented by the background intensity standard deviation.

Let us examine whether the error of both methods depends on the value of the found phase φ with constant noise and the same number of phase patterns N . To do this, we build the dependence of deviation ε between the measured phase value and the initial phase for different values of φ given in the interval $[0, 2\pi)$. The number of phase patterns $N = 50$. Standard deviation of the noise is 1% of the value of the background intensity A in expression (21.2).

Two hundred different sets of phase patterns were used for the reducement of the stochastic nature of the graphs. The calculated deviations of the measured phase maximum were recorded in the graph. Figure 21.3 shows the obtained dependences

Fig. 21.1 Dependence of intensity in the phase image on the phase shift without adding noise

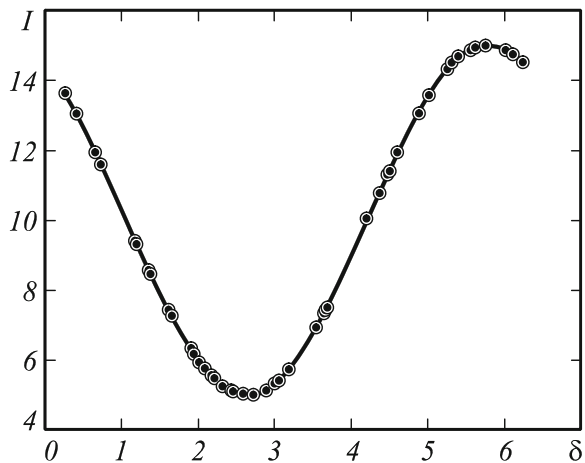


Fig. 21.2 Dependence of intensity in the phase image on the phase shift with added noise

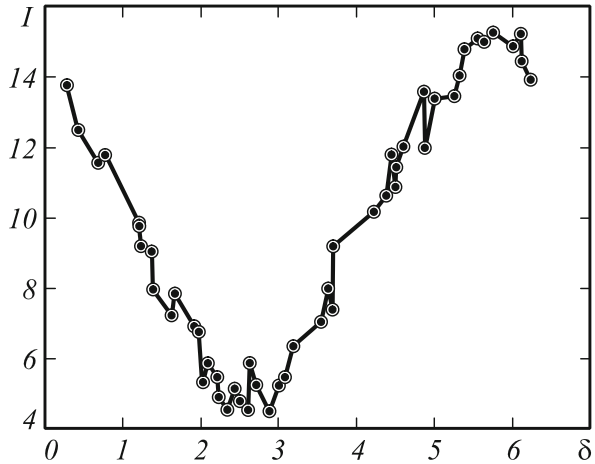
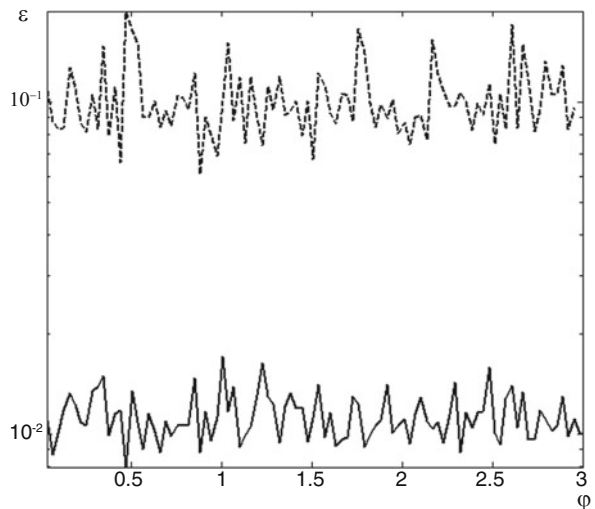


Fig. 21.3 Deviation of the phase, measured by the method of processing phase images, based on the vector representation of the system of equations (dashed line) and a steady method for determining the phase shift based on harmonic regression (solid line) at different values of the phase φ



of ε on φ . It can be seen that the proposed stable method provides at least an order of magnitude smaller deviation of the obtained phase, independent of the measured value of the initial phase φ .

In practice, phase patterns, as a rule, contain noises of additive nature. Therefore, it is useful to estimate the phase error, depending on the level of additive noise in phase images.

Let us estimate the deviation of the measured phase value from the initial ε , depending on the level of noise superimposed on the intensity distribution in the phase pattern with a constant number of shifts. Since the value of the initial phase φ does not affect ε , it is chosen equal to 0.5 radians. The number of phase patterns $N = 50$. H is noise standard deviation from the applied to the intensity distribution. H takes values in the range of 0–100% of the background intensity $A(2)$.

The resulting dependence of the measured phase deviation on the noise level is shown in Fig. 21.4. For the method of phase images processing, based on the vector representation of the system of Eq. (21.3), the error exceeds 100% with noise variance of more than 10%. The method of phase pattern processing based on the vector representation of the system of Eq. (21.3) gives unreliable results when the noise dispersion is over 10% of the background intensity. The proposed method for determining the phase shift based on harmonic regression provides an error of less than 50% for noise variance of less than 20%.

The number of implementations of phase images N is always limited in the experiment. The optimal number of realizations is required for decoding phase patterns with a given error. We conduct a comparative analysis of the methods in the case of phase recovery from a set of N phase image realizations. The error in determining the phase ψ is analyzed depending on the number N . The additive noise level was set constant with the standard deviation of 5% background intensity level. The phase value to be determined is unchanged and is set at 0.5 radians.

The results of the analysis methods are presented in Fig. 21.5. The error in determining the phase by an algorithm based on the vector representation of a system of transcendental equations does not qualitatively decrease with increasing N .

Even for small N ($N > 5$), the error of the method of decoding phase patterns based on the vector representation of the system of Eq. (21.3) is several times higher than that of the presented steady method based on harmonic regression. The simulation

Fig. 21.4 Deviation of the measured phase: the image processing method based on the vector representation of the system of Eq. (21.3) (dashed line) and a steady method for determining the phase shift based on harmonic regression (solid line)

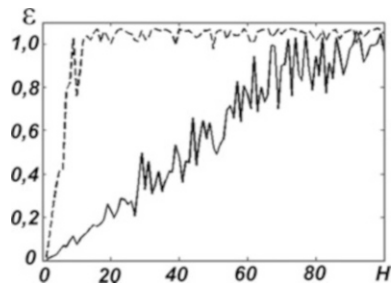
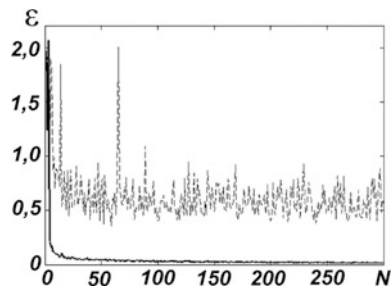


Fig. 21.5 Deviation of the measured phase from the number of shifts: the method of image processing based on the vector representation of the system of Eq. (21.3) (dashed line) and the steady method for determining the phase shift based on harmonic regression (solid line)



results generally demonstrate the stability of the proposed method to noise, its small error when working with a limited set of phase images and practical value.

Thus, a steady method for decoding the structured images has been proposed. It minimizes the measurement error of three-dimensional geometry in the presence of noise in the analyzed images. We consider a generalized algorithm that allows us to obtain a solution to the system of transcendental Eq. (21.3) with arbitrary phase shifts of the probing signal.

The stability of the method is shown in the presence of noise in a series of measurements of intensity of structured images at one point with constant variance and mathematical expectation. An estimate of the standard and measured deviations of the observed image is given to evaluate the reliability of the results. The necessary and sufficient condition for solving the problem by the proposed method is shown. The complexity of the algorithm is estimated depending on the number of photos with various structured images. A comparative analysis of the method presented in this work and the generalized algorithm for decoding phase images based on the vector representation of a system of transcendental equations is carried out. The results of the analysis demonstrate several times lower measurement error when working with a limited set of images.

The proposed method for processing images with spatially modulated phase illumination allows minimizing the error in determining the shift of the initial phase of the probing sinusoid. The received images of the measured object have the form of phase images with a stepwise shift of the initial phase of the probing sinusoid. The steady method for decoding the structured images allows minimizing the measurement error of three-dimensional geometry by the triangulation method with object surface arbitrary light scattering properties measured in a phase-inhomogeneous medium.

21.3 Method for Nonlinearity Compensation of the Source–Receiver Path of Optical Radiation in 3D Measurements Based on Phase Triangulation

The power characteristics of many input devices, printing, or visualization of images corresponds to a power law:

$$s = cr^\gamma, \quad (21.28)$$

where c and γ are the positive constants. Often, Eq. (21.28) is written as

$$s = c(r + \varepsilon)^\gamma, \quad (21.29)$$

in order to introduce the shift, that is, the initial brightness, when the photodetector input receives a zero optical signal. Devices used by the author as a source and

receiver of optical radiation also have a power type of energy characteristics. The graphs of dependences of s on γ for various values are shown in Fig. 21.6.

Most modern imaging devices have power dependence with an exponent ranging from 1.8 to 2.5. This trend originates from cathode ray tube monitors, in which the luminance brightness has power dependence on voltage. Figure 21.7 shows an image of a linear half-tone wedge, which is fed to the monitor input. The image on the monitor screen is darker than it should be.

Obviously, when using the method based on phase triangulation, it is necessary to control the linearity of the receiving-transfer path between the source of optical radiation and the image receiver. The presence of a nonlinear receiving-transfer characteristic of the path between the source and the receiver of spatially modulated illumination can cause difficult-to-predict systematic errors, which will depend on the magnitude of the phase shift (Fig. 21.8).

Now there are compensation methods that are based on serial or parallel compensating nonlinearity inclusion, introduction of the compensating nonlinear feedback and linear corrective devices synthesized on the invariance theory.

Fig. 21.6 Graphs for equations $s = cr^\gamma$, for different values of γ ($c = 1$ in all cases)

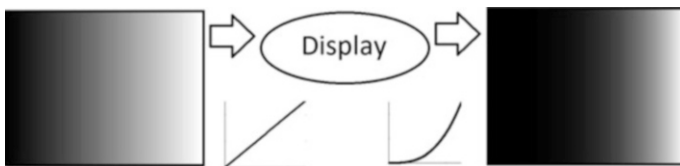
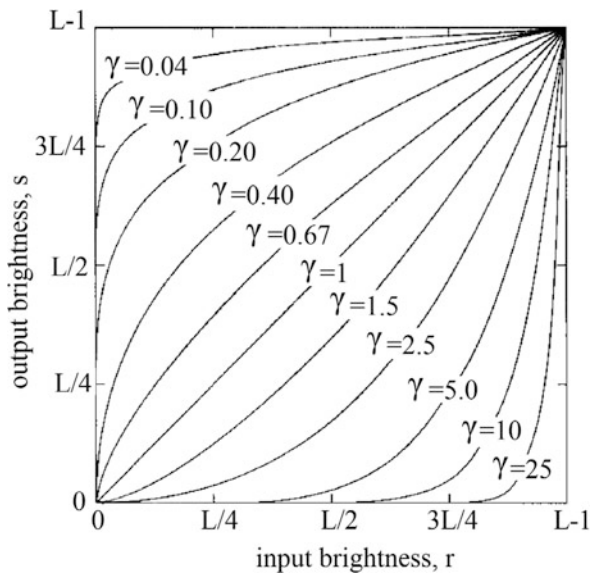
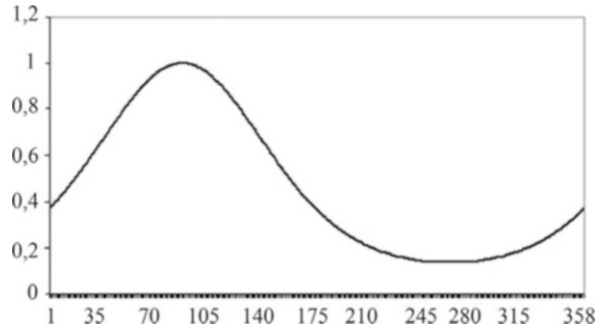


Fig. 21.7 Half-tone image with linear wedge (left) and monitor response to linear wedge (right)

Fig. 21.8 Type of sinusoidal signal after power conversion



The simplicity of implementation is advantage of sequential or parallel inclusion of compensating nonlinearity method. However, this method assumes the availability of information about the nonlinearity of characteristics of the receiving-transfer path, which in our case depends on the measured object reflective properties, external lighting, and the internal parameters of the optical radiation source and receiver. Therefore, the method of nonlinearity compensation on the basis of sequential or parallel inclusion of compensating nonlinearity is not applicable in our case.

The nonlinearity compensation method based on compensating nonlinear feedback is inappropriate to our problem. The feedback implies the presence of information about the distribution of structured illumination in the received images. The definition of illumination is the primary task of the optical triangulation method.

The method of modification by additional correction is proposed for compensation for the source–receiver path nonlinearity of optical radiation in 3D phase triangulation measurements.

The dependence of the intensity observed in the image on the intensity emitted by the light source can be represented by some nonlinear function K as follows:

$$I = K(U). \quad (21.30)$$

where U is the spatially modulated light intensity emitted to a small area of the investigated object; and I is the pixel intensity of the image the center of this small area is projected into.

If the object light scattering properties, the ambient lighting parameters, and the internal parameters of the structured illumination modulator and receiver do not change during the measurement process, the K function is identical for each pixel in the image. The inverse function K^{-1} can be calculated if K is smooth and continuous in the accepted values range.

Let us consider the nonlinearity compensation method of the optical radiation source–receiver path in 3D phase triangulation measurements. First, the calibration of the specified path is carried out. Then, to determine the function K , the measured object is lightened with a series of parallel halftone sinusoidal bands. The object

under study is consistently illuminated, providing uniform spatial modulation of the radiation source (the illumination intensity is uniform over the entire area of the radiator). The illumination intensity is changed linearly:

$$U^0(i) = U_0^0 + (i - 1) dU^0 \quad (21.31)$$

where i is the serial number of uniform illumination, $i = 1 \dots M$; U_0^0 is the intensity of the first implementation of illumination; and dU^0 is the increment step of illumination intensity.

The dependence of the intensity of the optical radiation source on the observed intensity in such an image is constructed for each point on the received images:

$$I(x, y) = K(x, y, U). \quad (21.32)$$

A function characterizing the nonlinearity of the source–receiver path of optical radiation is obtained. Then, the inverse function K^{-1} is built to restore the radiation intensity true value by the value of the registered intensity of the image at the point:

$$U = K^{-1}(x, y, I(x, y)). \quad (21.33)$$

It is possible to restore the intensity of the modulated optical radiation on the basis of the function K^{-1} , after obtaining investigation object images which are illuminated by parallel sinusoidal bands,

$$Y(x, y) = K^{-1}(x, y, I(x, y)). \quad (21.34)$$

Here $Y(x, y)$ is the light intensity distribution that is projected onto the measured object. The use of function $Y(x, y)$ instead of $I(x, y)$ in the decoding phase images method in phase triangulation eliminates the systematic phase measurement error of the testing sinusoid.

For the verification of the proposed method, we compare the results obtained by the steady decoding phase image method with and without compensation of the optical radiation source–receiver path. The results of the measured phase deviation from the known initial phase is estimated:

$$\varepsilon = |\psi - \varphi|, \quad (21.35)$$

where ψ is the phase found by compensation method.

Let us set the phase φ at initial position with taking into account typical intensity distribution in the phase images. Since the compared methods can work at arbitrary δ_t , shifts in the generation of phase images will have the format of an occasional set in the interval $[0, 2\pi)$. Phase images are formed with intensity of background $A = 10$ and visibility $V = 0.5$.

The intensity distribution in the phase images is set according to (21.28) with the addition of noise. Noise has random character with normal distribution. The level of noise will be estimated by the mean square deviation (RMS) from the intensity of background. We introduce a new parameter T : window width, where we will set different phase shifts in $(0, 2\pi)$ interval. At $T = 2\pi$, the phase shift can take all possible values. The introduction of the parameter T is due to the limitations of the photodetector range, which in the presence of non-switchable hardware and software adaptation automata can lead to uncertain results of intensity measurement in some areas of the formed phase shift values (see Fig. 21.9).

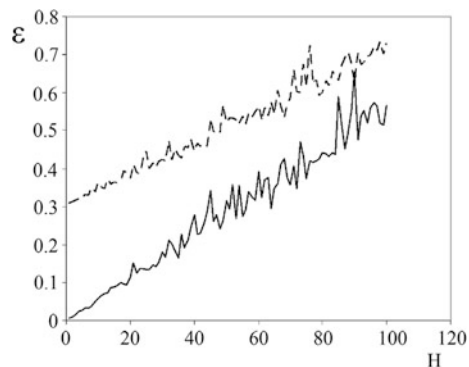
In practice, phase images, as a rule, contain noise of additive nature, imposed by the recording photodetector and elements of the receiving path. Therefore, it is useful to estimate the accuracy of phase determination depending on the level of additive noise.

Let us try to find the deviation ε of the phase from the initial one depending on the noise level imposed on the distribution of the intensity in the recorded phase images with known shift number and $T = 0.875\pi$. Recorded phase pattern number $N = 50$; parameter $\gamma = 1.5$; and RMS noise H imposed on the distribution of intensity is 0–100% of intensity of background. To decrease the stochastic character of the presented graphs, we use 200 different sets of recorded phase images and put the maximum acquired errors of the measured phase on the graph. Such procedure is followed in all the experiments described below.

The results presented in Fig. 21.9 show that in both cases the character of the phase error growth depending on the noise level $\varepsilon(H)$ in the interference patterns has a linear trend. However, at small values of noise level, this error taking into account the path nonlinearity compensation tends to zero, in contrast to the error of phase definition without compensation. If we do not compensate for the nonlinearity of the source–receiver path of optical radiation, then at $T < 2\pi$ the method of decoding phase images with arbitrary step-by-step shifts can give unreliable estimates.

Different sources and receivers of optical radiation have different parameters γ , determining the energy characteristics of the device according to (21.28). We

Fig. 21.9 Deviation of the value of phase from the level of noise in the distribution of intensity: the method of decoding images without nonlinearity compensation (dashed line) and that with nonlinearity compensation of the source–receiver path of optical radiation (solid line)



estimate the deviation ε depending on the level γ at constant number of shifts $N = 50$, noise level $H = 10\%$, and the parameter $T = 0.875\pi$.

Dependence $\varepsilon(\gamma)$ (see Fig. 21.10) shows that applying the method of the path nonlinearity compensation allows obtaining reliable values of the measured phase at any γ . In addition, from Fig. 21.11, it follows that in the vicinity of $\gamma = 1$, the compensation method cannot be used since in this vicinity the signal type after the power conversion remains unchanged (Fig. 21.12).

In experiments, the phase image realization number N is always limited. It is important to know the required number N for decoding phase images with a given accuracy. We perform a comparative analysis of methods in the case of phase recovery from a limited number of images N . We analyze the accuracy of phase determination depending on N (Fig. 21.10). The noise level will be a constant variable equal to 10% of background intensity of a standard deviation, the phase window $T = 0.875\pi$, and the parameter $\gamma = 1.5$. From graphs in Fig. 21.10, it follows that when using the method of decoding phase images without compensation for nonlinearity of the source–receiver path of optical radiation, the phase measurement error (i.e., deviation ε) decreases with increasing N , but converges to a value of about 0.34 radians, which is more than 10% of the measurement range. When compensating

Fig. 21.10 Deviation of the phase for various number of shifts: the method of decoding images without nonlinearity compensation (dashed line) and that with nonlinearity compensation of the source–receiver optical radiation path (solid line)

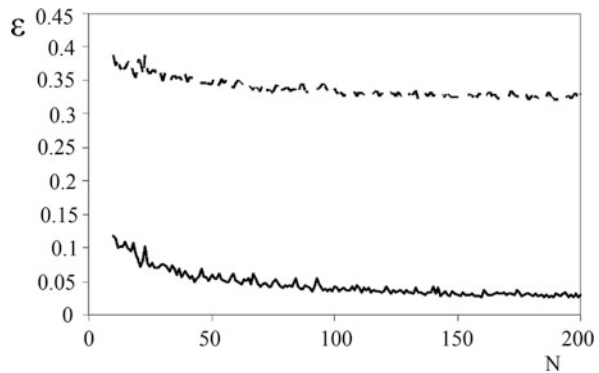


Fig. 21.11 Deviation of the phase from the value γ : the method of decoding images without nonlinearity compensation (dashed line) and that with nonlinearity compensation of the source–receiver path of optical radiation (solid line)

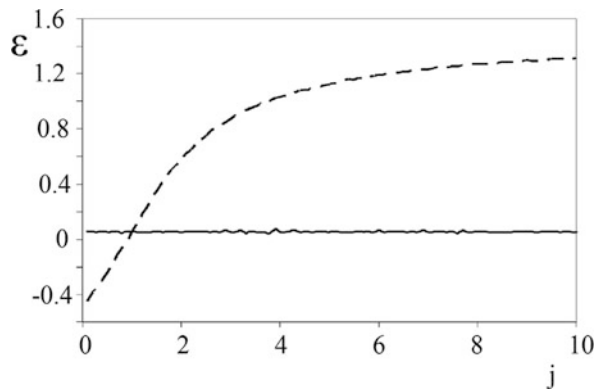
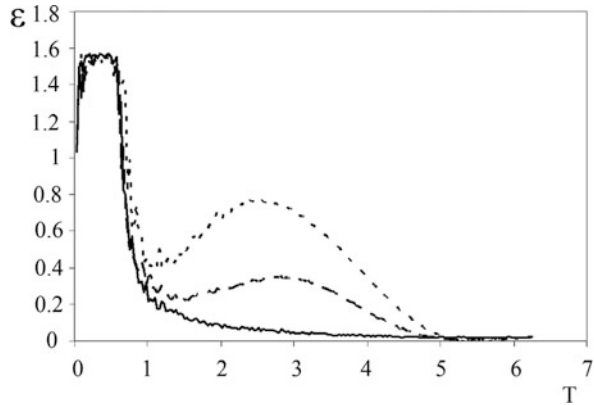


Fig. 21.12 Deviation of the measured phase at different window sizes T : the method of decoding phase images without compensation of the path nonlinearity $\gamma_1 = 1.5$ (large dotted line), $\gamma_2 = 2.5$ (small dotted line) and that with compensation for nonlinearity of the source–receiver path of optical radiation (solid line)



for the path nonlinearity, ε converges to a value of about 0.05 radians or 1.5% of the measuring range.

We estimate the phase measurement error for different window sizes T and constant values $\gamma_1 = 1.5$; $\gamma_2 = 2.5$; $H = 10\%$, $N = 50$. The results of estimations are shown in Fig. 21.10. At $T < 1$ rad, ε tends to 50% of the measurement range with and without nonlinearity compensation. At the given parameters H , N , and $T < 1$, the system (21.3) becomes degenerate and has no stable solution. At $T > 4.8$ radians, the error in determining the phase with and without compensation is almost the same. At $2 < T < 5$, the decoding phase image method without nonlinearity compensation leads to obtaining significant systematic errors.

Thus, the proposed method of nonlinearity compensation of the source–receiver path of radiation in 3D optic measurements based on structured light and phase triangulation minimizes the error of the phase measurements with arbitrary stepwise shifts in the presence of noise and power characteristics of the receiving-transfer devices. Comparison of the steady method of decoding interferograms and method proposed above shows that for T in the range $2 < T < 5$, the nonlinearity compensation improves the accuracy of phase triangulation at accidental phase shifts and random noise. The method of nonlinearity compensation allows to reduce the error in several times and to significantly increase the safety of 3D measurement results based on optical phase triangulation with structured light. It allows using modern inexpensive household devices, including those equipped with non-switchable hardware and software adaptation machines, as sources and receivers.

21.4 Comparing Methods of Structured Image Decoding at Nonlinearity of the Source–Receiver Path of Optical Radiation

In the measurement of a 3D profile using the phase triangulation in the images, there is often additive noise. In addition, most modern devices used to generate and input

images have an amplitude characteristic corresponding to the power law, often called “gamma correction” (21.28).

Several approaches are used to compensate for the nonlinearity of the source–receiver path of optical radiation in 3D measurements based on phase triangulation. The first is the use of a matched pair of the source and receiver of optical radiation, for which the nonlinearity of the transfer function is obviously absent, that is, $\gamma = 1$ in (21.28). This approach is found in specialized optoelectronic devices that use a single-frame decoding of a three-dimensional setting.

Another and more common approach is based on the application of the four-step phase triangulation method. The method works as follows. Four illuminations are projected on the surface of the measured object. The linear phase shift by adjacent images is $\pi/2$. Without taking into account the power transfer function, we obtain:

$$I_n(x, y) = I_b(x, y) + I_m(x, y) \cdot \cos\left(\varphi(x, y) + (n - 1) \frac{\pi}{2}\right), n = 1 \dots 4. \quad (21.36)$$

The phase $\varphi(x, y)$ can be calculated by the formula:

$$\varphi(x, y) = \arctan\left(\frac{I_4 - I_2}{I_1 - I_3}\right). \quad (21.37)$$

Next, we take into account the power transfer function in the form of a second-degree polynomial

$$I_n(x, y) = a_0 + a_1 S_n(x, y) + a_2 S_n^2(x, y) + \alpha, \quad (21.38)$$

where a_0, a_1, a_2 , and α are the coefficients and S is the received intensity of the phase image with power correction of brightness in accordance with expression (21.28). Given the properties of trigonometric functions, from expressions (21.35), (21.37), (21.38), the following can be obtained:

$$\varphi(x, y) = \arctan\left(\frac{S_4 - S_2}{S_1 - S_3}\right). \quad (21.39)$$

This method is resistant to additive noise and automatically compensates for nonlinear distortions in the measurement results. The disadvantage of the method is the need for all four measurements to fit in the dynamic range of the optical radiation receiver, which is not always possible. For example, in the case of measuring objects with complex profiles and arbitrary light scattering properties in a wide range of values, it is almost impossible to match the source and receiver of radiation. In practice, the used number of phase shifts is often larger than 4. In this case, the obtained phase images are analyzed in fours having successive shifts by $\pi/2$ relative to each other, and the results of the measured phase are averaged. The result will be a fairly accurate and reliable measurement method.

The third approach is more universal. It is based on the phase image decoding method, which allows discarding unreliable measurements and performing phase recovery in the images at an arbitrary set of phase shifts of the probing phase image [20]. The phase value $\phi(x, y)$ can be represented as

$$\phi(x, y) = \varphi(x, y) + \delta(x, y), \quad (21.40)$$

where $\delta(x, y)$ is the initial phase shift at the formed spatial illumination. Then the expression (21.2) can be presented as:

$$I(x, y) = I_b(x, y) + I_{\cos}(x, y) \cos(\delta) + I_{\sin}(x, y) \sin(\delta), \quad (21.41)$$

$$\varphi(x, y) = -\arctan\left(\frac{I_{\sin}(x, y)}{I_{\cos}(x, y)}\right). \quad (21.42)$$

The value of the phase $\varphi(x, y)$ is determined from the condition of minimization of the residual function between theoretical and experimental data:

$$S(I_b, I_{\sin}, I_{\cos}) = \sum_{i=1}^N I_i - I_b - I_{\cos} \cos(\delta_i) + I_{\sin} \sin(\delta_i), \quad (21.43)$$

$$\frac{\partial S}{\partial I_b} = 0; \quad \frac{\partial S}{\partial I_{\sin}} = 0; \quad \frac{\partial S}{\partial I_{\cos}} = 0. \quad (21.44)$$

This method requires direct compensation for nonlinearity of the source–receiver path of optical radiation. Otherwise, its application will lead to systematic deviations of the measured phase [21]. With the help of calibration of the source–receiver path of optical radiation, the transfer function is set in the form of dependence:

$$Y(x, y) = K^{-1}(x, y, I(x, y)), \quad (21.45)$$

where $Y(x, y)$ is the light intensity distribution that is projected onto the measured object. The use of function $Y(x, y)$ instead of $I(x, y)$ allows excluding the systematic measurement error of the phase of the probing sinusoid. Since the dependence $Y(x, y)$ automatically compensates for the background illumination $I_b(x, y)$, the expressions (21.43) and (21.44) will be reduced to the following form:

$$S(I_{\sin}, I_{\cos}) = \sum_{i=1}^N \left(K^{-1}(x, y, I_i(x, y)) - I_{\cos}(x, y) \cos(\delta_i) + I_{\sin}(x, y) \sin(\delta_i) \right), \quad (21.46)$$

$$\frac{\partial S}{\partial I_{\sin}} = 0; \quad \frac{\partial S}{\partial I_{\cos}} = 0. \quad (21.47)$$

This method is much more laborious, since it requires an additional procedure of calibration of the source–receiver path of optical radiation, but it is more versatile and reliable compared to the four-step method.

For its substantiation, it is required to analyze the error of phase determination in decoding phase images on the basis of iterative four-step method and steady method of decoding phase images with compensation of nonlinearity of the source–receiver path of optical radiation.

The main sources of measurement error of the phase triangulation method are noise on phase images and insufficient dynamic range of the optical radiation receiver. Since in the case of radiation intensity that does not fall within the dynamic range, the recorded intensity in the phase image is unreliable, such data will be discarded, and the phase shift will be calculated from the rest set of reliable measurements.

It is shown in [22] that the phase determination error can be estimated as:

$$\theta = \frac{\Delta I}{\sqrt{N} \cdot I}, \quad (21.48)$$

where N is the number of phase shifts, and $\Delta I/I$ is the relative error of the intensity measurement by the optical radiation receiver.

Below are the results of analysis of the phase measurement error in decoding phase images based on iterative four-step method and steady method of phase image decoding with compensation for nonlinearity of the source–receiver path of optical radiation. The analysis was performed at different noise levels in phase images, different values of the gamma correction coefficient, and different consistency parameters for sensitivities of the source and receiver of optical radiation.

Let the intensity of the radiation source scattered on the surface of the measured object vary in the range $[0 \dots 1]$ in relative dimensionless units. The operating range of the optical radiation receiver is $[a, b]$. Parameter a takes the values $[-1 \dots 1]$, and parameter b takes the values $[0 \dots 2]$. Figure 21.13 gives examples of the source intensity (a) and the observed receiver intensity (b, c, d) at different values of parameters a and b and at the noise level of 5% on the received images. Changes in all parameters allow estimating the measurement error at different light scattering properties of the measured object surface.

Figure 21.14 presents the theoretical error of phase measurement in the absence of noise in phase images. It is obvious that under ideal conditions the measurement error will be zero. In addition, the graph shows that in the area where the inequality $a > b$ is performed, the measurements lose their physical meaning because the dynamic range of the radiation receiver takes an incorrect value.

Figures 21.15, 21.16, 21.17, and 21.18 show estimates of the error of phase determining at the noise level of 5% in the phase images and Figs. 21.19, 21.20, 21.21, and 21.22 at a noise level of 10%. Each graph shows measurement errors by a four-step method (darker surface) and a steady method of decoding phase images (lighter surface). The vertical axis and the surface color indicate the standard deviation of the measured phase, and the horizontal axes show the values of parameters A and B, reflecting characteristics of consistency of the source and receiver of optical radiation at the measurement point. Errors are shown at parameter $\gamma = 0.25$, $\gamma = 0.5$, $\gamma = 1$,

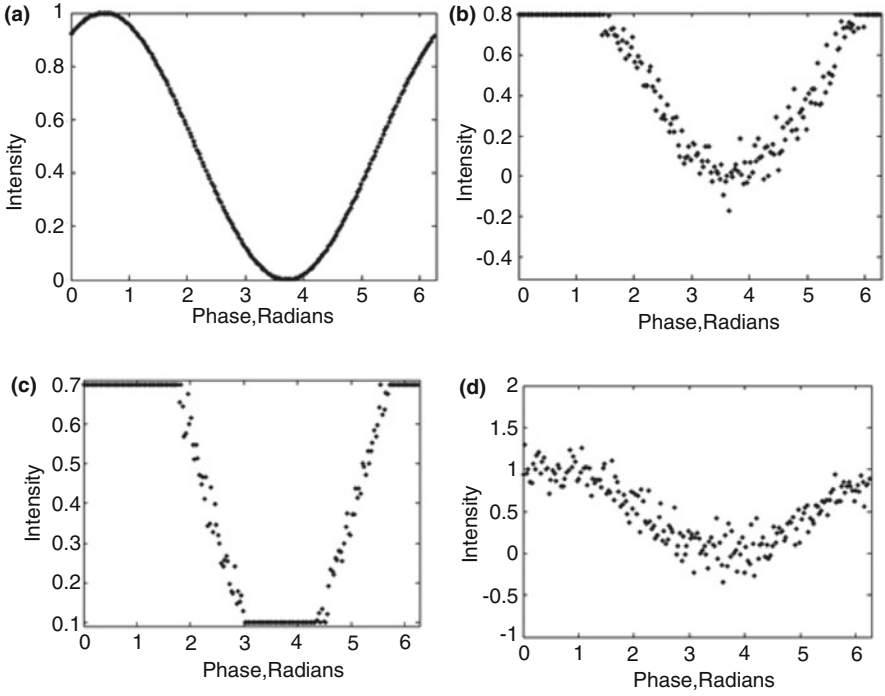


Fig. 21.13 Examples of intensity of the source (a) and the observed intensity of the receiver with various values of parameters a and b (b, c, d)

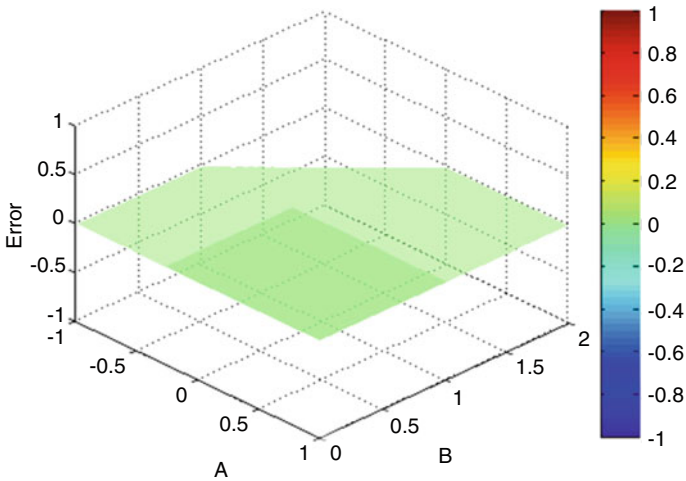


Fig. 21.14 An error in determining the phase from the consistency parameters of the receiver and source of radiation A and B in the absence of noise in the phase patterns

Fig. 21.15 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in phase images of 5%, $\gamma = 0.25$

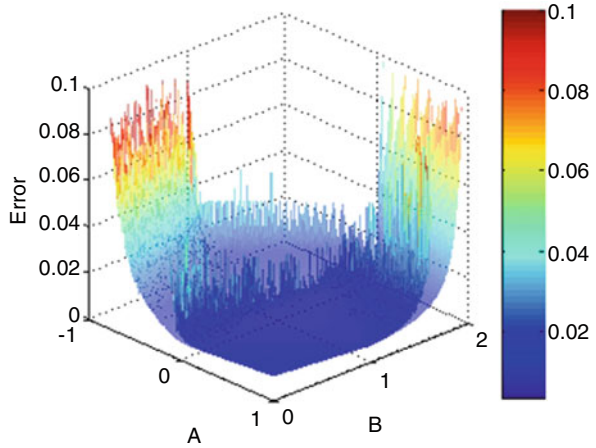


Fig. 21.16 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 5%, $\gamma = 0.5$

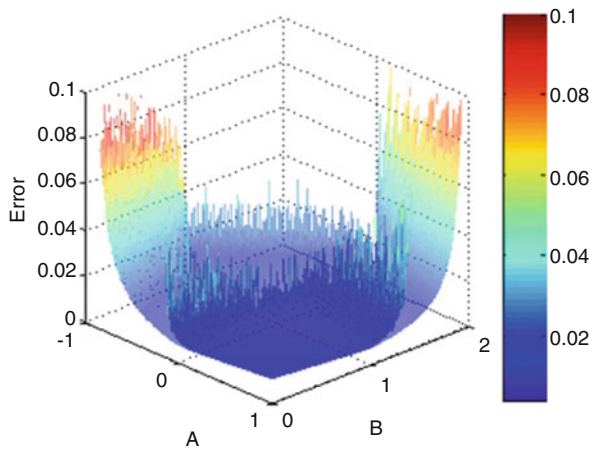


Fig. 21.17 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 5%, $\gamma = 1$

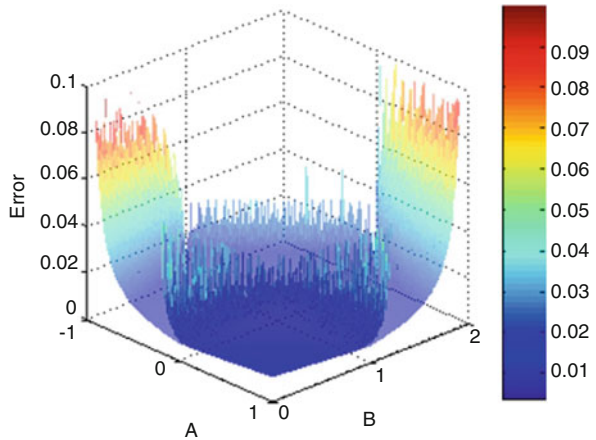


Fig. 21.18 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 5%, $\gamma = 2$

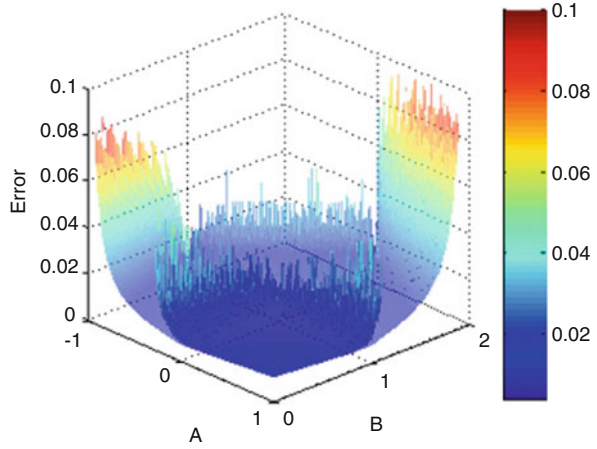


Fig. 21.19 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 10%, $\gamma = 0.25$

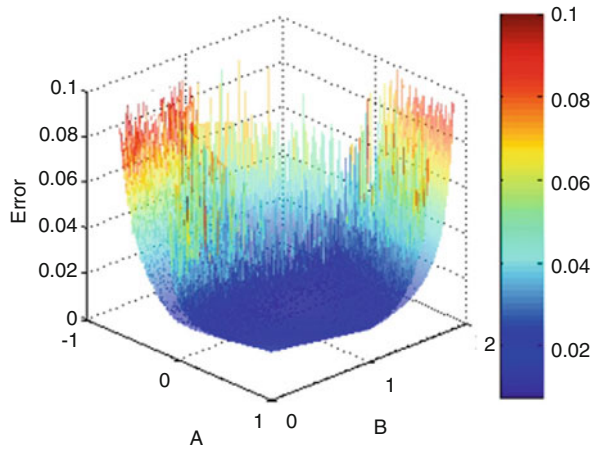


Fig. 21.20 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 10%, $\gamma = 0.5$

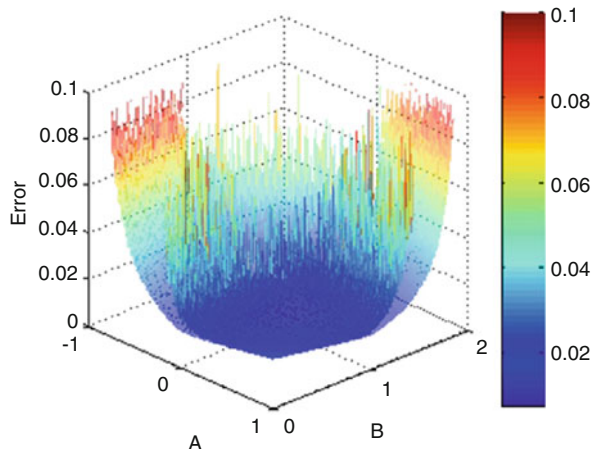


Fig. 21.21 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 10%, $\gamma = 1$

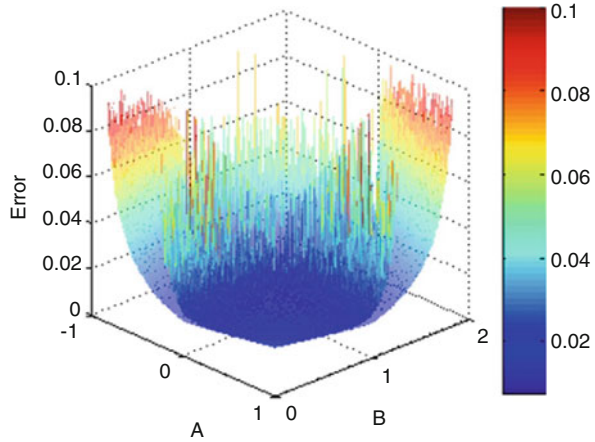
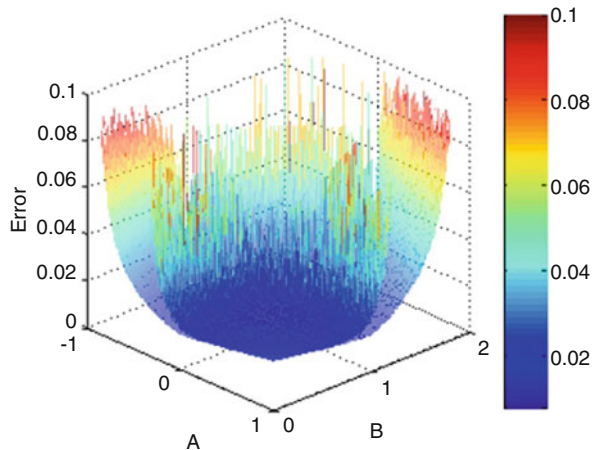


Fig. 21.22 An error of phase determination from the consistency parameters of the receiver and source of radiation A and B; noise level in the phase images of 10%, $\gamma = 2$



and $\gamma = 2$. The measurement results show that in the region where the operating range of the optical radiation receiver is consistent with the range of the emitted intensity, the error of the methods is approximately the same. When leaving the consistency area, the error of the four-step method increases significantly faster than that of the steady method of decoding phase images.

Figures 21.23, 21.24, 21.25, and 21.26 show three-dimensional surface sections (Fig. 21.22) at $\gamma = 2$ and the noise level of 10% in phase images. The results in the graphs show that for all the values of consistency parameters of the receiver and the radiation source, the method based on the steady decoding of phase images provides the measurement error that is, at least, not worse than that in the four-step method.

Figures 21.27 and 21.28 show the accuracy of phase determination by the four-step method and the steady method of phase image decoding at different consistency parameters of the receiver and source of radiation A and B, for small values of noise of 2% and 10% in the phase images. These graphs show that the error of the measured

Fig. 21.23 An error of phase determination from the consistency parameters of the receiver and source of radiation by four-step method (solid line) and steady method of phase image decoding (dashed line). Noise level in the phase images of 10%, $\gamma = 2, B = 1$

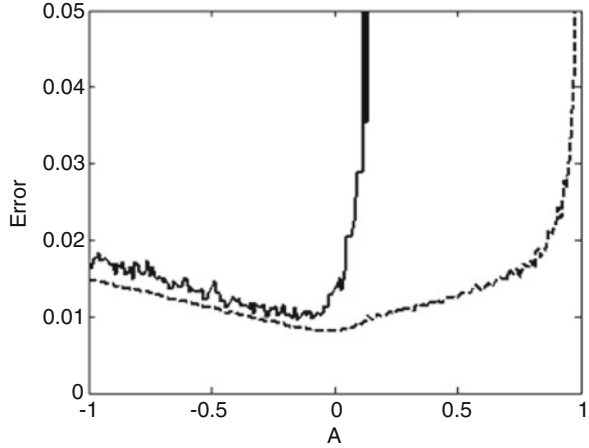


Fig. 21.24 An error of phase determination from the consistency parameters of the receiver and source of radiation by four-step method (solid line) and steady method of phase image decoding (dashed line). Noise level in the phase images of 10%, $\gamma = 2, B = 1.5$

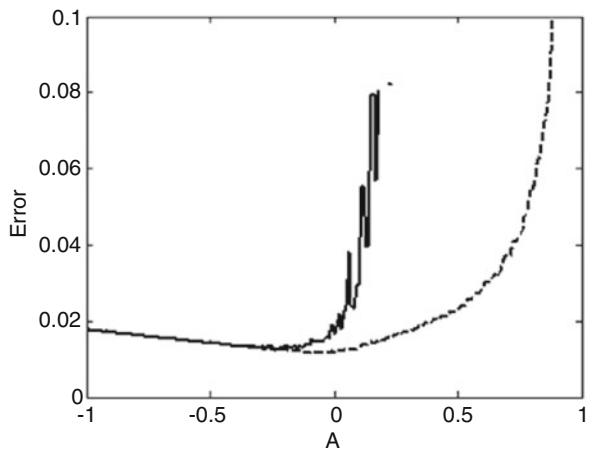


Fig. 21.25 An error of phase determination from the consistency parameters of the receiver and source of radiation by four-step method (solid line) and steady method of phase image decoding (dashed line). Noise level in the phase images of 10%, $\gamma = 2, A = 0$

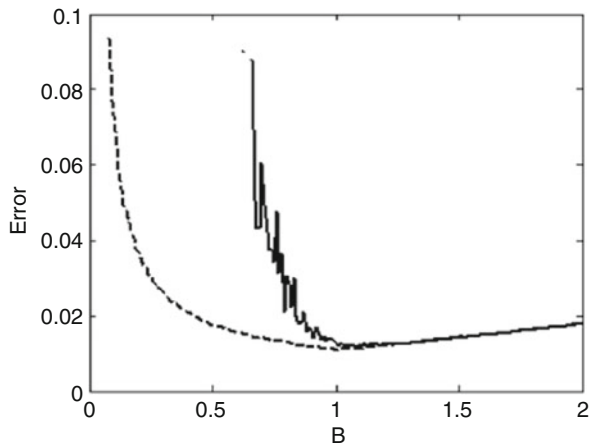


Fig. 21.26 An error of phase determination from the consistency parameters of the receiver and source of radiation by four-step method (solid line) and steady method of phase image decoding (dashed line). Noise level in the phase images of 10%, $\gamma = 2, A = -0.5$

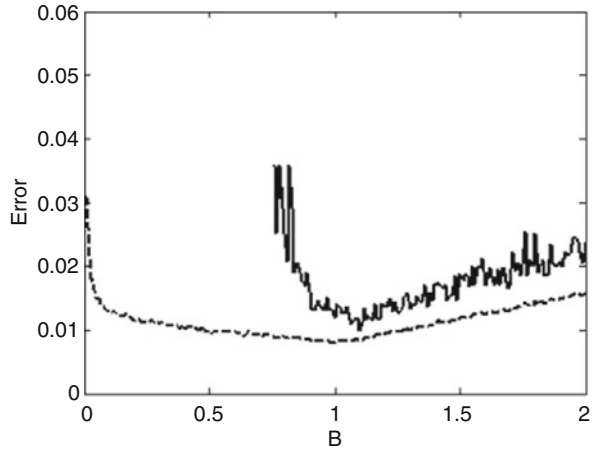


Fig. 21.27 An error of phase determination by four-step method (solid line) and steady method of phase image decoding (dashed line) from the consistency parameters of the receiver and source of radiation A and B at noise level in the phase images of 2%

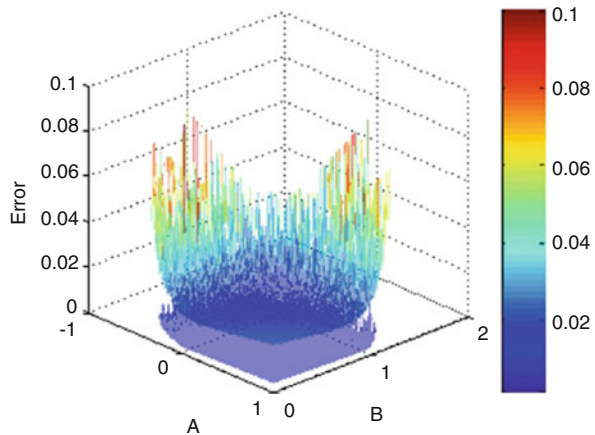
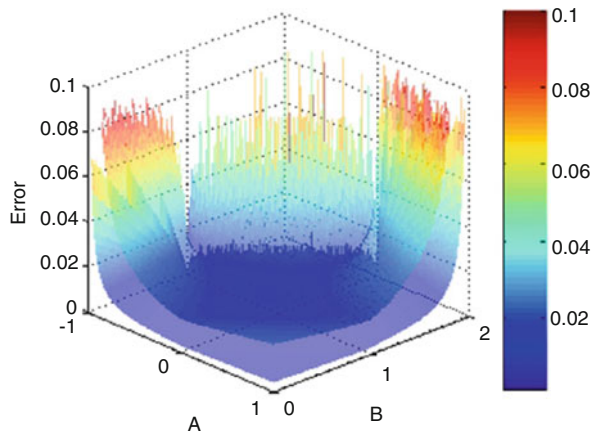


Fig. 21.28 An error of phase determination by four-step method (solid line) and steady method of phase image decoding (dashed line) from the consistency parameters of the receiver and source of radiation A and B at noise level in the phase images of 10%



phase depends on the noise level in the phase patterns. Moreover, in the field of consistency, both methods have almost identical level of error. This is confirmed by the graphs in Figs. 21.3 and 21.4.

Based on the results obtained and the experiments carried out, the following conclusion can be drawn. It is necessary to calculate the initial phase shift only by a set of reliable measurements when measuring by phase triangulation in a limited dynamic range of the receiver and with arbitrary light scattering properties of the measured object surface.

21.5 Methods for Expanding the Dynamic Range of Phase Triangulation Measurements

One of the disadvantages of the phase step method is the limited range of measured coordinates. Phase values over the image can be unambiguously restored only within the period.

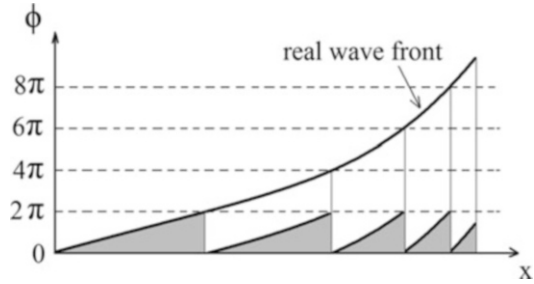
Currently, the problem of phase ambiguity has not been solved. There are many known phase field expansion algorithms [23], using known data on the investigated phase field to calculate the full phase, that is, full period number corresponding to the wave path difference, for example:

- Sign of phase change in the transition through the period
- Approximate values of the full phase obtained from other measurement sources and type of measured wave front (smoothness, continuity of derivatives)
- Change in the band color in interferometers with the same wave paths, change in the band contrast, etc.
- Difference in phases obtained from several measurements of one object with changed band value

Most of the phase ambiguity elimination algorithms are based on the analysis of the spatial structure of the phase field. The full phase is determined by its expansion, that is, by successive addition or subtraction of 2π to or from the phase value at the adjacent point, if the difference between them exceeds a certain threshold (Fig. 21.29). This procedure is based on the assumption that there are no sharp jumps (more than a period) at the points of transition through the period. To trace the transition boundary, the number of periods has to be an order of magnitude smaller than that of points in the detector array. This is only possible when analyzing a smooth phase front. The addition of 2π can be considered as an extrapolation process. At that phase shift at the previous restored points is considered for determining the phase shift at the following points. The hypothesis of the phase transition existence at some point of the phase field is accepted depending on the results of the analysis of its vicinity.

