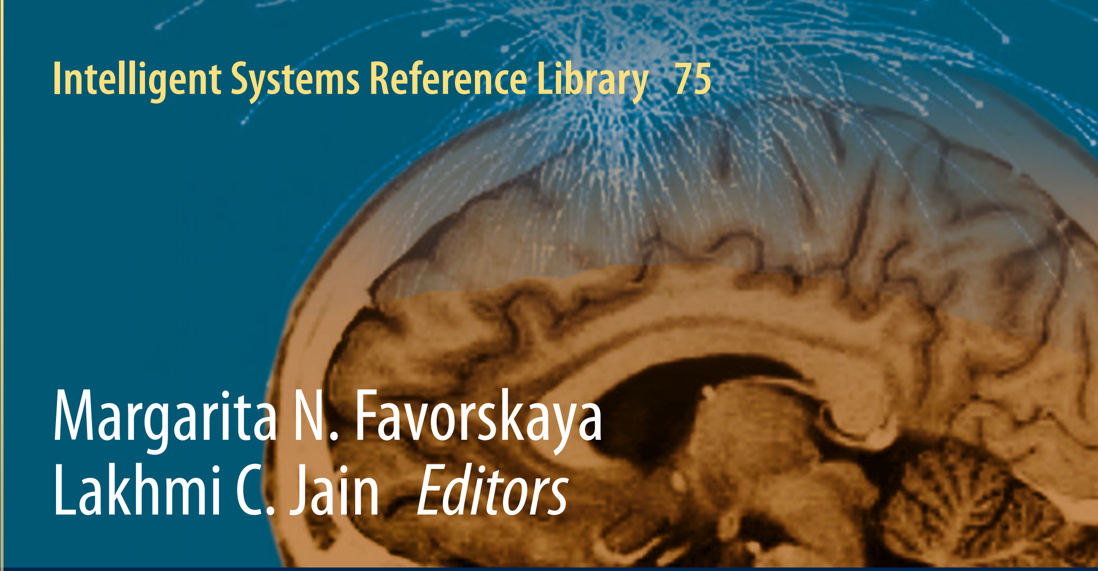
Intelligent Systems Reference Library 75



Margarita N. Favorskaya
Lakhmi C. Jain *Editors*

Computer Vision in Control Systems-2

Innovations in Practice

Springer

Intelligent Systems Reference Library

Volume 75

About this Series

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included.

More information about this series at http://www.springer.com/series/8578

Margarita N. Favorskaya · Lakhmi C. Jain
Editors

Computer Vision in Control Systems-2

Innovations in Practice

Springer

Editors

Margarita N. Favorskaya
Department of Informatics and Computer Techniques
Siberian State Aerospace University
Krasnoyarsk
Russia

Lakhmi C. Jain
Faculty of Education, Science, Technology and Mathematics
University of Canberra
Canberra
Australia

ISSN 1868-4394 ISSN 1868-4408 (electronic)
ISBN 978-3-319-11429-3 ISBN 978-3-319-11430-9 (eBook)
DOI 10.1007/978-3-319-11430-9

Library of Congress Control Number: 2014951912

Springer Cham Heidelberg New York Dordrecht London

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

In keeping with the overview of the mathematics underlying computer vision given in Volume 1, this volume introduces a rich landscape that spans a broad spectrum of computer vision applications. This landscape includes a very original view of human action recognition, a rich selection of novel methods in robot navigation systems, promising applications of face recognition methods, visual panorama reconstructions using images captured by mobile robot cameras, motion estimation methods based on salient feature points, intelligent robot motion control-based robotic visual perception (closely related to human perception), a novel approach to automatic surveillance based on the description of complex scenes and inter-object relationships, innovations in avionics, in-situ position estimation in navigating autonomous underwater vehicles, a thorough coverage of digital image filtration methods, as well as a good overview of image segmentation systems for monitoring such things as the Earth's surface, disease diagnosis, and technical object safety. As a result, this volume provides an in-depth view of computer vision applications in harmony with the computer vision theory in Volume 1. Appropriately, this volume begins with a detailed survey of the contributions by chapter authors: *Practical Matters in Computer Vision* (L.C. Jain, M. Favorskaya). This initial chapter is followed by a host of interesting practical views of computer vision.

The central motifs in this volume reflect a sensitive and remarkable understanding of computer vision in practice. These motifs are threefold.

1. Image Structures

Human Action Recognition (S. Al-Ali, M. Milanova, H. Al-Rizzo, V.L. Fox),
Efficient Denoising Algorithms for Intelligent Recognition Systems (A. Priorov, K. Tumanov, V. Kolokhov),
Image Segmentation Based on Two-Dimensional Markov Chains (E. Medvedeva, E. Kurbatova).

2. Image and Video Measurement

Real Time Audience Analysis System (V. Khryashchev, L. Shmaglit, A. Shemyakov),
Panorama Construction from Multi-view Cameras in Outdoor Scenes (L.C. Jain, M. Favorskaya, D. Novikov),
Real-Time Method of Contextual Image Description and Its Application in Robot Navigation and Intelligent Control (K.I. Kiy),
Perception of Audio Visual Information for Mobile Robot Motion Control Systems (S. Pleshkova, A. Bekiarski, S.S. Dehkharghani, K. Peeva).

3. Image-Based Signal Analysis

Adaptive Surveillance Algorithms Based on the Situation Analysis (N. Kim, N. Bodunkov),
Enhanced, Synthetic and Combined Vision Technologies for Civil Aviation (O. Vygolov, S. Zheltov),
Navigation of Autonomous Underwater Vehicles Using Acoustic and Visual Data Processing (I. Burdinsky, A. Myagotin).

A comprehensive view of applications in pattern recognition and image analysis are given in this volume. These applications are ably presented by the volume contributors.

I strongly recommend this volume and its companion volume as a concise and very original introduction to image analysis and its applications.

June 2014

James F. Peters
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB, Canada

and

Faculty of Arts and Sciences
Department of Mathematics
Adıyaman University
Adıyaman, Turkey

Preface

The research book is a continuation of our previous book, which is focused on the recent advances in computer vision methodologies and technical solutions using conventional and intelligent paradigms. The contemporary solutions based on advanced mathematical achievements emphasize more information and visual monitoring in natural and human environment. The real challenge of designing such observation models is to make them close to realistic visualization and interpretation of events in our world.

This book presents some of the research results from some of the most respectable researchers in the field of computer vision including some innovative applications in practice. The contributions include the recent methodologies for human action recognition, real-time audience analysis system, panorama construction from multiview cameras in outdoor scenes, real-time applications in robot navigation and intelligent control, adaptive surveillance algorithms, vision technologies for civil aviation, navigation of autonomous underwater vehicles, denoising algorithms for intelligent recognition systems, and image segmentation based on 2D Markov chains.

The book is directed to professors, researchers, and software developers working in the areas of digital video processing and computer vision technologies.

We wish to express our gratitude to the authors and reviewers for their contribution. The assistance provided by Springer-Verlag is acknowledged.

Russia
Australia

Margarita N. Favorskaya
Lakhmi C. Jain

Contents

About the Editors

Margarita N. Favorskaya received her engineering diploma from Rybinsk State Aviation Technological University, Russia, in 1980 and was awarded a Ph.D. by S.-Petersburg State University of Aerospace Instrumentation, S.-Petersburg, in 1985. Since 1986 she worked as an Associate Professor of Siberian State Aerospace University, Krasnoyarsk. Margarita Favorskaya defended her doctorial dissertation in Siberian Federal University in 2011. Since 2011 she is a Professor and a Head of Department of Informatics and Computer Techniques at Siberian State Aerospace University.

Her main research interests are digital image and videos processing, pattern recognition, fractal image processing, artificial intelligence, information technologies, and remote sensing. She is the author or the co-author of nearly 130 scientific publications and 20 educational manuals in these fields. Margarita Favorskaya is a member of KES organization, IPC member of International Conferences, and Co-Chair of Invited Sessions. She serves as a Reviewer, a Guest Editor, and an Associate Editor in International Journals.

Lakhmi C. Jain is with the Faculty of Education, Science, Technology, and Mathematics at the University of Canberra, Australia and University of South Australia, Australia. He is a Fellow of the Institution of Engineers Australia.

Dr. Jain founded the KES International for providing a professional community the opportunities for publications, knowledge exchange, cooperation, and teaming. Involving around 5,000 researchers drawn from universities and companies world-wide, KES facilitates international cooperation and generate synergy in teaching and research. KES regularly provides networking opportunities for professional community through one of the largest conferences of its kind in the area of KES. www.kesinternational.org.

His interests focus on the artificial intelligence paradigms and their applications in complex systems, security, e-education, e-healthcare, unmanned air vehicles, and intelligent agents.

Chapter 1
Practical Matters in Computer Vision

Lakhmi C. Jain and Margarita N. Favorskaya

Abstract A brief description of researches close to implementation in technical systems is represented in this chapter. Human action recognition and audience analysis systems as well as smart software tool for panorama construction help for a well-being of a human. The application of novel methods in robot navigation systems and the perception of audio visual information for mobile robots are the issues of other innovative investigations. The adaptive comprehensive surveillance algorithms for situation analysis, the enhanced, synthetic, and combined vision technologies for civil aviation, and the navigation techniques reflect the recent achievements in machine vision for robotics and autonomous vehicles. Also the efficient denoising algorithms and the image segmentation based on 2D Markov chains are useful in intelligent recognition systems.

Keywords Video surveillance · Face recognition · Panorama construction · Robot navigation · Avionics · Autonomous vehicle

1.1 Introduction

Video surveillance has a wide variety of applications in outdoor and indoor environment. Even a single video sequence provides the redundant data while information from multi-cameras combining the data from other sensors (acoustic signals, deep of scene data, odometer data, tactile information, etc.) is transformed in large scale data. On the other hand, in practice only the recognition of some objects or events as well as

L.C. Jain (✉)
Faculty of Education, Science, Technology and Mathematics, University of Canberra,
Canberra, ACT 2601, Australia
e-mail: lakhmi.jain@unisa.edu.au

M.N. Favorskaya
Institute of Informatics and Telecommunications, Siberian State Aerospace University,
31 Krasnoyarsky Rabochy, Krasnoyarsk 660014, Russian Federation
e-mail: favorskaya@sibsau.ru

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_1

a scene classification is required. Therefore, a development of intelligent systems is the urgent goal for scientific community in computer vision scope. The requirements of real-time implementation with the desired degree of accuracy are the main contradictory criteria for most vision-based systems. Often the analysis of rich experimental results permits to find a way for the reasonable simplification of high-cost algorithms in order to receive the innovative practical decisions.

1.2 Chapters Included in the Book

Two volumes “Mathematical Theory” and “Innovations in Practice” are included in the presented book. Ten original chapters are devoted to development the human surveillance monitoring systems, algorithms and software tools for robots’ navigation in various environments, and hardware for novel avionics solutions based on enhanced vision technologies.

Chapter 2 presents a fast growing field of research—a human action recognition extracted from videos as an important scope with numerous applications in the area of computer vision [1]. Applications for human action recognition include video surveillance, video indexing, and human-computer interface. In the case, where a video is segmented to contain a single implementation of human activity, the objective of the system is to classify video data into its respective activity category [2]. In this chapter, a number of existing methods for human activity recognition are discussed. First, the object extraction is done in each frame by performing background subtraction. Second, the frame normalization is applied by using a horizontal frame alignment and a resizing. Third, an Aligned Motion Image (AMI) is computed. Fourth, the video normalization is applied in each video allowing the cropping boundary box to go around the AMI, and subsequently unifying the size for all videos. Finally, a structure similarity measurement is applied to find the similarity between the different AMIs [3]. At the end, an application example of a new algorithm for human action recognition is presented. These results are very close to the recognition of a human observer. The result is about 96.774 % of correct recognitions by using all frames or “complete sequence” in each video. The result is about 98.925 % by using the first 30 frames or “sub sequence” from each video. The obtained experimental results demonstrate a high level of accuracy and efficiency of the proposed methodology.

Chapter 3 is devoted to automatic video data analysis of face and gender recognition. The promising practical applications of face recognition algorithms can be used in modern biometric control system, visitors calculation systems, throughput control on the entrance of office buildings, airports, automatic systems of accident prevention, intelligent human-computer interfaces, and some others [4]. The gender recognition can be applied to collect and estimate the demographic indicators [5]. Besides that it can be an important pre-processing step of person identification, the gender recognition allows twice to reduce the number of candidates for analysis, and thus twice to accelerate the identification process. A human age estimation is another problem in the field of computer vision, which is connected with face analysis [6]. Among its possible applications one should note an electronic customer relationship management, a security control, a surveillance monitoring, and biometrics. In order to organize a completely automatic system, the classification algorithms are utilized in the combination with a face detection algorithm, which selects candidates for further analysis [7]. The quality of face detection step is critical to the final result of the whole system, as inaccuracies at face position determination can lead to wrong decisions at the stage of recognition. To solve the task of face detection, the AdaBoost classifier is utilized. The detected fragments are preprocessed to align their luminance characteristics and transform them into a uniform scale. On the next stage, the detected and preprocessed image fragments are passed to the input of gender recognition classifier, which makes a decision on their belonging to one of two classes. The same fragments are also analyzed by the age estimation algorithm, which divides them into several age groups. To estimate the period of a person's stay in the range of camera's visibility, a face tracking algorithm is proposed. This chapter describes the main algorithmic techniques utilized in different stages of the proposed video data analysis system, which provides collection and processing of information about the audience in real time. The level of gender and age classification accuracy are estimated in real-life situations. The algorithms proposed in this chapter incorporate the universal machine learning techniques, and thus can be applied to solve other object classification tasks.

Chapter 4 investigates the issues of panorama construction by using images received from several cameras, which are maintained on mobile robots, or by hand-held shooting [8]. The main researches presented in this chapter connect with geometrical alignment of selected frames, color enhancement in shadow and bright areas, and seamless frames stitching. The accuracy of each algorithm is achieved by a high computational cost and non-real time realization. Therefore, several ways have proposed to increase the time execution with a suitable accuracy reduction [9]. A stitching of frames is the main core in the panorama construction, which includes the procedures for detection and matching of the similar regions. In this research, the feature points approach is applied, particularly in speeding robust feature detector. The estimation of feature point's correspondences is executed by using famous RANdom SAmple Consensus (RANSAC) algorithm. The luminance enhancement of selected frames is the necessary processing stage. The classical retinex algorithm normalizes dark areas and provides a result image with large contract values. The Enhanced Multi-Scale Retinex (EMSR) algorithm equalizes adaptively the spectral ranges of dark and bright areas for a single frame [10]. A special function stretches the spectral ranges with low and high intensity values due to a reduction of the middle spectral range. The improvement of visibility into the stitching areas is often required in final panoramic images [11]. The point-based rendering attracts a high interest in geometric modeling as an alternative to triangle meshes. The following improvement of blending is connected with the multi-scale or the multi-orientation sub-bands with a number of bands, not more than 3–4.

The main idea is to blend the sun-bands of image with various blending degree. The rich experimental materials are represented in this chapter.

Chapter 5 studies the motion estimation methods based on salient feature points such as Harris corner points, scale invariant feature points, etc. in robot navigation systems. Such artifacts as a fast motion with changing directions, overlapping of moving objects, and a complex background require a novel technique in real-time navigation and control tasks [12]. This technique ought to be able to cope with detection and tracking of boundary curves of the main objects in the image and the objects themselves. In fast motion (e.g. in sport games), a human vision provides the navigation of a sportsman based on a small number of features such as critical objects perceived as contrast (frequently, colored) blobs [13]. The idea to provide such description into a real-time computer vision is in the backbone of the proposed method. It is well known that the lack of real-time techniques of stable image segmentation limits the capabilities of mobile robots to understand scenes and solve the localization (categorization) problem. The chapter describes a version of the required technique and its application in indoor (outdoor) robot navigation. The proposed method makes it possible to select objects in complex scenes and provide their recognition based on the obtained generalized geometric descriptions in the language of collections of intervals [14]. The main restriction is connected with overall low saturation of colors in the image. Several applications of a novel real-time method of contextual image description are presented in the chapter. In the first application, a robot is navigated using artificial color visual landmarks (e.g. ones composed of colored rectangles). Visual landmark may be put on a wall, be in hands of a human, or be mounted on another robot (moving landmarks). In the second application, a robot finds doors (opened or closed), windows, walls, etc., while running in indoor environment.

Chapter 6 introduces a precise mobile robot motion control with clear orientation in the area of robot perception and observation [15]. In this chapter, the mobile robot audio and visual systems are outlined. They use data from corresponding audio (microphone array) and video (mono, stereo, or infrared cameras) sensors, accompanied with laser range finder sensor. The audio and video information captured from the sensors is used in the perception audio visual model. The proposed model performs a joint processing of audio visual information and determines a current mobile robot position (current space coordinates) in the area of robot perception and observation. The captured from audio visual sensors information is estimated with suitable algorithms developed for a speech and image quality estimation [16]. The preprocessing methods for increasing a quality and minimizing the errors of mobile robot position are developed. The current space coordinates, determined from a laser range finder, are used as supplementary information of mobile robot position. Also space coordinates are applied for error calculation and comparison with the results from audio visual mobile robot motion control. In the development of the mobile robot perception audio visual model, some methods are used [17]. The RANSAC method estimates parameters of a mathematical model from a set of observed audio visual coordinate data. The method of direction of arrival determines a localization of sound source direction with microphone array of speaker sending voice commands to the mobile robot. The method for speech recognition classifies the voice commands sending from the speaker to the robot. The current mobile robot position is calculated from a joint usage of perceived audio visual information. It is used in appropriate algorithms for mobile robot navigation, motion control, and objects tracking: a map-based or map less methods, path planning and obstacle avoidance, simultaneous localization and mapping, data fusion, etc. The error, accuracy, and precision of the proposed mobile robot motion control with perception of audio visual information are analyzed and estimated from the results of the numerous experimental tests, which are presented at the end of this chapter. The experiments are carried out mainly with simulations of the algorithms listed above. The parallel computing methods in implementation of the developed algorithms permit to reach a real time robot navigation and motion control using perceived audio visual information from the mobile robot audio visual sensors.

Chapter 7 provides some automatic surveillance solutions under an uncertainty and a variability of characteristics for an observed scene such as inaccuracy definition of objects coordinates and the mutual location of objects, illumination distortions and partial or complete objects overlapping, shadows and noises. The implementation of adaptive comprehensive algorithm for image processing and analysis is one of directions improving the efficiency of surveillance under uncertain conditions [18]. The algorithm provides a possibility to observe the object actions under complex and changing surveillance conditions. Also such approach is useful for situation analysis in less informative scenes. The novelty of this approach is based on the original descriptions of objects into complex scenes and inter-objects relationships. Such descriptions involve the elements of language contingency management that makes them more robust to various destabilizing factors. Algorithms of situation analysis provide a decision-making based on the choice of valid extracted features in accordance with the target task [19]. The entropy estimations (initial, current, and final) permit to decrease an uncertainty of complex scenes with large numbers of moving objects. The situation analysis reduces the initial and/or current entropy estimations during the procedure of object detection [20]. The software tool for a situation analysis is realized by using specialized database, knowledge base, and model descriptions. Database contains the descriptions of observed objects and inter-object relationships (spatial, temporal, causal, etc.). Knowledge base includes production rules describing the causal relations between objects and situations. It provides the final decision-making for control system. The model descriptions are built by considering the target tasks and environment models based on apriori and current information. The designed adaptive algorithms may be used in many applications for mobile robots and vehicles of various types including unmanned aerial vehicles.

Chapter 8 presents the innovations in avionics solutions aimed to enhance a flight visibility and a situational awareness of a flight crew. Such solutions are based on Enhanced Vision System (EVS), Synthetic Vision System (SVS), and Combined Vision System (CVS) [21]. These systems provide a supplemental view of external cabin space for a flight crew using technical vision, computer graphics, and augmented reality [22]. The chapter addresses the main aspects of the EVS/SVS/CVS technologies development and includes the main topics such as the EVS/SVS/CVS typical applications, the overview of well-known commercial and experimental systems, the cores of advanced EVS/SVS/CVS technologies, among others. The EVS generates an enhanced image of external cabin space in a real time. The main EVS functions connect with image enhancement, image fusion from visible and infrared ranges, interconnections of multispectral visual information, creation of superresolution image by using a set of low resolution frames, video stabilization, automatic binding and visual combination of enhanced image with a flight symbology, automatic runway and obstacles detection in the landing zone and taxiing, etc. The digital terrain modeling based on a photogrammetric processing of 2D/3D data and a synthesized image creation based on navigation, terrain, and obstacles layers fusion are the main tasks of the SVS [23]. The representation of topological map as vector graphical patterns of flight corresponding to real environment is produced by the on-board computer equipment. It is needed to create 3D view of external cabin space with enough scene depth to sense a relative distance to real objects by a flight crew. The automatic recognition and the flight symbols representation of potentially dangerous events are also provided by the SVS. The CVS binds and integrates the sensory and geospatial information to implement a human-machine interaction based on virtual and augmented realities. Such enhanced images are shown on the primary flight display. The chapter contains some examples of the EVS, the SVS, and the CVS images, which have been obtained during the research and development program initiated by Russian State Research Institute of Aviation Systems “GosNIIAS”.

Chapter 9 discusses the accurate in-situ position estimation of navigation system for an Autonomous Underwater Vehicle (AUV). Among the different navigation principles, acoustic and vision-based ones became the most popular [24]. The acoustic navigation uses the Time-Of-Flight (TOF) measurements of ultrasound waves propagating from a stationary buoy or a set of buoys with known location to a hydrophone, which is maintained on AUV. The vision-based navigation is based on the analysis of snapshots series receiving from an on-board optical camera. The operating range of the acoustic systems is typically from 1 m up to 10 km, while the working range of the optical navigation is reduced to several cm. A vehicle mission is to take a certain orientation in the space relatively to an underwater target. An operation model includes two steps called as a long-distant and a near-distant guidance. The long-distant guidance is based on the TOF measurements of acoustic signals, which can be one-way synchronous or one-way asynchronous signals from an acoustic buoy or two-way asynchronous guidance, when the AUV transmits a pilot signal, which is replicated by the buoy and is registered back on the vehicle side. The one-way asynchronous guidance was selected. Such approach does not require any synchronization and provides a possibility to navigate multiple AUVs. In general, the long-distant guidance can be viewed as a problem of time delay minimization between the transducer (acoustic buoy) and the receiver (AUV’s hydrophone) [25]. This problem is divided into two sub-tasks including the accurate TOF measurements and the estimations translation into engine control signals.

Addressing the first task, a cross-correlation technique based on pseudo-noise sequences is applied. For the second task, a Proportional-Integral-Derivative (PID) controller is used. The further positioning is continued by an image analysis for series of digital snapshots. The near-distant guidance aims to improve the position of the AUV and justify its course in accordance to an underwater target [26]. An image processing algorithm computes a lateral transition, rotation, and scaling between a snapshot and a stored target sample. An input snapshot is convolved with a Gaussian kernel in order to reduce an image noise and remove the non-significant details. The normalized gradient image is transformed to binary image by a predefined threshold. A target center is calculated as a center of mass. Then the unknown angular difference and scale factor are estimated in a log-polar system. The total computational complexity of the developed algorithm is $O(MN \log(\max(M, N)))$, where $M * N$ is the number of pixels in the digital image. The PID controller is used to manipulate the position and orientation of the AUV during a near-distant guidance mode. In order to evaluate an accuracy and robustness of the developed model, a 3D simulator was implemented, where the series of numerical experiments with different underwater targets was carried out. The experiments have shown a high accuracy of the proposed approach, which may be successfully used in real AUV navigation system.

Chapter 10 examines various digital image filtration algorithms [27], which are useful in great variety of video devices – cameras, mobile phones, scanners. These algorithms eliminate the distortions and other blurring effects and improve "raw" images for further specific applications. Among the most known filtration models are an Additive White Gaussian Noise (AWGN) model and a mixed noise model. As the AWGN model is suitable for description of effect of multiple noise sources caused by digital devices. A mixed noise model describes better a Complementary Metal-Oxide Semiconductor (CMOS) matrix noise. Therefore, these models were taken as the primary for investigations. For denoising purposes, algorithms based on Principal Component Analysis (PCA) and non-local processing were used [28]. The idea of the PCA is in a change of image representation in order to reduce the data dimensionality with minimizing of the mean square error. The core functions are a block representation of image, a blocks' transform into a set of the PCA coefficients, a coefficients processing, and an image reconstruction. The non-local processing is connected with the evaluation of a certain image's pixel using an average of all weighted values of pixels in a neighborhood. The calculation of pixel weight is based on a degree of similarity between neighborhoods in a noisy and denoisy images. However, most of the modern filtration algorithms implemented for grayscale or color images cannot be directly applied for "raw" (color filter array) images having the dependencies between color components. In addition, each of them possesses its own pros and cons in terms of image reconstruction quality. The main problems of the quality of reconstructed images, which researchers try to evaluate, are a Gibbs effect, which becomes highly noticeable in images containing objects with a high brightness contrast in outer edges, and an edge blurring in images. Both of these effects highly degrade an image perception and could not be suited for high demands [29]. The chapter provides the analysis of such effects and their compensation in a filtration stage. The listed features were formulated in order to implement a series of denoising algorithms, each of those was specifically designed to achieve a high image quality. They can be applied in multimedia transmission systems, digital video broadcasting, pattern recognition, object tracing, and other practical applications. The recommendations of further development and limitations of use are situated at the end of chapter.

Chapter 11 provides the methods of image segmentation in systems for monitoring of the Earth surface, disease diagnosis, and safety of various large technical objects [30, 31]. A complex inhomogeneous background in image and sometimes a low signal-to-noise ratio do not allow to apply the simple solutions. It is proposed to use the mathematical theory of conditional Markov processes, the representation of g-bit grayscale images as a set of g-binary planes, and the entropy approach for calculation the state elements probabilities [32]. This approach has allowed to develop the novel efficient methods of contour and texture segmentation for objects of interest in images. For each element, certain information content is calculated to detect the contours of objects. By comparison the obtained values with a predetermined threshold, one can decide, the given point belongs for a contour or not. The developed method of contour segmentation using the calculated value of information content detects the objects of interest with a high accuracy and requires significantly less computational resources than conventional methods (Canny, Laplacian of Gaussian, Roberts, Prewitt, and Sobel). To detect the object contours in image transmitted over a noisy radio channel, it is necessary the efficient filtering of images. For image restoration distorted by the white Gaussian noise, it is proposed to use 2D nonlinear filtering algorithm. Having more accurate estimates of the image’s elements states, and taking into account the transitions probability between elements, it is possible to outline the edges of objects of interest. The proposed method is effective under the signal-to-noise ratio for an input of receiver device up to 9 dB. Also the method of texture segmentation has been proposed for the determination of extensive areas with similar statistical characteristics of satellite images such as areas of the forest, the areas of urban developments, etc. The novel method is based on the calculation of average transition probabilities in the image’s elements using a slicing window. Experimental results confirm that the developed algorithms for objects and texture areas detection in images are effective in terms of quality and processing speed.

1.3 Conclusion

All included chapters contain the innovative decisions in computer vision for control and surveillance systems. To receive the real-time implementations is the main goal for close to practice tasks. The efforts direct on the development of robust, exact, and fast methods and algorithms. Many of represented works have an experimental software implementation as a result of previous many-years researches. The chapters include the recent methodologies for human action recognition, real-time audience analysis system, panorama construction from multi-view cameras in outdoor scenes, real-time applications in robot navigation and intelligent control, adaptive surveillance algorithms, enhanced, synthetic and combined vision technologies for civil aviation, navigation of autonomous underwater vehicles, efficient denoising algorithms for intelligent recognition systems, image segmentation based on 2D Markov chains. Each chapter explains in detail algorithmic and software/hardware implementations in the chosen area of researches.

References

1. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput. Surv. 43(3): 16:1–16:43
2. Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: European conference on computer vision (ECCV 2006), pp 428–441
3. Amraji N, Mu L, Milanova M (2011) Shape–based human actions recognition in videos. In: 14th international conference on human-computer interaction: design and development approaches, vol. 1, pp 539–546
4. Li SZ, Jain AK (2005) Handbook of face recognition. Springer, Berlin
5. Makinen E, Raisamo R (2008) An experimental comparison of gender classification methods. Pattern Recogn Lett 29(10):1544–1556
6. Fu Y, Huang TS (2010) Age synthesis and estimation via faces: a survey. IEEE Trans Pattern Anal Mach Intell 32(11):1955–1976
7. Khryashchev V, Ganin A, Golubev M, Shmaglit L (2013) Audience analysis system on the basis of face detection, tracking and classification techniques. In: International multi-conference of engineers and computer scientists (IMECS 2013) 1:446–450
8. Haenselmann T, Busse M, Kopf S, King T, Effelsberg W (2009) Multi perspective panoramic imaging. Image Vis Comput 27(4):391–401
9. Kwon OS, Ha YH (2010) Panoramic video using Scale Invariant Feature Transform with embedded color-Invariant values. IEEE Trans Consum Electron 56(2):792–798
10. Favorskaya M, Pakhirka A (2012) A way for color image enhancement under complex luminance conditions. In: Watanabe T, Watada J, Takahashi N, Howlett RJ, Jain LC (eds) Intelligent interactive multimedia: systems and services. Springer, Berlin
11. Zhao G, Lin L, Tang Y (2013) A new optimal seam finding method based on tensor analysis for automatic panorama construction. Pattern Recogn Lett 34(3):308–314
12. Bonin-Font F, Ortiz A, Oliver G (2008) Visual navigation for mobile robots: a survey. J Intell Robot Syst 53(1):263–296
13. Kiy KI, Dickmanns ED (2004) A color vision system for analysis of road scenes. In: IEEE intelligent vehicle'04 symposium, pp 54–59
14. Kiy KI (2010) A new real-time method for description and segmentation of color images. Pattern Recogn Image Anal Adv Math Theory Appl 20(2): 169–176
15. Jarvis R (2008) Intelligent robotics: past, present and future. Int J Comput Sci Appl Technomathematics Res Found 5(3):23–35
16. Bekiarski Al, Pleshkova Sn (2009) Microphone array beamforming for mobile robot. In: 8th WSEAS international conference on circuits, systems, electronics, control and signal processing (CSECS'2009), pp 146–149
17. Dehkharghani SSh, Bekiarski Al, Pleshkova Sn (2012) Application of probabilistic methods in mobile robots audio visual motion control combined with laser range finder distance measurements. In: Biolek D, Volkov K, Ng KM (eds) Advances in circuits, systems, automation and mechanics. WSEAS Press, Greece

18. Tulum K, Durak U, Yder SK (2009) Situation aware UAV mission route planning. In: IEEE aerospace conference, pp 1–12
19. Osipov GS, Smirnov IV, Tikhomirov IA (2012) Formal methods of situational analysis: experience from their use. Autom Doc Math Linguist 46(5):183–194
20. Leishman RC, McLain TW, Beard RW (2014) Relative navigation approach for vision-based aerial GPS-denied navigation. J Intell Rob Syst 74(1–2):97–111
21. Bailey RE (2012) Awareness and detection of traffic and obstacles using synthetic and enhanced vision systems. NASA technical memorandum, 2012-217324 NASA, pp 54–60
22. Kumar SV, Kashyap SK, Kumar NS (2014) Detection of runway and obstacles using electro-optical and infrared sensors before landing. Defense Sci J 64(1):67–76
23. Vizilter Yu, Zheltov SY (2012) Geometrical correlation and matching of 2D image shapes. ISPRS Ann Photogrammetry Remote Sens Spat Inf Sci 1–3:191–196
24. Sangekar M, Thornton B, Ura T (2012) Wide area seafloor observation using an autonomous landing vehicle with adaptive resolution capability. Oceans 2012:1–9
25. Burdinsky IN (2012) Guidance algorithm for an autonomous unmanned underwater vehicle to a given target. Optoelectron Instrum Data Process 48(1):69–74
26. Bezruchko F, Burdinky I, Myagotin A (2011) Global extremum searching algorithm for the AUV guidance toward an acoustic buoy. IEEE OCEANS'2011, pp 1–7
27. Buades A, Coll B, Morel JM (2005) A review of image denoising algorithms, with a new one. Multiscale Model Simul 4:490–530
28. Katkovnik V, Foi A, Egiazarian K, Astola J (2010) From local kernel to nonlocal multiple-model image denoising. Int J Comput Vision 86(8):1–32
29. Priorov A, Tumanov K, Volokhov V, Sergeev E, Mochalov I (2013) Applications of image filtration based on principal component analysis and nonlocal image processing. IAENG Int J Comput Sci 40(2):62–80
30. Martin D, Fowlkes C, Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 26(5):530–549
31. Wang H, Dong Y (2008) an improved image segmentation algorithm based on otsu method. In: International symposium on photoelectronic detection and imaging: related technologies and applications, SPIE 6625, pp 1–8
32. Petrov EP, Trubin IS, Medvedeva EV, Smolskiy SM (2013) Mathematical models of video-sequences of digital half-tone images. In: Atayero AA, Sheluhin OI (eds) Integrated models for information communication system and networks: design and development. IGI Global, Hershey

Chapter 2
Human Action Recognition: Contour-Based and Silhouette-Based Approaches

Salim Al-Ali, Mariofanna Milanova, Hussain Al-Rizzo and Victoria Lynn Fox

Abstract Human action recognition in videos is a desired field in computer vision applications since it can be applied in human computer interaction, surveillance monitors, robot vision, etc. Two approaches of features are investigated in this chapter. First approach is a contour-based type. Four features are investigated in this approach such as Cartesian Coordinate Features (CCF), Fourier Descriptors Features (FDF), Centroid-Distance Features (CDF), and Chord-Length Features (CLF). The second approach is a silhouette-based type. Three features are investigated in this approach such as Histogram of Oriented Gradients (HOG), Histogram of Oriented Optical Flow (HOOF), and Structural Similarity Index Measure (SSIM) features. All these features are simple to compute, efficient to classify, and fast to calculate. Therefore, these features demonstrate a promising field for human action recognition. Moreover, the classification is achieved using two classifiers: K-Nearest-Neighbor (KNN) and Support Vector Machine (SVM). The experimental results demonstrated that these features have a promising potential and useful for the human action recognition in videos.

S. Al-Ali (✉) · M. Milanova
Department of Computer Science, University of Arkansas at Little Rock,
2801 S. University Avenue, Little Rock, AR 72204, USA
e-mail: sgsaeed@ualr.edu

M. Milanova
e-mail: mgmilanova@ualr.edu

H. Al-Rizzo
Department of System Engineering, University of Arkansas at Little Rock,
2801 S. University Avenue, Little Rock, AR 72204, USA
e-mail: hmalrizzo@ualr.edu

V.L. Fox
Department of Applied Science, University of Arkansas at Little Rock,
2801 S. University Avenue, Little Rock, AR 72204, USA
e-mail: vlfox@ualr.edu

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_2

Keywords Human action recognition · Contour-based features · Silhouette-based features · K-Nearest-Neigbor · Supprt Vector Machine · Image and video understanding · Machine learning · Data mining

2.1 Introduction

Currently, computer application fields are playing significant role in multiple aspects of our lives. One important field is a computer vision, which has received a lot of attention during the past three decades due its wide applications. The human action recognition is an important goal of research on computer vision and image processing. Identifying, annotating, recognizing, and clustering human actions in videos have captured more and more attention because of its useful applications that support many different applications such as human–computer interaction, robot vision machine, human surveillance monitoring system, multimedia indexing and retrieval, entertainment environments, and healthcare systems [1, 2, 3].

The human action recognition in videos is a computer method for recognizing and identifying, what kind of action is happening in videos. In order to design and implement this program, there are many challenges such as foreground object, background scene, and camera setting. The foreground object, which is the human in this case, has many variations such as size, colour, shape, static or moving object, etc. The background scene, which is a whole image in a frame except the foreground object, has many variations such as lighting, occlusion, cluttered, static or moving background scene (based on camera setting). The camera setting is an important factor in the human action recognition because it has its own recording variations such as static or moving in all (left, right, up, or down) directions, zooming (in or out), speeds of recording (slow or high), recording types (2D or 3D), colors types in recording videos (black/white, colored, or grayscale color), etc. Moreover, the same action is performed in different ways by the same person, for example, the speed of walking is different although that the walking action is for the same person. Another challenging problem, which is more difficult and very realistic, is that the same action performed by different people. Although of these challenges, the human action recognition in videos is desired and required for many computer vision applications.

In recognizing human actions in videos, many researches have been reported in this field as shown in the survey paper [4], however, there still need to improve and develop new effective approaches. The presented chapter is a new investigation of two main features approaches (contour-based and silhouette-based) for human action recognition in videos.

In this chapter, main structure of human action recognition is defined. The structure mainly consists of three stages: human object tracking, feature extraction, and action classification. Some examples regarding literature researches of these three stages and the recent related works are given in Sect. 2.2. Subsequently, background subtraction [5, 6] is explained in detail as an example for stage of human object tracking in videos in Sect. 2.3. Next, two approaches: contour-based and silhouette based for feature extraction from the tracked human object are described in Sects. 2.4 and 2.5, respectively. The final stage in the human action recognition is action classification stage that used to classify and identify the action happening in human action testing video. Two classifiers: KNN [7, 8, 9], and SVM [10, 11, 12, 13] are used as examples for action classification stage. More details about the classifiers and their based techniques are explained in Sect. 2.6. Two modes (training and testing) of the presented algorithm for human action recognition are described in Sect. 2.7. Experimental results are discussed in Sect. 2.8. Finally, Sect. 2.9 conducted the conclusion.

2.2 Human Action Recognition in Videos

The main goal of human action recognition in videos is to identify the unknown actions happening in these videos. This goal is achieved by analyzing the frames of these videos to form and build a series of discriminant features that can be classified efficiently in term of accuracy, speed, and simplicity. The main structure of the human action recognition consists of three main stages: human object tracking, feature extraction, and action classification. The first stage has to answer the question of how to detect or segment and track the human object in each frame of the video sequences. The second stage has to answer the question of how to extract, represent, and then build feature vector from the tracked human object that result from the first stage. The third stage has to answer the question of how to classify extracted features from the second stage by applying an effective classification algorithm. Sometimes, this stage supported by data mining process to reduce dimensionality of the extracted features. In the next sections, answers and details about these three stages will be provided. The main structure of a human action recognition system is depicted in Fig. 2.1.

Section 2.2.1 addresses the human object tracking. The issues of feature extraction and action classification are discussed in Sects. 2.2.2 and 2.2.3, respectively. A literature review about human action recognition in Weizmann dataset is represented Sect. 2.2.4.

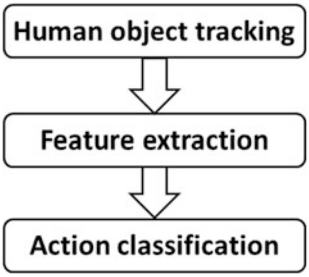


Fig. 2.1 Main structure of the human action recognition

2.2.1 Human Object Tracking

The human object tracking is a process of tracking a human object moving over sequence (time) of digital images (frames) in videos [14]. Generally, this process consists of two components: frame processing (local) and video processing (global). The first is achieved by the human object detection or segmentation. The human detection is the process of locating a human object in a frame of video. The segmentation of human object is the process of partitioning a frame into multiple segments (areas or sets). One of these separated segments represents the human object. Both detection and segmentation are mainly related to one frame (digital image) in videos and, therefore, are called frame processing. The second component is achieved by applying frame processing over all-frames (video) or sub-frames (sub video), thus, it is called video processing.

During the past three decades, many researchers solved problem of tracking objects in still image, in a frame, or videos. These solutions are achieved by several ways: point detection, image segmentation, and background modeling. First, the point detection is used for tracking based on some interesting points such as corners, or intersection points such as Harris detector [15], Scale-Invariant Feature Transform (SIFT) [16], affine invariant interest point detector [17], kernel-based object tracking [18], and Kanade-Lucas-Tomasi (KLT) detector [19]. Second, the image segmentation is a process to partition a digital image (frame) into multiple segments (sets of separated areas), used to track an object such as mean-shift [20], graph-cut [21], and active-contours [22]. Third, the background modeling is also another process used in the tracking. The goal is to obtain and build a model for the background scene. Then, the object extraction is achieved by subtracting each frame from this model, such as running Gaussian average [23], temporal median filter [24, 25], Mixture Of Gaussian (MOG) [26, 27], eigenbackground [28], and dynamic texture background [5]. More details and examples for human object tracking using a background subtraction in Weizmann human action dataset [2] are explained in Sect. 2.3.

2.2.2 Feature Extraction

The feature extraction is a process of extracting a set of features to represent some useful measurements or characteristics of a frame or video. These features are computed carefully from a frame, sub-frames, or all-frames in video efficiently in order to capture most important meaningful details. The goal of feature extraction is to provide a classifier by good feature in terms of accuracy and speed. This goal can be achieved by two ways: minimizing feature details as much as possible and at the same time maximizing features discrimination in order to increase accuracy and speed of classification in the next stage. There are several ways to enhance the feature extraction [29]. First, extracting spatial information is more related to frame details (coordinates, shape, size, texture, etc.). Second, extracting temporal information is more related to video details (motion, time, derivatives, etc.). Third, eliminating redundancy in information of both frame and video in order to reduce valueless features since much of information is of little or no value. Finally, integrating more than one kind of features together to get better performance in analysis and classification.

Most features are extracted from a shape, which represents both contour and silhouette of the tracked object implicitly. Moreover, this shape is used in template matching and human action recognition by some researches [30, 31].

In the past three decades, feature extraction or detection attracted the attention of researchers due to its useful applications [32]. For example, a content-based video retrieval [32] is the ability to recognize actions correctly that leads to automatic annotation of huge video dataset. Different types of features that have been extracted from human object are investigated for human action recognition purposes. In this chapter, the features used to recognize a human action in videos are categorized mainly into two types: contour-based and silhouette-based. On the one hand, the contour-based features are mainly obtained from boundary points that surround silhouette of human object. While on the other hand, the silhouette-based features are mainly obtained from whole region body of silhouette for human object. More details and examples about contour-based feature and silhouette-based features extraction are explained in Sects. 2.4 and 2.5, respectively.

2.2.3 Action Classification

The final step in any recognition is to feed the features into a classifier, which is adopted one of the classification algorithms, for example, such as K-Nearest-Neighbor (KNN) [7, 8, 9], Support Vector Machine (SVM) [10, 11, 12, 13], Adaptive boosting (Adaboost) [33], Hidden Markov Model (HMM) [34, 35], Artificial Neural Network (ANN) [36, 37], etc. More information about some of these algorithms is explained in Sect. 2.6.

Generally, the goal of all these algorithms is to classify extracted feature in testing video sample (testing mode) and identify its class membership or its closest neighbor based on features that conducted from training video samples (training mode). Thus, a recognition classifier has to be trained using the training observations. There are three types of learnings based on the labeled and non-labeled classes of the observations such as supervised, unsupervised, and semi-supervised (self-supervised) learnings [38, 39]. The supervised machine learning [39] is defined such that all training examples (observations) are labeled into classes, thus the system (machine) is trained with feature observations and their class labels. Thus, the goal is to find a membership class (classify into one class of the trained classes) for any a given testing example. Naturally, the supervised learning is used in most classification algorithms. In unsupervised machine learning [38, 39], the given training observations are not labeled into classes, thus the system is trained with only feature observations. Therefore, the system has two goals: first one is to cluster and find the classes for training examples and second goal is to find a membership class after clustering all training examples for a given testing example. Thus, the unsupervised machine learning is more complicated than the supervised learning. Naturally, the unsupervised learning is used in clustering but not in classification. In the semi-supervised (self-supervised) [39, 40], some of a given training observations are labeled while others not. This learning is a combined between both previous types, and it is located in middle between supervised and unsupervised learning in term of difficulty.

Moreover, in order to evaluate computed results statistically cross-validation is used. It is a technique to assess results based on a statistical analysis. The cross-validation is the estimation for accuracy of a model or an algorithm based on how to use the dataset during training and testing modes. Mainly, there are three techniques: 2-fold, cross-validation, K-fold cross-validation, and leave-one-out cross-validation [41]. More details and examples about classifier types and techniques are explained in Sect. 2.6.

2.2.4 Human Action Recognition in Weizmann Dataset: Literature Review

There are many approaches and methods to recognize human actions in videos over the past three decades. In this section, the recent literature review related to human action recognition in videos is presented. Despite of these methods and approaches, the human action recognition is still attractive for many computer vision researchers because still has plenty of challenges need to be solved and it is growing demands for many applications. For example, more realistic human action dataset requires for researchs in accuracy and speed challenges for building an efficient real-time human action recognition in videos.

Aggarwal and Ryoo [4] classified human action recognition into two main approaches. First, single layered approach is based on sequence of images to describe human actions. Second, the hierarchical approach is applied on more than one of single layered approach. On the one hand, the single layered approach is classified based on model into space-time, and sequential approaches. On the other hand, the hierarchical approach is classified based on their methodology into statistical, syntactic, and description approaches. The space-time approaches are classified based on feature types into space-time volumes, trajectories, and space-time features. This approach is named due that its features are computed from space (spatial) and time (temporal). In this chapter, all experiments are based on this space-time approach because both (contour-based and silhouette-based) features are extracted from Aligned Silhouettes Image (ASI), which is an accumulation of all frames in one video to form one image that captures all spatial and temporal features together.

Amraji et al. [30] presented human action recognition system based on shape of human. The shape representation is computed using Fourier Descriptors (FDs) as features. These features are projected into eigen-space by using Principal Component Analysis (PCA). The KNN is employed as a classifier for human action recognition. The Weizmann dataset [2] is used to test the system with only five actions from ten actions in the dataset. They recorded 86 % success recognition rate. In this chapter, the FDs are extracted from the contour of the ASI and used as a feature without any projection. Then, the KNN and the SVM classifiers are used. The best achieved result is 93.548 %, which is the first contribution of this chapter.

Gorelick et al. [2] employ contours of human silhouettes as space-time features for action recognition. In their algorithm, the Poisson's equation is solved based on contour coordinate points. The solution is required to compute coefficients of the Poisson's equation by using multigrid solution. Then, several properties are extracted based on these coefficients such as local saliency, action dynamics, shape structure, and orientation. These shape properties or Poisson features are employed as a sequence for a number of frames in each video action. Gorelick et al. [2] created the Weizmann dataset, which contains ten different human actions done by nine actors, applied their recognition system on these dataset. The researchers recorded 100 % correct recognition rate using all sequence of frames in videos for classification using variant median Hausdroff distance. This distance is used because of the differences among videos in term of video length in frames. It is used to find the distances between any two sequences (testing sample and each of training samples) and the action with minimum distance is identified as predicted action. Also, they achieved 97.83 % correct rate using sliding window of eight frames with jumping four step frames. They used 923 space-time cubes in their experiments. This means that a number of all frames in all videos is approximately 4,096 frames totally. In this investigation, the Weizmann dataset was downloaded from their website [2] and computed number of frames in all videos is 5,687 frames exactly. This difference means that all available frames, which effect on the recognition results, not used in their experiment. In all our experiments in this chapter, all frames are used without any exception. Therefore, the results are more relastic since built based on all available frames in all videos.

Sadek et al. used a chord-length as a feature for human action recognition. This feature demonstrates high accurate, robust, compact, and efficient results [3]. The computation of feature is based on a chord-length function as a local feature, and gravity or shape centroid motion for snippet frames as a global feature. The SVM is employed for the classification. Sadek et al. [3] applied their algorithm in Weizmann dataset. They recorded result 97.8 % correct recognition rate on Weizmann dataset. They combined more than one feature together. In this chapter, each feature is tested alone separately before combination to compare results with each other.

Dalal and Triggs [42] presented a very robust visual object recognition feature, which is called Histogram of Oriented Gradients (HOG), used effectively to recognize objects in visual method. Also, Dalal and Triggs [43] used another histogram feature, which is Histogram of Oriented Optical Flow (HOOF). By employing these features, the human object is detected. For classification, in both HOG and HOOF, a linear SVM is used. Both HOG and HOOF are mainly used in object recognition. In this chapter, both descriptors are employed as the features for human action recognition and achieved very good results in term of accuracy.

Chaudhry et al. [1] employed the HOOF and the Binet-Cauchy kernels on nonlinear dynamical systems for recognition of human actions. Four different kernels are used to measure distance between two histograms. These kernels are: geodesic, Minimum Difference of Pairwise Assignment (MDPA), chi-square, and histogram kernels. The minimum distance is selected as a prediction result for classification. Authors reported 94.4 % correct recognition rate on Weizmann dataset. In this chapter, the HOOF is employed directly as a feature and achieved 97.849 %, which is the second contribution in this chapter.

Al-Ali and Milanova [44] employed an Aligned Motion Image (AMI) as a feature for human action recognition. Each video sample is represented by an AMI. Then, the Structural Similarity Index Measure (SSIM) is used to measure the distances among these images. In the last experiment of this chapter, the SSIM is employed but as a feature for representing each video. Authors reported 98.924 % correct recognition rate.

The contributions of this chapter are the following. A novel simple algorithm for human action recognition provides very good results in term of accuracy in contour-based features. Almost optimal results are conducted in term of accuracy using silhouette-based features. The structural similarity for human action recognition is employed. Finally, a comparison among contour-based and silhouette-based feature results is presented in this chapter.

2.3 Human Object Tracking in Weizmann Dataset

The goal of human object tracking in videos is to separate a human silhouette from each its background scene of each frame in the videos. In order to achieve this goal specifically in Weizmann dataset [2], several pre-processing processes are required. The description of Weizmann human action dataset is located in Sect. 2.3.1. The following Sects. 2.3.2–2.3.7 include the representations of various types of processes such as background subtraction, detection of direction, horizontal alignment, computing of an Aligned Silhouettes Image (ASI) process, unifying direction, and cropping bounding box, respectively.

2.3.1 Weizmann Human Action Dataset

The Weizmann human actions dataset [2] is used to test our presented algorithms in this chapter. This dataset is recorded from a still (non-moving) camera, therfore

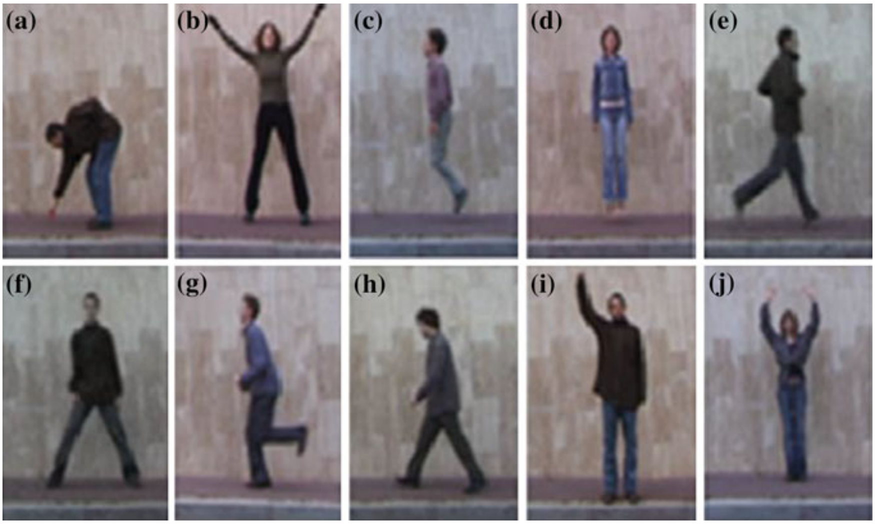


Fig. 2.2 Weizmann dataset frame example for ten different human actions: **a** bending, **b** jumping jack, **c** jumping forward, **d** jumping in place, **e** running, **f** gallop sideway, **g** skip jumping, **h** walking, **i** one hand waving, **j** two hands waving

background of scene is a still image. The only moving thing is the human object. The dataset contains 93 low-resolution (180 × 144, with speed rate 50 fps) video samples for human actions. Figure 2.2 depicts some frame examples of human action videos in the Weizmann dataset.

There are ten different human actions in this dataset such as bending, jumping jack, jumping forward, jumping in place, running, gallop sideway, skip jumping, walking, one hand waving, and two hands waving. These actions are performed by nine actors. Each actor performed each action once, except one actor ("Lena"), who performed three of the actions (running, skip jumping, and walking) twice. One action, the object is moving from left to right and other it is vice versa. The Weizmann dataset videos samples have different lengths in term number of frames recorded in each video.

2.3.2 Background Subtraction Process

Background subtraction is an example about human object tracking. It is a sort of background modeling process, which is used to separate foreground object from background scene. As mentioned above, the goal for this process is to obtain and build a model of background scene. But in some cases, the scene is already available or very easy to build especially (for example), when videos are recorded from a still (not moved) camera. Thus, there is no need to build and estimate out the

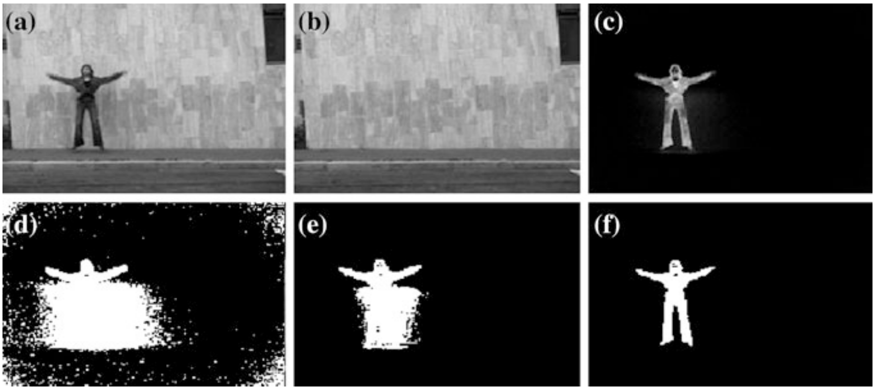


Fig. 2.3 Background subtraction process for a frame of jumping jack action in Weizmann dataset: **a** original frame in gray-scale image, **b** its background scene frame in gray-scale image, **c** background subtraction result image with noise, **d** thresholding into black and white image with noise, **e** thresholding into *black* and *white* with less noise, **f** thresholding into *black* and *white* image almost without noise, which is the result silhouette image

background model. If a background scene is available, the object tracking is simply achieved by subtracting each frame from its available background scene. Figure 2.3 shows the background subtraction process for a frame of jumping jack action in the Weizmann dataset.

At the beginning of this process, all frames in all videos and all background scenes are converted from Red Green Blue (RGB) colors into gray-scale colors. After that, frames are subtracted from their background scene. The result of subtraction is a noisy image, thus a proper threshold is used to extract the human silhouette. The final result is a logical (binary) image containing only the silhouette in white color (1's value), and the background scene in black color (0's value).

2.3.3 Direction Detection Process

The detection of direction is a process of finding movement path for human actions, which are directed from left to right or right to left. Specifically, the actions that have displacement in location, for example in the Weizmann dataset such as walking, jumping forward, skip jumping, side jumping, and running actions. Other actions in the dataset have movement displacements such as bending, jumping in place, jumping jack, waving in one hand, and waving in two hands actions. In order to obtain the movement direction for human action in silhouette videos, a contour of silhouette is obtained in the first frame of video. Then, a center point of the contour is calculated by finding mean of the contour coordinates. Next, a frame center is also calculated suing height and width of the frame. These contour and frame centers used in two purposes: first, it is used to detect displacement direction and,

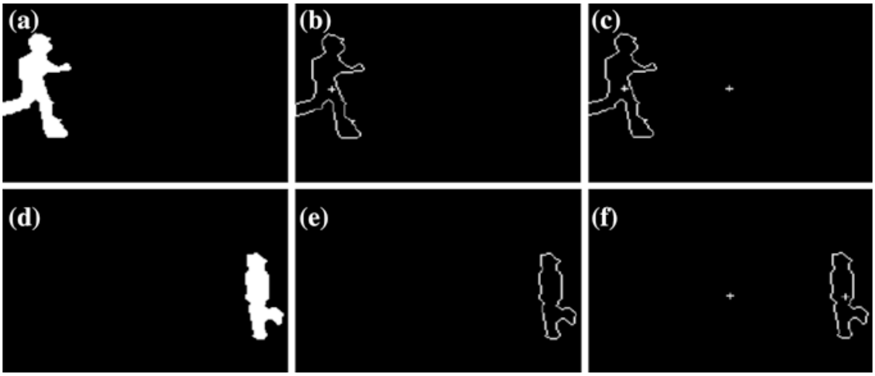


Fig. 2.4 Direction of movement detection process, (*top row*) running from *right* to *left* direction and (*bottom row*) running from *left* to *right* direction: **a**, **d** first frame of silhouette for running, **b**, **e** obtain a center of silhouette, **c** comparing silhouette center with center of frame, which means direction of movement is from *right* to *left*, **f** comparing silhouette center with center of frame, which means direction of movement is from *left* to *right*

second, it is used to align each silhouette horizontally into frame center. By comparing *x*-dim of the silhouette center with the *x*-dim of the frame center, the direction is detected. The direction of action is saved to be used later to unifying direction process. This process is important influence for actions that have displacement movement because same actions with different direction have different features but are symmetric. Figure 2.4 shows detection of direction process for two running actions with different movement directions in the Weizmann dataset.

2.3.4 Horizontal Alignment Process

This process is used for aligning all frames of silhouettes in each video horizontally. These videos are obtained as a result of background subtraction process. All frames are aligned horizontally into the *x*-dim of the frame center. Notice that here all frames are aligned into *x*-dim (horizontally) but not into *y*-dim (vertically). There are a few important advantages of this process. First, it helps to form a consistent aligned silhouette image later. Second, it solves the problem of different number of frames in each video. Finally third, it forms a very discriminant features for each video in the dataset. Figure 2.5 shows the horizontal alignment process for one of skip jumping frames.

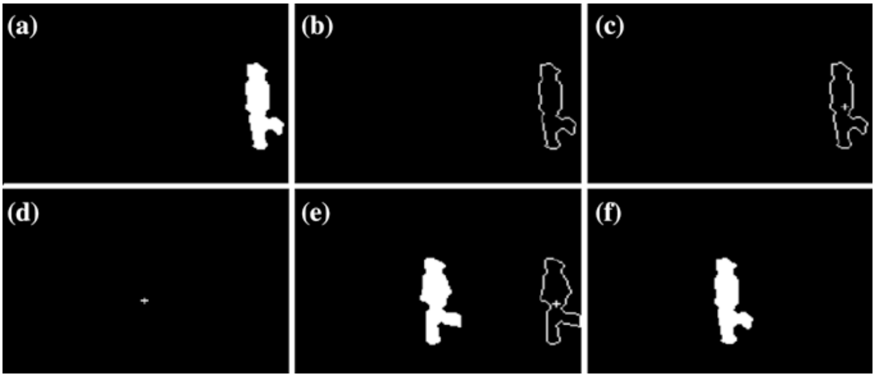


Fig. 2.5 Horizontal alignment for silhouette frame of skip jumping action in Weizmann dataset: **a** silhouette image frame, **b** contour of silhouette, **c** *center* of contour, **d** horizontal frame center, **e** align the silhouette into horizontal center of frame, **f** final result is an aligned silhouette image

2.3.5 Computing Aligned Silhouettes Image Process

This process is used to obtain an image for each video action calculated from all aligned silhouette frames to form an image that captures the most important features of the action video [44]. After completion of horizontal alignment process, all frames of silhouettes in all videos are aligned in x-dim of the frame's center. Then, the ASI is calculated for each video in the dataset. The ASI is the accumulation for silhouette images that are obtained by the summation of all binary silhouette images in each video. The pixels of the formed ASI consist of an integer value from 1 to n, where n is a number of frames in video, since each silhouette frame is a binary image. The idea of the ASI in this research work is inspired from Motion History Image (MHI) [45], Motion Energy Image (MEI) [45], and Gait Energy Image (GEI) [46].

After the ASI is computed, a thresholding (elimination) step is applied. For example, pixels that have intensity value less than 5 are eliminated and converted into value of 0, while others that have intensity value greater than or equal 5 are converted into 1 value. The contribution for this thresholding step is an improvement in accuracy of correct recognition rate. In this step, all pixels that have small value are eliminated. For example, pixel with 1 value means this pixel appears only in one frame during all frames in video. Thus, this pixel can be considered as most likely as a rarely occurred or a noise, so it is eliminated. This process has two advantages: first one is to find such images that have discriminant features for recognition and second one is to solve the problem of different number of frames in each video in the dataset. Figure 2.6 shows the ASI images for two different video actions (bending and jumping forward). The final result of computing ASI process is a binary image for each video that used in the next feature extraction stage.

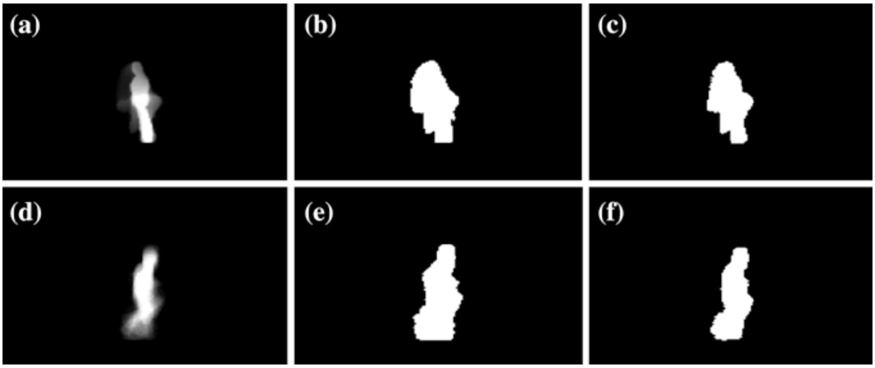


Fig. 2.6 Aligned silhouettes images for two human silhouette action videos in Weizmann dataset, (*top row*) for bending action and (*bottom row*) for jumping forward action: **a**, **d** summation of all frames, **b**, **e** binary images for the summation and thresholding equal 1, **c**, **f** binary images for the summation and thresholding equal 5

2.3.6 Unifying Direction Process

The unifying direction is a process of converting all actions that have displacement into one direction, for example, from left to right. The advantage of this process is to form more robustness feature, since it gets rid the difference in built features that resulted for same actions but opposite in direction. In Sect. 2.3.3, the direction for each video is detected. This process is used to unify the direction for all actions that have movement displacement such as jumping forward, jumping jack, skip jumping, running, and walking. Each of these actions has a direction either from left to right or from right to left. Thus, all videos with these actions are unified into one direction. The process is achieved by flipping all ASI images (over *y*-axis) from one direction to other. The unification process is not for all videos, it is only for videos with actions have displacement. This process starts after process of computing ASI because instead of unifying all frames in video separately, only one ASI image is unified. Therefore, this step is increased speed of processing.

2.3.7 Cropping Bounding Box Process

The cropping bounding box is a process for obtaining a smallest box that surrounds all silhouette pixels for the ASI, then crop this box area. This process starts by taking each binary ASI and traces its region boundaries. The trace is achieved by using logical OR operation for rows and columns of the ASI. Each row or column will have 0 logical value means it does not have any pixel of silhouette. By this operation, the bounding box is detected.

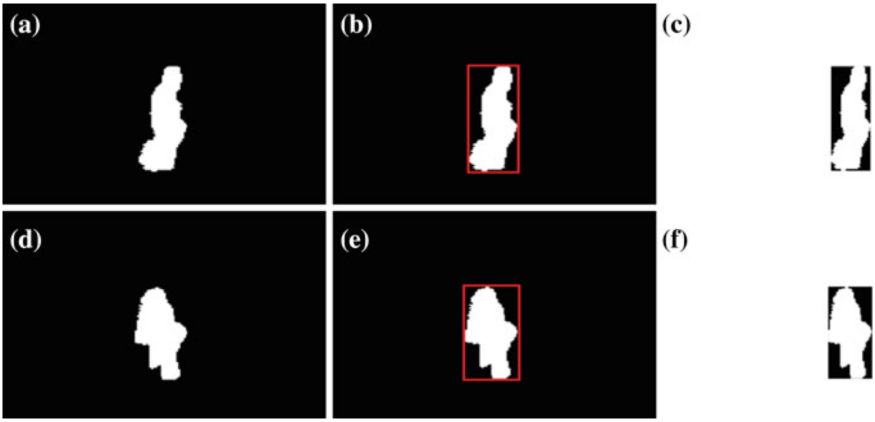


Fig. 2.7 Cropping bounding box process for bending and jumping forward actions in Weizmann dataset: **a**, **c** original ASI images for two actions, respectively, **b**, **d** bounding box (in *red color*) for their silhouettes in ASI images, **c**, **f** cropped (separated) bounding box regions

Then, by tracking first one (1) logical value in rows (from top to down and vice versa) and columns (from left to right and vice versa), bounding box will be detected. As a result, the bounding box of the ASI image is obtained. The bounding has two points in 2D coordinates, which are the left-top and right-bottom of the silhouette. Figure 2.7 shows the cropping bounding box process for two actions in Weizmann dataset.

2.4 Contour-Based Feature Extraction

Contour-based features are directly extracted from contour boundary coordinate points surround silhouette of the ASI image. There are many types of contour-based features such as the Cartesian Coordinate Feature (CCF), the Fourier Descriptor Feature (FDF) [47, 48, 49], Centroid-Distance Feature (CDF) [50, 51], and Chord-Length Feature (CLF) [3, 50, 51]. All these contour-based features are normalized between 0 and 1 values, before being used in the action classification stage.

All these types of contour-based features are explained in Sects. 2.4.1–2.4.4, respectively.

2.4.1 Cartesian Coordinate Feature

The CCF is represented by Cartesian 2D coordinate function. Each coordinate consists of two numbers (x, y) generated from each point on the boundary points of contour. In order to obtain boundary coordinate points, *bwboundaries*, which is a

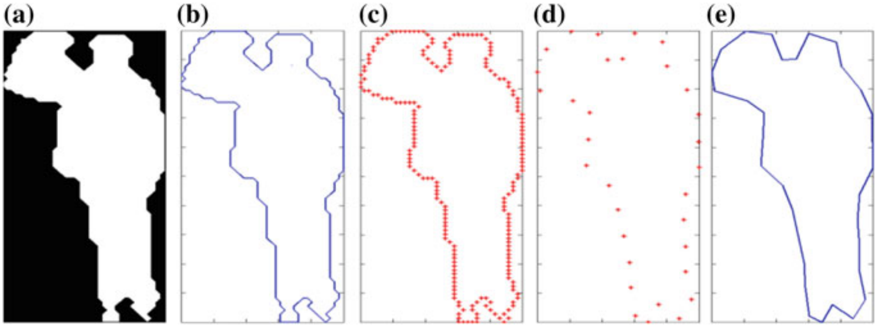


Fig. 2.8 Cartesian coordinates and Fourier descriptors of the ASI for one hand waving action in Weizmann dataset: **a** an ASI image, **b** contour of an ASI image, **c** Cartesian coordinate of the contour, **d** 32 Fourier Descriptors (FDs), **e** plotting of these 32 FDs

function in Matlab [52], is used to trace boundary. The result of this CCF function, which is N Cartesian coordinate points, is shown in Eq. 2.1, where t is an integer, $t \in [1..N]$, N is a number of points on boundary of contour, (x, y) are boundary coordinate points in 2D space, and (x_c, y_c) is a center of gravity for boundary points.

$$CCF(t) = [[x(t) - x_c], [y(t) - y_c]] \tag{2.1}$$

Figure 2.8a–c depict process of obtaining Cartesian coordinates features for one hand waving action in Weizmann dataset.

Until this end, each video will have different length in terms number of boundary points for each ASI image in 2D space. Thus, these different length numbers have to be set into one equal length number. The equalization can be achieved using interpolation [53, 54, 55] or using Fourier Descriptors (FDs) [47, 48, 49]. The interpolation is a method used to unify the number of boundary points for the contour of the ASI image. This method is achieved by constructing or estimating some unknown boundary points based on the known surrounding boundary points; the unknown point values are usually within a range between known values. There are two main types of interpolation based on how to use the known data (boundary points). First type is the global interpolation that employs all the boundary point values to find the unknown values. The second type is the local interpolation that employs a fixed number of known nearest boundary point values.

2.4.2 Fourier Descriptor Feature

The FDs are based on Discrete Cosine Transform (DCT) [47], which is a mathematical operation function for converting time domain into frequency domain. A surprising and importance feature of the FDs is ability to represent any 2D closed shapes independent of their location (translation), scaling, rotation, and starting point [47, 49]. Therefore, the first motivation of using the FDs is due its properties. The second motivation is to unify number points representing each shape boundary for all frames in videos. In short, the FDs are used to describe the contour (boundary) of any closed contour in 2D space based on the DCT methods. The FDs are presented by Eq. 2.2, where z is a complex number function, $[x(t), y(t)]$ are Cartesian boundary points of contour in 2D space, t is an integer such that $t \in [1, N]$, N is a number of points on boundary, and symbol (i) refers to imaginary part of the complex number.

$$z(t) = x(t) + iy(t) \tag{2.2}$$

The FDs function F, based on z, can be calculated using the DCT function [47] from Eq. 2.3, where k is an integer such that $1 \leq k \leq N$, e is the exponential function.

$$F(k) = \frac{1}{N} \sum_{t=1}^{N} z(t) e^{-\frac{j2\pi tk}{N}} \tag{2.3}$$

The expression $z(t)e^{-j2\pi tk/N}$ can be computed from Eq. 2.4, where $p = 2\pi tk/N$, and other parameters are defined above.

$$z(t)e^{-jp} = [x(t) \cdot \cos(p) + y(t) \cdot \sin(p) - x(t) \cdot j \cdot \sin(p) + y(t) \cdot j \cdot \cos(p)] \tag{2.4}$$

It is obvious from Eq. 2.4 that complex numbers are transformed into a linear combination of sins and cosines curves in the frequency domain. In order to reconstruct function $z(t)$, the inverse of the DCT, which is $z'(t)$ based on $F(k)$, is provided by Eq. 2.5.

$$z'(t) = \sum_{k=1}^{N} F(k) e^{\frac{j2\pi tk}{N}} \tag{2.5}$$

However, the approximation of z can be reconstructed by using the function $z'(t)$ with less number of Fourier coefficients such that$1 \leq k \leq p$ and $p < N$. This approximation is useful to unify the number of points for all contours. The reconstructed points by using part or all coefficients are known as the FDF. Figures 2.8d, e show the FDFs that reconstructed from 32 Fourier coefficients and their plots in 2D space, respectively. It is obvious that a plot of contour based on the FDF coordinates, which is shown in Fig. 2.8e, is very similar to the plot of the contour based on all original Cartesian coordinates, which is shown in Fig. 2.8b. The main difference between these two figures is that first figure captures all (low and high) details of the contour, while second figure captures only most important (high) details and ignoring other (low) details.

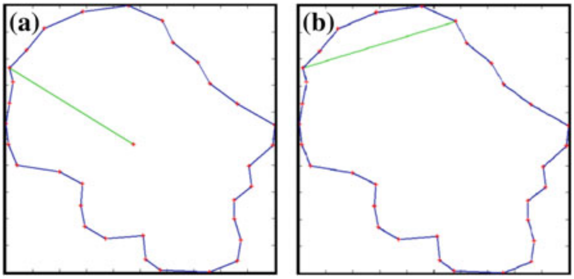


Fig. 2.9 Contour-based features of the ASI for bending action in Weizmann dataset: **a** the CDF between centroid point and a point on boundary of contour, **b** the CLF with jump displacement (w = 5) between two points on boundary of contour

2.4.3 Centroid-Distance Features

The CDFs are features calculated by obtaining the distance (magnitude) between each boundary point based on the FDF and centroid point of the contour in 2D space. In this case, the centroid-distance is calculated from Eq. 2.6, where $CDF(t)$ is a centroid-distance function, $[x(t), y(t)]$ are the FDF coordinates, and x_c, y_c are centroid point coordinates of contour.

$$CDF(t) = \left[(x(t) - x_c)^2 + (y(t) - y_c)^2\right]^{1/2} \tag{2.6}$$

Figure 2.9a depicts one distance from the CDF for the ASI contour of a bending action in Weizmann dataset.

2.4.4 Chord-Length Features

The CLFs are features calculated by obtaining length (magnitude) between two points on boundary based on the FDF points of contour. There is a fixed length in term of the number of points (jump displacement step), denoted by w, separating these two points. These features are calculated from Eq. 2.7, where $CLF(t)$ is a chord-length function, and w is an integer represents jump displacement step.

$$CLF(t) = [(x(t) - x(t+w))^2 + (y(t) - y(t+w))^2]^{1/2} \tag{2.7}$$

In Fig. 2.9b, the CLF feature is depicted with jump displacement step (w = 5) for the contour of the ASI for a bending action in the Weizmann dataset.

2.5 Silhouette-Based Feature Extraction

The silhouette-based features are directly extracted from silhouette. The silhouette is a whole body region inside contour of human object. There are several types of silhouette-based features such as Histogram Of Gradient (HOG) [42, 43], Histogram Of Optical Flow (HOOF) [1], and Structural Similarity Index Measure (SSIM) [56]. All these silhouette-based features are normalized, before being used in the action classification stage. The following Sects. 2.5.1–2.5.3 are devoted for extraction the HOG, the HOOF, and the SSIM features, respectively.

2.5.1 *Histogram of Oriented Gradient Feature*

The HOG is a feature used to capture occurrences of gradient orientation of pixels in overlapping windows of an image. The computation of the HOG is based on magnitudes and angles of these gradients [42, 43]. This feature is extracted from image based on two parameters: number of overlapping windows on this image ($N \times N$), and number of bins (B) for the gradients angles. Briefly, the HOG is computed through several steps. The gradients of an image are computed by filtering this image with horizontal kernel [–1, 0, 1] and vertical kernel $[-1, 0, 1]^{-1}$. Then magnitudes and angles are computed based on the computed gradients. Next, the image is separated into $N \times N$ overlapping windows. For each window, angles are binned into B orientation bins based on their angles' values. For each bin, sum of gradient magnitudes is calculated. After that, these sums, which are equal to the number of bins for each window, are normalized. At the end, $N \times N \times B$ normalized numbers are obtained. These numbers are called the HOG feature descriptors for the image. Figure 2.10a shows the HOG binned orientation of gradient angles into 8 bins.

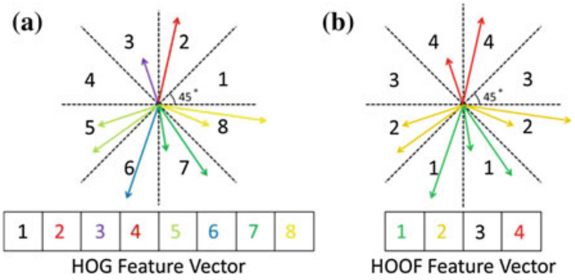


Fig. 2.10 Building histograms based on binned orientation, the angles of gradients control bins location of magnitudes (*colored arrows*), while magnitudes of gradients control lengths of these *arrows*: **a** the HOG with 8 bins, **b** the HOOF with 4 bins, each symmetric angles are binned in one orientation

2.5.2 Histogram of Oriented Optical Flow Feature

The HOOF feature is used to capture optical flow of motion in an image based on the gradient orientation of pixels' intensities in an image [1, 57]. The computation of the HOOF is also based on magnitudes and angles of these gradients. Briefly, the HOOF is computed through several steps. The gradients of image are computed as the same as the HOG by filtering an image with two kernels: horizontal kernel [–1, 0, 1] and vertical kernel $[–1, 0, 1]^{-1}$. Then angles and magnitudes are computed for these gradients. Next, all symmetric angles over y-axis are binned into B orientation bins based on the values of these angles. For each bin, a sum of gradient magnitudes is calculated. These sums, which are equal to number of bins, are normalized. At the end, B normalized numbers are obtained. These numbers are called the HOOF feature descriptors of image. Figure 2.10b shows the HOOF binned orientation of angles to 4 bins.

The HOOF is similar to the HOG but with some differences. First, the HOOF does not require for sliding windows to be overlapping, because it represents the optical flow of motion in an image, though, this image can be divided into several equal non-overlapping windows [1, 42]. Second, the HOOF is binned symmetric angles over y-axis together to overcome problem of detection movement direction while the HOG does not. Third difference is the number of feature descriptors in the HOG is $N \times N \times B$ while in the HOOF is B without sliding windows and is $N \times N \times B$ with non-overlapping sliding windows.

2.5.3 Structural Similarity Index Measure Feature

The SSIM feature is used to find an index measurement for similarity between any two (original and distorted) images [56]. This SSIM feature is first time used for human action recognition in [44]. This measurement is effective, especially when there is some difference between these two images in intensity, brightness, and contrast. The SSIM is computed based on three statistical factors (loss of correlation, luminance distortion, and contrast distortion) [44, 56]. The computation of the SSIM for image vectors (x, y) is shown by Eq. 2.8, where $x = \{x_{\,i}\, | i = 1, 2, \ldots, N\}$ are intensities of first image, N is a number of pixels in an image, $y = \{y_i | i = 1, 2, \ldots, N\}$ are intensities of second image, x and $\bar{y}$ are means of x, y respectively, σ_x^2 and σ_y^2 are Mean Square Error (MSE) of x, y, respectively, and σ_{xy} is a correlation coefficient between the x and y.

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{2.8}$$

The output of the SSIM is a quality Q with a dynamic range of [–1, 1]. When Q has 1 value means, there are full match between these two images and, when Q has –1 value means, there are significant differences between these images. The biggest positive value means the good similarity between these images, while smallest negative value means the good dissimilarity. In order to compute the SSIM features between x and y vector images, a sliding window moves over these images and the total Q is computed from Eq. 2.9, where M is a number of sliding windows and Q_j is a quality index of the jth window.

$$Q = \frac{1}{M}\sum_{j=1}^{M} Q_j \tag{2.9}$$

Briefly, there are few parameters that affect in calculation of the SSIM features [56]. First, image dimensions (*width*, *height*) are size of image in term of pixels. Second, intensity range (L) is the dynamic range for intensity values of two images. Third, tow constants ($c1$, $c2$) are small constants used in the SSIM formula to avoid division by zero. Finally forth, size of sliding window (*width, height*) is preliminary determined.

2.6 Action Classification

This is a final step in any recognition system. The action classification is based on used algorithm (method). The goal of algorithm is to classify features of testing and identify its class membership by obtaining its nearest neighbor in training samples, such as the KNN classifiers. Sometimes the goal is to train classifier about training samples in training mode and classify an unknown action of a testing sample in testing mode, such as the SVM classifiers. Moreover, these methods are divided into several algorithm techniques.

The outcome result of any classifier can be evaluated using cross-validation techniques such as: 2-fold cross-validation, k-fold cross-validation, and leave-one-out cross-validation. The 2-fold cross validation is the simplest technique, called holdout [41]. Dataset are separated into two sets, one is called training set and other is called testing set. The training set is used in training while other set is used in testing. The k-fold cross-validation is an improvement version of 2-fold [41]. Dataset are separated into k equal sets. The validation is repeated k times. Each time, one set is used for testing and others for training. Leave-one-out cross validation is the most common technique and it is a kind of k-fold but k is taken maximum possible value (logical extreme), which is equal to number of data (N) in the dataset [41]. Dataset are separated into N sets. Each time, one set is used for testing and others for training.

Two different kinds of classifiers KNN (Sect. 2.6.1) and SVM (Sect. 2.6.2) are used with two different techniques as Leave-One-Video-Out (LOVO) and Leave-One-Actor-Out (LOAO). The LOVO is a kind of leave-one-out technique and the LOAO is a kind of *k*-folds cross validation technique. For all experiments, a supervised learning is used. All data are labeled with class memberships except the testing data sample is used without class membership.

2.6.1 K-Nearest Neighbor Classifier

The KNN is the simplest method used for classification, clustering, regression, etc. [7, 8, 9]. The KNN is used in machine learning, pattern recognition, and data mining. It obtains class membership for some testing feature descriptor based on its nearest neighbor from training feature descriptors in feature space. The testing is classified by a majority vote of its *K* nearest neighbors. Due to time required for classifying, the KNN is called lazy learning because the KNN will go over all training samples to find the nearest neighbor for testing sample, therefore, it takes long time, if the training samples are numerous.

In the KNN, there are three parameters are used. First parameter, *K* is set up to number of voting members. Second, distance metric type is set up into: Euclidean (squared difference), cityblock (absolute difference), cosine metrics, etc. Third, the rule for selecting estimated class for testing sample is set up into: nearest neighbor, random, etc. The KNN classifier is calculated the distances (*d*) between testing sample (*x*) and each training sample (*m*) provided from Eq. 2.10, where *d* is a distance metric, *x* is a testing sample, *m* are training samples, $j = [1, 2, \ldots, N]$, *N* is a number of training samples.

$$d(x, m_j) = \arg\ \min_j \{d(x, m_j)\} \tag{2.10}$$

The distance *d* is argument as the minimum distance (nearest neighbor) among distances between *x* and each training sample. The class membership for action with minimum distance is defined as a class membership for testing sample.