There is a known method for measuring the phase shift on interferograms using an equivalent wavelength. Information about the wavelength of light sources is used a priori.

Fig. 21.29 Wave front measured on module 2π



Optical differences of wave path at any point of the field (x, y) are determined from expressions:

$$\Phi_a + 2\pi n = \frac{2\pi}{\lambda_a} \text{OPX}, \tag{21.49}$$

at wavelength λ_a and

$$\Phi_b + 2\pi n = \frac{2\pi}{\lambda_b} \text{OPX}, \tag{21.50}$$

at wavelength λ_b . Deducting (21.50) from (21.49) and selecting the optical difference of wave path, we obtain:

$$\text{OPX} = \frac{\Phi_a - \Phi_b}{2\pi} \lambda_{\text{eq}} + (n_a - n_b) \lambda_{\text{eq}}, \tag{21.51}$$

where

$$\lambda_{\text{eq}} = \frac{\lambda_a \lambda_b}{|\lambda_a - \lambda_b|}. \tag{21.52}$$

Thus, it is possible to determine the phase front with a period equal to the equivalent wavelength λ_{eq} .

The above-mentioned methods of increasing the measurement range are not suitable for the problem of measuring 3D geometry of large objects based on the spatiotemporal modulation of the optical radiation source, since there is no a priori information about the 3D geometry of the measured object.

Below we consider the method of the full phase recovery using the integer analysis, which does not use a priori information about the measured object. The method is based on a series of measurements at different values of interference bands. The band value is determined by the difference in the optical wave path, in which the interference bands change for a period. The band value depends on the angle between the interfering beams, the transmittance of the medium, or the wavelength of the light source.

This work applies a modification of the full phase recovery method using the integer analysis.

The modification consists of the fact that the measured object is consecutively illuminated by a series of phase images with a multiple of the phase period. The phase transition is sought only within one period. Then, in the presence of a set of phase images with multiple periods and images where the phase shift does not transit through the period (this is determined by the depth of the measuring setting), it is possible to restore the value of the phase shift in the phase images with the smallest period.

Let the phase images be measured and the phase values are decoded for five fields that have multiple periods $N_1 \dots N_5$ of a probing sinusoid:

$$N_2 = 2N_1, N_3 = 2N_2, N_4 = 2N_3, N_5 = 2N_4. \quad (21.53)$$

Let $\varphi_1 \dots \varphi_5$ be phase values at one point in five fields corresponding to different periods of the probing sinusoid $N_1 \dots N_5$. Then the value of the resulting phase ϕ_{res} can be calculated by the following algorithm:

$$\phi_2 = \varphi_2 + 2\pi \cdot \text{INT} \left(\frac{(2 \cdot \varphi_1) - \varphi_2}{2\pi} \right), \quad (21.54)$$

$$\phi_3 = \varphi_3 + 2\pi \cdot \text{INT} \left(\frac{(2 \cdot \phi_2) - \varphi_3}{2\pi} \right), \quad (21.55)$$

$$\phi_4 = \varphi_4 + 2\pi \cdot \text{INT} \left(\frac{(2 \cdot \phi_3) - \varphi_4}{2\pi} \right), \quad (21.56)$$

$$\phi_{\text{res}} = \varphi_5 + 2\pi \cdot \text{INT} \left(\frac{(2 \cdot \phi_4) - \varphi_5}{2\pi} \right), \quad (21.57)$$

where function $\text{INT}(x)$ takes the following values:

$$\begin{cases} \text{INT}(x) = 1, & 0.5 \leq x < 1 \\ \text{INT}(x) = 0, & -0.5 < x < 0.5 \\ \text{INT}(x) = 1, & -0.5 > x \geq -1 \end{cases} \quad (21.58)$$

The obtained phase field ϕ_{res} provides a measurement range to correspond to the period of the probing sinusoid N_5 and the sensitivity to correspond to the period N_1 .

The proposed method of expanding the dynamic range of phase measurements provides an increase in the dynamic range to the limits due to the resolution of the source and receiver of spatially modulated optical radiation.

21.6 Method for Estimating the Optimal Frequency of Spatial Modulation in Phase Triangulation Measurements

In the phase triangulation method, the phase error depends on the number of phase images N and the relative error of the photodetector $\Delta I/I$. The measurement error of Z coordinate (setting depth), according to [24], can be estimated as follows:

$$\Delta z = \frac{\Delta\varphi \cdot p}{2\pi \cdot \tan \theta} = \frac{\Delta I \cdot p}{2\pi \cdot I \cdot \sqrt{N} \cdot \tan \theta}, \quad (21.59)$$

where p is the spatial modulation period of radiation, and θ is the angle of triangulation.

It follows from expression (21.59) that the error in determining the z coordinate is proportional to spatial modulation period of radiation. To minimize the measurement error by the phase triangulation method, it is necessary to minimize the spatial modulation period of optical radiation.

It is obvious that as the spatial modulation period of radiation decreases, the range of the setting depth measurement decreases as well. Over the image, the phase values can be unambiguously restored only within one period. To increase the measurement range by the phase triangulation method, various methods of phase field expansion are actively used in interferometry. There are known algorithms for phase field expansion, using a priori data on the object under study to determine the full phase, that is, the number of full periods [18]. There are also some known algorithms for restoring the full phase using integer analysis when the object is illuminated by phase image series with different multiples of the spatial modulation of radiation [19]. The most coming triangulation methods for 3D measurements using structured light illumination are the methods of expanding the measurement range using phase steps and the method of binary coding of pixels [25, 26]. These approaches provide the best measurement accuracy when projecting the least number of structured illuminations.

There are fundamental limitations on the resolution of the image formed by the optical elements of the measuring system. Due to the nonlinear distortion of the optical elements of the measuring system, the limited depth of field of the optical elements of the system, it is impossible to obtain an absolutely sharp image. Therefore, it is necessary to choose the frequency of spatial modulation of radiation based on the following considerations. First, the maximum frequency of spatial modulation in the received image must be less than the frequency of the equivalent low-pass filter, which is the optical system of the meter. Second, in order to achieve the minimum measurement error, the frequency should be maximized.

This chapter presents a method for estimating the optimal frequency of spatial modulation of radiation for three-dimensional measurements based on phase triangulation, which provides the smallest error in measuring a given depth z .

The dependence of the image on the photodetector on the intensity distribution formed on the object surface by the radiation source can be represented as a convolution of the pulse response function of the system and the intensity distribution function of the image formed on the surface of the measured object by the radiation source:

$$g(x, y) = \iint h(x - x_1, y - y_1) f(x_1, y_1) dx_1 dy_1 + n(x, y), \quad (21.60)$$

where g is the image formed on the photodetector, h is the pulse response of the optical system or the scattering function of the point source, f is the distribution function of the intensity of the image formed on the surface of the measured object by the radiation source, and n is the noise in the image. The noise function n in the image in addition to the noise of the photodetector includes the background brightness distribution of the measured object. Since the intensity of the generated illumination is significantly higher than the background brightness of the measured object and, especially, than the noise of the photodetector, the inequality is performed:

$$\iint \iint h(x - x_1, y - y_1) f(x_1, y_1) dx_1 dy_1 dx dy \gg \iint n(x, y) dx dy, \quad (21.61)$$

where integration is done throughout the image. In the frequency space, expression (1.19) takes the form:

$$G(u, v) = H(u, v) F(u, v) + N(u, v). \quad (21.62)$$

Since the formed structured illumination has a pronounced modulation direction (the radiation intensity is modulated along the selected, as a rule, horizontal coordinate), then we limit ourselves to considering the one-dimensional case.

A standard approach can be used to experimentally determine the pulsed response function of the optical system. Spatial low-frequency binary grid in the form of several wide white light lines is projected on the surface of the object. The photodetector detects the brightness distribution $G_0(u)$. The function $F_0(u)$ characterizes the intensity distribution on the surface of the measured object in the absence of noise and any optical distortion. The value of the function $F_0(u)$ is obtained from a priori information about the illumination formed on the surface of the measured object using the obtained function $G_0(u)$.

For example, the function $F_0(u)$ can be obtained as follows:

$$F_0(u) = \text{sign}(\phi_{\text{Low}}(G_0(u))), \quad (21.63)$$

where the function sign gives 1, if the value is positive, and -1 if it is not positive. The function ϕ_{Low} is the linear low-frequency filter whose cutoff frequency is obviously higher than the spatial frequency of the observed binary grid, projected onto the surface of the measured object.

Then the function H can be defined as:

$$H(u) = \frac{G_0(u) - N(u)}{F_0(u)}. \quad (21.64)$$

According to expression (21.59), the optimal spatial period of radiation modulation will be at the minimum value (p/I) or $(1/wI)$, where w is the frequency of spatial modulation of radiation, and I is the amplitude of the signal in the received images. Since the ideal infinite harmonic signal in the frequency representation is expressed by the delta-function, then

$$G_w(u) = H(u)\delta_w(u) = H(w), \tag{21.65}$$

$\delta_w(u)$ is the delta function equal to 1 at point w , and $G_w(u)$ is the intensity dependence formed on the photodetector at illumination in the form of a harmonic signal with frequency w . Then the amplitude of the harmonic signal with frequency w , observed in the images, will be proportional to the value of $H(w)$. The problem of determining the optimal spatial frequency of radiation modulation is reduced to determining the frequency w , at which $H(w) \cdot w \rightarrow \max$.

Since the noise frequency distribution $N(w)$ is unknown, it is impossible to calculate the function $H(w)$ using expression (21.64). Ignoring the noise in this case is impossible, because high-frequency components will inevitably increase when divided into high-frequency component of the “ideal” signal $F_0(u)$.

The following approach is used to estimate the function $H(w)$. The point blur function, which describes the dependence $H(w)$, has to sufficiently accurately repeat the normal distribution:

$$H(u) = Ae^{-\frac{u^2}{\sigma^2}}. \tag{21.66}$$

Then expression $H(w) \cdot w \rightarrow \max$ at

$$w = \frac{\sigma}{\sqrt{2}}. \tag{21.67}$$

From (21.64), we obtain:

$$\frac{G_0(u)}{F_0(u)} = Ae^{-\frac{u^2}{\sigma^2}} + \frac{N(u)}{F_0(u)}. \tag{21.68}$$

We further assume that the noise distribution $N(u)$ has a substantially smaller amplitude than $H(u)$. Then the expression $\frac{N(u)}{F_0(u)}$ in the low-frequency region will be significantly less than $H(w)$. Therefore, in the low-frequency domain, a function $\frac{G_0(u)}{F_0(u)}$ can be used to estimate $H(u)$.

Based on the assumption that the parameter A in expression (21.66) is equal to $H(0)$, and using the least squares method, we obtain:

$$\sigma = \frac{\int \sqrt{\log\left(\frac{G_0(0)}{F_0(0)}\right) - \log\left(\frac{G_0(u)}{F_0(u)}\right)} \cdot du}{\int u \cdot du}. \tag{21.69}$$

Here integration is carried out only over the low-frequency part of the spectrum. On the basis of expressions (21.67) and (21.69), we can obtain an estimate of the optimum frequency of the harmonic signal.

The practical implementation and verification of the proposed adaptive phase triangulation method for 3D measurements based on structured lighting were performed experimentally. NEC VT570 digital projector with a spatial resolution of 1024×768 was used as a source of spatially modulated radiation. Digital camera Logitech C910 with a resolution of 1920×1080 was used as a receiver of optical radiation. The projector formed illumination on the surface of the measured object in the form of a set of several light lines equidistant on the flat surface. The aim of the experiment was to determine the optimal spatial frequency of the projected signal for this measurement scheme based on the adaptive phase triangulation method.

As a result of the analysis of the recorded images on the surface of the measured object, the function $G(u)/F(u)$ has been obtained and the point blur function $H(u)$ has been found (Fig. 21.30). The optimal period of the probing signal for this measuring configuration has been found to be 38 pixels in our experiments.

Then a similar experiment was carried out, but with especially decreased sharpness of the optical recording system. The analyzed signals in the first and second experiments are shown in Fig. 21.31. It can be seen that the brightness signal fronts along the horizontal direction in the images in the second experiment are substantially tumbled down. The optimal period of the harmonic signal for this configuration of the optical measuring circuit should be significantly longer than in the first case.

The obtained function $G(u)/F(u)$ and the found point blur function $H(u)$ are presented in Fig. 21.32. The optimal period of the harmonic signal observed by the photodetector for this configuration of the optical system was 105 pixels.

The results shown prove the usefulness and availability of the proposed method for estimation of the best radiation modulation frequency for measuring three-dimensional geometry using phase triangulation and structured radiation. In the case of defocusing the optical elements of the measuring system, the optimal spatial modulation frequency of the radiation will be significantly lower than in the case of a well-coordinated and focused optical system of the meter.

Fig. 21.30 Dependence $G(u)/F(u)$ in the frequency representation (solid line) and the found blur function of the system point (dotted line)

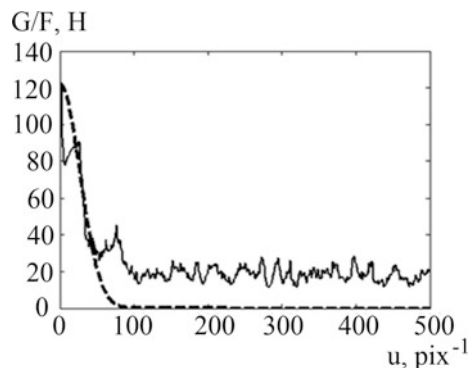


Fig. 21.31 Fragments of the analyzed intensity dependences in the case of a well-focused system (solid line) and in the case of a defocused system (dotted line)

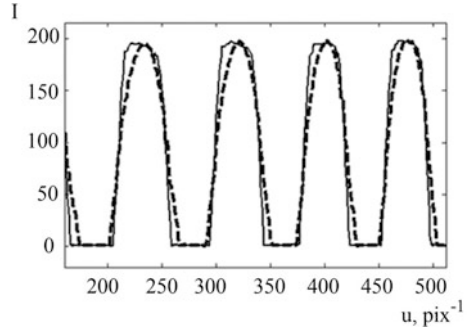
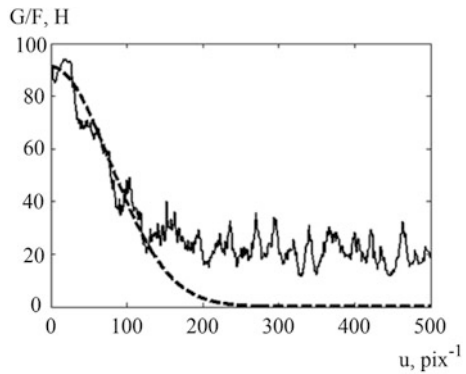


Fig. 21.32 Dependence $G(u)/F(u)$ in the frequency representation (solid line) and the found blur function of the system point (dotted line) for the case of defocused optical system



21.7 Conclusion

This section presents the methods of phase triangulation, providing higher metrological characteristics of measuring systems, as well as expansion of the functionality and range of applications of optoelectronic systems for geometric control in production conditions. The use of a steady method of phase image decoding will minimize the measurement error of three-dimensional geometry by phase triangulation using structured lighting. The method of nonlinearity compensation reduces the error several times and significantly increases the reliability of the results of 3D measurements based on phase triangulation and allows using modern inexpensive household devices, including those equipped with non-switchable hardware and software adaptation machines, as sources and receivers. The proposed method of expanding the dynamic range of phase measurements provides an increase in the dynamic range to the limits conditioned by the resolution of the source and receiver of spatially modulated optical radiation. The proposed method for estimating the optimal frequency of spatial radiation modulation for 3D measurements based on phase triangulation and structured lighting minimizes the error of phase determination for the used optoelectronic elements.

Acknowledgments This research was supported in part by RFBR (project No 18-08-00910) and was carried out under state contract with IT SB RAS.

References

1. Gorthi, S. S., & Rastogi, P. (2010). Fringe projection techniques: Whither we are? *Optics and Lasers in Engineering*, 48, 133–140.
2. D'Apuzzo, N. (2006). Overview of 3D surface digitization technologies in Europe. In *Proc. SPIE*, pp. 1–13.
3. Zhang, S. (2010). Recent progresses on real-time 3-D shape measurement using digital fringe projection techniques. *Optics and Lasers in Engineering*, 48(2), 149–158.
4. Lindner, L., Sergiyenko, O., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., Rodriguez-Quinonez, J. C., Murrrieta-Rico, F. N., Ivanov, M., Tyrsa, V., & Basaca, L. C. (2017). Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 44(4), 532–541.
5. Lindner, L. (2016). Laser scanners. In O. Sergiyenko & J. C. Rodriguez-Quinonez (Eds.), *Developing and applying optoelectronics in machine vision*. Hershey, PA: IGI Global. 38.
6. Lindner, L., Sergiyenko, O., Rivas-Lopez, M., Ivanov, M., Rodriguez-Quinonez, J., Hernandez-Balbuena, D., Flores-Fuentes, W., Tyrsa, V., Murrrieta-Rico, F. N., & Mercorelli, P. (2017). Machine vision system errors for unmanned aerial vehicle navigation. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on, Edinburgh*.
7. Lindner, L., Sergiyenko, O., Rivas-Lopez, M., Valdez-Salas, B., Rodriguez-Quinonez, J. C., Hernandez-Balbuena, D., Flores-Fuentes, W., Tyrsa, V., Medina Barrera, M., Murrrieta-Rico F., & Mercorelli, P. (2016). UAV remote laser scanner improvement by continuous scanning using DC motors. In *Industrial Electronics Society, IECON 2016, Florence*.
8. Lindner, L., Sergiyenko, O., Rivas-Lopez, M., Valdez-Salas, B., Rodriguez-Quinonez, J. C., Hernandez-Balbuena, D., Flores-Fuentes, W., Tyrsa, V., Medina, M., Murrrieta-Rico, F., Mercorelli, P., Gurko, A., & Kartashov, V. (2016). Machine vision system for UAV navigation. In *Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), International Conference on, Toulouse*.
9. Chen, L., Liang, C., Nguyen, X., Shu, Y., & Wu, H.-W. (2010). High-speed 3D surface profilometry employing trapezoidal phase-shifting method with multi-band calibration for colour surface reconstruction. *Measurement Science and Technology*, 21(10), 105309.
10. Lohry, W., & Zhang, S. (2014). High-speed absolute three-dimensional shape measurement using three binary dithered patterns. *Optics Express*, 22, 26752–26762.
11. Wissmann, P., Schmitt, R., & Forster, F. (2011). Fast and accurate 3D scanning using coded phase shifting and high speed pattern projection. In *Proceedings of the IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 108–115. IEEE.
12. Zuo, C., et al. (2013). High-speed three-dimensional shape measurement for dynamic settings using bi-frequency tripolar pulse-width-modulation fringe projection. *Optics and Lasers in Engineering*, 51(8), 953–960.
13. Zhang, S., & Yau, S.-T. (2007). Generic nonsinusoidal phase error correction for threedimensional shape measurement using a digital video projector. *Applied Optics*, 46(1), 36–43.
14. Zhang, S., & Huang, P. S. (2007). Phase error compensation for a 3-D shape measurement system based on the phase-shifting method. *Optical Engineering*, 46(6), 063601–063601-9.
15. Song, L., et al. (2015). Phase unwrapping method based on multiple fringe patterns without use of equivalent wavelengths. *Optics Communication*, 355, 213–224.
16. Armangue, X., Salvi, J., & Battle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7), 1617–1635.

17. Guzhov, V. I. (1995). Practical aspects of phase measurement in interferometry. *Avtometriya*, 5, 25–31.
18. Guzhov, V. I., & Solodkin, Y. N. (1992). Accuracy analysis of determination of phase total difference in integer interferometers. *Avtometriya*, (6), 24–30.
19. Indebetouw, G. (1978). Profile measurement using projection of running fringes. *Applied Optics*, 17, 2930–2933.
20. Takeda, M., & Mutoh, K. (1983). Fourier transform profilometry for the automatic measurement of 3-D object shapes. *Applied Optics*, 22(24), 3977–3982.
21. Bruning, J. H., Herriott, D. R., Gallagher, J. E., Rosenfeld, D. P., White, A. D., & Brangaccio, D. J. (1974). Digital wave-front measuring for testing optical surfaces and lenses. *Applied Optics*, 13, 2693–2703.
22. Dvoynishnikov, S. V., Kulikov, D. V., & Meledin, V. G. (2010). Optoelectronic method of contactless reconstruction of the surface profile of complexly shaped three-dimensional objects. *Measurement Techniques*, 53(6), 648–656.
23. Takeda, M., & Yamamoto, H. (1994). Fourier-transform speckle profilometry: Three-dimensional shape measurements of diffuse objects with large height steps and/or spatially isolated surfaces. *Applied Optics*, 33(34), 7829–7837.
24. Gruber, M., & Hausler, G. (1992). Simple, robust and accurate phase-measuring triangulation. *Optik*, 3, 118–122.
25. Inokuchi, S., & Sato, K., et al. (1984). Range-imaging system for 3-D object recognition. In *Proceeding of 7th International Conference Pattern Recognition, Montreal, Canada*, pp. 806–808.
26. Stahs, T., & Wahl, F. (1992). Fast and versatile range data acquisition. In *IEEE/RSJ International Conference Intelligent Robots and Systems, Raleigh, NC*, pp. 1169–1174.

Chapter 22

Detection and Tracking of Melt Pool in Blown Powder Deposition Through Image Processing of Infrared Camera Data



Sreekar Karnati and Frank F. Liou

Abbreviations

AM	Additive manufacturing
BPD	Blown powder deposition
CCD	Charge coupled device
CNC	Computer numerical control
IR	Infra-red
JSR	Just solidified region
LoG	Laplacian of Gaussian

22.1 Introduction

Additive Manufacturing (AM) is a component fabrication methodology where the desired geometries are built by selectively adding material in a layer by layer fashion. This approach is complementary to conventional subtractive fabrication methodologies where the desired geometries are machined to shape by removing material from a starting block of material. AM has been theorized to offer cost-effective means to fabricate complex geometries which are otherwise challenging to produce. Through various implementations, AM processes are being developed and commercialized for several materials such as plastics, metals, ceramics, and even organic materials. The avenues offered through AM are often cited to be exciting and groundbreaking.

S. Karnati (✉) · F. F. Liou
Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA
e-mail: skw92@mst.edu; liou@mst.edu

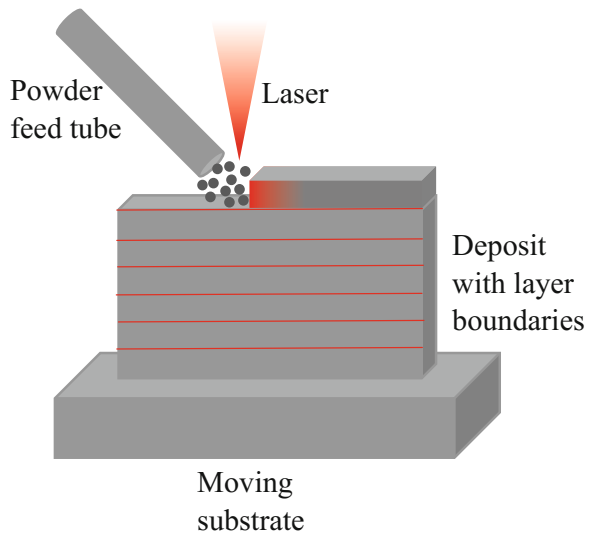
Among the different modalities of AM, metal-based AM is of great interest to the industries such as bio-medical [1], aerospace [2], defense [3], automobile [4], and construction industries [5]. AM is known to provide cost-efficient means of fabricating components with low buy to fly ratios and shorter lead times. The capability of AM to fabricate complex geometries can eliminate the need for assembly and thereby improve manufacturing and functioning efficiencies. AM has also been used to repair, recondition, and remanufacture existing components [6]. Such a revival of the end of life components can lead to immense cost savings and thereby growth in the industry [7].

While the benefits of AM are obvious, the process is very complicated and the outcome from the process is dependent on various factors. Depending on the modality, the factors of influence are subject to change. Metal-based AM modalities differ in terms of power source, the form of feedstock, types of process variables, and capabilities [8]. For the sake of clarity and brevity, discussion in this chapter will be limited to blown powder deposition.

Blown Powder Deposition (BPD) is a variant of laser metal deposition [8]. In this process, a moving melt pool is created on a substrate material by the use of a laser and a motorized worktable. A stream of powder is introduced into the melt pool through a feed tube; this powder is melted and then consolidated on to the substrate material. By moving the melt pool in a pre-planned fashion, the material is deposited in a layer by layer sequence. Upon successful deposition of such layers, the components of required geometries are fabricated. A schematic diagram of BPD is shown in Fig. 22.1.

The quality and properties of components obtained through BPD are dependent on process variables such as laser power, scan speed, powder feed, and layer thickness [9–11]. Creating and sustaining a melt pool capable of producing material with

Fig. 22.1 A schematic diagram of a front view of a BPD process indicating the relative positions of the laser beam, the powder feed tube, and the substrate. The red lines indicate the layer boundaries from the preplanned deposition path



required properties while ensuring successful fabrication of the required geometry is a challenging problem [12]. In the aim of understanding the influence of process parameters on material properties, researchers have mostly relied on experimentation and subsequent testing [10]. The composition of the material under discussion also influences the interaction and influence of the process parameters. This forces researchers to repeat similar studies on every new material.

Analysis from destructive testing of deposits only yields information on the final outcome and leaves in-process phenomenon to speculation. In order to realize the in-process phenomenon, researchers have often resorted to numerical analysis such as finite element modeling [13], cellular automaton [14], molecular dynamics [15], and phase field modeling [16]. Researchers have used such techniques to model, cooling rate [17], melt pool characteristics [18], temperature profile microstructure [19, 20], residual stresses [21], etc. In order to establish the validity of these models, the attributes of deposits such as melt pool dimensions, microstructure features, residual strain, and the temperature during deposition have been used. While such approaches have known to be successful, wide-scale incorporation of these models is not feasible. Often times these studies are limited in their scope due to assumptions necessary to facilitate successful calculation, loss of relevancy due to change in setup or limitations on computational resources.

Vision-based real-time monitoring of the deposition processes and closed loop control is a potential solution for realizing in-process phenomenon and performing a controlled deposition process. Real-time monitoring and potential closed-loop control of AM processes have been a topic of study for a while. Song et al. have successfully developed a system of three CCD cameras and a pyrometer for real-time tracking and control of deposition height. While doing so they were also able to measure and control the melt pool temperature [22]. The required control actions were performed by manipulating the power value during deposition. Similar process monitoring techniques have been developed for another localized heating process, i.e., welding. Real-time monitoring and correction during plasma welding and laser welding were achieved to study weld pool diameter, the surface of the weld pool, the weld plume size, etc. Kovacevic et al. utilized a CCD camera to capture a laser illuminated weld pool for information on surface detail. By its study, they were able to perform real-time correction during the process [23]. This camera was configured to capture light in the wavelength range of the irradiating laser. Manipulation of arc current, shield gas flow rate, and feed speed were used as a means to perform the required control. Zhang et al. utilized a spectrometer to analyze the plasma-developed laser lap welding. Simultaneously, a CCD camera was used to co-axially monitor the shape of the weld pool [24]. The intensity of characteristic peaks pertaining to constituent elements was observed to understand the welding practice. By integrating image processing and edge detection methods, they identified potential defects during the process. Huang et al. utilized an IR camera to acquire temperature information and implement interference analysis on their hybrid laser and TIG welding system. By doing so, they were successful in tracking the weld seam during the process [25]. Similarly, multiple systems were developed using pyrometers, CCD cameras, acoustic sensors, etc. to monitor the process [26–29]. The above-

mentioned control and monitoring setups were dedicated to observing a characteristic feature such as the size of melt or weld pool, temperature, the shape of weld plume, etc. Decision criteria based on this information were established through iterative experimentation. However, the analysis did not involve decoupling the monitored data to understand the solidification process. These setups were validated through qualification of final geometry. In this chapter though, a processing methodology for obtaining representative insight into the process of solidification is discussed. The influence and capacity of feedback systems are presented.

22.2 Influence of Feedback Systems

The authors theorize, controlling and maintaining a fixed melt pool size and ensuring consistent material deposition is central to setting up a robust deposition process. Incorporation of feedback mechanisms for compensating in-process inconsistencies and ensuring consistent layer thickness could be an approach for reliable fabrication. Pan and his colleagues at Missouri S&T developed a BPD system with two feedback systems aimed at managing the energy within the deposit and ensuring consistent material buildup [30].

Due to the localized heating in BPD, there are steep temperature gradients within and around the melt pool. Also, due to the large melting points of most metals, the melt pool is visibly hot and distinguishable. This enables the prospect of incorporating vision-based process monitoring systems based on cameras, both visible and infrared. Such cameras are excellent choices for gathering spatial and thermal information. However, these cameras yield large amounts of data. Feedback systems based on camera data are often slow due to the computational overhead originating from the need for massive data processing. For a fast and dynamic process such as BPD, incorporating camera-based feedback systems is still a challenging problem. However, vision-based sensing is still viable with analog high-speed sensors such as photodiodes and pyrometers. Due to the low-resolution aspect of these sensors, decoupling complex concurrent phenomenon can be challenging. There is a need for capturing such sensor data under various known conditions to establish process signatures.

In this section, the fabrication of AISI 316 stainless steel thin wall structures using BPD under the influence of two closed-loop feedback systems is discussed. Stainless powder in the particle size range of $-100/+325$ mesh was used to perform the deposition. A 1 kW fiber laser with a wavelength of 1064 nm was used in combination with a CNC worktable to perform these depositions. The CNC deposition system involved the use of stepper motors to achieve actuation. The process of micro-stepping was used to achieve smooth motion during deposition.

A FLIR A615 thermal infrared camera was used to monitor the deposition process. The A615 is a longwave infrared camera, which involves the use of micro-bolometers as the sensors. It is sensitive to infrared light in the spectrum range of 7.5–13.5 μm . The camera has a maximum resolution of 640×480 pixels. Under the calibrated

conditions, the camera is capable of sensing temperature up to a resolution of 50 mK. The lens used for the current set of analyses had a field of view of 25°. The camera was capable of reaching a frame rate of 200 frames per second.

Pan's feedback systems were used during these depositions. These systems can be classified as an energy management system and a height control system. These systems have been proven to successfully fabricate thin wall structures that met the design requirements [30]. Incorporation of these control systems facilitated near constant bead thickness, reliable material build, and good surface quality. Deposits fabricated with and without the influence of these control systems are shown in Fig. 22.2. The deposit fabricated under the influence of the control systems is visually better and has more consistent geometric features. The surface roughness was also better when fabricating under the influence of the control systems. The logic and setup of these control systems are discussed in sections below.

22.2.1 Energy Management System

From Planck's law, the spectral radiance of a black body at a constant temperature can be analytically modeled. The spectral radiance of a black body at a given wavelength can be calculated from Eq. (22.1).

$$B_{\lambda}(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \quad (22.1)$$



Fig. 22.2 Thin wall deposits fabricated without (left) and with (right) the control of feedback systems

Where, $B_{\lambda}(\lambda, T)$ is the spectral radiance observed at a wavelength λ , T is the absolute temperature of the blackbody, k_B is the Boltzmann constant, h is the Plank's constant, and c is the speed of light in the medium. From Wein's law, the peak of the spectral radiance curve occurs at wavelengths that is inversely proportional to the temperature of the black body. From these laws, it can be inferred that for a fixed shape and fixed thermal gradient the total spectral radiance in a given spectral range should be fixed. The energy management system monitors and attempts to control this radiance. This control is expected to indirectly correlate with the melt pool. In order to maintain the required radiance value, the system manipulates the input power during deposition. In case the measured radiance value is high, the control system drives the power down and vice versa. The logic flow of the energy management system is shown in Fig. 22.3.

While the radiance of the deposition site was monitored, the significance of the numerical value was not analyzed as a part of this study and setup. There is merit in understanding the contribution of each source, i.e., molten material and the just solidified material. However, this requires a more specialized investigation which goes beyond the scope of this work. For the current study, the optimized setting for

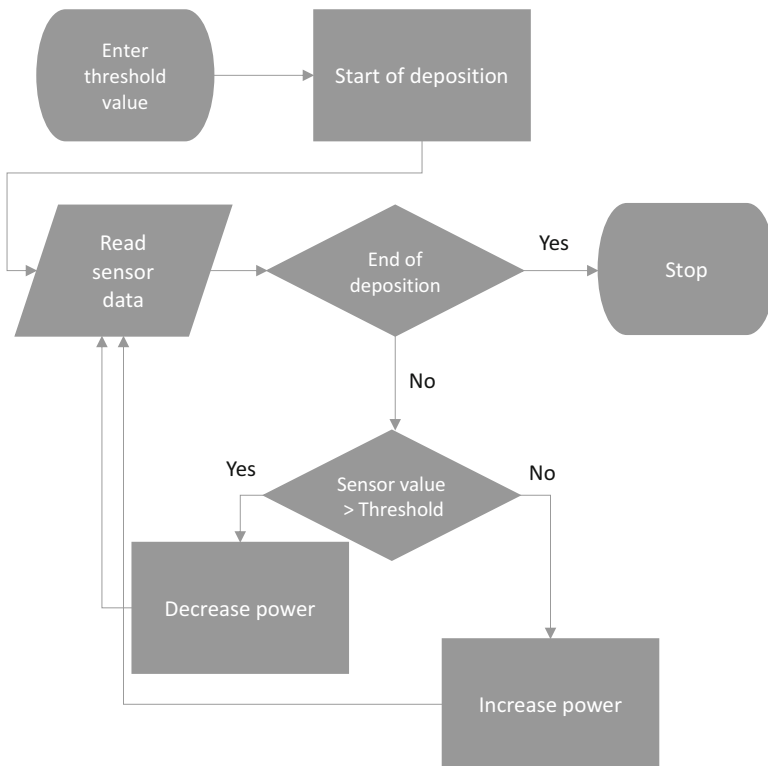


Fig. 22.3 Flowchart detailing the workflow of the energy management system

the energy management system was evaluated experimentally through trial and error. The optimal condition was chosen based on the size of the high temperature region during deposition. The outcome material quality was also inspected to ensure good bonding between successive layers and no porosity.

22.2.2 Height Control System

For the purposes of distance measurement, triangulation methods are often used in industrial environments using automation and vision-based sensing [31, 32]. The process typically involves lasers and light sensors to calculate the distance between two points. Extensions of this implementation have been used to even create 2D and 3D topographic profiles. There is a benefit to using such implementations in additive manufacturing. Monitoring build height enables reliable part fabrication and in-process correction [33, 34]. A similar implementation was used in the current setup as well. Instead of a laser-based setup, a temperature sensor based setup was used for triangulation.

The height control system is used to monitor and control the building height during BPD. The feedback system employs a non-contact temperature sensor to keep track of the top of the deposit. This sensor was a dual color pyrometer procured from WilliamsonIR. The camera was capable of measuring temperatures above 900 °C. Due to the dual color implementation, the pyrometer can bypass the issues of emissivity variance. The system uses a temperature threshold to assess the material buildup. The system employs a go/no go strategy to ensure the required build height during deposition. The control system manipulates the motion system to go forward when the intended buildup is attained and slows down or stops motion when the material buildup is below the requirement. These decisions are taken along the entire deposition tool path. A flowchart detailing the workflow of the height control system is shown in Fig. 22.4.

The energy management system and the height control system work in conjunction to facilitate reliable deposition during BPD. One of the biggest issues in BPD that leads to problems during deposition is capture efficiency. In order to maintain consistent layer thickness, the capture efficiency of the melt pool has to remain constant. In case the efficiency is more than nominal, the material buildup will be larger than the layer thickness. If the efficiency is lower than nominal, the buildup will be lower than the layer thickness. The pileup of these differences will lead to unsuccessful deposition. Under the right conditions, when the capture efficiency is close to nominal and the right power, feed rate, and scan speed are set, the deposition reaches a steady state and doesn't need any intervention.

Typically, the right deposition conditions are evaluated using a lengthy and expensive design of experiments study. However, in this case, the energy management system and the height control system can be used to tune the machine settings and obtain the right deposition parameters. While identifying a domain of viable parameters settings is made easy, picking the best setting by visually inspecting the

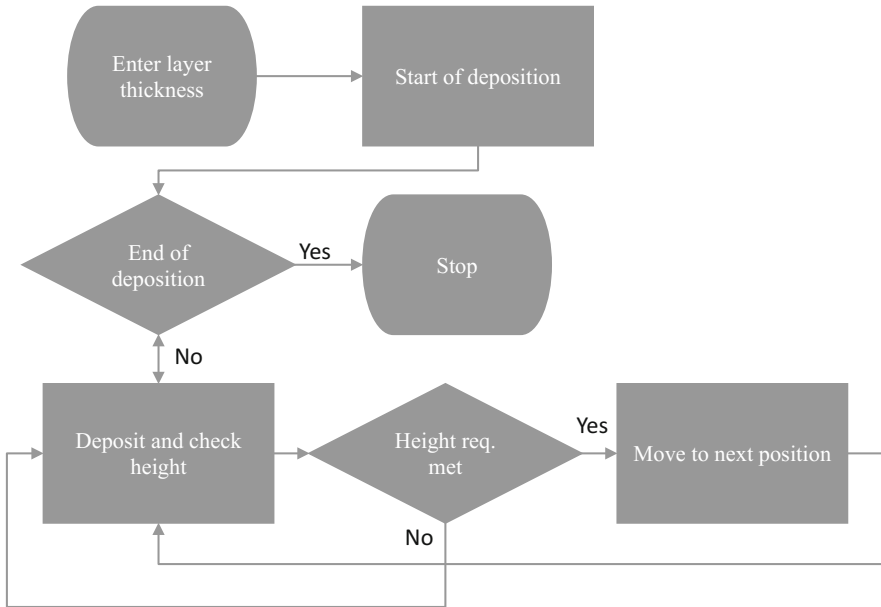


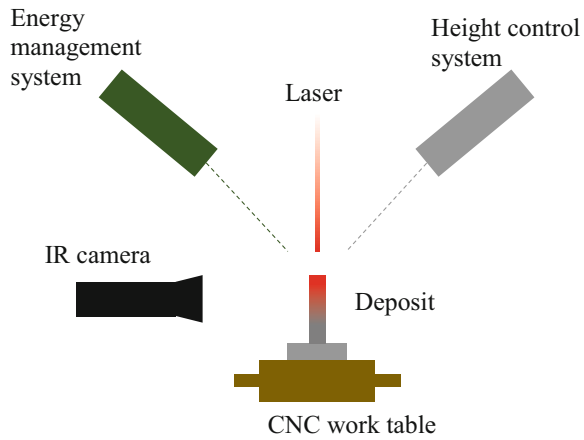
Fig. 22.4 Flowchart detailing the workflow of the height control system

deposit quality is not optimal. The complex interaction of the energy management system and height control system could lead to largely varying thermal conditions which could lead to inhomogeneity in material properties [10, 35, 36]. While many factors influence the material properties, thermal history can be considered as one of the most influential factors. Using thermal cameras to capture and analyze thermal history during deposition could serve as an evaluation tool to pick the appropriate process parameters.

A schematic drawing of the deposition setup is shown in Fig. 22.5. The IR camera was oriented normal to the deposit in order to capture the deposition process from a front view perspective. In this orientation, the camera is capable of recording a longitudinal view of the active region during deposition. The IR thermography is a surface measurement technique and does not provide insight into the variation of these measurements along the thickness of the deposit. The temperature gradients beneath the surface cannot be measured using this technique.

The variation in the total area of the high temperature region was used as a metric to assess the influence and difference with varying control system parameters. The high temperature region was defined as the region whose radiance is above a threshold value. The chosen threshold was set to the value corresponding to a temperature of 50° lower than the solidus temperature of 316 stainless steel. This was supposed to imply that the area on the deposit meeting the threshold criteria would constitute the material that had just solidified. However, that is not the case. The above equations and discussions pertaining to radiance are limited to that of a black body. In reality,

Fig. 22.5 Schematic side view diagram illustrating the setup of the feedback systems on the BPD system



most materials do not behave like a black body. Their radiance at a given temperature is a partial fraction of that of a black body at a similar temperature. These bodies are referred to as grey bodies. The radiative powder (W) from a grey body at a temperature (T) with a surrounding temperature (T_c) can be defined as Eq. (22.2). This equation, the Stefan–Boltzmann Law, is obtained by integrating the Plank’s law for all wavelengths. Emissivity (ϵ) is used to account for the lower radiation values of grey bodies. The value of the emissivity for grey bodies falls between 0 and 1. The “ σ ” is the Stefan–Boltzmann constant.

$$W = \epsilon \sigma A \left(T^4 - T_c^4 \right) \tag{22.2}$$

The infrared cameras measure the radiative power given off a grey or black body. Under the right setup conditions and setup calibration, these power values can be processed and interpreted as temperatures. Multiple sources contribute to the total radiative power value measured by the IR camera (W_{camera}). The breakdown of different sources contributing to the total radiative power measured by the IR camera is shown in Fig. 22.6. The body of interest, hot body, is shown in red. The temperature of the hot body is T_{HB} . The emissivity of the hot body is ϵ . The black body radiative power from the hot body is W_{HB} . The reflected temperature is T_{RT} and the radiative component is W_{RT} . The atmospheric temperature is T_{atm} and the radiative power component is W_{atm} . The transmissivity of the atmosphere is τ . The makeup of the total radiative power as measured by the IR camera is expressed as Eq. (22.3). The emissivity and transmissivity values are used to decouple the radiative power values measured by the camera. The decoupled power components are used to estimate the absolute temperature values.

$$W_{\text{camera}} = \epsilon \tau W_{\text{HB}} + (1 - \epsilon) \tau W_{\text{RT}} + (1 - \tau) W_{\text{atm}} \tag{22.3}$$

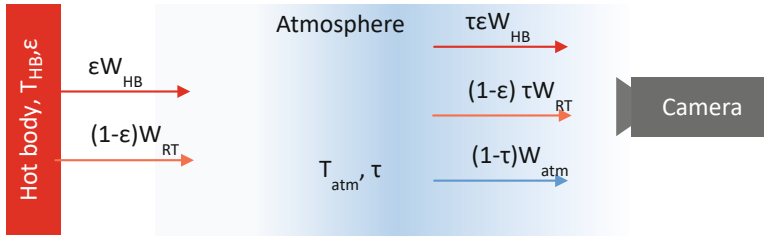


Fig. 22.6 Illustration demonstrating the breakdown of the total radiative powder measured by the IR camera

These power measurements are also influenced by the setup conditions and are sensitive to the range of temperature under study. The calibration of the camera and measurement setup is therefore essential for obtaining true temperature measurements. The complexity and sensitivity of the calibration and setup process make it challenging to obtain true temperature values. However, these challenges are minimized when used for purposes of differential thermography. The analysis discussed in this chapter involves the use of differential thermography and not absolute temperature measurement. A calibrated setup is doable in cases where the range of measured temperatures is small and the emissivity values are also known. However, the emissivity values for a given material are subject to change with temperature, wavelength, and phase change. Typically, the emissivity of metal drops when a phase change happens from solid to liquid state. This is attributed to the rise in reflectivity of the molten material. The same is true for the case of 316 stainless steel. In the case of BPD, the range of temperatures seen is very large and the process also involves phase change. Adding to this complexity, depending on the setup, the region of interest can be moving with the field of view. By the definition set for the high temperature region in the above paragraphs of discussion, this region could include portions or all of the melt pool. Further discussion on the constitution of the high temperature region will be continued in the next sections of the chapter.

22.2.3 Variation in the High Temperature Region

The deposition of 25 mm × 25 mm 316 stainless steel thin wall structures under the control of feedback systems for different powder feed settings was monitored using the IR camera. For a fixed threshold value on the energy management system and layer thickness, the deposition was performed at three different powder feed settings. A powder feed rate of 10, 30, and 50 gm/min was used for deposition. Argon gas was used to shield the deposition from oxidation.

A snapshot of the BPD process during the fabrication of a thin wall 316 stainless steel structure is shown in Fig. 22.7. This was generated using a false color rendering to realize the variation in temperature, where purple indicates the lower end of the

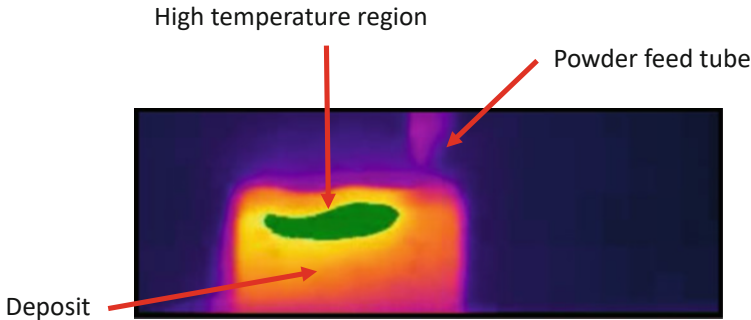


Fig. 22.7 Infrared thermograph of BPD of 316 stainless steel captured from a front view perspective. The green color represents the high temperature region

data and yellow indicates the higher end of the data. The deposit and powder feed tube can be seen within the picture. The thermal data was processed to identify the pixels that met the threshold criteria for the high temperature region. The pixels corresponding to the high temperature region are highlighted in green as shown in Fig. 22.7.

During deposition, the height control system checks to ensure sufficient material buildup. In case the required level was not reached, the control system slows down or stops the movement. By doing so, the material deposition is continued in the same location and when the required material buildup is attained the deposition system moves on to the next position along the tool path. In situations where the worktable is slowed down or stopped, continuous heating by the laser in the same location leads to heat retention. This can lead to slower cooling rates and in extension weaker material. These issues are supposed to be eliminated through the intervention of the energy management system. In cases of heat buildup, the energy management system lowers the power of the laser and attempts to mitigate heat retention. While this is the expected outcome from the control systems, the final outcome needs to be investigated. Typically destructive testing is employed to draw estimations of cooling rate and heat retention during deposition. However, with the IR camera incorporation, the size of a high temperature region can be monitored to more reliably visualize and optimize the deposition process. The evolution of the size of the high temperature region for each of the powder feed settings can be seen in Fig. 22.8.

From Fig. 22.8, there is a definite variation in the total area of the high temperature region with varying powder feed rate. At the lowest powder feed rate of 10 gm/min, the overall trend in the area of the high temperature region was found to be increasing. However, as the powder feed rate was increased, say for 30 gm/min, the overall value of the high temperature region was substantially reduced. However, steep rises and drops similar to those seen in the case of 10 gm/min still persisted. Upon further increasing powder feed rate to 50 gm/min, the overall area value stayed more consistent and the rises/drops in the values were also reduced. Among the considered

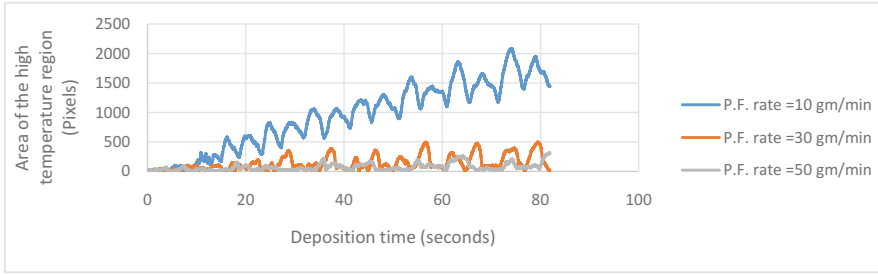


Fig. 22.8 Variation in the total area of the high temperature region during BPD with varying powder feed (P.F.) rate

powder feed rates, 50 gm/min yielded the most consistent values for the area of the high temperature region.

The overall rise in the area of the high temperature region for the 10 gm/min feed rate was suspected to be the intervention of the height control system. Due to the lower powder feed rate, meeting the requirements for material buildup would be slower in the case of 10 gm/min when compared to 30 and 50 gm/min feed rates. This implies that when the height control system slows down the deposition, there is a rise in heat retention. While the energy management system does lower the power, its intervention is insufficient in avoiding the overall increase in the area of the high temperature region. With the increase in powder feed rate to 30 gm/min, the intervention of the height control system seems to have been reduced and this is reflected in the total area data. However, the energy management system is still not entirely successful in eliminating issues of heat retention. This suggests at this powder feed value, the issue could be from cumulative error. The control system is suspected to be unable to sense minute differences in the material buildup. These differences can stack up over layers and warrant intervention from the height control system. In that instance, while there is a rise in the total area of the high temperature region, the intervention of the energy management system is successful in bringing down the overall value. The power feed rate of 50 gm/min appears to be near optimal. The control systems managed to keep the area of the high temperature region near consistent. Therefore, the optimal powder feed setting among the three settings was concluded to be 50 gm/min.

While deposits made at all the powder feed settings met the required geometric criteria, without the IR camera monitoring, the influence of the control systems would have been challenging to analyze. The analysis of the area of the high temperature region led to valuable insights into the process. However, further dissection of the thermal data is possible. The variation of the emissivity values caused by phase change can help break down the thermal data into the just solidified region and melt pool region.

22.3 Melt Pool Identification

The emissivity of the solid 316 stainless is higher than the emissivity of the liquid phase [23, 37]. Also, the emissivity of 316 stainless with an oxidized surface is higher than that of one with a shiny metallic finish. Therefore, from Eq. (22.3), at solidus temperature, the radiation from the melt pool in BPD will be lower than the radiation from the solid metal. If the pixels corresponding to solid and liquid weren't identified and the appropriate emissivity values were not assigned, the temperature measurements from the IR camera can be misleading. For example, if the emissivity for the entire field of view in Fig. 22.7 was set to 0.95, the pixels corresponding to the melt pool would appear colder than the material at solidus temperature. While this, in reality, is not true, this phenomenon can be leveraged to identify the pixels constituting the melt pool.

In an image, a sharp difference in contrast stemming from differences in details is referred to as an edge. By employing edge detection techniques, the pixels in the IR image corresponding to the melt pool could be identified. Since 316 stainless steel is not a eutectic composition, it has a freezing range. Over this range of temperatures, the material gradually transitions from solid to liquid and it is referred to as mushy zone. In BPD, a vague boundary can be seen between the completely liquid phase and the completely solid phase. However, due to the low magnification and resolution of the IR camera (640×120) in the current setup, resolving this fuzzy boundary is not expected to be feasible. Therefore, a step-like transition from solid to liquid and vice versa is expected. Due to this lack of distinction between melt pool and mushy zone, from here on the melt pool and the mushy zone are collectively referred to as the melt pool.

$$e_{\text{Just solidified region}} > e_{\text{Melt pool+mushy zone}} > e_{\text{Ambient region}} \quad (22.4)$$

$$W_{\text{Just solidified region}} > W_{\text{Melt pool+mushy zone}} > W_{\text{Ambient region}} \quad (22.5)$$

Figure 22.9 shows a schematic representation of the boundaries formed during deposition and the comparison of the emissivity values. Boundaries in Fig. 22.9 indicate the edges that are possible due to the emissivity differences. Due to the variation in emissivity values, the effective radiative power values and thereby the calculated temperature values also vary and follow the same trend. The effective radiative power is the power value as measured by the camera. The descending order of emissivity and radiative power values are shown in Eqs. (22.4) and (22.5). Through edge detection, these characteristic regions such as the Just Solidified Region (JSR) and the melt pool can be identified. The presence of these differences can be realized by performing a discrete thermal gradient analysis across the horizontal and vertical directions of the thermograph. The peaks observed in these gradients were expected to bear correlation with the transitions in the material's emissivity

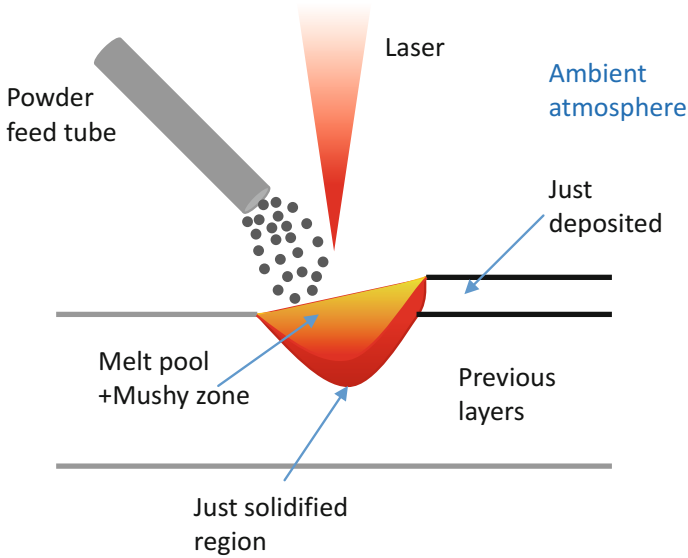


Fig. 22.9 An illustration of boundaries between different regions of interest and corresponding differences in emissivity

values. The evaluated vertical and horizontal gradients of temperatures (Fig. 22.7) during deposition are shown in Figs. 22.10 and 22.11. The equations used to calculate the discrete gradient are Eqs. (22.6) and (22.7). $x_{i,j}$ represents the pixel value in the (i, j) positions of an image.

$$\text{Discrete gradient in } y = x_{i,j} - x_{i,j-1} \quad (22.6)$$

$$\text{Discrete gradient in } x = x_{i,j} - x_{i-1,j} \quad (22.7)$$

Figure 22.10 indicates peaks in the discrete vertical gradient of thermal data from Fig. 22.7; these peaks are expected to represent the vertical transition from the ambient region to deposit. Peaks around the melt pool can also be noticed. Figure 22.11 depicts the discrete horizontal gradient of the thermal data from Fig. 22.7. The peaks in this plot are expected to show the horizontal transitions from the ambient region to deposit and vice versa. The transitions around the melt pool can also be seen in this gradient plot. Laplace edge detection technique was used to isolate the location of these transitions in the deposit. Standard functions from python libraries were implemented to identify these edges. The foreseen edges were captured after a series of smoothing, gradient and edge detection operations. The detailed implementation of processing is presented below.