In all KNN experiment, two techniques are used. First, the LOVO is leave-one-out cross-validation technique is employed. Thus, all videos in the dataset are used for training except one video is used for testing. Second, the LOAO is 9-fold cross validation techniques, therefore, all videos separated into number of actors, which is nine sets. One actor (set) is used for testing and others for training.

2.6.2 Support Vector Machine Classifier

The SVM is a binary classifier, which separates some feature descriptors by an optimal hyperplane used as a decision function [11, 10, 12, 13]. This hyperplane is represented as a separation, hence called a margin classifier. The SVM can be used to perform a linear or non-linear classification based on using kernels. Once the SVM is trained on features of training samples, the classifier can make decisions about some features testing sample regardless absence of this feature in the testing sample. The classification is performed such as a human is making a decision.

In all SVM experiments, one technique (LOAO) is used. The dataset is separated into nine folds. Each fold represents one actor in the dataset videos. The classification is repeated 9 times. Each time, one fold (actor) is used for testing and others are used to train the classifier. By end of the 9 times, all videos are used in testing and training modes.

2.7 Human Action Recognition in Videos Algorithm

This section provides details about presented algorithm for human action recognition in videos. This algorithm consists of two modes mainly represented in Sects. 2.7.1 and 2.7.2. First, the training mode is a program to train algorithm about human actions using already classified video samples, as depicted in Fig. 2.11a. Second, the testing mode is a program for classifying the unknown action happened in a video sample and identifying its class membership, as depicted in Fig. 2.11b. Usually the training mode is first started, then testing mode will be executed after that.

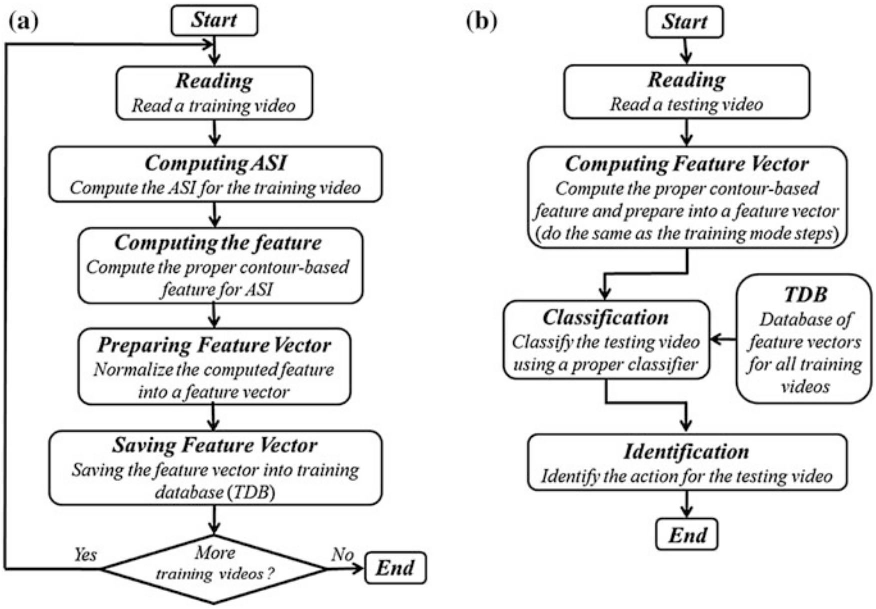

(a)
Start
Reading
Read a training video
Computing ASI
Compute the ASI for the training video
Computing the feature
Compute the proper contour-based feature for ASI
Preparing Feature Vector
Normalize the computed feature into a feature vector
Saving Feature Vector
Saving the feature vector into training database (TDB)
More training videos ?
Yes
No
End
(b)
Start
Reading
Read a testing video
Computing Feature Vector
Compute the proper contour-based feature and prepare into a feature vector (do the same as the training mode steps)
Classification
Classify the testing video using a proper classifier
TDB
Database of feature vectors for all training videos
Identification
Identify the action for the testing video
End

Fig. 2.11 Flow charts of human action recognition algorithms: **a** training mode, **b** testing mode

2.7.1 Training Mode

The training mode is always starts before testing mode in human action recognition. It consists of several processes: reading a training video, computing the ASI from the video, computing (contour-based or silhouette-based) features based on the ASI, preparing feature vector, and saving the feature vector in Training DataBase (TDB). All these steps are repeated for each sample of training video samples in Weizmann dataset. The main structure for training mode of human action recognition in videos is depicted in Fig. 2.11a.

This mode is started by reading a training video process that reads a video sample from the Weizmann dataset. After reading, a process of computing the ASI from this video is performed. Both first and second processes in this algorithm belong to human object tracking stage of the human action recognition system, as depicted in Fig. 2.11. In reading process, all videos in the dataset have (avi) format type. After finishing the reading of all frames, frame by frame in video, the ASI computing process starts in several internal pre-processing steps that have been explained in Sect. 2.3. Briefly, these pre-processing steps involve background subtraction, detection of direction, horizontal alignment, computing the ASI, unifying direction, and bounding box detection processes, respectively.

The rest of processes in training algorithm belong to feature extraction stage of human action recognition, as depicted in Fig. 2.11.

Continuously, after the first and second processes, the third process starts for computing a proper feature. This process employs one of seven different features (contour-based or silhouette-based) that have been explained in Sects. 2.4 and 2.5, respectively. The contour-based involves the CCF, the FDF, the CDF, and the CLF features, while the silhouette-based involves the HOG, the HOOF, and the SSIM features. Regardless of feature type, all features require the ASI that produced in the second process of training mode. For contour-based, the contour of the ASI is first obtained, then a proper features are extracted. While for silhouette-based, features are extracted directly from the ASI. Then, the process of preparing a feature vector is performed. The goal of this process is to normalize the feature vector and make it invariant to scaling and translation with little rotation. Subsequently, for the KNN classifier, the prepared feature vector is saved in the TDB while for the SVM, the classifier has to be trained to find the optimal separation hyperplane for making a decision in testing mode. All processes of training mode are repeated for all training samples in the dataset. At the end of training mode, the TDB have feature vectors for all training videos.

2.7.2 Testing Mode

The testing mode is the second mode of the human action recognition in videos. This mode consists of several processes such as reading a testing video, computing feature vector for the testing video, action classification based on the training feature vectors in the TDB, and identifying action happened in a testing video. Figure 2.11b depicts main structure of testing mode for human action recognition system.

This mode starts by reading a testing video sample, frame by frame, from Weizmann dataset in the same manner as the reading process of training mode. After that, a process of computing feature vector is achieved by a few internal steps. These steps involve computing the ASI for a testing video, computing proper features for this ASI, and preparing the feature vector. This feature vector has to be formatted as the same as all feature vectors in the TDB. Until this end, these steps are common with steps of training mode and have to be exactly in the same manner in every detail, thus comparison of classification algorithm is performed successfully.

Then, the process of classification feature vector for testing video is applied using one of action classification methods that described in Sect. 2.6. The classification process begins using one of two different classifier algorithms: the KNN or the SVM. For the KNN, the feature vector is based on feature vectors in the TDB, which is created in training mode. For the SVM, the created feature vector is projected and classified based on the trained classifier. Subsequently, a final identification is applied to identify the human action that occurred in a testing video based on a result of classifier.

2.8 Experimental Results

The Weizmann dataset is used to test the presented human action recognition system. A background subtraction and several sequence processes are used for human tracking stage. The ASI images are employed to represent each video in the dataset. One feature of contour-based or silhouette-based types is used for feature extraction stage. Also, two classifiers KNN and SVM are used, which are based on two techniques: the LOVO and the LOAO by using 9-folds cross validation. These classifiers are employed for action classification stage. Moreover, the best conducted results for 21 different experiments based on feature and classifier types are presented. Generally, two groups are presented in this section, 12 contour-based experiments for first group and 9 silhouette-based experiments for other. In both groups, two classifiers with different techniques are tested to recognize an action that occurred in videos of Weizmann human action dataset. Figure 2.12 depicts bounding box for contour images. Figure 2.13 depicts bounding box of silhouette images. Both image examples are extracted from different actions in Weizmann dataset.

The descriptions of experiments for various features extraction, including the CCF, the FDF, the CDF, the CLF as well as the HOG, the HOOF, and the SSIM, one can find in Sects. 2.8.1–2.8.7, respectively. Also, the discussion of experimental results is located in Sect. 2.8.8.

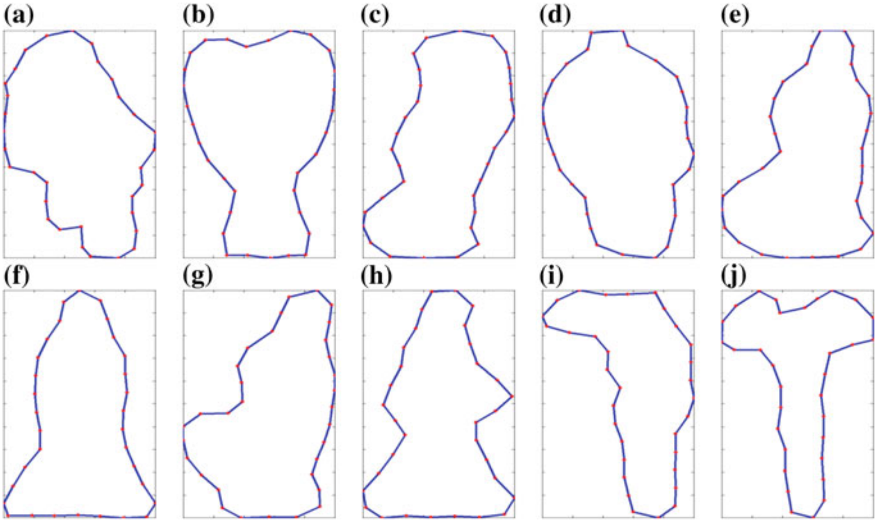


Fig. 2.12 Bounding boxes of contours for the ASI images used to extract contour-based features for 10 different human actions in Weizmann dataset: **a** bending, **b** jumping jack, **c** jumping forward, **d** jumping in place, **e** running, **f** gallop sideway, **g** skip jumping, **h** walking, **i** one hand waving, **j** two hands waving

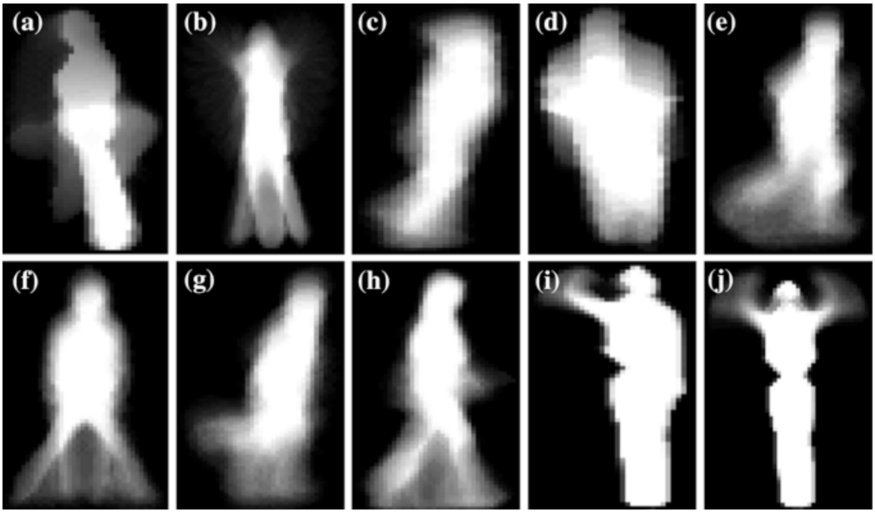


Fig. 2.13 Bounding boxes of the ASI images used to extract silhouette-based features for 10 different human actions in Weizmann dataset: **a** bending, **b** jumping jack, **c** jumping forward, **d** jumping in place, **e** running, **f** gallop sideway, **g** skip jumping, **h** walking, **i** one hand waving, **j** two hands waving

Table 2.1 Cartesian coordinates feature (CCF) experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of boundary points = 16	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	89.247
2.	No. of boundary points = 27	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cosine, rule = nearest	91.397
3.	No. of boundary points = 22	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	**92.473**

2.8.1 Cartesian Coordinate Feature Experiment

The CCF experiments are one of the contour-based feature types. This feature is extracted from contour of the ASI for each video action in the dataset. For the CCF feature, three experiments are examined, based on the classifier used. In these experiments, only one parameter is used for setting a feature, which is a number of Cartesian coordinate points used for boundary of contour. The summary of setup of parameters and results for the CCF experiments are listed in Table 2.1.

In the first experiment, the KNN is used as a classifier to identify action in the testing sample. The best result recorded for this experiment is 89.247 % of correct recognition rate. For the CCF feature, a number of boundary points is set up to 16 points. For the KNN classifier, the LOVO classification technique is used. The number of voting (K) is set up to 1 value. The Euclidean is used to measure distances, and the nearest neighbor rule is used for identifying action in a testing sample.

In the second experiment, the KNN is used as a classifier. The best result recorded for this experiment is 91.397 % of correct recognition rate. For the CCF feature, a number of the used boundary points is set up to 27 points. For the KNN classifier, the LOAO (9-cross validation) technique is used. The number of voting (K) is set up to 1. The Cosine is used to measure distances, and the nearest neighbor is used as a rule to identify the action.

In the third experiment, the SVM classifier is used for classification. The best result recorded for this experiment is 92.473 % of correct recognition rate. For the CCF feature, a number of the used boundary points is set up to 22 points. For classifier, a multi-class SVM type is used. The LOAO (9-cross validation) is used as a classification technique. Also, the linear kernel is used for this classifier.

Table 2.2 Fourier descriptor feature (FDF) experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of Fourier descriptors (FDs) = 18	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	**93.548**
2.	No. of Fourier descriptors (FDs) = 18	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cityblock, rule = nearest	91.397
3.	No. of Fourier descriptors (FDs) = 80	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	91.397

2.8.2 Fourier Descriptor Feature Experiments

The FDF experiments are contour-based feature type. This feature is extracted from the contour of the ASI for each video action in the dataset. For the FDF feature, three experiments are examined, based on the classifier. In these experiments, one parameter is used for setting of feature, which is a number of the FDs used to represent boundary of contour. The summary of setup for parameters and results for the FDF experiments are listed in Table 2.2.

In the first experiment, the KNN is used as a classifier to identify the action. The best result recorded for this experiment is 93.548 % of correct recognition rate, which is the best recognition rate achieved in the contour-based feature types. For the FDF feature, a number of the FDs points is set up to 18 points. For the KNN classifier, the LOVO technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure distances, and the nearest neighbor rule is used.

In the second experiment, the KNN is used as a classifier. The best result recorded for this experiment is 91.397 % of correct recognition rate. For the FDF feature, a number of the used FDs points is also set up to 18 points. For the KNN classifier, the LOAO (9-folds validation) technique is used. The number of voting (K) is set up to 1. The cityblock is used to measure distance, and the nearest neighbor rule is used.

In the third experiment, the SVM classifier is used for classification. The best result recorded for this experiment is 91.397 % of correct recognition rate. The number of the FDs points is set up to 80 points. For classifier, the multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. Also, the linear kernel is used as a base for this classifier.

2.8.3 Centroid-Distance Feature Experiments

The CDF experiments are one of the contour-based types. This feature is extracted based on the FDF. For the CDF feature, three experiments are conducted, based on the

Table 2.3 Centroid-distance feature (CDF) experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of Fourier descriptors (FDs) = 18	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	**92.473**
2.	No. of Fourier descriptors (FDs) = 18	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cityblock, rule = nearest	92.473
3.	No. of Fourier descriptors (FDs) = 96	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	86.021

classifier. In these experiments, one parameter is only used for feature setting. This parameter is a number of the FDs used for boundary representation. The summary of setup for parameters and results for the CDF experiments are listed in Table 2.3.

In the first experiment, the KNN is used as a classifier to identify the action. The best result recorded for this experiment is 92.473 % of correct recognition rate. For the CDF feature, a number of the FDs points is set up to 18 points. For the KNN classifier, the LOVO technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure distances, and the nearest neighbor rule is used for identifying the action.

In the second experiment, the KNN is used as classifier. The best result recorded for this experiment is 92.473 % of correct recognition rate. For the CDF feature, a number of the FDs points is also set up to 18 points. For the KNN classifier, the LOAO (9-folds validation) technique is used. The number of voting (K) is set up to 1. The cityblock is used to measure distances, and the nearest neighbor rule is used.

In the third experiment, the SVM classifier is used for classification. The best result recorded for this experiment is 86.021 % of correct recognition rate. For the CDF feature, a number of the FDs points is set up to 96 points. For classifier, multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. Also, the linear kernel is used for this classifier.

2.8.4 Chord-Length Feature Experiments

The CLF experiments are contour-based feature type. This feature is extracted from the FDF. For the CLF feature, three experiments are experimented, based on classifier type. In these experiments, two parameters are used for setting the FDF. The first parameter is a number of used FDs for boundary representation. The second one is a jump displacement in term of number of points separating two FDs points of chord on boundary of contour. The summary of setup for parameters and results for the CLF experiments are listed in Table 2.4.

Table 2.4 Chord-length feature (CLF) experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of Fourier descriptors (FDs) = 30, jump displacement = 12	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	**89.247**
2.	No. of Fourier descriptors (FDs) = 30, jump displacement = 12	KNN	Leave-one-actor-out, 9-folds cross validation, distance = cityblock, rule = nearest, K = 1	89.247
3.	No. of Fourier descriptors (FDs) = 86, jump displacement = 28	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	89.247

In the first experiment, the KNN is used as a classifier to identify the action. The best result recorded for this experiment is 89.247 % of correct recognition rate. For the CLF feature, a number of the FDs points is set up to 30 points and a jump displacement is set up to 12 separation points. For the KNN classifier, the LOVO technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure the distances and the nearest neighbor rule is used to identify the action.

In the second experiment, the KNN is used as a classifier. The best result recorded for this experiment is 89.247 % of correct recognition rate. For receiving of the CLF feature, a number of the FDs points is also set up to 30 points and a jump displacement is set up to 12 separation points. For the KNN classifier, the LOAO (9-folds validation) technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure distances and the nearest neighbor rule is used.

In the third experiment, the SVM classifier is used for classification. The best result recorded for this experiment is 89.247 % of correct recognition rate. For receiving of the CLF feature, a number of FDs points is set up to 86 points and a jump displacement is set up to 28 separation points. For the SVM classifier, the multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. The linear kernel is used as base for this classifier.

2.8.5 Histogram of Oriented Gradient Feature Experiments

The HOG feature experiments are one of the silhouette-based feature types. This feature is extracted directly from the ASI. For the HOG feature, three experiments are conducted, based on classifier type. In these experiments, two parameters are used for the HOG feature. The first parameter is a number of overlapping windows. The second one is a number of bins for orientation of angles. The summary of setup for parameters and results for the HOG experiments are listed in Table 2.5.

Table 2.5 Histogram of oriented gradient (HOG) feature experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of overlapping windows = (6 × 6), No. of bins = 12	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	**98.924**
2.	No. of overlapping windows = (6 × 6), No. of bins = 12	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cityblock, rule = nearest	97.849
3.	No. of Overlapping windows = (2 × 6), No. of bins = 11	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	97.849

In the first experiment, the KNN classifier is used to identify the action. The best result recorded for this experiment is 98.924 % of correct recognition rate, which is the best recognition rate achieved in both contour-based and silhouette-based feature types. For the HOG feature, a number of overlapping windows is set up to (6 × 6) windows and a number of bins is set up to 12 bins. For the KNN classifier, a number of voting (K) is set up to 1. The LOVO technique is used. The Euclidean is used to measure distances, and the nearest neighbor rule is used to identify the action.

In the second experiment, the KNN is used as a classifier. The best result recorded for this experiment is 97.849 % of correct recognition rate. For the HOG feature, a number of overlapping windows is set up to (6 × 6) windows and a number of bins is set up to 12 bins. For the KNN classifier, the LOAO technique (9-folds validation) is used. The number of voting (K) is set up to 1. The nearest neighbor is used as a rule based on Euclidean distance.

In the third experiment, the SVM classifier is used for classification. The best result recorded for this experiment is 97.849 % of correct recognition rate. For the HOG feature, a number of overlapping windows is set up to (2 × 6) windows and a number of bins is set up to 11 bins. For a classifier, the multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. The linear kernel is used as base for this classifier.

2.8.6 Histogram of Oriented Optical Flow Feature Experiments

The HOOF feature experiments silhouette-based feature type. This feature is also extracted directly from the ASI. For this feature, three experiments are examined, based on classifier type. In these experiments, two parameters are used for the HOOF feature. The first parameter is a number of non-overlapping windows.

Table 2.6 Histogram of oriented optical flow (HOOF) feature experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	No. of non-overlapping windows = (5 × 5), No. of bins = 7	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	96.774
2.	No. of non-overlapping windows = (5 × 5), No. of bins = 7	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cityblock, rule = nearest	**97.849**
3.	No. of non-overlapping windows = (8 × 3), No. of bins = 3	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	96.774

The second one is a number of oriented bins for angles. The summary of setup for parameters and results for the HOOF experiments are listed in Table 2.6.

In the first experiment, the KNN classifier is used to identify the action. The best result that recorded for this experiment is 96.774 % of correct recognition rate. For the HOOF feature, a number of non-overlapping windows is set up to (5 × 5) windows and a number of bins is set up to 7 bins. For the KNN classifier, the LOVO technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure the distances, and the nearest neighbor rule is used.

In the second experiment, the KNN is used as a classifier. The best result that recorded for this experiment is 97.849 % of correct recognition rate. For the HOOF feature, a number of non-overlapping windows is set up to (5 × 5) windows and a number of bins is set up to 7 bins. For the KNN classifier, the LOAO technique (9-folds validation) is used. The number of voting (K) is set up to 1. The cityblock metric is used to measure distances. The nearest neighbor rule is used to identify the action.

In the third experiment, the SVM classifier is used for classification. The best result that recorded for this experiment is 96.774 % of correct recognition rate. For the HOOF feature, a number of non-overlapping windows is set up to (8 × 3) windows and a number of bins is set up to 3 bins. For classifier, the multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. The linear kernel is used as base for this classifier.

2.8.7 Structure Similarity Index Measure Feature Experiments

The SSIM feature experiments are one of the silhouette-based types. This feature is also extracted directly from the ASI. For this feature, three experiments are conducted, based on classifier type. In these experiments, four parameters are used for

Table 2.7 Structure similarity index measure (SSIM) features experiments setting and results

Exp. no.	Feature parameters	Classifier type	Classifier parameters	Correct recognition rate
1.	Image dimension = (50 × 36), L = 4, C1 = 0.03, C2 = 0.01, window = (2 × 2)	KNN	Leave-one-video-out, K = 1, distance = euclidean, rule = nearest	**98.924**
2.	Image dimension = (50 × 36), L = 4, C1 = 0.03, C2 = 0.01, window size = (2 × 2)	KNN	Leave-one-actor-out, 9-folds cross validation, K = 1, distance = cityblock, rule = nearest	94.623
3.	Image dimension = (47 × 48), L = 7, C1 = 0.03, C2 = 0.01, window size = (2 × 2)	SVM	Leave-one-actor-out, 9-folds cross validation, classifier = multi-class, kernel = linear	96.774

the SSIM feature. The first parameter is a dimension size of the ASI, which represents dimensions of an image. The second parameter is a dynamic range of intensity values. The third parameter is two small constants used to overcome problem of division by zero. The fourth parameter is a size of overlapping windows used to obtain one SSIM feature value. The summary of setup for parameters and results for SSIM experiments are listed in Table 2.7.

In the first experiment, the KNN classifier is used to identify the action. The best result that recorded for this experiment is 98.924 % of correct recognition rate. For the SSIM feature, an image dimension is set up to (50 × 36) for all ASIs. A dynamic range is set up to (L = 4) value. Two small constants are set up to (C1 = 0.03 and C2 = 0.01). The overlapping window is set up to (2 × 2). For the KNN classifier, the LOVO technique is used. The number of voting (K) is set up to 1. The Euclidean is used to measure the distances, and the nearest neighbor rule is used.

In the second experiment, the KNN is used as a classifier. The best result that recorded for this experiment is 94.623 % of correct recognition rate. For the SSIM feature, an image dimension is set up to (50 × 36) for all ASIs. The dynamic range is set up to (L = 4) value. Two small constants are set up to (C1 = 0.03 and C2 = 0.01). The number of overlapping window is set up to (2 × 2). For the KNN classifier, the LOAO technique (9-folds validation) is used. The number of voting (K) is set up to 1. The cityblock metric is used to measure distances among samples, and the nearest neighbor rule is used.

In the third experiment, the SVM classifier is used for classification. The best result that recorded for this experiment is 96.774 % of correct recognition rate. For the SSIM feature, an image dimension is set up to (47 × 48) for all ASIs. The dynamic range is set up to value 7. Two small constants used to avoid division by zero problem are set up to (C1 = 0.03 and C2 = 0.01). The number of overlapping window is set up to (2 × 2). For classifier, the multi-class SVM type is used. The LOAO (9-folds validation) is used as a classification technique. Also, the linear kernel is used as base for this classifier.

2.8.8 Experimental Results Discussion

In order to test the presented algorithm, different experiment results are conducted. Summary for all results are listed in Table 2.8.

For feature extraction, seven different types are used. Four are contour-based feature type such as the CCF, the FDF, the CDF, and the CLF. Also, three are silhouette-based feature type such as the HOG, the HOOF, and the SSIM. These features are used for human action recognition in Weizmann dataset [2]. Moreover, for each of both (contour-based and silhouette-based) features, two different types of the classifiers (the KNN and the SVM) are used with different techniques. Totally, 21 experiments are achieved to recognize actions in Weizmann dataset. As listed as in Table 2.8, for contour-based feature type, the best result 93.548 % is achieved for the FDF using the KNN-LOVO, which is the KNN classifier based on

Table 2.8 Results of all contour-based and silhouette-based feature type experiments using seven features and three classifiers

Exp. no.	Feature-based	Feature type	Classifier type	Corrects recognition/all	Wrongs recognition/all	Correct recognition rate %
1.	Contour-based	CCF	KNN-LOVO	83/93	6/93	89.247
2.			KNN-LOAO	85/93	8/93	91.397
3.			SVM	86/93	7/93	92.473
4.		FDF	KNN-LOVO	87/93	6/93	**93.548**
5.			KNN-LOAO	85/93	8/93	91.397
6.			SVM	85/93	8/93	91.397
7.		CDF	KNN-LOVO	86/93	7/93	92.473
8.			KNN-LOAO	86/93	7/93	92.473
9.			SVM	80/93	13/93	86.021
10.		CLF	KNN-LOVO	83/93	10/93	89.247
11.			KNN-LOAO	83/93	10/93	89.247
12.			SVM	83/93	10/93	89.247
13.	Silhouette-based	HOG	KNN-LOVO	92/93	1/93	**98.924**
14.			KNN-LOAO	91/93	2/93	97.849
15.			SVM	91/93	2/93	97.849
16.		HOOF	KNN-LOVO	90/93	3/93	96.774
17.			KNN-LOAO	91/93	2/93	97.849
18.			SVM	90/93	3/93	96.774
19.		SSIM	KNN-LOVO	91/93	2/93	98.924
20.			KNN-LOAO	88/93	5/93	94.623
21.			SVM	90/93	3/93	96.774

leave-one-video-out technique. For silhouette-based feature type, the best result 98.924 % is achieved for the HOG using also the KNN-LOVO classifier. This result is the best result achieved among all 21 experiments. Generally, the results of silhouette-based are similar to each other and contour-based are also. The silhouette-based are better than contour-based feature types.

For contour-based feature type, the FDF is performed better result in term of accuracy than others. This is due that the FDF is capturing the high details and ignoring the low details in the contour. This is one of main characteristics for the FDs. The CCF is using all coordinates for contour and employing interpolation techniques for unifying number of boundary points for each contour. Therefore, accuracy results of the FDF are better than the CCF. Others CDF and CLF are converting the FDF into the distances (lengths) instead of using them directly. Thus, there results are close to each other but the FDF is more accurate. All silhouette-based features achieved very similar results to each other. The HOG and the SSIM achieve slightly more accurate result than the HOOF. This is due than both are using an overlapping windows to extract the feature, while the HOOF is based on non-overlapping windows.

2.9 Conclusion

In this chapter, the goal of human action recognition is achieved based on two different types (contour-based and silhouette-based) of features and by using three different types of classification (the KNN-LOVO, the KNN-LOAO, and the SVM). The results proved that silhouette-based features are better, in term of accuracy rate, than contour-based features, due to three reasons. First, silhouette-based features implicitly contain the contour, which is the boundary of silhouette. Second, silhouette-based features cover interior regions while contour-based features are covering border line. Third, silhouette-based features contain different intensity values that help to build better recognition feature in term of accuracy rate. For example, the silhouette-based features are capturing different intensity values and some possible gap region(s) that located interior of silhouette, while the contour-based ignores all these intensity values and regions of silhouette.

Moreover, the comparison between these two features (contour-based and silhouette-based), in term of computation time, shows that the contour-based is faster than the silhouette-based. This is due to a number of pixels are used in computation in contour is small while in the silhouette-based this number is large and very large comparing into the contour-based. In other words, the contour-based features are performing computation only on the boundary (sub pixels) points while the silhouette-based are performing computation over all pixels of an image (silhouette region and empty region). The contour-based approach is easier and simpler than silhouette-based, in term of complexity, because silhouette-based method employs an additional calculation such as orientation of bins in the HOG and the HOOF and also variables multitude in the SSIM.

The accuracy of classifiers and techniques demonstrates that these classifiers are close to each in term of accuracy in both contour-based and silhouette-based types. Although, the accuracy of the KNN-LOVO is slightly better than other (the KNN-LOAO and the SVM) classifiers, in term of correct recognition rate, since the best results are recorded using the KNN-LOVO in both contour-based and silhouette-based features. In all experiments, results in all three types of classifiers, reflected closeness, in term of accuracy, to each other, as listed in Table 2.8.

Moreover, the comparison between these three classifiers, in term of computation time, shows that the SVM is faster than other two KNN classifiers. The SVM requires more time in training mode and little time for testing mode, while the KNN requires more time in the testing. The time of testing is important for classification. Also, both KNN classifiers are almost similar to each other but slower than the SVM in testing mode. Therefore, the KNN is generally called lazy classifiers. The SVM classifier is more complex than others, in term of complexity, because it is based on kernels and margins. The KNN classifiers are simpler to compute and easier to use compared to the SVM.

References

1. Chaudhry R, Ravichandran A, Hager G, Vidal R (2009) Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: IEEE conferences on computer vision and pattern recognition (CVPR'2009), pp 1932–1939
2. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2007) Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell 29(29):2247–2253. http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html. Accessed 15 June 2014
3. Sadek S, Al-Hamadi A, Michaelis B, Sayed U (2012) Chord length shape features for human activity recognition. ISRN machine vision, article ID 872131. doi:10.5402/2012/872131
4. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput Surv 43 (3):16:1–16:43
5. Monnet A, Mittal A, Paragios N, Ramesh V (2003) Background modeling and subtraction of dynamic scenes. In: IEEE international conferences on computer vision (ICCV'2003), vol 2, pp 1305–1312
6. Piccardi M (2004) Background subtraction techniques: a review. In: Proceeding IEEE international conferences on systems, man and cybernetics, vol 4, pp 3099–3104
7. Deza E, Deza MM (2009) Encyclopedia of distances. Springer, Berlin
8. Duda RO, Hart PE, Stork DG (2000) Pattern classification, 2nd edn. Wiley-Interscience, New York
9. Elkan C (2011) Nearest neighbor classification. doi:10.1007/978-0-387-39940-9_2920
10. Ben-Hur A, Weston J (2010) A user's guide to support vector machines. In: Carugo O, Eisenhaber F (eds) Data mining techniques for the life sciences. Humana Press a part of Springer Science + Business Media, LLC 2010, New York
11. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167

12. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27:1–27:2
13. Gunn SR (1998) Support vector machines for classification and regression. University of Southampton, Technical report MP-TR-98-05, Image speech and intelligent systems group
14. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38(4):1–45
15. Harris C, Stephens M (1988) A combined corner and edge detector. In: 4th Alvey vision conferences, pp 147–151
16. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
17. Mikolajczyk K, Schmid C (2002) An affine invariant interest point detector. In: Proceedings of the 7th European conferences on computer vision (ECCV'2002), vol 1, pp 128–142
18. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–575
19. Shi J, Tomasi C (1994) Good features to track. In: IEEE conferences on computer vision and pattern recognition (CVPR'1994), pp 593–600
20. Comaniciu D, Meer P (1999) Mean shift analysis and applications. In: International conferences on computer vision (ICCV'1999),vol 2, pp 1197–1203
21. Shi J, Malik J (1997) Normalized cuts and image segmentation. In: IEEE conferences on computer vision and pattern recognition (CVPR'1997), pp 731–737
22. Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. Int J of Comput Vis 22 (1):61–79
23. Wren C, Azarbayejani A, Darrell T, Pentland A (1997) Pfinder: real-time tracking of the human body. IEEE Trans Pattern Anal Mach Intell 19(7):780–785
24. Lo BPL, Velastin SA (2001) Automatic congestion detection system for underground platforms. In: International symposium on intelligent multimedia, video and speech processing (ISIMP'2001), pp 158–161
25. Cucchiara R, Grana C, Piccardi M, Prati A (2003) Detecting Moving Objects, Ghosts, and Shadows in Video Streams. IEEE Trans Pattern Anal Mach Intell 25(10):1337–1342
26. Stauffer C, Crimson WEL (1999) Adaptive background mixture models for real-time tracking. In: IEEE conferences on computer vision and pattern recognition (CVPR'1999), vol 2, pp 246–252
27. Stauffer C, Grimson WEL (2000) Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell 22(8):747–757
28. Oliver NM, Rosario B, Pentland AP (2000) A Bayesian computer vision system for modeling human interactions. IEEE Trans Pattern Anal Mach Intell 22(8):831–843
29. Foschi PG, Kolippakkam D, Liu H, Mandvikar A (2002) Feature extraction for image mining. In: 8th international workshop multimedia information systems, pp 103–109
30. Amraji N, Mu L, Milanova M (2011) Shape-based human actions recognition in videos. In: 14th international conferences on human–computer interaction: design and development approaches, vol 1, pp 539–546
31. Zhao H, Liu Z (2009) Shape-based human activity recognition using edit distance. In: 2nd international congress on image and signal processing (CISP'2009), pp 1–4
32. Zivkovic Z, Heijden van der F, Petkovic M, Jonker W (2001) Image segmentation and feature extraction for recognizing strokes in tennis game videos. In: 7th annual conferences of the advanced school for computing and imaging (ASCI'2001), pp 262–267
33. Vezhnevets A, Vezhnevets V (2005) Modest AdaBoost—teaching AdaBoost to generalize better. In: 15th international conferences on computer graphics and applications (GraphiCon'2005)
34. Rabiner LR, Juang BH (1986) An introduction to hidden Markov models. IEEE ASSP Mag 3 (1):4–16
35. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286

36. Kwok KL (1989) A neural network for probabilistic information retrieval. In: 12th annual international ACM SIGIR conferences on research and development in information retrieval (SIGIR'1989), vol 23(SI), pp 21–30
37. Stergiou C, Siganos D (2014) Neural network. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html. Accessed 15 June 2014
38. Domingos P (2012) A few useful things to know about machine learning. Mag Commun ACM 55(10):78–87
39. Ozgur A (2004) Supervised and unsupervised machine learning techniques for text document categorization. MSc thesis, Bogazici University
40. Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Synth Lect Artif Intell Mach Learn 3(1):1–130
41. Schneider J Cross validation. http://www.cs.cmu.edu/~schneide/tut5/node42.html. Accessed 13 Feb 2014
42. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE conference on computer vision and pattern recognition (CVPR'2005), vol 1, pp 886–893
43. Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: European conferences on computer vision (ECCV'2006), pp 428–441
44. Al-Ali S, Milanova M (2014) Human action recognition in videos using structure similarity of aligned motion images. Int j reasoning-based intell syst (IJRIS), 6(1/2):7182
45. Han J, Bhanu B (2006) Individual recognition using gait energy image. IEEE Trans on Pattern Anal Mach Intell 28(2):316–322
46. Huang C, Hsieh C, Lai K, Huang W (2011) Human action recognition using histogram of oriented gradient of motion history image. In: IEEE 1st international conferences on instrumentation, measurement, computer, communication and control (IMCCC'2011), pp 353–356
47. Gonzalez R, Woods R, Eddins S (2009) Digital image processing using Matlab, 2nd edn. Gatesmark Publishing, Knoxville
48. Kauppinen H, Seppanen T, Pietikainen M (1995) An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Trans Pattern Anal Mach Intell 17(2):201–207
49. Léon RD, Sucar L (2000) Human silhouette recognition with Fourier descriptors. In: 15th international conferences on pattern recognition (ICPR'2000), vol 3, pp 709–712
50. Zhang D, Lu G (2004) Review of shape representation and description techniques. Pattern Recogn 37(1):1–19
51. Zhang D, Lu G (2003) A comparative study on shape retrieval using Fourier descriptors with different shape signatures. J Vis Commun Image Represent 14(1):41–60
52. MathWorks Inc.: MATLAB version R2013a (8.1.0.604) win 64-bit software, February (2013)
53. Li Z, Li X, Li C, Cao Z (2010) Improvement on inverse distance weighted interpolation for ore reserve estimation. In: Proceedings of the fuzzy systems and knowledge discovery (FSKD'2010), PP 1703–1706
54. Luo H, He X (2011) An improved inverse distance weighted interpolation method for InSAR tropospheric delay error corrections. In: International conferences on information science and technology (ICIST'2011), pp 480–482
55. Revesz P, Li L (2002) Constraint-based visualization of spatial interpolation data. In: IEEE 6th international conferences on information visualization, pp 563–569
56. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structure similarity. IEEE Trans Image Process 13(4):600–612
57. Pers J, Sulic V, Kristan M, Perse M, Polanec K, Kovacic S (2010) Histograms of optical flow for efficient representation of body motion. Pattern Recogn Lett 31(11):1369–1376

Chapter 3
The Application of Machine Learning Techniques to Real Time Audience Analysis System

Vladimir Khryashchev, Lev Shmaglit and Andrey Shemyakov

Abstract An application for video data analysis based on computer vision methods is presented in this chapter. The proposed system consists of five consecutive stages: face detection, face tracking, gender recognition, age classification, and statistics analysis. The AdaBoost classifier is utilized for face detection. A modification of Lucas and Kanade algorithm is introduced on the stage of face tracking. Novel gender and age classifiers based on adaptive features and support vector machines are proposed. More than 90 % accuracy of viewer's gender recognition is achieved. All stages are united into a single system of audience analysis. The system allows to extract all possible information about depicted people from the input video stream, aggregate and analyze this information in order to measure different statistical parameters. The proposed software solution can find its applications in different areas, from digital signage and video surveillance to the automatic systems of accident prevention and intelligent human-computer interfaces.

Keywords Image recognition · Face detection · Gender classification · Age estimation · Machine learning · Object tracking · Support vector machines

3.1 Introduction

Automatic video data analysis is very challenging problem. In order to find a particular object in a video stream and automatically decide, if it belongs to a particular class, one should utilize a number of different machine learning techniques and algorithms, solving object detection, tracking, and recognition tasks [1–3]. A lot of different algorithms, using such popular techniques as principal

V. Khryashchev (✉) · L. Shmaglit · A. Shemyakov
P.G. Demidov Yaroslavl State University, 14 Sovetskaya st., Yaroslavl 150000
Russian Federation
e-mail: dcslab@uniyar.ac.ru

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_3

component analysis, histogram analysis, artificial neural networks, Bayesian classification, adaptive boosting learning, different statistical methods, and many others, have been proposed in the field of computer vision and object recognition over recent years. Some of these techniques are invariant to the type of analyzed object, others, on the contrary, are utilizing aprioristic knowledge about a particular object type such as its shape, typical color distribution, relative positioning of parts, etc. [4]. In spite of the fact that in the real world there is a huge number of various objects, a considerable interest is being shown in the development of algorithms for analysis of a particular object type—the human faces. The promising practical applications of face recognition algorithms can be automatic number of visitors calculation systems, throughput control on the entrance of office buildings, airports and subway, automatic systems of accident prevention, intelligent human-computer interfaces, etc.

The gender recognition, for example, can be used to collect and estimate demographic indicators [5–8]. Besides, it can be an important pre-processing step, when solving the problem of person identification, as gender recognition allows twice to reduce the number of candidates for analysis (in case of identical number of men and women in a database), and thus twice to accelerate the identification process.

The human age estimation is another problem in the field of computer vision, which is connected with face area analysis [9]. Among its possible applications one should note the electronic customer relationship management (such systems assume the usage of interactive electronic tools for automatic collection of age information of potential consumers in order to provide the individual advertising and services to clients of various age groups), the security control and the surveillance monitoring (for example, an age estimation system can warn or stop underage drinkers from entering bars or wine shops, prevent minors from purchasing tobacco products from vending machines, etc.), the biometrics (when age estimation is used as a part that provides ancillary information of the users' identity information, and thus decreases the whole system identification error rate). Besides, the age estimation can be applied in the field of entertainment, for example, to sort images into several age groups or to build an age-specific human-computer interaction system, etc. [9].

In order to organize a completely automatic system, the classification algorithms are utilized in the combination with a face detection algorithm, which selects candidates for further analysis [10–15]. In this chapter, a system, which extracts all the possible information about depicted people from the input video stream, aggregates and analyses in order to measure different statistical parameters (Fig. 3.1).

The following metrics are calculated:

- The count—the number of viewers, who have paid an attention to a particular product or have watched the advertisement.
- The Opportunity To See (OTS)—the number of potential viewers, who were close to the presented product or advertising media.

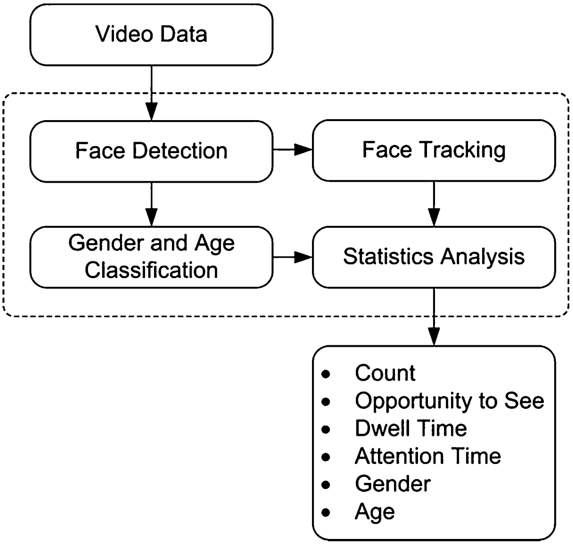


Fig. 3.1 Block diagram of the proposed application for video analysis

- The dwell time—the average time, during which potential viewers have been in the visibility range to the presented product or advertising media.
- The attention time—the average time, when the viewer was watching the object of interest.
- The gender—a viewer gender (man/woman).
- The age—a viewer age group (child/youth/adult/seniors).

The quality of face detection step is critical to the final result of the whole system, as inaccuracies at face position determination can lead to wrong decisions at the stage of recognition. To solve the task of face detection the AdaBoost classifier, described in paper [16], is utilized. The detected fragments are preprocessed to align their luminance characteristics and transform them to uniform scale. On the next stage the detected and preprocessed image fragments are passed to the input of gender recognition classifier, which makes a decision on their belonging to one of two classes ("Male", "Female"). The same fragments are also analyzed by the age estimation algorithm, which divides them into several age groups. The proposed gender and age classifiers are based on a non-linear Support Vector Machine (SVM) classifier with a Radial Basis Function (RBF) kernel. To extract information from image fragment and to move it in a lower dimension feature space, the adaptive feature generation algorithm is proposed. The adaptive features are trained by means of optimization procedure according to a Linear Discriminant Analysis (LDA) principle. The resulted classifier consists of the following steps: the color space transform, the image scaling, the adaptive feature set calculation, and the SVM classification with preliminary kernel transformation.

To estimate the period of a person's stay in the range of camera's visibility, the face tracking algorithm is used [17–20]. Generally speaking, the task of tracking is to match same objects on different frames of video sequence. The object tracking itself is a difficult problem, as it is influenced simultaneously by the following factors:

- The variation of image parameters, scene illumination, and camera noise.
- The presence of objects with varying form (for example, a running person).
- The temporary occlusion of analyzed objects due to overlapping by other objects.
- The existence of several moving objects at the same time with similar features and crossed trajectories.
- The distortions due to wrong segmentation of objects at the previous processing stages.

The main approach to age estimation is a two-level scheme, where on the first step special features [21] are extracted from the analyzed image fragment (the best results are reached applying the combination of various feature descriptors such as an Active Appearance Model (AAM), a Histogram of Oriented Gradients (HOG), a Local Binary Patterns (LBP), a Discriminative Scale-Invariant Feature Transform (DSIFT), etc.), and on the second step a classifier is used to find areas in the resulted feature space, corresponding to certain ages (directly or by means of a set of binary classifiers and a voting scheme). The best results of classification can be reached by utilizing a combination of various approaches such as the SVM, artificial neural networks, random forests, etc [19]. Difficult hierarchical schemes, applied to classifier design, also allow to achieve an advantage in some cases [22].

The study of age estimation performance under variations across race and gender [23] discovered that a crossing race and a gender can result in significant error increase. The age facial features of men and women and also of representatives of various races significantly differ from each other. Thus, the optimum strategy of training is to form a number of separate training sets and construct an independent classifier for each analyzed category. The age estimation is then conducted with preliminary division of all input images into defined categories in order to choose a suitable classifier for each image. The problem of such scheme with preliminary division into categories lies not only in the increase of computational complexity of the total classifier but, mainly, in the significant increase of the required training database capacity.

The framework, described above, is utilized for age estimation of faces varying by their relative position to camera (frontal, panning in horizontal, and vertical direction) [24]. In work [25], a different strategy is suggested to improve the accuracy of age estimation under the facial expression changes. It lies in the search of correlation between faces with different facial expression and in the conversion of initial feature space into a space, where features are similar (become invariant) for neutral, smiling and faces with a sad expression. The automatic facial alignment based on eye detection is suggested in [26] to reduce the influence of face position variation.

The rest of the chapter briefly describes main algorithmic techniques utilized on different stages of the proposed system such as face detection (Sect. 3.2), face tracking (Sect. 3.3), gender recognition (Sect. 3.4), and age estimation (Sect. 3.5). The level of gender and age classification accuracy is estimated in real-life situations.

3.2 Face Detection

To solve the problem of face detection, an algorithm, suggested by Viola and Jones [20], was chosen. It utilizes a learning procedure based on adaptive boosting [27–30]. This procedure consists of three parts:

Step 1. Integral image representation.

The integral image at location (x, y) contains the sum of the pixels above and to the left of x, y, inclusive provided by Eq. 3.1, where $ii(x, y)$ is the integral image, $i(x', y')$ is the original image.

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y') \tag{3.1}$$

The integral image representation (Eq. 3.1) allows to speed up the calculation of a rectangular feature set as any rectangular sum can be computed in four array references.

Step 2. Learning classification functions using the AdaBoost algorithm.

For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples is misclassified. Thus, a weak classifier ($h(x, f, p, \theta)$) consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality $h(\cdot)$ computed by Eq. 3.2, where x is a 24 × 24 pixel sub-window of an image.

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Step 3. Combining classifiers in a cascade structure.

Cascade structure allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions. The cascade structure of a resulted classifier is schematically presented in Fig. 3.2. It consists of N layers, each of which represents a classifier generated by the AdaBoost learning procedure. The considered algorithm is one of the most widely used to solve the problem of face detection in digital images. It is a part of computer vision library OpenCV [30].

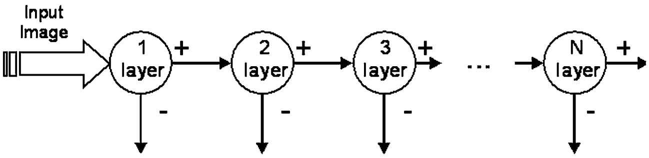


Fig. 3.2 Schematic depiction of the detection cascade

3.3 Face Tracking

Nowadays there exist several approaches to the realization of object tracking in video sequences. For real time applications, methods based on the estimation of optical flow are most widely used. Generally speaking, an optical flow can be defined as two dimensional projections of objects motion on an image plane, representing object pixel's trajectories. An optical flow can be calculated as on the basis of tracking of all image pixels (full optical flow) and on the basis of tracking of particular feature pixels (sparse optical flow) [4]. The disadvantage of full optical flow is its low resistance to image distortions caused by the presence of noise. Thus, the calculation of sparse optical flow is used more often in practice.

An algorithm, proposed by Lucas and Kanade [31], was chosen as the basic approach to solve the problem of optical flow calculation. With the help of this algorithm, the coordinates of feature pixels on the current image frame are calculated out of their coordinates on the previous frame. Then the estimation of new position and size of a tracked object is performed on the basis of the original algorithm represented in Fig. 3.3.

Step 1. Object offset calculation.

First, the differences in coordinates between features on the previous frame and current features are calculated. The resulted values are processed by median filter in order to smooth the spikes, obtained due to inaccuracies in tracking of some certain pixels. Such inaccuracies may be, for example, caused by the presence of noise. After that filtered offsets of feature pixels are averaged. The obtained average value is used as a total offset of the whole object.

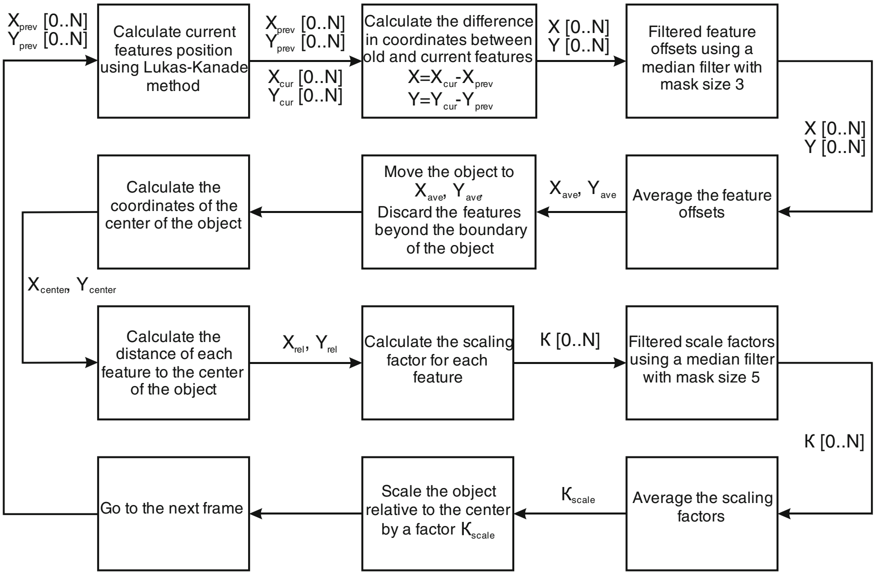

X_{prev} [0..N]
Y_{prev} [0..N]
Calculate current features position using Lukas-Kanade method
X_{prev} [0..N]
Y_{prev} [0..N]
X_{cur} [0..N]
Y_{cur} [0..N]
Calculate the difference in coordinates between old and current features
$X=X_{cur}-X_{prev}$
$Y=Y_{cur}-Y_{prev}$
X [0..N]
Y [0..N]
Filtered feature offsets using a median filter with mask size 3
X [0..N]
Y [0..N]
Average the feature offsets
X_{ave}, Y_{ave}
Move the object to X_{ave}, Y_{ave}, Discard the features beyond the boundary of the object
Calculate the coordinates of the center of the object
X_{center}, Y_{center}
Calculate the distance of each feature to the center of the object
X_{rel}, Y_{rel}
Calculate the scaling factor for each feature
K [0..N]
Filtered scale factors using a median filter with mask size 5
K [0..N]
Average the scaling factors
K_{scale}
Scale the object relative to the center by a factor K_{scale}
Go to the next frame

Fig. 3.3 The scheme of modified tracking algorithm

Step 2. Scaling coefficient calculation.

Due to the fact that the object is scaled relative to its center, for scaling coefficient calculation the coordinates of all feature pixels are recalculated relative to the center of tracked object. Then the distance from each feature to the center of the object is calculated. The resulted values are processed by median filter and then averaged. The obtained average value is rounded to the second digit and used as a scaling coefficient of tracked object. After the estimation of position and size of a tracked object on the current frame, feature pixels, which lie outside the defined border, are rejected.

The face window scaling and offset calculation is not the only problem, which Lucas-Kanade method does not solve. The other two ones are a face overlapping and a face crossing. In this research, three modifications were proposed to solve the problems described above. The first one (Lucas-Kanade-1) defines a tracking object offset and a scaling factor as the averages of key pixels corresponding characteristics. The second one (Lucas-Kanade-2) introduces a median filtration to improve the overall performance. The third one (Lucas-Kanade-3) detects an overlapping and crossing by dividing the window into square regions and labeling each key pixel to the corresponding region. If a pixel moves out of its region, as it is shown in Fig. 3.4, then such pixel is removed from the further consideration being suspected as an overlapped one. In order to compare the proposed modifications, a test sequence containing very difficult movement was chosen as it is presented in Fig. 3.5.

The third modified algorithm comprises all necessary tracking difficulties including overlapping, fast movement, and camera trembling. In Fig. 3.6, the performance of the proposed Lukas-Kanade modifications in the chosen test video sequence is presented. The tracking rate is defined as a relation of a number of frames, on which tracking window follows the object, to the total number of frames in a video segment.

The results show that the proposed second and third modifications allow to achieve better tracking rate compared to classic Lucas-Kanade-1. The algorithm Lucas-Kanade-2 misses the object only once due to overlapping, while the algorithm Lucas-Kanade-3 does not make mistakes in such conditions but misses the object at the very end of test sequence due to camera trembling. Thus, it can be concluded that the proposed face tracking technique requires further improvements.

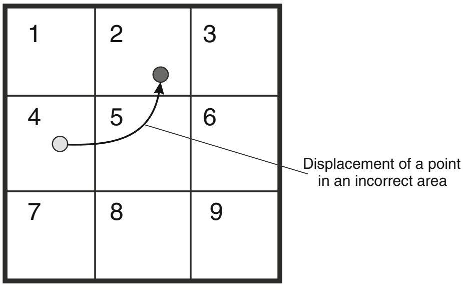


Fig. 3.4 Overlapping and crossing pixel detection

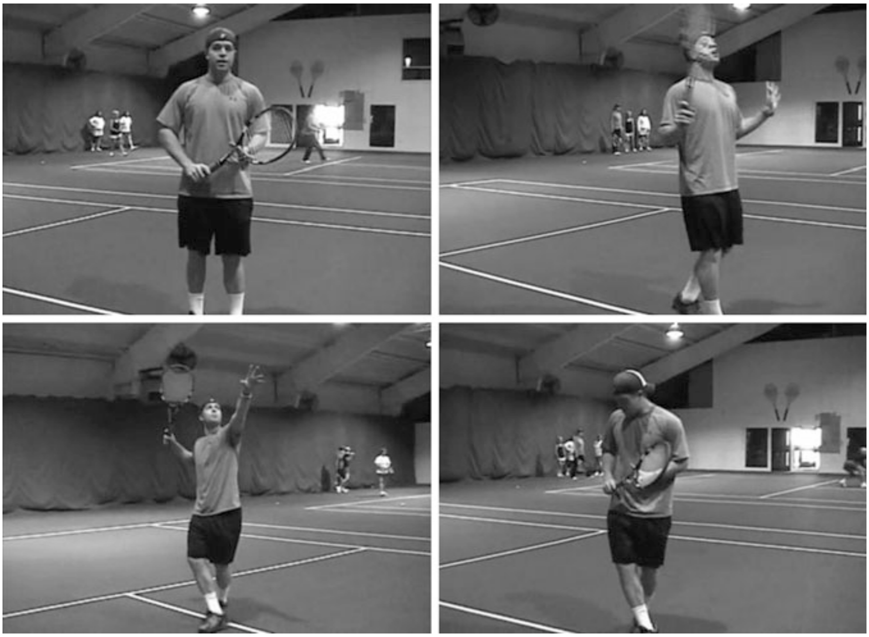


Fig. 3.5 Test video sequence

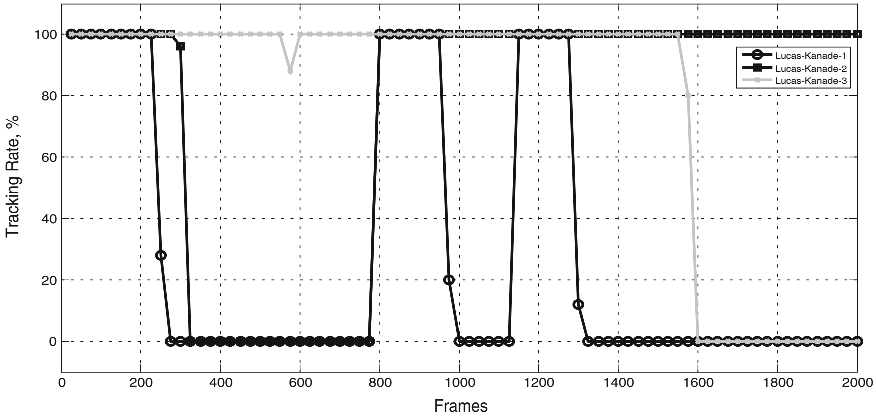


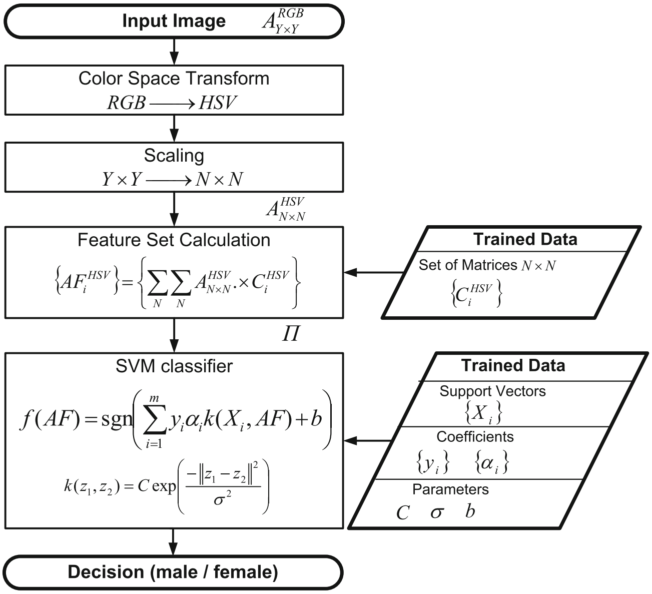
Fig. 3.6 Tracking algorithms testing results

3.4 Gender Recognition

The new gender recognition algorithm proposed in this chapter is based on a nonlinear SVM classifier with RBF kernel. To extract information from image fragment and to move to a lower dimension feature space, an adaptive feature generation algorithm is proposed, which is trained by means of optimization procedure according to the LDA principle. In order to construct a fully automatic face analysis system, the gender recognition is used in connection with the AdaBoost face detection classifier, described in Sect. 3.2. The detected fragments are pre-processed to align their luminance characteristics and transform them to a uniform scale.

The classifier is based on Adaptive Features and SVM (AF-SVM). Its operation includes several stages, as it is shown in Fig. 3.7. The AF-SVM algorithm consists of the following steps: the color space transform, the image scaling, the adaptive feature set calculation, and the SVM classification with a preliminary kernel transformation.

The input image $A^{RGB}_{Y \times Y}$ is converted from RGB to HSV color space (allowing to separate brightness and color components of an image) and is scaled to fixed image resolution $N \times N$. After that a set of features $\{AF_i^{HSV}\}$ is calculated, where each feature (Eq. 3.3) represents the sum of all rows and columns of element-by-element matrix product of an input image and a coefficient matrix C_i^{HSV} with resolution $N \times N$, which is generated by the training procedure.



Input Image $A^{RGB}_{Y\times Y}$
Color Space Transform
$RGB \longrightarrow HSV$
Scaling
$Y\times Y \longrightarrow N\times N$
$A^{HSV}_{N\times N}$
Feature Set Calculation
$\{AF_i^{HSV}\} = \left\{\sum_N\sum_N A^{HSV}_{N\times N} . \times C_i^{HSV}\right\}$
Trained Data
Set of Matrices $N\times N$
$\{C_i^{HSV}\}$
Π
SVM classifier
$f(AF) = \mathrm{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i k(X_i, AF) + b\right)$
$k(z_1, z_2) = C\exp\left(\frac{-\|z_1 - z_2\|^2}{\sigma^2}\right)$
Trained Data
Support Vectors
$\{X_i\}$
Coefficients
$\{y_i\}$ $\{\alpha_i\}$
Parameters
C σ b
Decision (male / female)

Fig. 3.7 Scheme of the proposed gender classification algorithm

$$AF_i^{HSV} = \sum_N \sum_N A_{N\times N}^{HSV} \times C_i^{HSV} \tag{3.3}$$

The obtained feature vector is transformed using a Gaussian RBF kernel by Eq. 3.4.

$$k(z_1, z_2) = C \exp\left(\frac{-\|z_1 - z_2\|^2}{\sigma^2}\right) \tag{3.4}$$

Kernel function parameters C and σ are defined during training. The resulted feature vector serves as an input to a linear SVM classifier, which decision rule is specified by Eq. 3.5.

$$f(AF) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(X_i, AF) + b\right) \tag{3.5}$$

The set of support vectors $\{X_i\}$, the sets of coefficients $\{y_i\}$, $\{\alpha_i\}$ and the bias b are obtained at the stage of classifier training.

Both gender recognition algorithm training and testing require big size color image database. The most commonly used image database for the tasks of human faces recognition is the FERET database [32], but it contains insufficient number of faces of different individuals, that is why the authors collected own image database, gathered from different sources (Table 3.1, Fig. 3.8).

Faces in the images from the proposed database were detected automatically by the AdaBoost face detection algorithm. After that, false detections were manually removed, and the resulted dataset consisting 10,500 image fragments (5,250 for each class) was obtained. This dataset was divided into three independent image sets: the training, the validation, and the testing. The training set was utilized for feature generation and a SVM classifier construction. The validation set was

Table 3.1 The proposed training and testing image database parameters

Parameter	Value
The total number of images	8,654
The number of male faces	5,250
The number of female faces	5,250
Minimum image resolution	640 × 480
Color space format	RGB
Face position	Frontal
People's age	From 18 to 65 years old
Race	Caucasian
Lighting conditions, background, and facial expression	No restrictions

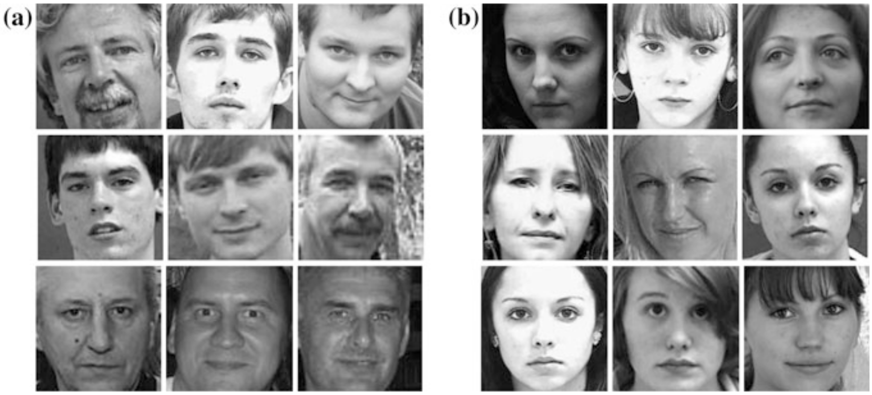


Fig. 3.8 Detected fragments from the proposed image database: **a** male faces, **b** female faces

required in order to avoid the effect of overtraining during the selection of optimal parameters for the kernel function. A performance evaluation of the trained classifier was carried out with the use of the testing set.

The training procedure of the proposed AF-SVM classifier can be divided into two independent parts: the feature generation and the SVM construction and optimization. Let us consider the feature generation procedure. It consists of the following basic steps:

- The RGB → HSV color space transform of the training images (all further operations are carried out for each color component independently).
- The scaling training images to fixed image with resolution $N \times N$.
- The coefficient matrix C_i^{HSV} random generation.
- The feature value AF_i^{HSV} calculation for each training fragment.
- The utility function F calculation (Eq. 3.6) as a square of a difference between feature averages calculated for "male" and "female" training image datasets and divided by the sum of feature variances [9].

$$F = \frac{\left(\left\langle \{AF_i^{HSV}\}_M \right\rangle - \left\langle \{AF_i^{HSV}\}_F \right\rangle\right)^2}{\sigma\{AF_i^{HSV}\}_M + \sigma\{AF_i^{HSV}\}_F} \tag{3.6}$$

- The iteratively in a cycle (until the number of iterations exceeds some preliminary fixed maximum value): the random generation of coefficient matrix $\tilde{C}_i^{HSV}$ inside the fixed neighborhood of matrix C_i^{HSV}, the feature value $A\tilde{F}_i^{HSV}$ calculation for each training fragment, the calculation of the utility function $\tilde{F}$, the transition to a new point ($F \to \tilde{F}, C \to \tilde{C}$), if $\tilde{F} > F$.
- The saving the matrix C_i^{HSV} after exceeding the maximum number of iterations.
- The return to beginning in order to start the generation of the next $(i + 1)$ feature.

An optimization procedure, described above, allows to extract from an image only the information, which is necessary for class separation. Besides, features with higher utility function value have higher separation ability. The feature generation procedure has the following adjusted parameters: the training fragments resolution (N), the number of training images for each class (M), and the maximum number of iterations (T). The following values, as a compromise between reached separation ability and the training speed, were empirically obtained (Eq. 3.7).

$$N = 65 \quad M = 400 \quad T = 10^5 \tag{3.7}$$

The 1,000 features have been generated for each color component. At the second stage of a training, these features have been extracted from training images and then they were used to learn the SVM classifier. The SVM construction and optimization procedure included the following steps:

- The calculation of feature set generated on the first stage of training for each training fragment.
- The normalization of feature values.
- The learning of the SVM classifier with different parameters of the kernel function.
- The Recognition Rate (RR) calculation using validation image dataset.
- The calculation of optimal kernel function parameters maximizing the RR.
- The learning of final SVM classifier with the found optimal kernel function parameters.

The goal of the SVM optimization procedure is to find a solution with the best generalization ability, and, thus, with the minimum classification error. The adjusted parameters are: the number of features in a feature vector ($N2$), the number of training images for each class ($M2$), and the kernel function parameters σ and C.

A grid search was applied to determine optimal kernel parameters: the SVM classifier was constructed varying $C = 10^{k_1}$ and $\sigma = 10^{k_2}$, where k_1 and k_2 are all combinations of integers from the range [–15 … 15]. During a search, the recognition rate was measured using validation image dataset. The results of this procedure are presented in Fig. 3.9. The maximum RR (about 80 %) was obtained for $C = 10^6$ and $\sigma = 10^8$.

Besides, the dependence of classifier performance from the number of features extracted from each color component N_2 and the number of training images for each class M_2 was investigated (Fig. 3.10). The analysis shows that each feature has essential separation ability, and at N_2 = 30 the RR value reaches 79.5 %. At the same time the growth of the RR is observed both with the growth of N_2 and M_2 due to the accumulation of information about considered classes inside the classifier. Thus, to obtain a compromise between quality and speed the following parameters, values N_2 = 30 and M_2 = 400 were chosen.

Let us turn to the results of the proposed AF-SVM algorithm comparison with state-of-the-art classifiers: the SVM [33] and a Kernel Direct Discriminant Analysis (KDDA) [34]. The classifier AF-SVM was trained according to a technique, given

Fig. 3.9 The dependence of the RR from kernel function parameters $C = 10^{k_1}$ and $\sigma = 10^{k_2}$

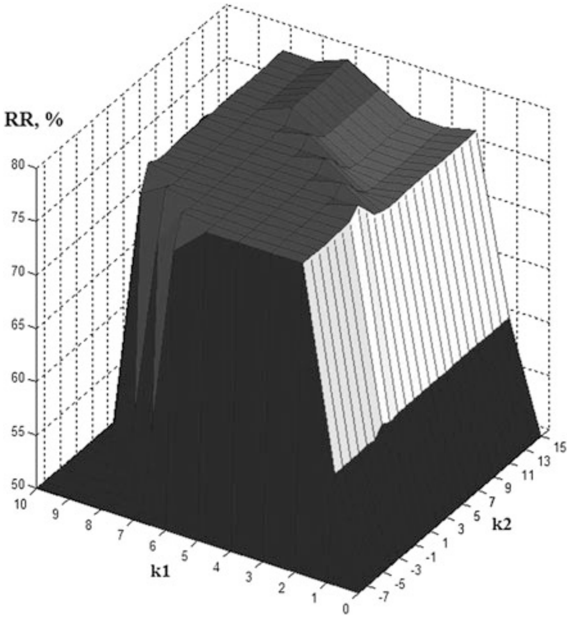
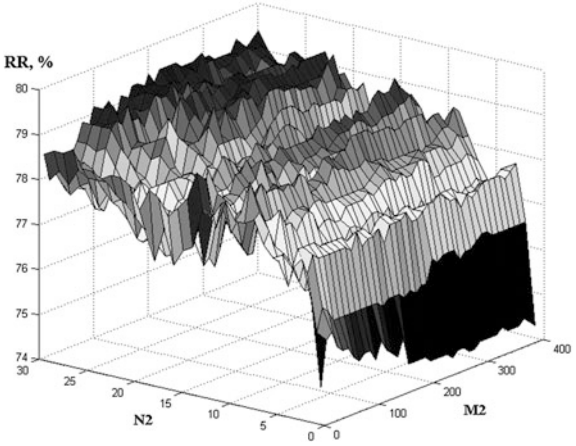


Fig. 3.10 The dependence of the RR from training procedure parameters



above. The SVM and the KDDA classifiers have far less adjustable parameters as they are working directly with image pixel values instead of feature vectors. To construct these classifiers the same training base, as for the AF-SVM classifier, was used. The following conditions also were identical for all three considered classifiers: the number of training images for each class, the training fragments resolution, and the image preprocessing procedure. The optimization of the SVM and the KDDA kernel function parameters was held using the same technique and the same validation image dataset as used in case of the AF-SVM classifier. Thus, equal conditions for independent comparison of considered classification algorithms, using testing image dataset, were provided.

For representation of classification results, the Receiver Operator Characteristic curve (ROC-curve) was utilized. As there are two classes, one of them is considered to be a positive decision and the other—a negative. The ROC-curve is created by plotting the fraction of true positives out of the positives (TPR = *true positive rate*) versus the fraction of false positives out of the negatives (FPR = *false positive rate*) by using various discrimination threshold settings. The advantage of ROC-curve representation lies in its invariance to the relation between the first and the second error type's costs.

The results of the AF-SVM, the SVM, and the KDDA testing are presented in Fig. 3.11 and Table 3.2. The computations were held on a personal computer with the following configuration: operating system—Microsoft Windows 7; CPU type—Intel Core i7 (2 GHz) 4 cores; memory size—6 Gb.

The analysis of testing results shows that the AF-SVM is the most effective algorithm considering both recognition rate and operational complexity. The AF-SVM has the highest RR among all tested classifiers—79.6 % and is faster than the SVM and the KDDA approximately by 50 %. Such advantage is explained by the

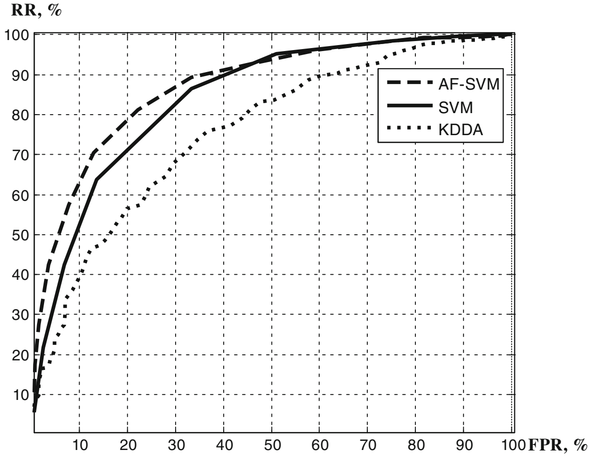


Fig. 3.11 ROC-curves of tested gender recognition algorithms

Table 3.2 Comparative analysis of tested algorithms performance

Algorithm parameter	SVM		KDDA		AF-SVM	
Recognition rate	True	False	True	False	True	False
Classified as "male" (%)	80.0	20.0	75.8	24.2	80.0	20.0
Classified as "female" (%)	75.5	24.5	65.5	34.5	79.3	20.7
Total classification rate (%)	77.7	22.3	69.7	30.3	79.6	20.4
Operation speed (faces/s)	44.0		45.0		65.0	

fact that the AF-SVM algorithm utilizes a small number of adaptive features, each of which carries a lot of information and is capable to separate classes, while the SVM and the KDDA classifiers work directly with a huge matrix of image pixel values.

Let us consider the possibility of classifier performance improvement by the increase of the total number of training images per class from 400 to 5,000. Experiments show that the SVM and the KDDA recognition rates cannot be significantly improved in that case. Besides, their computational complexity increases dramatically with the growth of the training dataset. This is explained by the fact that while the number of pixels, which the SVM and the KDDA classifiers utilize to find an optimal solution in a high dimensional space, increases it becomes harder and even impossible to find an acceptable solution for the reasonable calculation time.

In the case of the AF-SVM classifier the problem of the decrease of the SVM classifier efficiency with the growth of training database can be solved by use of a small number of adaptive features, holding information about a lot of training images at once. For this purpose, there was suggested a procedure, when each feature should be trained using a random subset (containing 400 training images per class) from the whole training database (containing 5,000 images per class). Thus, each generated feature will hold the maximum possible amount of information required to divide the classes, and a set of features will include the information from each of 10,000 training images.

On the stage of feature generation, 300 features were trained according to the technique described above. After that, the SVM classifier utilizing these features was trained similarly as before. Besides, the number of training images for the SVM construction equal to 400 is preserved, and, thus, the operation speed of the final classifier remained the same as in previous experiments is equal 65 faces processed per s. The results of the AF-SVM algorithm trained using expand dataset (M = 5,000) and the initial AF-SVM classifier (M = 400) comparison are presented in Table 3.3 and Fig. 3.12.

The results show that the AF-SVM algorithm together with the proposed training setup allow to improve significantly the classifier performance in case of increasing the training database size to 5,000 images per class. The RR of nearly 91 % is achieved. It should be also noted that the adaptive nature of feature generation procedure allows to use the proposed AF-SVM classifier for the recognition of any other object in an image (in addition to faces).

Table 3.3 Recognition rate of the AF-SVM algorithm trained on datasets of different size

Algorithm parameter	AF−SVM (M = 5,000)		AF−SVM (M = 400)	
Recognition rate	True	False	True	False
Classified as “male” (%)	90.6	9.4	80.0	20.0
Classified as “female” (%)	91.0	9.0	79.3	20.7
Total classification rate (%)	90.8	9.2	79.6	20.4

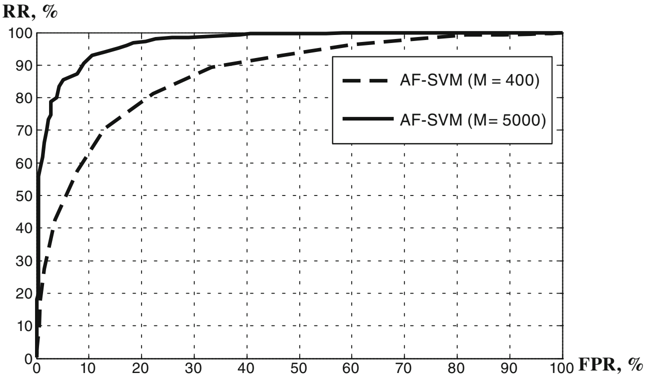


Fig. 3.12 ROC-curves for the AF-SVM algorithm trained on datasets of different size

3.5 Age Estimation

The proposed age estimation algorithm realizes a hierarchical approach (Fig. 3.13). First of all, the input fragments are divided into three age groups: less than 18 years old, from 18 to 45 years old, and more than 45 years old. After that the results of classification on the first stage are further divided into seven new groups, each of which is limited to one decade. Thus, the problem of multi-class classification is reduced to a set of Binary "one-against-all" Classifiers (BC). These classifiers calculate ranks for each of the analyzed classes. Then the total decision is obtained by the analysis of the previously received histogram of ranks.

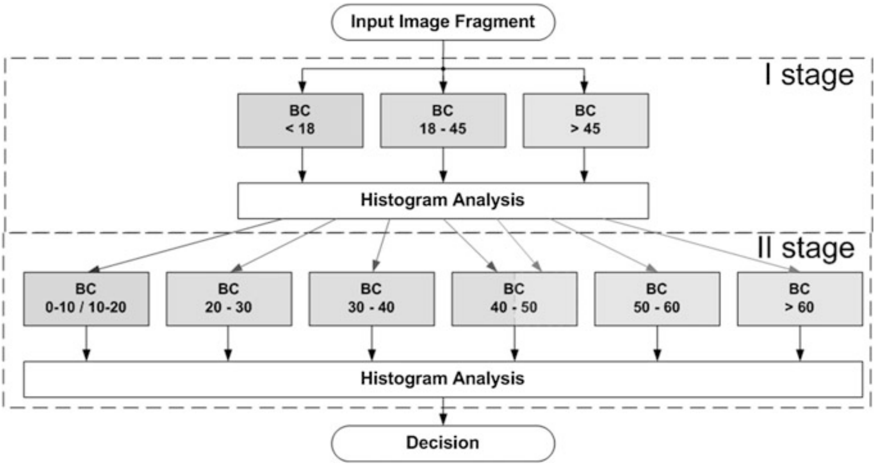


Fig. 3.13 Block diagram of the proposed age estimation algorithm

Table 3.4 First stage, the image database parameters

Class label database capacity	<18	18–45	>45	Total
Training images per class	2,000	2,000	3,000	7,000
Testing images per class	226	400	5,31	1,157
Total number of images	2,226	2,400	3,531	8,157

Table 3.5 First stage, the AF-SVM summary

Training parameters		Value
Number of binary classifiers		3
Number of color components used		3
Number of adaptive features generated	Per color component	48
	Per binary classifier	144
	Total	432

Two level scheme of the BCs construction is applied with the transition to adaptive feature space, similar to described earlier, and the SVM classification with the RBF kernel. The input fragments are preprocessed to align their luminance characteristics and transform them to a uniform scale. The preprocessing includes the color space transformation and scaling, both similar to that of gender recognition algorithm. Features, calculated for each color component, are combined to form a uniform feature vector.

The training and the testing require a huge enough color image database. The state-of-the-art image databases MORPH [35] and FG-NET [36] as well as the own image database gathered from different sources, which consisted of 10,500 face images, were used. Faces on the images were detected automatically by the AdaBoost face detection algorithm.

Total number of seven thousand images were used for age classification algorithm training and testing at the first stage (Table 3.4). Three BCs were constructed utilizing 144 adaptive features each (Table 3.5).

The ROC-curves of binary classifies are presented in Fig. 3.14. It is clear, that the main problem is to distinguish an age group from 18 to 45 years old.

The classification results are as follows (Table 3.6): 82 % accuracy for young age group, 58 % accuracy for middle age group, and 92 % accuracy for senior age group. The age classification rate in a three age group division problem achieved 77.3 %.

The BCs of the second stage were constructed similar to those of the first stage described above. A visual example of age estimation by the proposed algorithm on its first and second stages is presented in Fig. 3.15a, b.