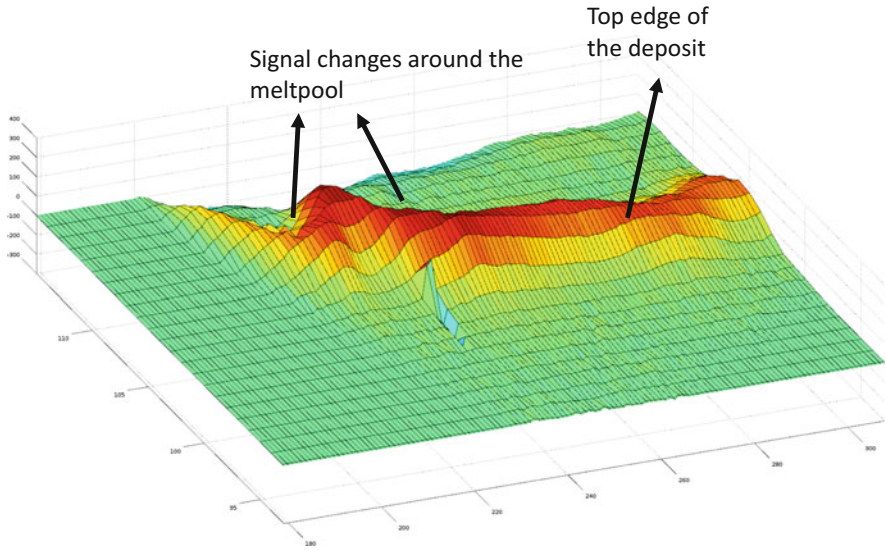


Fig. 22.10 Peaks occurring around the melt pool and the top edge of the deposit in the surface plot of the discrete gradient of the thermal data along the vertical direction

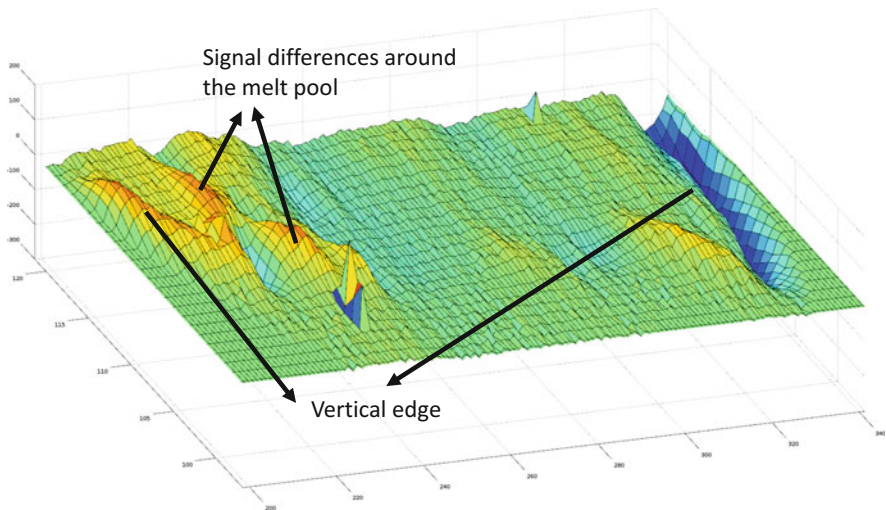


Fig. 22.11 Peaks occurring around the melt pool and the top edge of the deposit in the surface plot of the discrete gradient of the thermal data along the horizontal direction

1. Moving median

Figure 22.12 shows a false color rendering of the output generated after applying a moving median filter. The iron color palette was chosen for rendering, where black is the lowest value and white is the highest value. The moving median was performed by picking the median values among every five consecutive frames. The moving median operation was expected to remove powder particles, oxidation flashes, and Johnson noise. Equation (22.8) details the moving median operation. The $x_{i,j,t}$ is the pixel value at the (i, j) position in an image acquired at time t . $mx_{i,j,t}$ is the median value gathered across pixels in the same position across five consecutive time steps. This median data was used for further processing.

$$mx_{i,j,t} = \text{Median}(x_{i,j,t-5}, x_{i,j,t}) \quad (22.8)$$

2. Gaussian blur and Laplacian transform

A function to perform the Gaussian blur and the Laplace transform (see Eq. (22.9)) was implemented on data outputted in the previous step. The Gaussian blur minimizes spatial noise across the pixels of the image. This operation is necessary since the second derivative operations are very sensitive to noise. Typically these operations are implemented through convolution operations with pre-calculated kernels. The output image is shown in Fig. 22.13. This data from here on is referred to as LoG (Laplacian of Gaussian).

A more localized search was then performed by limiting the search domain to the region inside the deposit boundary. A search for edges with a lower threshold yielded the melt pool boundary solidified region. The processed image with melt pool (red), just solidified region (yellow), and the deposit boundary (blue) shown in Fig. 22.15.

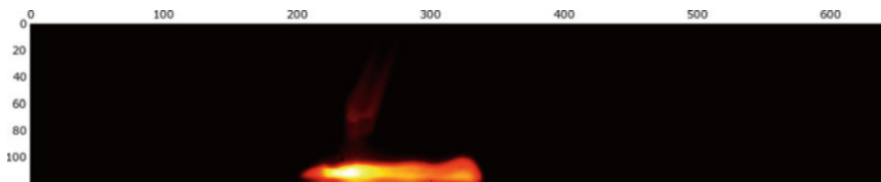


Fig. 22.12 False color image post moving median operation

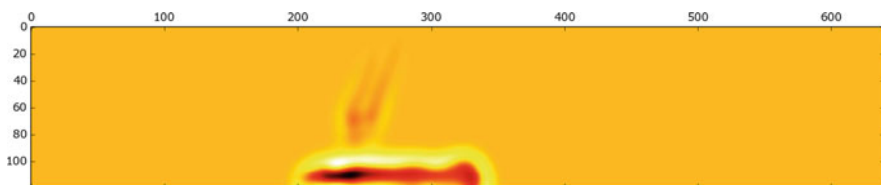


Fig. 22.13 Spatial variation in double discrete deviate of the thermal data from Fig. 22.12

$$L(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (22.9)$$

3. Edge detection

The LoG data was analyzed for identifying edges, and an appropriate threshold value was set to eliminate noise and capture only the most significant changes. The presence of an edge is realized as a zero crossing in the LoG data. A change in sign across consecutive pixels was used to identify these zero crossings. The edges that were identified are shown as a binary image (see Fig. 22.14). The site of an edge was set as 1 and the rest were set to zero. The identified boundaries were found to be those of the deposit and the powder feed tube. While the deposit boundaries were easily identified, the melt pool boundaries were not captured. The transitions corresponding to the melt pool boundary were not substantial enough to meet the set threshold. Thereby the melt pool boundary was not captured at this stage.

Implementing the above steps of image processing led to the generation of data which are visualized in Fig. 22.16 for better understanding. Figure 22.16 shows the first layer of the deposition before the steady state was obtained. As discussed in the previous section, the variation in the just solidified region is substantial before steady state was achieved. This expected due to the intervention of the control systems. Figure 22.17 show snapshots from during deposition after steady state was achieved. The size of the melt pool and the just solidified region remained almost constant post steady state.

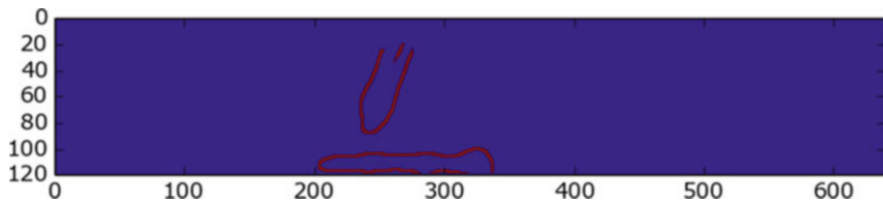


Fig. 22.14 The edges of the deposit and powder feed tube (red)

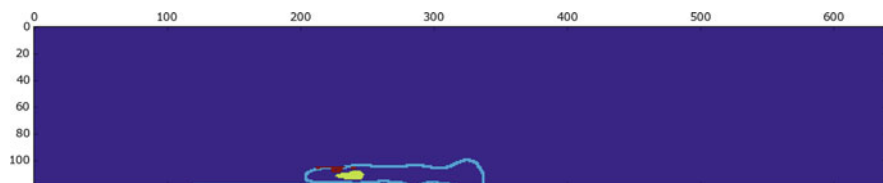


Fig. 22.15 The melt pool (red) and just solidified region (yellow) boundary of the deposit (sky blue)

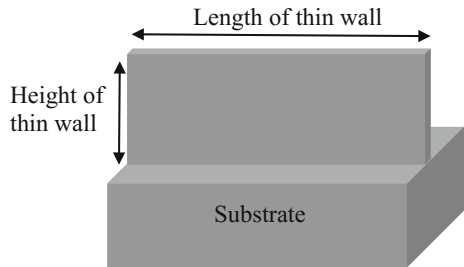


Fig. 22.16 The first layer of deposition, melt pool (white), just solidified region (yellow), and deposit boundary (red), left to right progression in the deposition



Fig. 22.17 Melt pool (white), just solidified region (yellow), and deposit boundary (red) after a steady state was achieved by the control system, left to right progression of deposition

Fig. 22.18 Illustration of thin wall substrates used for line scan depositions



22.3.1 Sensitivity and Repeatability

In order to assess the sensitivity of this technique, depositions on substrates machined to the shape of thin wall structures were analyzed. These substrates were machined to have different lengths and heights of the thin wall structure. The shape of these thin wall substrates is defined in Fig. 22.18. These experiments involved four layers of an end to end deposition on each of the substrates. These depositions were monitored using the IR camera to capture the thermal history. To simplify the complexity of the process, these depositions were carried out without the control of the feedback systems.

The analysis of IR data from these deposits indicated a definite impact of substrate length and thin wall height. This influence was theorized to be a consequence of different degrees of heat retention. The difference in heat retention was attributed to the difference in the size of these thin walls and the resulting differences in conduction losses. Larger sizes of high temperature regions were seen on substrates with a smaller length. Due to the small track length, heat retention was observed to be prominent. On the other hand, the deposits with low thin wall heights showed very small sizes of melt pool and just solidified region. The close proximity of the melt pool to the substrate due to the small height of the thin wall leads to low levels of heat retention and in extension small sizes of melt pool and just solidified regions. The analysis of thermal data from deposits on substrates with taller thin walls was also in agreement with this conclusion. Due to the large thin wall heights and distance from

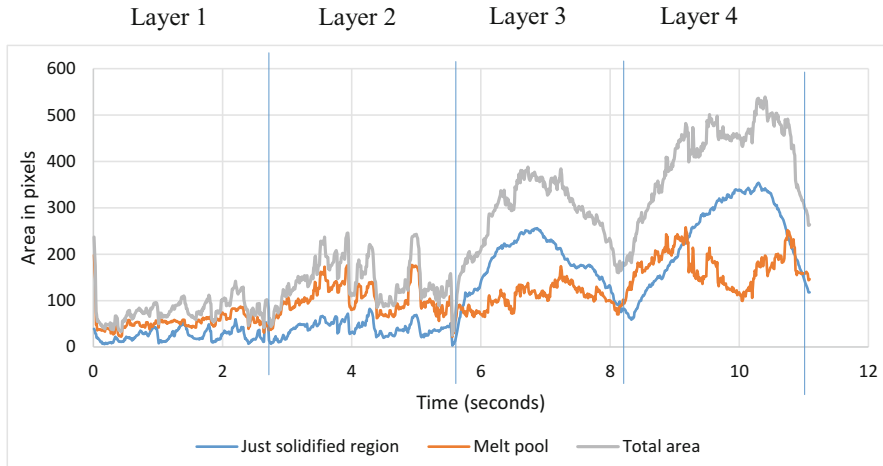


Fig. 22.19 Variation in areas of just solidified regions and melt pool on a thin wall substrate

the substrate, the heat retention was observed to be significant. Consequently, the total area of the melt pool and just solidified region were observed to be substantially higher. These subtle differences in sizes of melt pool were registered and identified using this monitoring methodology. Analysis of replications of these depositions yielded similar conclusions reliably [38].

The rise and drops in the total area of the high temperature region seen in Fig. 22.8 were attributed to the rise in heat retention. These fluctuations are suspected to be the consequences of thin wall geometry. At the edges of the thin wall, the total area of the melt pool was observed to increase. This phenomenon was observed to be more prominent with increasing heat retention. At the edges, due to the localized heating of BPD, an increase in the size of the melt pool resulted in a corresponding decrease in the size of the just solidified region. However, the overall high temperature region was seen to increase. Previously, this observation was not apparent from only monitoring the high temperature region from the temperature data. Identification of melt pool data is necessary to understand the complete picture during BPD. The variation in the area of the regions of interest is shown in Fig. 22.19.

22.4 Conclusions

Acquisition of thermal history through an IR camera was identified to be a viable method for monitoring blown powder deposition process. A methodology was laid out for qualitative analysis of the thermographic data acquired from the IR camera. Monitoring the total area of the material meeting a temperature based criterion was found to be instrumental in assessing the influence of closed loop feedback systems.

An image processing methodology based on edge detection was developed to analyze the IR data. The difference in emissivity values of the solid and liquid phases of metal resulted in sharp changes in the IR data. Edge detection techniques were used to successfully identify these phase transitions. Identification of these transitions led to successful identification of the melt pool.

Acknowledgments The financial support from National Science Foundation Grant # CMMI-1625736 and the Intelligent Systems Center (ISC) at Missouri S&T is appreciated.

References

1. Murr, L. E., Gaytan, S. M., Medina, F., et al. (2010). Next-generation biomedical implants using additive manufacturing of complex, cellular and functional mesh arrays. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 368, 1999–2032. <https://doi.org/10.1098/rsta.2010.0010>.
2. Vishnu Prashant Reddy, K., Meera Mirzana, I., & Koti Reddy, A. (2018). Application of additive manufacturing technology to an aerospace component for better trade-off's. *Materials Today: Proceedings*, 5, 3895–3902. <https://doi.org/10.1016/J.MATPR.2017.11.644>.
3. Busachi, A., Erkoyuncu, J., Colegrove, P., et al. (2017). A review of additive manufacturing technology and cost estimation techniques for the defence sector. *CIRP Journal of Manufacturing Science and Technology*, 19, 117–128. <https://doi.org/10.1016/J.CIRPJ.2017.07.001>.
4. Kumar Dama, K., Kumar Malyala, S., Suresh Babu, V., et al. (2017). Development of automotive FlexBody chassis structure in conceptual design phase using additive manufacturing. *Materials Today: Proceedings*, 4, 9919–9923. <https://doi.org/10.1016/J.MATPR.2017.06.294>.
5. Delgado Camacho, D., Clayton, P., O'Brien, W. J., et al. (2018). Applications of additive manufacturing in the construction industry—A forward-looking review. *Automation in Construction*, 89, 110–119. <https://doi.org/10.1016/J.AUTCON.2017.12.031>.
6. Oter, Z. C., Coskun, M., Akca, Y., et al. (2019). Benefits of laser beam based additive manufacturing in die production. *Optik (Stuttgart)*, 176, 175–184. <https://doi.org/10.1016/J.IJLEO.2018.09.079>.
7. Ford, S., & Despeisse, M. (2016). Additive manufacturing and sustainability: An exploratory study of the advantages and challenges. *Journal of Cleaner Production*, 137, 1573–1587. <https://doi.org/10.1016/J.JCLEPRO.2016.04.150>.
8. (10AD) Standard Terminology for Additive Manufacturing Technologies BT—Standard Terminology for Additive Manufacturing Technologies.
9. Lewis, G. K., & Schlienger, E. (2000). Practical considerations and capabilities for laser assisted direct metal deposition. *Materials and Design*, 21, 417–423. [https://doi.org/10.1016/S0261-3069\(99\)00078-3](https://doi.org/10.1016/S0261-3069(99)00078-3).
10. Shamsaei, N., Yadollahi, A., Bian, L., & Thompson, S. M. (2015). An overview of direct laser deposition for additive manufacturing; Part II: Mechanical behavior, process parameter optimization and control. *Additive Manufacturing*, 8, 12–35. <https://doi.org/10.1016/j.addma.2015.07.002>.
11. Dinda, G. P., Dasgupta, A. K., & Mazumder, J. (2009). Laser aided direct metal deposition of Inconel 625 superalloy: Microstructural evolution and thermal stability. *Materials Science and Engineering A*, 509, 98–104. <https://doi.org/10.1016/J.MSEA.2009.01.009>.
12. Mazumder, J., Dutta, D., Kikuchi, N., & Ghosh, A. (2000). Closed loop direct metal deposition: Art to part. *Optics and Lasers in Engineering*, 34, 397–414. [https://doi.org/10.1016/S0143-8166\(00\)00072-5](https://doi.org/10.1016/S0143-8166(00)00072-5).

13. Peyre, P., Aubry, P., Fabbro, R., et al. (2008). Analytical and numerical modelling of the direct metal deposition laser process. *Journal of Physics D: Applied Physics*, *41*, 025403. <https://doi.org/10.1088/0022-3727/41/2/025403>.
14. Javaid, M., & Haleem, A. (2017). Additive manufacturing applications in medical cases: A literature based review. *Alexandria Journal of Medicine*, *54*(4), 411–422. <https://doi.org/10.1016/J.AJME.2017.09.003>.
15. Wang, Y., Wei, Q., Pan, F., et al. (2014). Molecular dynamics simulations for the examination of mechanical properties of hydroxyapatite/poly α -n-butyl cyanoacrylate under additive manufacturing. *Bio-medical Materials and Engineering*, *24*, 825–833. <https://doi.org/10.3233/BME-130874>.
16. Sahoo, S., & Chou, K. (2016). Phase-field simulation of microstructure evolution of Ti–6Al–4V in electron beam additive manufacturing process. *Additive Manufacturing*, *9*, 14–24. <https://doi.org/10.1016/J.ADDMA.2015.12.005>.
17. Amine, T., Newkirk, J. W., & Liou, F. (2014). An investigation of the effect of direct metal deposition parameters on the characteristics of the deposited layers. *Case Studies in Thermal Engineering*, *3*, 21–34. <https://doi.org/10.1016/J.CSITE.2014.02.002>.
18. Qi, H., Mazumder, J., & Ki, H. (2006). Numerical simulation of heat transfer and fluid flow in coaxial laser cladding process for direct metal deposition. *Journal of Applied Physics*, *100*, 024903. <https://doi.org/10.1063/1.2209807>.
19. Zheng, B., Zhou, Y., Smugeresky, J. E., et al. (2008). Thermal behavior and microstructure evolution during laser deposition with laser-engineered net shaping: Part II. Experimental investigation and discussion. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, *39*, 2237–2245. <https://doi.org/10.1007/s11661-008-9566-6>.
20. Zheng, B., Zhou, Y., Smugeresky, J. E., et al. (2008). Thermal behavior and microstructural evolution during laser deposition with laser-engineered net shaping: Part I. Numerical calculations. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, *39*, 2228–2236. <https://doi.org/10.1007/s11661-008-9557-7>.
21. Mazumder, J., Choi, J., Nagarathnam, K., et al. (1997). The direct metal deposition of H13 tool steel for 3-D components. *JOM*, *49*, 55–60. <https://doi.org/10.1007/BF02914687>.
22. Song, L., Bagavath-Singh, V., Dutta, B., & Mazumder, J. (2012). Control of melt pool temperature and deposition height during direct metal deposition process. *International Journal of Advanced Manufacturing Technology*, *58*, 247–256. <https://doi.org/10.1007/s00170-011-3395-2>.
23. Kovacevic, R., & Zhang, Y. M. (1997). Real-time image processing for monitoring of free weld pool surface. *Journal of Manufacturing Science and Engineering*, *119*, 161. <https://doi.org/10.1115/1.2831091>.
24. Zhang, Y., Zhang, C., Tan, L., & Li, S. (2013). Coaxial monitoring of the fibre laser lap welding of Zn-coated steel sheets using an auxiliary illuminant. *Optics and Laser Technology*, *50*, 167–175. <https://doi.org/10.1016/j.optlastec.2013.03.001>.
25. Huang, R.-S., Liu, L.-M., & Song, G. (2007). Infrared temperature measurement and interference analysis of magnesium alloys in hybrid laser-TIG welding process. *Materials Science and Engineering A*, *447*, 239–243. <https://doi.org/10.1016/J.MSEA.2006.10.069>.
26. Li, L. (2002). A comparative study of ultrasound emission characteristics in laser processing. *Applied Surface Science*, *186*, 604–610.
27. Gao, J., Qin, G., Yang, J., et al. (2011). Image processing of weld pool and keyhole in Nd:YAG laser welding of stainless steel based on visual sensing. *Transactions of the Nonferrous Metals Society of China*, *21*, 423–428. [https://doi.org/10.1016/S1003-6326\(11\)60731-0](https://doi.org/10.1016/S1003-6326(11)60731-0).
28. Saeed, G., & Zhang, Y. M. (2007). Weld pool surface depth measurement using a calibrated camera and structured light. *Measurement Science and Technology*, *18*, 2570–2578. <https://doi.org/10.1088/0957-0233/18/8/033>.
29. Luo, M., & Shin, Y. C. (2015). Vision-based weld pool boundary extraction and width measurement during keyhole fiber laser welding. *Optics and Lasers in Engineering*, *64*, 59–70. <https://doi.org/10.1016/J.OPTLASENG.2014.07.004>.

30. Pan, Y. (2013). *Part height control of laser metal additive manufacturing process*. Missouri University of Science and Technology.
31. Garcia-Cruz, X. M., Sergiyenko, O. Y., Tyrsa, V., et al. (2014). Optimization of 3D laser scanning speed by use of combined variable step. *Optics and Lasers in Engineering*, *54*, 141–151. <https://doi.org/10.1016/J.OPTLASENG.2013.08.011>.
32. Lindner, L., Sergiyenko, O., Rodríguez-Quíñonez, J. C., et al. (2016). Mobile robot vision system using continuous laser scanning for industrial application. *Industrial Robot: An International Journal*, *43*, 360–369. <https://doi.org/10.1108/IR-01-2016-0048>.
33. Donadello, S., Motta, M., Demir, A. G., & Previtali, B. (2018). Coaxial laser triangulation for height monitoring in laser metal deposition. *Procedia CIRP*, *74*, 144–148. <https://doi.org/10.1016/J.PROCIR.2018.08.066>.
34. Donadello, S., Motta, M., Demir, A. G., & Previtali, B. (2019). Monitoring of laser metal deposition height by means of coaxial laser triangulation. *Optics and Lasers in Engineering*, *112*, 136–144. <https://doi.org/10.1016/J.OPTLASENG.2018.09.012>.
35. Lewandowski, J. J., & Seifi, M. (2016). Metal additive manufacturing: A review of mechanical properties. *Annual Review of Materials Research*, *46*, 151–186. <https://doi.org/10.1146/annurev-matsci-070115-032024>.
36. Carroll, B. E., Palmer, T. A., & Beese, A. M. (2015). Anisotropic tensile behavior of Ti–6Al–4V components fabricated with directed energy deposition additive manufacturing. *Acta Materialia*, *87*, 309–320. <https://doi.org/10.1016/J.ACTAMAT.2014.12.054>.
37. Roger, C. R., Yen, S. H., & Ramanathan, K. G. (1979). Temperature variation of total hemispherical emissivity of stainless steel AISI 304. *Journal of the Optical Society of America*, *69*, 1384. <https://doi.org/10.1364/JOSA.69.001384>.
38. Karnati, S. (2015). *Thermographic investigation of laser metal deposition*. Missouri University of Science and Technology.

Chapter 23

Image Processing Filters for Machine Fault Detection and Isolation



Ranjan Ganguli

Acronyms

ACO	Ant colony optimization
ACS	Ant colony system
ECS	Environment cooling system
EGT	Exhaust gas temperature
FDI	Fault detection and isolation
FIR	Finite impulse response
HPC	High pressure compressor
HPT	High pressure turbine
IIR	Infinite impulse response
LPC	Low pressure compressor
LPT	Low pressure turbine
MAE	Mean absolute error
N1	High rotor speed
N2	Low rotor speed
RM	Recursive median
SM	Simple median
TCC	Turbine cooling casing
WF	Fuel flow
WRM	Weighted recursive median

R. Ganguli (✉)

Department of Aerospace Engineering, Indian Institute of Science (IISc), Bangalore, Karnataka, India

e-mail: ganguli@iisc.ac.in

23.1 Introduction

Fault detection and isolation (FDI) typically uses measured data and machine vision in conjunction with identification, optimization, or soft computing algorithms [1–3]. There are two broad types of faults: single faults and gradual faults. Single faults are typically preceded by a sharp change in the signal. Gradual faults lead to a slow change in the signal which can be approximated with a linear variation. These signals can be viewed as images acquired by machine vision. For example, stereo vision systems [4] and noise removal systems [5] play an important role in structural and condition monitoring of systems. Mohan and Poobal [6] provide a recent review of crack detection in image processing. Such approaches typically use images along with optimization methods and pattern recognition tools to isolate the system fault.

The gas turbine is a machine which is widely used for power generation. It is used for aircraft engines, and therefore automated processes which detect and isolate faults in this machine are of great interest. Algorithms developed for FDI of such machines based on image processing filters are discussed in this chapter. These algorithms have applicability to all machines where sensor data is available in the form of 1D images.

Typical signals used for gas turbine machine diagnostics are exhaust gas temperature (EGT), fuel flow (WF), high rotor speed (N1), and low rotor speed (N2). These four basic sensors are present in almost all jet engines. The signals considered for gas turbine machine diagnostics are called “measurement deltas” which are deviations between sensor measurements of a “damaged” engine compared to a “good” engine. For an ideal undamaged engine, the measurement deltas are zero. The measurement deltas obtained from operational machines are typically non-zero and are also contaminated with noise. Fault detection and isolation (FDI) algorithms are used to detect and isolate the machine fault. Here “detection” is the process of identifying if a fault is present or not. Errors in detection can lead to false alarms. Also, “isolation” is the process of identifying the type of fault. Typically, fingerprint charts are used for fault isolation and these relate the measurement deltas produced to a given change in the machine state. For example, Table 23.1 presents the fingerprint chart for a 2% deterioration in the efficiency of the engine modules [7].

Fingerprint charts represent a linearized model evaluated at a selected engine operating point. Such tables are obtained from thermodynamics and are available with aircraft engine manufacturers.

Table 23.1 Fingerprints for selected gas turbine machine faults for $\eta = -2\%$

Module faults/measurement deltas	$\Delta\text{EGT (C)}$	$\Delta\text{WF}\%$	$\Delta\text{N2}\%$	$\Delta\text{N1}\%$
High pressure compressor (HPC)	13.60	1.60	-0.11	0.10
High pressure turbine (HPT)	21.77	2.58	-1.13	0.15
Low pressure compressor (LPC)	9.09	1.32	0.57	0.28
Low pressure turbine (LPT)	2.38	-1.92	1.27	-1.96
Fan	-7.72	-1.40	-0.59	1.35

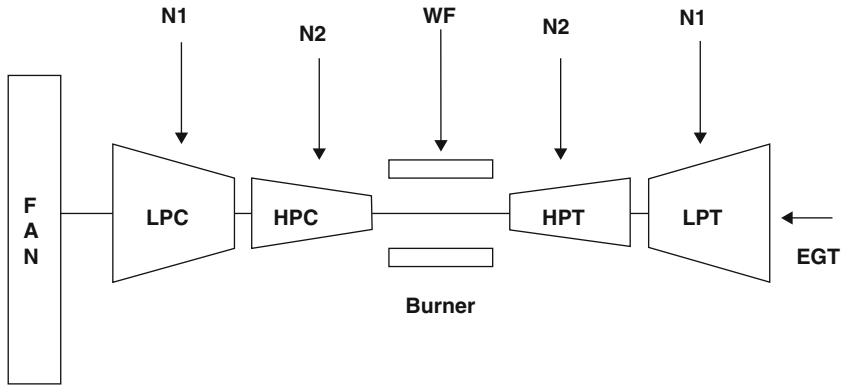


Fig. 23.1 Schematic representation of jet engine and its four key measurements

Figure 23.1 shows the engine modules and basic measurements for a typical turbofan engine. At the most basic level, fault isolation would indicate if the fault is present in the fan, high pressure compressor, low pressure compressor, high pressure turbine, or low pressure turbine module. These are coupled faults within the major modules of the engine. Other system faults such as handling and ECS (environment control system) bleed leaks and failures, variable stator vane malfunctions, TCC (turbine case cooling) malfunctions as well as certain instrumentation faults can also be considered as single faults [2]. We will only consider module faults to create the ideal signal for damaged engine in this study. Such faults can emanate from different physical processes but the signature is shown through the sensor measurement deltas. Once the fault is isolated to the module level, the maintenance engineer can then focus only on these modules for repair work.

The accuracy of FDI algorithms improves if noise is removed from the gas path measurement signals while preserving features indicating a single fault such as sharp trend shifts [8, 9]. Typical linear filters such as the moving average filter and exponential average filter can perform as good smoothers for gas turbine signals. The moving average filter is a simple FIR (Finite Impulse Response) filter with equal weights and the exponential average is an IIR (Infinite Impulse Response) filter. While linear filters can remove noise, they smooth out the sharp trend shifts which can indicate a single fault event. Thus, machine signals are more akin to images than typical 1D signals as there is a need for edge preservation and smoothing filters.

Therefore, nonlinear filters originating from image processing area such as the median filter were proposed for noise removal from gas turbine signals [8]. Other computational architectures for noise removal from jet engine signals include the auto-associative neural network [10], radial basis neural networks [9], myriad filter [11], and recursive median (RM) filter [12]. The RM filter is an efficient alternative to the median filter and converges rapidly to the root signal when compared to the simple median (SM) filter. The SM filter can take many passes before converging to the root signal. However, the RM filter can lead to a phenomenon called “streaking,”

which involves creation of artificial step-like artifacts in the signal. This problem can be removed by introducing weights, resulting in the weighted recursive median (WRM) filter.

The WRM filters have integer weights and the optimal calculation of these weights for a given application is an important problem in filter design. The design space of the weights of WRM filters is multimodal (shows the presence of several local minima) and an exhaustive search of the design space can be used to find the weights [7]. However, this exhaustive search approach is very computationally intensive and there is a need for more efficient algorithms for solving this filter weight optimization problem. In this chapter, ACO is used to design a WRM filter for use as a data smoothing preprocessor in gas turbine diagnostics. A schematic of this procedure is shown in Fig. 23.2. We focus on the “noise reduction” aspect of gas turbine diagnostics shown in Fig. 23.2.

Few researchers have addressed the problem of optimization of median filter weights. Algorithms for calculating the integer weights of weighted median filters were proposed [13]. Both recursive and non-recursive filters were considered but the study focused on center weights. A numerical approach for the optimization of recursive median filters was presented in Arce and Paredes [14]. Uday and Ganguli [7] searched over the low integer space (1, 2, and 3) to find the optimal weights. They found that higher integer weights led to duplication in the filter and the low integer space was sufficient for the given problem.

Filter design spaces are often multimodal, implying that there can be more than one minimum point. Therefore, use of gradient-based numerical optimization can lead to a local minimum point. To address this issue, the application of global optimization methods in filter design has grown substantially. Particle swarm optimization was applied to solve the parameter estimation problem of nonlinear dynamic rational filters [15]. Genetic algorithms were used for optimizing stack filters using a root mean square error norm [16]. Ant colony optimization was used for the design of IIR filters [17]. Since the error surface of IIR filters is typically multimodal, global optimization methods such as ACO are amenable for their design. ACO is a relatively new approach for solving combinatorial optimization problems. The main characteristics of ACO are positive feedback, distributed computation, and

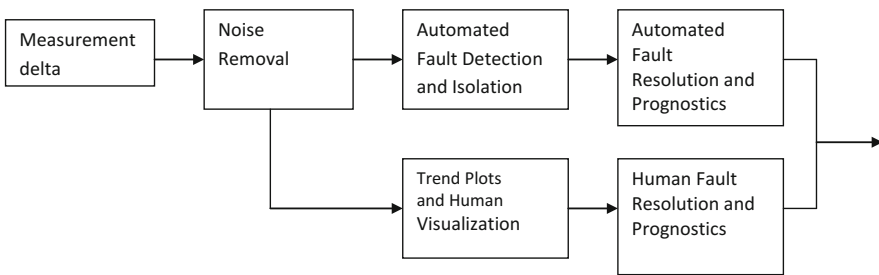


Fig. 23.2 Schematic representation of gas turbine diagnostics system

the use of a constructive greedy heuristic [18]. Note that a heuristic method is an approach to solving problems that employs a practical sequence of steps which is not guaranteed to be optimal, but is sufficient for obtaining useful results for applications. Heuristics can be called rules of thumb or educated guesses. Since ACO is a heuristic method, it gives satisfactory solutions for many difficult optimization problems, but these solutions may not prove to be optimal in the mathematical sense. Furthermore, the convergence of heuristic methods cannot be guaranteed.

In this chapter, we address the problem of finding the integer weights of WRM filters using ACO to design a useful image processing filter. The algorithm is demonstrated for signals simulating jet engine single (abrupt) and gradual faults. Some image processing filters are discussed next. These filters are widely used in 2D image processing and can be easily used for 1D image processing. This chapter is based on an earlier work by Raikar and Ganguli [19].

23.2 Image Processing Median Filters

The mean and the median represent simple measures of central tendency. The mean of a set of number is an FIR filter with uniform weights. The median filters are based on the median operation, typically performed over an odd sample of numbers. The SM (simple median) filter with length of window of $N = 2n + 1$ can be represented as [20]:

$$y_k = \text{median} (x_{k-n}, x_{k-n+1}, \dots, x_k, \dots, x_{k+n-1}, x_{k+n}) \quad (23.1)$$

Here x_k and y_k are the k th sample of the input and output sequences, respectively, and n represents integers ensuring that the window length N is odd for easy calculation of the median. The calculation of the median needs that these numbers be sorted from lowest to highest and then the central number be selected as the median output. The SM filter needs a large number of iterations to converge to a desired output. A 5-point SM filter can be written as $y_k = \text{median}(x_{k-2}, x_{k-1}, x_k, x_{k+1}, x_{k+2})$ since $N = 5 \Rightarrow n = 2$. The 5-point SM filter has a window length of 5 and a two-point time lag as it needs measurements at the time points $k + 1$ and $k + 2$ to predict the output at k . Since most current jet engines have many data points available during each flight, a two-point time lag for image processing is acceptable.

Since median filters took many iterations to converge, researchers came up with the recursive median filter which has memory. A recursive median (RM) filter for window length $N = 2n + 1$ can be represented as:

$$y_k = \text{median} (y_{k-n}, y_{k-n+1}, \dots, x_k, \dots, x_{k+n-1}, x_{k+n}) \quad (23.2)$$

RM filters converge fast compared to SM filters. A 5-point RM filter can be written as $y_k = \text{median}(y_{k-2}, y_{k-1}, x_k, x_{k+1}, x_{k+2})$, where the use of previously filtered

output values y_{k-1} and y_{k-2} point to the *recursive* nature of this filter. Again, this filter has a two-point time delay. However, RM filters suffer from streaking or introduction of step-like artifacts in the signal.

The WRM filter is a modified version of RM filter, where *integer* weights are assigned to each data point in the filter window. The output of a weighted recursive median filter with window length $N = 2n + 1$ is given by:

$$y_k = \text{median}(w_{-n} \circ y_{k-n}, w_{-n+1} \circ y_{k-n+1}, \dots, w_0 \circ x_k, \dots, w_{n-1} \circ x_{k+n-1}, w_n \circ x_{k+n}) \quad (23.3)$$

Here \circ stands for duplication and w are the integer weights. Duplication means that a particular sample x_k is repeated w_k times before taking the median of the array. For example, $(4 \circ x_1)$ is the same as $(x_1 x_2 x_3 x_4)$, i.e., the value x_1 is duplicated four times. As an example, consider a 5-point WRM filter given as $y_k = \text{median}(2 \circ y_{k-2}, y_{k-1}, 3 \circ x_k, x_{k+1}, 2 \circ x_{k+2})$. This is identical to $y_k = \text{median}(y_{k-2}, y_{k-2}, y_{k-1}, x_k, x_k, x_k, x_{k+1}, x_{k+2}, x_{k+2})$. Again, this filter will have a two-point time delay. The filter in Eq. 23.3 has weight set $(w_{-n}, w_{-n+1}, \dots, w_0, \dots, w_{n-1}, w_n)$, where there are $N = 2n + 1$ weights. For a 5-point filter, the weight set is $(w_{-2}, w_{-1}, w_0, w_1, w_2)$. The weights often have considerable influence on the filter performance.

23.3 Gas Path Measurement Images

A turbofan jet engine in general consists of five modules: fan (FAN), low pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT), and low pressure turbine (LPT), as shown in Fig. 23.1. Air sucked into the engine is compressed in the FAN, LPC, and HPC modules; combusted in the burner; and then expanded through the HPT and LPT modules to produce power. This power is the main deliverable of the gas turbine machine and needs to be supplied without problems. Sensors are placed on this machine to observe its state at any given time. The four sensors N1, N2, WF, and EGT represent low rotor speed, high rotor speed, fuel flow, and exhaust gas temperature, respectively. These measurements provide information about the condition of these modules and are used for engine condition monitoring. In this chapter, an ideal root signal ΔEGT with implanted HPC and/or HPT faults is used for testing the filters. Similarly, root signals for ΔN1 , ΔN2 , and ΔWF can be derived. Here the delta “ Δ ” refers to the deviation from the baseline “good” engine.

Test signals for faults in jet engines were used to demonstrate the optimal WRM filter [7]. For a new undamaged engine, the measurement delta is zero. For a typical engine which goes into service, the measurement deltas slowly increase with time due to *deterioration* as the number of flights increase. While deterioration increases gradually as flight hours and cycles accumulate, *single faults* lead to sudden, abrupt,

or step changes in the signal. For this study, a step change in measurement deltas of 2% or more is regarded as a large enough trend to be interpreted as a single fault event [2]. Gaussian noise is added to the simulated measurement deltas using the typical standard deviations for ΔEGT (deviation in exhaust gas temperature from a good baseline engine) as 4.23 °C. These values are obtained by a study of typical engine measurement deltas. Measurement deltas are created using: $z = z^0 + \theta$, where θ is the noise and z^0 is the baseline measurement delta without any fault (ideal signal). Thus, z is the simulated noisy signal. Therefore, a filter ϕ is required to eliminate noise from data and return a filtered signal \hat{z} for accurate condition monitoring: $\hat{z} = \phi(z) = \phi(z^0 + \theta)$.

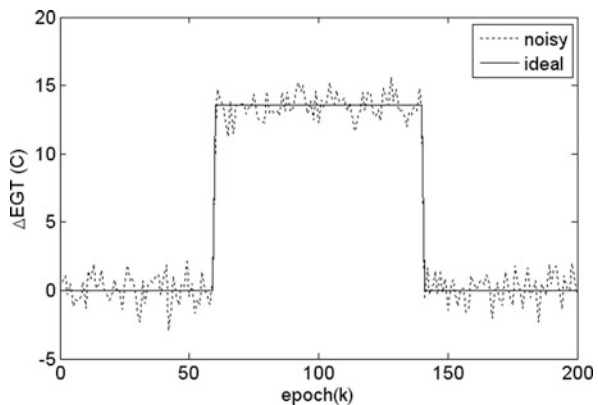
Three different types of signals are considered for designing the WRM filter using ACO. Though these signals are outlined for gas turbine diagnostics, they are applicable for any general machine FDI problem as the abrupt fault and long-term deterioration are characteristics of all signals used for condition monitoring. These signals are:

1. Step signal (abrupt fault or single fault)
2. Ramp signal (gradual fault)
3. Combination signal (comprising of both abrupt and gradual fault)

Each of the signals contains 200 data points which represent a time series of engine data available for signal processing. The data comes in at each epoch k (Fig. 23.3) and the filtered value is calculated using the N -point WRM filter. Figure 23.3 therefore represents an image for a single fault.

The filter of window length N processes the data as it comes into the information processing system with a time lag of n (Eq. 23.3). We use a 5-point filter in this chapter. So the WRM filter works on a stream of 200 data points ($x_1, x_2, x_3, \dots, x_{198}, x_{199}, x_{200}$) to yield filter output ($y_1, y_2, y_3, \dots, y_{198}, y_{199}, y_{200}$) as given below:

Fig. 23.3 Image representing a single fault and its repair



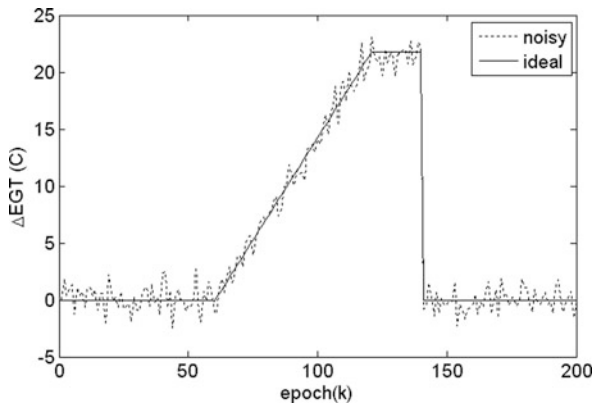
$$\begin{aligned}
 k = 1, y_1 &= x_1 \\
 k = 2, y_2 &= x_2 \\
 k = 3, y_3 &= \text{median}(w_{-2} \circ y_1, w_{-1} \circ y_2, w_0 \circ x_3, w_1 \circ x_4, w_2 \circ x_5) \\
 k = 4, y_4 &= \text{median}(w_{-2} \circ y_2, w_{-1} \circ y_3, w_0 \circ x_4, w_1 \circ x_5, w_2 \circ x_6) \\
 &\vdots \\
 k = 100, y_{100} &= \text{median}(w_{-2} \circ y_{98}, w_{-1} \circ y_{99}, w_0 \circ x_{100}, w_1 \circ x_{101}, w_2 \circ x_{102}) \\
 &\vdots \\
 k = 198, y_{198} &= \text{median}(w_{-2} \circ y_{196}, w_{-1} \circ y_{197}, w_0 \circ x_{198}, w_1 \circ x_{199}, w_2 \circ x_{200}) \\
 k = 199, y_{199} &= x_{199} \\
 k = 200, y_{200} &= x_{200}
 \end{aligned}
 \tag{23.4}$$

We see that at $k = 3$, to get y_3 requires x_4 and x_5 . So the filter has a two-point time delay. Also, for the last two points in the time series we use the input value of the data. However, in normal operation, the data points continue to stream in as the aircraft engine continues to accrue flights. The data point to be processed for fault detection is thus available with a two-point time delay. This data point can be used by trend detection algorithms which are typically based on derivatives [12]. So fault detection occurs with only a two-point time delay for the 5-point filter.

The ideal signal in Fig. 23.3 represents a single fault that may be due to any damage. Data point $k = 60$ represents the onset of this fault. The damage caused is identified as a 2% fall in HPC efficiency and the HPC module is repaired at point $k = 140$. This signal is created based on the fingerprint chart given in Table 23.1. In Fig. 23.4, the development of the HPT fault is illustrated by use of the ramp signal and the image corresponds to a gradual fault.

This fault differs from the HPC fault since it does not occur suddenly as it develops due to engine deterioration. Again, the maximum value of EGT here corresponds to a 2% fall in HPT efficiency. Here, the growth is gradual and is approximated by a

Fig. 23.4 Image representing a gradual fault followed by its repair



linear function from points $k = 40-120$. From $k = 120$ the HPT fault remains steady and is finally repaired at $k = 140$. The step and ramp signals represent the two types of faults considered individually.

Now, Fig. 23.5 shows a combination signal, wherein both types of faults may occur one after the other. This is a more practical case since any jet engine is susceptible to both these faults.

A signal to noise ratio of 1.5 is used for the numerical results. A 5-point WRM filter is considered and this filter processes the 200 point measurement delta signal with a window of 5 points and a time delay of two points.

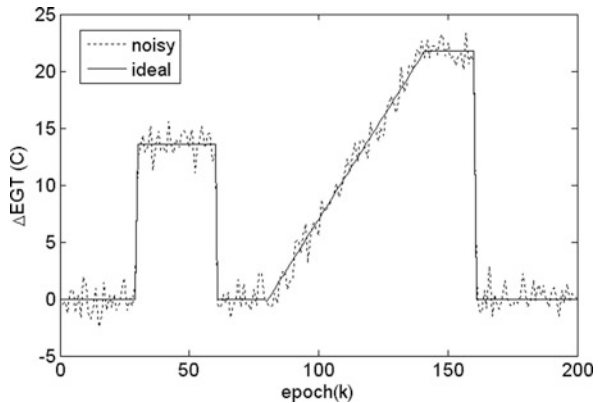
23.3.1 Objective Function

To get a quantitative idea of noise reduction, the mean absolute error (MAE) is considered for each signal ($\bar{N} = 200$) and $M = 1000$ random realizations are used to get an estimate of the error. These random realizations can be considered to be simulated signals of noisy data. These random signals are $z = z^0 + \theta$ and are obtained by adding a different noisy sample to the ideal measurement for each case. These noisy signals are generated via Matlab which is also used for all the results in this chapter. We illustrate the ACO algorithm for a filter of window length equal to five. For finding the optimum weights of this filter, we have to minimize the objective function:

$$f(w) = f(w_{-2}, w_{-1}, w_0, w_1, w_2) = \frac{1}{M} \sum_{i=1}^M \frac{1}{\bar{N}} \sum_{j=1}^{\bar{N}} |\hat{z}_j - z_j^0| \tag{23.5}$$

Since the weights $w = (w_{-2}, w_{-1}, w_0, w_1, w_2)$ are integer design variables, the problem is a combinatorial optimization problem. Also, in Eq. (23.5) \hat{z} is the WRM

Fig. 23.5 Image representing single fault and its repair followed by gradual fault and its repair



filtered signal and z^0 is the ideal or root signal. By minimizing the function in Eq. (23.5), we want to identify the weights which minimize the difference between the ideal signal and the noisy signal over a large number of noisy points. Note that the objective function with the 1000 random samples is only created to find the optimal filter weights. Once the filter weights are found, the filter can be tested and used on new random samples.

23.4 Ant Colony Optimization

ACO is a biologically inspired stochastic method which is highly appropriate for combinatorial optimization problems [21]. Such problems have a finite number of solutions with discrete design variables. Ants are able to find the shortest paths needed to travel from their nest to a food source by using *stigmergy*, which is a form of indirect communication conducted by changing the environment. Ants use sign-based *stigmergy* where an individual ant leaves markers on the path. While these markers do not solve the problem by themselves, they modify the behavior of other ants in a way that helps them in problem solution.

The motivation for the ACO algorithm came from some experiments performed on Argentine ants which unraveled the science behind their optimal path finding capabilities. An innovative experiment was performed using a double bridge between an ant nest and a food source. Here each bridge is of the same length as shown in Fig. 23.6.

If the bridges are of identical length, after some time, the ants start to take one of the bridges to the food source. If the experiment is repeated several times, it is found that the probability of selection of any one bridge is about 0.5. The biological explanation of the ant behavior is as follows. Once the ants leave the nest, they will move randomly around their starting location until some find the bridge. Some ants will randomly start on bridge A while others will randomly start on bridge B. Now, ants deposit a chemical called *pheromone* when they travel along a path. They prefer to follow a path with higher pheromone deposits. Since, there is no pheromone

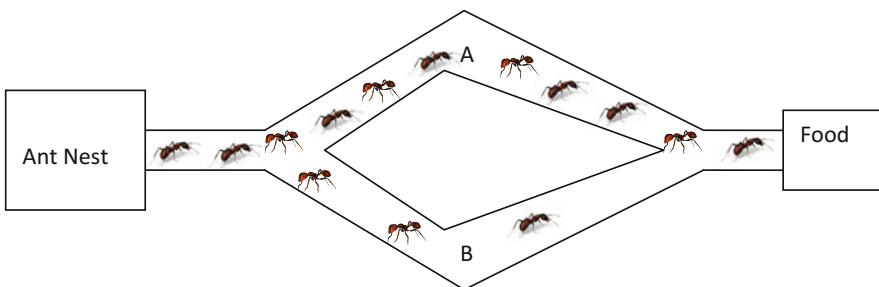


Fig. 23.6 Two-bridge experiment with ants for bridges of equal length

initially on either bridge A or B, the probability of ants taking either bridge is equal at 0.5. Once the ants discover the food source, they will pick up some food and return back to the nest. This process will lead to ants traveling on both bridges until through random chance; a few more ants take, say, bridge A. After this point, the *pheromone trail* on bridge A will strengthen, making it more attractive for the ants. Another important point to appreciate is that pheromone keeps evaporating and so the pheromone trail on bridge B will weaken. After some time, almost all ants will take bridge A.

Of course, taking only one bridge to the food source when two bridges of equal length are available is not very smart from the optimization viewpoint, though it may have other reasons based on sociology. However, ants are prisoners of their swarm intelligence which becomes very useful in the situation where two paths of different lengths are present, as shown for the two bridges in Fig. 23.7.

In this case, the ants again start out initially in a random manner and take both the bridges with equal probability assuming that the lengths are too large to be estimated by their vision. The ants which choose the shorter bridge B reach the food source first. When the ant wants to return to its nest, it comes to point 2 and finds that bridge B has a higher level of pheromone because of less evaporation. The probability of choosing bridge B therefore becomes more. As this process continues, the positive feedback effect means that more pheromone is put on bridge B and less evaporation of pheromone takes place on bridge B. A positive feedback loop is created and after some time, most ants will travel by bridge B to the food source. We can see that ants are capable of a high degree of *self-organization* using the stigmergy principle. By modifying the environment using pheromones, they can collectively perform complex functions despite their poor vision. In fact, some species of ants are completely blind but are still able to find the shortest path.