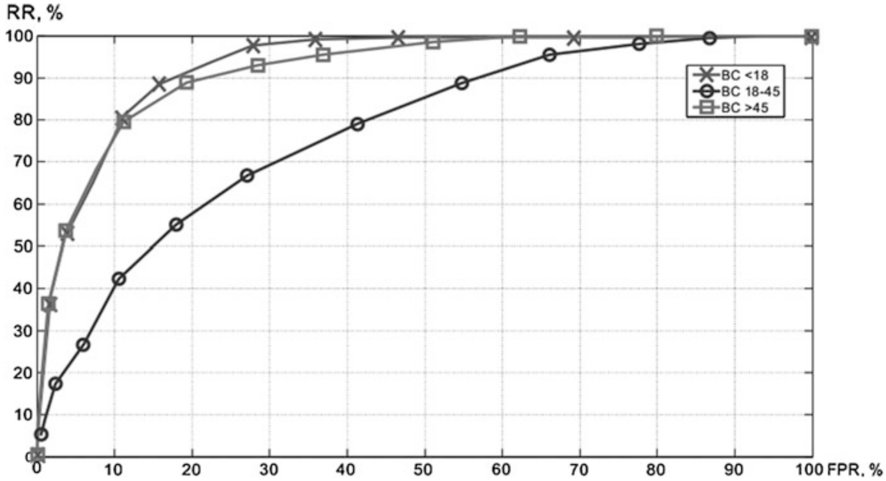


Fig. 3.14 Binary classifiers ROC-curves

Table 3.6 First stage, the age classification results

Decision ground truth	<18 (%)	18–45 (%)	>45 (%)
<18	82	14	4
18–45	22	58	20
>45	3	5	92

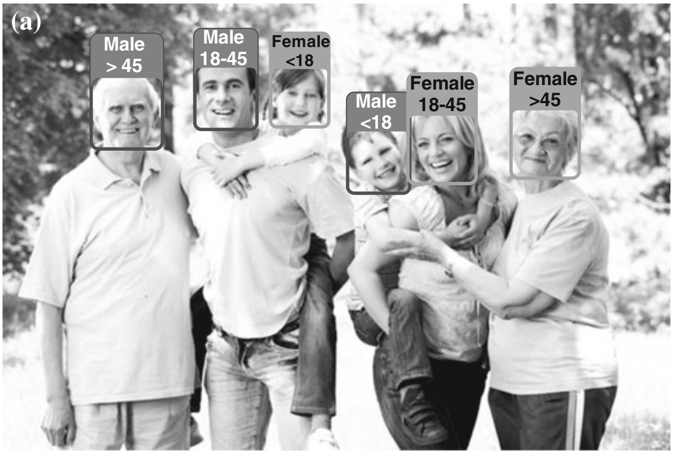


Fig. 3.15 Visualization of the proposed system age estimation performance: **a** at the first stage, **b** at the second stage

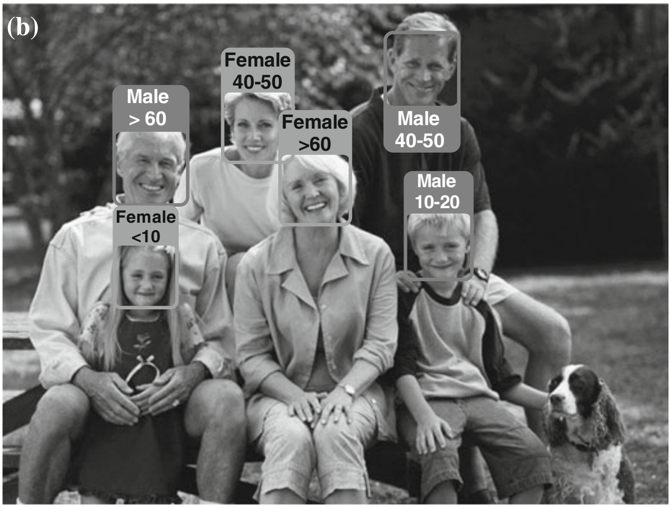


Fig. 3.15 (continued)

3.6 Conclusion

The system, described in this chapter, provides the collection and processing of information about the audience in a real time. It is fully automatic and does not require people to conduct it. No personal information is saved during the process of operation. A modern efficient classification algorithm allows to recognize a viewer's gender with more than 90 % accuracy. Our classification results are as follows: 82 % accuracy for young age group, 58 % accuracy for middle age group, and 92 % accuracy for senior age group. The age classification rate in a three age group division problem achieved 77.3 %. The noted features allow to apply the proposed system in various spheres of life: the places of mass stay of people (stadiums, movie theaters, and shopping centers), the transport knots (airports, railway, and auto stations), the border passport and visa control check-points, etc.

References

1. Alpaydin E (2010) Introduction to machine learning, 2nd edn. The MIT Press, Cambridge
2. Sammut C, Webb GI (eds) (2011) Encyclopedia of machine learning. Springer Science + Business Media, LLC, New York
3. Li SZ, Jain AK (eds) (2005) Handbook of face recognition. Springer, New York
4. Szeliski R (2010) Computer vision: algorithms and applications. Springer, London
5. Makinen E, Raisamo R (2008) An experimental comparison of gender classification methods. Pattern Recogn Lett 29(10):1544–1556
6. Tamura S, Kawai H, Mitsumoto H (1996) Male/female identification from 8 to 6 very low resolution face images by neural network. Pattern Recogn Lett 29(2):331–335

7. Khryashchev V, Priorov A, Shmaglit AL, Golubev M (2012) Gender recognition via face area analysis. In: World congress on engineering and computer science, pp 645–649
8. Khryashchev V, Ganin A, Golubev M, Shmaglit L (2013) Audience analysis system on the basis of face detection, tracking and classification techniques. In: International multi-conference of engineers and computer scientists (IMECS 2013), vol 1, pp 446–450
9. Fu Y, Huang TS (2010) Age synthesis and estimation via faces: a survey. IEEE Trans Pattern Anal Mach Intell 32(11):1955–1976
10. Sung KK, Poggio T (1998) Example-based learning for view-based human face detection. IEEE Trans Pattern Anal Mach Intell 20(1):39–51
11. Maydt J, Lienhart R (2002) Face detection with support vector machines and a very large set of linear features. In: IEEE international conference on multimedia and expo (ICME'2002), pp 309–312
12. Yang MH, Roth D, Ahuja N (2000) A SNoW-based face detector. In: Advances in neural information processing systems (NIPS'1999) vol 12:855–861
13. Juell P, Marsh R (1996) A hierarchical neural network for human face detection. Pattern Recogn 29(5):781–787
14. Rowley HA, Baluja S, Kanade T (1998) Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell 20(1):23–38
15. Lin SH, Kung SY, Lin LJ (1997) Face recognition/detection by probabilistic decision-based neural network. IEEE Trans Neural Netw 8(1):114–132
16. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: International conference on computer vision and pattern recognition, vol 1, pp 511–518
17. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38(4):art No 13
18. Comaniciu D, Ramesh V, Andmeer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–575
19. Shi J, Tomasi C (1994) Good features to track. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 593–600
20. Tao H, Sawhney H, Kumar R (2002) Object tracking with bayesian estimation of dynamic layer representations. IEEE Trans Pattern Anal Mach Intell 24(1):75–89
21. Sung EC, Youn JL, Sung JL, Kang RP, Jaihie K (2010) A comparative study of local feature extraction for age estimation. In: IEEE international conference on control automation robotics & vision (ICARCV'2010), pp 1280–1284
22. Thukral P, Mitra K, Chellappa R (2012) A hierarchical approach for human age estimation. In: IEEE international conference on acoustics, speech and signal processing (ICASSP'2012), pp 1529–1532
23. Guodong G, Guowang M (2010) Human age estimation: what is the influence across race and gender. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW'2010), pp 71–78
24. Zhen L, Yun F, Huang TS (2010) A robust framework for multiview age estimation. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW'2010), pp 9–16
25. Guodong G, Xiaolong W (2012) A study on human age estimation under facial expression changes. In: IEEE conference on computer vision and pattern recognition (CVPR'2012), pp 2547–2553
26. Wang HL, Yau WY, Chua XL, Tan YP (2010) Effects of facial alignment for age estimation. In: IEEE international conference on control automation robotics & vision (ICARCV'2010), pp 644–647
27. Kriegman D, Yang MH, Ahuja N (2002) Detecting faces in images: a survey. IEEE Trans Pattern Anal Mach Intell 24(1):34–58
28. Hjelmas E (2001) Face detection: a survey. Comput Vis Image Underst 83(3):236–274
29. Zhao W, Chellappa R, Phillips P, Rosenfeld A (2003) Face recognition: A literature survey. ACM Comput Surv (CSUR'2003) 35(4):399–458

30. Open Source Computer Vision Library. http://sourceforge.net/projects/opencvlibrary/. Accessed 14 June 2014
31. Lucas B, Kanade T (1981) An iterative image registration technique with an application to stereo vision. Imaging Understanding Workshop, pp 121–130
32. Phillips PJ (2000) The FERET evaluation methodology for face recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22(10):1090–1104
33. Burges C (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2:121–167
34. Gao H, Davis J (2006) Why direct LDA is not equivalent to LDA. Pattern Recogn Lett 39 (5):1002–1006
35. Ricanek K, Tesafaye T (2006) MORPH: a longitudinal image database of normal adult age-progression. In: IEEE 7th international conference on automatic face and gesture recognition, pp 341–345
36. The FG-NET Aging Database. http://www-prima.inrialpes.fr/FGnet/html/benchmarks.html. Accessed 14 June 2014

Chapter 4
Panorama Construction from Multi-view Cameras in Outdoor Scenes

Lakhmi C. Jain, Margarita N. Favorskaya and Dmitry Novikov

Abstract The applications of panoramic images are wide spread in computer vision including navigation systems, object tracking, virtual environment creation, among others. In this chapter, the problems of multi-view shooting and the models of geometrical distortions are investigated under the panorama construction in the outdoor scenes. Our contribution are the development of procedure for selection of "good" frames from video sequences provided by several cameras, more accurate estimation of projective parameters in top, middle, and bottom regions in the overlapping area during frames stitching, and also the lighting improvement of the result panoramic image by a point-based blending in a stitching area. Most proposed algorithms have high computer cost because of mega-pixel sizes of initial frames. The reduction of frames sizes, the use of CUDA technique, or the hardware implementation will improve these results. The experiments show good visibility results with high stitching accuracy, if the initial frames were selected well.

Keywords Panorama construction · Image stitching · Projective transformation · Image selection · Robust detectors · Retinex algorithm · Texture blending

L.C. Jain (✉)
University of Canberra, Canberra, ACT 2601, Australia
e-mail: lakhmi.jain@unisa.edu.au

M.N. Favorskaya · D. Novikov
Siberian State Aerospace University, Institute of Informatics and Telecommunications,
31 Krasnoyarsky Rabochy, Krasnoyarsk 660014, Russian Federation
e-mail: favorskaya@sibsau.ru

D. Novikov
e-mail: novikov_dms@sibsau.ru

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_4

4.1 Introduction

The panoramic representation of outdoor scenes is required in many applications involving computer vision systems in robotics [1, 2], industry, transport, surveillance systems [3], computer graphics, virtual reality systems [4], medical applications [5], etc. A digital panorama can be obtained by two ways: the use of a special panoramic camera or the analysis of many images received from a regular camera. On the one hand, the specified sensors, e. g. panoramic lens [6] or a fisheye [7] with a wide Field Of View (FOV) were applied at the first years of digital image processing. These sensors have high cost and provide images or videos with substantial distortions, which are not suitable for a user. On the other hand, a set of images can be provided by a single camera maintained on a tripod and rotated through its optical center, by using a single omni-directional camera, by multiple cameras pointing in different directions, or using a stereo panoramic camera. In the first case, the panoramic mapping and tracking is based on the assumption that only rotational movements of camera are available. This assumption allows the mapping of current frame onto a cylinder to create 2D panoramic image. A set of images or frames from video sequences ought to be composed to create a single panoramic image.

The process of panorama construction includes three main steps as mentioned below:

- The image acquisition, when a series of overlapping images are acquired [8].
- The images alignment, for which the parameters of geometrical transformations are required [9].
- The images stitching, when all aligned images are merged to create a composite panoramic image with the color correction [10].

These three steps have different content according to mono-perspective and multi-perspective panoramic images. The mono-perspective shooting means that the images are produced with a tripod-mounted camera. A fixed focal point is situated in the center of projection. Only viewing direction is altered by camera rotations around vertical or vertical/horizontal axes. The multi-perspective shooting is taken from changing viewpoints and provides the patches, from which a panoramic image consists. This makes a seamless stitching procedure very difficult or even impossible because a scene changing according to various viewpoints cannot be aligned in a common case. Most of researches apply a conception of mosaics, which will be considered in Sect. 4.3. The smart approach connects with morphing or image metamorphosis application. This idea was proposed by Beier and Neely [11] and then was developed by Haenselmann et al. in the research [12]. The image in the middle of the metamorphosis is called the interpolated image. Then the color values are also interpolated and displayed in a panoramic image pixel by pixel. Such morphing requires a straightening of some lines in the panoramic image or reconstruction the textured non-visible regions. This issue is required in the following development to make a panoramic image more realistic.

Additionally the forth step of panorama construction can be mentioned. This is a panorama improvement because the lighting and the color alignment are often required to compensate the shadows and the light-struck regions in separate images. Such step determines a final visibility of panoramic image.

The chapter is organized as follows. Section 4.2 explains a problem statement of multi-view shooting and the models of possible geometrical distortions. Related work is discussed in Sect. 4.3. Section 4.4 provides a procedure of the intelligent frames selection from multi-view video sequences. Section 4.5 derives the correspondence of reliable feature points for a seamless stitching in selected frames. The issues of panorama lighting improvement are located in Sect. 4.6. The experimental results and calculations for feature points detection and projective parameters are situated in Sect. 4.7. Section 4.8 summarizes the presented work and addresses the future development.

4.2 Problem Statement

Various types for receiving a set of initial images, discussing in Sect. 4.1, always have the geometrical or the geometrical/lighting distortions. The ideal case during the outdoor shooting is practically absent (Fig. 4.1a). A scheme of geometrical distortions appearing by a hand-held shooting is presented in Fig. 4.1b. The multi-cameras shooting, when cameras maintain on a moving platform, has similar distortions caused by vibrations and deviations under a movement in the real environment. If cameras are calibrated, then a panoramic image construction occurs faster and more accurate. However, the extended task—the use of non-calibrated cameras is more interesting for practice.

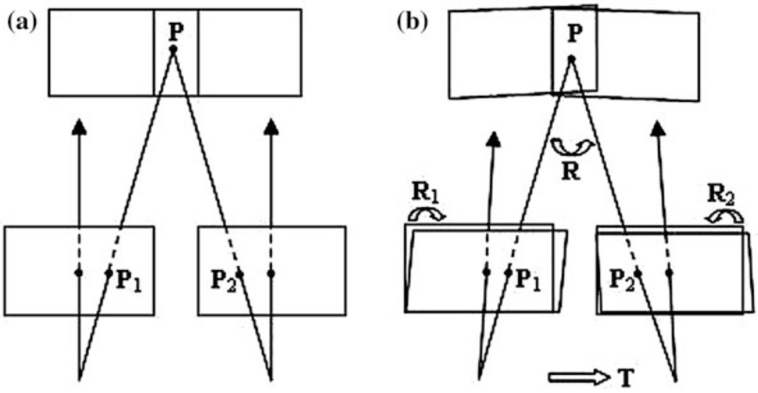


Fig. 4.1 Scheme of a hand-held shooting with two points of shooting: **a** ideal vertical parallel image planes, **b** rotation and translation of image planes, which cause the geometrical distortions in images

One of the main tasks is to determine the parameters of such geometrical distortions. In literature, there are many models from the simple to the complicated ones including the perspective and the homographic transformations. If the possibility of stereo calibration exists, then a rotation matrix **R** and a translation matrix **T** are connected by Eq. (4.1), where $\mathbf{P}_1$ and $\mathbf{P}_2$ are points from two images in Euclidian coordinate system with coordinates (x_1, y_1) and (x_2, y_2), respectively (Fig. 4.1).

$$\mathbf{P}_1 = \mathbf{R}^T(\mathbf{P}_2 - \mathbf{T}) \tag{4.1}$$

The mapping from coordinate $\mathbf{p} = (x, y, f)$ to 2D cylindrical coordinates (θ, h) are calculated by Eq. (4.2), where θ is a panning angle, h is a scanning line, f is a focal length of a regular camera.

$$\theta = \arctan\left(\frac{x}{f}\right), \qquad h = \frac{y}{\sqrt{x^2 + f^2}} \tag{4.2}$$

The cylindrical projection is easy to calculate, when the focal length is known and is not changed. However, this model does not consider the camera rotations around vertical axis. The spherical projection has enough degree of freedom and allows both vertical and horizontal rotations. An artifact of both approaches is a disturbing fish eye-like appearance in the synthesized image.

Usually an eight-parameter planar projection is used under the assumption that a camera has small rotations. The eight parameters are characterized a rotation within an image plane, a perspective turn in vertical and horizontal directions, a scaling factor, and the horizontal and vertical translations. For two points with coordinates (x_i, y_i) and (x_j, y_j) in two images, a perspective transformation has a view expressed by Eq. (4.3), where **H** is a 3 × 3 invertible non-singular homography matrix (homographies and points are defined as a non-zero scalar).

$$\begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.3}$$

In the case of projective transformation, the above Eq. (4.3) can be written as Eq. (4.4), $h_9 = 1$.

$$x_j = \frac{h_1 x_i + h_2 y_i + h_3}{h_7 x_i + h_8 y_i + 1}, \quad y_j = \frac{h_4 x_i + h_5 y_i + h_6}{h_7 x_i + h_8 y_i + 1} \tag{4.4}$$

A pair of corresponding points provides two following equations (Eq. 4.5).

$$\begin{aligned} h_1x_i + h_2y_i + h_3 - h_7x_ix_j - h_8x_jy_i = x_j \\ h_4x_i + h_5y_i + h_6 - h_7x_iy_j - h_8y_iy_j = y_j \end{aligned} \qquad (4.5)$$

To compute eight parameters of **H** matrix (excluding h_9), the four-point correspondences are required. The contribution of this chapter includes the detection of accurate feature points correspondences, depending from an image content. Also the issue of lighting alignment after images stitching will be considered in details.

4.3 Related Work

Two main approaches to create the panoramic images are known. The first one is based on the patches [13–15] and the second one uses the feature points [1, 16]. However, the goal of these approaches is to find the "best" patches or feature points and their the "best" correspondence in the overlapping area of two neighbor images. Let us consider the related works according to both approaches.

The creation of an image mosaic is a popular method to increase a field of camera view, combining the several views of scene in the panoramic image. Such algorithms direct on the normalization of projective transformations between images called the homography. However, the creation of high quality mosaic requires the color and lighting corrections to avoid seams caused by the moving objects and the photometric variations across the blending boundaries. Kim and Hong formulated the blending problem as a labeling problem of Markov Random Fields (MRF) [17]. The authors used the image patches centered at the regular spatial grid. In each grid, a collection of image patches from the registered images are generated.

To find a smooth and consistent with the collected patches, Kim and Hong proposed to define an energy function $E(f)$ using the MRF formulation given by Eq. (4.6), where N is a set of pairs of neighboring grids, P is a grid space representing a pixel location in the image plane, $D_p(f_p)$ is a data penalty function in grid p (patch location) with label (image number) f_p, and $\mathrm{V}_{p,q}(f_p, f_q)$ is a pairwise smoothness constraint function of two neighboring grids p and q with labels f_p and f_q, respectively.

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{pq \in N} V_{p,q}(f_p, f_q) \qquad (4.6)$$

The energy $E(f)$ can be minimized by using the graph-cuts algorithm. Kim and Hong used so called α-expansion graph-cuts algorithm, which minimizes an energy function by the cyclically iterations through every possible α label. During an iteration, the algorithm determines a possibility of improvement for the current labeling by changing of some grids. The penalty function $D_p(f_p)$ is determined by Eq. (4.7), where C is a constant, $B_\Omega(i, j)$ represents a similarity measure (an average value) between two images I_i and I_j in the patch region Ω.

$$D_p(f_p) = \sum_{\mathrm{k} \neq \mathrm{f_p}} \min\left(B_{\Omega_p}(k, f_p),\ C\right) \tag{4.7}$$

The smoothness constraint $V_{p,q}(f_p, f_q)$ in the overlapping region Ω_{pq} between neighboring patches is defined by empirical dependence expressed by Eq. (4.8), where $S_{p,q}(f_p, f_q)$ is a general form of a smoothness cost (Eq. 4.9), $R_{p,q}(f_p, f_q)$ encourages the labels to consist of a small number of uniform regions reducing the noise-like patterns and simplifies the shape of boundaries (Eq. 4.10), λ_s is a smoothness coefficient, λ_r is used to encourage the labels with a small number of the uniform regions. The recommendations of Kim and Hong are $\lambda_s = 1$ and $\lambda_r = 10$.

$$V_{p,q}(f_p, f_q) = \lambda_s S_{p,q}(f_p, f_q) + \lambda_r R_{p,q}(f_p, f_q) \tag{4.8}$$

$$S_{p,q}(f_p, f_q) = B_{\Omega_{p,q}}(f_p, f_q) \tag{4.9}$$

$$R_{p,q}(f_p, f_q) = \begin{cases} 1 & f_p \neq f_q \\ 0 & o.w. \end{cases} \tag{4.10}$$

The MRF approach was applied to analyze the neighboring patches with moving objects. Also a simple exposure algorithm was proposed to reduce seams near the estimated boundaries by the correction of intensities.

Also a spectral analysis can be applied for patches detection and correspondence. The robust panoramic image mosaics in the complex wavelet domain were presented by Bao and Xu [18]. The panoramic mosaics are built by a set of alignment transformations between the image pixels and the viewing direction. The full planar projective panorama and the cylindrical panoramic image, which is warped in the cylindrical coordinates (assuming that a focal length is known and a camera has a horizontal position), are considered in this research. The multispectral mosaicing for enhancement of spectral information from the thermal infra-red band and visible band images, directly fused at the pixel level, was suggested by Bhosle et al. [19]. The authors developed a geometric relationship between the visible band of panoramic mosaic and the infrared one. A creation of panoramic sequences with video textures had been considered by Agarwala et al. [20] based on the smoothness constraints between patches with a low computational cost.

Let us consider the second approach, connecting with extraction of feature points. The matching of the corresponding points between two and more images is used successfully in many computer vision tasks particularly in panorama construction [21]. The detection and description of feature points are the connected procedures. The distinctiveness and the robustness to photometric and geometrical transformations are two main criteria for feature point extraction [22]. Deng et al. considered the using of the panorama views instead of standard projection views in 3D mountain navigation system [23]. This approach is based on the features of interest. The authors specify the reference points located between the features of interest, trace them along the line of sight, and then determine the visibility of these features in 3D mountain scene. This permits to avoid occlusions of close objects by using a perspective projection for rendering of 3D spatial objects.

A modified Hough transform for the feature detection in panoramic images was proposed by Fiala and Basu [24]. The authors modified the well known Hough transform in such manner that only horizontal and vertical line segments were detected by the edge pixels mapping in a new 2D parameter space. The recognition of horizontal and vertical lines is very useful for mobile robots navigation in urban environment, where the majority of line edge features are either horizontal or vertical.

An efficient method to create the panoramic image mosaics with multiple images was proposed by Kim et al. [25]. This method calculates the parameters of a projective transformation in the overlapped area of two given images by using four seed points. The term “seed point” means the highly textured point in the overlapped area of the reference image, which is extracted by using a phase correlation. The authors assert that because a Region Of Interest (ROI) is restricted in the overlapped areas of two images, more accurate correspondences can be obtained. The proposed algorithm includes the following steps:

- The input a pair of images.
- The extraction of overlapped areas by using a phase correlation method.
- The histogram equalization and the selection of seed points.
- The detection of seed points’ correspondence by using the weighted Block Matching Algorithm (BMA).
- The parameters calculation of projective transformation.
- The estimation of focal length.
- The mapping of image coordinates in the cylindrical coordinates.
- The calculation of displacement between two images.
- The blending of warped images using a bilinear weight function.

The parameters of a projective transformation are determined by using four seed points, which ought to exist in the overlapped area and not more then three of them should be collinear. The overlapped area is divided into four sub-areas, where the central pixel of the *k*th block with maximum variance is selected as the seed point q_i into the *i*th sub-area. Equation (4.11) provides a seed point q_i calculation, where $\sigma^2_{k,i}$ is a variance, $M_{k,i}$ is a mean value of the *k*th block into the *i*th sub-area, h_g is a histogram of a gray level g values, and G_{max} is a maximum of gray level value ($G_{max} = 255$).

$$q_i = \operatorname{agr} \max_k \left| \sigma^2_{k,i} \right|, \quad 0 \le i \le 3$$
$$\sigma^2_{k,i} = \sum_g^{G_{\max}} g^2 \cdot h_g - M^2_{k,i} \tag{4.11}$$

Such approach, representing by Kim et al. is enough simple. However, if the overlapped areas are the high textured regions, then Eq. (4.11) can provide several seed points with equal values of variance of the *k*th block into the *i*th sub-area, and the choice of single point will be non-determined. Also as any statistical model, the proposed decision is a noise dependent.

More common approach was proposed by Zhu et al. as a 2D/3D realistic panoramic representation from video sequence with dynamic and multi-resolution capacities [26]. The authors investigated the moving objects in the static scenes, which are taken by a hand-held camera undergoing 3D rotation (panning), zooming, and small translation. In this case, the motion parallax can be a non-zero value, but it is neglected due to small translation. The authors assumed that the camera covers the full 360° field of view around the camera, and the rotation is almost around its nodal point. The algorithm includes three steps such as the interframe motion estimation, the motion accumulation and classification, and a Dynamic and Multi-Resolution (DMR) 360° panoramic model generation.

The camera motion has six degrees of freedom: three translation and three rotation components. A coordinate system XYZ is attached to the moving camera with the origin O in the optical center of camera. An alternative interpretation is a scene movement with six degree of freedom. The authors represented three rotation angles (roll, tilt, and pan) through the interframe difference by (α, β, γ) into a rotation matrix **R** and three translation components $\mathbf{T} = (T_x, T_y, T_z)^{\mathrm{T}}$. 3D point $\mathbf{X} = (x, y, z)^{\mathrm{T}}$ with the image coordinates $\mathbf{u} = (u, v, 1)^{\mathrm{T}}$ (UOV is a plane of image) at time instant *t* in the current frame moved from point $\mathbf{X}' = (x', y', z')^{\mathrm{T}}$ at the time instant *t* in the reference frame with the image coordinates $\mathbf{u}' = (u', v', 1)^{\mathrm{T}}$. The relation between the 3D coordinates is expressed by Eq. (4.12).

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T} \tag{4.12}$$

These authors [26] used the simplified 2D rigid interframe motion model provided by Eq. (4.13), where $s \approx f/f'$ is a scale factor associated with zoom and Z-translation, f' and f are the camera focal lengths before and after the motion, $(t_u, t_v) \approx (-\gamma_f, \beta_f)$ is a translation vector representing pan/X-translation and tilt/Y-translation, and α is a roll angle.

$$\begin{cases} s \cdot u' = u + \alpha v + t_u \\ s \cdot v' = v - \alpha u + t_v \end{cases} \tag{4.13}$$

The motion model, representing by Eq. (4.13), is actual in far away scenes. The least square solution of motion parameters s, t_u, t_v, and α can be obtained by given more than two pairs of corresponding points between two frames. The errors of approximation can be corrected by the mosaic algorithm [27, 28]. The color frames had the size 384 × 288 pixels that permitted to process five frames per second.

Langlotz et al. proposed to use the Features from Accelerated Segment Test (FAST) keypoints for the mapping and tracking purposes [29]. The FAST keypoints are extracted at each frame to create the panoramic map. The authors try to extend the dynamic range of images by the choice of the anchor camera frame and the using of pixel intensities as a baseline for all further mappings. Then the differences of intensities are computed for all pairs of matching keypoints in the original frame and the panoramic map. The average difference of these point pairs is used to improve the current frame before its inclusion in the panorama image.

Sometimes the panoramic view morphing is used to generate a set of new images from different points of view, if two basic views of static scene uniquely determine a set of views on the line between the optical centers of cameras, and a visibility constraint is satisfied. Seitz and Dyer proved this statement in their research [30]. Often the panoramic images are required in medical applications for improvement the ultrasound images, in ophthalmology, etc. A fast and automatic mosaicing algorithm to construct the panoramic images for cystoscopic exploration was presented by Hernández-Mier et al. [31]. The algorithm provides the panoramic images without affecting the application protocol of cystoscopic examination.

Method of optimal stitching based on tensor analysis was proposed by Zhao et al. in the research [10]. This method has the advantage of causing less artifacts in the final panorama despite the presence of complex radiometric distortions such as vignetting by a new function of seam cost. An approach for image stitching based on Hu moments and Scale-Invariant Feature Transform (SIFT) features is presented by Fathima et al. [32]. The authors propose to determine the overlapping region of images using the combined gradient and invariant moments to reduce the processing area of features extraction. The selection of matching regions is realized by gradient operators based on the edge detection. After definition of the dominant edges, the images are partitioned in equal sized blocks and compared with each other to find the similarity. Then the SIFT features are extracted, and their correspondence are defined by RANdom SAmple Consensus (RANSAC) algorithm [33]. The authors assert that their approach reduces 83 % of unreliable features.

Yan et al. combined the probability models of appearance similarity and keypoint correspondences in a maximum likelihood framework, which is named as Homography Estimation based on Appearance Similarity and Keypoint correspondences (HEASK) [34]. The probability model of the keypoint correspondences is based on the mixture of Laplacian distributions and the uniform distribution, which is proposed by the authors. Also the authors built a probability model of the appearance similarity. The authors named the method with only the term of a keypoint correspondence fitness as the HEASK-I, and named the method with only the term of an image appearance similarity as the HEASK-II. The experimental results show that the HEASK achieves a good trade-off between the keypoint correspondences and the image similarity in comparison of the RANSAC, Maximum Likelihood Estimator SAC (MLESAC) [35], and the Logarithmical RANSAC (Lo-RANSAC) [36].

The analysis of related works shows that some complex issues of panorama creation especially through the multi-view cameras shooting require the additional research and development. This helps to achieve the reliable results in the construction of panoramic image acceptable for user surveillance in various tasks of computer vision.

4.4 Intelligent Selection and Overlapping of Representative Frames

Let several video sequences be entered from the multi-view cameras, which are maintained on a moving platform or vehicle. All six parameters including three angles and three coordinates in a camera 3D coordinate system relatively a platform 3D coordinate system are known. The main requirement connects with the overlapping regions between images, approximately 10 % of image area. The conditions of real shooting such as vibrations, deviations of lens systems, lighting change, etc. damage the ideal calculation.

The primary efforts are directed on the background analysis of image. This stage involves many methods and processing algorithms and has a high computational cost. The selection of representative frames provides the "good" frames from video sequences without artifacts. This procedure is represented in Sect. 4.4.1. An overlapping analysis of selected frames is discussed in Sect. 4.4.2.

4.4.1 Selection of Representative Frames

Let a moment of multi-cameras shooting is determined but with the ms error because cameras can be the non-synchronized, have different constructive types, and various access time for writing a video sequence in the inner device. Also a process of panorama creation has a large duration, and the sampling of forth-sixth panoramic images per s is a good result in a current experimental stage. The main acquisitions of representative frames selection are mentioned below:

- The selected frame is a sharp fame.
- The selected frame ought to have an equal median lighting.
- The selected frame ought to be contrast and has a color depth.
- An overlapping area between two frames from neighbor cameras ought to have a maximum value.

- It is desired that the vertical and horizontal lines in a frame satisfy to affine or perspective transformations.

For these purposes, a set of well known filters such as Laplacian [37], high dynamic range [38], and morphological [39] filters, also a Hough transformation to detect the vertical and horizontal lines [40] can be applied automatically in a parallel mode. The received estimations have a different contribution in the final decision. For example, a blurred frame ought to be rejected immediately. A frame with the non well-defined lines would like to be changed. The parameters of lighting, contrast, and color can be improvement during the following processing. The overlapping area strongly depends from the cameras positions and can be tuned preliminary. It is required to track the low threshold value of overlapping area. In spite of some computational cost of such preprocessing filtering, this step determines the accuracy of final panoramic result.

4.4.2 Overlapping Analysis of Selected Frames

For small cameras rotations, the eight-planar perspective transformation is the most appropriate approach. If the translation is a non-linear, then the perspective transformation (Eq. 4.3) by homogeneous coordinates introduced by Maxwell [41, 42] and later applied to computer graphics by Roberts [43] is often used. Equation (4.14) provides such homogeneous transformation, where w is a warping parameter.

$$\begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ w \end{bmatrix} \tag{4.14}$$

One can see the examples of frames overlapping in the case of the affine model (Fig. 4.2) and the perspective model (Fig. 4.3).

The homogeneous transformation is more complex in comparison of other procedures. The complexity of such transformations causes the necessity to find a way of frames matching with a high accuracy. The application of classical pattern recognition methods is non-useful here because of the unknown distortions of objects views. In this case, patches or feature points matching are recommended. In recent researches, the feature points correspondence is often used for the seamless stitching of selected frames. These issues will be discussed in Sect. 4.5.

The overlapping under the projective and the homogeneous transformations leads to the parallax effects. Let us notice that the affine transformation is the simplest case, which is rarely met in the panorama construction, and has not significant problems during frames stitching. Some examples of parallax effect are

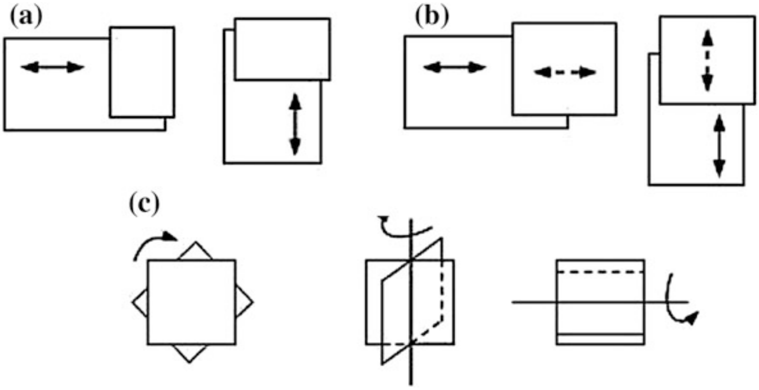


Fig. 4.2 Variants of frames overlapping under the affine transformation: **a** horizontal and vertical translations, **b** horizontal and vertical scaling and translations, **c** rotations relatively three coordinate axes

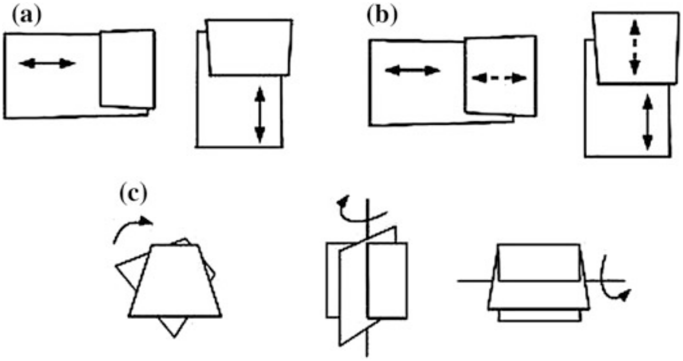


Fig. 4.3 Variants of frames overlapping under the projective transformation: **a** horizontal and vertical translations, **b** horizontal and vertical scaling and translations, **c** rotations relatively three coordinate axes

situated in Fig. 4.4. The experiments show the interesting property: if the initial frames will be reduced, for example, in two or four times, then some artifacts disappear (Figs. 4.4e, f, g). It may be explain that the scaled down panoramic images have more “rough” structure of feature points, and their number is significantly reduced in comparison with the initial image. However, the quality of such fragments has been lost. Let us discuss the problems and possible decisions for seamless stitching of the selected frames.

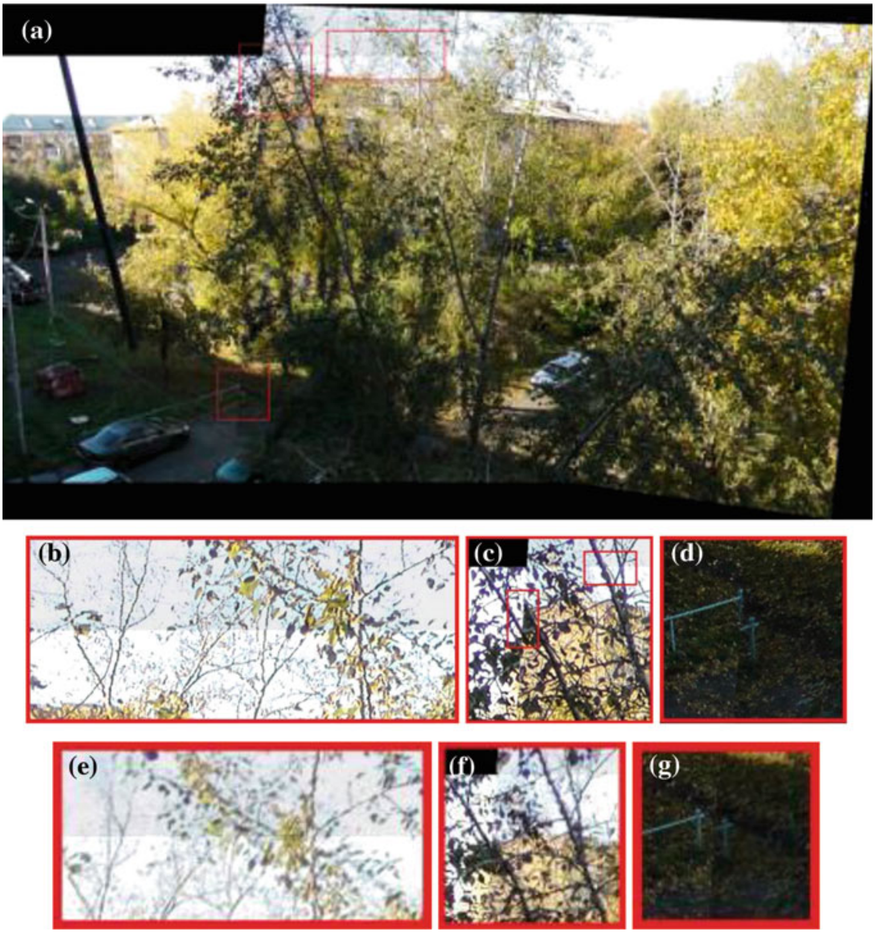


Fig. 4.4 A panoramic creation with parallax effects, the item "Trees from Window": **a** stitching panorama, **b** *top* good stitching (*without lighting alignment*), **c** *top* artifact, **d** *bottom* artifact, **e** *top* good stitching in reduced image, **f** *top* good stitching in reduced image, **g** *bottom* artifact in reduced image

4.5 Stitching of Selected Frames

A stitching of frames is the main core in the panorama construction, which includes the procedures of similar regions detection, similar regions matching, and some additional procedures determined by the content of selected frames. In Sect. 4.3, two main approaches were considered to determine the similar regions. More rough approach connects with so called "patches", which are usually regions from 3 × 3 pixels to 11 × 11 pixels. More accurate approach is based on the technique of feature points detection. Great variety of feature detectors determines various relations accuracy/computational cost. In this research, the feature points approach is developed. To accelerate a computational time, a pre-filtering was used before standard RANSAC application. In recent years, RANSAC or its modifications remain the basic algorithm to estimate the feature points correspondence.

The additional procedures are executed in the particular cases, for example, when the object with a high speed is situated in all selected frames, and it is required to understand its location in the panoramic image. Sometimes fragments in the selected frames have such big warping that the interpolation or the morphing procedures provide the better panoramic image. The feature points detection, matching, and correspondence will be discussed in Sects. 4.5.1–4.5.3, respectively. Section 4.5.4 provides the image projection and the geometrical improvement of panorama. A visualization of high speed objects in panoramic images is briefly considered in Sect. 4.5.5.

4.5.1 Feature Points Detection

To create an operator, which combines the feature points detection and the lighting alignment in stitching images simultaneously, is possibly in the future. At present, they are two original procedures. The analysis of the existing feature points detectors in panorama construction show that Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Feature (SURF), and corner detectors are the most popular among others. Our recommendations are to use the SURF detector (because it is faster than the SIFT detector while both detectors demonstrate the close results) in the landscape outdoor scenes and the corner detectors jointly with a Hough transform (to find lines and correct a parallax effect) in the indoor scenes.