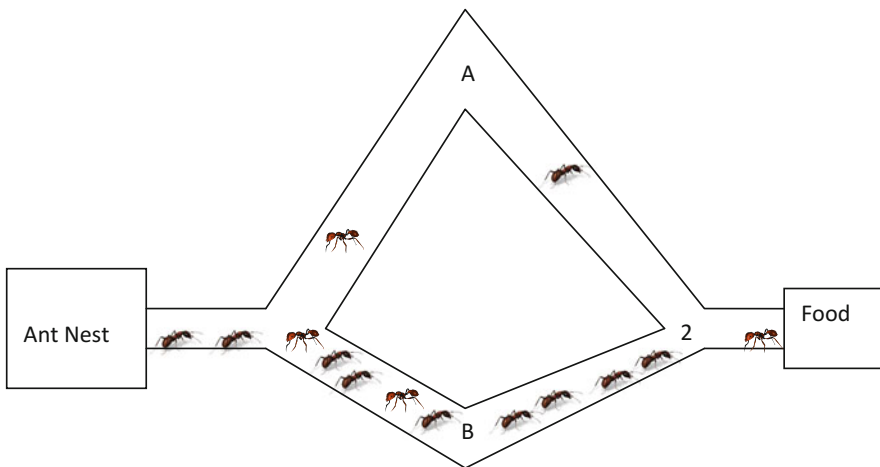


Fig. 23.7 Two-bridge experiment with ants for bridges of unequal length

The behavior of ants could be used for optimization algorithms which involve finding good paths through graphs [18, 22]. In ACO, several generations of artificial ants search for good solutions. Note that “good” may not be “optimal” in a mathematical sense but will be a practical and useful outcome. We use the word “ant” to refer to the “artificial ant” in the ACO algorithm in further discussions. Every ant of a generation constructs a solution in a step by step manner while going through several probabilistic decisions. In general, ants that find a good solution mark their paths through the decision space by putting some amount of pheromone on the edges of the path. The ants of the next generation are attracted by the pheromone trail left behind by the earlier ants, so they search the solution space near good solutions. In addition to the pheromone values, the ants are also guided by some problem-specific heuristics (a rule of thumb specific to the problem being solved). The ACO amalgamates a priori information about a promising solution with a posteriori information about previously obtained good solutions.

ACO was used to solve combinatorial optimization problems [21, 23]. An assignment problem is a combinatorial optimization problem where a set of items or objects is assigned to a set of resources or locations. Such assignments can be represented as a mapping from a set I to a set J . The objective function to be minimized is a function of the assignment done. Consider the pair (i, j) where i is an item and j is a resource. The pheromone trail τ_{ij} is the desirability of assigning i to j .

We can see that the WRM filter optimization is an assignment problem, wherein integer weights are assigned to a particular data point of a WRM filter with the objective of minimizing the mean absolute error over M samples. We seek to find the weight vector w which minimizes Eq. 23.5. Consider the 5-point WRM filter with weights $w = (w_{-2}, w_{-1}, w_0, w_1, w_2)$. We want to assign weights from set of integers $(1, 2, 3, 4)$ to minimize the error in Eq. 23.5.

23.4.1 Ant Colony Algorithm

This section describes the various components of ACO.

Set of initial solutions: This set is created using solutions that are not repeated and no two solutions in this set can be converted into each other by swapping of elements.

Pheromone trail matrix: A key feature of ACO is the pheromone trail management. Along with the objective function, pheromone trail values are applied to construct new solutions from the existing ones. Pheromone trail values measure the desirability of having an element in the solution. These are maintained in a pheromone matrix T with elements τ_{ij} .

23.4.2 Filter Weight Optimization

This section discusses about the application of ACO to WRM filter optimization problem.

Initialization of solution: Initially, every ant k is allotted a randomly selected solution w^k such that no two ants have the same initial solution. A total of m ants are used. Each initial ant solution is improved using a local search procedure and the best solution is labelled as w^* .

Pheromone trail initialization: Pheromone matrix component τ_{ij} measures the desirability of allotting weight w_i to a j th data point in the N -point filter. In the weight assignment problem, T matrix size is $N \times \text{Max}$, where N is the number of points in the WRM filter and Max is the maximum positive integral value that a weight can be assigned. Here $\text{Max} = 4$ and $N = 5$ are considered. The pheromone matrix T is created by setting all the pheromone trails τ_{ij} to the same initial value τ_0 . The pheromone trails determine the quality of the solution obtained, hence τ_0 must take a value that depends on value of the solution. Therefore, we chose to set $\tau_0 = Q/f(w^*)$, where w^* represents the best initial solution and Q is a constant to be found from numerical experiments.

Solution construction: Pheromone trail based modification is applied by every ant to its own solution w^k . It consists of the following steps. Any arbitrary filter data point r of the N -point filter is selected and then a second point s is chosen such that $s \neq r$ and the weights w_r and w_s are swapped in the current solution w . The second index s is selected such that the value of $\tau_r w_s + \tau_s w_r$ is maximum. By exploiting the pheromone trail, a new solution \hat{w}^k is obtained for each ant, which gives the most desirable path with the highest pheromone value.

Local search modification: Local search involves exploring of the neighborhood of current solution \hat{w}^k . It involves changing weights w_r while keeping the other weights constant to produce \hat{w}^k . The improvement is recorded as $\Delta(\hat{w}^k, r, s)$ which is the difference in objective function $f(w)$ when weight w_r is changed with s , where s can be any integral weight excluding w_r . This procedure is repeated for all the data points of the filter. Using the objective function as a measure, we find the optimum solution \tilde{w}^k . If no improvement is found, then no change is made to the earlier solution \hat{w}^k obtained by the ants.

Pheromone trail modification: There are several different pheromone update rules. For the current problem, we use the ant colony system (ACS) pheromone update. Each ant applies this update:

$$\tau_{ij} = (1 - \alpha) \tau_{ij} + \alpha \tau_0 \quad (23.6)$$

Here, α is a parameter that controls pheromone evaporation and is named the pheromone evaporation rate ($0 < \alpha < 1$).

Terminating condition: The termination condition is reached when a predefined number of ant generations ($niter$) finish completing their search in the solution space.

The different steps of algorithm are enumerated as pseudocode below.


```

Generate  $m$  ants with each ant being given different
weight permutation  $w^k$ .
/*Initialization*/
Generate all possible different non-arranged
permutations and randomly assign them to  $m$  ants.
Improve the weights  $w^1, w^2, w^3, \dots, w^m$  by the local
search procedure. Let  $w^*$  be the best solution.
Initialize the pheromone matrix  $T$ 
/* main loop */
For  $i = 1$  to  $niter$ 
    /* solution construction */
    For each permutation  $w^k(i)$  ( $1 \leq k \leq m$ ) do
        Apply  $r$  pheromone trail swaps to  $w^k(i)$  to obtain  $\hat{w}^k(i)$ 
        Apply the local search procedure to  $\hat{w}^k(i)$  to obtain
 $\tilde{w}^k(i)$ 
    End For
    /* pheromone trail updating */
    Update the pheromone trail matrix
    End for if /* terminating condition */
The parameters  $Q$ ,  $niter$ ,  $m$ , and  $\alpha$  are obtained from numerical experiments.

```

23.5 Numerical Experiments

The ACO was tested for different parameter setting of Q , number of iterations $niter$, number of ants m , and pheromone evaporation rate α for 100 noisy ramp input signals. The number of ants m was varied from 2 to 10 while number of iterations $niter$ was varied from 1 to 10. The evaporation rate was varied from 0.1 to 0.9 and a rate of 0.4 was found to be good.

The number of ants is the main parameter for a good quality solution. The solutions improve with increasing number of ants as seen from Table 23.2 for one case with three iterations. The optimum number of ants is 8–10. Therefore, ten ants are selected for finding out best value of number of iterations.

The optimum number of iterations was found to be three on the basis of solution quality and simulation time as seen from Table 23.3. The final parameter settings are selected to be: $\alpha = 0.4$, $Q = 0.1$, $niter = 3$, and $m = 10$. Numerical experimentation shows that the parameters obtained are not dependent on the type of noisy signal used. The ACO algorithm is finally applied to the three different types of noisy test signals.

The optimal filter weights obtained using ACO are shown in Table 23.4. The performance of WRM filter is compared with the performance of SM and RM filters in Table 23.5. These comparisons are for a different set of 1000 noisy data points compared to those used for finding the filter optimal weights.

Table 23.2 Change in objective function with number of ants ($niter = 3$)

No. of ants	MAE value	Best weight
1	0.3209	[1 4 3 1 3]
2	0.2899	[3 3 2 2 4]
3	0.3206	[3 1 3 4 1]
4	0.2854	[3 1 2 1 3]
5	0.2854	[4 1 2 1 4]
6	0.2854	[3 1 2 1 3]
7	0.3206	[3 1 3 4 1]
8	0.2817	[4 1 2 2 3]
9	0.2854	[4 1 2 1 4]
10	0.2771	[4 1 2 2 3]

Table 23.3 Change in objective function with number of iterations ($m = 10$)

Iteration	MAE value	Weight
2	0.2795	[3 1 2 1 3]
3	0.2771	[4 1 2 2 3]
4	0.2771	[4 1 2 2 3]
5	0.2795	[3 1 2 1 3]

Table 23.4 Optimal WRM filter weights

Signal type	w_{-2}	w_{-1}	w_0	w_1	w_2
Step	4	2	3	1	4
Ramp	4	1	2	2	3
Combination	3	1	2	1	3

Table 23.5 Mean absolute error for the filters

Signal type	SM filter	RM filter	WRM filter
Step signal	0.3638	0.3031	0.2426
Ramp signal	0.3856	0.3739	0.2773
Combination signal	0.3999	0.3930	0.3027

To quantify the advantage of using the optimal WRM filter for noisy data, we define a noise reduction measure as follows:

$$\rho = 100 \frac{\text{MAE}^{\text{noisy}} - \text{MAE}^{\text{filtered}}}{\text{MAE}^{\text{noisy}}} \quad (23.7)$$

Table 23.5 shows the improvement shown by the WRM filters. The WRM filter with weights given in Table 23.4 provides a noise reduction of about 52–64%. In contrast, the RM filter yields noise reduction of 41–55% and the SM filter shows a noise reduction of only 40–46%. Note that the RM filter could be considered as a WRM filter with unit weights. The improvement between SM and RM filter results is due to the introduction of recursion in the RM filter. The improvement between the RM and WRM results takes place because of optimal weights obtained using ACO. The values of noise reduction in Table 23.6 clearly justify the improved performance of WRM filters over SM and RM filters as illustrated by the simulated signals from gas turbine diagnostics.

Table 23.6 Noise reduction for simple median, recursive median, and ant colony designed weighted recursive median filters

Signal type	ρ^{SM}	ρ^{RM}	ρ^{WRM}
Step signal	45.87	54.90	63.90
Ramp signal	42.49	44.23	55.13
Combination signal	40.04	41.07	51.94

The WRM filter provides a noise reduction of about 52–64%. In particular, compared to the noisy signal, the noise reduction is 64%, 55%, and 52% for the step signal, ramp signal, and the combination signal, respectively. Therefore, ACO represents an effective approach for the development of WRM filters.

23.6 Conclusions

Removing noise from gas turbine measurement signals before subjecting them to fault detection and isolation algorithms is an important component of machine diagnostics. Nonlinear filters such as the WRM filter are attractive for these problems as they do not smooth out the step changes in signals which typically indicate the onset of a single fault. However, filters such as the WRM filter proposed for gas turbine machine diagnostics need to be optimized for the specific application.

The problem of finding optimal integer weights of WRM filters using ACO is addressed in this chapter. An analogy between the WRM filter weight optimization problem and the quadratic assignment problem is discovered. Images simulating abrupt and gradual faults are contaminated with noise and then used to find the WRM filter weights which minimize the noise. Numerical experiments are used to find the best parameters required for the ACO application. The WRM filters presented in this show noise reductions of 52–64% relative to the noisy signal compared to 41–55% for the RM filter, which uses unit weights. For the step, ramp, and combination images considered, noise reduction gains of about 9, 11, and 10% are obtained through filter optimization due to the use of weights obtained by ACO.

Acknowledgement The author thanks Mr. Chintan Raikar, Undergraduate Student from IIT Mumbai for running the simulations in this chapter.

References

1. Chauhan, V., & Surgenor, B. (2015). A comparative study of machine vision based methods for fault detection in an automated assembly machine. *Procedia Manufacturing*, 1, 416–428.
2. Volponi, A. J., Depold, H., Ganguli, R., & Daguang, C. (2003). The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparative study. *Journal of Engineering for Gas Turbine and Power*, 125(4), 917–924.

3. Lu, P. J., Zhang, M. C., Hsu, T. C., & Zhang, J. (2001). An evaluation of engine fault diagnostics using artificial neural networks. *Journal of Engineering for Gas Turbine and Power*, 123(2), 340–346.
4. Rodríguez-Quiñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
5. Miranda-Vega, J. E., Rivas-Lopez, M., Flores-Fuentes, W., Sergiyenko, O., Rodríguez-Quiñonez, J. C., & Lindner, L. (2019). Methods to reduce the optical noise in a real-world environment of an optical scanning system for structural health monitoring. In *Optoelectronics in machine vision-based theories and applications* (pp. 301–336). IGI Global.
6. Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57(2), 787–798.
7. Uday, P., & Ganguli, R. (2010). Jet engine health signal denoising using optimally weighted recursive median filters. *Journal for Engineering in Gas Turbine and Power*, 132(4), 41601–41608.
8. Ganguli, R. (2002). Fuzzy logic intelligent system for gas turbine module and system fault isolation. *Journal of Propulsion and Power*, 18(2), 440–447.
9. Ganguli, R. (2012). *Gas turbine diagnostics*. Boca Raton, FL: CRC Press.
10. Lu, P. J., & Hsu, T. C. (2002). Application of autoassociative neural network on gas-path sensor data validation. *Journal of Propulsion and Power*, 18(4), 879–888.
11. Surender, V. P., & Ganguli, R. (2004). Adaptive myriad filter for improved gas turbine condition monitoring using transient data. *ASME Journal of Engineering for Gas Turbines and Power*, 127(2), 329–339.
12. Ganguli, R., & Dan, B. (2004). Trend shift detection in jet engine gas path measurements using cascaded recursive median filter with gradient and laplacian edge detector. *ASME Journal of Engineering for Gas Turbine and Power*, 126, 55–61.
13. Yang, R., Gabbouj, M., & Neuvo, Y. (1995). Fast algorithms for analyzing and designing weighted median filters. *Signal Processing*, 41(2), 135–152.
14. Arce, G. R., & Paredes, J. L. (2000). Recursive weighted median filter admitting negative weights and their optimization. *IEEE Transactions in Signal Processing*, 48(3), 768–779.
15. Lin, Y. L., Chang, W. D., & Hsieh, J. G. (2008). A particle swarm approach to nonlinear rational filter modeling. *Expert Systems with Applications*, 34(2), 1194–1199.
16. Zhao, C., & Zhang, W. (2005). Using genetic algorithm optimizing stack filters based on MMSE criterion. *Image and Vision Computing*, 23(10), 853–860.
17. Karaboga, N., Kalinli, A., & Karaboga, D. (2004). Designing digital IIR filters using ant colony optimization algorithm. *Engineering Applications of Artificial Intelligence*, 17(3), 301–309.
18. Dorigo, M., Maniezzo, V., & Colnari, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transaction on Systems, Man and Cybernetics Part B Cybernetics*, 26(1), 29–41.
19. Raikar, C., & Ganguli, R. (2017). Denoising signals used in gas turbine diagnostics with ant colony optimized weighted recursive median filters. *INAE Letters*, 2(3), 133–143.
20. Brownrigg, D. R. K. (1984). The weighted median filter. *Communications of the ACM*, 27(8), 807–818.
21. Dorigo, M., & Stutzle, T. (2005). *Ant colony optimization*. New Delhi: Prentice Hall of India Private Limited.
22. Boryczka, U., & Kozak, J. (2015). Enhancing the effectiveness of ant colony decision tree algorithms by co-learning. *Applied Soft Computing*, 30, 166–178.
23. Gambardella, L. M., Taillard, E. D., & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *The Journal of the Operational Research Society*, 50(2), 167–176.

Chapter 24

Control and Automation for Miniaturized Microwave GSG Nanoprobing



Alaa Taleb, Denis Pomorski, Christophe Boyaval, Steve Arscott, Gilles Dambrine, and Kamel Haddadi

Acronyms

CPW	Coplanar waveguide
DUT	Device under test
GaN nanowires	Gallium nitride nanowires
GSG	Ground signal ground
HF	High frequency
HIL	Hardware-In-the-Loop
MEMS	Microelectromechanical systems
PID controller	Proportional–integral–derivative controller
RF	Radio frequency
SEM	Scanning electron microscope

A. Taleb
Univ. Lille, CNRS, Centrale Lille, UMR 9189—CRISTAL—Centre de Recherche en Informatique, Signal et Automatique de Lille, Lille, France

D. Pomorski (✉)
Univ. Lille, CNRS, Centrale Lille, UMR 9189—CRISTAL—Centre de Recherche en Informatique, Signal et Automatique de Lille, Lille, France
Univ. Lille, IUT A—Département GEII, Lille, France
e-mail: denis.pomorski@univ-lille.fr

C. Boyaval · S. Arscott · G. Dambrine
Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520—IEMN, Lille, France
e-mail: christophe.boyaval@univ-lille.fr; steve.arscott@univ-lille.fr;
gilles.dambrine@univ-lille.fr

K. Haddadi
Univ. Lille, CNRS, Centrale Lille, ISEN, Univ. Valenciennes, UMR 8520—IEMN, Lille, France
Univ. Lille, IUT A—Département GEII, Lille, France
e-mail: kamel.haddadi@univ-lille.fr

SWNT	Single-walled nanotube
VNA	Vector network analyzer

24.1 Introduction

24.1.1 Context

To drive the progress of the miniaturization of electronic circuits, new metrological issues related to the dimensional and electrical characterization of nanoelectronic devices must be addressed [1]. In addition, the electrical characterization of high-impedance 1D or 2D based nanodevices in the microwave regime is still challenging [2]. A typical high-frequency (HF) device characterization is built up with a vector network analyzer, a probe station equipped with a pair of microwave GSG probes aligned manually through a microscope or a camera system onto calibration substrates and test devices [3, 4]. Conventional HF test structures require probing pads around $50 \times 50 \mu\text{m}^2$ to accommodate the probe tip geometry (center-to-center pitch of $100 \mu\text{m}$, contact area of $20 \times 20 \mu\text{m}^2$). The extrinsic parasitic capacitance associated with the pad in the range of 50 fF is therefore not compatible to address the metrology of nanodevices. Furthermore, actual visualization and displacement/positioning techniques are not accurate enough to ensure a repeatable contact between the probe tips and the pads at the micro- and nanoscale.

Intensive research has been described in the literature to address RF metrology at the nanoscale. In 2005, the first measurements of the high-frequency conductance of a metallic single-walled nanotube (SWNT) with resistance below $200 \text{ k}\Omega$ inserted in a coplanar waveguide (CPW) transmission line were performed up to 10 GHz [5]. In 2008, an on-wafer technique and calibration method are developed for broadband electrical characterization of GaN nanowires up to 40 GHz [6]. In 2010, to improve the vector network analyzer (VNA) sensitivity, an individual SWNT is inserted in a specific high impedance Wheatstone bridge helps to reduce the impedance mismatch between the VNA and the high-impedance nanodevice [7]. Other indirect measurements including nanotube transistors acting as resonators [8] or microwave detectors [9] have been proposed to demonstrate their GHz operation. Despite these pioneering works, in the era of shrinking GSG probing structures, a gap between commercially available probes and those required to characterize nanodevices still remains.

The objective of this work is to develop a new generation of on-wafer probing instrumentation dedicated to HF quantitative characterization of micro- and nanodevices. At such scale, visualization, accuracy of alignment, positioning, and repeatability require suited techniques. In the solution proposed, the probes are mounted on nanopositioners, and the visualization is ensured by a scanning electron microscopy (SEM) rather than optics. This method suffers from measurement repeatability and accuracy issues. Consequently, we have developed a unique instrument that is a compromise between conventional on-wafer probe station and microscopy tools. We have fabricated microelectromechanical system (MEMS) technology-based miniaturized microwave ground-signal-ground (GSG) probes

[10]. In contrast to conventional macroscopic on-wafer probing structures, micrometric CPW test structures have been designed and fabricated to accommodate the miniaturized probes and to ensure quasi-transverse electromagnetic (quasi-TEM) mode propagation to the nanoscale devices embedded in the test structures [11]. The probes are mounted on nanopositioners, and imaging is ensured by an SEM. A detailed study on the development of the nanorobotic on-wafer probe station is given in [12].

24.1.2 Short Description of the SEM

The SEM [13] (Fig. 24.1) consists of observing the topography of a surface (a substrate). Its operation relies essentially on the detection of the secondary electrons (1) emerging from the observed surface (2) under the impact of a primary electron beam (3) which scans it. The images obtained from the substrate (Fig. 24.2) have a separating power often less than 5 nm and a great depth of field.

Without going into details of the internal functioning of the SEM, the general idea of this work consists of positioning a probe consisting of three aligned points on an element (pattern) of the substrate (Fig. 24.2). It is therefore a system with four degrees of freedom requiring the control of the displacements of three nanopositioners SmarAct™ (Fig. 24.3) [14] in X, Y, and Z and the control of one nanopositioner around an axis of rotation θ .

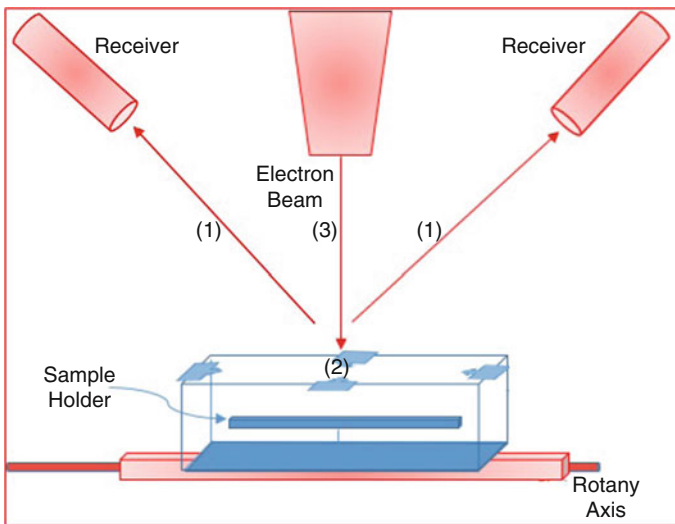


Fig. 24.1 Schematic layout of the SEM

24.1.3 Specifications

Our specifications are as follows:

First, we propose to model and control any linear nanopositioners. In this study, we focus on three linear nanopositioners X , Y , and Z (Sect. 24.2).

Both linear nanopositioners X and Y can be controlled by taking into account a minimum response time. The linear nanopositioner Z must be controlled without overshooting the set point (in order to avoid any crashing of the probe tips on the DUT).

Section 24.3 provides an approach for controlling the nanopositioner in θ in order to align the probe on the image.

Finally, a simple approach for detecting points of interest (Harris method) allows determining the set point value of each nanopositioner in X , Y , or Z (Sect. 24.4).

The overall process allows us to position the probe accurately at any point of the substrate.

24.2 Modeling and Control of a Linear Nanopositioner Using LabVIEW™

24.2.1 Central Idea of This Study

The main idea of this study is to fully master the nanomanipulator control chain from two elementary blocks using LabVIEW™ (Fig. 24.4) [15, 16]:

- A set point or control block
- A block for acquiring the actual position of the nanopositioner

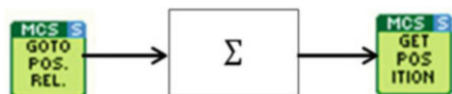
These two elements allow a real-time interaction with the nanopositioner (Fig. 24.5).

So we can use a conventional control loop to control the nanomanipulator (Fig. 24.6). A HIL (Hardware-In-the-Loop) test is thus carried out.



Fig. 24.4 Blocks used with LabVIEW™: (a) Set point block. (b) Position acquisition block

Fig. 24.5 Control of a nanopositioner with LabVIEW™



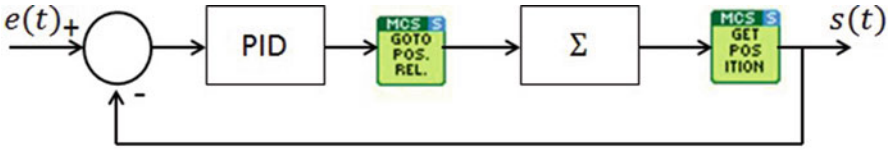


Fig. 24.6 Control loop for the nanopositioner, where $e(t)$ represents the required position of the nanopositioner—it is the set point; $s(t)$ is its measured/actual position; the PID controller continuously applies a correction based on proportional, integral, and derivative terms from the error value $\varepsilon(t)$ which is the difference between the required position (the set point $e(t)$) and the measured/actual position $s(t)$

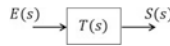


Fig. 24.7 Open-loop transfer function of the nanopositioner, where $E(s)$ is the Laplace transform of the required (desired) position $e(t)$; $S(s)$ is the Laplace transform of the actual position $s(t)$; $T(s)$ is the transfer function of the nanopositioner. $T(s)$ represents the Laplace transform of the impulse response of the nanopositioner

In order to obtain an accurate control of the nanomanipulator, we must identify its transfer function. We considered that the system is linear. This assumption is not contradicted by the experiments.

We propose to identify its linear transfer function using a (or some) basic identification method(s).

24.2.2 Modeling

After testing two open-loop identification methods, we propose a more accurate closed-loop identification technique.

24.2.2.1 Identification of the Open-Loop Transfer Function of the Nanopositioner

First, let us try to identify the open-loop transfer function of the nanopositioner (Fig. 24.7).

For instance, the set point value is fixed at 1000 nm. Figure 24.8 shows the required position (in red) and the actual position (in blue) of the nanopositioner with respect to time. For a greater clarity of the curves, the X-axis is graduated in tenths of a second (1 unit = $\frac{1}{10}$ s), which represents the sampling period of the measured position.

The nanopositioner reacts globally as a pure integrator. This property allows us to assume that the closed-loop permanent error (difference between the output and

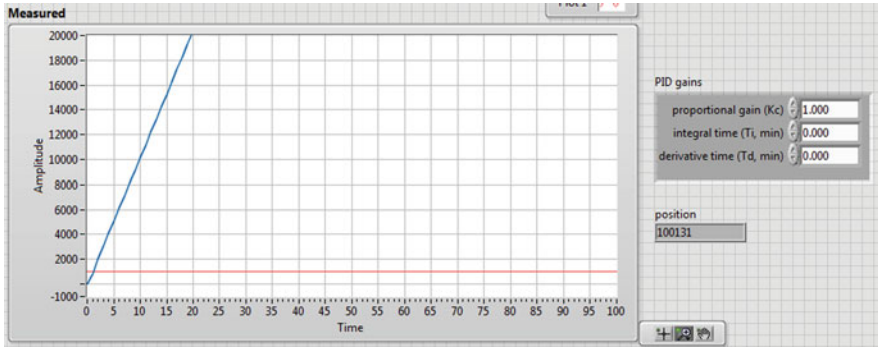


Fig. 24.8 Open-loop response with a set point of 1000 nm of the nanopositioner

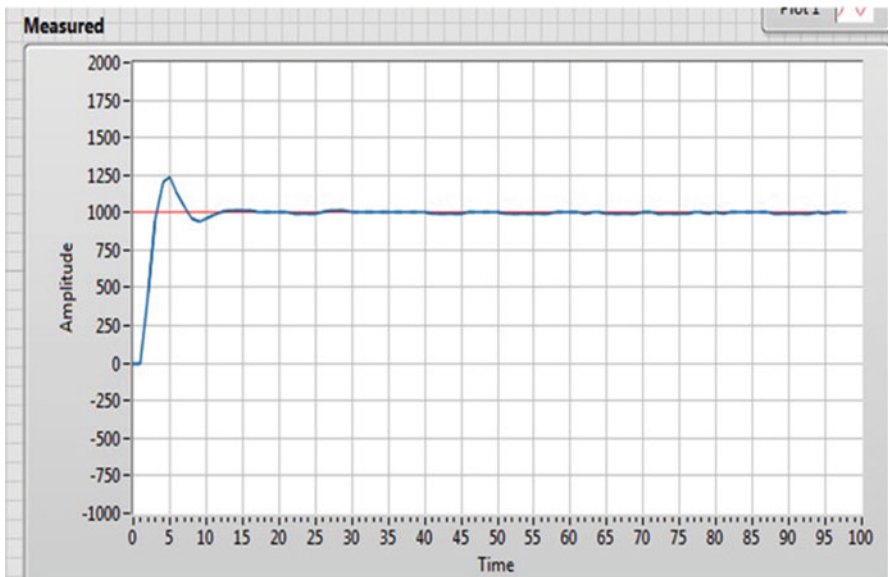


Fig. 24.9 Closed-loop response of the nanopositioner for a set point of 1000 nm

the set point for a constant set point value when $t \rightarrow \infty$) is zero. This is confirmed by the actual closed-loop response (Fig. 24.9) without PID controller (or with a proportional controller with a gain factor equal to 1).

A proportional correction seems sufficient (Fig. 24.10).

The identification of the nanopositioner transfer function can be performed only by the hypothesis of a mathematical expression.

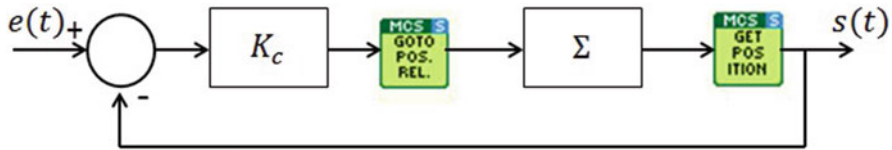


Fig. 24.10 A proportional correction of the nanopositioner

First hypothesis

If we consider that the nanopositioner behaves as a pure integrator, its transfer function is as follows:

$$T(s) = \frac{K}{s} \quad (24.1)$$

From the identification, we obtain $K = 1.05$.

However, the closed-loop nanopositioner shows that for some values of K_c , there is an overshoot of the set point value (e.g., Fig. 24.9).

Consequently, the first hypothesis must be rejected, in favor of a second-order model at least.

Second hypothesis

The transfer function is as follows:

$$T(s) = \frac{K}{s(1 + \tau s)} \quad (24.2)$$

The expression “ $(1 + \tau s)$ ” will have a preponderant role only when $t \rightarrow 0$ and it can be considered as a time delay (dead time τ), using the Taylor series expansion: $e^{-\tau s} \sim \frac{1}{1 + \tau s}$.

From the previous identification, $\tau = 1 \text{ unit} = 100 \text{ ms}$. The transfer function thus becomes

$$T(s) = \frac{1.05}{s(1 + s)} \quad (24.3)$$

However, the closed-loop responses of the model (Fig. 24.11) and the real system (Fig. 24.9) are too distinct to consider the model to be correct.

We can conclude that the method used to identify the open-loop transfer function is not accurate, given the sampling period of 100 ms. We have therefore chosen to identify the parameters K and τ differently using an identification of the closed-loop transfer function.

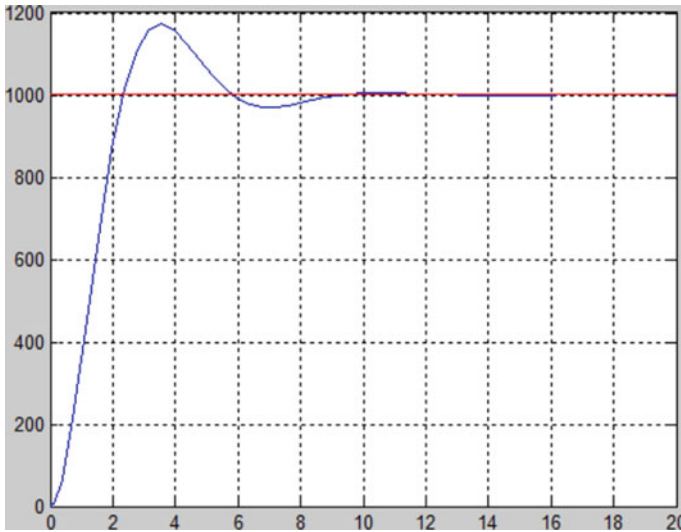


Fig. 24.11 Closed-loop response of the model (Eq. 24.3) for a set point of 1000 nm

24.2.2.2 Nanopositioning in Closed Loop

Taking into account the second hypothesis for the open-loop transfer function $T(s) = \frac{K}{s(1+\tau s)}$ and a proportional controller with a gain factor equal to K_c , the closed-loop transfer function of the nanopositioner is as follows:

$$H(s) = \frac{S(s)}{E(s)} = \frac{1}{1 + \frac{1}{KK_c} s + \frac{\tau}{KK_c} s^2} \tag{24.4}$$

We can rely on the well-known equations of the automatic control of a linear system:

- The canonical form of a second-order transfer function

$$H(s) = \frac{S(s)}{E(s)} = \frac{K}{1 + \frac{2z}{\omega_n} s + \frac{1}{\omega_n^2} s^2} \tag{24.5}$$

where K is the gain, z is the damping factor, and ω_n is the undamped natural frequency.

- The first overshoot with respect to the damping factor

$$D = e^{\frac{-\pi z}{\sqrt{1-z^2}}} \tag{24.6}$$

- The period of the pseudo-oscillation with respect to the damping factor and the undamped natural frequency

$$T = \frac{2\pi}{\omega_n \sqrt{1-z^2}} \quad (24.7)$$

From Eqs. (24.4)–(24.7), the parameters τ and K can be determined as functions of T and D .

Comparing Eqs. (24.4) and (24.5), we obtain

$$K = 1; \quad \omega_n = \sqrt{\frac{K \cdot K_c}{\tau}}; \quad z = \frac{1}{2\sqrt{\tau \cdot K \cdot K_c}} \quad (24.8)$$

By multiplying ω_n by z , we get τ , that is,

$$\tau = \frac{1}{2 \cdot \omega_n \cdot z} \quad (24.9)$$

From Eqs. (24.7) and (24.9), we obtain

$$\tau = \frac{1}{2 \cdot \frac{2\pi}{T \sqrt{1-z^2}} \cdot z} \quad (24.10)$$

And using Eq. (24.6),

$$\tau = \frac{-T}{4 \cdot \ln D} \quad (24.11)$$

The above equation allows determining τ .

By dividing ω_n by z from Eq. (24.8),

$$K \cdot K_c = \frac{\omega_n}{2 \cdot z} \quad (24.12)$$

From Eq. (24.7), $\omega_n = \frac{2\pi}{T \cdot \sqrt{1-z^2}}$ and from Eq. (24.6), $z = -\frac{1}{\pi} \sqrt{1-z^2} \ln D$. So we get

$$\frac{\omega_n}{z} = -\frac{2 \cdot \pi^2}{T \cdot (1-z^2) \cdot \ln D} \quad (24.13)$$

From Eq. (24.6), we also have

$$z^2 = \frac{(\ln D)^2}{\pi^2 + (\ln D)^2} \quad (24.14)$$

Substituting z^2 in Eq. (24.13),

$$\frac{\omega_n}{z} = -\frac{2 \cdot (\pi^2 + (\ln D)^2)}{T \cdot \ln D} \quad (24.15)$$

And therefore, from Eq. (24.12)

$$K \cdot K_c = -\frac{\pi^2 + (\ln D)^2}{T \cdot \ln D} \quad (24.16)$$

In conclusion, Eqs. (24.11) and (24.16) can be used to determine the parameters K and τ from:

- The identification of the parameters T and D of the step response of the closed-loop nanopositioner
- The knowledge of K_c of the controller fixed by the user

Experimentally, the closed-loop nanopositioner is subject to a gain $K_c = 1$. We identified the following parameters: an overshoot D of 25% and a period of oscillations $T = 800$ ms (i.e., 8 units). From Eqs. (24.11) and (24.16), we get

$$\tau = 1.443 \text{ units} = 144.3 \text{ ms} \quad \text{and} \quad K = 1.0632 \quad (24.17)$$

The transfer function of the open-loop nanopositioner is thus

$$T(s) = \frac{1.0632}{s(1 + 1.443s)} \quad (24.18)$$

Finally, the closed-loop responses of the model (Fig. 24.12) and the real nanopositioner (Fig. 24.9) are strictly identical.

24.2.3 Control with LabVIEW™

For all tested values of K_c , the closed-loop responses of the model and the nanopositioner are strictly identical.

Some values of K_c are as follows:

- For $K_c = 0.33$, the nanopositioner has a minimal response time, that is, the damping factor of the closed-loop nanopositioner is $z = \frac{1}{\sqrt{2}}$ (Fig. 24.13a)
- For $K_c = 0.163$, there is no overshoot of the nanopositioner, that is, $z = 1$ (Fig. 24.13b)

Whatever the set point and gain K_c , in the physical limits of the nanopositioner, the closed-loop responses of the real nanopositioner (Fig. 24.14a) and the model

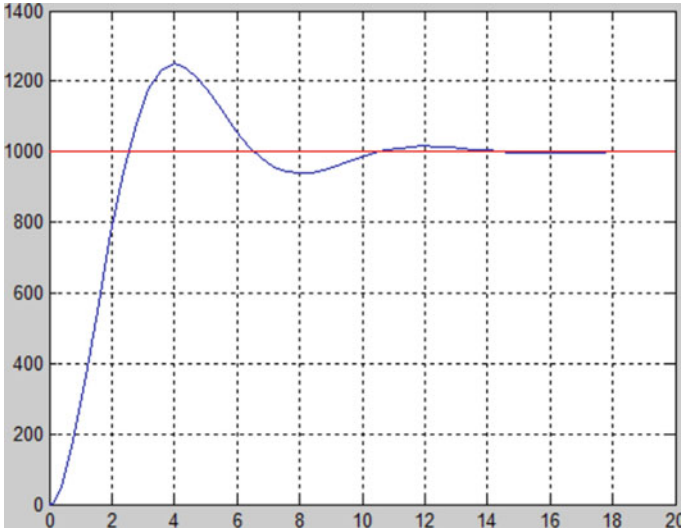


Fig. 24.12 Closed-loop response of the model (Eq. 24.18) for a set point of 1000 nm

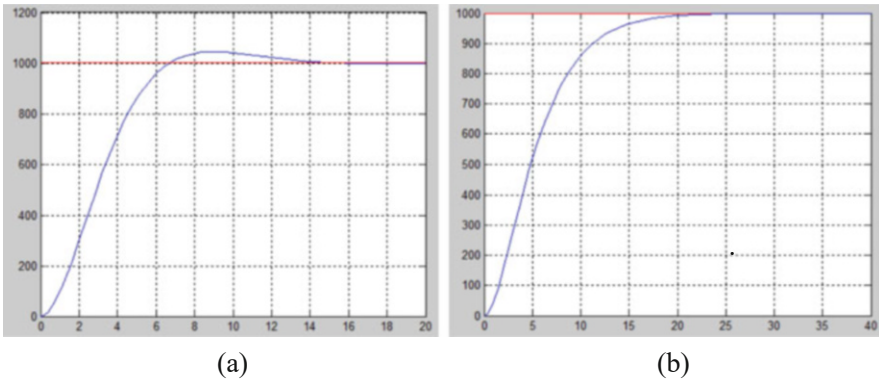


Fig. 24.13 Control of the nanopositioner: (a) $K_c = 0.33$; (b) $K_c = 0.163$

(Fig. 24.14b) are strictly identical. As an example, the responses to a set point of 2000 nm and $K_c = 1$ are given below.

We can conclude that the nanopositioner has been correctly modeled. The minimal response time of this one is obtained for $K_c = 0.33$.

Depending on the Z axis, the nanopositioner must not overshoot the set point value, otherwise the probe may be crushed (thus breaking) onto the substrate. Just set $K_c = 0.163$. The controls in X and Y can be realized in minimal response time ($K_c = 0.33$).

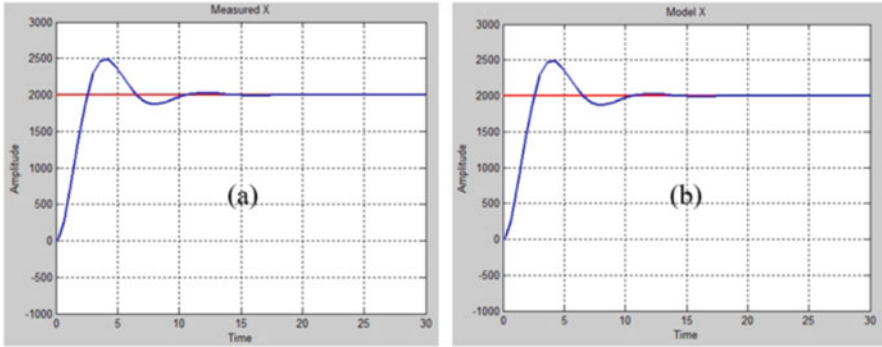


Fig. 24.14 (a) Nanopositioner response, (b) model response; set point = 2000 nm and $K_c = 1$

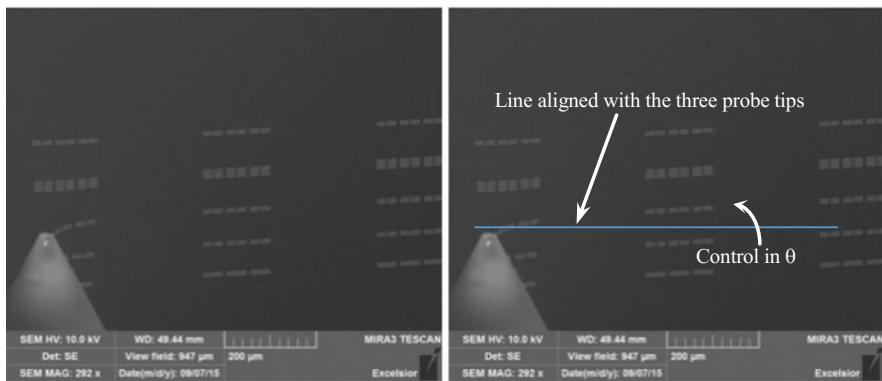


Fig. 24.15 Principle of alignment of probe tips with the patterns

24.3 Angular Control: Feasibility Study with Matlab™

In this section, we propose to determine a simple real-time method for aligning the substrate patterns on an imaginary line (blue line in Fig. 24.15) through the three probe tips, by using an angle control of the sample holder.

This feasibility study is based on simple and efficient image processing, which can be performed under Matlab™.

Figure 24.15 shows a depth image whose pixels are gray levels (0 for black, corresponding to a distant pixel; 255 for white, corresponding to the closest pixel).

An exhaustive solution consists of scanning all the possible rotations of the image (function `imrotate(image, theta)` under Matlab™) and retaining the one for which the sum of the gray levels on the imaginary blue line is maximum. There is thus a maximum of white pixels on this line (Fig. 24.16).

Fig. 24.16 Alignment of probe tips with the patterns



Another solution would be to use the gradient method to obtain a local maximum that corresponds to a locally optimal angle. The general idea of this method is to perform a rotation as long as the sum of the gray levels is higher than in the previous iteration.

The implementation of these solutions (exhaustive method and gradient method) is not problematic and will allow one to control the sample holder in real time.

24.4 Determining Set Points of the Nanopositioners on X, Y, and Z Axes

In this section, we propose a simple image processing allowing the detection of forms on the one hand, and the detection of points of interest on the other hand. The coordinates of these points of interest will represent the set point values of the three linear nanopositioners X, Y, and Z.

24.4.1 Detecting the Patterns

The general idea is to transform a grayscale image into a binary image. A threshold of the gray levels allows to obtain a black and white image. This threshold can be determined as the median of the gray levels of the initial image.

If the resulting binary image is too noisy, a filtering can be carried out by neglecting, for instance, all solid forms made up of fewer than P pixels (function `bwareaopen (bw_image, P)` under Matlab™).

P is a parameter that must be much smaller than the size (number of pixels) of the patterns, so that they are not removed in this filtering operation. P must also be large enough to provide efficient filtering.

Fig. 24.17 Detection of the patterns



Figure 24.17 shows the result obtained by this simple method from the image of Fig. 24.16.

24.4.2 Detecting a Point to Reach

In order to detect areas of interest in the grayscale image, a Harris method can easily be used [17, 18] (e.g., function `corner(bw_image, Nb_corners)` under Matlab™).

This method is used, for instance, to extract the corners of the contours. It is based on the derivative of the gray levels to locate the points where the intensity varies strongly in one or more directions.

For a given pixel (u, v) , let us consider

- Its pixel intensity $I(u, v)$
- Its neighborhood $w(u, v)$ —Harris and Stephens propose the use of a smooth circular window as a Gaussian filter $w(u, v) = \exp(-(u^2 + v^2)/(2\sigma^2))$

The average change of intensity for a small displacement (x, y) is

$$E(x, y) = \sum_{u, v} w(u, v) \cdot (I(x + u, y + v) - I(u, v))^2 \quad (24.19)$$

Consider the Taylor expansion of the intensity function I over the area (u, v)

$$I(x + u, y + v) = I(u, v) + x \frac{\delta I}{\delta x} + y \frac{\delta I}{\delta y} + o(x^2, y^2) \quad (24.20)$$

where $\frac{\delta I}{\delta x}$ and $\frac{\delta I}{\delta y}$ are the partial derivatives of I .

We obtain the following relation:

$$E(x, y) = \sum_{u,v} w(u, v) \cdot \left(x \frac{\delta I}{\delta x} + y \frac{\delta I}{\delta y} + o(x^2, y^2) \right)^2 \tag{24.21}$$

By neglecting the term $o(x^2, y^2)$ for small displacements, $E(x, y)$ can be expressed in the form

$$E(x, y) = Ax^2 + 2Cxy + By^2 \tag{24.22}$$

with $A = \frac{\delta I^2}{\delta x} \otimes w$; $B = \frac{\delta I^2}{\delta y} \otimes w$; $C = \left(\frac{\delta I}{\delta x} \frac{\delta I}{\delta y} \right) \otimes w$, where \otimes is the convolution function.

$E(x, y)$ can also be expressed in the form

$$E(x, y) = (x, y) M(x, y)^t \tag{24.23}$$

with $M = \begin{pmatrix} A & C \\ C & B \end{pmatrix}$. M is called the structure tensor. It is a symmetrical and positive matrix.

The matrix M characterizes the local behavior of the function E .

Indeed, the eigenvalues of this matrix correspond to the principal curvatures associated with E :

- If the two eigenvalues are large, then the intensity varies strongly in all directions. We have a corner.
- If the two eigenvalues are small, then the region under consideration has an approximately constant intensity. We have a homogeneous area.
- If the two eigenvalues are very different, we are in the presence of an outline.

Instead of using eigenvalues, Harris and Stephen propose to detect corners based on the following formula:

$$R = \text{Det}(M) - k \cdot \text{trace}(M)^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{24.24}$$

where $\text{Det}(M) = A \cdot B - C^2$ and $\text{trace}(M) = A + B$.

k is an empirically determined constant; $k \in [0.04; 0.06]$.

The values of R are positive close to a corner, negative near a contour, and weak in an area of constant intensity.

The search for corners in an image therefore consists of finding the local maxima of R .

This approach gives excellent results on the SEM images (Fig. 24.18).

The coordinates of these points of interest represent the X , Y , and Z set point values of the nanopositioners.



Fig. 24.18 Detecting the points to reach

24.5 Conclusions

This chapter presented an interdisciplinary approach to the control of nanomanipulators. First, we used classical automatic linear tools to identify the transfer function of a system of three nanopositioners along the X , Y , and Z axes. This part allows the precise control of any nanomanipulator in LabVIEW™, with overshoot (according to a minimal response time in X and Y) or without overshoot (in order to avoid crushing of the probe tips on the substrate in Z) of the required set point. Second, we designed an angular control methodology (under Matlab™) to align the probe tips with the component. Finally, the detection of points of interest (use of the Harris detector) makes it possible to fix the set point value of each nanopositioner in X , Y , and Z .

Acknowledgment This work is supported by the French National Research Agency (ANR) under the EquipEx Excelsior (www.excelsior-ncc.eu).

References

1. The International Technology Roadmap for Semiconductors (ITRS). (2013). Retrieved from <http://www.itrs.net/Links/2013ITRS/2013Chapters/2013ERD.pdf>.
2. Happy, H., Haddadi, K., Théron, D., Lasri, T., & Dambrine, G. (2014). Measurement techniques for RF nanoelectronic devices: New equipment to overcome the problems of impedance and scale mismatch. *IEEE Microwave Magazine*, 15(1), 30–39.
3. Rumiantsev, A., & Doerner, R. (2013). RF Probe Technology. *IEEE Microwave Magazine*, 14, 46–58.

4. Daffé, K., Dambrine, G., Von Kleist-Retzow, F., & Haddadi, K. (2016). RF wafer probing with improved contact repeatability using nanometer positioning. In *87th ARFTG Microwave Measurement Conference Dig, San Francisco, CA*, pp. 1–4.
5. Yu, Z., & Burke, P. J. (2005). Microwave transport in single-walled carbon nanotubes. *Nano Letters*, *5*(7), 1403–1406.
6. Wallis, T., Intiaz, A., Nembach, H., Bertness, K. A., Sanford, N. A., Blanchard, P. T., & Kabos, P. (2008). Calibrated broadband electrical characterization of nanowires. In *2008 Conference on Precision Electromagnetic Measurements Digest, Broomfield, CO*, pp. 684–685.
7. Nougaret, L., Dambrine, G., Lepilliet, S., Happy, H., Chimot, N., Derycke, V., & Bourgoïn, J.-P. (2010). Gigahertz characterization of a single carbon nanotube. *Applied Physics Letters*, *96*(4), 042109-1–042109-3.
8. Li, S., Yu, Z., Yen, S.-F., Tang, W. C., & Burke, P. J. (2004). Carbon nanotube transistor operation at 2.6 GHz. *Nano Letters*, *4*(4), 753–756.
9. Rosenblatt, S., Lin, H., Sazonova, V., Tiwari, S., & McEuen, P. L. (2005). Mixing at 50 GHz using a single-walled carbon nanotube transistor. *Applied Physics Letters*, *87*(15), 153111.
10. El Fellahi, A., Haddadi, K., Marzouk, J., Arscott, S., Boyaval, C., Lasri, T., & Dambrine, G. (2015). Integrated MEMS RF probe for SEM station—Pad size and parasitic capacitance reduction. *IEEE Microwave and Wireless Components Letters*, *25*(10), 693–695.
11. Marzouk, J., Arscott, S., El Fellahi, A., Haddadi, K., Lasri, T., Boyaval, C., & Dambrine, G. (2015). MEMS probes for on-wafer RF microwave characterization of future microelectronics: design, fabrication and characterization. *Journal of Micromechanics and Microengineering—IOPscience*, *25*(7).
12. El Fellahi, A., Haddadi, K., Marzouk, J., Arscott, S., Boyaval, C., Lasri, T., & Dambrine, G. (2015, September). Nanorobotic RF probe station for calibrated on-wafer measurements. In *45th European Microwave Conference, Paris, France*, pp. 1–4.
13. Reichelt, R. (2007). Scanning electron microscopy. In *Science of microscopy* (pp. 133–272). New-York: Springer.
14. https://www.smaract.com/SmarAct_Catalog_v16.pdf
15. National instruments NI. *LabVIEW control design user manual*.
16. Halvorsen, H.-P., Department of Electrical Engineering, Information Technology and Cybernetics. *Control and simulation in LabVIEW*.
17. Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *4th Alvey Vision Conference*, pp. 147–151.
18. Mikolajczyk, K., & Schmid, C. (2002). An affine invariant interest point detector. In A. Heyden et al. (Eds.), *ECCV 2002, LNCS 2350* (pp. 128–142). Berlin; Heidelberg: Springer.

Chapter 25

Development of Design and Training Application for Deep Convolutional Neural Networks and Support Vector Machines



Fusaomi Nagata, Kenta Tokuno, Akimasa Otsuka, Hiroaki Ochi, Takeshi Ikeda, Keigo Watanabe, and Maki K. Habib

Abbreviations

AlexNet	AlexNet is the name of a well-known convolutional neural network designed by Alex Krizhevsky, which is the champion of the competition called ImageNet Large Scale Visual Recognition Challenge held in 2012
Back Propagation (BP) algorithm	BP algorithm is a famous method used in artificial neural networks to calculate weights between a large number of neurons. The BP algorithm is a generalized delta rule for multi layered neural networks, in which chain rules are applied to iteratively calculating the weights based on gradients and errors in the network

F. Nagata (✉) · K. Tokuno · A. Otsuka · H. Ochi · T. Ikeda
Sanyo-Onoda City University, Sanyo-onoda, Japan
e-mail: nagata@rs.socu.ac.jp; otsuka_a@rs.socu.ac.jp; ochi@rs.socu.ac.jp; t-ikeda@rs.socu.ac.jp

K. Watanabe
Okayama University, Okayama, Japan
e-mail: watanabe@sys.okayama-u.ac.jp

M. K. Habib
The American University in Cairo, Cairo, Egypt
e-mail: maki@aucegypt.edu

Caffe	Caffe means convolutional architecture for fast feature embedding, which is one of famous deep learning frameworks developed at University of California, Berkeley. Caffe is an open source library written in C++
Convolutional Neural Network (CNN)	A convolutional neural network called CNN is a class of deep neural networks, which has been most commonly applied to image recognition. The CNNs are typical applications based on the concept of deep learning and are known as one of the most powerful structures for image recognition
MATLAB	MATLAB is a high performance computing environment provided by MathWorks. In particular, MATLAB enables us to conduct matrix manipulations, plotting of data, implementation of algorithms, design of user interfaces, and development of application software written in standard languages such as C++, C#, Python, and so on
ReLU	ReLU means a rectified linear unit function. In the structure of recent deep neural networks, the ReLU is one of effective activation functions superior to conventional sigmoid functions, which is defined with only positive part of the input argument
Sequential Minimal Optimization (SMO) Algorithm	SMO is an effective algorithm to solve quadratic programming (QP) problems. The QP is one of nonlinear programming methods. The SMO was proposed by John Platt in 1998 and has been used to train support vector machines
Support Vector Machine (SVM)	In machine learning, the support vector machine called SVM is one of supervised learning models associated with learning algorithms which can analyze training data used for classification and regression problems. A binary class SVM model is a hyper plane that can clearly separate two categorized points in space. The hyper plane is drawn so that the points can be divided into two domains by a clear margin as large as possible
TensorFlow	TensorFlow is an open source software library which can be used for the development of machine learning software such as neural networks. TensorFlow was developed by Google and released from 2015

25.1 Introduction

Recently, deep learning techniques are gathering attention from researchers and engineers all over the world due to the high performance superior to conventional shallow neural networks. In this decade, several software development environments for deep neural networks (DNN) such as Caffe [1] and TensorFlow [2] have been introduced to researchers and engineers. In those development environments, C++ or Python is well used for development. Deep convolutional neural networks (DCNN) are typical applications based on the concept of DNN and are known as one of the most powerful structures for image recognition. However, for example, it may be difficult for students and junior engineers to develop and implement a practical DCNN using programming languages such as C++ or Python and to utilize it for anomaly detection in actual production systems. Generally speaking, it seems that the availability of user-friendly software that facilitates such applications without using programming languages skills, such as C++ or Python, have not been sufficiently developed yet.

Hence, this paper presents the development of user-friendly application development environment based on MATLAB system [3, 4] that facilitates two applications using DCNNs and support vector machines (SVMs). An application of DCNN for anomaly detection is developed and trained using many images to inspect undesirable defects such as crack, burr, protrusion, chipping, spot, and fracture phenomena which appear in the manufacturing process of resin molded articles. Automation of visual inspection process has been demanded from many different kinds of industrial fields because it is not easy to reduce the increase of undesirable human error associated with the length of successive working hours.

Besides DCNN, SVMs are supervised learning models with associated learning algorithms that analyze data sets used for classification and regression analysis. Not only have a linear classification ability based on the idea of margin maximized hyperplanes, but also SVMs have promising characteristics to efficiently perform a nonlinear classification using what is called the kernel trick, by implicitly mapping input data into high-dimensional feature spaces [5].

In the fields of measurement systems, for example, Flores-Fuentes et al. proposed a combined application of power spectrum centroid and SVMs to improve the measurement capability in optical scanning systems [6]. The energy signal center is found in the power spectrum centroid, in which the SVM regression method is used as a digital rectified to increase measurement accuracy for optical scanning system. Then, a technical research of an opto-mechanical system for 3D measurement was reported in detail, in which a multivariate outlier analysis was implemented to detect and remove atypical values, in order to improve the accuracy of artificial intelligence regression algorithms [7]. Also, although the architecture is not deep structure, Rodriguez-Quinonez et al. surveyed the dominant laser scanner technologies, gave a detailed description of their 3D laser scanner, and adjusted their measurement error by a once trained feed forward back propagation (FFBP) neural network with a Widrow-Hoff weight/bias learning function [8]. Surface measurement systems (SMS) allow accurate measurements of surface geometry for three-dimensional

computational models creation. There are cases where contact avoidance is needed; these techniques are known as non-contact surface measurement techniques. To perform non-contact surface measurements, there are different operating modes and technologies, such as lasers, digital cameras, and integration of both. Each SMS is classified by its operation mode to get the data, so it can be divided into three basic groups: point-based techniques, line-based techniques, and area-based techniques. Real et al. provided useful topics about the different types of non-contact surface measurement techniques, theory, basic equations, system implementation, actual research topics, engineering applications, and future trends [9]. The description seems to be particularly beneficial for students, teachers, researchers, and engineers who want to implement some visual inspection system.

In this paper, two kinds of SVMs are, respectively, incorporated with the two trained DCNNs to classify sample images with high recognition rate into accept as OK or reject as NG categories, in which compressed features obtained from the DCNNs are used as the input for the SVMs. The two types of DCNNs used for generating feature vectors are our designed sssNet and the well-known AlexNet [10, 11]. The designed applications of the SVMs and their evaluation are introduced. The usability and operability of the proposed design and training application for DCNNs and SVMs are demonstrated and evaluated through design, training, and classification experiments.

25.2 Design and Training Application for DCNNs and SVMs

A large number of image files with different kinds of defect features and their paired labels for classification are needed to construct a reliable DCNN-based anomaly detection system with generalization ability. To deal with this serious need, a dialogue-based application named the similar image generator was first developed that can easily produce a lot of similar images with sequential number from an original image for training. For example, similar images with fracture defect can be generated as shown in Fig. 25.1 by rotating, translating, scaling an original image, changing the brightness, the resolution, or the file format such as JPG, BMP, PNG, and so on.

Then, a DCNN and SVM design application as shown in Fig. 25.2 was developed using App Designer provided by MATLAB. Deep Learning Toolbox (Neural Network Toolbox), Statistics and Machine Learning Toolbox, Parallel Computing Toolbox, Computer Vision System Toolbox, and Image Processing Toolbox were optionally installed for the development on MATLAB. Main DCNN design parameters on number of layers, filter size, pooling size, padding size, and width of stride can be easily given through the user-friendly dialogue. As an example, Fig. 25.3 shows a designed DCNN composed of three convolution layers. The first layer is positioned for input images with a resolution of $200 \times 200 \times 1$ given by a matrix

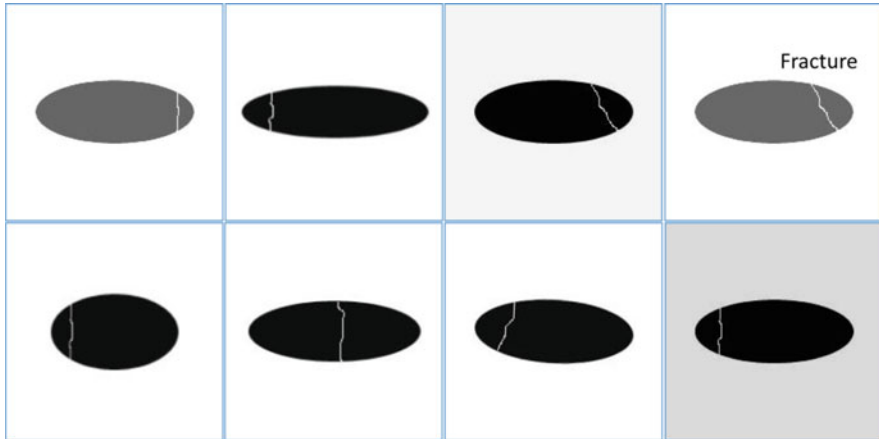


Fig. 25.1 Examples of generated images with a defect of fracture for training

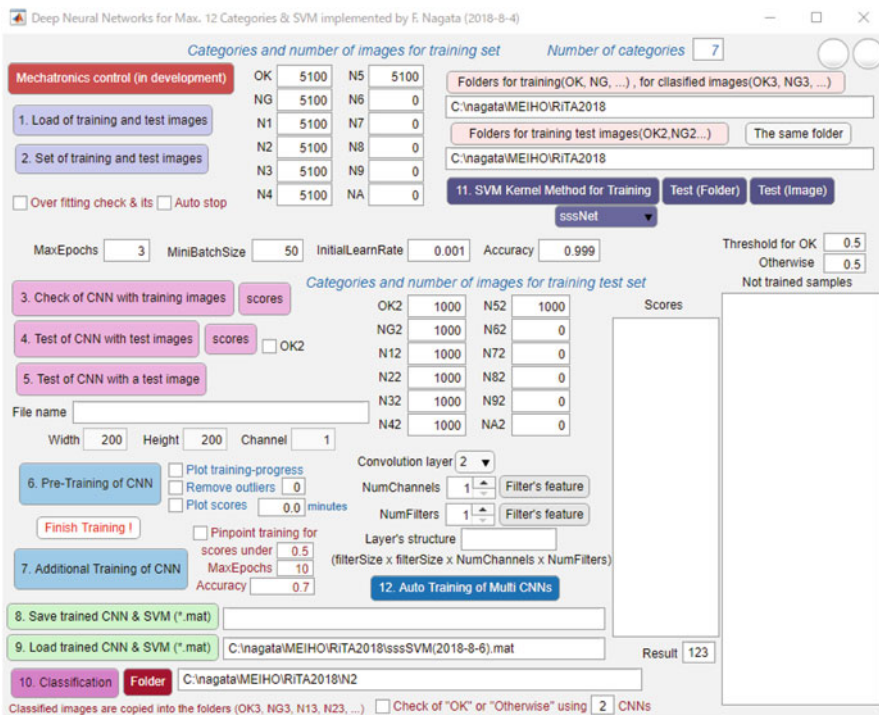


Fig. 25.2 The developed design and training application for DCNNs and SVMs

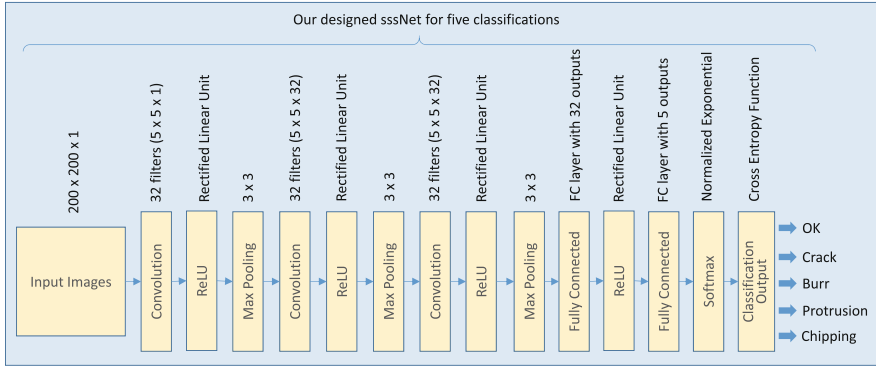


Fig. 25.3 An example of DCNN for five classifications designed by using the application shown in Fig. 25.2

with zerocenter normalization. The second, fifth, and eighth layers are convolution ones which severally have 32 filters. It is known that convolution layers perform the translation invariance and compositionality required for computer vision. In the convolution layers, the filters are applied to each image while sliding from the left top to the right bottom in the image based on the value of the stride. Note that each filter has channels as many as the number of feature maps in the previous layer. Activation functions called rectified linear unit (ReLU) are located at third, sixth, ninth, and twelfth layers. The ReLUs are given by

$$f(u) = \max(0, u) \tag{25.1}$$

$$f'(u) = \begin{cases} 1 & (u > 0) \\ 0 & (u \leq 0) \end{cases} \tag{25.2}$$

In the context of deep neural networks, ReLU have been actively used as one of the most effective activation functions for back propagation algorithms. The 4th, 7th, and 10th layers are max pooling ones to reduce the dimensions of feature maps for computational efficiency. The sizes of pooling, stride, and padding are given as [3 3], [2 2], and [0 0 0], respectively. If the n th image for training is given to the input layer, then the 14th softmax layer produces the probability $p_{ni} (i = 1, 2, \dots, 5)$ called the score for five categories. The probability $p_{ni} (i = 1, 2, \dots, 5)$ generated from the 14th softmax layer for five categories is calculated by

$$p_{ni} = \frac{e^{y_{ni}}}{\sum_{k=1}^5 e^{y_{nk}}} \tag{25.3}$$

where $y_n = [y_{n1} \ y_{n2} \ y_{n3} \ y_{n4} \ y_{n5}]^T$ is the output vector from the 13th fully connected layer corresponding to the n th input image. In this case, the loss function called cross entropy is calculated by

$$\bar{E} = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^5 t_{nk} \log(y_{nk}) \quad (25.4)$$

where $t_n = [t_{n1} \ t_{n2} \ t_{n3} \ t_{n4} \ t_{n5}]^T$ means the n th desired output vector for five categories, i.e., $[1 \ 0 \ 0 \ 0 \ 0]^T$, $[0 \ 1 \ 0 \ 0 \ 0]^T$, $[0 \ 0 \ 1 \ 0 \ 0]^T$, $[0 \ 0 \ 0 \ 1 \ 0]^T$, or $[0 \ 0 \ 0 \ 0 \ 1]^T$. N is the total number of image samples in the training set. The cross entropy is also used to tune the values in each filter in back propagation algorithm during iterative training process.

25.3 Review of Back Propagation Algorithm for Implementation

The authors of this chapter have implemented the back propagation algorithm into some systems [12–15]. The first system designed for a feedforward force controller learned the contact motion which was the relation between the contact force and the velocity at the tip of robot arm[12]. The second system named the effective stiffness estimator was developed for a desktop NC machine tool with a compliant motion capability. The estimator finally allowed the machine tool to generate a desired damping needed for a stable force control system without undesirable large overshoots and oscillations [13, 14]. Further, the third system was considered to deal with the problem concerning the learning performance to large scale teaching signals, so that a simple and adaptive learning technique for sigmoid functions could be proposed. The validity and control effectiveness of the learning technique were verified through simulation experiments using the dynamic model of PUMA560 manipulator with six degrees of freedoms. In this section, the important back propagation (BP) algorithm for training is reviewed for an easy implementation in software development using a simple three-layered neural network with two inputs and two outputs as shown in Fig. 25.4, in which a standard sigmoid function is applied as an activation function of each neuron. It is known that BP algorithms are also applied to the training of filters in CNNs. The sigmoid function and its derivative are generally given by

$$f(s) = \frac{1}{1 + e^{-s}} \quad (25.5)$$

$$f'(s) = f(s)\{1 - f(s)\} \quad (25.6)$$

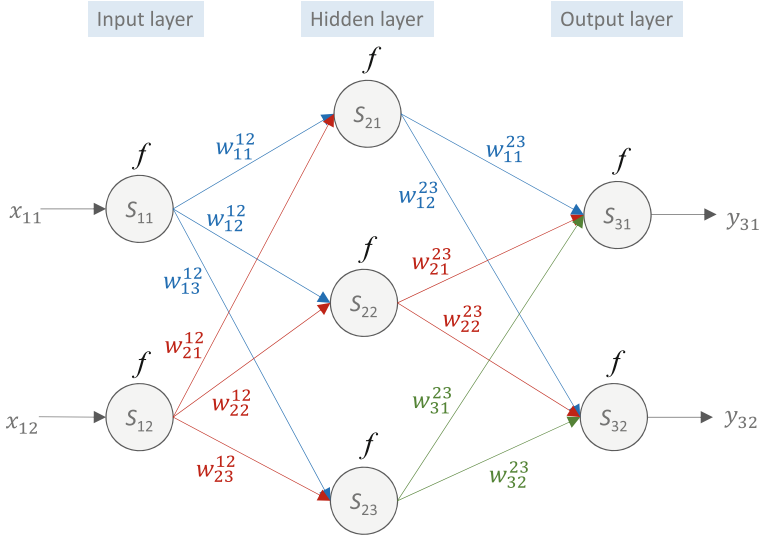


Fig. 25.4 Three-layered neural network to review the back propagation algorithm

where s is the state of a neuron. Weights between the last hidden layer and the output layer are updated through the calculations based on the generalized delta rule. The weights in this example are actually updated based on the rule as written by

$$e_{31} = d_1 - y_{31} \tag{25.7}$$

$$e_{32} = d_2 - y_{32} \tag{25.8}$$

$$w_{11}^{23} = w_{11}^{23} + \eta f(s_{21}) f(s_{31}) \{1 - f(s_{31})\} e_{31} \tag{25.9}$$

$$w_{12}^{23} = w_{12}^{23} + \eta f(s_{21}) f(s_{32}) \{1 - f(s_{32})\} e_{32} \tag{25.10}$$

$$w_{21}^{23} = w_{21}^{23} + \eta f(s_{22}) f(s_{31}) \{1 - f(s_{31})\} e_{31} \tag{25.11}$$

$$w_{22}^{23} = w_{22}^{23} + \eta f(s_{22}) f(s_{32}) \{1 - f(s_{32})\} e_{32} \tag{25.12}$$

$$w_{31}^{23} = w_{31}^{23} + \eta f(s_{23}) f(s_{31}) \{1 - f(s_{31})\} e_{31} \tag{25.13}$$

$$w_{32}^{23} = w_{32}^{23} + \eta f(s_{23}) f(s_{32}) \{1 - f(s_{32})\} e_{32} \tag{25.14}$$

where d_1 and d_2 are the components in the desired output vector to be trained. $y_3 = [y_{31} \ y_{32}]^T$ is an output vector from the network. $e_3 = [e_{31} \ e_{32}]^T$ is the error vector between the desired and the actual outputs. w_{ij}^{pq} is the weight between the i th unit in p th layer and the j th unit in q th layer. $x_1 = [x_{11} \ x_{12}]^T$ is the input vector to be directly $s_1 = [s_{11} \ s_{12}]^T$. η is the learning rate. Also, for example, the state s_{31} and s_{32} are linearly calculated by

$$s_{31} = w_{11}^{23}f(s_{21}) + w_{21}^{23}f(s_{22}) + w_{31}^{23}f(s_{23}) \quad (25.15)$$

$$s_{32} = w_{12}^{23}f(s_{21}) + w_{22}^{23}f(s_{22}) + w_{32}^{23}f(s_{23}) \quad (25.16)$$

Next, update process of weights between the hidden layer and the input layer is explained. The calculation of error e_{pi} for p th layer, i th unit in the hidden layer is a little bit more complex. For example, e_{21} , e_{22} , and e_{23} are obtained by

$$e_{21} = w_{11}^{23}f(s_{31})\{1 - f(s_{31})\}e_{31} + w_{12}^{23}f(s_{32})\{1 - f(s_{32})\}e_{32} \quad (25.17)$$

$$e_{22} = w_{21}^{23}f(s_{31})\{1 - f(s_{31})\}e_{31} + w_{22}^{23}f(s_{32})\{1 - f(s_{32})\}e_{32} \quad (25.18)$$

$$e_{23} = w_{31}^{23}f(s_{31})\{1 - f(s_{31})\}e_{31} + w_{32}^{23}f(s_{32})\{1 - f(s_{32})\}e_{32} \quad (25.19)$$

so that, the weight w_{11}^{12} , w_{12}^{12} , w_{13}^{12} , w_{21}^{12} , w_{22}^{12} , and w_{23}^{12} are calculated by

$$w_{11}^{12} = w_{11}^{12} + \eta f(s_{11})f(s_{21})\{1 - f(s_{21})\}e_{21} \quad (25.20)$$

$$w_{12}^{12} = w_{12}^{12} + \eta f(s_{11})f(s_{22})\{1 - f(s_{22})\}e_{22} \quad (25.21)$$

$$w_{13}^{12} = w_{13}^{12} + \eta f(s_{11})f(s_{23})\{1 - f(s_{23})\}e_{23} \quad (25.22)$$

$$w_{21}^{12} = w_{21}^{12} + \eta f(s_{12})f(s_{21})\{1 - f(s_{21})\}e_{21} \quad (25.23)$$

$$w_{22}^{12} = w_{22}^{12} + \eta f(s_{12})f(s_{22})\{1 - f(s_{22})\}e_{22} \quad (25.24)$$

$$w_{23}^{12} = w_{23}^{12} + \eta f(s_{12})f(s_{23})\{1 - f(s_{23})\}e_{23} \quad (25.25)$$

25.4 Design and Training Experiments of Designed DCNN

25.4.1 Test Trial of Design and Training of a DCNN for Binary Classification

Table 25.3 tabulates the main parameters of DCNN training in case of two categories, which are non-defective and defective articles named OK and NG, respectively. The category of NG includes the defects of burr, protrusion, and crack. The training was conducted by using a single PC with a Core i7 CPU and a GPU (NVIDIA GeForce GTX 1060). In this DCNN training, it first took about several minutes until the categorization accuracy reached to 0.95. The accuracy is the result of discriminant analysis obtained by dividing the number of correctly classified images by that of images in the entire data set. Then, the DCNN was additionally fine-trained to enhance the accuracy to 1. It also took several minutes to the completion. After the DCNN was trained, the classification result of the training set was checked based on the scores from the softmax layer. Figure 25.5 shows the scores of classification of OK and NG using all the images in training set (total number of images is 2040).

Table 25.3 Parameters designed for DCNN training

Filter size in convolution layers	$5 \times 5 \times 1$
Padding of convolution layers	[2 2 2 2]
Stride of convolution layers	[1 1]
Pooling size	[3 3]
Padding of max pooling layers	[0 0 0 0]
Stride of max pooling layers	[2 2]
Max epochs	30
Mini batch size	200
Learning rate	0.002 to 0.0001
Desired categorization accuracy	0.999
Number of OK images	1020
Number of NG images	1020

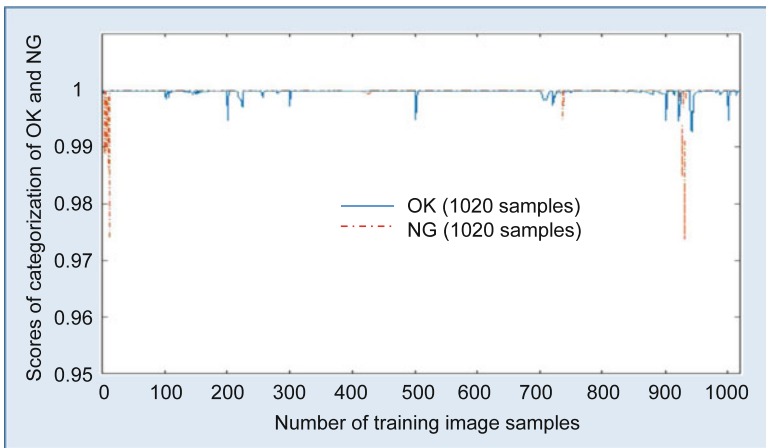


Fig. 25.5 Scores of classification of OK and NG using the all images in the training set (the total number of images = 1020 + 1020)

As can be seen, it is observed that all the 2040 images in the training set can be well discriminated with each score more than 0.97. Next, the generalization of the trained DCNN was simply evaluated using the test images with a feature of burr, protrusion, or crack as shown in Fig. 25.6 which were not included in the training set. Figure 25.7 shows the classification scores evaluated using the ten test images including a feature of defect, in which it is observed that “image2.jpg” and “image9.jpg” are not well categorized. To cope with this problem, we considered to pinpointedly improve the recognition ability to these two types of defects. Figure 25.8 shows the additional 10 training images a little bit deformed from “image2.jpg” in Fig. 25.6.

To enhance the classification ability further to the images shown in Figs. 25.6 and 25.8, the pretrained DCNN was additionally retrained using the reorganized training set consisting of original 2040 images, additional 20 OK ones, 10 NG ones in Fig. 25.6, and 10 NG ones in Fig. 25.8. After the additional training, the training



Fig. 25.6 Test images with a feature of NG which were not included in the training images

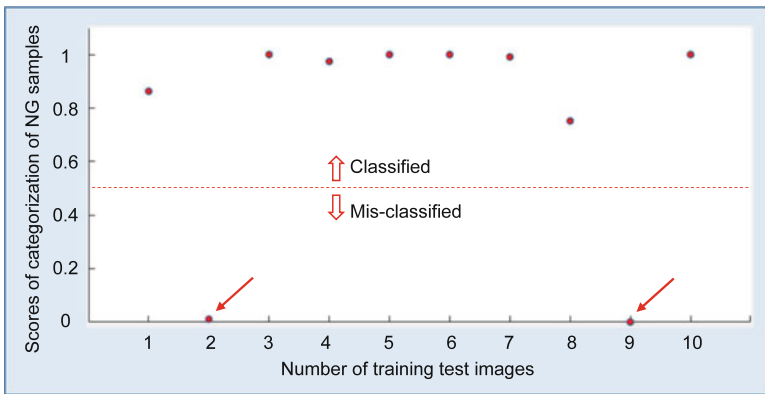


Fig. 25.7 Scores of classification of NG test images shown in Fig. 25.6



Fig. 25.8 Additional 10 training images with a protrusion which is a little bit deformed from “image2.jpg” shown in Fig. 25.6

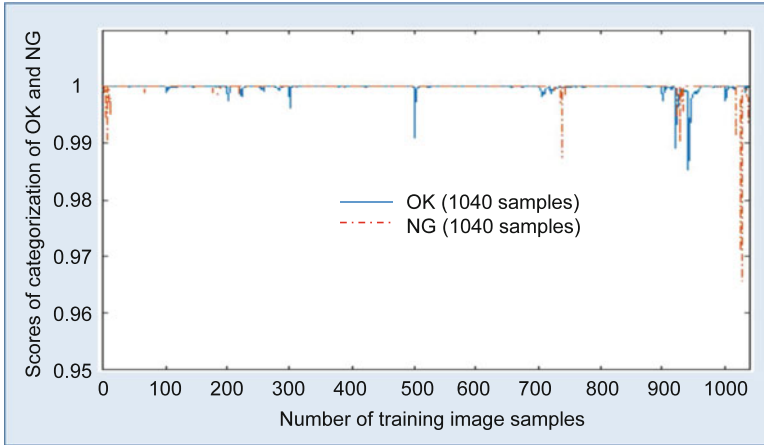


Fig. 25.9 Check of scores of categorization OK and NG using the images in training set (total number of images = 1040 + 1040), in which images shown in Figs. 25.6 and 25.8 are included

situation was checked based on the scores of categorization OK and NG using the images in reorganized training sets (total number of images = 1040 + 1040) including the images in Figs. 25.6 and 25.8. Figure 25.9 shows the result. It is observed that the recognition ability to additional images can be improved efficiently and pinpointedly. The additional training function introduced in this section is effective to reconstruct an updated DCNN when miscategorized images are found in training test process.

25.4.2 Test Trial of Design and Training for Five Categories

The DCNN designed for the binary classification of resin molded articles is extended and applied to classifying images into typical five defective categories as shown in Fig. 25.3, in which the category of NG is subdivided into typical defects seen in resin molding process such as crack, burr, protrusion, and chipping. An epoch means a full pass through the entire training data set, i.e., $5100 \times 5 = 25,500$ images are used for training process. First, a pretraining using randomly initialized weights is conducted through the period from the first epoch to sixth one, where the desired categorization accuracy is set to 0.999. Then, a fine training using the pretrained weights is successively conducted through the period from the seventh epoch to tenth one, where the desired categorization accuracy is increased to 0.9999. After the fine training, it is confirmed from the experiments that the designed DCNN with 15 layers shown in Fig. 25.3 can be well trained to classify resin molded articles into five categories, through the training process using 25,500 gray scale image samples with the resolution of 200×200 .

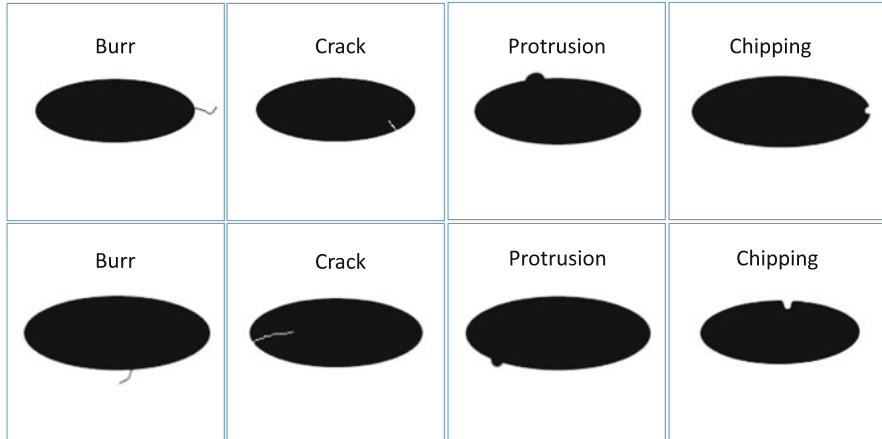


Fig. 25.10 Examples of four kinds of defects which are seen in production process of resin molded articles

Finally, after further adding 300 images with different features into each training set, the trained DCNN is additionally trained, i.e., by using the $5400 \times 5 = 27,000$ images. Then, to simply check the generalization ability of the trained DCNN, a training test set consisting of 100 images \times 5 categories are prepared. Figure 25.10 shows some of the images in the training test set. After the testing, it is confirmed that the categorization accuracy to the test images is $492/500 = 98\%$, so that it is concluded that the obtained DCNN can perform satisfactory generalization.

25.5 Support Vector Machines Based on Trained DCNNs

In the previous section, two types of DCNNs for two or five classification are designed, trained, and evaluated using the proposed DCNN design application. In this section, another approach using two types of support vector machines (SVMs) is introduced. It is expected that the DCNN designed in the previous section may be able to give more characterized feature vectors to the SVMs. Actually, the most important function which is required to a defect inspection system is to remove defective products from all products. It is not allowed that any defective product is mixed into lots of non-defective products. To cope with this serious need, two types of SVMs shown at the lower parts in Figs. 25.11 and 25.12 are tried to be designed and trained using the proposed application shown in Fig. 25.2. It is expected that the trained SVMs will be able to classify input images into OK or NG category including a small defect such as crack, burr, protrusion, chipping, spot, and fracture.

As for the first SVM, our designed DCNN named sssNet is used to extract the feature vector $\mathbf{x} = [x_1, x_2, x_3, \dots, x_{32}]^T$ from each inputted image. Figure 25.11 illustrates the designed SVM for binary classification whose input is the feature vector

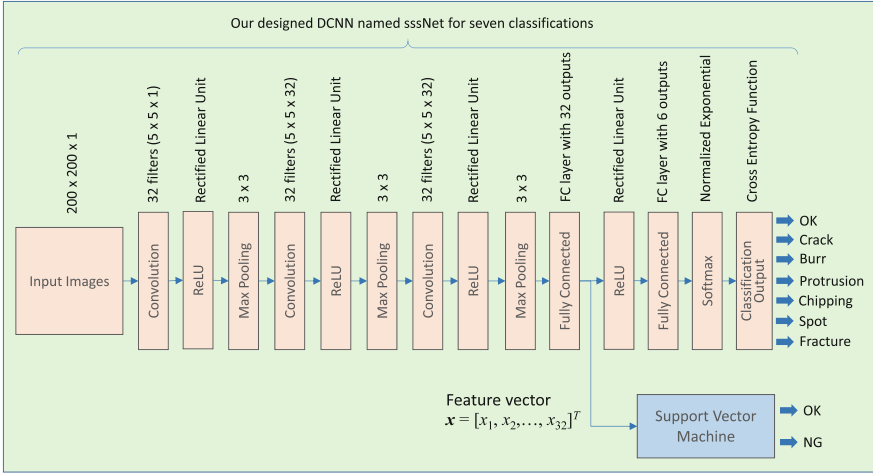


Fig. 25.11 The proposed SVM for binary classification to which feature vectors generated from our designed DCNN named sssNet are given

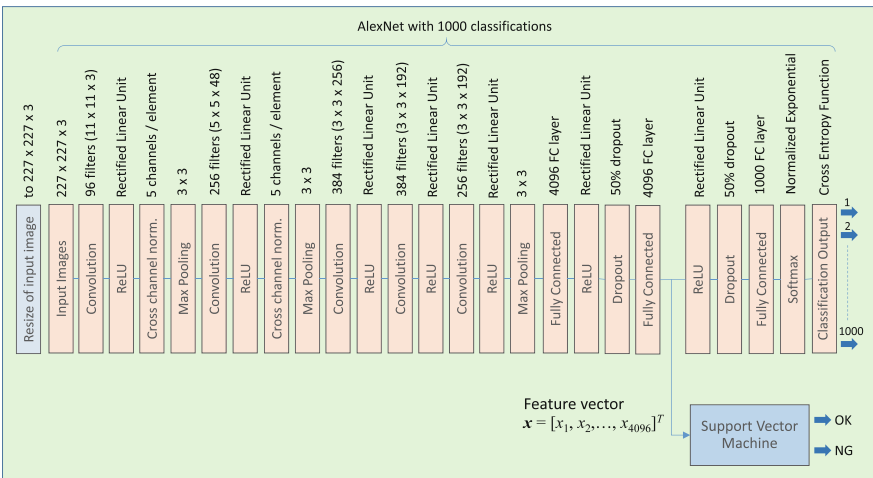


Fig. 25.12 The proposed SVM for binary classification to which feature vectors generated from AlexNet are given

generated from the 1st fully connected layer (11th layer) in the sssNet. Gaussian kernel function is used for one class unsupervised training of the SVM, in which 5100 OK images used in the pretraining in the Sect. 25.4.2 are reused. Sequential minimal optimization (SMO) algorithm [16] is applied to solve the quadratic programming (QP) of SVM. It took about several minutes for training the SVM. After training the SVM, a classification experiment was conducted to check the generalization ability to unlearned NG images. Figure 25.13 shows the classification results using

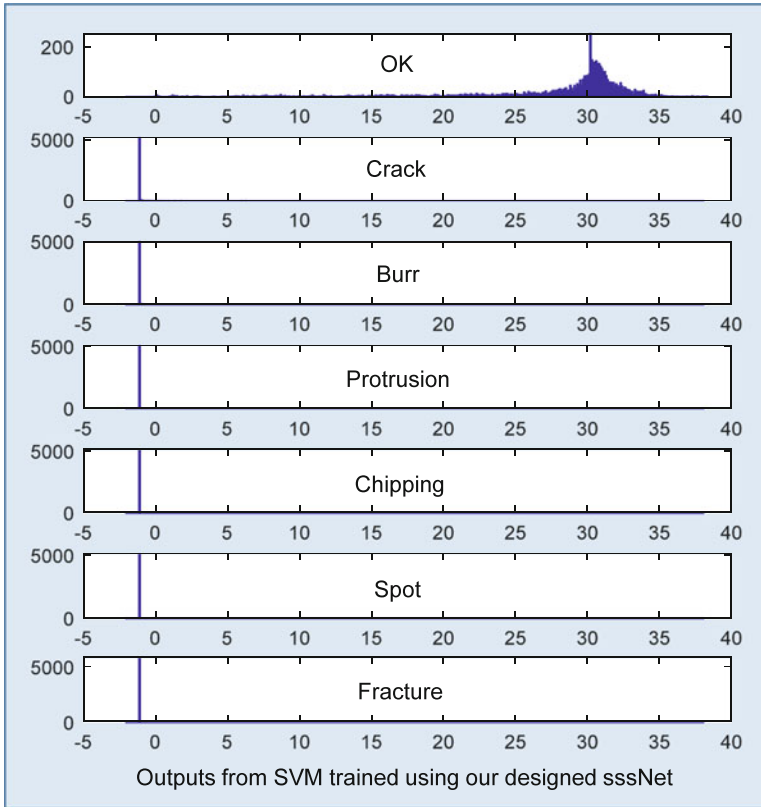


Fig. 25.13 Classification results using the SVM shown in Fig. 25.11, in which horizontal and vertical axes denote the output from the SVM trained with our designed sssNet and the number of image samples, respectively

the SVM shown in Fig. 25.11. The horizontal and vertical axes denote the output values from the SVM trained with our designed sssNet and the number of image samples, respectively. It is observed from Fig. 25.13 that the SVM can discriminate NG images from OK ones.

As for the second SVM, well-known DCNN called AlexNet is used to extract the feature vector $\mathbf{x} = [x_1, x_2, x_3, \dots, x_{4096}]^T$ from each inputted image. The AlexNet trained using one million images can classify test images into 1000 object categories such as a keyboard, mug, pencil, many kinds of animals, and so on. It is known that the AlexNet learned abundant feature representations of images covering a wide range of objects. If the trained AlexNet receives an image with the resolution of $227 \times 227 \times 3$, then the label of an object featuring in the image and the probability, i.e., score of the categorized object, are produced. Figure 25.12 illustrates another binary class SVM whose input is the feature vector generated from the 2nd fully connected layer (20th layer) in the AlexNet. Similarly, 5100 OK images

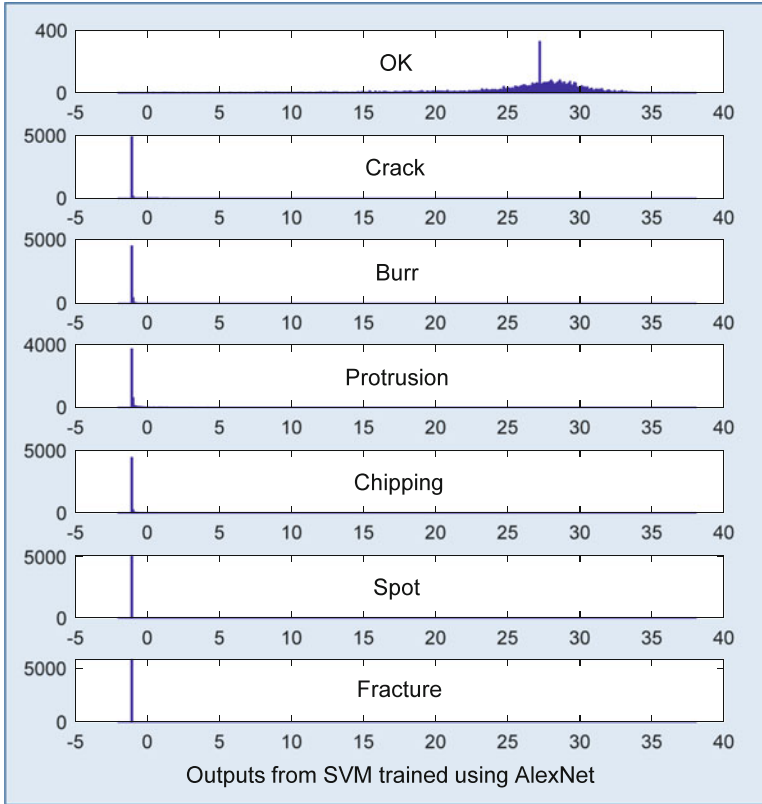


Fig. 25.14 Classification results using the SVM shown in Fig. 25.12, in which horizontal and vertical axes denotes the output from the SVM trained using AlexNet and the number of image samples, respectively

used in the pretraining in the Sect. 25.4.2 were reused for one class unsupervised training of the SVM. It also took about several minutes for training. After training the SVM, a classification experiment was conducted to check the generalization ability to unlearned NG images. Figure 25.14 shows the classification results using the SVM shown in Fig. 25.12. It is observed from Fig. 25.14 that the SVM with AlexNet can also discriminate NG images from OK ones with the almost same reliability as the SVM with sssNet. Actually, lengths of feature vectors generated from sssNet and AlexNet are quite different as 32 and 4096; however, almost the same discrimination ability can be obtained. In the case of the target features as shown in Figs. 25.1, 25.6, 25.8, and 25.10, the feature vector with 4096 components given to SVM seems to be somewhat redundant.

25.6 Conclusions

In this decade, deep learning, in particular, DCNN has been eagerly focused on by researchers and engineers in order to apply to various kinds of inspection systems. However, it seems that user-friendly design and training tools without using programming languages such as C++ and Python have not been well provided yet. In this chapter, a design and training application for DCNNs with multiple classification and SVMs with binary classification is presented. As the first application test trial, a DCNN is designed using the application to detect defects such as crack, burr, protrusion, chipping, and fracture phenomena seen in the manufacturing process of resin molded articles. A similar image generator is also proposed to efficiently generate a large number of images transformed from original ones by rotating, scaling, and changing brightness, etc. After the designed DCNN is pretrained using those images, classification experiments are conducted using test images in order to simply check the generalization. Based on the results, an additional fine training method is applied and evaluated to cope with mis-classified images, so that the classification ability can be efficiently and pinpointedly improved to a desired level of categorization accuracy. Generally, the objective of training in machine learning is to enhance the ability of generalization to unlearned environments. The additional training introduced in this chapter may proceed to the opposite direction of the objective of training or cause different kinds of problems. However, in daily production process, it will be effective for the construction of a practical visual inspection system. Here, the practicality means that the additionally trained DCNN will never miss the defects which have been misrecognized once. As the second application test trial, two kinds of SVMs with trained DCNNs, i.e., our designed sssNet and well-known AlexNet, for binary classification are designed, trained, and evaluated to discriminate NG sample images from OK ones, so that it is confirmed that the SVM with our designed sssNet can perform almost the same recognition ability as that with AlexNet in spite of the much shorter feature vector.

Finally, the authors apologize that unfortunately this chapter cannot show real photos including defective plastic parts due to the obligation of confidentiality with a joint research and development company.

References

1. Cengil, E., Cnar, A., & Ozbay, E. (2017). Image classification with caffe deep learning framework. In *Proceedings of 2017 International Conference on Computer Science and Engineering (UBMK)*, Antalya (pp. 440–444).
2. Yuan, L., Qu, Z., Zhao, Y., Zhang, H., & Nian, Q. (2017). A convolutional neural network based on tensorflow for face recognition. In *Proceedings of 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing (pp. 525–529).

3. Nagata, F., Tokuno, K., Tamano, H., Nakamura, H., Tamura, M., Kato, K., et al. (2018). Basic application of deep convolutional neural network to visual inspection. In *Proceedings of International Conference on Industrial Application Engineering (ICIAE2018)*, Okinawa (pp. 4–8).
4. Nagata, F., Tokuno, K., Otsuka, A., Ikeda, T., Ochi, H., Tamano, H., et al. (2018) Design tool of deep convolutional neural network for visual inspection. In *Proceedings of The Third International Conference on Data Mining and Big Data (DMBD2018), Springer-Nature LNCS Conference Proceedings 10943*, Shanghai (pp. 604–613).
5. Cristianini, N., & Shawe-Taylor, J. (2000) *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.
6. Flores-Fuentes, W., Rivas-Lopez, M., Sergiyenko, O., Gonzalez-Navarro, F. F., Rivera-Castillo, J., Hernandez-Balbuena, D., et al. (2014). Combined application of power spectrum centroid and support vector machines for measurement improvement in optical scanning systems. *Signal Processing*, 98, 37–51.
7. Flores-Fuentes, W., Sergiyenko, O., Gonzalez-Navarro, F. F., Rivas-Lopez, M., Rodriguez-Quinonez, J. C., Hernandez-Balbuena, D., et al. (2016). Multivariate outlier mining and regression feedback for 3D measurement improvement in opto-mechanical system. *Optical and Quantum Electronics*, 48(8), 403.
8. Rodriguez-Quinonez, J. C., Sergiyenko, O., Hernandez-Balbuena, D., Rivas-Lopez, M., Flores-Fuentes, W., & Basaca-Preciado, L. C. (2014). Improve 3D laser scanner measurements accuracy using a FFBP neural network with Widrow-Hoff weight/bias learning function. *Opto-Electronics Review*, 22(4), 224–235.
9. Real, O. R., Castro-Toscano, M. J., Rodriguez-Quinonez, J. C., Sergiyenko, O., Hernandez-Balbuena, D., Rivas-Lopez, M., et al. (2019). Surface measurement techniques in machine vision: Operation, applications, and trends. In *Optoelectronics in machine vision-based theories and applications* (pp. 79–104). Hershey: IGI Global.
10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, NV (pp. 1097–1105).
11. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
12. Nagata, F., & Watanabe, K. (2002). Learning of contact motion using a neural network and its application for force control. In *Proceedings of the 4th Asian Control Conference (ASCC2002)* (pp. 420–424).
13. Nagata, F., Mizobuchi, T., Tani, S., Watanabe, K., Hase, T., & Haga, Z. (2009). Impedance model force control using neural networks-based effective stiffness estimator for a desktop NC machine tool. *Journal of Manufacturing Systems*, 28(2/3), 78–87.
14. Nagata, F., Mizobuchi, T., Hase, T., Haga, Z., Watanabe, K., & Habib, M. K. (2010). CAD/CAM-based force controller using a neural network-based effective stiffness estimator. *Artificial Life and Robotics*, 15(1), 101–105.
15. Nagata, F., & Watanabe, K. (2011). Adaptive learning with large variability of teaching signals for neural networks and its application to motion control of an industrial robot. *International Journal of Automation and Computing*, 8(1), 54–61.
16. Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14 (pp. 1–24).

Chapter 26

Computer Vision-Based Monitoring of Ship Navigation for Bridge Collision Risk Assessment



Xiao-Wei Ye, Tao Jin, and Peng-Peng Ang

Acronyms

IPT Image process technology
DL Deep learning
SHM Structural health monitoring

26.1 Introduction

Since the 1990s, the tonnage of the ship is getting bigger, more and more ships are in the same channel. So the possibility of the bridge pier being struck by ships increases. Developed water transport brings a serious problem, namely the bridge collision problem, while promoting economic development. At present, due to the large-scale construction of bridges across rivers, the scale of navigation vessels, coupled with changes in bridge environment (such as flow rate, wind speed, curve, erosion, and siltation), have caused bridge–ship crashes, which often bring huge loss of life and property. Research focusing on the protection of bridges has been conducted by scholars or engineers all around the world.

Bridge reinforcement is one way to improve the bridge viability, especially for aged bridges. Vu and Stewart [1] developed an empirical formula of the ampere density for the corrosion of chloridion and carried out research on the whole life cycle reliability of reinforced concrete bridges under the corrosion of chloridion. Choe et al. [2] established a probability degradation model of load-carrying capacity

X.-W. Ye (✉) · T. Jin · P.-P. Ang
Department of Civil Engineering, Zhejiang University, Hangzhou, China
e-mail: cexwye@zju.edu.cn; cetaojin@zju.edu.cn; cepengpengang@zju.edu.cn

for reinforced concrete bridges under chloridion corrosion and studied the seismic vulnerability of the model. Li et al. [3] implemented experimental study on the influence of corrosion on reinforced concrete piers. Simon et al. [4] investigated the influence of chloridion corrosion on the seismic response and vulnerability of reinforced concrete bridges. Alipour et al. [5] carried out experimental study on the influence of chloridion corrosion on the anti-seismic property of reinforced concrete continuous beam bridges with different spans, pier heights, and diameters.

At present, the measures for the bridge to deal with ship collision are mainly passive anti-collision, that is, when designing the bridge structure, the designer should reasonably consider the ability of the bridge to resist the impact of the ship or adopt passive anti-collision measures to reduce the direct effect of ship collision on the bridge. Passive anti-collision measures are expensive. Many experts are carrying out the anti-collision design of the bridge pier, and the purpose of the anti-collision design of the bridge pier is to prevent the bridge pier from being damaged due to the excessive impact force of the ship and to ensure the safety of the bridge structure. By adopting different types of anti-collision facilities, it is possible to effectively reduce or prevent the force of the ship on the bridge pier, thereby ultimately protecting the bridge. In recent years, researchers in many countries are studying the bridge pier anti-collision maintenance devices, such as shield system, support pile system, floating tether system, artificial island and reef protection, and floating protection system. But these methods can only reduce the damage of the pier struck by ships, it does not fundamentally solve the problem that the bridge pier being struck by ships. Through research, human error is the primary cause of ship collision accidents, including operational errors, lack of necessary navigation information (non-navigable holes), lack of skills, and controller misconduct.

Simulation or test research on ship-bridge collision problem was widely conducted by many groups. Zhu et al. [6] proposed an innovative foam-filled lattice composite bumper system with fiber-reinforced polymer skins and a foam-web core as protective structures for bridge piers against ship collision. Guo et al. [7] researched the optimal sensor placement for damage detection of bridges subject to ship collision, an optimal sensor placement method targeting post-collision damage detection of bridges was proposed. Fang et al. [8] proposed an innovative large-scale composite bumper system for bridge piers against ship collision. Liu and Gu [9] simulated whole ship-bridge collision process by nonlinear dynamic finite element method, a scenario of a 40,000 DWT oil tanker colliding with a bridge across the Yangtze River was designed for simulation. Minorsky [10] investigated the relationship between the deformation of steel structures and energy transition by multi ship-bridge collision tests, based on a collision accident of a nuclear power ship against a bridge. Meir-Dornberg [11] conducted scale ratio collision test to obtain the impact force, impact energy, and deformation of ship prow. Sha and Hao [12] built a detailed finite element model of a barge and simulated the damage feature during the process of collision with a single pier and discussed the influence of ship velocity, mass, pier size, and impact location. Fan and Yuan [13] considered the pile soil interaction in the simulation and also investigated the influence of material and initial stress. Wan et al. [14] conducted the numerical simulation and quasi-static

compression test of simplified bow model to study the static stiffness characteristic of the ship bow for comparison. Jiang and Chorzepa [15] assessed energy absorption capacity of the floated collision-prevention device made of fiber-reinforced plastic on several bridges. Simulations or prevention measures or energy was investigated by more groups worldwide [16–21].

Anti-collision piers or similar passive measures might protect the bridge at a very high cost for manpower or money, the best way to protect the bridge is to reduce the collision probability. With the development of computer vision technology and image acquisition equipment, vision-based techniques have been developed and applied in practical application [22–28]. This technology is increasingly popular among scholars and engineers due to the advantages including long range, non-contact, high accuracy, time saving, low cost, multi-function, etc. It is much easier for computer vision technology to be deeply integrated with other techniques like signal processing, automation, and artificial intelligence. Also, it is easier to be carried out on other platforms like submersibles, vehicles, airplanes, or satellites. Computer vision technology will take pictures or videos of the desired area by cameras and apply image process technology (IPT) or deep learning (DL) methods to track or identify the target for purposed parameters like displacement, location, size, etc. Thanks to the possible large visual field, cost-effective realization of tracking purpose for large areas is financially feasible. Vision-based techniques have been applied to a variety of structural health monitoring (SHM) tasks for bridges including deflection measurement [29], bridge line shape measurement [30], evaluation of carrying capacity [31], calibration of finite element model [32], modal analysis [33], damage identification [34], cable tension monitoring [26], and assistance for weigh-in-motion system [35].