The SURF is a fast modification of the SIFT, which detects feature points by using the Hessian matrix. The determinant of the Hessian matrix achieves the extremum in the point of maximum changing of intensity gradient. The SURF detects well spots, corners, and edges of lines. The Hessian is invariant to rotation and intensity, but not invariant to scale. The SURF uses a set of scalable filters for the Hessian matrixes calculation. The gradient in a point is computed by the Haar filters. The feature points detection is based on a calculation for determinant of the Hessian matrix det(**H**) by Eq. (4.15), where $L_{xx}(P, \sigma)$, $L_{xy}(P, \sigma)$, $L_{yx}(P, \sigma)$, and $L_{yy}(P, \sigma)$ are convolutions the second derivative of Gaussian $G(P)$ with a function describing a frame I_p in a point P along OX axis, diagonal in the first quadrant, OY axis, and diagonal in the second quadrant, respectively, σ is a mean square. Equation (4.16) provides an example of the convolution along OX axis.

$$\det(\mathbf{H}(P,\sigma)) = L_{xx}(P,\sigma)\ \cdot L_{yy}(P,\sigma) - \left(L_{xy}(P,\sigma)\right)^2 \tag{4.15}$$

$$L_{xx}(P, \sigma) = \frac{\partial^2}{\partial x^2} G(P) * I_P \tag{4.16}$$

In practice, the SUFR uses a binary approximation of Gaussian Laplacian, which is called Fast Hessian, and Eq. (4.15) is replaced by Eq. (4.17), where coefficient 0.9 means an approximate character of calculations.

$$\det(\mathbf{H}(P, \sigma)) = L_{xx}(P, \sigma) \ \cdot L_{yy}(P, \sigma) - 0.9 \cdot \left(L_{xy}(P, \sigma)\right)^2 \tag{4.17}$$

The invariance to scale is provided by partitioning a set of scales (9, 15, 21, 27, etc.) in the octaves. Each octave includes four filters with different scales: the first octave involves (9, 15, 21, 27) scales, the second octave involves (15, 27, 39, 51) scales, the third octave involves (27, 51, 75, 99) scales, and etc. The number of octaves is usually equal 5–6. Theoretically, it is enough to cover scales from 1 to 10 in an image with sizes 1,024 × 768 pixels. The octave filters are not calculated for all pixels. The first octave uses the each secondary pixel, the second octave uses the each forth pixel, the third octave uses the each eighth pixel, and etc. The true extremum cannot be agreed with the calculated extremum. For this purpose, a search is initiated (an interpolation by a quadratic function) until a derivative will not be much close to 0.

In such manner, a list of detected feature points is created. The following step is the calculation of orientations. The Haar filters permit to indicate and normalize a vector of orientation. A noise gives the additional gradients in directions, which are different from a direction of the main gradient. Thus, the additional filtrating is required.

After feature points detection, the SUFR creates its descriptors as a vector with 64 elements or with extended 128 elements. Such vectors characterize the gradient fluctuations in surrounding of feature point. For a descriptor creation, a square area $20 \cdot s$, where s is a scale value, is built. The first octave uses the area 40 × 40 pixels around a feature point. This square orients along a priority vector direction and is divided into 16 sub-squares with sizes 5 × 5 pixels. In each sub-square, a gradient is determined by using of the Haar filters. Four components in each sub-square have a view provided by Eq. (4.18), where ΣdX and ΣdY are the sum of gradients, $\Sigma|dX|$ and $\Sigma|dY|$ are the sum of gradients modules.

$$\sum dX, \quad \sum |dX|, \quad \sum dY, \quad \sum |dY| \tag{4.18}$$

Four components from Eq. (4.18), which are multiplied on 16 sub-squares, provide a 64-value descriptor in surrounding of feature point. Additionally, the received values are weighed by the Gauss filter with $\sigma = 3.3 \cdot s$. This is required for reliability of the descriptor to noises in the remote areas from a feature point.

4.5.2 Feature Points Matching

The feature points are detected in the whole frame because a prediction of overlapping area position is impossible, when many frames are stitched. A feature points matching is a step, when only "good" feature points correspondence are remained. The task of matching is realized as a multi-search of nearest neighbor for a selected current point in a feature space. Under a term "nearest neighbor", such point is understood, which locates in a minimal distance to the selected current point. One can use Euclid metric or some others, however it is the difficult task to compare the 64-valued vectors. The good decision is a representation of all detected descriptors as the *k*-dimensional tree (*k-d* tree). The *k-d* tree is such data structure, which divides *k*-dimensional space to order the points in this space. The *k-d* trees are a modification of binary trees, and usually used in a multi-dimensional space. The following operators are applied to the *k-d* trees:

- The *k-d* tree creation.
- The adding of element.
- The removal of element.
- The tree balancing.
- The search of nearest neighbor by using a key value.

For elements matching, it is enough only two operators: the *k-d* tree creation and the search of a nearest neighbor for a couple of input frames. In this research, the following node structure was applied. Each node includes a single point in feature space called a descriptor. Additionally, the following data are stored:

- The index of frame, from which the descriptor was received.
- The index of separating hyperplane.
- The left sub-tree.
- The right sub-tree.

The index of separating hyperplane is calculated during a tree creation. The left and right sub-trees include the necessary references for the data structure. To detect the position of each element, the index of separating hyperplane is used. The simple procedure sorts the input array of descriptors in the increased or decreased manner relative the element having the index of separating hyperplane points. Then the input array of descriptors is divided into two sets—the left and the right. All elements of the left set have a component value, for which the index of separating hyperplane points, less than the element has, which is written in a current node of a tree, and vice versa for the right set (the nonstrict inequalities are used). The hyperplane is described by Eq. (4.19), where i is an index of separating hyperplane, a_i is a value of the ith component of descriptor, which is written in a current node, X_i is the ith component of basis.

$$X_i = a_i \tag{4.19}$$

The last step of algorithm is a recursive call of procedure to create the *k-d* tree for left and right sub-sets, and a loading of descriptors values as the parameters of left and right sub-trees.

The *k-d* trees have a possibility of fast search of nearest neighbor without considering all remaining points. A mean time of one nearest neighbor is estimated as $O(\log n)$, where n is a number of tree elements. A mean time of feature points matching can be estimated as $k \cdot n \cdot O(\log n)$, where k is a number of neighbors (against $k \cdot n^2$ for a full search). The search by the *k-d* trees is the directional search because a current node is chosen under the determined rule. The *i*th component of a target vector is compared the *i*th component in a current node, where i is an index of separating hyperplane. If the *i*th component of target vector less than the *i*th component in a current node, then a search is continue in a left sub-tree, otherwise in a right sub-tree.

If a nearest neighbor was not detected in a close sub-tree, then the search is continued in a far sub-tree. The goal of nearest neighbor algorithm is to find the nearest value, which cannot be the exact value. This specialty is very important for the SURF descriptor. In spite the SURF descriptor determines a small area, which is invariant to many transformations, including a noise up to 30 %, the exact matching between regions in various images is unlikely.

After feature points matching, the procedure of features points estimation is started. Usually the RANSAC or its modifications are used for these purposes.

4.5.3 Feature Points Correspondence

The quality of feature points matching is checked by the RANSAC algorithm. The concept of this algorithm is based on separation of initial data on outliers (noises, failure points, and random data) and inliers (points, which satisfy a model). The iterations of the RANSAC are divided logically in two stages. The first stage is a choice of points and a model creation, and includes the following steps:

- Step 1. The choice of n points randomly from a set of initial points $\mathbf{X}$ or based on some criterion.
- Step 2. The calculation of θ parameters of model using a function M. Such model is called a hypothesis.

The second stage is a hypothesis verification:

- Step 1. The point correspondence for the given hypothesis is checked by using estimator E and threshold T.
- Step 2. The points are marked as inliers or outliers.

- Step 3. After bypass of all points, the algorithm determines a quality of current hypothesis. If a hypothesis is the best, then it replaces the previous best hypothesis.

As a result, the last hypothesis is remained as the best hypothesis. One can read about the RANSAC algorithm in details in the researches [44–49]. As any stochastic algorithm, the RANSAC has some disadvantages. The upper boundary of computing is absent. Therefore, a suitable result cannot be achieved for the acceptable time. Also the RANSAC can determine only a single model for the initial data set with the acceptable probability. For panoramic images, the first disadvantage can be compensated by CUDA technology application or hardware decisions. The second disadvantage is removed by an algorithmic improvement for the projective parameters of panorama image (Sect. 4.5.4).

The research of Yan et al. [34] is the RANSAC development. The authors combined the probability models of appearance similarity and keypoint correspondences in a Maximum Likelihood framework. Such novel estimator is called as Homography Estimation based on Appearance Similarity and Keypoint correspondences (HEASK). The novelties of the HEASK are connected with a distribution of inlier location error, which is represented by a Laplacian distribution, and with the similarity between the reference and transformed image by the Enhanced Correlation Coefficient (ECC) feature. Yan et al. realized their algorithm in a RANSAC-based framework and consistently achieved an accurate homography estimation under different transformation degrees and different inlier ratios.

4.5.4 Image Projection and Geometrical Improvement of Panorama

The RANSAC algorithm provides many feature points correspondences. In the task of panorama construction, it is important to determine the parameters of affine/projective/homography transformations for following recalculation image coordinates. For projective transformation, four corresponding feature points are required. They provide eight coefficients $h_1 \ldots h_8$ from Eq. (4.3). The full homography transformation was not investigated in this research.

The proposed algorithm for selection of corresponding feature points includes two scenarios. The first one selects points in the whole overlapping image area randomly. The second scenario divides the overlapping area in three regions—top, middle, and bottom. The algorithm calculates coefficients of transformation in each region separately. The last approach is required for images with different models of distortions. This is the case of image warping, non-well investigated issue in the theory of computer vision. The joint for different models of distortions in the result panoramic image may be provided by interpolation or warping.

The image projection consists in a multiplication of the coordinate vector for each pixel on the coefficients of transformation matrix. The new pixel coordinates are pointed in the coordinate system connecting with the second image (on which the first image is projected). During such pixels projection, the following problems appear:

- The new pixel coordinates have the fractional values.
- The projective image losses a rectangle shape. The regions appear, in which initial points overlap each other or where the new points appear without any color information.

Therefore, an interpolation task is required. The traditional interpolation methods such as bilinear, bicubic, or spline interpolation cannot be recommended because of high computational cost [50, 51]. In this research, a type of linear interpolation based on three predetermined points was realized as a simple and fast decision.

The affine transformation provides a non-equaled uniform scale distribution, when a cell of image grid remains a rectangle. Under the projective and homography transformations, a scale distribution is becoming the non-equal and the non-uniform. Therefore, a cell of image grid transforms to a non-regular rectangle. The method of linear interpolation by using three values (vertexes of triangle) is fast algorithm. First, it is required to build such triangles. For this purpose, method of square comparison, vector method, beam tracing, among others, can be applied. In this research, vector method was used as the fastest algorithm. Second, the linear interpolation is executed by using three values. It includes three following steps:

- Step 1. An image is divided in non-overlapping triangles in such manner, that a sum of triangles squares would be equal to a square of total image projection. Then for all triangles Step 2 and Step 3 are executed.
- Step 2. A calculation of distances (d_1, d_2, d_3) from the vertexes of triangle to the required point and a normalization of these distances so that their sum is equaled 1.
- Step 3. A calculation of color function values in the required point as a sum of the weighing function values in the vertexes of triangle (multiplied on the normalized distances).

A procedure of panorama stitching is very complex with different transformation models in top, middle, and bottom parts of overlapping area. It is necessary to use the compulsory lines straightening or the morphing in the final panoramic image. The morphing parameters can be fitted between two unmatched fragments in the initial frames. However, these issues require the following investigations.

4.5.5 Visualization of High Speed Objects in Panoramic Images

In common case, a video sequence involves visual projections of objects moving with high speed. Due to such high speed, an object can appear in several frames selected for panorama construction. Two variants of representation exist:

- Show a moving object in a single from the selected frames. It will be mean that a moving object is a single in a scene.
- Show all views of a moving object in the selected frames. It will map a trajectory of moving object in a scene.

The choice depends from a goal of panorama application. This issue is outline of the current research. Previously, the scenes without objects moving with high speed were used for experiments.

4.6 Lighting Improvement of Panoramic Images

It is difficult to guess, that a shooting is executed under the ideal lighting conditions. Usually, this assumption is inversely. All digital processing methods and algorithms are based on the analysis of intensity or color functions, describing an image. Therefore, the issues of lighting are very important. Two tasks can be formulated relatively to a panorama construction—the enhancement of initial selected frames (Sects. 4.6.1 and 4.6.2) and the improvement of visibility of stitching areas (Sect. 4.6.3).

4.6.1 Application of Enhancement Multi-scale Retinex Algorithm

Between the known approaches for spectrum enhancement of color images such as histogram approach, homomorphic filtering, and the Retinex algorithm, the last one is the most suitable for panoramic images. This is explained that the Retinex (formed from the words "retina" and "cortex") algorithm is the advanced method, which simulates the adaptation of human vision for dark and bright regions [52].

The Single-Scale Retinex (SSR) algorithm demonstrates the best results for a grey-scale images processing and has difficulties for color images. The Multi-Scale Retinex (MSR) algorithm provides the processing of color images. The 1D retinex function $R_i(x, y, \sigma)$ according to the SSR-model calculates differences of logarithmic functions given by Eq. (4.20), where $I_i(x,y)$ is an input image function in the ith spectral channel, c is a scale coefficient, sign "*" represents a convolution of the input image function $I_i(x,y)$, and the surrounding function $F(x, y, c)$. Often the surrounding function $F(x, y, c)$ has a view of Gaussian function including a scale vector $\boldsymbol{\sigma}$.

$$R_i(x, y, \sigma) = \log\{I_i(x, y)\} - \log\{F(x, y, c)^* I_i(x, y)\} \tag{4.20}$$

The MSR-model $R_{Mi}(x, y, \mathbf{w}, \boldsymbol{\sigma})$ in the ith spectral channel is calculated by Eq. (4.21), where $\mathbf{w} = (w_1, w_2, \ldots, w_m)$, $m = 1, 2, \ldots, M$ is a weight vector of 1D Retinex functions in the ith spectral channel $R_i(x, y, \boldsymbol{\sigma})$, $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, $n = 1, 2, \ldots, N$ is a scale vector of 1D output Retinex function. A sum of weigh vector $\mathbf{w}$ components is equaled 1.

$$R_{M_i}(x, y, \mathbf{w}, \boldsymbol{\sigma}) = \sum_{n=1}^{N} \mathrm{w}_n R_i(x, y, \sigma_n) \tag{4.21}$$

The basic SSR and MSR algorithm improve the shadow regions well. However, they are non-useful for bright region processing. The Enhanced Multi-Scale Retinex (EMSR) algorithm based on an adaptive equalization of spectral ranges in dark bright regions simultaneously was developed by Favorskaya and Pakhirka [53]. The EMSR algorithm uses a special curve representing as a logarithmic dependence of image function for low values of intensity and a logarithmic dependence of inverse image function for upper values of intensity.

The application of such algorithms permits to balance the lighting in the initial images. The experiments show that a contour accuracy becomes higher especially in dark regions. However, a total visibility is worse because of too sharp edges of visual objects. For this purpose, the edge smoothing procedure can be recommended.

4.6.2 The Edges Smoothing Procedure

All Retinex similar algorithms increase a sharp of result image significantly. This is not essential for computer processing. However, a visibility of result image can not satisfy the user.

The image sharpness improves the details in a result image by the blurring application. Some known filters solve this problem, for example, the High pass filter, the Laplacian filter, or the Unsharp masking filter. All these filters increase the sharpness values by a contrast amplification of the tonal transitions. The main disadvantage of the High pass and the Laplacian filters consists in sharpening not only image details but also a noise. An unsharp masking filter blurs a copy of original image by Gauss function and determines a subtraction between the received image and input image, if their differences exceed some threshold value.

The Enhanced Unsharp Masking (EUM) filter proposed by Favorskaya and Pakhirka [53] improves the output image by joined compositing of contour performance and equalization performance based on empirical dependences. Let notice that the application of the EMSR algorithm and/or the edge smoothing procedure is not always required in a panoramic image.

4.6.3 Blending Algorithm for Stitching Area

The improvement of visibility in the stitching areas is often required in final panoramic images. A transparency-blending method, based on α-composite transparent surface layers in a back-to-front order to generate the effect of transparency, is not suitable for panorama improvement. The Point-Based Rendering (PBR) approach attracts a high interest in geometric modeling and rendering primitives as an alternative to triangle meshes. The PBR-blending is used to interpolate between overlapping point splats within the same surface layer to achieve the smooth rendering results. Zhang and Pajarola [54] proposed a deferred blending concept, which enables the hardware accelerated transparent PBR with combined effects of multi-layer transparency, refraction, specular reflection, and per-fragment shading.

Recently, the region-based image algorithms were developed to improve the pixel-based algorithms. They include the Poisson fusion algorithm [55], the Laplacian pyramid transform, contrast pyramid, discrete or complex wavelet transform [56], curvelet transform [57], contourlet transform [58], among others. The interesting approach for combining a set of registered images in a composite mosaic with no visible seams and minimal texture distortion was proposed by Gracias et al. [59]. In this research, a modification of the PBR-blending was applied, which includes the steps as mentioned below:

- Step 1. The segmentation of blending area near a stitching line. First, the Weighed Coefficients Masks (WCMs) are built for each of input images. The size of the WCMs is equal to the size of input image. Each element has a back-proportional value of distance between the center of input image and a current point. Then the received WCMs are imposed each other, and their elements are compared to build the Binary Blending Masks (BBMs). If a value of current element in the WCM cannot be compared with any element from the WCM of other image, then the corresponding value of the BBM receives the minimal value. (It means that a current pixel is located in a final image.) If a value of current element in the WCM is less than a value of current element in the WCM of other image, then a maximum value is assigned to the corresponding element of the BBM (a current pixel is not moved in a final image), and vise verse.
- Step 2. After the BBMs forming, these masks are blended by Gauss filter in three sub-bands. Other pixels from the input images are moved in final panoramic image without any changing.

The example of lighting enhancement and seamless blending is situated in Fig. 4.5. Two initial images were hand-held received by using Nikon D5100 camera with the autoexposure.

The following improvement of blending is connected with the multi-scale or the multi-orientation sub-bands framework with a number of bands, not more than 4. The main idea is to blend the sub-bands of image with various blending degree (if a frequency is low, then a blending degree is high).

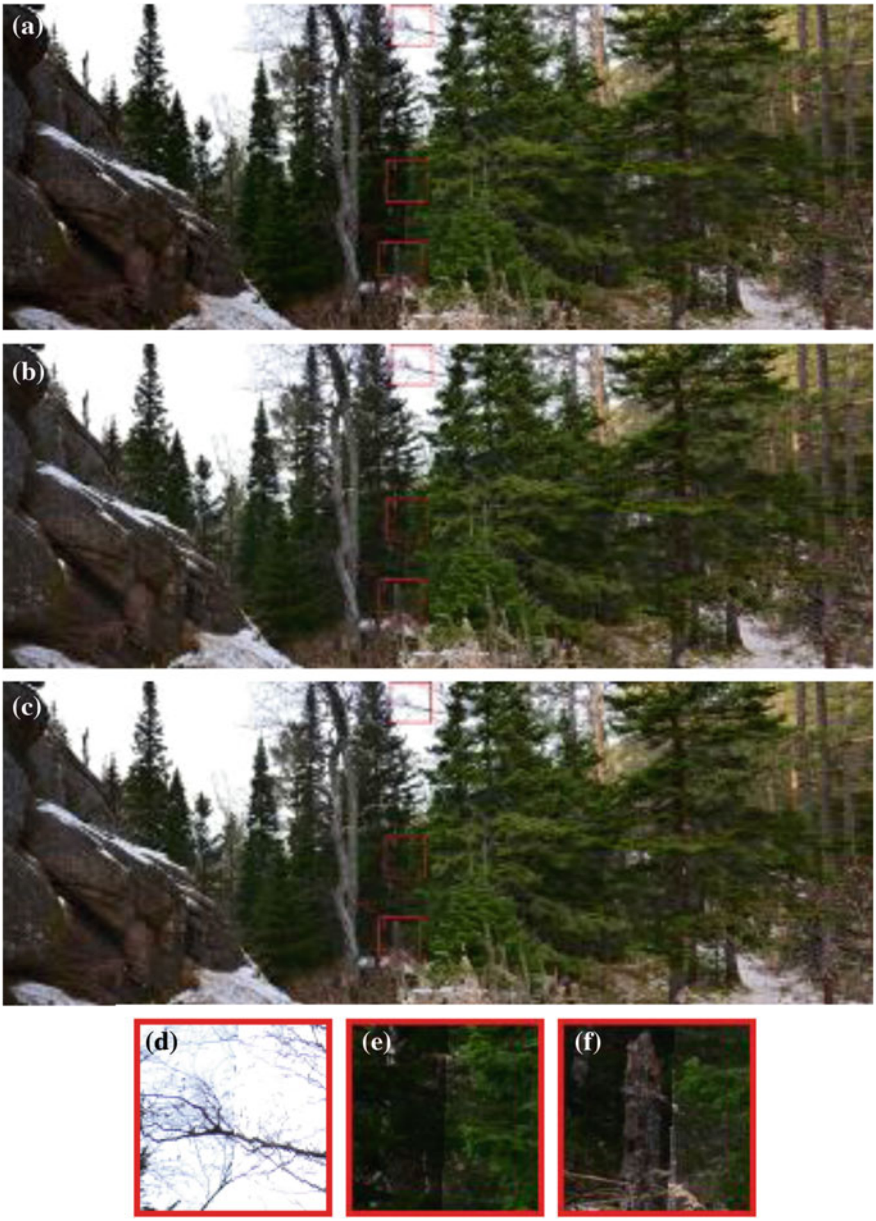


Fig. 4.5 Example of lighting enhancement and the PBR-blending by using of Gauss filter in three sub-bands: **a** stitching panoramic image, **b** stitching panoramic image with lighting enhancement, **c** stitching panoramic image with lighting enhancement and blending, **d**, **e**, **f** fragments with artifacts, **g**, **h**, **i** fragments with lighting enhancement, **j**, **k**, **l** fragments with lighting enhancement and blending

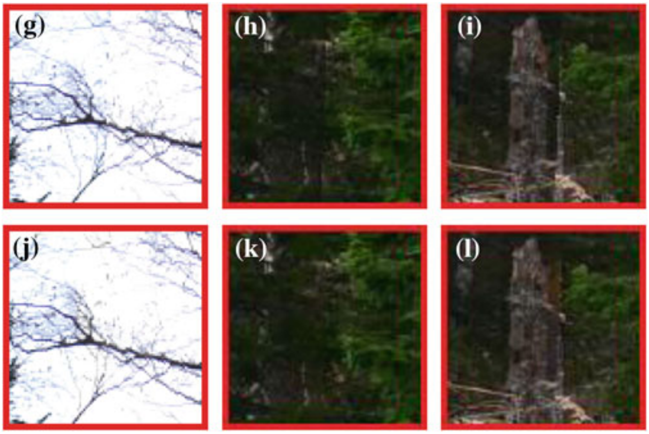


Fig. 4.5 (continued)

4.7 Discussion of Experimental Results

The software tool "Panorama Builder", v. 1.07 supports the main functions for a panorama construction such as frames or images loading, execution of automatic stitching procedure, save of results, and tuning of algorithm parameters. The designed software tool is written on C# language and uses the open libraries OpenCV and EmguCV. The calculations are realized by the central processor card and the graphic card, if it supports NVIDIA/CUDA technology. All discussed and proposed algorithms were realized in this program.

Figures 4.6, 4.7, 4.8, and 4.9 demonstrate a visual tracking of basic algorithmic operations to receive a final panoramic image. The initial images represented in Figs. 4.6, 4.7, 4.8, and 4.9 were received during the hand-held shooting by using Panasonic HDC-SD800 camera with the weighted auto exposure. Figures 4.6 and 4.7 show the close example with bad and good stitching. The bad stitching in Fig. 4.6 can be explained by the unsuccessful selection of frames, when a building tower crane was in motion with different directions of a jib. The final visual results in Figs. 4.7, 4.8 and 4.9 are enough appropriate.

The number of feature point correspondences in left image, right image, and overlapping area as well as the calculation time for different sizes of initial images 640 × 480 (VGA), 800 × 600 (SVGA), 1,024 × 768 (XGA), and 1,280 × 960 from Figs. 4.6, 4.7, 4.8 and 4.9 are presented in Table 4.1. During experiments, the different values of the SURF parameter and a brightness threshold, which influences on a number of feature points correspondences, was applied. The time of homography processing directly depends from a quantity of common feature point correspondences. All results are given for PC configuration: Intel Pentium Dual-Core T4300 @ 2.10 GHz, 3 Gb RAM.

The analysis of data from Table 4.1 shows that the increment of resolution leads to the increased number of feature points detected in the input images. If the number of feature points increases, then the computational cost becomes higher. Therefore, a calculation time for the homography parameters increases. The stitching results are different by a mutual location of fragments for various values of the SURF brightness threshold. However, the lines of stitching save their visibility. The item "Road" is characterized by the increased number of feature points. This is explained by a wide overlapping area of the initial images.

Also the coefficients of homography matrixes for such examples are shown in Table 4.2.

The following investigations were connected with the parameters behavior during the perspective transformation. The results of top, middle, and bottom regions in an overlapping area are situated in Fig. 4.10. In Table 4.3, the coefficients of homography matrix of top, middle, and bottom regions are represented.

As it is seen from Fig. 4.10, the middle and bottom models are close, the top model is differed. This is explained by shooting from the Earth surface. However, such differences have not a significant value in landscape images.

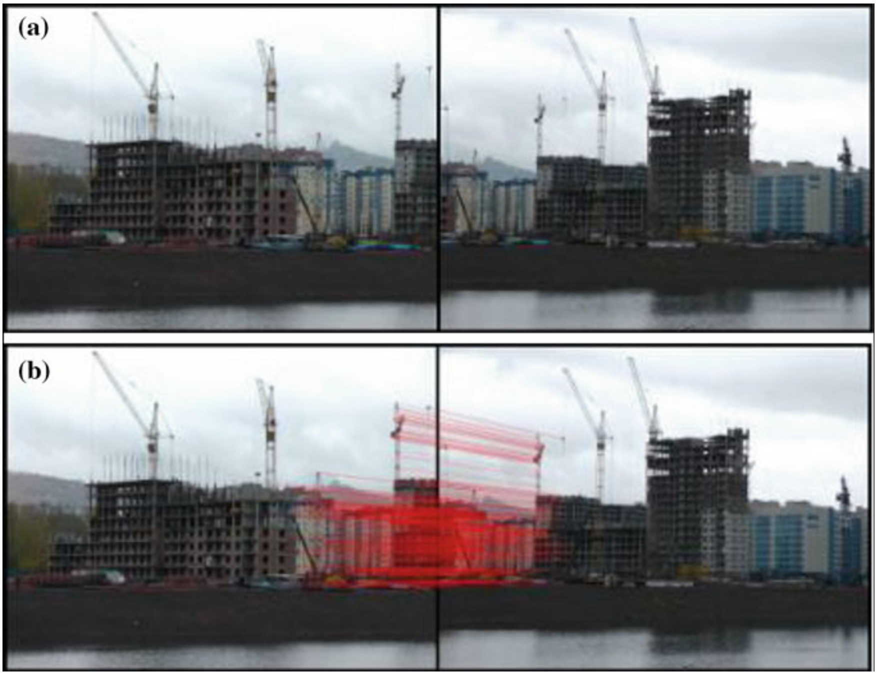


Fig. 4.6 The item "Houses failure": **a** input images, **b** feature points matching, **c** projection result, **d** final panoramic image, **e** *top* fragment with an artifact stitching, **f** good *middle* fragment, **g** *bottom* fragment with an artifact stitching

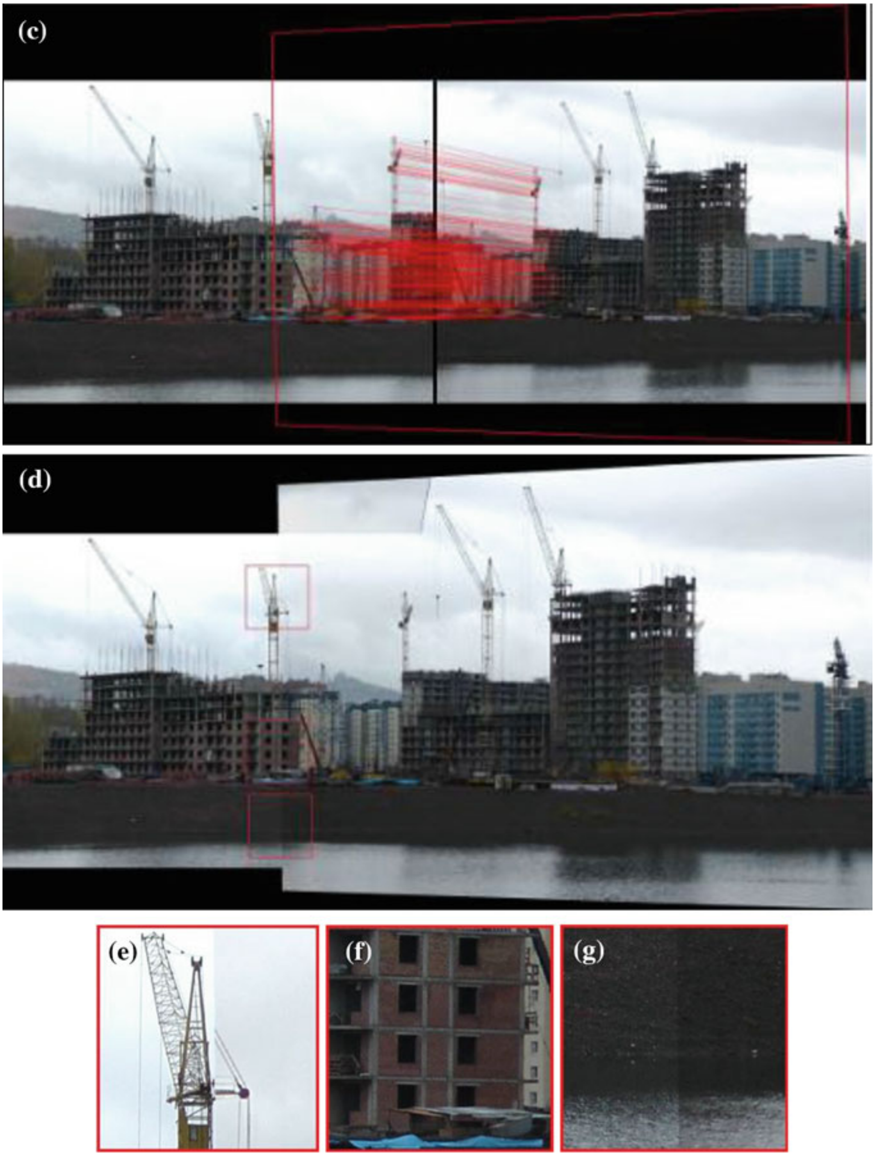


Fig. 4.6 (continued)

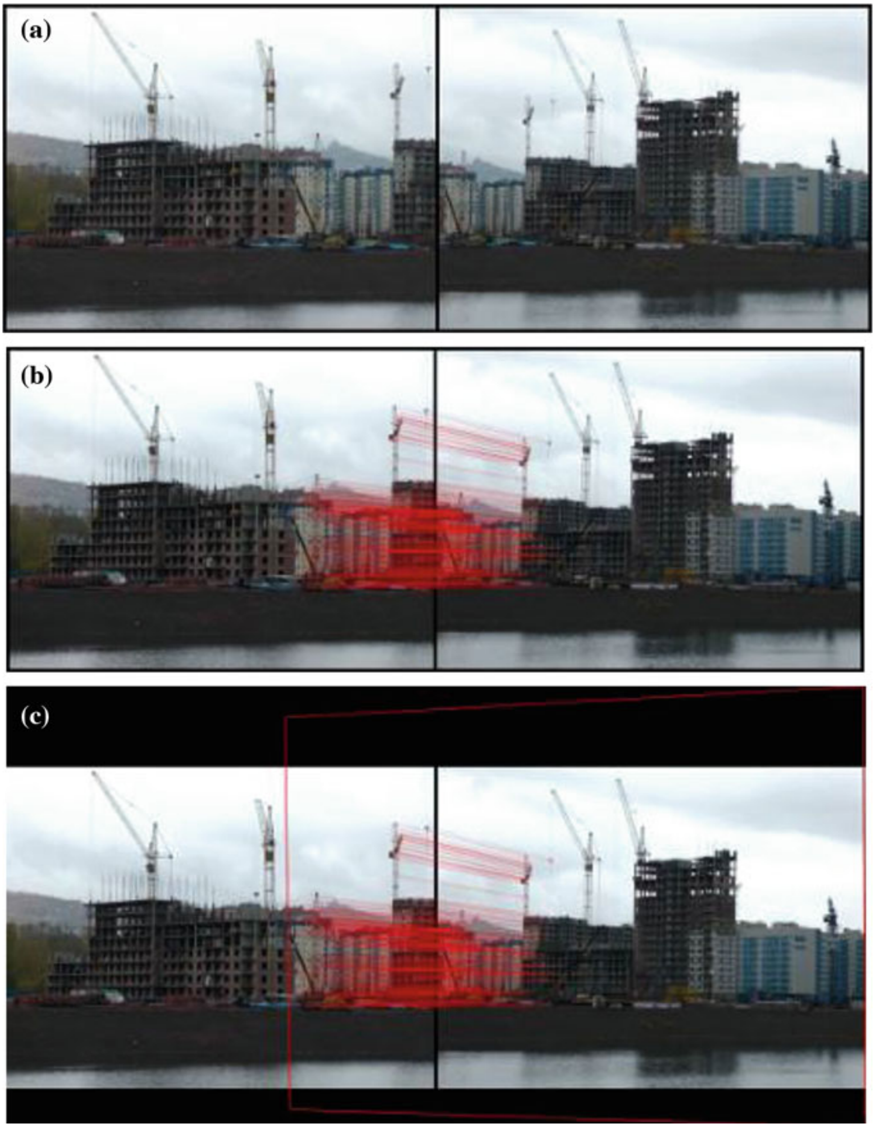


Fig. 4.7 The item "Houses": **a** input images, **b** feature points matching, **c** projection result, **d** final panoramic image, **e** good *top* fragment, **f** good *middle* fragment, **g** *bottom* fragment with an artifact stitching

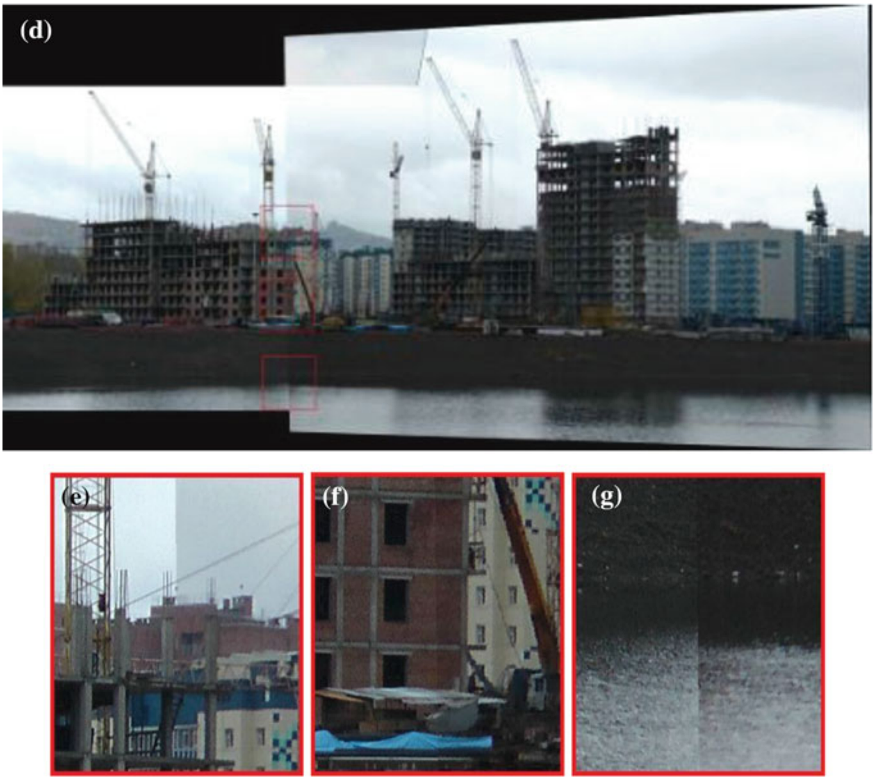


Fig. 4.7 (continued)

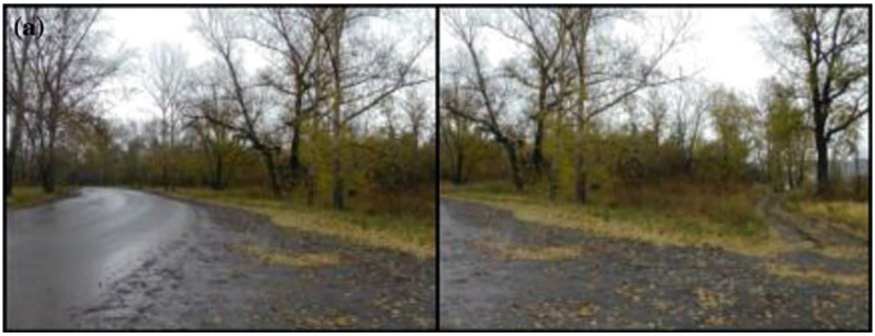


Fig. 4.8 The item “Road”: **a** input images, **b** feature points matching, **c** projection result, **d** final panoramic image, **e** good *top* fragment, **f** *middle* fragment with an artifact stitching, **g** *bottom* fragment with an artifact stitching

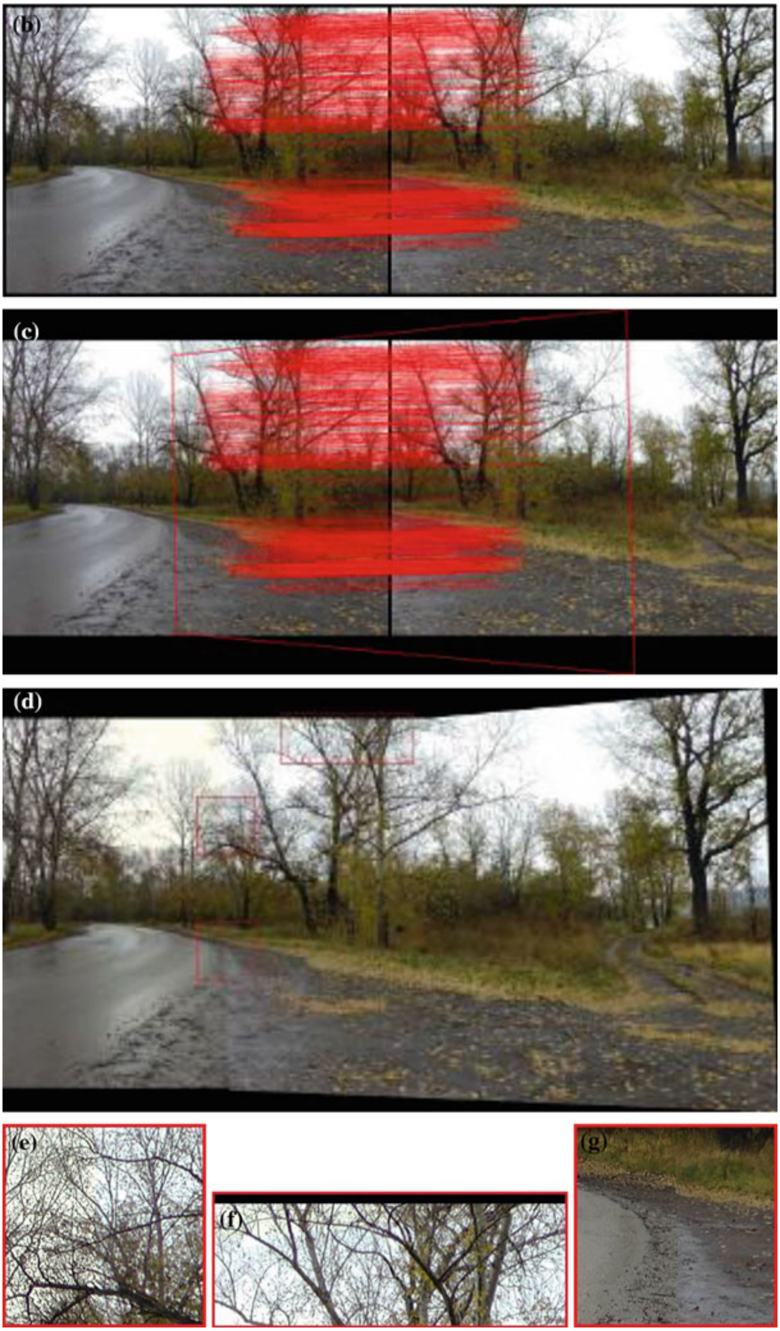


Fig. 4.8 (continued)