The detection methods of ships with the help of IPTs have been explored by several groups. Jiao et al. [36] developed a densely connected multiscale neural network for ship detection, feature maps were densely connected, and reduction of the weight for easy examples was implemented. Liu et al. [37] established an inshore ship detection method by shape and context information; an active contour model was developed for segmenting of water and land target. Li et al. [38] proposed an inshore ship detection method by ship head and body information, ship head features were obtained by transformed domain of polar coordinates, and ship boundary was detected by the saliency of directional gradient. Liu et al. [39] proposed a ship detection framework by using convolutional neural networks, aiming at the prediction of bounding box for ship with orientation angle information. Liu et al. [40] developed a two-stage detection approach for automatic ship target detection; mean-shift smoothing algorithm and hierarchical ship target detection were adopted in the two stages, respectively. Lin et al. [41] inserted a task partitioning model in a fully convolutional network to deal with the problem of inshore ship detection; also, a ship detection framework was established to improve the robustness of detection. The establishment of an active anti-collision system, proactively reminding the ship of illegal operation, real-time assessment of ship collision risk, and active warning of ship collision hazard can avoid the occurrence of ship collision accidents or reduce the severity of ship collision accidents to the greatest extent.

This chapter presents a computer vision-based ship–bridge anti-collision system for an ancient arch bridge across the Beijing–Hangzhou Grand Canal in Hangzhou, China. The structural condition of the arch bridge is analyzed, and the passing ships are investigated and classified into three levels. IPTs are adopted for tracking of incoming ships to obtain the target parameters including velocity, direction, etc. The ship–bridge collision risk is assessed by a comprehensive multifactor method. Pretest is conducted on a river before the implementation on the ancient bridge where construction nearby is strictly controlled for the protection of the bridge and the scenic spot around.

26.2 Engineering Background

26.2.1 Bridge Introduction

The Gongchen Bridge was built in the year of 1631 in Ming Dynasty, and it is a mark of the destination for the Beijing–Hangzhou Grand Canal, as shown in Fig. 26.1. It is an arch pedestrian bridge connecting the east and west shores of the Grand Canal, with a total length of 92 m, covering three spans and a height of 16 m. The bearing structure is the arch ring, and soil was filled inside the bridge. The main span is 15.8 m, and other two spans are 11.9 m. The width is 5.9 m in the mid-span and 12.8 m at the two ends. The main structure is made of boulder strips which are rare nowadays, leading to the difficulty to repair or replace it, and the foundation was made of spiling. Due to the precious historical value and social value, the bridge was granted as a provincial cultural relic for better protection in 2005 and was promoted to be an important heritage site under state protection in 2013. This place near the bridge was set to be a scenic spot attracting thousands of tourists every year.



Fig. 26.1 The Gongchen Bridge. (a) Side view. (b) Front view

26.2.2 Navigation Condition

Ever since ancient times, the Grand Canal served as a significant path for the transportation of passengers and cargo. Although the status is not comparable to the past nowadays, it still serves as an important local passageway for commercial cargo transportation or visiting purpose. The channel where the Gongchen Bridge locates is a V level channel which allows the ships under 300 tons to go. The ships travel under the bridge could roughly be divided into three classes: the ship for cargo transportation is around 300 tons, the sightseeing boat for visiting traveling is around 100 tons, and the other small ships including maritime ships or privately owned fish boats. The cargo ships could be as long as 45 m and as wide as 11 m, and in practical loading, they might be heavier. The small ships will not threat the safety of the bridge, so it is beyond discussion. The typical types of the two large ships are shown in Fig. 26.2.

26.2.3 Collision Incidents Analysis

Due to the famous status for sight visiting and the important role for local transportation, the traffic on the channel for either traveling or cargo transportation is very heavy. However, the main span is only 15.8 m, and the arch ring is quite slender compared to the ship or the size of the bridge. The cargo ship drivers will have to be quite cautious to drive, and they have to stand in a line to go through the bridge. Compared to the cargo ships, the precious bridge is small and vulnerable, as shown in Fig. 26.3. As is shown in Fig. 26.3(b), when the cargo ship passes through the main span, the distance between the arch ring and the ship is so close that careless driving or irregular current could lead to collision incidents.



Fig. 26.2 Typical ships in the channel. (a) Sightseeing boat. (b) Cargo ship



Fig. 26.3 Heavy traffic under the bridge. (a) Fleet of cargo ships. (b) Passing circumstances

Fig. 26.4 Damage of arch ring



Plenty of collision accidents have happened in recent years, among them, there are several relatively serious incidents that show how the situation is like. In August 1996, the arch ring of the main span was hit twice that created cracks in the stone ring. In October 1998, the south east corner of the main span was hit so heavy that a stone strip with a length of 3 m long and 0.65 m width weighting 1.4 tons was broken into three pieces. In June 2007, four anti-collision piers were constructed to protect the bridge. However, on September 26, 2005, and November 23, 2005, the same place in the northeast corner was hit twice, leading to the wrecking of a stone strip. In 2008, a cargo ship hit the southeast anti-collision pier and slanted it. On August 31, 2008, a cargo ship hit the southeast anti-collision pier and nearly wrecked it. On January 6, 2016, the northeast corner was hit, and a stone strip was broken into two pieces, leading to bridge repair construction. Figure 26.4 illustrates the arch ring that has been hit plenty of times by passing ships. As shown in Fig. 26.4, there are wide cracks that reduce the structural strength, also the plenty of abrasion marks stand for the many small collision incidents that frequently happen. The anti-collision piers

did help to protect the bridge, but they will have to be repaired frequently to maintain protection capacity. Construction under water is more expensive and less reliable, and it might disturb the navigation.

26.2.4 Significance of the System

Due to the significant role for historical value, social value, and tourism value, it is necessary to build an anti-collision system to protect the bridge. Anti-collision piers have been constructed as a passive defense measure to protect the bridge, but it is not able to reduce the impacting incidents. The piers have to be repaired or even rebuilt as have already happened each time after the collision, leading to the cost of lots of money and manpower as well as the blocking of the channel for hours or days. Computer vision-based system is an initiative measure to alter the passive defending condition to active defending condition by monitoring the incoming ships and alerting actively. On the one hand, by the reduction of collision incidents, the bridge will be protected better. On the other hand, by reducing the incidents caused by the carelessness of the shipowners, the shipowners could potentially save the punishment fine. Also, the experience of development and implementation of the computer vision-based anti-collision system could be a reference for other cases in anti-collision action for bridges.

26.3 Ship–Bridge Anti-Collision System

The computer vision-based ship–bridge anti-collision system mainly consists of a monitoring and tracking system, a risk assessment and early warning system, and a post recording system, as shown in Fig. 26.5. The tracking system will track the incoming ships to obtain the ship type, velocity, and direction for risk evaluation. The risk assessment and early warning system will evaluate the risk levels and warn accordingly. The post recording and evaluation systems are applied to record the possible collision and evaluate the intensity for post measures. By the combination of these three systems, an active anti-collision system is established to reduce the probability of collision. The workflow of the computer vision-based ship–bridge anti-collision system is shown in Fig. 26.5. The design goals of the bridge–ship anti-collision monitoring and early warning system include (1) 24-h continuous monitoring of sailing ships within a certain range upstream of the bridge, including navigable and non-navigable areas; (2) identify the size, position, speed, direction, track, and quantity of the ship within the monitoring range and carry out the bridge–ship anti-collision warning according to the identified result; (3) after the occurrence of the early warning event, it is possible to communicate and alarm the ship by means of sound, light, communication, etc.; (4) the user interface is simple, easy to use, and user-friendly; (5) after accumulating a certain amount of data, ship

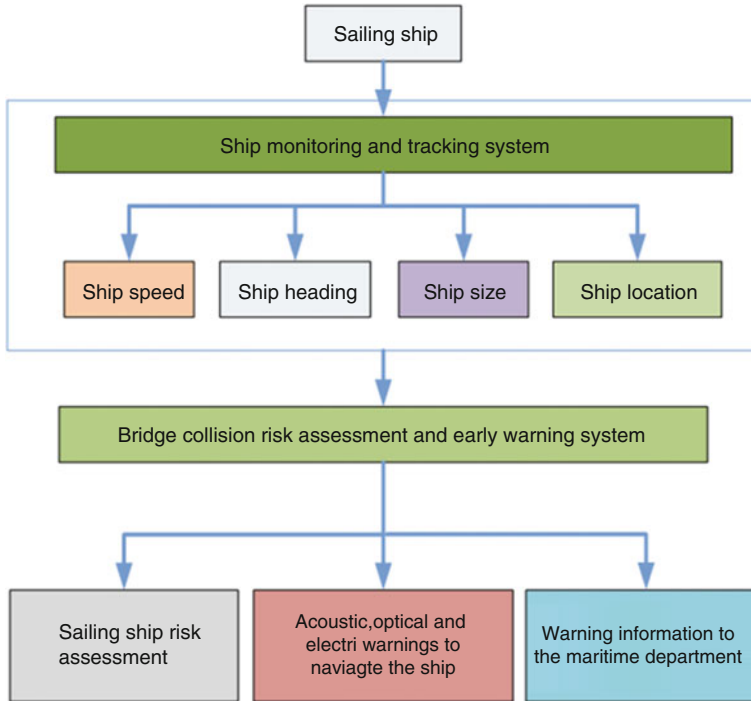


Fig. 26.5 Workflow of the system

transient prediction warning and active warning shall be carried out; (6) provide warning information to the maritime department and cooperate with the maritime department to deal with illegal navigation ships.

26.3.1 Monitoring and Tracking System

The theoretical core of the computer vision-based monitoring of ship navigation for bridge collision risk assessment method is the ship tracking method, including image pre-processing and target tracking method. The image pre-processing method mainly comprises image graying, image enhancement, image binarization, and image filtering steps. The general steps are as follows: implement graying to the input image to facilitate subsequent image processing; enhance image contrast by gray scale stretching; binarize the image to segment the background and objects; and apply median filter to eliminate image interference and noise.

The target tracking method consists of the frame difference method and the background learning algorithm. The basic principle of the frame difference method is that after the pre-process of the images, the gray values of the corresponding pixels

of two images at the adjacent time step are obtained. During the continuous tracking of moving targets, the dynamic change of gray values of the continuous image of moving targets is obtained by the frame difference method. The tracking of moving targets could then be realized by the change of gray values.

First, the sequent images f_1, f_2, \dots, f_t are collected for a continuous time period for background extraction. Assume that $f_k(x, y), f_{k+1}(x, y)$, and $f_{k+2}(x, y)$ are adjacent three frames of sequent images ($1 \leq k + 2 \leq t$) and that (x, y) are the coordinates of the corresponding frame image, as shown in Fig. 26.6. The k -th frame is adopted as a background image. Subtracting the k -th frame with the $(k + 1)$ -th frame, and we can get the difference image of adjacent images and mark M_1 for the center of the image, shown in Fig. 26.7(a). Subtracting the k -th frame with the $(k + 2)$ -th frame, we can get the difference image of adjacent images and mark M_2 for the center of the image, as shown in Fig. 26.7(b). Thus, continuous difference image is continuously obtained by the frame difference method, and continuous centroids of the difference image are obtained, achieving the continuous tracking of moving targets. The calculation method of the centroid points M_1, M_2 coordinates (\bar{X}, \bar{Y}) is:

$$\bar{X} = \frac{\sum x \sum y \cdot X \cdot f(x, y)}{\sum x \sum y \cdot f(x, y)} \tag{26.1}$$

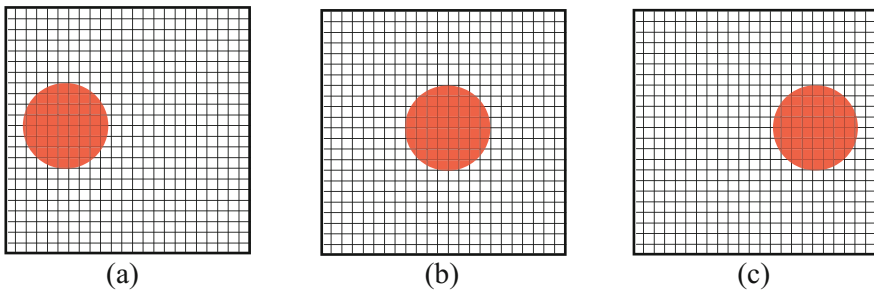
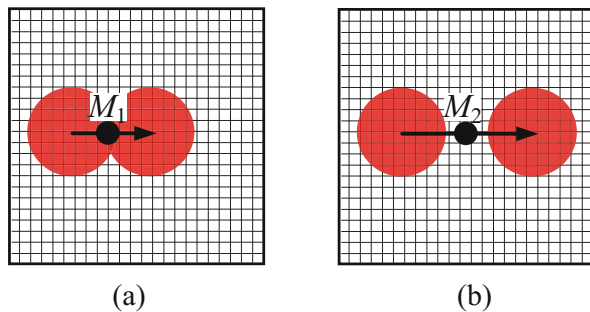


Fig. 26.6 Adjacent images. (a) k -th frame. (b) $(k + 1)$ -th fame. (c) $(k + 2)$ -th frame

Fig. 26.7 Frame difference method. (a) k -th and $(k + 1)$ -th frame. (b) k -th and $(k + 2)$ -th frame



$$\bar{Y} = \frac{\sum x \sum y \cdot Y \cdot f(x, y)}{\sum x \sum y \cdot f(x, y)} \quad (26.2)$$

The basic idea of the background learning algorithm is to model the background of the image. When the background model is established, the current image is compared with the background model, and the moving target could be detected according to the comparison result. The background model used in this system is the single Gaussian background model.

The basic idea of the single Gaussian background model is: the gray value of each pixel in the image is regarded as a random process and the probability that the gray value of a pixel of the point is subject to a Gaussian distribution. Let $I(x, y, t)$ denotes the pixel gray value of the pixel point (x, y, t) at time t , then

$$P(I(x, y, t)) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{-\frac{(x-\mu_t)^2}{2\sigma_t^2}} \quad (26.3)$$

where μ_t and σ_t are the expected value and standard deviation of the Gaussian distribution of the gray value of the pixel at time t , respectively. The background model for each pixel consists of an expected value μ_t and a deviation σ_t .

For illustration, an image of the scene on a river was obtained as the background image. The real-time tracking of the sailing vessel can be obtained by the background learning algorithm and the setting of the appropriate threshold, as shown in Fig. 26.8. Figure 26.8(a) is the video monitoring image, and Fig. 26.8(b) is the target tracking image processed by the background learning algorithm. There are influencing factors like clouds and hills. Hills remain static while clouds will move, and the system is still able to detect the coming ships.

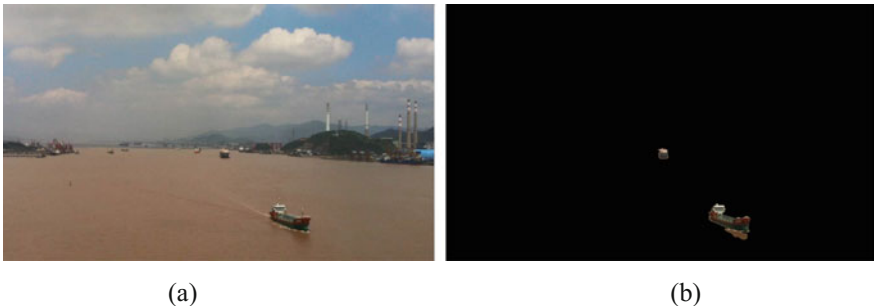


Fig. 26.8 Test results of background learning algorithm. (a) Original image. (b) Extraction of ships

26.3.2 Risk Assessment and Early Warning System

26.3.2.1 Warning Area Division

The width of the Grand Canal around the bridge is 80 m. The bridge has three spans, allowing small ships to pass in two spans in two directions simultaneously, while cargo ships could only go through the main span one by one. The nearest bridge in the north direction is 650 m away, and the nearest bridge in the south direction is 450 m away. Given a comprehensive consideration of the bridge condition, channel condition, ship types, ship velocity, and hardware for the computer vision system, the monitoring area for the bridge is 250 m in both sides.

According to the potential collision risk and navigation channel, the monitoring area was divided into four parts: traffic zone, danger zone, warning zone, and tracking zone. The traffic zone is an area with a width of 12.8 m centering on the center line of the navigation channel. The danger zone is within 50 m to the bridge and 0.5 m to the edge of the arch ring. Warning zone has a length of 150 m beyond the danger zone and 1.5 m from the arch ring.

For ships entering the tracking zone, the tracking system automatically tracks their dynamic navigation tracks. The risk assessment and early warning system are based on the measured data including the distance, speed, heading, nearest meeting point, and nearest meeting time of the ship in the bridge area. Also, the tracking system will monitor whether the ship has risky phenomenon, such as overspeed, yaw, etc. When the ship's illegal operation behavior is found, the system will take the initiative to issue a reminder to avoid the ship collision accident caused by the ship's illegal operation.

For ships entering the warning zone, risk assessment and early warning system will assess the ship collision risk in real time. The ship collision risk model consists of the minimum safety encounter distance and the shortest maneuver time. The minimum safe encounter distance and the shortest maneuver time are determined by the ship speed, ship type, tonnage, scale, and the environmental conditions of the ship's circumference. The risk assessment and early warning system receives the measured data such as the distance, speed, heading, nearest meeting point, and nearest meeting time of the tracking system in the bridge area and will continuously compute the minimum safety meeting distance and the time that the minimum safety is reached, and dynamic judgment of whether there is a danger of ship collision in real time.

For ship entering the danger zone which is in danger of collision, it will take the initiative to issue a ship collision warning and notify the ship to take evasive measures to avoid the ship collision accident to the greatest extent. The ship's illegal reminder or ship collision warning will be issued through the walkie-talkie or horn. Of course, it is extremely difficult to completely avoid a ship collision accident. When a ship collision accident is unavoidable, risk assessment and early warning of bridge-ship collision system automatically issues a ship collision warning and

strives for the personnel on the bridge and ships under the bridge to avoid casualties as much as possible.

26.3.2.2 System Warning Trigger Method

When the ship enters the warning zone, the system automatically initiates a transient prediction warning. The transient prediction warning is based on the current navigation state of the ship and the trajectory of the past short time to predict the behavior state and movement position of the ship in a short time and is based on the predicted result; it is determined whether the current warning level needs to be improved, so as to obtain more alarm response time. Active prediction warning is based on the navigational trajectory, speed, heading, and other information of the ship in the monitoring area to extract the navigation characteristics of the ship and proactively predict the ship collision probability and provide early warning. The system warning trigger classification is shown in Fig. 26.9.

26.3.2.3 Early Warning Event Risk Assessment

Bridge–ship collision warning should consider the possibility and danger of bridge–ship collision. The possibility of bridge–ship collision is considered by the relative position of the ship from the bridge. The risk of a bridge–ship collision is related to the ship weight, speed, and direction of impact of a particular structure. It is difficult to obtain accurate data on the weight of the ship through vision monitoring, so the

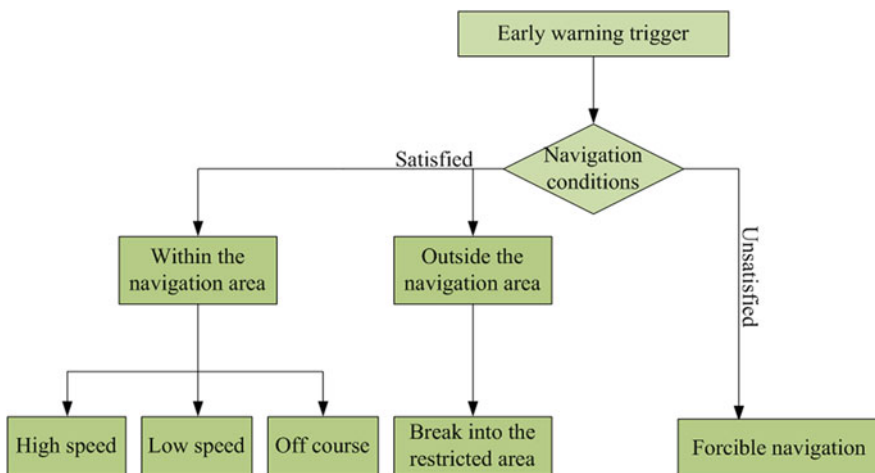


Fig. 26.9 System warning trigger classification

length of the ship is used instead of the weight of the ship to evaluate the risk of ship collision.

When an early warning event occurs, the system will obtain the corresponding risk assessment score based on the location of the early warning event, ship size, speed, heading, and other information and then perform weighted summation to calculate the early warning score and determine the warning level in a short time.

The system obtains the corresponding risk assessment score based on the information such as the position of the ship, the size of the ship, the speed, and the heading and then performs weighted summation to calculate the early warning score. The specific formula is as follows. The risk score of the monitoring area, ship size, navigation speed, and ship navigation direction are shown in Table 26.1.

$$Q = \sum_{i=1}^4 A_i \cdot \omega_i \tag{26.4}$$

where Q is the warning score, A_i is the characteristic risk score of the warning event behavior, as shown in Table 26.1, and ω_i is the feature weight of the warning event behavior, as shown in Table 26.2.

Then, according to the above-mentioned ship early warning score, the warning level of the bridge–ship collision is determined and the corresponding early warning measures are taken, as shown in Table 26.3.

Table 26.1 Risk score

Dangerous situation	Risk score		Early warning score calibration		
Monitoring area	A ₁	1	Primary area		
		0.5	Secondary area		
		0.25	Tertiary area		
Ship size	A ₂		Ship size definition	Ship tonnage	Ship length
		1	Large	>1000 t	>61 m
		0.5	Middle	500 t~1000 t	47~61 m
		0.25	Small	>50 t~<500 t	20~47 m
		0	Very small	<5 t	<20 m
Sailing speed	A ₃		Ship speed definition	Ship speed	
		1	Fast	≥5 km	
		0.5	Slow	<5 km	
Ship sailing direction	A ₄		Ship heading definition	Ship heading	
		1	Near the bridge	±71.5° (the direction in which the ship is facing forward)	
		0.5	Away from the bridge	Other directions	

Table 26.2 Analysis weights of different factors

Serial number	Monitoring area	Ship size	Sailing speed	Ship sailing direction
Weights ω_i	0.3	0.3	0.1	0.3

Table 26.3 Warning score and warning measures

Serial number	Interval of warning score	Alert level
1	[0, 0.7]	No warning
2	(0.7, 0.8]	Primary warning
3	(0.8, 0.9]	Intermediate warning
4	(0.9, 1]	Emergency warning

26.3.3 Post Recording System

The post recording system is combined with the intelligent video monitoring system to help the recording of potential collision. Video cameras were mounted on the bridge. When the tracking system detected abnormal tracks, it will trigger the recording system to record the scene happening. It helps to retain evidence for post collision process.

Due to the recording purpose, the hard disk of the recording system shall be large enough to restore video for around 2 weeks. The resolution of the camera shall be around 1920×1080 pixels to demonstrate clear details of the ships and bridge in case there is a collision incident. Accordingly, cameras with Ethernet port are adopted for its fast transmission speed. Also, due to the outdoor environment, the camera shall be robust enough to withstand temperature change. Considering the summer and winter temperature here, the working temperature shall be between -20 and $+65$ °C.

26.4 Field Test

Due to the strict law for protecting the scenic area around the Gongchen Bridge, a field test was conducted in another area for validation of the system, before the implement on the ancient bridge. The investigated bridge is located on the main channel of the ship import and export in Taizhou, China. The tides near the bridge area are urgent, and the ship flow is large with complicated ship types. There are many docks and berths, and the navigation environment is complicated. The bridge area is 1700 m wide, and the main channel is 900 m wide. According to the measured traffic volume of the maritime department, there are 272 ships per day in the waters of the bridge area. Due to the complexity of the navigation environment, this site selection can better verify the effectiveness of the active ship anti-collision system.

26.4.1 System Summary

The system is a bridge–ship anti-collision intelligent video monitoring system based on computer vision technology and IPTs. The hardware platform was built by video monitoring technology which mainly includes several industrial cameras, zoom lens, and a computer. The advanced digital IPTs are adopted as the core of video signal analysis theory. This computer vision-based ship–bridge anti-collision system is built to achieve the main objectives of static and dynamic identification, real-time data display, track description, database preservation, and multi-level intelligent alarm system. The system is shown in Fig. 26.10.

26.4.2 Monitoring Interface

The tracking system is part of the multi-stage active anti-collision system, and its system software has the following functions: real-time ship monitoring, ship navigation depiction, monitoring data display, warning information display, data storage, report generation, instant help, and system parameter configuration.

The monitoring images of the three cameras can be displayed simultaneously in real time, as shown in Fig. 26.11. Three levels of monitoring areas, separated by yellow and red lines, can be displayed in the picture, and static and dynamic ships can be locked in real time with a rectangular frame if the identified ship is visible. In addition, three monitoring picture windows can be scaled as shown in Fig. 26.11.

The three cameras can display the route trajectory of the ship moving in real time, and the background color of the track map also shows three levels of monitoring areas. In addition, the track map has coordinate display, and the coordinate value range can be calibrated and input according to the actual value. As shown in Fig. 26.12, the

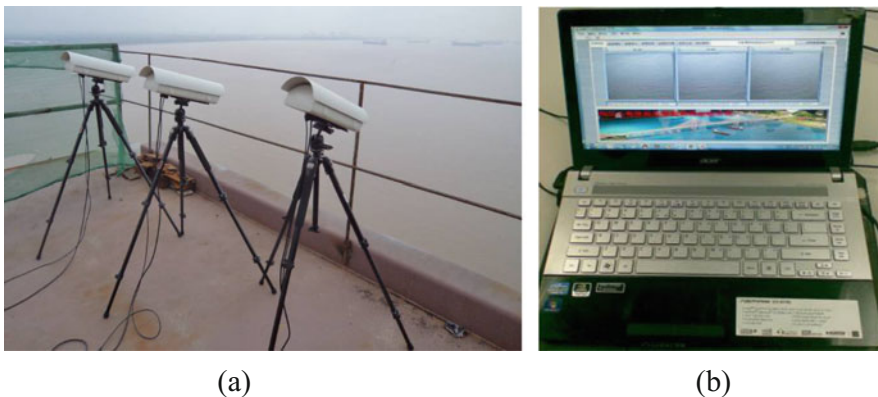


Fig. 26.10 Field test of the system. (a) Cameras. (b) Computer and software



Fig. 26.11 Real-time monitoring interface

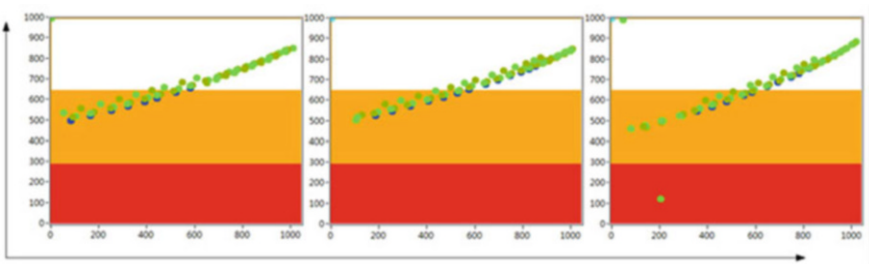


Fig. 26.12 Ship track depiction

three cameras track the number of identified ships, and the data of each ship's size, coordinates, speed, and direction can be displayed in real time.

26.4.3 Warning System

According to the size, speed, direction, and monitoring area of the ship monitored by the three cameras, the weighted sum is used to obtain the early warning score, and the warning score is displayed in real time. At the same time, the warning score is divided into three levels (primary warning, intermediate warning, and emergency warning), and respectively light the lights of different colors through the corresponding threshold, as shown in the following figure. When it is an emergency warning, the alarm ringtone is automatically triggered to alarm, as shown in Fig. 26.13.

As shown in Fig. 26.13, the data recognized by camera 1 is weighted to obtain a final warning score of 0.72, which is within the primary warning range, so the green light is on; the data recognized by camera 3 is weighted to obtain a final warning score of 1, which is within the emergency warning range, so the red light is on. The restricted area can be artificially defined in the three camera monitoring screens, as shown in the blue rectangle of Fig. 26.14. When the monitoring ship enters the restricted area, it automatically lights up red and directly triggers the alarm sound.

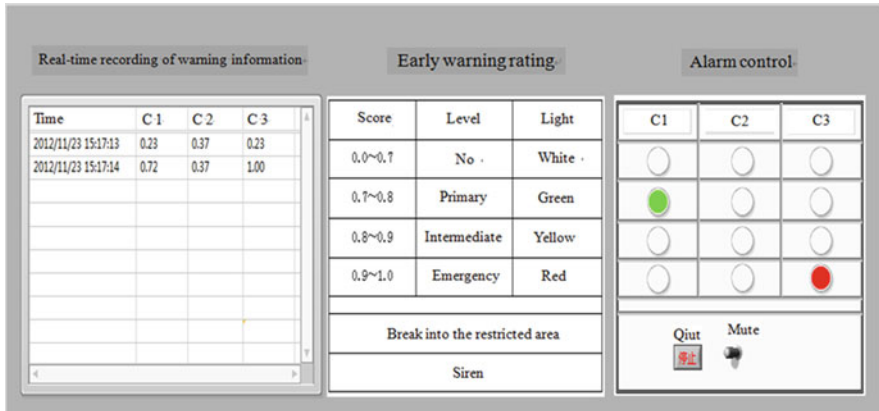


Fig. 26.13 Warning information display

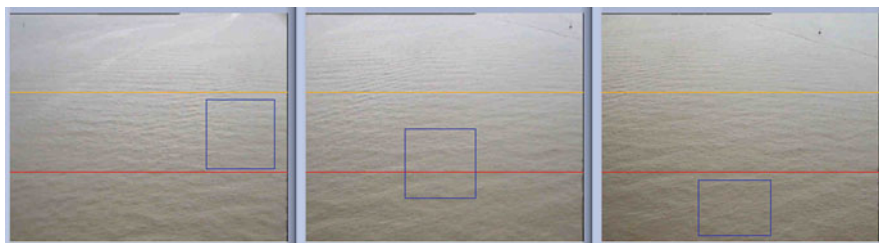


Fig. 26.14 Restricted area setting

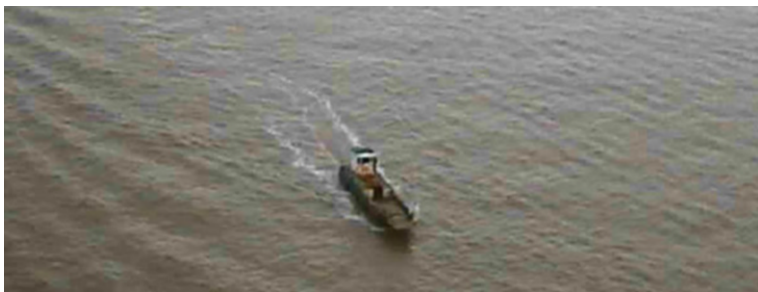


Fig. 26.15 Calibration ship

26.4.4 Ship Identification

The size of the calibration ship is about 3 m × 20 m, the speed is about 2 m/s, and it is basically kept at a constant speed. The video picture is shown in Fig. 26.15. The system requires calibration and test to facilitate static and dynamic identification.

According to the site environment, the debugged camera monitors the ships parked on the river surface, and the software system can identify multiple mooring ships, as shown in Fig. 26.16. The static ships are far away from the bridge, yet the system could still detect the existence of them. This allows sufficient time to identify the coming target and makes a potential warning in time.

According to the site environment, the debugged camera monitors the ships sailing on the river, and the software system can identify multiple ships that are sailing, as shown in Fig. 26.17. The results indicated that multi-ships on the river could be tracked steadily, it is important for practical application when there are more incoming ships.

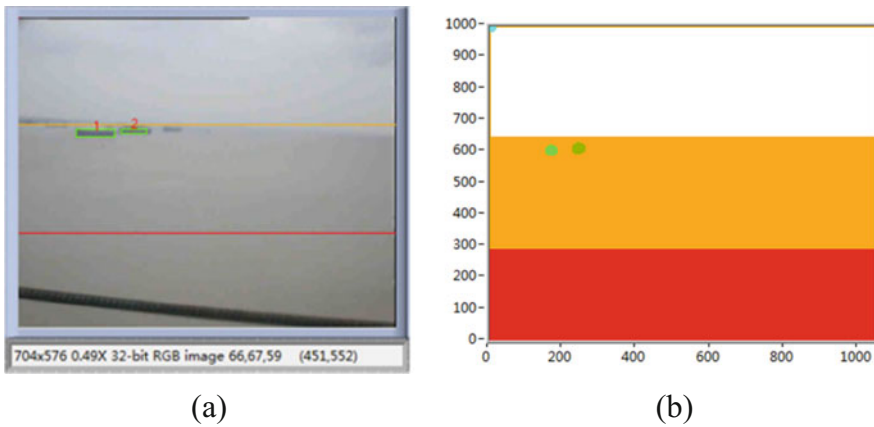


Fig. 26.16 Static monitoring. (a) Monitoring area. (b) Identification result

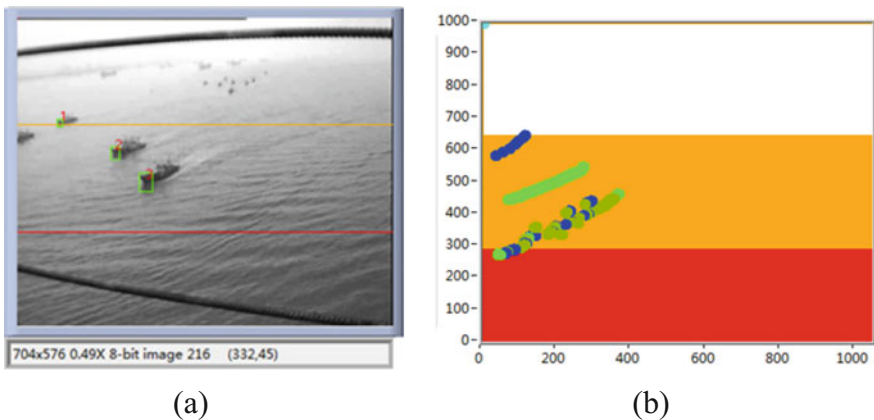


Fig. 26.17 Dynamic monitoring. (a) Monitoring area. (b) Identification result

26.5 Conclusions

A computer vision-based system was developed for ship–bridge collision protection. The system features are mainly reflected by (1) introducing intelligent video monitoring into the field of bridge engineering, actively monitoring the navigation of the waters in the bridge area, and actively reminding the ship of illegal operation and proactively alerting the ship to danger, maximally avoiding ship collision accident; (2) when a ship collision accident occurs, an alarm is issued in time to take the initiative to evacuate the personnel, personnel on the bridge and ships under the bridge could be alerted in advance to avoid casualties; (3) when a ship collision accident occurs, the recording camera system is used to record the whole course of the ship collision accident, so that the bridge owner or manager takes an active advantage in the ship collision accident claim. Also, the system has the functions of multilevel warning zone, an active reminder of ship violation operation, real-time assessment of ship collision risk, ship collision risk active warning, and ship collision accident automatic alarm.

The system utilizes all-round monitoring and active early warning to maximize the avoidance of ship collision accidents or reduce the severity of ship collision accidents. It avoids or reduces the direct or indirect losses to the economy and society caused by potential ship collision accidents, obtaining the economic and social benefits. Active anti-collision systems have significant cost advantages over passive collision avoidance systems. Therefore, the multistage active anti-collision system not only reduces direct or indirect losses but also saves bridge collision avoidance costs. With the construction of a bridge over the rivers and the development of shipping business, there will be an increasing risk of potential collision incidents. With the help of such a computer vision-based ship–bridge anti-collision system, active measures could be taken to reduce the collision risk and remarkable social and economical benefit will be obtained.

Acknowledgments The work described in this chapter was jointly supported by the National Science Foundation of China (Grant Nos. 51822810, 51778574), the Zhejiang Provincial Natural Science Foundation of China (Grant No. LR19E080002), and the Fundamental Research Funds for the Central Universities of China (Grant No. 2019XZZX004-01).

References

1. Vu, K. A. T., & Stewart, M. G. (2000). Structural reliability of concrete bridges including improved chloride-induced corrosion models. *Structural Safety*, 22(4), 313–333.
2. Choe, D. E., Gardoni, P., Rosowsky, D., & Haukaas, T. (2008). Probabilistic capacity models and seismic fragility estimates for RC columns subject to corrosion. *Reliability Engineering & System Safety*, 93(3), 383–393.
3. Li, J. B., Gong, J. X., & Wang, L. C. (2009). Seismic behavior of corrosion-damaged reinforced concrete columns strengthened using combined carbon fiber-reinforced polymer and steel jacket. *Construction and Building Materials*, 23(7), 2653–2663.

4. Simon, J., Bracci, J. M., & Gardoni, P. (2010). Seismic response and fragility of deteriorated reinforced concrete bridges. *Journal of Structural Engineering*, *136*(10), 1273–1281.
5. Alipour, A., Shafei, B., & Shinozuka, M. (2011). Performance evaluation of deteriorating highway bridges located in high seismic areas. *Journal of Bridge Engineering*, *16*(5), 597–611.
6. Zhu, L., Liu, W. Q., Fang, H., Chen, J. Y., Zhuang, Y., & Han, J. (2019). Design and simulation of innovative foam-filled lattice composite bumper system for bridge protection in ship collisions. *Composites Part B: Engineering*, *157*, 24–35.
7. Guo, Y. L., Ni, Y. Q., & Chen, S. K. (2017). Optimal sensor placement for damage detection of bridges subject to ship collision. *Structural Control & Health Monitoring*, *24*(9). <https://doi.org/10.1002/stc.1963>.
8. Fang, H., Mao, Y. F., Liu, W. Q., Zhu, L., & Zhang, B. (2016). Manufacturing and evaluation of large-scale composite bumper system for bridge pier protection against ship collision. *Composite Structures*, *158*, 187–198.
9. Liu, J. C., & Gu, Y. N. (2002). Simulation of the whole process of ship-bridge collision. *China Ocean Engineering*, *16*(3), 369–382.
10. Minorsky, V. U. (1958). An analysis of ship collisions with reference to protection of nuclear power plants (No. NP-7475). *Journal of Ship Research*, *3*(2), 1–4.
11. Meir-Dornberg, K. E. (1983). Ship collisions, safety zones, and loading assumptions for structures in inland waterways. *VDI-Berichte*, *496*(1), 1–9.
12. Sha, Y. Y., & Hao, H. (2012). Nonlinear finite element analysis of barge collision with a single bridge pier. *Engineering Structures*, *41*, 63–76.
13. Fan, W., & Yuan, W. C. (2014). Numerical simulation and analytical modeling of pile-supported structures subjected to ship collisions including soil-structure interaction. *Ocean Engineering*, *91*, 11–27.
14. Wan, Y. L., Zhu, L., Fang, H., Liu, W. Q., & Mao, Y. F. (2019). Experimental testing and numerical simulations of ship impact on axially loaded reinforced concrete piers. *International Journal of Impact Engineering*, *125*, 246–262.
15. Jiang, H., & Chorzepa, M. G. (2015). Evaluation of a new FRP fender system for bridge pier protection against vessel collision. *Journal of Bridge Engineering*, *20*(2). [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0000658](https://doi.org/10.1061/(ASCE)BE.1943-5592.0000658).
16. Fu, T. S., Garcia-Palencia, A. J., Bell, E. S., Adams, T., Wells, A., & Zhang, R. (2016). Analyzing prerepair and postrepair vibration data from the Sarah Mildred Long Bridge after ship collision. *Journal of Bridge Engineering*, *21*(3). [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0000856](https://doi.org/10.1061/(ASCE)BE.1943-5592.0000856).
17. Consolazio, G. R., & Cowan, D. R. (2005). Numerically efficient dynamic analysis of barge collisions with bridge piers. *Journal of Structural Engineering*, *131*(8), 1256–1266.
18. Davidson, M. T., Consolazio, G. R., & Getter, D. J. (2010). Dynamic amplification of pier column internal forces due to barge-bridge collision. *Transportation Research Record*, *2172*, 11–22.
19. Consolazio, G. R., Davidson, M. T., & Cowan, D. R. (2009). Barge bow force-deformation relationships for barge-bridge collision analysis. *Transportation Research Record*, *2131*, 3–14.
20. Yuan, P., & Harik, I. E. (2008). One-dimensional model for multi-barge flotillas impacting bridge piers. *Computer-Aided Civil and Infrastructure Engineering*, *23*(6), 437–447.
21. Zhu, B., Chen, R. P., Chen, Y. M., & Zhang, Z. H. (2012). Impact model tests and simplified analysis for flexible pile-supported protective structures withstanding vessel collisions. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, *138*(2), 86–96.
22. Xu, Y., & Brownjohn, J. M. W. (2018). Review of machine-vision based methodologies for displacement measurement in civil structures. *Journal of Civil Structural Health Monitoring*, *8*(1), 91–110.
23. Feng, D. M., Feng, M. Q., Ozer, E., & Fukuda, Y. (2015). A vision-based sensor for noncontact structural displacement measurement. *Sensors-Basel*, *15*(7), 16557–16575.
24. Feng, D. M., & Feng, M. Q. (2017). Identification of structural stiffness and excitation forces in time domain using noncontact vision-based displacement measurement. *Journal of Sound and Vibration*, *406*, 15–28.

25. Feng, D. M., & Feng, M. Q. (2017). Experimental validation of cost-effective vision-based structural health monitoring. *Mechanical Systems and Signal Processing*, 88, 199–211.
26. Feng, D. M., Scarangelo, T., Feng, M. Q., & Ye, Q. (2017). Cable tension force estimate using novel noncontact vision-based sensor. *Measurement*, 99, 44–52.
27. Dong, C. Z., Ye, X. W., & Jin, T. (2018). Identification of structural dynamic characteristics based on machine vision technology. *Measurement*, 126, 405–416.
28. Wu, L. J., Casciati, F., & Casciati, S. (2014). Dynamic testing of a laboratory model via vision-based sensing. *Engineering Structures*, 60, 113–125.
29. Khuc, T., & Catbas, F. N. (2017). Computer vision-based displacement and vibration monitoring without using physical target on structures. *Structure and Infrastructure Engineering*, 13(4), 505–516.
30. Tian, L., & Pan, B. (2016). Remote bridge deflection measurement using an advanced video deflectometer and actively illuminated LED targets. *Sensors-Basel*, 16(9), 1344.
31. Lee, J. J., Cho, S., Shinozuka, M., Yun, C. B., Lee, C. G., & Lee, W. T. (2006). Evaluation of bridge load carrying capacity based on dynamic displacement measurement using real-time image processing techniques. *International Journal of Steel Structures*, 6(5), 377–385.
32. Feng, D. M., & Feng, M. Q. (2015). Model updating of railway bridge using in situ dynamic displacement measurement under trainloads. *Journal of Bridge Engineering*, 20(12). [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0000765](https://doi.org/10.1061/(ASCE)BE.1943-5592.0000765).
33. Chen, J. G., Adams, T. M., Sun, H., Bell, E. S., & Buyukozturk, O. (2018). Camera-based vibration measurement of the World War I memorial bridge in Portsmouth, New Hampshire. *Journal of Structural Engineering*, 144(11). [https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002203](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002203).
34. Khuc, T., & Catbas, F. N. (2018). Structural identification using computer vision-based bridge health monitoring. *Journal of Structural Engineering*, 144(2). [https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001925](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001925).
35. Ojio, T., Carey, C. H., O'Brien, E. J., Doherty, C., & Taylor, S. E. (2016). Contactless bridge weigh-in-motion. *Journal of Bridge Engineering*, 21(7). [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0000776](https://doi.org/10.1061/(ASCE)BE.1943-5592.0000776).
36. Jiao, J., Zhang, Y., Sun, H., Yang, X., Gao, X., Hong, W., Fu, K., & Sun, X. (2018). A densely connected end-to-end neural network for multiscale and multiscale SAI ship detection. *IEEE Access*, 6, 20881–20892.
37. Liu, G., Zhang, Y. S., Zheng, X. W., Sun, X., Fu, K., & Wang, H. Q. (2014). A new method on inshore ship detection in high-resolution satellite images using shape and context information. *IEEE Geoscience and Remote Sensing Letters*, 11(3), 617–621.
38. Li, S., Zhou, Z. Q., Wang, B., & Wu, F. (2016). A novel inshore ship detection via ship head classification and body boundary determination. *IEEE Geoscience and Remote Sensing Letters*, 13(12), 1920–1924.
39. Liu, W. C., Ma, L., & Chen, H. (2018). Arbitrary-oriented ship detection framework in optical remote-sensing images. *IEEE Geoscience and Remote Sensing Letters*, 15(6), 937–941.
40. Liu, Z. Y., Zhou, F. G., Bai, X. Z., & Yu, X. Y. (2013). Automatic detection of ship target and motion direction in visual images. *International Journal of Electronics*, 100(1), 94–111.
41. Lin, H. N., Shi, Z. W., & Zou, Z. X. (2017). Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 14(10), 1665–1669.

About the Authors

Hadi Aliakbarpour received his PhD degree in automation and robotics from the University of Coimbra, Coimbra, Portugal, under the supervision of Prof. J. Dias. He was a researcher with the Institute of Systems and Robotics, Coimbra, from 2007 to 2013. He is currently an assistant research professor with the Electrical Engineering and Computer Science Department (EECS), University of Missouri at Columbia, Columbia, MO, USA. He is also a faculty member of Computational Imaging and VisAnalysis (CIVA) Lab, at the EECS, University of Missouri. His research interests include computer vision, video analytics, remote sensing, large-scale multi-view 3D reconstruction and structure from motion, computer vision algorithm development for low-power embedded systems, and robotics. He has several publications in international journals, books, and conferences. Dr. Akbarpour holds a US patent related to his field of research and a number of (provisional) patent applications. He has actively participated in several research and industrial (technology transfer) projects within Europe and the USA.

Vicente Alarcon-Aquino received his BSc degree from the Veracruz Institute of Technology; MSc degree from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico; and the PhD and DIC degrees from Imperial College London, University of London, UK, all in Electrical Engineering. He is now a full professor in the Department of Computing, Electronics, and Mechatronics at Universidad de las Americas Puebla (UDLAP), Mexico. Previously, from 2006 to 2018 he held the positions of graduate coordinator, head of department, and member of several academic, research, and advisory councils at UDLAP. He has authored several research articles in refereed journals, books, and conference proceedings, has written a book on MPLS networks, and has numerous citations to his research articles. He is member of the Mexican national research system (SNI) level 1 and has been elected to membership of the Mexican Academy of Sciences (AMC). His research interests include network security and monitoring, network anomaly detection, wavelet-based signal processing, and multiresolution techniques.

Eteزاز Abo Al-Izam was born in 1992. She studied Computer Engineering at the University of Aleppo, Syria. She received her BSc degree in 2016. She is an MSc candidate. Her research interests lie in the fields of intelligent systems, stereo vision, embedded system, SLAM, Kalman filter, mobile robot, and deep learning. She is an experienced Android developer with a demonstrated history of working in the computer software company (Beetronix, Automata4Group), skilled in C#, WPF, Microsoft SQL Server, HTML5, CSS, Java Script, and Java.

Brice Allen is the lab manager and instructor in the Department of Physics and Engineering of Northwest Nazarene University in Nampa, Idaho. Brice is also a researcher in the Robotics Vision Lab and is currently working on using artificial intelligence for fruit yield estimation. Prior to moving to Northwest Nazarene University, Brice worked for Textron Defense Systems. Brice has a BS in Physics and Mathematics from Eastern Nazarene College in Quincy, Massachusetts.

Víctor Andaluz was born in Ambato, Ecuador, in 1984. He graduated in Electronics and Control Engineering from the Escuela Politécnica Nacional, Ecuador, in 2008, and obtained a PhD in Control Systems Engineering in 2011 from the Universidad Nacional de San Juan. Currently, he is professor and researcher at the Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador. His area of research is intelligent control applied to production systems, mobile robots, and rehabilitation of people disabilities.

Peng-Peng Ang is currently a Master in the Department of Civil Engineering at Zhejiang University, China. He is currently majoring in bridge engineering. His current research topic is structural health monitoring. He received his Bachelor's degree from the College of Civil Engineering at Hefei University of Technology in 2018. He won the third prize of the 2017 National Youth Science Innovation Experiment and Works Competition in Hefei. He participated in the innovation test of college students, and the experimental theme was the development of antipollution permeable concrete (2017). He was selected as an outstanding graduate of Anhui Province in 2018.

Steve Arscott was born in Plymouth (UK). He obtained a PhD from the University of Manchester (UK) in 1994. Following postdoctoral positions at the University of Leeds (UK) and the University of Lille (France) at the Institute of Electronics, Microelectronics, and Nanotechnologies (IEMN), he joined the Centre National de la Recherche Scientifique (CNRS) as a research scientist based at IEMN. His research interests focus on micro and nanotechnologies and their applications. He has made contributions in a broad spectrum of areas ranging from defects in semiconductors, THz electronics, piezoelectric materials, ferroelectric materials, frequency bulk acoustic wave devices, polymeric materials, microfluidics, electrospraying, electrospray tip technology, electrowetting, surface wetting and dewetting, evaporation of micro droplets, surface patterning and lithography, liquid films and bubbles, capillary effects, miniaturized fuel cells, micro and nanotechnology fabrication technologies, micro and nano electromechanical systems, miniaturized components for mass spectrometry, flexible and stretchable electronics and systems, cracking

studies in thin films, piezo-resistance and piezo-impedance in semiconductors, miniaturized probes for on-chip ground-signal-ground microwave measurements and for tip-enhanced Raman spectroscopy (TERS), fabrication of nanoparticles, and devices for fundamental spin studies.

Daniel Hernández-Balbuena was born on July 25, 1971. He received the BS degree from Puebla Autonomous University, Puebla, México, in 1996, and the MS degree from Ensenada Center for Scientific Research and Higher Education, Baja California, México, in 1999. He received his PhD degree in Baja California Autonomous University in 2010. His research interests are in the areas of time and frequency metrology, design and characterization of microwave devices and systems RF measurements, research applications of unmanned aerial vehicles, and image digital processing.

Isela Bonilla-Gutierrez received her BE degree in Communications and Electronics Engineering from the University of Colima, Mexico, in 2003; MSc degree in Electronics from the Autonomous University of Puebla, Mexico, in 2006; and PhD degree in Electrical Engineering from the Autonomous University of San Luis Potosí in 2011. She was an associate professor at the University of Sonora, Mexico, in 2011. In 2012 she joined the College of Sciences of the Autonomous University of San Luis Potosí, Mexico, where she is currently a professor of Electronic Engineering and Instrumentation. She has published more than 50 refereed papers in scientific journals and congresses. Her current research interests include impedance/force control, vision-based control, and rehabilitation robotics.

Christophe Boyaval was born in Lille (France). He joined the Institute of Electronics, Microelectronics, and Nanotechnologies (IEMN) in 1995. He is currently an engineer, and his main activities concern technological fabrication of micro- and nanodevices and characterization by means of microscopy techniques including scanning electronic microscopy (SEM). Recently, he has been involved in the development of new class of on-wafer prober operating in a SEM dedicated to the microwave characterization of nanodevices.

Duke M. Bulanon is an associate professor in the Department of Physics and Engineering of Northwest Nazarene University in Nampa, Idaho. Duke is directing the Robotics Vision Lab and is currently working on the development of a crop monitoring platform for specialty crops. Prior to moving in Idaho, Duke has worked in the development of a robotic harvesting system for oranges at the University of Florida. He has also developed automated system for the detection of citrus diseases using multispectral and hyperspectral imaging both in the lab and field applications. He has also worked in the area of Precision Agriculture as a Japan Society for the Promotion of Science Fellow at Hokkaido University. Duke received his Master's and PhD in Agricultural Engineering from Iwate University in Japan and has a BS Mechanical Engineering from the University of San Carlos (Philippines).