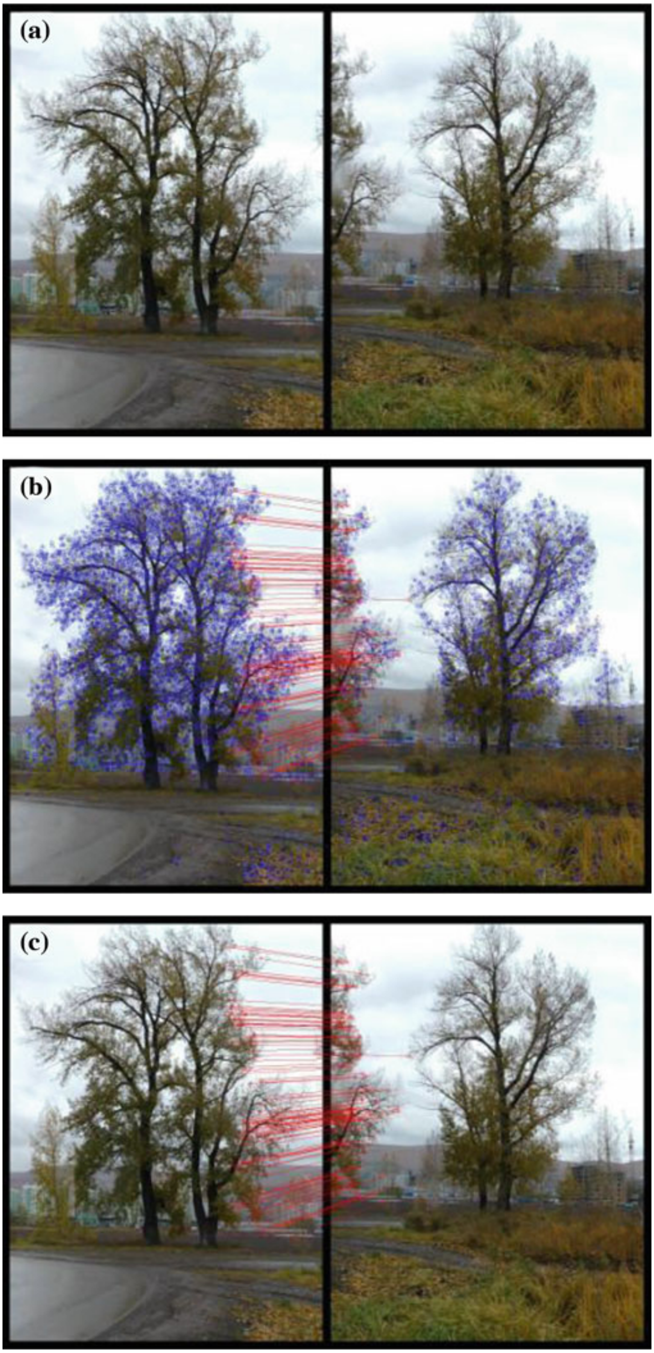


Fig. 4.9 The item “Trees”: **a** input images, **b** all detected feature points, **c** feature points matching, **d** projection result, **e** final panoramic image, **f** good *top* fragment, **g** good *middle* fragment, **h** good *bottom* fragment

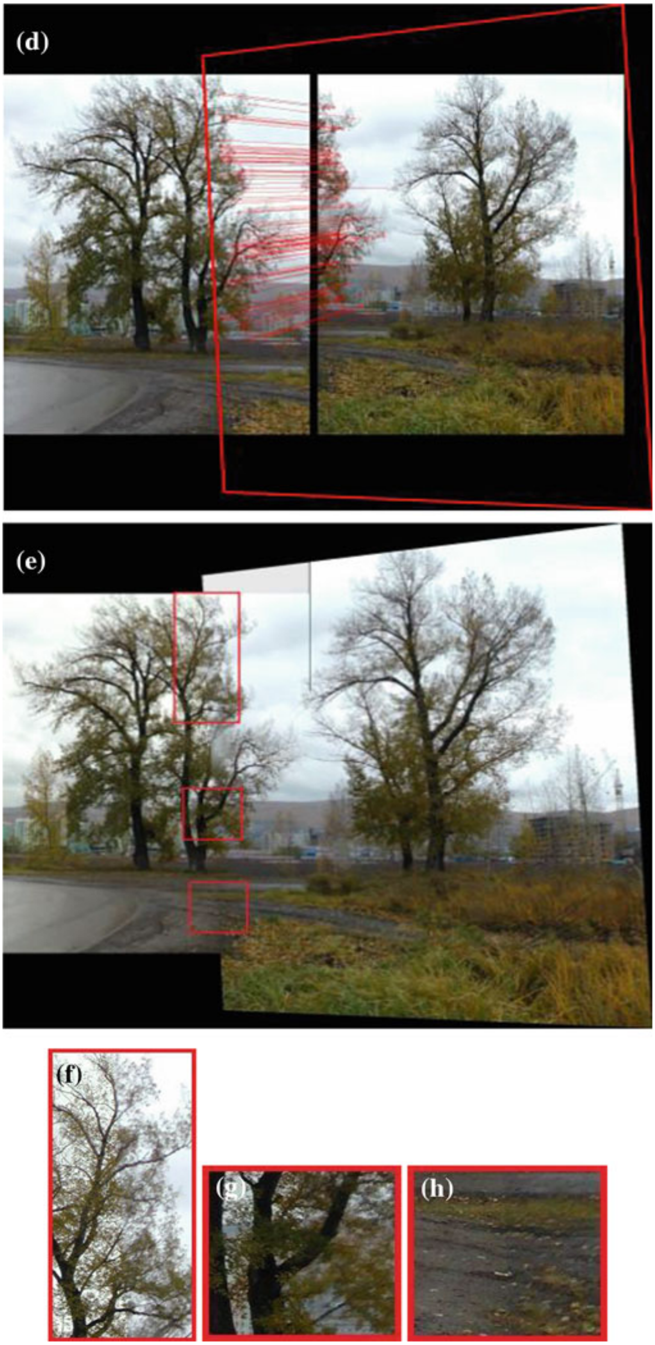


Fig. 4.9 (continued)

Table 4.1 The parameters of panoramic images creation

Items	Frame sizes, pixels	SUFR brightness threshold	Numbers of feature points correspondence			Time, s
			Left image	Right image	Common	Homography processing
Houses failure and Houses	640 × 480	500	592	830	72	0.784
	800 × 600	500	1,285	1,235	126	1.235
	1,024 × 768	500	1,828	1,747	162	2.141
		1,250	744	716	101	1.223
	1,280 × 960	500	2,631	2,643	241	3.350
		1,250	1,100	1,082	105	1.868
		1,500	859	875	87	1.679
Road	640 × 480	500	1,595	1,322	44	1.286
	800 × 600	500	2,636	2,626	279	2.686
	1,024 × 768	500	4,255	4,372	622	5.152
		1,250	2,438	2,459	284	2.649
	1,280 × 960	500	7,453	7,411	819	11.847
		1,250	4,322	4,175	575	5.360
		1,500	3,739	3,608	459	4.480
Trees	640 × 480	500	1,595	1,322	44	1.286
	800 × 600	500	2,527	2,069	62	2.232
	1,024 × 768	500	4,043	3,515	99	4.597
		1,250	2,603	1,828	65	2.436
	1,280 × 960	500	6,508	6,002	112	9.747
		1,250	4,134	3,168	60	4.704
		1,500	3,669	2,695	60	4.000

Table 4.2 Values of coefficients in homography matrixes

Parameter	Items			
	Houses failure	Houses	Road	Trees
h_1	1.12064947	1.12458137	0.77233217	1.09293649
h_2	0.01455373	0.02066196	0.01209813	0.04614185
h_3	1597.91839937	1666.59485125	1150.95532389	498.58782054
h_4	–0.04899116	–0.05201935	–0.09843960	–0.15573117
h_5	1.21681933	1.22309076	0.93694815	1.19738618
h_6	–280.89284415	–294.01541213	96.86158032	–52.85478987
h_7	–0.00004299	–0.00004421	–0.00009345	–0.00017982
h_8	0.00000109	0.00000252	–0.00000203	–0.00001127

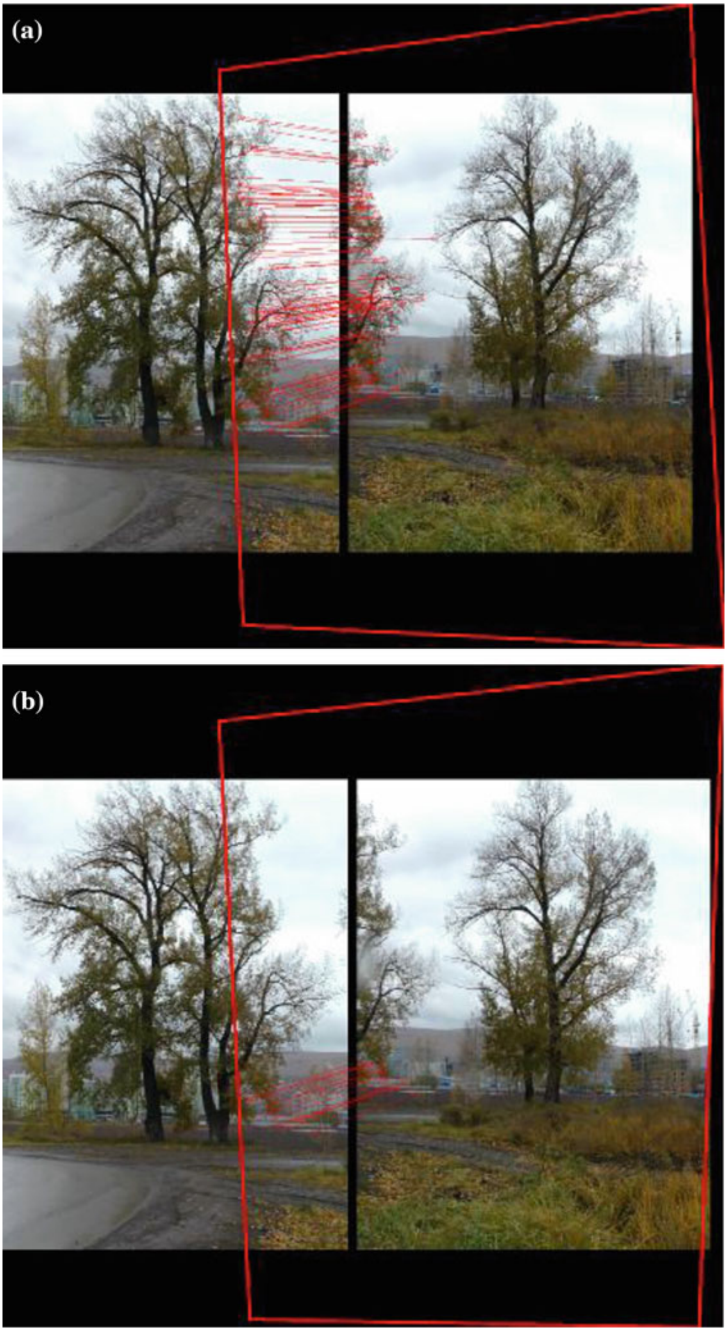


Fig. 4.10 The item “Trees” with various projective models: **a** total projection result, **b** *bottom* projection result, **c** *middle* projection result, **d** *top* projection result

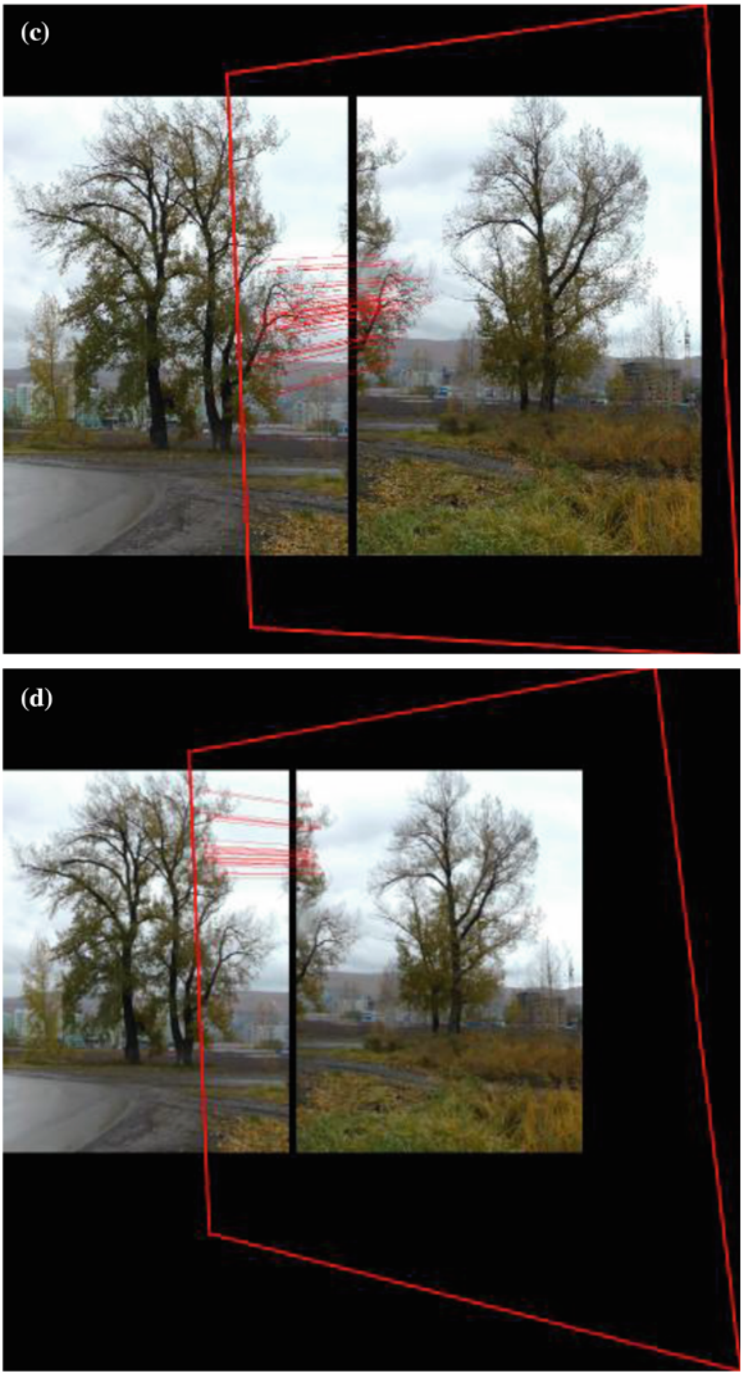


Fig. 4.10 (continued)

Table 4.3 Coefficients in homography matrix with various projective models for item “Trees”

Parameter	Region in overlapping area			
	Total	Bottom	Middle	Top
Total number of feature points	101	18	48	35
h_1	1.09293649	1.22312982	1.07997240	0.94064686
h_2	0.04614185	0.12969314	0.04410856	0.01186247
h_3	498.58782054	487.78575565	499.32996392	499.70470638
h_4	–0.15573117	–0.12683291	–0.16033099	–0.18758969
h_5	1.19738618	1.40435098	1.18993275	1.16906946
h_6	–52.85478987	–126.8548956	–50.06574907	–49.89502673
h_7	–0.00017982	–0.00012848	–0.00019690	–0.00039302
h_8	–0.00001127	0.0001145	–0.00001319	–0.00007330

4.8 Conclusion

In this chapter, some methods and algorithms were investigated for panorama construction from frames, which are received from multi-view cameras in the outdoor scenes. All steps of panorama construction were described with a special attention for the main issues—the geometrical and lighting alignment during the images stitching. Some novel procedures were proposed for selection “good” frames from video sequences, more accurate estimation of projective parameters in an overlapping area, and the lighting improvement of panoramic image by a point-based blending in the stitching area. The illustrations show well the visible intermediate results of images stitching. The calculated parameters of homography matrixes indicate on the necessity of following investigations in interpolation, morphing, and warping transformation for improvement of result panoramic images. Also it is important to accelerate an automatic panorama construction for practical implementation.

References

1. Briggs AJ, Detweiler C, Li Y, Mullen PC, Scharstein D (2006) Matching scale-space features in 1D panoramas. Comput Vis Image Underst 103(3):184–195
2. Dang TK, Worring M, Bui TD (2011) A semi-interactive panorama based 3D reconstruction framework for indoor scenes. Comput Vis Image Underst 115(11):1516–1524
3. Zhang W, Cham WK (2012) Reference-guided exposure fusion in dynamic scenes. J Vis Commun Image Represent 23(3):467–475
4. Chen H (2008) Focal length and registration correction for building panorama from photographs. Comput Vis Image Underst 112(2):225–230

5. Ni D, Chui YP, Qu Y, Yang X, Qin J, Wong TT, Ho SSH, Heng PA (2009) Reconstruction of volumetric ultrasound panorama based on improved 3D SIFT. Comput Med Imaging Graph 33(7):559–566
6. Powell I (1994) Panoramic lens. Appl Opt 33(31):7356–7361
7. Xiong Y, Turkowski K (1997) Creating image-based VR using a self-calibrating fisheye lens. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 237–243
8. Luong HQ, Goossens B, Philips W (2011) Joint photometric and geometric image registration in the total least square sense. Pattern Recogn Lett 32(15):2061–2067
9. Fan BJ, Du YK, Zhu LL, Tang YD (2011) A robust template tracking algorithm with weighted active drift correction. Pattern Recogn Lett 32(9):1317–1327
10. Zhao G, Lin L, Tang Y (2013) A new optimal seam finding method based on tensor analysis for automatic panorama construction. Pattern Recogn Lett 34(3):308–314
11. Beier T, Neely S (1992) Feature-based image metamorphosis. In: Procedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'92, vol 26, no 2. pp 35–42
12. Haenselmann T, Busse M, Kopf S, King T, Effelsberg W (2009) Multi perspective panoramic imaging. Image Vis Comput 27(4):391–401
13. Pazzi RW, Boukerche A, Feng J, Huang Y (2010) A novel image mosaicking technique for enlarging the field of view of images transmitted over wireless image sensor networks. J Mob Netw Appl 15(4):589–606
14. Jain DK, Saxena G, Singh VK (2012) Image mosaicking using corner techniques, Int Conf on Communication Systems and Network Technologies 79–84
15. Yang J, Wei L, Zhang Z, Tang H (2012) Image mosaic based on phase correlation and Harris operator. J Comput Inf Syst 8(6):2647–2655
16. Kwon OS, Ha YH (2010) Panoramic video using scale invariant feature transform with embedded color-Invariant values. IEEE Trans Consum Electron 56(2):792–798
17. Kim D, Hong KS (2008) Practical background estimation for mosaic blending with patch-based Markov random fields. Pattern Recogn 41(7):2145–2155
18. Bao P, Xu D (1999) Complex wavelet-based image mosaics using edge-preserving visual perception modeling. Comput Graph 23(3):309–321
19. Bhosle U, Roy SD, Chaudhuri S (2005) Multispectral panoramic mosaicing. Pattern Recogn Lett 26(4):471–482
20. Agarwala A, Zheng C, Pal C, Agrawala M, Cohen M, Curless B, Salesin D, Szeliski R (2005) Panoramic video textures. ACM Trans Graph 24(3):821–827
21. Brown M, Lowe DG (2007) Automatic panoramic image stitching using invariant features. Int J Comput Vis 74(1):59–73
22. Li C, Ma L (2009) A new framework for feature descriptor based on SIFT. Pattern Recogn Lett 30(5):544–557
23. Deng H, Zhang L, Ma J, Kang Z (2011) Interactive panoramic map-like views for 3D mountain navigation. Comput Geosci 37(11):1816–1824
24. Fiala M, Basu A (2002) Hough transform for feature detection in panoramic images. Pattern Recogn Lett 23(14):1863–1874
25. Kim DH, Yoon YI, Choi JS (2003) An efficient method to build panoramic image mosaics. Pattern Recogn Lett 24(14):2421–2429
26. Zhu Z, Xu G, Riseman EM, Hanson AR (2006) Fast construction of dynamic and multi-resolution 360° panoramas from video sequences. Image Vis Comput 24(1):13–26
27. Zhu Z, Riseman EM, Hanson AR (2004) Generalized parallel-perspective stereo mosaics from airborne videos. IEEE Trans Pattern Anal Mach Intell 26(2):226–237
28. Steedly D, Pal C, Szeliski R (2005) Efficiently registering video into panoramic mosaics. In: IEEE International Conference on Computer Vision (ICCV'2005), vol 2. pp 15–21
29. Langlotz T, Degendorfer C, Mulloni A, Schall G, Reitmayr G, Schmalstieg D (2011) Robust detection and tracking of annotations for outdoor augmented reality browsing. Comput Graph 35(4):831–840

30. Seitz SM, Dyer CR (1997) Viewing morphing: uniquely predicting scene appearance from basis images. DARPA Image Understanding Workshop, pp 881–887
31. Hernández-Mier Y, Blondel WCPM, Daula C, Wolf D, Guillemin F (2010) Fast construction of panoramic images for cystoscopic exploration. Comput Med Imaging Graph 34(7):579–592
32. Fathima AA, Karthik R, Vaidehi V (2013) Image stitching with combined moment invariants and SIFT features. Procedia Comput Sci 19:420–427
33. Fischler MA, Bolles RC (1881) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
34. Yan Q, Xu Y, Yang X, Nguyen T (2014) HEASK: Robust homography estimation based on appearance similarity and keypoint correspondences. Pattern Recognit 47(1):368–387
35. Torr PHS, Zisserman A (2000) MLESAC: a new robust estimator with application to estimating image geometry. Comput Vis Image Underst 78(1):138–156
36. Chum O, Matas J, Obdrzalek S (2004) Enhancing RANSAC by generalized model optimization. In: Asian Conference on Computer Vision, ACCV. pp 812–817
37. Nashat S, Abdullah A, Abdullah MZ (2012) Unimodal thresholding for Laplacian-based Canny-Deriche filter. Pattern Recogn Lett 33(10):1269–1286
38. Shen F, Zhao Y, Jiang X, Suwa M (2009) Recovering high dynamic range by multi-exposure retinex. J Vis Commun Image Represent 20(8):521–531
39. Dokládal P, Dokládalová E (2011) Computationally efficient, one-pass algorithm for morphological filters. J Vis Commun Image Represent 22(5):411–420
40. Fernandes LAF, Oliveira MM (2008) Real-time line detection through an improved Hough transform voting scheme. Pattern Recogn 41(1):299–314
41. Maxwell EA (1946) Methods of plane projective geometry based on the use of general homogeneous coordinates. Cambridge University Press, Cambridge
42. Maxwell EA (1951) General homogeneous coordinates in space of three dimensions. Cambridge University Press, Cambridge
43. Roberts LG (1965) Homogeneous matrix representations and manipulations of n-dimensional constructs, Technical Representation Document MS 1405, Lincoln Laboratory, MIT, Cambridge
44. Chum O, Matas J (2002) Randomized ransack with t(d,d) test. British Machine Vision Conference (BMVC'2002), vol 2. pp 448–457
45. Capel D (2005) An effective bail-out test for ransack consensus scoring. British Machine Vision Conference (BMVC'2005). pp 629–638
46. Matas J, Chum O (2005) Randomized RANSAC with sequential probability ratio test. In: 10th IEEE International Conference on Computer Vision, vol 2, pp 1727–1732
47. Chum O, Matas J (2008) Optimal randomized ransac. IEEE Trans Pattern Anal Image Underst 30(8):1472–1482
48. Raguram R, Frahm JM, Pollefeys M (2008) A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: 10th European Conference on Computer Vision, vol 2. pp 500–513
49. Cheng CM, Lai SH (2009) A consensus sampling technique for fast and robust model fitting. Pattern Recogn 42(7):1318–1329
50. Kiciak P (2011) Bicubic B-spline blending patches with optimized shape. Comput Aided Des 43(2):133–144
51. Kineri Y, Wang M, Lin H, Maekawa T (2012) B-spline surface fitting by iterative geometric interpolation/approximation algorithms. Comput Aided Des 44(7):697–708
52. Meylan L, Alleysson D, Süsstrunk S (2007) Model of retinal local adaptation for the tone mapping of color filter array images. J Opt Soc Am A: 24(9):2807–2816
53. Favorskaya M, Pakhirka A (2012) A way for color image enhancement under complex luminance conditions. In: Watanabe T, Watada J, Takahashi N, Howlett RJ, Jain LC (eds) Intelligent interactive multimedia: systems and services. Springer, Berlin
54. Zhang Y, Pajarola R (2007) Deferred blending: Image composition for single-pass point rendering. Comput and Graph 31(2):175–189

55. Sun J, Zhu H, Xu Z, Han C (2013) Poisson image fusion based on Markov random field fusion model. Inf Fusion 14(3):241–254
56. Mills A, Dudek G (2009) Image stitching with dynamic elements. Image Vis Comput 27 (10):1593–1602
57. Gómez F, Romero E (2011) Rotation invariant texture characterization using a curvelet based descriptor. Pattern Recogn Lett 32(16):2178–2186
58. Yang S, Wang M, Jiao L, Wua R, Wang Z (2010) Image fusion based on a new contourlet packet. Inf Fusion 11(2):78–84
59. Gracias N, Mahoor M, Negahdaripour S, Gleason A (2009) Fast image blending using watersheds and graph cuts. Image Vis Comput 27(5):597–607

Chapter 5
A New Real-Time Method of Contextual Image Description and Its Application in Robot Navigation and Intelligent Control

Konstantin I. Kiy

Abstract Computer vision and image understanding are of crucial importance in robotic systems of the future. In this chapter, a new real-time method of contextual image description is presented, and its application to image understanding problems, arising in robotics, is given and discussed. A new vector form of contextual description and segmentation of color images is proposed. This form, called STructural Graph (STG) of color bunches, conserves the geometric constraints in the image and projects real objects onto certain formal objects in the contextual vector description, which can be axiomatically (mathematically) described and found by the real-time segmentation algorithms. The relation between formal objects in the STG and real objects in the image is established. The concept of local and global contrast objects in the STG is put forward. Real-time algorithms for segmentation and detection of global contrast (salient) objects in the STG are described, which provide the stable segmentation components for solving scene recognition problems. These algorithms are based on the geometrized histogram method developed by the author. Selection of stable segmentation components in images is applied to the analysis of video sequences in order to find and recognize visual landmarks. Applications of the developed technique to autonomous robot navigation are presented and discussed.

Keywords Computer vision · Segmentation · Robot navigation · Image description

K.I. Kiy (✉)
Keldysh Institute of Applied Mathematics, 4 Miusskaya pl, Moscow 125047,
Russian Federation
e-mail: konst.i.kiy@gmail.com

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_5

5.1 Introduction

Despite a significant progress that has been made during the last two decades in visual robot navigation based mainly on salient feature points such as Harris corner points, Scale-Invariant Feature Transform (SIFT), etc. [1], there is a need for a new technique that has to be useful in fast motion with changing direction, overcrowded environment, and in cluttered background. This technique has to be able to cope with detection and tracking of boundary curves of the main objects in the image and the objects themselves. In fast motion (e.g. in sport games), the human vision provides the navigation of a sportsman based on a small number of features, such as critical objects, perceived as contrast (frequently colored) blobs. The idea to provide such a description in a real-time computer vision is the backbone of the proposed method. Methods based on a finite number of salient points are unstable under occlusion. For example, if one tries to find a geometric figure in the image based only on the positions of its vertices, then it may not be found especially, when some of these vertices are occluded. If this figure is located against the background of an object generating many salient points (e.g., bookshelves), then it is also difficult to find the figure. However, a human solves these problems without any difficulty. Since humans take into account many additional characteristics of the desired object, the problem is solved by them in a more general statement. For example, color and intensity of the region bounded by the desired figure provide the necessary information to detect the partially occluded figures. This means that region-based segmentation in combination with contour data can provide very important information in order to solve the problem at hand under more difficult conditions.

As a rule, the salient points make it possible to find reliably a simple object against a simple background. If region-based information is appended, then there is a chance to find reliably simple (and even rather complex) objects with occlusion against a complex background. If not the whole object is occluded and the body of the object with the hole caused by occlusion is employed in recognition, it can be better recognized than in the case, when one leans on the corner points only. One can also use the color information about the body of the object for more confident identification. In many publications, it was mentioned that the lack of real-time techniques of stable image segmentation limits the possibility of region-based methods in applications [2, 3]. The capabilities of mobile robots to understand scenes and to solve the localization (categorization) problems suffer from this lack as well [2]. This chapter describes a version of the required technique and its application in robot navigation. In this chapter, the indoor applications are mainly interested. Applications in outdoor environment will be described in detail in publications to follow. For example, the technique for finding objects by color contrasts and for controlling a robot following towards colored visual landmarks is developed.

The proposed approach is based on a new type of features attached to color images. Instead of simple functions of features, more complex objects, consisting of the functions of features furnished with structures describing the geometry of values distribution of these functions are employed. Using the proposed method for describing the geometry of value distribution, instead of classical histograms of functions, new features, called geometrized histograms, were introduced. The geometrized histograms of images were proposed in [4]. The first ideas of the geometrized histogram were presented in [5]. This method was successfully applied in navigation of an autonomous vehicle on bumpy dirt roads [6]. An early version of a geometrized histogram of a color image was presented in [7], where possible applications to the analysis of road scenes have been discussed.

The rest of the chapter is organized as follows. Related works are discussed in Sect. 5.2. Section 5.3 describes the method of geometrized histograms and introduces the concept of "contrast objects". In this section, the structural graph STG of a color image, and virtual contrast boundary curves, as well as "germs" of global objects in it, are defined. In Sect. 5.4, the bipartite graph of contrast boundary curves and germs of global contrast objects Left-Right Germs (LRGs) in the STG is proposed. The concepts of left-right germs of the global contrast objects are introduced formally in Sect. 5.4. Informally, they mean left (or right) parts of the boundaries of global objects, furnished with descriptions of the color characteristics of the objects near the boundary parts. Using the LRG, the informal concept of "global contrast objects" is interpreted formally in the STG as the connected components in the LRG. Based on the technique of the LRG, the algorithms for constructing global objects in the STG are also proposed. Section 5.5 presents the experimental software package implementing the proposed algorithms and some experimental results. In this section, the application of the developed technique to finding artificial visual landmarks (labels) is presented, and the results of experiments with a robot moving towards such labels are shown. Future work is also discussed. Conclusions are given in Sect. 5.6.

5.2 Related Works and Main Ideas

The region-based segmentation methods allow us to obtain important information about objects in the image. Information about regions bounded by contours provides the necessary data for motion tracking of contours in image sequences and matching contours in stereo pairs. However, the fact that the segmentation results are unstable under slight variations of lighting conditions and changing the point of view, emphasized in [2, 3], makes researchers to introduce the additional structures or to add a supervisor, labeling regions of interest, in order to solve practical tasks, arising in image analysis [2, 3]. It was mentioned in [3] that the existing segmentation algorithms do not always produce perceptually meaningful regions. This is the main reason for segmentation results to be very sensible to slow variations of viewing conditions. The goal of the proposed approach is to develop a technique providing the opportunity of using semantic information at early stages of the segmentation process in order to produce perceptually meaningful regions.

In any image of unknown nature, the hints or the cues are able to start the scene categorization process (for example, regions of interest, labeled by a supervisor or by some device like a laser range finder). As such cues, humans employ clearly visible objects with contrasting regions of their bodies. They demonstrate this ability even in fast motion (in sport games, fast driving, etc.), when it is impossible to see such small details as salient points. This does not depreciate the meaning of such important features in computer vision; they have to be perceived at the second glance. To follow these ideas, a new conceive image description is provided in the form of the structural graph of color bunches, its connection with the image itself is explained, and contrast elements in the STG are mathematically described. Then, the formal global contrast description of objects in the STG is created and algorithms for finding them are designed. Global contrast objects provide the stable segmentation components. They are the support points in the proposed image analysis and can be find automatically without any supervisor and under variation of viewing conditions. These formal global objects have both geometric and color descriptions, which are very brief as compared with the description of pixel objects in the image corresponding to them. However, global contrast objects in the STG have rather rich description, which makes it possible to find objects in the real image specified by certain shape and color characteristics. Using this technique, it is possible to solve recognition problems in real time. The next two sections are devoted to a presentation of these ideas.

5.3 Geometrized Histograms of Color Images and Segmentation

Both the region-based and contour-based segmentation methods are very important. The goal of this chapter is to produce a combined geometrical and statistical description of the image in order to join both segmentation approaches. To follow this idea, in the method of geometrized histograms, the information about the frequencies of values of the function specifying the image (either scalar functions, as in grayscale and InfraRed (IR) images or vector functions, as in color images) is combined with a certain geometrical description of its level sets. The method is based on the idea that the values of the intensity function are close not only in the case, when they are close (in some sense) as integers, but also when the sets, on which these values are taken, are close as subsets in the image plane. This assumption results in the fact that instead of an n-dimensional feature space of the image, a certain bundle of fibers is obtained. These fibers are the structures of segments in the straight line of one of the orthogonal axes of the image plane (vertical or horizontal), furnished with arrays of color and frequency characteristics. In topology, such bundles of fibers are called the fiber bundles. Let us follow here one of the main ideas of modern mathematics—to consider a complex object as a fiber bundle with a simple fiber.

If an image has a discrete one-dimensional representation, then any level set of any feature function is described by several segments, on which this function takes a particular value. A similar approximate description for level sets of a function defined on a narrow strip can be obtained, if the level sets are projected onto the lower side of this strip. In the discrete case, the projection of a set of pixels with a fixed value of the feature function onto the axis Os, corresponding to the lower side of the strip, consists of several connected segments of pixels in the axis. The only difference is that for a function on a straight line, the segments describing the level sets with different values of the function do not intersect each other. However, the corresponding segments of the feature function defined on the strip, which obtained by the projection onto the side of the strip, may intersect each other. Figure 5.1 illustrates this construction. In Fig. 5.1, two systems of segments for a function taking two values in the strip $\mathbf{St}_i$ are constructed. The level sets of this function in the strip consist of two sets of rectangles painted in different ways (the central part of the figure). After the projection, each system consists of two segments that are depicted separately on two examples of the axis Os for clearness (lower part). Actually on the axis Os of the strips side, they overlap each other. One can furnish each of the segments by the corresponding value of the feature function (1 or 2) and the cardinality numbers. These numbers are equal to the numbers of points in the corresponding level set projected onto the segment. Let us describe the construction more formally.

Let $\mathbf{L}_z$ be the set of points of the strip $\mathbf{St}_i$, at which a certain feature function $f(x, y)$ takes the value z. Then the projection of $\mathbf{L}_z$ onto the axis Os is a union of intervals (segments) $\mathbf{I}_{kz}$ on this axis $Pr(\mathbf{L}_z) = \cup_k \mathbf{I}_{kz}$. For each interval $\mathbf{I}_{kz}$, its cardinality obtained in this projection is calculated, which is equal to the number of points of the level set $\mathbf{L}_z$ in the image strip $\mathbf{St}_i$ that are projected onto this interval.

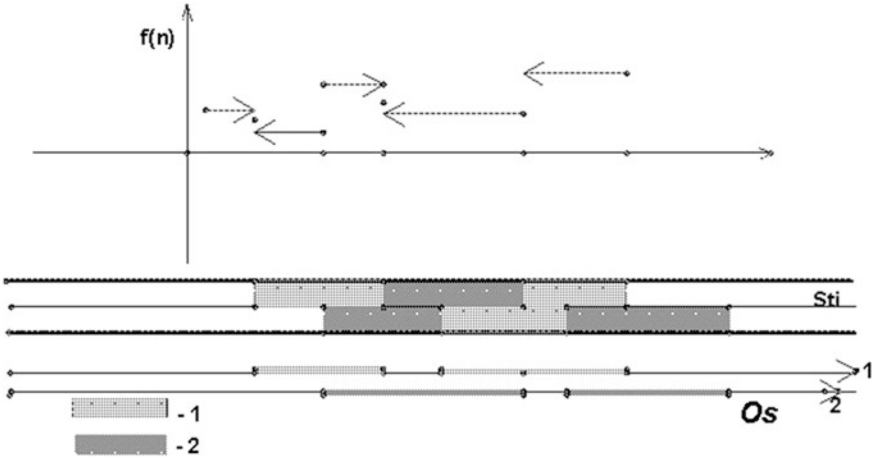


Fig. 5.1 Construction of approximate descriptions of level sets in a narrow strip; the *lower two lines* show the definition of the joint description

It is clear that the collection of cardinalities of intervals $\mathbf{I}_{kz}$ for all possible values z determines the classical histogram of the feature $f(x, y)$ in the strip $\mathbf{St}_i$. The collection of intervals $\mathbf{I}_{kz}$ describes approximately the region in the strip, where $f(x, y)$ takes the value z because $\mathbf{L}_z \subset Pr^{-1}(\cup_k \mathbf{I}_{kz})$, and the strip is narrow. The union of all intervals for all values of $f(x, y)$ within a selected strip defines a space of intervals in Os with a scalar function defined on them (cardinality). This space is called the "local geometrized histogram" ($\mathbf{HG}_i$) of the function $f(x, y)$ in the strip $\mathbf{St}_i$. Let the image $\mathbf{Im} = \cup_i \mathbf{St}_i$ be a union of nonintersecting narrow strips parallel to the axis Os, and let $\mathbf{Bs} = \{1, 2, \ldots, n\}$ be a finite ordered set numbering the strips $\mathbf{St}_i$, $i \in \mathbf{Bs}$. Then, as in the mathematical theory of fiber bundles, the global geometrized histogram of the function $f(x, y)$, corresponding to a given partition of the image into strips, is the triple (π, $\mathbf{HG}$, $\mathbf{Bs}$), where $\mathbf{HG} = \cup_i \mathbf{HG}_i$ is the union of all geometrized histograms of strips $\mathbf{St}_i$, and π is the projection mapping onto the base $\mathbf{Bs}$, which attaches to each interval of the geometrized histogram $\mathbf{HG}_i$ the serial number of its strip π: $\mathbf{HG} \rightarrow \mathbf{Bs}$. The global geometrized histogram describes approximately the geometry of value distribution of the function $f(x, y)$ in the image. The detailed discussion of the introduced geometrized histogram and examples of geometrized histograms of particular images can be found in [4].

The geometrized histogram of a color image and the set of color bunches are presented in Sect. 5.3.1. Section 5.3.2 provides the preliminary local segmentation in strips. The partial order relation and contrasts are described in Sect. 5.3.3. The structural graph of color bunches and continuous left and right contrast curves are proposed in Sect. 5.3.4.

5.3.1 The Geometrized Histogram of a Color Image and the Set of Color Bunches

Let us discuss briefly the construction of the geometrized histogram of a color image. The detailed description can be found in [4]. In this construction, two coordinate systems of color ratios and of overall grayscale intensity introduced in [4] are used. These two coordinate systems are described by the functions of pixels ($G/(G + B)$, $G/(G + R)$, I) and ($G/(G + B)$, $R/(R + G)$, I) and are employed in order to represent the color of a pixel of the given color image **CI** in a form suitable for further considerations. Here (R, G, B) are standard color coordinates and I is the grayscale intensity of the pixel. It can easily be seen that pairs of functions ($G/(G + B)$, $G/(G + R)$) and ($G/(G + B)$, $R/(R + G)$) determine two coordinate systems on the standard color triangle given by the equation $R + G + B = 1$ (except for the singular points of the ratios) [4].