Julio Francisco Hurtado-Campa was born in Mexicali, Baja California, Mexico, on June 28, 1998. He is a mechatronics engineering student currently enrolled in CETYS University, pursuing a Bachelor's degree as a member of the class of 2020.

As a college student, he has participated in numerous innovation and design projects in areas such as underwater robotics, bioengineering, avionics, telecommunications, and space systems. He has been involved in regional and national competitions regarding engineering and the humanities, obtaining top places in many of them. He is currently focusing on research and development of space systems and satellite telecommunications technology.

Cristobal Capiz-Gómez is a full-time professor at CETYS Universidad, Campus Mexicali, Mexico. He holds a Bachelor of Science in Cybernetics Electronics Engineering from CETYS University, Mexico, and a Master of Science from Arizona State University, USA. Professor Capiz currently teaches in the areas of digital electronic design, computer architecture, digital signal processing, and automation; also he currently is the coordinator of the Mechatronics Engineering Program.

Ricardo Carelli was born in San Juan, Argentina. He graduated in Engineering from the National University of San Juan, Argentina, and obtained a PhD degree in Electrical Engineering from the Universidad Nacional de Mexico (UNAM). He is a full professor at the Universidad Nacional de San Juan and a senior researcher of the National Council for Scientific and Technical Research (CONICET, Argentina). Professor Carelli is the director of the Instituto de Automática (INAUT), Universidad Nacional de San Juan (Argentina). His research interests are in robotics, manufacturing systems, adaptive control, and artificial intelligence applied to automatic control. Professor Carelli is a senior member of IEEE and a member of AADECA-IFAC.

Moisés J. Castro-Toscana received the engineering degree in Mechatronics from the Instituto Tecnológico de Mexicali, Mexico, in 2011, and the MC degree from the Instituto de Ingeniería of the Universidad Autónoma de Baja California, Mexicali, Mexico, in 2015. He is a professor at the Instituto Tecnológico de Mexicali, teaching basic science to engineering students. He is currently a full-time student on his fourth year of PhD in the Facultad de Ingeniería of Universidad Autónoma de Baja California in the Electronic and Instrumentation Department, working on the development of navigation application through inertial navigation systems. Also, he has worked on papers and book chapters on different topics such as Stereo Vision Systems, Laser Vision Systems, and Inertial Navigation Systems. He presented his results at ISIE 2017 international conference in Edinburgh, Scotland, and at the international conference of IECON 2018 in Washington D.C., USA.

Cesar Chavez-Olivares received his BE degree in Electronics Engineering from the Autonomous University of Aguascalientes, México, in 2006, and the MEng and PhD degrees in Electrical Engineering from the Autonomous University of San Luis Potosí, México, in 2009 and 2014, respectively. He was a CONACYT Research Fellow at the Aguascalientes Technological Institute, México, from 2014 to 2015. In 2015 he joined the Center of Engineering Sciences at the Autonomous University of Aguascalientes where he is currently a full-time professor of robot manipulators. His current research interests include biorobotics, haptic devices, identification, and control of robot manipulators.

Jason Colwell is an associate professor in the Department of Mathematics and Computer Science of Northwest Nazarene University in Nampa, Idaho. He holds a PhD in Mathematics from Caltech, and his current research interests include applications to image processing and organic chemistry.

Sergio Rolando Cruz-Ramírez is an assistant professor at Tecnológico de Monterrey, San Luis Potosí campus. He earned his PhD in Engineering from Osaka University, Japan, in 2009. His thesis project was related to a robotic dismantling system to assist workers in the construction field applying robotic arms, machine vision, and artificial intelligence. With more than 15 years of academic experience, Dr. Cruz has taught different courses including robotics, automatic control, capstone projects, vision, and programming, among others. He has always been committed to the academic development of the students, looking for better strategies to share the knowledge and proposing and carrying out educational innovation projects where students are enrolled in activities of real-life challenges. In addition, he has collaborated with companies coordinating different projects such as greenhouse automation, domotics, and industrial robots programming. Dr. Cruz has won several awards like a national contest with his Master's thesis, gotten scholarships for graduate studies in Mexico and Japan, and published papers in national and international congresses as well as journal articles in the field of robotics.

Gilles Dambrine received his engineering diploma, PhD degree, and the Habilitation à Diriger des Recherches (HDR) degree from the University of Lille, France, in 1986, 1989, and 1996, respectively. He was permanent researcher at the CNRS between 1989 and 1999. He is full professor of Electronic at the University of Lille since 1999. His main research interests are concerned with the characterization and modeling of ultimate low-noise devices for application in millimeter and submillimeter wave ranges. Over these few years, his research interests are oriented to the study of the microwave properties and applications of nanodevices. He is author and coauthor of 180 papers and communications and 11 chapters of books, in the field of microwave devices. Since 2010, he is deputy director of the Institute of Electronics, Microelectronics, and Nanotechnologies (IEMN), gathering about 500 researchers and PhD students. Since 2017, he is scientific delegate in charge of industrial partnership and innovation at the CNRS headquarter. He is the scientific coordinator of Nanoscience-Characterization-Center "ExCELSiOR" (www.excelsior-ncc.eu) and leader of European H2020 project (www.mmama.eu) on thin film materials microwave nano-characterization. He received the IEEE Fellow grade in 2016.

Sergey Vladimirovich Dvoynishnikov graduated from the Physics Department of Novosibirsk State University, doctor of technical science, and head of the laboratory of the fundamentals of safety and effective use of the reactor plants of Kutateladze Institute of Thermophysics of the Siberian Branch of the Russian Academy of Sciences. He is also winner of the Russian Federation Governmental Prize in the field of science and technology for young scientists. He is the author and coauthor of 145 scientific papers, including 2 monographs and 14 patents. His research interests are in the fields of optical-electronic diagnostic methods, digital signal processing,

laser Doppler anemometry, and optical triangulation. He proposed and implemented new complex methods of parametrical triangulation based on the modulation of the optical source and multidimensional regression analysis of the spatial and temporal ensembles of experimental data. Such approach ensured the measurement of the geometric parameters of static and dynamic objects in phase-inhomogeneous media with a record of small error. A number of informational diagnostic systems implementing methods of parametrical triangulation have been created. Created systems were adapted to the characteristics of phase-inhomogeneous media and real conditions of domestic production. They successfully passed industrial tests and introduced at metallurgical and machine-building enterprises of the Russian Federation.

Shiyu Feng is a PhD candidate in the School of Mechanical Engineering at Georgia Institute of Technology. He obtained his Master of Engineering degree from the University of California at Berkeley. He is currently a graduate research assistant in the Intelligent Vision and Automation Laboratory (IVALab) directed by Dr. Patricio Vela. His research work is exploring navigation solutions in perception space through stereo vision system, which can achieve scalability and adaptivity based on different computation resources. He is also introducing machine learning techniques into traditional navigation system to facilitate and improve the performance. As a doctoral student in ME program, he has the knowledge of system dynamics, controls, and vision-based sensory processing. He is also a graduate teaching assistant of the course ME2110: Creative Decisions and Design to train and guide students about robotics machine design. In summer 2018, he interned as an autonomous driving perception engineer at SF Motors.

Wendy Flores-Fuentes was born in Mexicali, Baja California, Mexico, in January of 1978. She received the Bachelor's degree in Electronic Engineering from the Autonomous University of Baja California in 2001, Master's degree in Engineering from Technological Institute of Mexicali in 2006, and PhD degree in Science, Applied Physics, with emphasis on optoelectronic scanning systems for SHM, from the Autonomous University of Baja California in June 2014. Until now she is the author of multiple impact journal articles in Elsevier, IEEE Emerald, Hindawi, Wiley, and Springer; 8 book chapters and 4 books in InTech, Nova Science Publisher, Lambert Academic Publisher, IGI Global, and Springer; and 30 proceedings articles in IEEE ISIE (2014–2017), IECON (2014 and 2018), the World Congress on Engineering and Computer Science (IAENG 2013), IEEE Section Mexico IEEE ROCC2011, and the VII International Conference on Industrial Engineering ARGOS 2014. Recently, she has organized and participated as chair of Special Session on “Machine Vision, Control and Navigation” at IEEE ISIE (2015, 2016, and 2017) and IECON (2018). She has been a reviewer of multiple articles in Taylor and Francis, IEEE, Elsevier, Sensors MDPI, and EEMJ. Currently, she is a full-time professor-researcher at Universidad Autónoma de Baja California, at the Faculty of Engineering, and is the coordinator of the Physics area in the Basic Science Department. She has been incorporated to CONACYT National Research System since 2015.

Juan C. Galan-Hernandez was born in Puebla, Mexico, in 1980. He received his PhD degree in Computer Science from the Universidad de las Americas Puebla, Mexico, and the MSc degree in Computer Science from the Autonomous University of Puebla, Mexico. In 2012, he spent a 6-month research staying at the Imperial College, London, UK. In 2014 he conducted a research collaboration with the National Institute of Astrophysics, Optics, and Electronics (INAOE), Puebla, Mexico. His current research interests include signal processing, audio and video compression, computer security, distributed systems, and deep learning.

Ranjan Ganguli is currently professor in the Aerospace Engineering Department of the Indian Institute of Science, Bangalore, India, since 2000. His research area spans computational engineering. He received his PhD and MS degrees from the Department of Aerospace Engineering at the University of Maryland, College Park, USA, in 1994 and 1991, respectively, and his BTech degree from the Indian Institute of Technology, Kharagpur, India, in 1989. He has also worked at GE and Pratt & Whitney for 3 years. He has published 205 journal papers, 128 conference papers, and 6 books. His books include *Engineering Optimization* and *Gas Turbine Diagnostics* published by CRC Press. He has also published *Smart Helicopter Rotors*, *Finite Element Analysis of Rotating Beams*, and *Structural Health Monitoring using Genetic Fuzzy System* published by Springer. He is a fellow of the American Society of Mechanical Engineers, an associate fellow of the American Institute of Aeronautics and Astronautics, a fellow of the Indian National Academy of Engineering, and a fellow of the Royal Aeronautical Society, UK. He is a senior member of IEEE. He was awarded Humboldt Fellowship in 2007, Fulbright Senior Research Fellowship in 2011, and the Royal Academy of Engineering Distinguished Visiting Fellowship in 2019. His research publications have been cited over 5500 times and his h-index is 44 in Google Scholar.

Néstor Aarón Orozco-García was born in Mexicali, Baja California, México, on August 12, 1975. He is a professor and an enthusiastic student of Robotics and Automation dedicated to teaching and research and system integration. He received a degree in Electronic Cybernetics Engineering in 1996 at the “Centro de Enseñanza Técnica y Superior” Campus, Mexicali. Néstor worked in the manufacturing industry for 7 years and later worked in a company dedicated to the development of industrial integration for automation systems and control equipment design, as project manager. He received a Master’s degree in Electronic Engineering from the Instituto Tecnológico de Mexicali in 2009. Almost at the same time, he ventured into academia, teaching subjects in the Cybernetics and Mechatronics engineering specialty at CETYS Universidad Campus Mexicali; in addition, he still works in a systems solutions development company for automation and control. Eventually he obtained the position of coordinator of the career in Cybernetics Electronic Engineering Program where he currently works. He participates in research projects, such as the construction of an intelligent transportation scheme for autonomous vehicle for use at the institution’s facilities.

Juan Manuel Terrazas-Gaynor graduated from CETYS University in Electronic Cybernetics Engineering in 2002 and obtained his Master’s and PhD degrees in

Semiconductor Engineering in 2007 and 2009, respectively, at the Institute of Engineering of Universidad Autónoma de Baja California. Since 2002 until the present time, he has worked on material characterization, electrical analysis, and reliability and failure analysis of semiconductor electronic devices. He currently works as a full-time professor at the Engineering School of CETYS University and as a researcher at the Center for Innovation and Design (CEID)—CETYS University. His main areas of interest are industrial electronics, advanced manufacturing, automation, and industrial IoT.

Pilar Gomez-Gil was born in Puebla, Mexico. She received her BSc degree in 1983 from the Universidad de las Americas A.C, Mexico; MSc in 1991 from Texas Tech University, USA; and PhD degree in 1998 from the same university, all in Computer Science. She is currently a titular researcher in the Computer Science Department at the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico. She is member of the Mexican national research system (SNI), level 1. Her research interests include artificial neural networks, time series prediction, signal and image processing, and pattern recognition.

Emilio J. Gonzalez-Galvan received the Bachelor's and Master's degrees in Mechanical Engineering from the University of Guanajuato, Mexico, in 1990 and 1991, respectively. From 1991 to 1996, he was a Fulbright scholar at the University of Notre Dame in Notre Dame, IN, where he obtained a PhD degree in Mechanical Engineering in 1995. He was a postdoctoral fellow at the same university in 1996. In that year he joined the College of Engineering at the Autonomous University of San Luis Potosí (UASLP) in Mexico where he is a professor and researcher. From 2003 to 2005, he was appointed president of the Mexican Robotics Association. From August 2007 to August 2008, he was a visiting scholar at the Massachusetts Institute of Technology. Currently, Dr. Gonzalez is the academic secretary of the College of Engineering at UASLP. He has published more than 120 refereed papers in scientific journals and congresses. His research and professional interests include vision-based robot control, human-robot interaction, and rehabilitation robotics.

Maki K. Habib obtained his PhD-Eng in Intelligent and Autonomous Robots, University of Tsukuba, Japan. He was with RIKEN Japan, RISO Laboratories, Japan, and visiting researcher at EPFL, Switzerland. He was visiting expert under Asian Development Bank; associate professor at UTM, Malaysia; and a senior manager at MCRMA, Malaysia. He was senior research scientist with GMD, Japan, and associate professor with Monash University. Then he was appointed as full professor at Swinburne University. He was invited professor at KAIST, Korea, and visiting professor at Saga University, Japan. He is currently a full professor at American University in Cairo, Egypt. He edited 6 books and has more than 230 papers published in internationally recognized journals and conferences. His research interests include human adaptive and friendly mechatronics, autonomous navigation, humanitarian demining, intelligent control, telecooperation, distributed teleoperation and collaborative control, wireless sensor networks and ambient intelligence, and biomimetic robots.

Kamel Haddadi (MSc Lille 2003, PhD Lille 2007) is an associate professor of Electronics and Electrical Engineering in the University of Lille. His main research interests are the development of microwave measurement techniques for nondestructive evaluation (NDE) and electrical nanometrology. He is in charge of microwave aspects in the frame of the French PIA Equipex Excelsior and European H2020 project (www.mmama.eu). He is member of the IEEE TC-25 RF Nanotechnology and associate editor of IEEE TIM and IET SMT. He acts regularly as a reviewer for several international journals and conferences. He has authored/coauthored more than 100 journal and conference papers.

Dalia Kass Hanna was born in 1987. She studied Control Engineering and Automation at the University of Aleppo, Syria. She received her MSc degree in 2016. Since 2011, she is a research assistant at the Dept. of Mechatronics Engineering, Laboratory of Integrated Mechatronics Systems. Her research interests lie in the fields of advanced intelligent mechatronics systems, embedded systems, Kalman filter, computer vision system, SLAM, intelligent motion approaches, robotics dynamics, and modern control systems.

Danilo Cáceres-Hernández was born on November 9, 1975. He graduated from José Dolores Moscote High School, Panamá, Panamá, on March 11, 1994. He received his BS in Electrical and Electronic Engineering from the Technological University of Panamá (UTP), in 2004. From January 1, 2002, to March 14, 2005, he was working at the Computer Laboratory at the Department of Electrical Engineering. From March 2005 to the present, he is a faculty member in the Department of Electrical Engineering at Technological University of Panamá. In 2011, he completed his MSc of Science in Electrical Engineering at the University of Ulsan. He received his PhD in Electrical Engineering from the University of Ulsan, South Korea, in February 2017. Since 2017, his research interests are focused on intelligent systems, robotics, autonomous navigation, and computer vision. Since 2017, he is currently a full-time professor in the Electrical Department, Universidad Tecnológica de Panamá.

Tyler Hestand is a student researcher in the Robotics Vision Lab of Northwest Nazarene University and has been working on the fruit yield estimation project since January 2018. Tyler is currently a senior taking up Bachelor of Science in Engineering (Electrical Engineering concentration) and Mathematics. He is currently an electrical intern at CTA Architects Engineers at their Boise office.

Takeshi Ikeda is currently a junior associate professor in the Department of Mechanical Engineering, Faculty of Engineering, Sanyo-Onoda City University. He received his Bachelor's degree in 2000, Master's degree in 2002, and Doctor of Engineering from the University of Fukui in 2006. He was an assistant professor at Kyushu University from 2006 to 2014. His research area is dynamics and modeling of mobile robot and mobile manipulator with carrying objects for safety and fastest guidance control, recognition of travel environment using a RGB-D sensor for mobile robot, research on visual recognition and harvest mechanism for autonomous agriculture robot, motion planning for remote control robot, interface for

handicapped person to control the devices or robot in welfare robotics, development of rehabilitation brace for flexor tendon, development of robot hand to grasp, handling or pick up the soft target, and control the robot using microcomputers, camera, and RGB-D sensor. He is a member of IEEE, JSME, and RSJ.

Mikhail Ivanov was born on July 18, 1989 in Kharkov, Ukraine. He received his BS and MS degrees in Kharkov National Aerospace University “KhAI,” Kharkiv, Ukraine, in 2010 and 2012, respectively. Mikhail Ivanov, in October 2013, has joined the Kharkov National Aerospace University and was working as the professor assistant and vice dean of Aircraft Control Systems Faculty. Since 2012 till the present time, he is represented by his research works in several International Congresses of IEEE in the USA and Great Britain and international conferences in Ukraine and Russia. He published two international journals with impact factor, book chapters, and nine papers on national and four papers on international conferences. In 2016 he was invited to receive the PhD degree in Engineering Institute, Autonomous University of Baja California, Mexico, with the topic “Distributed scanning for objects recognition in the behavior of mobile robotic group using the optical 3D technical vision system.”

Tao Jin is currently a PhD candidate in the Department of Civil Engineering at Zhejiang University, China. His research interests mainly contain computer vision and artificial neural networks-based structural health monitoring technology like displacement monitoring, identification of structural dynamic characteristics, crack detection, etc. Before his PhD education, he worked in a bridge and tunnel maintenance company as a technician for structural evaluation for different kinds of urban bridges and structural health monitoring system maintenance of several long-span bridges for 2 years. Mr. Jin received his Master’s degree from the Faculty of Architectural, Civil Engineering and Environment at Ningbo University in 2014 and received his Bachelor’s degree in the same department in 2011.

Kang-Hyun Jo (M’96–SM’16) received the PhD degree in Computer-Controlled Machinery from Osaka University, Suita, Japan, in 1997. After a year’s experience at ETRI as a postdoctoral research fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. He was the director or an AdCom member of the Institute of Control, Robotics, and Systems and the Society of Instrument and Control Engineers and the IEEE IES Technical Committee on Human Factors chair. He has also been involved in organizing many international conferences such as International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, International Conference on Human System Interactions, and Annual Conference of the IEEE Industrial Electronics Society. He is currently serving in the Administrative Committee of the IEEE IES, and he is an editorial board member of international journals, such as the *International Journal of Control, Automation, and Systems* and the *Transactions on Computational Collective Intelligence*. His research interests include computer vision, robotics, autonomous vehicle, and ambient intelligence.

Abdulkader Joukhadar was born in 1968. He received his PhD degree from Aberdeen University, UK, in 2004. Currently, he is an associate professor at the Department of Mechatronics Engineering, Faculty of Electrical and Electronic Engineering, University of Aleppo. The central theme of his research interest is in the field of robotics, robot control, probabilistic robotics, and intelligent control system.

Ivan Konstantinovich Kabardin graduated from the Physics Department of Novosibirsk State University and is a philosophy doctor of technical science and deputy head of the physical processes modeling Laboratory of Kutateladze Institute of Thermophysics of the Siberian Branch of the Russian Academy of Sciences. He is a winner of the Russian Federation Governmental Prize in the field of science and technology for young scientists. He is the author and coauthor of 118 scientific papers including 4 patents. His research interests are in the fields of heat and mass processes diagnostics via optical methods, laser Doppler anemometry, and particle image velocimetry. The PhD thesis of Kabardin Ivan Konstantinovich “Development and application of optical-laser techniques for improving the efficiency of wind generators” is devoted to the study of the vortex wake behind the rotor model of a wind generator using modern noncontact optical-laser diagnostic methods and the development of optical-laser methods for diagnosing icing of its blades. He experimentally investigated and diagnosed by optical-laser methods vortex structures that occur in the wind turbine rotor wakes. The technique of optical-laser diagnostics of unsteady eddy flows based on the integrated use of LDA and PIV measurement technologies has been developed. An optical-laser method has been developed for diagnosing the geometric dimensions of ice on the basis of total internal reflection.

Kenichi Kanatani received his BE, MS, and PhD in Applied Mathematics from the University of Tokyo in 1972, 1974, and 1979, respectively. After serving as professor of computer science at Gunma University, Japan, and Okayama University, Japan, he retired in 2013 and is now professor emeritus of Okayama University. He was a visiting researcher at the University of Maryland, USA; the University of Copenhagen, Denmark; the University of Oxford, UK; INRIA, France; ETH, Switzerland; University of Paris-Est, France; and Linköping University, Sweden. He is the author of K. Kanatani, *Group-Theoretical Methods in Image Understanding* (Springer, 1990); K. Kanatani, *Geometric Computation for Machine Vision* (Oxford University Press, 1993); K. Kanatani, *Statistical Optimization for Geometric Computation* (Elsevier, 1996; reprinted Dover, 2005); K. Kanatani, *Understanding Geometric Algebra* (CRC Press, 2015); K. Kanatani, Y. Sugaya, and Y. Kanazawa, *Ellipse Fitting for Computer Vision* (Morgan & Claypool, 2016); and K. Kanatani, Y. Sugaya, and Y. Kanazawa, *Guide to 3D Vision Computation* (Springer 2016). He received many awards including the best paper awards from IPSJ (1987), IEICE (2005), and PSIVT (2009). He is a fellow of IEICE and IEEE.

Sreekar Karnati is currently a research engineer for GE Global Research. He secured his PhD and Master’s degree in Mechanical and Manufacturing Engineering, respectively, from Missouri University of Science and Technology. He got his Bachelor’s in Manufacturing Science and Engineering from the Indian Institute

of Technology, Kharagpur. As a researcher of LAMP lab at Missouri S&T, he has assisted on several industrial and academic projects. He had effectively contributed on several NASA, Department of Energy, and Navy-sponsored projects. He has extensive experience with additively fabricating and characterizing several aerospace and advanced materials. He developed optimized process methodologies for fabricating flaw-free materials. His work involved fabrication of advanced materials such as functionally graded materials, high entropy alloys, and metal matrix composites. He also developed miniature tensile testing facilities at Missouri S&T. Dr. Karnati has extensive experience with planning, operation, and maintenance of both commercial and research-grade additive manufacturing machines. He has published over ten journal and conference publications on topics of additive manufacturing relating to blown powder and powder bed methodologies.

Vladimir G. Kartashov as born on July 03, 1958. In 1980 he graduated from Kharkiv Institute of Radio Electronics on specialty “Radio engineering,” Kharkiv, Ukraine. In 1990 he received his PhD degree on specialty “Radar and radio navigation.” In 2006 he was awarded the title of professor. He received the DSC (Postdoctoral) degree in Kharkiv National University of Radio Electronics, Ukraine, on specialty “Radio engineering and Television Systems” in 2003. He is the author of 6 monographs and 47 articles indexed in SCOPUS (H-index 4) and has 4 author certificates of the Soviet Union and 25 patents of Ukraine. He is a member of editorial college of several scientific and technical editions of the university. Currently, he is the head of the Department of Media Engineering and Information Radio Electronics Systems of Kharkov National University of Radio Electronics, Ukraine, and director of several doctorates and postdoctoral thesis. He delivered reports at international (IECON 2015, ESARS-ITEC 2016, IECON 2018) and local conferences. He is a member of scientific and technical and specialized councils of Ukraine. He is a member (by a corresponding member) of the Academy of Sciences of the applied radio electronics of Ukraine, Russia, and Byelorussia.

Gabriel Kerekes received his Master’s title in Geodetic Engineering from the Technical University of Civil Engineering of Bucharest, Romania, in 2014. Before graduating, he worked as a technical assistant at Estudio Pereda 4 in Malaga, Spain, for 3 months in 2013. During this time he coordinated and undertook field measurements like leveling and surveying for water works. In the beginning of 2014, he prepared his master thesis in collaboration with the Institute of Engineering Geodesy, University of Stuttgart, Germany. Shortly after graduating, he started working at intermetric GmbH in Limburg an der Lahn, Germany, as a surveying engineer. Almost all of the projects he was involved in implied high-precision measurements with total stations or GNSS instruments, geodetic network adjustments, or CAD planning for railways and infrastructure works. In the spring of 2017 he accepted a position as research associate at the Institute of Engineering Geodesy, University of Stuttgart. Besides supervising practical exercises, seminars, and Master’s theses for undergraduates and graduates, Gabriel Kerekes is conducting research in the field of terrestrial laser scanning for monitoring purposes. As regards

his publication activity, the papers he has written are concentrated on enhanced measurement systems based on robotic total stations.

Marina Kolendovska was born on November 21, 1974. She received her BS and MS degrees (Laurea) in Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, in 1998 and 2004, respectively. She received her PhD degree in Kharkiv National Polytechnic University on specialty “Medical devices, systems and complexes” in 1998. She has been the author of a large number of papers and holds five patents of Ukraine. Since 2004 until the present time, her research works represented in various international congresses. The Kharkiv National University of Radio Electronics did invite Dr. Kolendovska for associate professor of the Department of Media Engineering and Information Radio Electronic Systems in September 2017. She is currently the associate professor of the Department of Media Engineering and Information Radio Electronic Systems, Member of STC, PhD.

Vladimir G. Krasilenko was born on July 20, 1953, Vinnitsa Region, Ukraine. His educational background is as follows: Radio Engineers Diploma, 1975; Candidate of Sciences Degree (PhD), 1988; and Information Systems, Vinnitsa State Technical University. He held engineering positions and was the head of the Department of Research Institutes and Enterprises (1975–1982), he also was PhD student and lecturer assistant (1982–1988); senior, leading scientific researcher; head of Special Design and Technology Bureau of Vinnitsa National Technical University; leading scientist, Enterprise “Injector,” Science Research Institute of Videotechnic (1988–2001); associate professor; and professor of Information Technology Department, Vinnitsa Social Economy Institute of University of Ukraine (2001–2015). He has authored/coauthored more than 400 scientific works, including 188 inventions, about 80 articles in scientific journals and Proceedings of SPIE, 2 chapters published by InTech, and 5 tutorial books. Krasilenko was named Best Young Inventor of Ukraine in 1985 and was a member of SPIE in 1995 and senior member of SPIE in 2012. His research interests include optoelectronic devices and multifunctional logic elements for parallel image processing and for computing; neural networks; multi-valued, continuous matrix logic; recognition; cryptography; information protection; and analog-to-digital transformations.

Alexander A. Lazarev is an assistant professor at the Vinnytsia National Technical University (VNTU), Ukraine. He completed his graduate studies with a PhD from the VNTU in 2003 and his undergraduate studies at the VNTU with a MSc in Electronics Engineering in 1998. He has authored/coauthored more than 250 publications. His research interests include devices and logic elements for image processing and for computing; neural networks; multi-valued, continuous matrix logic; recognition; cryptography; information protection; and analog-to-digital transformations.

Otto Lerke has received his diploma grade in Geodesy and Geoinformatics from the University of Stuttgart in the year 2008. From 2009 to 2013 he served as an engineer at the construction company Klinger and Partner GmbH in Stuttgart, undertaking planning tasks in the landfill sector by the use of GIS and CAD tools. Currently he

is serving as a research associate at the Institute of Engineering Geodesy, University of Stuttgart. His research topics are machine guidance and multisensory systems. His primary interests include the integration of tachymeter measuring techniques into control closed-loop systems. His scientific activities include several technical publications, among which one was honored with the best paper award during the International Machine Guidance Conference (MCG) in the year 2018.

Lars Lindner was born on July 20, 1981, in Dresden, Germany. He received his MS degree in Mechatronics Engineering from the Technical University Dresden University in January 2009. He was working as graduate assistant during his studies at the Fraunhofer Institute for Integrated Circuits EAS in Dresden and also finished his Master's thesis there, with the title "Support of Rapid Control Prototyping of machine controls by transformation of hierarchical state machine graphs into IEC 1131 code of Programmable Logic Controllers." After finishing his career, he moved to Mexico and started teaching engineering classes at different universities in Mexicali, currently at the Engineering Faculty of the Autonomous University of Baja California campus Mexicali, where he received his PhD on January 2017. He completed his research stay at the Fraunhofer Institute IIS-EAS in Dresden, taking part in the project "Development of a software driver for a 2D laser scanner" in June/July 2015. Its academic products include various original research articles, published in specialized international journals with impact-factor, articles in national and international congresses, and various book chapters. In the year 2017 his research article "Mobile robot vision system using continuous laser scanning for industrial application" of the international journal *Industrial Robot* of the editorial Emerald received the distinction "Outstanding Paper." In September 2017, he was appointed as a Level 1 national researcher by the National System of Researchers CONACYT for the period 2018–2020. He is currently working full-time at the Engineering Institute of Autonomous University of Baja California for the Department of Applied Physics and is an active member of the research group "Optoelectronics and Automatic Measurements."

Frank F. Liou is the Michael and Joyce Bytnar professor of the Mechanical Engineering Department, Missouri University of Science and Technology. He has served as the director of the Manufacturing Engineering Program at Missouri S&T since the year 1999. He has published a book on rapid prototyping and engineering applications, along with over 300 technical papers. Dr. Liou's research has been focusing on additive manufacturing, including hybrid additive and subtractive process integration, path planning, multiscale multiphysics process modeling, and AM process monitoring and control. His research has been funded by AFRL, DOE, NASA, NAVAIR, NSF, and many industrial partners. Several related papers have been awarded as the best papers in various conferences (in years 1997, 2005, 2009, 2010, 2015), including some published in *Nature: Scientific Report*. The related work was selected as the most important innovations at Missouri S&T in the years 2013 and 2015. Dr. Liou has received several teaching, research, and service awards. Dr. Liou is a fellow of the American Society of Mechanical Engineers (ASME).

Ambrocio Loreda-Flores received his Bachelor's degree in Electrical Engineering from the Technological Institute of San Luis Potosí, Mexico, in 1999, and Master's and PhD degrees in Electrical Engineering from the Autonomous University of San Luis Potosí (UASLP), Mexico, in 2002, and 2007, respectively. In that year, he joined the College of Engineering at UASLP as a research assistant. In 2009 he joined the UASLP campus in Matehuala, Mexico, where he is currently a professor of Mechatronics Engineering. He has published more than 20 refereed papers in scientific journals and congresses. His research and professional interests include robotics, programming, and electronic design.

Fanzhe Lyu is a third-year Bachelor of Science Computer Engineering candidate in the School of Electrical and Computer Engineering at Georgia Institute of Technology. He is currently an undergraduate research assistant in the Intelligent Vision and Automation Lab (IVALab) under guidance from Dr. Patricio Vela. His research interest includes robot navigation under dynamic environments using perception space techniques and obstacle memorization under sensor limitation. He has also interned as a software engineer with NCR Corporation, where he gained experience in mobile apps development and distributed systems design. As a student in the Computer Engineering program, he also holds backgrounds in signal processing, computer systems and architecture, computer vision, and machine learning.

Vladimir Genrievich Meledin is a Doctor of Technical Science and chief researcher of Kutateladze Institute of Thermophysics of the Siberian Branch of the Russian Academy of Sciences. He is a winner of the Russian Federation Governmental Prize in the field of science and technology. Meledin V.G. is an expert in the field of complex technical systems and information control for optoelectronic diagnostics and monitoring of kinematics and the structure of gas and condensed matter. He is the author and coauthor of 300 scientific papers, including 3 monographs, and 65 patents. The main scientific results are associated with the creation of fundamental new methods of semiconductor laser Doppler anemometry of multiphase flows with natural light diffusers, which were the basis of a wide class of measuring information and control systems. He created new methods of forming, receiving, and processing of complex optoelectronic and acousto-optic signals and images in the conditions of photon limitation and methods of their processing and methods of dynamic information monitoring of three-dimensional geometry of objects. Innovative import-substituting systems have been created for science, defense, and industry, implemented at the largest enterprises of energy and transport, metallurgy, nuclear, engineering, oil, and other industries, which ensured the proven multibillion dollar economic effect and significantly increased the efficiency and safety in the real sector of the Russian economy.

Marco O. Mendoza-Gutierrez received his BE degree in Communications and Electronics Engineering from the University of Colima, Mexico, in 2003; MSc degree in Electronics from the Autonomous University of Puebla, Mexico, in 2006; and PhD degree in Electrical Engineering from the Autonomous University of San Luis Potosí, Mexico, in 2011. He was an associate professor at the University of

Sonora, Mexico, from 2011 to 2012. In 2014 he joined the College of Sciences at the Autonomous University of San Luis Potosí, Mexico, where he is currently a professor of biomedical engineering. He has published more than 50 refereed papers in scientific journals and congresses. His current research interests include vision-based control, robot control, and biorobotics.

Paolo Mercorelli received his MS (Laurea) degree in Electronic Engineering from the University of Florence, Florence, Italy, in 1992, and PhD degree in Systems Engineering from the University of Bologna, Bologna, Italy, in 1998. In 1997, he was a visiting researcher for 1 year in the Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, USA. From 1998 to 2001, he was a postdoctoral researcher with Asea Brown Boveri, Heidelberg, Germany. From 2002 to 2005, he was a senior researcher with the Institute of Automation and Informatics, Wernigerode, Germany, where he was the leader of the control group. From 2005 to 2011, he was an associate professor of Process Informatics with Ostfalia University of Applied Sciences, Wolfsburg, Germany. In 2010 he received the call from the German University in Cairo (Egypt) for a full professorship (chair) in Mechatronics which he declined. In 2011 he was a visiting professor at Villanova University, Philadelphia, USA. Since 2012 he has been a full professor (chair) of Control and Drive Systems at the Institute of Product and Process Innovation, Leuphana University of Lueneburg, Lueneburg, Germany. His current research interests include mechatronics, automatic control, signal processing, wavelets, sensorless control, Kalman filter, camless control, knock control, lambda control, and robotics.

Fabián N. Murrieta-Rico received BS and MSc titles of the Mexicali Institute of Technology (ITM) in 2004 and 2013, respectively. In 2017 he received his PhD in Materials Physics at the Center for Scientific Research and Higher Education of Ensenada (CICESE). He has worked as an automation engineer, systems designer, and university professor. Currently, he is a postdoctoral fellow at Facultad de Ingeniería, Arquitectura y Diseño from Universidad Autónoma de Baja California (UABC). His research has been published in different journals and presented at international conferences since 2009. He has served as a reviewer for different journals, some of them include *IEEE Transactions on Industrial Electronics*, *IEEE Transactions on Instrumentation and Measurement*, and *Sensor Review*. His research interests are focused on the field of time and frequency metrology, the design of wireless sensor networks, automated systems, and highly chemical detectors. He is currently involved in the development of new frequency measurement systems and highly sensitive sensors for the detection of chemical compounds.

Fusaomi Nagata received the BE degree from the Department of Electronic Engineering at Kyushu Institute of Technology in 1985 and the DE degree from the Faculty of Engineering Systems and Technology at Saga University in 1999. He was a research engineer with Kyushu Matsushita Electric Co. from 1985 to 1988 and a special researcher with Fukuoka Industrial Technology Center from 1988 to 2006. He is currently a professor at the Department of Mechanical Engineering, Faculty of Engineering, Tokyo University of Science, Yamaguchi, Japan, and also a

dean of the Faculty of Engineering. His research interests include deep convolutional neural networks for visual inspection of resin molded articles and intelligent control for industrial robots and its application to machining process (e.g., robot sander, mold polishing robot, desktop NC machine tool with compliance control capability, machining robot with robotic CAM system, and 3D printer-like data interface for machining robot development for wood material, aluminum PET bottle blow mold, LED lens mold, foamed polystyrene, and so on).

Diana V. Nikitovich is a dispatcher of Faculty for Radio Engineering, Telecommunication, and Electronic Instrument Engineering at the Vinnytsia National Technical University (VNTU), Ukraine. She graduated from the magistracy from Vinnitsa Social and Economic Institute in 2012 and her MSc and Bachelor's degrees in document management and information activities. She is the author/coauthor of about 50 publications. Her research interests include neural networks, devices, and logic elements for image processing, continuous matrix logic, recognition, information protection, cryptography, and analog-to-digital transformations.

Connor Nogales is a student researcher in the Robotics Vision Lab of Northwest Nazarene University. Connor is currently a senior taking up Bachelor of Science in Engineering (Electrical Engineering concentration).

Hiroaki Ochi received his BE degree and ME degree from the Department of Intelligent Mechanical Engineering, Faculty of Engineering, at Fukuoka Institute of Technology in 2011 and 2013, and his DE degree from the Department of Material Science and Production Engineering, Faculty of Engineering, at Fukuoka Institute of Technology in 2016. He is currently an assistant professor at the Department of Mechanical Engineering, Faculty of Engineering, Sanyo-Onoda City University, Japan. His research interests include controls and characteristic analyses of redundant driven systems such as musculoskeletal structural robots and wire-driven robots. In the research of musculoskeletal structural robots, he studied a stability analysis of the internal force generating between the muscles, a feedforward control method utilizing the stability, and an analysis of structural conditions that stabilize the internal force. In the other research, he examined an error evaluation of approximated inverse kinematics on a parallel wire-driven robot and a determination method of balancing tensile force on a tensegrity structural robot.

Akimasa Otsuka received his BE degree from the Department of Mechanical Engineering at Tokyo Institute of Technology in 2002. He received the ME degree in 2004 and the DE degree in 2009 from the Department of Mechanical and Environmental Informatics at Tokyo Institute of Technology. He was assistant professor from 2011 to 2015 at Tokyo University of Science, Yamaguchi, Japan. He was junior associate professor at the Department of Mechanical Engineering, Faculty of Engineering, Tokyo University of Science, Yamaguchi, Japan. He is currently junior associate professor at the Department of Mechanical Engineering, Faculty of Engineering, Sanyo-Onoda City University, Japan. His research interest includes quality engineering, industrial engineering, tolerancing, and geometrical modeling in mechanical design and swarm robotics. His skills are software development of

mass production simulation, statistical simulation, optimization technique, computer graphics, MATLAB, multi-robotics simulation, neural network, statistical analysis, measurement of surface form and roughness, and so on.

Kannappan Palaniappan is a professor in the Electrical Engineering and Computer Science Department. He has received several notable awards, including the National Academies Jefferson Science Fellowship (first in Missouri), the NASA Public Service Medal for pioneering contributions to (Big Data) scientific visualization of petabyte-sized archives, the Air Force Summer Faculty Fellowship, the Boeing Welliver Summer Faculty Fellowship, and MU's William T. Kemper Fellowship for Teaching Excellence. At NASA's Goddard Space Flight Center, he co-founded the Visualization and Analysis Lab that has produced a number of spectacular digital Earth visualizations used by search engines (Blue Marble), museums, magazines, and broadcast television. He is co-inventor of the Interactive Image Spreadsheet for handling large multispectral imagery, and he developed the first massively parallel semifluid cloud motion analysis algorithm using geostationary satellite imagery. In 2014, his team won first place at the IEEE Computer Vision and Pattern Recognition (CVPR) Change Detection Workshop video analytics challenge. In 2015, the team was a finalist in the CVPR Video Object Tracking Challenge, and in 2016, the team won the best paper award at the CVPR Automatic Traffic Surveillance Workshop and also was selected as a finalist for a best student paper award at the IEEE Engineering in Medicine and Biology Society conference (EMBC 2016). He has several US patents, including one for moving object detection using the flux tensor split Gaussian model and the other for fast bundle adjustment to accurately estimate the pose of airborne camera sensor systems. Research projects have been funded by the National Institutes of Health, the Air Force Research Laboratory, Army Research Laboratory, NASA, the National Science Foundation, and others. His current, multidisciplinary interests in computer vision, high-performance computing, data science, and biomedical image analysis range across orders of scale from subcellular microscopy at the molecular level to aerial and satellite remote sensing imaging at the macro level.

Ajay K. Pandey is a senior lecturer in Robotics and Autonomous Systems at the School of Electrical Engineering and Computer Science at Queensland University of Technology (QUT). He obtained a PhD in Physics on Organic Optoelectronics with *mention tres honourable* from the University of Angers, France. He holds a Master of Technology (MTech) degree in Laser Science and Applications and a Master of Science (MSc) degree in Physics with specialization in atomic, molecular, and laser spectroscopy. His research interest has the interdisciplinary mix of photonics, chemical physics, molecular electronics, neuroscience, and robotics. He has been recipient of prestigious fellowship awards including QUT-Vice Chancellor's Senior Research Fellowship and Australian Renewable Energy Agency (ARENA) Research Fellowship. He has published more than 50 research articles on advanced topics linked to photo detection, energy conversion, and energy up-converted electroluminescence. He leads an interdisciplinary research group at QUT that specializes in technological implementation of advanced materials for applications in

neuroscience, intelligent bionics, soft robotics, energy conversion, and night vision. He also serves as an editorial board member of *Scientific Reports*, a Springer Nature Group journal.

Oleksandr Poliarus was born on February 18, 1950, in Gadyach town of Poltava region (Ukraine). From 1967 to 1973 he studied at the Moscow Higher Technical School named after M. E. Bauman, after which he worked as an engineer at a research institute for 1 year. From 1974 to 1999 he served in the Armed Forces of the USSR and then Ukraine. In 1980 he graduated from the Military Radio Engineering Academy of Air Defense in Kharkiv and in 1985 defended his PhD thesis. He was in teaching positions of the Academy and since 1996 the head of the Department of Antenna-Feeder Devices. In 1994 he defended his doctoral thesis, and in 1999 he retired from the Armed Forces of Ukraine. Since September 2007, he is the head of the Department of Metrology and Life Safety of the Kharkiv National Automobile and Highway University. He is married, has two sons, and lives in Kharkiv, Ukraine. His research interests include signal processing, remote measurements, identification of nonlinear inertial systems, and inverse problems of measurement.

Yevhen Poliakov was born in 1985 in Pavlograd, Dnipropetrovsk region, Ukraine. In 2003 he graduated from Pavlograd secondary school number 9 and entered the Kharkiv National Automobile and Highway University. In 2008 he graduated from Kharkiv National Automobile and Highway University and received a full higher education in the specialty “Automated Control of Technological Processes.” From 2008 to 2012 he was a graduate student of the Department of Metrology and Life Safety of Kharkiv National Automobile and Highway University. From 2012 to present, he works at the Kharkiv National Automobile and Highway University as associate professor of the Department of Metrology and Life Safety. In 2014 he defended his PhD thesis on the theme “Improvement of methods of sensors dynamic errors decrease” (specialty 05.01.02 “Standardization, certification and metrological support”). He is married, has a daughter, and lives in Kharkiv, Ukraine. His research interests include inverse problems of measurement, estimation of parameters of quick changing processes, identification of dynamic characteristics of technical objects, and correction of dynamic measurement errors.

Denis Pomorski received his PhD degree from the University of Lille in Automatic Control and Industrial Computing in December 1991. From October 1992 to August 2001, he was a lecturer at the same university. In December 1999, he obtained the “Habilitation à Diriger des Recherches” Degree (accreditation to supervise research works) from the University of Lille in Physical Sciences. During the academic year 1999–2000, he was in the Delegation Position at CNRS. Since September 2001, he is a full professor within the CRISAL Laboratory, UMR 9189. His research interests are focused on fault-tolerant multi-sensor data fusion, detection, information theory, Shannon entropy, and Bayesian filter with applications in collaborative mobile robotics.

Miguel A. Ponce-Camacho is a full-time professor at CETYS University in Mexicali, Mexico, and holds a Bachelor’s degree in Physics Engineering from

Monterrey Institute of Technology (Mexico); a Master of Science from the Center for Scientific Research in Ensenada, Mexico; and a PhD in Nanotechnology from Universidad Autónoma de Baja California, Mexico. Dr. Ponce currently teaches the subject of Innovation and Development for graduate students at the Master of Engineering and Innovation and Industrial Electronics and Electromagnetic Theory for undergraduate students for the School of Engineering at CETYS University. Some of his recent works include titles such as “Fabrication of an optical prototype for light-matter experiments,” “Mathematical Modeling for Design Thinking Innovation Method based on Markov Chain Theory,” “High-performance quantum key distribution prototype system using a commercial off-the-shelf solution: experimental and emulation demonstrations,” and “A Novel Space Systems Management Methodology Based on Shortcomings and Strengths of Conventional System Engineering Tools used in a Design Thinking Framework.”

Luis Carlos Básaca-Preciado was born on October 25, 1985. He received a BS degree in Cybernetics and Electronics Engineering from CETYS University, Mexicali, Baja California, Mexico, in 2007. In 2013 he obtained his PhD in Optoelectronics from the Institute of Engineering of the Autonomous University of Baja California, México. As of 2014 he became a full-time professor, teaching automation, robotics, and instrumentation at the Graduate School of Engineering and in the Electronics and Mechatronics majors at the School of Engineering of CETYS University, Mexicali Campus. From 2015 to 2018 he coordinated the Program of Cybernetics Electronics Engineering Major. He currently focuses on research in 3D vision systems, intelligent transportation by autonomous vehicles, autonomous UAVs, underwater ROVs, and VR/AR and also participates in applied engineering projects linked to the industry. He has also been a mentor to student robotics teams since 2012 to promote science and engineering among young people. He has written 3 book chapters, 9 journal articles, and 14 conference documents, including international conferences such as IEEE ISIE, IECON, EEEIC, ROC&C, PAHCE, and IPC, in the USA, Italy, Brazil, and Mexico, where he holds a patent for a 3D vision system.

Juan Manuel Ramírez-Cortes received his BSc degree from the National Polytechnic Institute, Mexico; MSc degree from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico; and PhD from Texas Tech University, all in Electrical Engineering. From 1982 to 2007, he was with University of the Americas, Puebla, Mexico, in several academic and administrative roles as faculty member, department chair, and dean of the Engineering School. Since 2007 he has been at INAOE, Puebla, Mexico, where he has served as Electronics Department chair, research director, and currently titular researcher at the Electronics Department. Dr. Ramírez is an appointed member of the Administrative Committee of IEEE.

Luis Roberto Ramírez-Hernández received the Engineering degree in Mechatronic in the Facultad de Ingeniería of the Universidad Autónoma de Baja California, México, in 2010, and the MC degree in the Instituto de Ingeniería of the Universidad Autónoma de Baja California, México, in 2015. Regarding his MC degree, he works in the area of artificial intelligence creating a matrix computation library for graphics

processing unit (GPU cards). Actually, he is a professor of the Universidad Autónoma de Baja California, Mexico, and he is a full-time student on his second year of PhD in the Facultad de Ingeniería of the Universidad Autónoma de Baja California in the Electronic and Instrumentation Department. His current research interest is applications of instrumentation and control of cameras and robots manipulators.

Luis A. Raygoza received his BE degree in Electronic and Digital Communications Engineering from the Autonomous University of Aguascalientes (UAA), Mexico, in 2002; MSc degree in Electrical Engineering from the Technological Institute of Aguascalientes, Mexico, in 2004; and the PhD degree in Electrical Engineering from the Autonomous University of San Luis Potosí, Mexico, in 2010. Currently, he is a professor and researcher at UAA. He has been dedicated to developing applied research projects, linked to the Mexican industry through the National Council of Science and Technology (CONACYT) since 2014. In 2017 he joined INFOTEC Aguascalientes, Mexico, as a lecturer. His most important developments are related to the design of electronic platforms for automatic production control machines. His research interests include 3D modeling, multiple camera environments, mobile robotics, and electronic embedded systems for automatic control.

Miguel Reyes-García was born on September 29, 1989, in Hidalgo, Mexico. He received his BS degree in mechatronics engineering from the Polytechnic University of Baja California (UPBC), Mexicali, Baja California, Mexico, in 2014. Currently he studies a MS degree in Engineering Physics area at the Engineering Institute of Autonomous University of Baja California (UABC), Mexico, with the topic “Theoretical method to reduce the positioning error in a scanning laser system, using an embedded digital controller and direct current motors” working with the development prototype named “Technical Vision System,” which uses laser triangulation to determine 3D coordinates of any object under observation and developed by members of the Applied Physics Department of Engineering Institute of UABC. He published a paper on international conferences.

Moisés Rivas-López was born on June 1, 1960. He received his BS and MS degrees in Autonomous University of Baja California, México, in 1985 and 1991, respectively. He received his PhD degree in the same university, on specialty “Optical Scanning for Structural Health Monitoring,” in 2010. He was the editor of 2 books and has written 5 book chapters and 48 journal and proceedings conference papers. Since 1992 until the present time, he has presented different works in several International Congresses of IEEE, ICROS, SICE, and AMMAC in the USA, England, Japan, Turkey, and Mexico. In 1997–2005, he was dean of the Engineering Institute of Autonomous University Baja California; also in 2006–2010, he was rector of the Polytechnic University of Baja California. Since 2013 till the present time, he is a member of the National Researcher System and now is the head of the Physical Engineering Department, Engineering Institute, Autonomous University of Baja California, Mexico.

Flavio Roberti was born in Buenos Aires, Argentina, in 1978. He graduated in Engineering from the Universidad Nacional de San Juan, Argentina, in 2004,

and obtained a PhD degree in Control Systems Engineering from the Universidad Nacional de San Juan, Argentina, in 2009. He is currently an associate professor at the Instituto de Automática (INAUT), Universidad Nacional de San Juan, and associate researcher of the National Council for Scientific and Technical Research (CONICET, Argentina). His research interests are robotics, wheeled mobile robots, mobile manipulators, visual servoing, passivity-based visual control, and assistive robotics.

Julio Cesar Rodríguez-Quiñonez was born in October 1985. He received his BS degree in CETYS, Mexico, in 2007. He received his PhD degree from Baja California Autonomous University, México, in 2013. He is currently head of the Department of Electrical Engineering in Engineering Faculty, Autonomous University of Baja California. Since 2016, he is a senior member of IEEE. He is involved in the development of optical scanning prototype in the Applied Physics Department and research leader in the development of a new stereo vision system prototype. He holds 2 patents referred to dynamic triangulation method and has been editor of 2 books, has written over 50 papers and 8 book chapters, and has been a reviewer for *IEEE Sensors Journal*, *Optics and Lasers in Engineering*, *IEEE Transaction on Mechatronics*, and *Neural Computing and Applications* of Springer; he participated as a reviewer and session chair of IEEE ISIE conferences in 2014 (Turkey), 2015 (Brazil), 2016 (USA), and 2017 (UK) and IECON 2018 (USA). His current research interests include automated metrology, stereo vision systems, control systems, robot navigation, and 3D laser scanners.

Veronica A. Rojas-Mendizabal earned her Doctor's degree in Electronics and Telecommunications from CICESE Research Center in Baja California, Mexico. She earned her Master's degree in Telecommunications and Telematics from the Catholic University of Bolivia (UCB) in La Paz, Bolivia. She took postgraduate-level courses in Satellite Communications with the Regional Center for Education in Space Science and Technology for Latin America and the Caribbean (CRECTEALC), a United Nations space educational center in Puebla, Mexico. She is currently a professor and researcher at CETYS University in Mexicali, Baja California, Mexico.

Oscar A. Rosete-Beas was born in Tijuana, Baja California, México, on July 7, 1992, and he is an adjunct professor at CETYS University, Mexicali Campus. He received his Bachelor's degree in Mechatronics Engineering from Universidad Autónoma de Baja California Campus Mexicali in 2015 and he is currently working toward his Master's degree at CETYS. He began working in the manufacturing industry in 2013. During the past years, his works have focused upon developing robotics and industrial automation systems while teaching subjects in the Cybernetics and Mechatronics engineering specialty at CETYS Universidad Campus Mexicali. Mr. Rosete has collaborated on peer-reviewed articles on robotics and industrial automation and currently participates in research projects, such as the construction of an intelligent transportation scheme for autonomous vehicle for use at the institution's facilities and IoT Implementation for Predictive Maintenance including Lambda Architecture and OPC UA.

Marta Rostkowska graduated from Poznań University of Technology in 2013 receiving the BSc and MSc in Automatic Control and Robotics. She is currently a PhD student at the same university. Her current research interests include applications of computer vision, software engineering for embedded systems, and mobile robotics.

Kevin B. Ruiz-López is currently pursuing a Master's degree in Innovation and Engineering at CETYS Universidad. In 2016, he graduated with honors from the same university and won an academic medal along the way. In the time between his graduation and starting his Master's, he went on an exchange trip with AISEC to Athens, Greece, where he worked as an engineer for a startup company in the tourism sector. While in Greece, he worked as a backend developer and gained experience with Internet of things (IoT). Kevin is passionate about developing technologies that contribute to society in the healthcare sector and to improve their quality of life. His dream is to be able to make comfortable prosthesis for people in need that feel just like an extension of themselves, so that every person can live with the least restrictions possible in their day-to-day life. Kevin is a medal-winning archer from Sonora, Mexico, and you can usually find him working on projects at his university.

Javier E. Sanchez-Galan received his BSc degree in Computer Systems Engineering at the Universidad Tecnológica de Panama (UTP) in 2006. In 2007 he was awarded a full scholarship from the Panamanian government (IFARHU-SENACYT) to pursue postgraduate studies at McGill University, in Montreal, Canada. In 2010, he completed his MSc in Computer Science (Bioinformatics) at the McGill Centre for Bioinformatics focusing in Computational Gene Regulation. In 2015 he finished his PhD in Experimental Medicine at McGill's Faculty of Medicine and his work on chemo-informatics applied to maternal-fetal health. Since 2013, he works as a research scientist at the UTP in the Production and Agro-industrial Research Center, where he is a coordinator of the Research Group in Biotechnology, Bioinformatics, and Systems/Synthetic Biology (GIBBS). He also works as a professor in the Department of Computer Systems Engineering at UTP at the graduate and undergraduate levels. Since February 2015, he also works as an adjunct researcher at the Institute for Scientific Research and High Technology Services AIP (INDICASAT-AIP). His research interests are focused on the development of innovative approaches applying computational data analysis of massive data sets coming from biological, medical, environmental, and agricultural fields.

Jorge A. Sarapura was born in Orán, Salta, Argentina. He graduated with honors with the degree of Electronic Engineer of the Universidad Nacional de Tucumán, Argentina, in 2006. He received his Master's degree in Control Systems Engineering from the Universidad Nacional de San Juan (UNSJ), Argentina, in 2013. Since 2006, he has been with the Instituto de Automática (INAUT) of the Universidad Nacional de San Juan, where he currently works as a researcher and assistant professor of Analog Electronic, and also finished his studies to access a PhD degree in Control Systems Engineering. From 2011 to 2014, he also worked in the Department of Electromechanics of the Universidad Nacional de San Juan as an assistant professor of Electrical Machines and Electrical Measurements. His research interests include nonlinear and adaptive control, robotics, precision agriculture, control based on

artificial vision, and systems identification with application to manipulator robots, unmanned aerial vehicles, and mobile robots.

Volker Schwieger received his Dipl.-Ing in Geodesy from the University of Hannover, today Leibniz University, Germany, in 1989. After a period of approximately 2 years of civilian service, he was engaged as a research associate at the Geodetic Institute of the University of Stuttgart. In 1998 he did his Dr.-Ing at the same university on “An elementary error model for GPS monitoring measurements.” In 2000 he changed as a postdoc to the German Research Centre for Geoscience working on orbit determination of GPS and low orbit satellites. In 2002 he accepted a postdoctoral position at the Institute of Applications of Geodesy to Engineering at the University of Stuttgart. In 2003 he was appointed as head of the Department of “Measurement Techniques” within the institute, and in 2004 he habilitated on the topic “Nonlinear sensitivity analysis taking moving objects as examples.” In 2010 he was appointed as professor and head of the Institute of Applications of Geodesy to Engineering at the University of Stuttgart, which he renamed into Institute of Engineering Geodesy (IIGS). From 2010 to 2015 he was the elected spokesperson of the Center for Transportation Research at the University of Stuttgart (FOVUS). Since 2011 he acts as the head of working group III “Measurement Methods and Systems” of the German Association for Surveying (DVW), and in 2015 he was appointed as chair of the International Federation of Surveyors Commission 5 “Positioning and Measurement.” Additionally, since 2017 he serves as the dean of The Faculty of Aerospace Engineering and Geodesy. He has published more than 170 papers that cover the complete field of engineering geodesy.

José María Sebastián was born in Madrid, Spain, in 1959. He received his BS degree in Electrical Engineering, MS degree in Control Engineering, and PhD degree in Computer Vision from the Universidad Politécnica de Madrid, Madrid, in 1979, 1982, and 1987, respectively. He is currently a teacher in the Control Department, Industrial Engineering School, Universidad Politécnica de Madrid, where he has been since 1982. He teaches courses in computer vision and control engineering. His research interests include distance learning, teleoperation, and computer vision. Dr. Sebastian is a member of the International Society for Optical Engineers and the International Federation of Automatic Control.

Guna Seetharaman was appointed as senior scientist (ST) for Advanced Computing Concepts and chief scientist for Computation, in June 2015. He joined the NRL Center for Computational Science after a successful tenure (2008–2015) as a principal engineer of Computing Architectures and Video Exploitation at the Information Directorate of Air Force Research Laboratory, Rome, NY. He was elevated to the rank of fellow of IEEE, in 2015, for his contributions to high-performance computer vision algorithms for airborne applications. As the chief scientist of Center for Computational Science, he leads high impact research on high performance computing, novel architectures, high throughput low-latency networked computing, video analytics, autonomy, and C4ISR areas. He held research active academic tenured positions at the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base,

Dayton, OH (2003–2008), and at the Center for Advanced Computer Studies (1988–2003), University of Louisiana at Lafayette, LA. He earned his PhD, in Electrical and Computer Engineering from the University of Miami, FL, in 1988; MTech degree in Electrical Engineering from The Indian Institute of Technology, Madras, in 1982; and his BEng degree in Electronics and Communication Engineering from the University of Madras, in 1980. His recent works have focused on high-performance computing for video exploitation: computer vision, machine learning, content-based image retrieval, persistent surveillance, and computational science and engineering. He was a member of the AFIT-based core team for demonstrating and transitioning a wide area persistent imaging and surveillance system known as Angel Fire. At AFRL, he led a program C4ISR Enterprise to the Edge (CETE) aimed at pushing processing closer to sensors and successfully demonstrated high-performance computational algorithms for airborne video analysis: for 3D mapping, tracking, and exploitation. He also started a new program named content and context aware trusted routers (C2TR) as a robust mechanism for agile high-performance computing across large networks. His team won the best algorithm award in the IEEE CVPR 2014 Video Change Detection Challenge. They won the best paper award at the IEEE Workshop on Automatic Traffic Surveillance, IEEE CVPR-2016. His team is among the top for high-performance video trackers in the IEEE ICCV 2015 competition. He co-founded Team CajunBot—a participant in DARPA Grand Challenge—featuring two unmanned vehicles. He led the LiDAR data processing and obstacle detection efforts in Team CajunBot, demonstrated in 2005 and 2007 DARPA Grand Challenges. He has published more than 180 peer-reviewed articles in computer vision, low-altitude aerial imagery, parallel computing, packet routing, VLSI-signal processing, 3D displays, nanotechnology, micro-optics, and 3D video analysis. He guest-edited *IEEE COMPUTER* special issue devoted to unmanned intelligent autonomous vehicles, Dec 2006. He also guest-edited a special issue of the *EURASIP Journal of Embedded Systems* in the topics of intelligent vehicles. He is an associate editor of the ACM Computing Surveys. He has served as the general chair of IEEE Workshop on Computer Architecture for Machine Perception 2003 and co-chaired the technical program committee of IEEE AIPR2014. He is a member of the academic honor societies Tau Beta Pi, Eta Kappa Nu, Upsilon Pi Epsilon, and Phi Beta Delta. He served as the elected chairman of IEEE Mohawk Valley Section, Region 1. He is also a Paul Harris Fellow of the Rotary International.

Oleg Sergiyenko was born on February 9, 1969. He received his BS and MS degrees (Laurea) in Kharkiv National University of Automobiles and Highways, Kharkiv, Ukraine, in 1991 and 1993, respectively. He received his PhD degree in Kharkiv National Polytechnic University on specialty “Tools and methods of non-destructive control” in 1997 and DSc (Postdoctoral) degree in Kharkiv National University of Radioelectronics in 2018. He has been the author of 1 book and editor of 5 books and has written 23 book chapters and 111 papers indexed in SCOPUS (h-index 12) and holds 2 patents of Ukraine and 1 patent of Mexico. Since 1994 until the present time, his research works represented in several International Congresses of IEEE, ICROS, SICE, IMEKO, etc. The Engineering Institute of

Baja California Autonomous University did invite Dr. Sergiyenko for a researcher position in December 2004. He is currently the head of Applied Physics Department of Engineering Institute of Baja California Autonomous University, Mexico, and director of several Master's and Doctorate thesis. He was a member of Program Committees of various international and local conferences, participating as session chair of IEEE ISIE and IECON conferences in 2014–2019 (ISIE2014, IECON2014, ISIE2015, IECON2016, ISIE2017, IECON2018, ISIE2019). He is a member of Scientific Council on Electric Specialties in the Autonomous University of Baja California and Academy of Engineering. He is a member (academician) of the Academy of Applied Radioelectronics of Ukraine, Russia, and Belorussia. He did receive the award of “Best session presentation” in IECON2014 in Dallas, USA, and IECON2016 in Florence, Italy. He did receive as coauthor the award of “Outstanding Paper in the 2017 Emerald Literati Network Awards for Excellence.”

Ali Shahnewaz obtained an MSc in Computer Engineering from Politecnico Di Milano, Italy. He is a current PhD student at the Queensland University of Technology, Brisbane, Australia, where he is conducting research on intelligent imaging systems for robotic vision at system on chip (SoC) as well as network on chip (NoC) level. He is passionate about problem formulation and solution space exploration and has published five technical papers. Robust robotic vision, 3D reconstruction, data fusion of homogeneous and heterogeneous modality of images, and automatic segmentation are the main focus areas of his current research.

Piotr Skrzypczyński received his PhD and DSc degrees in Robotics from Poznań University of Technology (PUT) in 1997 and 2007, respectively. Since 2010 he is an associate professor at the Institute of Control, Robotics, and Information Engineering (ICRIE) of PUT and head of the Mobile Robots Laboratory at ICRIE. His current research interests include autonomous mobile robots, simultaneous localization and mapping, multi-sensor fusion, and artificial intelligence in robotics.

Justin S. Smith is a PhD candidate in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. He is advised by Dr. Patricio Vela, director of the Intelligent Vision and Automation Laboratory (IVALab). His research interests are centered on autonomous navigation by mobile robots in 3D environments. His focus is on utilizing perception space and deep learning to improve navigation on robots with limited computational resources. His background includes computer architectures, control theory, computer vision, and machine learning. He has interned with the Air Force Research Lab, taught undergraduate classes on circuit fundamentals, and mentored students on robotics projects.

Oleksandr Sotnikov was born on June 12, 1958. In 1980 he graduated from Kharkov higher military command school of rocket troops named by Marshal of Soviet Union Krylov N.I. on specialty radio engineering systems of complexes, Kharkiv, Ukraine. In 1986 he received his PhD degree on Specialty Armament and Military Technique. In 2014 he was awarded the title of professor. He received the DSc (Postdoctoral) degree in Ivan Kozhedub Kharkiv National Air Force University, Ukraine, on specialty “armament and military technique” in 2007. He is the author

of 4 monographs, 180 reasons, and index in SCOPUS (H-index 6) and has 4 author certificates of the Soviet Union and 25 patents of Ukraine. He is a member of the editorial college of several of scientific and technical editions of the university. Currently, he is the leading research worker of Ivan Kozhedub Kharkiv National Air Force University, Ukraine, and director of several doctorates and Postdoctoral thesis. He delivered reports at international (MSMW'2004, UWBUSIS 2004, MRF 2005, ЕЛIT 2016) and local conferences. He is a member of scientific and technical and specialized councils of Ukraine. He is a member (by a corresponding member) of the Academy of Sciences of the applied radio electronics of Ukraine, Russia, and Byelorussia.

Oleg Starostenko received his BSc and MSc degrees in Computer science from Lviv State University of Ukraine in 1982 and PhD degree in Mathematics and Physics from the Autonomous University of Puebla, Mexico, in 1996. He is currently full-time professor in the Department of Computing, Electronics, and Mechatronics at the Universidad de las Americas Puebla, Mexico. He has authored more than 215 research articles in several refereed journals, books, and conference proceedings. His current research fields are the access, retrieval, transmitting, and processing of multimedia information in distributed environments. He belongs to the Mexican National System of Researchers (Level I).

Alaa Taleb received her Bachelor's degree in Electrical Engineering from Beirut Arab University, Lebanon, in 2015. In 2016 she obtained two Master's degrees: MSc in Technology of Medical and Industrial Systems from Lebanese University, Lebanon, and Master SMaRT from University of Lille, France. Her research work was supervised by Kamel Haddadi from the IEMN laboratory and Denis Pomorski from the CRIStAL laboratory. In addition to her Master's degrees, Alaa Taleb has completed various internship projects in the field of energy in several companies in Lebanon. She is currently a technical engineer at Jung Middle East in Dubai since 2017.

Oleksandr Tymochko was born on August 12, 1964. In 1985, he graduated from the Kharkiv Higher Military Aviation School of Radio Electronics, Kharkiv, Ukraine, with a degree in automated control systems. He received his PhD degree in Kharkiv Military University, Ukraine, on specialty "Military cybernetics, computer science, systems analysis, modeling systems and combat operations" in 1996, and DSc (Postdoctoral) degree in Ivan Kozhedub Kharkiv National Air Force University, Ukraine, on specialty "Navigation and traffic control" in 2013. In 2014 he was awarded the title of professor. He is the author of 4 monographs and 23 articles indexed in SCOPUS (H-index 4) and has 8 patents of Ukraine. He is the editor-in-chief of the *Information Processing Systems* journal. Currently, he is a professor at the Ivan Kozhedub Kharkiv National Air Force University, Ukraine. He is the director of several Doctorates and Postdoctoral thesis. He delivered reports at international (EWDTW'05, EWDTW'06, DESSERT, 18) and local conferences. He is a member of the scientific and technical council of the Ivan Kozhedub Kharkiv National Air Force University. He is a member of the specialized councils of the Ivan Kozhedub

Kharkiv National Air Force University and the N.E. Zhukovsky Kharkiv National Aerospace University “KhAI.”

Juan Marcos Toibero received his BEng degree in Electronic Engineering from the Facultad Tecnológica Nacional, Argentina, in 2002, and PhD degree in Control Systems from the Instituto de Automatica at the Universidad Nacional de San Juan, Argentina, in 2007. His main works are related to nonlinear control of robotic platforms and robotics applications. He is with the National Council for Scientific and Technological Research of Argentina since 2011 as adjunct researcher. He leads different technological projects, and his current scientific research is at the Institute of Automatics of San Juan, Argentina. His research interests include wheeled mobile robots; manipulators force/impedance; switched, hybrid, nonlinear control methods applied to automatic control; visual servoing; image processing applications; and human-robot interaction. He is currently working on the autonomous control of ASVs and the supervision and analysis of sport motions with RGBd cameras.

Kenta Tokuno is a 1st year student at the Graduate School of Science and Technology, Sanyo-Onoda City University. His current research interest includes “Development of a design tool for deep convolutional neural networks” and “Application of DCNN design tool for visual inspection systems.” He is a student member of JSME (Japan Society of Mechanical Engineers).

Vera Tyrsa was born on July 26, 1971. She received her BS and MS degrees in Kharkiv National University of Automobiles and Highways, Kharkiv, Ukraine, in 1991 and 1993, respectively (Honoris Causa). She received her PhD degree in Kharkiv National Polytechnic University on specialty “Electric machines, systems and networks, elements, and devices of computer technics” in 1996. She has written 3 book chapters and more than 50 papers and holds 2 patents of Mexico and Ukraine. From 1994 till the present time, she was represented by her research works in several international congresses in the USA, England, Italy, Japan, Ukraine, and Mexico. In April 1996, she joined the Kharkiv National University of Automobiles and Highways, where she holds position of associate professor of the Electrical Engineering Department (1998–2006). In 2006–2011, she was invited by the Polytechnic University of Baja California, Mexico, for professor and researcher position. Currently she is professor of Engineering Faculty of Baja California Autonomous University. Her current research interests include automated metrology, machine vision systems, fast electrical measurements, control systems, robot navigation, and 3D laser scanners.

Jesús Elías Miranda-Vega was born in 1984 and received a BE degree in Electrical and Electronic Engineering from Los Mochis Institute Technology in Sinaloa, Mexico, in 2007, and a Master’s degree in Electronic Engineering from the Mexicali Institute of Technology, Mexicali, Mexico, in 2014. He joined the Engineering Institute at the Autonomous University of Baja California (UABC) optoelectronics lab as a PhD student, Mexicali, Mexico, in August 2016. His current research interest includes machine vision, data signal processing, the theory and optoelectronics devices, and their applications.

Patricio A. Vela is an associate professor in the School of Electrical and Computer Engineering and the Institute of Robotics and Intelligent Machines, at Georgia Institute of Technology, USA. Prof. Vela earned his BSc degree in 1998 and his PhD degree in Control and Dynamical Systems in 2003, both from the California Institute of Technology, where he did his graduate research on geometric nonlinear control and robotics. In 2004, Dr. Vela was a postdoctoral researcher on computer vision with School of ECE, Georgia Tech. He joined the ECE faculty at Georgia Tech in 2005. His research interests lie in the geometric perspectives to control theory and computer vision. Recently, he has been interested in the role that computer vision can play for achieving control-theoretic objectives of (semi-)autonomous systems. His research also covers control of—and machine learning for—nonlinear systems, typically robotic systems.

Ross D. Jansen-van Vuuren obtained a PhD degree in Organic Chemistry in 2012 at the University of Queensland. The focus of his research was the development of filter-less, color-selective organic photodetectors to replace silicon in conventional image sensors. He then spent 8 months investigating ways in which tertiary educators based in developed nations could partner with and share resources with scientists based in under-resourced, low-income countries. He was then appointed as a Postdoctoral Research Fellow at the University of Queensland in the Centre for Organic Photonics & Electronics (COPE) where he worked on the development of (poly)dendrimers for OLEDs with applications in large, flexible lighting modules for 3 years. Since November 2017, **Ross** has been working as a postdoctoral research fellow with Professor Philip Jessop at Queen's University (Canada). His research focuses on the development of draw agents for forward osmosis, an energy-efficient water purification technology. He maintains an interest in the field of photodetection for applications in machine vision.

Marek Waśik graduated from Poznań University of Technology in 2012 receiving the BSc and MSc in Automatic Control and Robotics. He is currently a PhD student at the same university. His research interests include legged robots, design of mechatronic systems, and multi-sensor fusion.

Keigo Watanabe received BE and ME degrees in Mechanical Engineering from the University of Tokushima in 1976 and 1978, respectively, and a DE degree in Aeronautical Engineering from Kyushu University in 1984. From 1980 to 1985, he was a research associate at Kyushu University. From 1985 to 1990, he was an associate professor in the College of Engineering, Shizuoka University. From April 1990 to March 1993, he was an associate professor, and from April 1993 to March 1998, he was a full professor in the Department of Mechanical Engineering at Saga University. From April 1998, he was with the Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University. Currently, he is with the Department of Intelligent Mechanical Systems, Graduate School of Natural Science and Technology, Okayama University, Japan. His research interests include intelligent signal processing and control using soft computing, bioinspired robotics, and nonholonomic systems.

Xiao-Wei Ye is currently a professor in the Department of Civil Engineering at Zhejiang University, China. His research interests include structural health monitoring, fatigue of steel structures, and structural reliability. Dr. Ye received his PhD degree from the Department of Civil and Environmental Engineering at The Hong Kong Polytechnic University in 2010. He has been serving as a guest editor for several SCI-indexed international journals and invited as a peer reviewer for more than 30 SCI-indexed international journals. He is a recipient of a Gold Medal Prize at the 37th International Exhibition of Inventions in Geneva (2009) for the invention of the “Mega-structure Diagnostic and Prognostic System” and a Gold Medal Prize at the 41st International Exhibition of Inventions in Geneva (2013) for the invention of the “Intelligent Ship-bridge Anti-collision Surveillance System.” Dr. Ye has served as PI or co-PI for more than 20 research projects. He has published more than 100 papers including over 60 peer-reviewed journal articles. He also has ten Chinese patents and four Chinese computer software copyrights.

Further Readings

- López, M. R., & Flores-Fuentes, W. (2016). *Robust control-theoretical models and case studies*. ISBN: 978-953-51-2424-5, Print ISBN: 978-953-51-2423-8. Retrieved from <http://www.intechopen.com/books/robust-control-theoretical-models-and-case-studies>
- Rivas-Lopez, M., Flores, W., & Sergiyenko, O. (Eds.). (2017). *Structural health monitoring: Measurement methods and practical applications*. BoD–Books on Demand. ISBN: 978-953-51-3254-7, Print ISBN: 978-953-51-3253-0. Retrieved from <https://www.intechopen.com/books/structural-health-monitoring-measurement-methods-and-practical-applications>
- Rivas-Lopez, M., Sergiyenko, O., Flores-Fuentes, W., & Rodríguez-Quiñonez, J. C. (Eds.). (2019). *Optoelectronics in machine vision-based theories and applications*. IGI Global. ISBN10: 1522557512, ISBN13: 9781522557517. Retrieved from <https://www.igi-global.com/book/optoelectronics-machine-vision-based-theories/192034>
- Sergiyenko, O., Flores-Fuentes, W., & Tyrsa, V. (2017). *Methods to improve resolution of 3D laser scanning*. Lambert Academic Publisher. Managed by OmniScriptum AraPers GmbH Bahnhofstraße 28, D-66111 Saarbrücken. ISBN-10: 6202007559, ISBN-13: 978-620-2-00755-9. Retrieved from <https://www.lap-publishing.com/catalog/details/store/tr/book/978-620-2-00755-9/methods-to-improve-resolution-of-3d-laser-scanning?search=Methods%20to%20improve%20resolution%20of%203D%20Laser%20Scanning>
- Sergiyenko, O., & Rodríguez-Quiñonez, J. C. (Eds.). (2016). *Developing and applying optoelectronics in machine vision*. IGI Global. ISBN-10: 1522506322, ISBN-13: 978-1522506324. Retrieved from <https://www.igi-global.com/book/developing-applying-optoelectronics-machine-vision/147652>

Index

A

- Adaptive binary arithmetic coding (ABAC), 646–649
- Additive manufacturing (AM)
 - fabrication methodology, 711
 - mathematical models, 89
 - metal-based, 712
 - robotic devices, 88
- Agricultural machine vision applications
 - guidance and control, 205–207
 - plant identification, 203–204
 - process control, 204–205
- Analog-digital coding, 90, 91
- Ant colony optimization (ACO)
 - algorithm, 744, 746
 - amalgamates, 744
 - filter weight, 745–746
 - food source, 743
 - motivation, 742
 - optimization algorithms, 744
 - two-bridge experiment, 742, 744
 - WRM filter, 736, 739
- Artificial intelligence
 - in agricultural applications, 202
 - machine vision (*see* Machine vision)
 - multi-agent systems, 390
 - regression algorithms, 772
- Automated reflector-based target recognition
 - fine pointing/aiming, 145–146
 - rough pointing/coarse search, 144, 145
- Automated storage and retrieval system (AS/RS), 494–495
- Autonomous vacuum cleaners, 496
- Autonomous vehicle
 - electric autonomous cars, 486

- individual transport, 486
- levels of automation, 487–489
- main components, 489–493

B

- Binocular vision systems
 - artificial biological
 - applications, 258
 - SVS applications, 256–258
 - vision model, 243–248
 - coordinates, 250
 - factor conversion, 248–249
 - projection coordinates, 249
 - SVS, 248
- Biological inspirations, 105, 122, 742
- Blossom isolation
 - apple trees, 216–217
 - data transformations, 208–212
 - peach trees, 218–219
 - sample image, 208
 - testing, 212–214
 - tree, 214–215
- Blown powder deposition (BPD)
 - AM, 711
 - analog high-speed sensors, 714
 - destructive testing, 713
 - height control system, 717
 - infrared thermograph, 721
 - melt pool data, 729
 - metal-based AM, 712
 - quality and properties, 712–713
 - vision-based real-time monitoring, 713
- Bundle adjustment (BA), 78, 268, 314–317, 581, 828

- Business models
 - autonomous car, 529
 - expensive vehicles, 527
 - generation Z consumer profile, 527–528
 - model canvas, 528
- C**
- Calibration
 - camera
 - and measurement setup, 720
 - to-range, 260
 - virtual, 42, 43
 - eccentric layout, 138
 - image distortions, 18
 - panoramic images, 40–42
 - radar equipment, 178
 - source–receiver path, 693
 - static and dynamic identification, 805
 - subsystems, 39–40, 43, 44
 - 3D phase triangulation measurements, 686
 - transformation parameters, 145
- Camera parameter optimization, 260, 328, 341, 588
- Camera-space manipulation (CSM), 325–328
- Central vision, 52–55
- Color filter, 77, 208, 213, 214, 216, 218, 219, 227, 230
- Commercial autonomous vehicles, 495–496
- Complementary metal-oxide semiconductor (CMOS)
 - architectures, 6
 - vs. CCD, 6
 - image sensing technology, 76
 - MPixel, 161
 - 3D active imaging system, 72
 - transistors, 76
- Computer vision
 - anti-collision piers, 791
 - bridge
 - introduction, 792
 - reinforcement, 789
 - collision incidents analysis, 793–795
 - detection methods, 791
 - navigation, 793
 - ship collision, 790
 - significance, 795
 - simulation/test research, 790
- Condition monitoring, 734, 739
- Continuous logical, CL ADC
 - 8-bit ADC, 120, 123
 - CM-6, 111–117
 - 8-DC-(G), 117–119
 - equivalence models, 107–111
 - modification, 107–111
 - theoretical foundations, 107–111
 - time diagrams, 122
- Control of nanomanipulators, 767
- Conventional imaging systems
 - camera, 68
 - MHP/OHP, 20–21
 - OPDs (*see* Organic photodetectors (OPDs))
- Convolutional neural network (CNN)
 - description, 659, 660
 - domain, 79
 - and NN, 96
 - R-CNN, 658
- Correlation navigation systems (CENS)
 - classification, 540
 - current and reference images, 565–567
 - FR locating probability, 547
 - functioning process, 541
 - multi-threshold selection, 568–572
 - radiometric, 560–564
 - three-dimensional form object binding, 555–560
 - unimodal decision function, 572–574
- Current mirror (CM), 91, 95, 97, 101, 103, 105, 119
- D**
- Data transferring
 - communication, 406
 - feedback implementation, 410–413
 - leader based communication, 408–410
 - path bridging, 407
 - spanning tree protocol, 406
 - swarm robotics, 407–408
- Deep convolutional neural networks (DCNN)
 - back propagation algorithm, 776–778
 - design and training application, 772–776, 778–780
 - DNN concept, 772
 - programming languages, 772
 - SVMs, 772, 773
 - test trial, 781, 782
- Deep learning approaches, 79–80
- Digital image processing, 91, 149, 207
- E**
- Echo signals
 - concentrated objects, 186–189
 - distributed objects, 179–186
 - formation, 174

- MAR, 190–195
 - reflected signal, 173
 - small-scale irregularities, 173
- Edge detection, 713, 723, 724, 727, 730
- Electronic angle measurement
 - code-based, 142
 - incremental, 142–143
- Electronic distance measurement (EDM)
 - phase-based method, 141–142
 - pulse method, 140–141
- Electronics
 - angle measurement (*see* Electronic angle measurement)
 - receiver systems, 16
 - sensor, 62
 - wave function, 13
- Environment sensing
 - kinect sensor, 499–500
 - LiDAR sensor, 497–499
- Equivalence (non-equivalence) functions, 93, 113
- Equivalent models, 93

F

- Fault detection and isolation (FDI)
 - algorithms, 735
 - engine modules, 735
 - filter design spaces, 736
 - fingerprints, 734
 - gas turbine, 734
 - jet engine, 735
 - types, 734
 - WRM filters, 736–737
- Feedback systems
 - energy management system, 715–717
 - height control system, 717–720
 - high temperature region, 720–722
 - visual, navigation, 234–235
- Field test
 - interface monitoring, 803–804
 - ship identification, 805–806
 - system summary, 803
 - warning system, 804–805
- Flying robots (FR)
 - automatic navigation, 538
 - autonomous navigation solution, 555
 - CENS (*see* Correlation navigation systems (CENS))
 - image modeling, 548
 - navigation and obstacle, 539
 - theory, 538

- Fovea centralis
 - human eye, 635
 - HVS based, 632
 - image compression (*see* Image compression)
 - receptors, 636
- Fovea centralis hierarchical tree (FVHT)
 - algorithm, 644
 - block diagram, 643, 644
 - simulation results, 644–645
- Fruit yield estimate, 219, 232
- Fundamental matrix, 281, 309–314, 318

G

- Gabor filter, 662–664
- Gas path measurement images
 - ideal signal, 740
 - objective function, 741–742
 - test signals, 739
 - time delay, 740
 - turbofan jet engine, 738
 - types of signals, 739
- Georegistration, 580, 589
- Global planning
 - costmap, 443
 - discrete spaces, 434–436
 - planning in continuous spaces, 436
- Global positioning system (GPS)
 - autonomous vehicle, 491
 - beacon system, 512
 - calculation of position, 512
 - INS measurements, 518
 - measurement, 305
 - navigation system, 171
 - passive and exteroceptive sensors, 513
 - satellite information, 513
 - sensors, 199
 - signals, 492
 - SLAM, 514
- Ground signal ground (GSG) probe
 - context, 752–753
 - SEM, 753–754
 - specifications, 754

H

- Homography
 - camera sensor pose measurements, 581
 - experiments, 588–592
 - feature track building, 582
 - geographical regions, 579–580

- Homography (*cont.*)
 GPS and IMU, 580
 imaging model, 582–584
 moving objects, 580
 optimization, 584–588
 related work, 581–582
- Human visual system (HVS), 7, 17, 631
- Hybrid sensor
 avoiding obstacles, 46–49
 central vision, 49–50
 detection of objects, 44–45
 tracking of objects, 45–46
- I**
- Identification and PID controller, 756, 757
- Image-assisted total station (IATS)
 feature extraction
 SIFT, 151
 SURF, 151–153
 and laser tracker measurement, 167
 processing fundamentals, 149–150
 regression line, 166
 robotic total station, 136
 static object recognition, 148
- Image based controller, 607–608
- Image compression
 block-DCT methods, 641
 block diagram, 639
 foveated images, 641–642
 JPEG image coding, 640
 JPEG2000 standard, 640
 quantization equation, 639
 ROI (*see* Fovea centralis hierarchical tree (FVHT))
- Image intensity transformation
 analog 64-input and 81-input neuron
 equivalentor, 106–107
 Mathcad, 98–100
 nonlinear transformations, 97–98
 OrCad PSpice, 100–106
 SLECONS, 92–96
- Image processing
 algorithms, 44
 blossom isolation (*see* Blossom isolation)
 BPD (*see* Blown powder deposition (BPD))
 centroid of path plane, 235
 classification, 202
 DCNN (*see* Deep convolutional neural networks (DCNN))
 feature extraction, 201
 FDI (*see* Fault detection and isolation (FDI))
 fundamentals, 149–150
 IATS (*see* Image-assisted total station (IATS))
 median filters, 737–738
 MVS, 5
 numerical experiments, 746–748
 preprocessing, 201
 ROS stereo, 475
 segmentation, 201
- Image processor (IP), 91, 95, 111, 116, 119
- Image registration, 580, 581, 585
- Image sensor
 CMOS, 6
 consumer digital photography, 4
 fabrication, 15
 Foveon X3, 7
 machine vision (*see* Machine vision)
 passive imaging technology, 62
 performance metrics, 7–8
 photosensor material, 5
 ROIC, 24
- Infinitesimal rotation
 lie algebra, 297–300
 parameterize, 318
 quadratic approximation, 302
- Infrared (IR) thermography, 7189
- Inorganic-based imaging systems
 crosstalk, 12
 DR, 10
 illuminant variation, 11
 image sensors, 8
 incompatibility, 11
 low bandgap, 12
 weak light absorption, 10
- Integer discrete cosine transform (iDCT), 643, 649
- Integrated colour pixel (ICP), 7, 8
- Intelligent transportation scheme
 mechanical design and kinematic model,
 525, 526
 SLAM, 523–524
- Intensity transformation, 92–96, 98–100, 105, 119
- K**
- Kalman filter (KF)
 Bayes filter and belief update, 274–275
 Bayesian rule, 271–273
 EKF linearization technique, 275
 MLE, 270–271
 probabilistic inference, 271–273
 UKF stochastic linearization technique,
 275
- Kinematic mode, 162–167

L

LabView™

- control, 761–763
- idea, 755–756
- nanopositioning, 759–761
- open-loop transfer function, 756–759

Landmarks, 503–504

Laser metal deposition, 712

Lie algebra

- angular velocity, 294–296
- BA, 314–317
- exponential expression of rotation, 296–297
- infinitesimal rotations, 297–300
- matrix computation, 309–314
- maximum likelihood, 303–308
- optimization of rotation, 300–303
- parameterization of rotation, 294
- rotation matrices, 294
- small rotations, 294–296
- 3D pose computing, 293

Linear dynamic range (LDR), 9, 10

Localization sensors

- GPS, 512–513
- wheel encoders, 510–512

Local planning

- navigation, 437
- optimal trajectory synthesis, 438–439
- reactive methods, 437–438
- 3D environment representation, 439–440
- velocity/control space methods, 438

M

Machine vision

- actuation, 202
- agricultural applications (*see* Agricultural machine vision applications)
- applications, 198–199
- DCNN (*see* Deep convolutional neural networks (DCNN))
- definition, 198
- digital cameras, 5–7
- experimental ground vehicle platform, 236–237
- image
 - acquisition, 200–201
 - processing (*see* Image processing)
 - sensing, 4–5
 - sensor photodiodes, 7–8
- orchard management, 199
- peach orchard navigation (*see* Peach orchard navigation)
- robotics (*see* Robotics)

- scene constraints, 200
- stereo imaging, 199
- SVM (*see* Support vector machines (SVM))
- visual feedback system, 234–235

Map building, 504–506

Mapping sensors

- kinect, 508–510
- LiDAR, 507–508

Mathematical operators, 96–97

Matlab™, 763–764

Maximum likelihood estimator (MLE), 270–271

- lie algebra, 318
- rotation estimation, 303–308

Melt pool

- boundaries, 724
- BPD, 723
- edge detection techniques, 723, 727
- Gaussian blur, 726–727
- JSR, 723
- Laplacian transform, 726–727
- peaks, 724, 725
- repeatability, 728–729
- sensitivity, 728–729
- solidified region, 727
- spatial variation, 726

Microwave measurement, 813

Mixed sensor processors

- continuous-discrete automation, 90
- DOEP, 91
- EMs, 89
- MAAM and MHAM, 89
- mutual 2D function, 88
- OE-VLSI circuits, 90
- optical learning, 90
- self-learning, 89
- VMO ADC, 91

Mobile autonomous robots (MAR)

- amplitude jump of signals, 190–195
- echo signals (*see* Echo signals)
- EMW, 171, 176–179
- GPS, 171
- navigation, 172–176
- radiofrequency range, 172

Mobile industrial robots, 495

Mobile robots, 360–362

- applications, 511
- dynamic model, 350–351
- FOV, 457
- geometric description, 380
- localization process, 505
- machine vision (*see* Machine vision)
- model, 380–381

- Mobile robots (*cont.*)
- navigation, 473
 - omnidirectional cameras, 34
 - passivity-based control laws, 348
 - pioneer 3-AT, 417
 - radar antenna, 174
 - self-contained vision sensor, 35
 - trajectory, 365
 - vision-based, 258
- Modulated light scanners, 686
- Multi-camera models, 254–256
- Multivision systems
- multi-camera models (*see* Multi-camera models)
 - SVS, 259–260
 - trinocular vision models (*see* Trinocular vision models)
- N**
- Nanopositioners
- in closed loop, 759–761
 - LabView™ (*see* LabView™)
 - linear, 755
 - open-loop transfer function, 756–758
 - patterns detection, 764, 765
 - points, 765–767
 - probes, 753
 - set points, 764
- Navigation
- benchmarking, 468–469
 - experiments
 - campus world, 472, 473
 - office world, 472, 473
 - review of outcomes, 473–475
 - sector world, 471–472
 - FR (*see* Flying robots (FR))
 - machine vision (*see* Machine vision)
 - MAR, 172–176
 - robot, 36, 50
 - robotic group (*see* Robotic group)
 - scenario configuration, 466–468
 - SLAM, 513–518
 - SVS, 259
 - 3D sensing (*see* 3D sensing)
 - UAV, 204
 - in vision-based aerial imaging, 15
 - world synthesis, 465–466
- Neuron-equivalentor (Neqs), 119
- Neuroscience, 440–442
- Nonlinear analysis
- filters, 228, 735, 748
 - processing, 275
 - transformations, 97–98
- O**
- Object detection (OD)
- applications in robotics, 80
 - digital elevation map, 501
 - Gaussian model, 828
 - geometry-based clusters, 502
 - image processing, 167
 - and matching, 153–155
 - pairwise and combinatorial matching, 581
 - position determination, 155–161
 - probabilistic occupancy map, 500, 501
 - scene flow segmentation, 501, 502
- Object isolation alternate method
- car, 230
 - conclusion, 233
 - spatial mapping, 230–233
 - stereo camera operation, 231–232
- Object tracking
- autonomy, 347
 - classified, 348
 - color-based, 46
 - feature selection, 381–383
 - kinematic image-based, 161–162
 - module, 37
 - passivity-based proposals, 348
 - principles, 156
 - time aspects, 148
- Omnidirectional vision, 39
- On-wafer probe station, 752, 753
- Optoelectronics
- dual analog neuron-equivalentors, 119
 - geometric control, 707
 - integration, 8
 - OPD, 19
 - optics, 20
 - signal processing, 541–546
- Orchard management
- machine vision system, 199
 - navigation (*see* Navigation)
 - RGB values, 209
 - sample image, 208
 - system models (*see* System models)
 - theoretical analysis, 349
- Organic photodetectors (OPDs)
- active 3D imaging, 18–19
 - AK-SHB 810 model camera, 15
 - baseline, 18
 - broadband/narrowband, 14
 - chemical structures, 13–14
 - CMOS image sensors, 15
 - colour sensing systems, 14
 - depth measurement technology, 17–19
 - disruptive technology, 13

- electronics cost, 16
 - image sensors, 15
 - Si-photodiodes, 13
- P**
- Parallax
 - image registration technique, 581
 - magnitude, 586
 - occlusion, 580, 581
 - stereo vision system, 62
 - Passivity based controller
 - analysis, 609
 - kinematics, 622
 - robustness analysis, 622–623
 - UAV dynamic model, 620–621
 - vision system, 621
 - visual servoing control, 608
 - Passivity-based visual controller design
 - compensation, 356–357
 - kinematic-based controller, 353–356
 - robustness analysis, 358–359
 - simulation and experimental results
 - mobile manipulator, 362–371
 - mobile robot, 360–362
 - robotic manipulator, 371–376
 - vision system, 352–353
 - Path planning
 - A* algorithm, 519–520
 - D* algorithm, 520–522
 - secondary objectives placement, 405–406
 - technical vision system, 403–405
 - Path tracking, 332–337
 - Peach orchard navigation
 - agricultural operation, 233
 - experimental ground vehicle platform, 236–237
 - visual feedback system, 234–235
 - Peripheral vision, 50–52
 - Phase triangulation
 - dynamic range, 700–702
 - nonlinearity compensation, 684–690
 - optical radiation source–receiver path, 676
 - phase measurement error, 690
 - proposed method, 690
 - spatial modulation, 703–707
 - steady method, 676–684
 - structured image decoding, 690–700
 - 3D object geometry measurement, 675
 - Photodetectors, 8–10
 - electrical analog inputs, 95
 - harmonic signal, 705
 - in image sensing (*see* Organic photodetectors (OPDs))
 - intensity distribution, 703
 - materials/image processing technologies, 5
 - performance metrics, 9
 - transmission of light, 7
 - Photogrammetry, 135
 - Photo-sensing material
 - conventional imaging (*see* Conventional imaging systems)
 - image sensor, 8
 - OPT platform, 23
 - organic–inorganic hybrid layer, 23
 - Phototransistors, 21–24
 - Planning in perception space (PiPS)
 - collision checking, 445–450
 - egocylindrical perception space, 451–454
 - modifications, 444–445
 - navigation, 442–444
 - trajectory scoring (*see* Trajectories)
 - Plant identification application
 - image processing (*see* Image processing)
 - over-constraining the sample data, 227–229
 - yield estimation
 - statistical and probabilistic results, 226
 - transition, 220
 - weight values, 221–226
 - Pose estimation, 287
 - Position based controllers
 - analysis, 606
 - visual servoing control, 606
 - Powder feed, 712, 720–722, 727
- R**
- Radar detection, *see* Mobile autonomous robots (MAR)
 - Radiometrics
 - autonomous and noise-cancel FR, 539–541
 - correlation-extreme navigation
 - decisive function, 546–547
 - RM and CENS, 541–546
 - FR, 538
 - impact analysis, 542–554
 - Region of interest (ROI), 635, 636, 646, 658, 668–671, 720
 - Risk assessment
 - bridge collision, 796
 - early warning event, 800–802
 - post recording system, 802
 - system warning trigger method, 800
 - warning area division, 799–800
 - Robotic group
 - autonomous vehicles, 390
 - data transferring networks (*see* Data transferring)

- Robotic group (*cont.*)
 - effectiveness, 423–424
 - nature swarm adaption, 390–392
 - swarm projects, 394–397
 - tasks, 392–394
- Robotic image-assisted total stations
 - coaxial layout, 137
 - correction function, 139
 - eccentric layout, 137, 138
 - IATS, 136
 - principle layout, 137
 - Trimble S7, 138
 - working principles (*see* Electronic distance measurement (EDM))
- Robotics
 - conventional vision technology, 60
 - medical, 60
 - MIS, 60
 - 3D image construction
 - dynamic vision, 68–70
 - sensor, 62
 - shape from shading, 66–68
 - stereo vision, 62–66
 - UAV (*see* Unmanned aerial vehicles (UAV))
- Robotic total stations (RTS)
 - accuracy, 135
 - angles and distances, 134
 - optics and mechanics, 134
 - science community, 135
 - tachymeter, 134–135
 - theodolites/total stations, 136
- Robots
 - autonomy, 33–34
 - biological vision, 34
 - hardware of the sensor, 37–38
 - hybrid field of view, 35
 - natural evolution, 34
 - related work, 36–37
 - self-contained perception unit, 35
 - software and calibration (*see* Calibration)
- S**
- Scale-invariant feature transform (SIFT), 151
 - detector step, 151
 - detector/descriptor pair, 53
 - feature matching procedures, 50
 - features, 588
 - image-based object recognition, 151
 - Sparse stereo, 54
 - and SURF, 153
- Scanning electron microscopy (SEM), 752–754
- Self-driving car, 4, 60, 495
- Self-learning equivalent convolutional neural structures (SLECNS), 92–96, 122
- Sensors
 - hardware, 37–38
 - hybrid (*see* Hybrid sensor)
 - image intensity transformation (*see* Image intensity transformation)
 - machine vision (*see* Machine vision)
 - robotics (*see* Robotics)
- Ship-bridge collision
 - anti-collision system, 795–796
 - monitoring and tracking system, 796–798
 - multifactor method, 792
 - vision-based system, 807
- Signal processing
 - computer vision technology, 791
 - RM channel CENS, 541–546
 - theory, 72
 - time-pulse, 90
- Signal sensor, *see* Machine vision
- Simultaneous localization and mapping (SLAM)
 - control flow process, 523, 524
 - diagram, 523, 524
 - multi-robot map alignment, 256
 - navigation
 - control, 513–518
 - and mapping, 268
 - nonlinear problems, 275
 - robot localization problem, 79
- Smart campus
 - kinematic model, 525, 526
 - mechanical design, 525, 526
 - SLAM, 523–525
- Speeded-up robust feature (SURF)
 - algorithm, 151–153
 - feature matching procedures, 50
- Stairways detection
 - CNN, 658
 - experimental results, 665–667
 - Gabor filter, 662, 663
 - HH, 662
 - localization and recognition, 658, 664
 - MDPG, 660
 - proposed algorithm, 665
 - synthesized image stair set, 664
 - testing image sets, 661
- Stereo
 - distance measurements, 43
 - OPDs, 17–18
 - perspective camera, 38
 - photometric, 67
 - virtual camera, 38

- vision, 62–66
 - See also* Stereoscopic vision systems (SVS)
 - Stereo cameras, 463–464, 475–478
 - Stereoscopic vision systems (SVS)
 - angles and coordinates, 248
 - artificial biological applications, 256–258
 - constraint
 - collinearity, 278–280
 - coplanarity, 280–282
 - biological processes, 243
 - depth information, 243
 - digital scene reconstruction, 242
 - epipolar geometry, 276, 280–282
 - extended KF, 276
 - feature extraction, 242
 - images acquisition, 242
 - localization approach, 286–287
 - multivision applications, 259–260
 - pattern matching, 243
 - perspective projection, 278–280
 - pose tracking, 284–286
 - system geometry, 242
 - 3D
 - point, 247
 - reconstruction stage, 278
 - trinocular applications, 258–259
 - uncertainties, 282–284
 - unscented KF, 277
 - Stereo vision-based scanners
 - binocular, 62, 63
 - depth, 63
 - dynamic vision, 68–70
 - energy minimization function, 64–65
 - epipolar geometry, 280–282
 - extended Kalman filter, 276
 - feature-based methods, 66
 - local methods, 64
 - matching algorithms, 63–64
 - OPDs, 17–18
 - perspective projection, 278–280
 - semi-global method, 65
 - shape from shading, 66–68
 - SLAM algorithm, 276
 - ToF sensors, 60
 - vehicle detection system design, 398
 - Structural health monitoring (SHM)
 - bridges, 791
 - implementation of approach, 422
 - SVS, 257
 - 3D coordinates, 242
 - Structured image decoding
 - error of phase, 695–699
 - intensity of the source, 694
 - nonlinear distortions, 691
 - power transfer function, 691
 - probing phase image, 692
 - source–receiver path, 692
 - substantiation, 693
 - theoretical error, 693
 - 3D profile, 690–691
 - three-dimensional surface sections, 697
 - 3D profile, 690–691
 - Structured light scanners
 - classes, 73
 - constrained (indoor)/unconstrained (outdoor), 61
 - depth measurement sensors, 74–75
 - dynamic vision, 70
 - 3D measurements, 703
 - system architecture, 76
 - Structure from motion (SfM), 78–79
 - Support vector machines (SVM)
 - binary classification, 483
 - classification results, 784, 785
 - DCNN (*see* Deep convolutional neural networks (DCNN))
 - regression method, 772
 - on trained DCNNs, 782–785
 - Surface characterization, 329–331
 - Surface mapping
 - data exchange, 416–420
 - modeling system structure, 415–416
 - objects extraction, 420–422
 - simulation frameworks, 413–415
 - System models
 - kinematic, 351–352
 - mobile robot, 350–351
 - nomenclature, 349
 - robotic manipulator, 350
- ## T
- Target tracking, 147–148
 - Technical vision system
 - data reduction, 400–402
 - historical background, 398
 - structure and working principles, 399–400
 - 3D active imaging system
 - advantages, 71
 - OPDs, 18–19
 - SfM, 78–79
 - structured light, 73–78
 - ToF, 71–73
 - 3D measurements
 - double-baseline stereo algorithm, 259
 - high-accuracy, 257
 - high-speed sensor, 509

- 3D measurements (*cont.*)
 - optical radiation, 691
 - opto-mechanical system, 772
 - 3D sensing
 - characteristics, 304
 - complementary roles, 432, 433
 - global path, 433
 - mobile robot navigation, 434
 - navigation (*see* Navigation)
 - 3D vision system
 - development, 523
 - robotic group, 397–398
 - Time-of-flight (ToF), 71–73
 - high-quality depth maps, 19
 - light source/transmitter, 71
 - OPDs, 19
 - photosensor, 72
 - single-photon detection, 73
 - spot scanner, 73
 - surface points, 19, 71
 - 3D dense data sources, 439
 - ultrasonic wave receiver, 490
 - Tracking
 - automated reflector-based target
 - recognition, 144–146
 - IATS, 148–156
 - object module (*see* Object tracking)
 - particle filter, 53
 - path-generation, 323–325
 - principles, 135
 - simulation results, 612–619
 - UAV (*see* Unmanned aerial vehicles (UAV))
 - UKF and stereo vision, 286
 - Traffic signs, 502–503
 - Trajectories
 - arbitrary shapes, 322
 - cost functions
 - global path, 457
 - goal point, 461–462
 - obstacle, 459–460
 - path, 462–463
 - scoring, 458
 - simplification, 458
 - CSM, 325–328
 - egocircle measurements, 455–457
 - experimental validation, 337–343
 - geodesic mapping, 322–323
 - heat-induced deformations, 323
 - internal controllers, 598
 - length, 418, 419
 - path-generation, 323–325
 - propagation, 455–457
 - robot's end effector, 378
 - storage, 455–457
 - time derivative, 610
 - tracking
 - maneuver, 323–325
 - problem, 322
 - Trinocular vision models
 - arbitrary, 254
 - divergent, 253, 254
 - parallel model, 251–252
 - right triangular, 250–251
 - surrounding, 253
 - SVS, 258–259
- U**
- UKF-based image filtering
 - feature correspondence extraction, 268, 269
 - SFM, 268
 - SLAM, 268
 - 3D model, 268
 - Unmanned aerial vehicles (UAV)
 - agricultural environments, 597–598
 - controller analysis, 610–611
 - and IMU, 598
 - models
 - dynamic, 600–601
 - kinematic, 599
 - robustness and stability properties, 599
 - visual servoing, 598
- V**
- Video compression, 632, 642–643
 - Video stabilization, *see* Homography
 - Vision system
 - binocular stereo, 62, 63
 - image
 - processing, 601–602
 - sensing, 4–5
 - kinematics, 602–605
 - machine vision (*see* Machine vision)
 - omnidirectional, 39
 - robotic, 23
 - SVS (*see* Stereoscopic vision systems (SVS))
 - trajectories (*see* Trajectories)
 - Visual servoing controllers
 - biological artificial SVS, 258
 - dynamic compensation, 610
 - passivity properties, 598
 - position based (*see* Position based controllers)

W

Wavelet-based coding approaches

ABAC, 646–648

AFV-SPECK algorithm, 648, 649

simulation results, 649–650

Wavelet transforms

classic algorithms, 632

compression algorithms, 631

data compression, 632–635

digital image, 630

DWT, 637–638

Fourier analysis, 636

fovea centralis, 635–636

HVS, 631

JPEG, 632

prediction, 638

two-dimensional signals, 637