Figure 5.2 presents the standard color triangle and explains the meaning of the color ratios introduced. This figure illustrates the fact that the ratios $G/(G + B)$, $G/(G + R)$, and $R/(R + G)$ are constant on segments originating at the vertices **R**, **B**, and **G** of the color triangle, respectively, and passing to the opposite sides of the

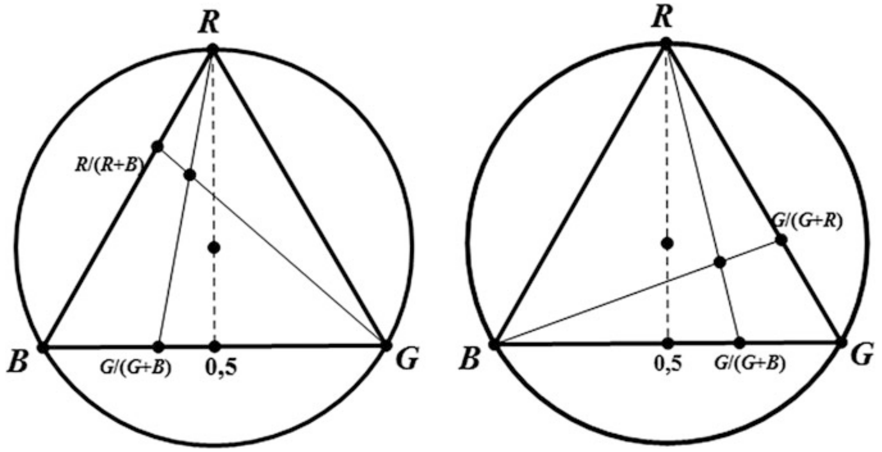


Fig. 5.2 *Color triangle* with segments of constancy of *color ratios*

color triangle. The value of the corresponding ratio is the coordinate of the other end point of the segment on the sides **BG**, **GR**, and **RG** of the color triangle, respectively. For example, if $G/(G + B) = 0.5$, then the second end of the segment is at the center of the side **BG**. It is clear from the figure that the shown pairs of segments (color ratios) determine uniquely points inside the color triangle. It is possible to express the (H, S) coordinates of the color triangle in terms of color ratios [4]. If the following notation: $(G/(G + B), G/(G + R)) = (C_{23}, C_{21})$ is introduced, then the formula for the saturation S will take the form: $S = (C_{21} + C_{23} - 4C_{23}\, C_{21}))/(C_{21} + C_{21}(1 - C_{21}))$. Let us introduce the "linear" hue, which is uniformly parameterized by sides of the color triangle, starting from **R**. For example, on the side **RG** the linear hue coincides with the function $H_1 = G/(R + G{+}B)$.

The formula for calculating the linear hue in terms of the color ratios is as follows: $H_1 = (C_{23}\,(2C_{21} - 1))/(C_{21} - 2C_{23}\,(1 - C_{21}))$. It is known that the real hue H is uniformly parameterized by the length of the arc of the circle, into which the color triangle is inscribed. Suppose that H_1 has K grades. Using a special array of numbers $Hue[i]$, $i = 0, \ldots, K - 1$, calculated in advance, the "real hue" H can find in terms of H_1, $H = Hue[H_1]$. This method of calculating reduces the computational cost.

In these constructions, the coordinate systems $G/(G + B)$, $G/(G + R)$), and $(G/(G + B), R/(R + G))$ are used in the left and right half-triangles, respectively, as it is shown in Fig. 5.2. The color ratios determine the balance of color components R, G, B at each pixel of the image. Instead of considering three geometrized histograms of color coordinates, only two geometrized histograms can be consider: a single geometrized histogram that joins the color information of the image and the geometrized histogram of the grayscale intensity I. The color information can be joined in a geometrized histogram of a certain characteristic function CF, constructed based on $G/(G + B)$ as the geometrized histogram of a one-dimensional feature function *CF* in the way described above. To save the complete color information, the additional structures are used that reflect the value of the opponent ratio function (either *G/*(*G* + *R*) or *R/*(*R* + *G*)). These opponent ratio functions are the second coordinates of the coordinate systems in both rectangular triangles. In numerous experiments with real-world images, it was shown that the *G/*(*G* + *B*) ratio has the most efficient power for discrimination among other ratios and color coordinates. Let us explain, how to modify the *G/*(*G* + *B*) feature to improve its discriminating ability. Since for a fixed value of *G/*(*G* + *B*) different values of *G/*(*G* + *R*) (or *R/*(*R* + *G*)) give different values of hue and saturation of the pixel, the intervals of the geometrized histogram of *G/*(*G* + *B*) corresponding to its constant values may be inhomogeneous in color (in the geometrized histogram of this one-dimensional feature). For example, adjacent regions in the image with different color ranges that have the same values of *G/*(*G* + *B*) but different values of the other ratio determine under the projection onto the axis Os a single interval and are not discriminated by this geometrized histogram. To provide the color homogeneity of such intervals, this feature should be modified taking into account the values of the other ratio. Using the values of the second ratio, additional values of the main feature *G/*(*G* + *B*) are introduced artificially. For example, if the hue of the pixel belongs to the yellow range, then the characteristic function *CF* coincides with *G/*(*G* + *B*). If the pixel hue belongs to the red (green) range, then the value of *CF* differs from *G/*(*G* + *B*) by *M* (2*M*), where *M* is the number of grades of the feature *G/*(*G* + *B*). In a similar way, the red, violet, and blue ranges are separated. In a way, this procedure is analogous to and inspired by the lift to the universal covering in topology (the hue becomes a continuous function, when the lift of the hue is considered to the universal covering of the circle, which is the helical curve). This procedure increases the discriminating ability of *G/*(*G* + *B*). It also allows avoiding erroneous segmentation results demonstrated by many segmentation algorithms for regions with gradually changing colors.

Further modifications of *CF* are introduced in order to improve the distinctive abilities of the geometrized histogram and to take into account the fact that in dark objects color characteristics may have discontinuous behavior. It is generally known that humans perceive dark, low-saturated objects as purely dark (without any color). Experiments with many images have shown that color insensitivity has different thresholds for intensity and saturation for different color ranges. Based on these experiments, a number of thresholds for intensity and saturation were introduced for several selected color ranges (red, orange, yellow, green, etc.). Also the introduction of several additional grades of *CF* was proposed. These grades are assigned to colorless pixels in the corresponding color range, whose intensity and saturation are smaller than the thresholds introduced. In the same way, when a set of pixels is rather bright, humans do not see the color of this set and perceive it as colorless. As for dark pixels, one can introduce thresholds of intensity and saturation, and add several grades of *CF*, to select bright colorless pixels. Thus, the value of *CF* at a pixel is *G/*(*G* + *B*) or *G/*(*G* + *B*) shifted by *M* (2*M*) or the grade of colorlessness (in the dark or bright range) of the pixel in the corresponding color range.

A one-dimensional geometrized histogram for *CF* was constructed in the sense explained above. To conserve complete color information for each segment of the geometrized histogram, the conventional histogram of deviations from one half of the opponent color ratios (either *G/*(*G* + *R*) or *R/*(*R* + *G*)) was collected. To complete color information encapsulated in the each segment of the color geometrized histogram in the course of its construction, the mean value and lower and upper bounds of the values of the grayscale intensity of the pixels projected onto this particular segment were found. In this way, the information contained in the intensity *I* is added to the geometrized histogram of the color image. Using this information, some parameters are prescribed to each segment **Sg** of the geometrized histogram in the color image:

- The interval [beg_{Sg}, end_{Sg}] of the beginning and end points of the colored segment **Sg** on the axis Os.
- The range $\Delta_H^{Sg} = \left[H_{min}^{Sg}, H_{max}^{Sg}\right]$ and the mean value H_{mean}^{Sg} of the hue.
- The range $\Delta_S^{Sg} = \left[S_{min}^{Sg}, S_{max}^{Sg}\right]$ and the mean value S_{mean}^{Sg} of the saturation.
- The range $\Delta_I^{Sg} = \left[I_{min}^{Sg}, I_{max}^{Sg}\right]$ and the mean value of the grayscale intensity component I_{mean}^{Sg}.
- The cardinality of the segment $Card^{Sg}$ (approximately the number of points in the strip with the specified color characteristics that are projected onto the colored segment **Sg** on the axis Os).

Denote by $dens(Sg) = Card^{Sg}/(end_{Sg} - end_{Sg} + 1)$ the density of **Sg**.

Since in real-world images color characteristics of pixels vary in a wide range, the geometrized histogram of a color image contains many segments that have similar color characteristics and strong intersection. The number of colored segments in the geometrized histogram is too big to solve qualitative recognition problems arising in practical application. However, numerous examples of real-world images have shown that the geometrized histogram has significant distinctive abilities. As a rule, the clusters of segments with close color characteristics correspond to real objects in strips. In the next subsection, the algorithms for obtaining clusters of segments of the geometrized histogram will be presented in order to generate color bunches (clusters of segments).

5.3.2 Preliminary Local Segmentation in Strips

After introducing the classifying space, let us describe the mathematical procedures for finding objects in the image. To restore the complete color information, the second coordinate in the coordinate systems of color ratios introduced on the color triangle (*G/*(*G* + *R*) or *R/*(*R* + *G*)) is employed. In each strip, using the classical histogram of deviations of *G/*(*G* + *R*) (or *R/*(*R* + *G*)) from 0.5, the ranges of variation of hue and saturation, as well as their mean values, are attached to each colored segment of the geometrized histogram of CF. Let us explain, how the description for each colored segment of the geometrized histogram in the Hue, Saturation, grayscale Intensity coordinate system (H, S, I) is obtained. The range of variation of I and the mean grayscale value in each of the intervals were obtained in the course of finding the segment as the projection of points of the strip with the fixed value of the characteristic function CF. Let us pass to finding the ranges of variation of H and S and their mean values for each segment of the geometrized histogram. Assume that the second ratio ($G/(G + R)$ or $R/(R + G)$, depending on the considered coordinate system) is measured in grades and takes values from 0 to 64. When the pixels that are projected onto a particular segment of the geometrized histogram are collected, the deviations of the second color ratio at these pixels from the mean value (which is equal to 32) are calculated. For the whole segment, the histogram of deviations of the second color ratio from the mean value is computed $Hist(n)$, where $n = 0, \pm1, \pm2, \ldots$ is the value of deviations. Here $Hist(n)$ specifies the fraction (portion) of pixels projected onto the segment at hand that have the deviation from 32 (0.5) equal to n. Each deviation n determines uniquely the value of the hue and saturation H_n and S_n of the corresponding pixels.

Since for each pixel of the segment the value of the first color ratio ($G/(G + B)$) is fixed and the both color ratios determine uniquely the point of the color triangle, H_n and S_n values can be found. (The value of the second color ratio is $32 + n$.) Using the frequencies $Hist(n)$, the H_{mean} and S_{mean} values corresponding to the segment can be found. To specify the ranges of variation of H and S for the segment, the global maximum and the extreme left and right "significant" local maximums of $Hist(n)$ can also be detected. The extreme left and right significant local maximums determine the ranges of variation of H and S, defined by H_{min}, S_{min}, and H_{max}, S_{max} corresponding to the left and right maximums of $Hist(n)$, respectively. One can determine the boundary points of the ranges of H and S variation by using the extreme maximums in order to avoid outliers of values of the features H and S for a particular segment. The global maximum gives additional features H_{freq} and S_{freq}, which specify the values of H and S that are most frequently occurred. This allows us to attach to each segment of the geometrized histogram the ranges of variation and mean values of hue and saturation. In this way, the color range of the segment in the coordinate system (H, S) is described. The transition to this coordinate system is explained by the fact that it is convenient and necessary in the clustering procedures described in what follows.

Remark. Note that the calculations are considerably reduced by dealing with deviations of the other ratio from 0.5 (mean grade 32) at the level of pixels and by the transition to the (H, S, I) coordinate system at the level of segments of the geometrized histogram.

At the first stage of constructing local objects within each strip of the image, using the original clustering methods developed, the homogeneous bunches (clusters) of colored segments in the geometrized histogram of this strip are detected, which determine certain local objects within the strip. To select seeds for clusters, the following "survival" procedure is used.

Survival Procedure. Consider an array *Sur* of dimension of either *dimX* or *dimY* equal to the corresponding dimension of the image (horizontal or vertical), depending on what strips vertical or horizontal are employed. In the case, when the survival procedure is applied separately in several color ranges, the number of arrays Sur_i introduced for each color range is equal to the number of these ranges. The segments of the geometrized histogram are mapped on the array *Sur* (Sur_i), and the segment with the highest density (in the sense of *dens*(*Sg*) introduced above) survives at each point of the array *Sur* (Sur_i). The serial number of the survivor in the geometrized histogram is written in each entry of the array *Sur* (Sur_i). Then for each segment, the fraction (portion) of its points is calculated, where this segment is a survivor. This portion determines the visibility of the segment (or the visibility of the segment in a particular color range). The segment that dominates the other segments at all its points has the maximal visibility. The set of visible segments is divided into several groups depending on their visibilities. Starting from the group of most visible segments, the segments from visibility groups are taken successively as seeds of clusters in the order of decreasing the length of segments within each group. Next the grouping procedure is applied using the selected seeds.

In the grouping procedure, the segments of the geometrized histogram with a sufficiently big intersections and similar color characteristics are grouped. These segments correspond to sets of pixels in the strip that are projected onto connected segments. Any two of them have strong intersection in the sense of the following pseudo-metrics provided by Eq. 5.1 for intervals (segments) **I** and **J** on the axis Os with lengths *L*(**I**) and *L*(**J**).

$$\begin{aligned} &d_i(\mathbf{I},\mathbf{J}) = 1 - \rho_i(\mathbf{I},\mathbf{J}) \;\; i = 1,\, 2,\, \ldots \;\; \rho_1(\mathbf{I},\mathbf{J}) = L(\mathbf{I}\cap\mathbf{J})/\min(L(\mathbf{I}), L(\mathbf{J})) \\ &\rho_2(\mathbf{I},\mathbf{J}) = L(\mathbf{I}\cap\mathbf{J})/\max(L(\mathbf{I}), L(\mathbf{J})) \end{aligned} \tag{5.1}$$

Grouping procedure. Each segment of the geometrized histogram **Sg** is characterized by the following features:

- The beginning and end in the axis Os $\mathbf{Int}^{Sg}$ = [*beg*, *end*], determining the part of the strip, where the points projected onto this segment are located.
- The range of hue value [H_{min}, H_{max}] and the mean hue value H_{mean}.
- The range of saturation value [S_{min}, S_{max}] and the mean saturation value S_{mean}.
- The range of grayscale intensity value [I_{min}, I_{max}] and its mean value I_{mean}.
- The cardinality of the segment $Card^{Sg}$ equal to the number of points of the strip projected onto the segment.

In the grouping procedure, the segments are added to the chosen seed, if they are close in color characteristics and to the seed segment according to proximity measures (Eq. 5.1). There is a system of rules that with due account of the listed features of the seed segment $\mathbf{Sg}_0$ and of the investigated segment $\mathbf{Sg}_1$ decides, whether or not to add the latter to the cluster determined by the seed. Different rules for determining the proximity in color characteristics are used, depending on the comparative lengths of the intervals $\mathbf{Int}_0$ and $\mathbf{Int}_1$, their proximity determined by Eq. 5.1, and their densities. As a result, several clusters will be obtained.

At the next stages, the survival procedure is applied for the remaining segments. Such procedure is performed several times, each time for a specified hue range. This allows finding even small colored objects against different backgrounds. After survival procedures for different hue ranges, the grouping procedures are applied again. The obtained clusters of colored segments are called color bunches. Informally, the color bunches are generated by colored segments that have similar color characteristics and "large" intersections in accordance with the pseudo-distances in Eq. 5.1. As a union of colored segments, each of these bunches is characterized by definite mean values of hue, saturation and grayscale intensity, as well as by the ranges of variation of these parameters. There is a system of rules that determines the color characteristics of the bunch based on color characteristics of the colored segments generating this bunch. In addition to these numerical characteristics, each color bunch has a geometric description, an interval on Os, which is determined by the corresponding intervals of its members-segments. Based on each color bunch, using its numerical characteristics as the data for local thresholding, a certain set of points in the strip can be recovered. In this way, "vertical" segmentation in the strip is performed. As a result of the vertical segmentation in the strip, the integrated local objects, called color bunches, are obtained. Each color bunch is a junction of several segments of the geometrized histogram of the color image. A color bunch *b* of the geometrized histogram of the color image is characterized by the following parameters mentioned below:

- The interval $[beg_b, end_b]$ on the axis Os.
- The range $\Delta_{\mathrm{H}}^{b} = \left[H_{min}^{b}, H_{max}^{b}\right]$ and the mean value H_{mean}^{b} of the hue.
- The range $\Delta_{\mathrm{S}}^{b} = \left[S_{min}^{b}, S_{max}^{b}\right]$ and the mean value S_{mean}^{b} of the saturation.
- The range $\Delta_{\mathrm{I}}^{b} = \left[I_{min}^{b}, I_{max}^{b}\right]$ and the mean value I_{mean}^{b} of the intensity.

Each color bunch has the cardinality $Card^b$ equal to the sum of cardinalities of segments of the geometrized histogram united in this bunch. The set of color bunches *Bun*(**CI**) or simply *Bun* contains the most important color and geometric information about the image. This becomes clear, if one considers the rectangle corresponding to the investigated image (with the same dimensions), divides the rectangle into the strips in the same way, and superimposes the obtained color bunches on the middle lines of the strips. Many examples of such pictures can be found in [8]. More interesting examples are obtained, when the video sequences of frames generated by color bunches of the real-world frames superimposed on the middle lines of the empty rectangles [9] are watched. The information contained in video sequences of color bunches is sufficient for humans to clearly understand the main content of the scenes, including motion understanding in complex environments. It is clear that to understand the events in video sequences of color bunches, the whole power of the human brain is used. However, the fact that it is possible to recognize scenes, based only on hundreds of colored segments, makes us sure that there exists a hypothetic mathematical theory that can provide similar results completely automatically. In this chapter, the first steps are outlined in constructing of this theory.

To do this, the next steps of segmentation on the set of color bunches such as "horizontal" segmentation along the axis Os and the "vertical" segmentation, involving color bunches of several strips will be discussed. The latter stage requires more sophisticated considerations involving global arguments. The approach to global segmentation presented in this chapter differs from that proposed in [4]. This is connected with an attempt to bridge contour-based and region-based approaches. The new approach also makes it possible to use the semantic information at earlier stages of the segmentation process. In the next subsection, the concepts of contrast and similar color bunches are presented, which are applicable for adjacent bunches both in the same strip and in the adjacent strips.

5.3.3 Partial Order Relation and Contrasts on the Set of Color Bunches

If a narrow strip except for points belonging to textured parts of the strip is evaluated visually in a color image, then the strip can be divided into several rectangular background parts with small details against this background. Some of these small details may be very important (traffic lights, turn signals, and brake lights of cars, etc.). However, the objects of our first interest are the background rectangles that are given by the intersection of global homogeneous objects with this particular strip. In this subsection, the additional structures, which make it possible to find "basic" color bunches in each strip, are specified on the set of color bunches *Bun*. Using these structures, one can be able to select textured parts and find a covering of non-textured parts by basic color bunches. Consider a color bunch b with the characteristics $\mathbf{Int}^b = [beg_b, end_b]$, $\Delta_{\mathrm{H}}^b = \left[H_{min}^b, H_{max}^b\right]$, H_{mean}^b, $\Delta_{\mathrm{S}}^b = \left[S_{min}^b, S_{max}^b\right]$, S_{mean}^b, $\Delta_{\mathrm{I}}^b = \left[I_{min}^b, I_{max}^b\right]$, I_{mean}^b, and $Card^b$.

Denote by $dens(b)$ the density of the bunch $dens(b) = Card^b/L([beg_b, end_b])$, L $([beg_b, end_b] = end_b - beg_b + 1$. Consider all color bunches b_i, whose intervals $\mathbf{Int}_i$, $i = 0, \ldots, n$, intersect the interval $\mathbf{Int}$, $b_0 = b$. Then the function $Max(x) = \max_i (dens (b_i))$ (the maximum of densities of all color bunches passing through the point x) can be introduced.

Definition 1 A color bunch b is dominating on an interval $[c, d] \subset [beg_b, end_b]$ in Os, if at each $x \in [c, d]$, $Max(x) = dens(b)$.

Definition 2 Let St_w be the width of the strip $\mathbf{St}$. If $dens(b) > St_w/2$, then b is called a generating bunch.

In what follows, the concepts of contrast (similarity) for adjacent bunches are introduced. Consider two bunches in the same strip b_1 and b_2 with the intervals $\mathbf{Int}_1$ and $\mathbf{Int}_2$, $\mathbf{Int}_{12} = \mathbf{Int}_1 \cap \mathbf{Int}_2$. Then b_1 and b_2 are in a general position, if both bunches are dominating (generating) ones and $\mathbf{Int}_{12} \neq \varnothing$. One can introduce a partial order relation on the set of dominating (generating) bunches. The bunch b_1 is the left (right) adjacent of $b_{2,}$ if $beg_{b1} < beg_{b2}$ ($end_{b1} > end_{b2}$). One can consider chains of dominating (generating) bunches that cover the non-textured parts of the strip. To perform further segmentation steps, let us produce a reasonable criterion for similarity (dissimilarity) of adjacent generating (dominating) color bunches.

As a rule, the particular objects that are clearly visible for humans in the real world have neighborhoods on the sides of the object boundary (the background close to the object of interest) that are in contrast with the object part near the boundary. A human vision is able to distinguish even very small contrasts. The human ability to notice small contrasts is partially based on the ability to analyze the distribution of feature values. In grayscale images, humans can distinguish the regions that differ only in few grades of grayscale in the case, when the intensity ranges of the regions are non-overlapping or have small intersection. In addition, in different ranges of H, S, I the distinctive abilities of human vision are different. For example under low illumination, the perceptibility of colors strongly varies. There are many references in [10] of psychologists on the determination of limits in color variations invisible for human vision. The results have been shown to be dependent on the color range. Note that these experiments have been conducted under the condition of constant intensity. Using these hints, the distinguishing functions on the set of color bunches may be introduced. The problem is solved in a more general statement, taking into account the grayscale component and accepting results that are not so precise. The decisions will be obtained of the following three types:

- The considered bunches are similar with several degrees of similarity.
- The considered bunches are in contrast with several degrees of contrast.
- The decision is not made because additional considerations are required.

Let us describe the procedure of finding contrast and similar bunches. Based on numerous own experiments with real-world color images and using the investigations of psychologists and neurophysiologist [10], the rules for the determination of contrast can be proposed. This problem is not purely mathematical and has to be solved taking into account the specific features of human vision. Since this approach is oriented to the robotic applications, the human operator should understand the decisions made by robots based on a computer vision. Therefore, a robot vision has to be as close to the human one as possible. The mechanisms and algorithms of human vision have not been investigated in necessary detail yet. Thus, the single way, that is available now, is to synthesize algorithms based on other principles convenient for computer implementation and to try to obtain on numerous images and video sequences the results that suit our perception. A base of productions was designed in order to solve the posed problem. At present, this base contains more than 2,500 lines of program code. The author is continually extending the list of rules in order to improve the ability of the base to deal with real-world images. It is expected to publish the text of the procedure implementing the contrast (similarity) determination including the list of rules in the Internet, when it will be developed in more or less final form. Since the base is rather bulky, in this chapter only the main ideas, on which it leans, are presented.

Consider two adjacent bunches b_i, $i = 1, 2$. These bunches are characterized by intervals $\mathbf{Int}_i$ on the axis Os (geometry) and three intervals of color characteristics $\mathbf{Int}_i^f = \left[f_{min}^i, f_{max}^i\right]$, where $f = H, S, I$, and the three mean values of these color characteristics f_{mean}^i. Here f_{min}^i and f_{max}^i are the minimum and maximum of the corresponding feature. Denote by $df(b_1, b_2)$ the absolute values of the difference of mean values for the corresponding features of the color bunches. To eliminate random deviations, $\mathbf{Int}_i^f$ are symmetrized relative to f_{mean}^i, $\mathbf{Int}_i^f = \left[\left(f_{mean}^i - \Delta\right), \left(f_{mean}^i + \Delta\right)\right]$, where $\Delta = \min\left(\left(f_{mean}^i - f_{min}^i\right), \left(f_{max}^i - f_{mean}^i\right)\right)$. To measure the proximity of the intervals of color characteristics $\mathbf{Int}_1^f$ and $\mathbf{Int}_2^f$ and the geometric intervals $\mathbf{Int}_i$, Eq. 5.1 is used. The values of H and I are divided into eight and six zones, respectively. For each zone of H and I, the production rules are formulated separately. Saturation S takes values from 0 to 15 (maximum saturation). For each value of S and every zone of H and I, the thresholds for possible and impossible deviations of mean values of H, S, I: $PosTh_f(k)$ and $ImPosTh_f(k)$ are chosen, where f is one of the color characteristics, and k is the serial number of the zone of H. Let us develop the production rules for small values of saturation S, $0 \leq S \leq 3$. When $S = 0$, then the grayscale intensity is the most significant feature. When saturation grows from 1 to 3, the intervals between lower and upper thresholds in hue become narrower. In this range, the threshold of darkness can be also introduced for each value of S, when the bunches are considered as colorless and purely dark. The rules for the general case start from $S = 4$.

However, the introduced thresholds are taken to eliminate the simplest cases, in which it is possible to judge based on mean values only. In the majority of cases, when the deviations of mean values belong to intervals between possible and impossible deviations, more elaborate criteria are employed, applying the proximity measures Eq. 5.1 to the variation intervals of color characteristics. To determine the proximity of the bunches in hue, saturation, and intensity functions $hue\_close(b_1, b_2)$, $sat\_close(b_1, b_2)$, and $inten\_close(b_1, b_2)$ are introduced with the set of values (3, 2, 1, 0, −1, −2, −3). Based on $df(b_1, b_2)$ and $d_{min}\left(\mathbf{Int}_1^f, \mathbf{Int}_2^f\right)$, $d_{max}\left(\mathbf{Int}_1^f, \mathbf{Int}_2^f\right)$, the appropriate values from 3 (best similarity) to −3 (best contrast) through 0 (indefinite contrast) are assigned to $hue\_close(b_1, b_2)$, $sat\_close(b_1, b_2)$, and $inten\_close(b_1, b_2)$. Then based on the introduced functions of proximity of color characteristics, the discriminating function $Discr(hue\_close, sat\_close, inten\_close) = Discr(b_1, b_2)$ is constructed, which takes the values from the set (4, 3, 2, 1, 0, −1, −2, −3, −4). This function assigns to the pair of bunches the degrees of similarity (4, 3, 2, 1) or contrast (−1, −2, −3, −4). The value 0 means that the decision was not made.

Definition 3 Let b_1 be a dominating bunch, and let b_2 be the right (left) adjacent dominating bunch such as $\mathbf{Int}_{12} \neq \varnothing$. If according to $Discr(b_1, b_2)$ the bunches are contrasting, then the left (right) end beg_{b1} (end_{b1}) of the bunch b_1 is called the left (right) virtual boundary point in the set of bunches, and b_1 (b_2) is the left (right) contrast bunch to b_2 (b_1).

5.3.4 Structural Graph of Color Bunches and Continuous Left and Right Contrast Curves on It

On the set *Bun*(**CI**), let us introduce a graph structure. The set *Bun*(**CI**), furnished with a graph structure, is called the structural graph of the color image *STG*(**CI**) or simply *STG*. Let us explain how to introduce the graph structure on *Bun*(**CI**). Denote by Bun^i(**CI**) or simply by Bun^i color bunches belonging to $\mathbf{St}_i$. Any two bunches b_1, b_2 belonging to the same strip such that the intersection of their geometric components is not empty $\mathbf{Int}_{12} \neq \varnothing$ are connected by an edge $ed(b_1, b_2)$. These edges are called "horizontal" edges. Each horizontal edge is equipped with the value of the contrast function $Discr(b_1, b_2)$. In the same way, the "vertical" edges are constructed that connect bunches belonging to adjacent strips. Assume that $b_1 \in Bun^i$ and $b_2 \in Bun^j$. Since for all strips the intervals $\mathbf{Int}_i \in$ Os, $i = 1, 2$, belong to the same axis Os, they can be compared for color bunches of different strips. Similarly, the intersection $\mathbf{Int}_{12}$ of the intervals determined by b_1 and b_2 is also defined for bunches belonging to different strips. Let $j = i + 1$. Let us connect b_1, b_2 by a vertical edge, if $\mathbf{Int}_{12} \neq \varnothing$, and assign again to this edge the value of $Discr(b_1, b_2)$. The set *Bun*(**CI**) with the introduced edges is called the structural graph of color bunches *STG*(**CI**) or simply *STG*. Our goals are as follows: to find the sequences of color bunches in adjacent strips that correspond to real boundaries of objects in the image and to develop the algorithms for finding such sequences, which will provide the brief descriptions of real boundary curves in the image.

For these purposes, let us introduce two types of extremal vertical edges. They are called left and right extremal vertical edges. Left (right) extremal vertical edges connect color dominating (generating) bunches b_1, b_2 in adjacent strips, whose left (right) ends are left (right) virtual contrast points. Bunches b_1, b_2 are connected by a left (right) extremal edge, if b_2 is the continuous extension of the bunch b_1 determined in the following way. Assume for definiteness, that b_1 has the left contrast end and define its left continuous extension. The case of the right contrast end and the right continuous extension is defined similarly. Consider the color bunches $b_2 \in Bun^{i+1}$ such that each b_2 has the left contrast end, and $\mathbf{Int}_{12} \neq \varnothing$. For the intervals $\mathbf{Int}_j$ determining the geometric component of b_j, $j = 1, 2$, the preliminary right continuous extension $\mathbf{Int}_j^c$ is defined. This extension is the preliminary segmentation in the horizontal direction. Consider a sequence of right adjacent bunches b_j^k of the same strip such that: (1) $b_j^0 = b_j$ and b_j^{l+1} is the right adjacent for b_j^l and (2) b_j^l and b_j^{l+1} are similar in the sense of $Discr(b_j^l, b_j^{l+1})$. Note that the transition from b_j^l to b_j^{l+1} is estimated by $Discr(b_j^l, b_j^{l+1})$ as the transition from one interval with similar color characteristics to another.

A kind of weak segmentation along the strip, changing the geometric component of the bunch and leaving the color characteristics the same, is perfumed. Therefore, consider the virtual bunch b_1^c, whose color characteristics are equal to the color characteristics of b_1, while the interval $\mathbf{Int}_1$ is replaced by $\mathbf{Int}_1^c$. The local extension of the contrast boundary point beg_1 of the bunch b_1 to the next strip is detected and then tested whether this left contrast point belongs to a continuous sequence of left

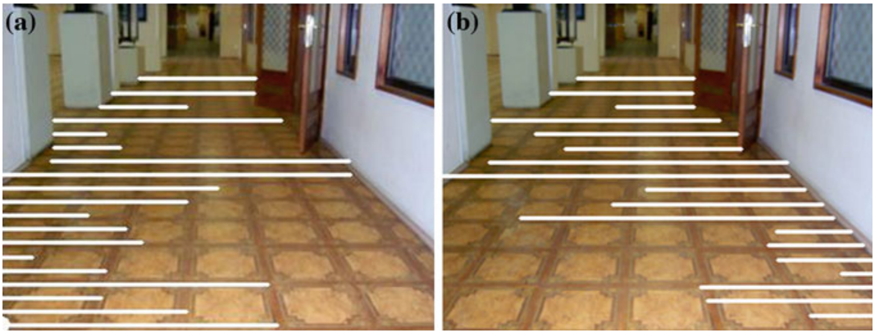


Fig. 5.3 Example of *left* and *right germs* (*left* and *right* contrast boundary *curves*) of a global object (the *corridor floor*): **a** the *left germ*, **b** the *right germ*

contrast points corresponding to a certain boundary of a real region in the image. Thus, in the procedure the color characteristics in the close vicinity are used, and $\mathbf{Int}_j^c$ allow us to understand whether the points bound a certain global region. In the same way, instead of the bunch $b_2 \in Bun^{i+1}$, one can consider the virtual bunch b_2^c. Then for values of 1, 2 of $Discr(b_1^c, b_2^c)$ (the strongest contrast), the bunches b_2 are found such that $\mathbf{Int}_j^c$, $j = 1, 2$ have a sufficiently big intersection in the sense of measures (Eq. 5.1), and the end beg_2 of b_2 is the most proximate to beg_1 of b_1.

Definition 4 Assume that $b_{j,} j = n_1, \ldots, n_k$, is a sequence of bunches with left (right) contrast boundaries, $b_j \in \mathbf{St}_j$, such that the edge $ed(b_j, b_{j+1})$ is left (right) extremal (the extension is continuous), then the sequence of the left (right) ends beg_j (end_j) of b_j is called a virtual left (right) contrast boundary curve in the STG. The sequence of such bunches b_j is called a left (right) germ of a contrast object in the STG bounded by this left (right) virtual contrast boundary curve.

An example of left and right germs of a contrast object in the STG can be found in Fig. 5.3. In this figure, the left and right germs of the floor are presented. As a rule, the homogeneous regions in the image generate the continuous sequences of left (right) contrast points in the STG, as well as left (right) germs of contrast objects in the STG, and vice versa. Numerous experiments have shown that the continuous sequences of contrast boundary points, marked on the middle lines of strips, determine the certain boundaries of regions in the image.

5.4 Construction of Global Contrast Objects in STG

To generate "global contrast objects" from left and right germs of contrast objects, let us introduce on the set of left and right germs the structure of a bipartite graph.

Definition 5 Let G_l and G_r be left and right germs of contrast objects. Then G_l and G_r are continuously connected germs, if bunches of these germs (at least in one strip) belong to a sequence of adjacent similar bunches, connecting them (in the sense of the contrast-similarity function *Discr* introduced).

Let us introduce the graph LRG. Left vertices of the LRG are generated by left germs of contrast objects, and right vertices of the graph are generated by right germs of contrast objects. The vertices G_l and G_r are connected by an edge, if these germs are continuously connected in the sense of the above definition. Candidates for objects are connected components in the LRG (all bunches in G_l and G_r and all chains of similar bunches that join bunches of G_l and G_r). It is clear that real regions in the image (possibly partially occluded) should generate connected components in the LRG. Let us present algorithms for finding connected components in the LRG.

Assume that G_{l1} is a left germ of a contrast object $G_{l1} = (b_j^{l1},\ldots, b_k^{l1})$, where b_i^{l1} is a bunch that belongs to the strip $\mathbf{St}_i$, $1 \leq j \leq k \leq N$, and N is the number of parallel, nonintersecting strips, into which the image *CI* is divided. Each bunch b_i^{l1} has a left contrast boundary bunch. Let $M_{\max}$ be a maximum of the numbers of color bunches in the strips $\mathbf{St}_i$ over all $1 \leq i \leq N$. Let us introduce two matrices $Trace^l(N, M_{max})$ and $Trace^r(N, M_{max})$ that fixes the traces of the constructed left and right germs of contrast objects on the STG. $Trace^l(i, j) = 0$ ($Trace^r(i, j) = 0$), if in $\mathbf{St}_i$ there is no left (right) germs that involves the bunch with the number j. Otherwise, $Trace^l(i, j)$ or ($Trace^r(i, j)$) is equal to the serial number of the left or right germs that passes through this bunch plus one. These matrices are formed in the procedure for constructing left (right) germs of global contrast objects.

Consider the construction of a connected component on the bipartite graph LRG that contains $G_{l1.}$ To construct this component, let us successively move from bunches b_i^{l1} through the right adjacent bunches that are similar to b_i^{l1} and look at the values of $Trace^r(i, j)$ until a germ of a right contrast object will be met. Moving along all strips, a set of right contrast germs $RG(S_{l1})$ will be attached to G_{l1} Then in this way, moving to the left, a set of left germs $LG(RG(S_{l1}))$ will be attached to the obtained set of right germs $RG(S_{l1})$. At this step, we attach to G_{l1} the sets $RG(S_{l1})$ and $LG(RG(S_{l1}))$. Assume that L and R left and right germs of contrast objects have been constructed, respectively. Note that $G_{l1} \in LG(RG(S_{l1}))$. Proceeding in this way, the whole connected component is obtained. In the inverse path from right contrast germs, $Trace^l(i, j)$ is used instead of $Trace^r(i, j)$. The connected component may contain d left germs G_{l1}, …, G_{ld} and m right germs G_{r1}, …, G_{rm}. This collection of left and right germs, as well as the intermediate similar bunches that connect the corresponding left and right germs, determine a geometrical figure with boundaries specified by the boundary points of the germs G_{l1}, …, G_{ld} and G_{r1}, …, G_{rm}. Also the color characteristics of all bunches, which are involved in the obtained figure, are known. In this way, a template for recognition of objects in the image given by color characteristics and geometry will be obtained. The results of the process of attaching to each left germ G_{li} a set of right germs can be specified in the matrix $Con(i, j)$, $1 \leq i \leq L$, $1 \leq j \leq R$. If there is no connection between the ith and jth germs, then $Con(i, j) = 0$, otherwise $Con(i, j) = c$, where c is the number of strips, via which the ith and jth germs are connected. This matrix determines all connected components in the LRG.

Using the developed technique, the complex contrast objects can be found in the STG. Since in real world actual objects frequently are not homogeneous with respect to color characteristics, but contrast objects are clearly discernible in the background of other objects, the proposed segmentation scheme can formalize segmentation of contrast objects, whose color characteristics vary strongly (lighting, shadows, etc.). This is especially true for indoor scenes, when there are several sources of light, which are located at relatively short distances to objects. Color variations can be taken into account by using several degrees of similarity in *Discr* (*b*, *lb*) (*Discr*(*b*, *rb*). As in human vision, in this approach sometimes, one can mainly take into account shape (given by the end points of left and right germs) and contrasts. In this case, one can lean on color characteristic to a lesser degree.

If all bunches of the constructed contrast object from the STG are overlaid on the middle lines of the corresponding strips, then a geometric pattern in the image is obtained. The procedure of constructing color bunches [4] proves that a certain approximation of parts of real objects in the image is obtained. In addition, the boundaries of the object in the STG (end points of contrast germs G_{l1}, …, G_{ld} and G_{r1}, …, G_{rm}) approximate the boundaries of the real object. This means that the search procedures can be implemented in images using only computations in the STG. This immensely accelerates the search procedures and makes it possible to solve these problems in real time. One may compare with [11], where the required goal was not achieved. In the next section, the application of the technique developed to robot navigation is described and an example is given.

5.5 Applications to the Navigation of Robots in Indoor Environments

The developed algorithms for constructing the structural graph of a color image *STG*(**CI**), left and right contrast boundary curves (left and right germs of global objects), the graph LRG of connections between left and right germs, and global contrast objects in the STG, based on the proposed technique, have been implemented as a program (vision) system written in C++ in Windows and Linux operating systems. A visualization program was developed to illustrate each step of processing. Under Linux, using OpenCV tools, this vision system was connected with video input from cameras (including network cameras) and video files, as well as with video records of processing results. The performance of the vision system is about 20 fps for sequences of images of dimension 640 × 480 for personal computers with processors from Intel i5, i7 series. This vision system was installed on an “Amur” robot produced by “Sensorika” international laboratory. This robot is controlled by a remote computer via a network. The necessary data are transmitted via Wi-Fi from the robot to the remote computer. In the inverse direction, control commands are sent to the motors of the robot wheels in order to realize motion plans produced by the remote computer. A camera is mounted on the robot and provides video sequences for solving different navigation problems.

Consider the application of the developed technique to finding visual landmarks in the navigation of the “Amur” robot. Two types of problems have been solved. In the first type, the robot tries to find rather small visual landmarks. As a model problem of this type, the finding and tracking a label, consisting of several colored geometrical figures, is considered. The algorithms based on the proposed technique have a rather general nature and can be applied to finding different types of posters and signs located at different places of indoor and outdoor environments (e.g., emergency exit, road signs, construction signs, breakdown triangles on roads, etc.). In addition using these artificial landmarks, it is possible to control the robot by a human in presentations and to guide the motion of a group of robots. Tuning these algorithms to the problem of finding things in rooms given by a template or a verbal description (e.g., in order to assist disabled persons [2, 11]) is under consideration. With respect to the problems of the second type, one mention challenges of finding rather big landmarks such as doors, windows, and walls in order to provide navigation of robots in indoor environments based mainly on recognition rather than on measurements, in the way humans do.

As an example of a problem of the first type, consider a robot motion to an artificial visual label assembled of three colored rectangles (see Figs. 5.4 and 5.5). The label may be fixed to an object in the environment (e.g., to a wall) or be in the hands of a human moving in front of the robot at a certain distance (Fig. 5.4). It is

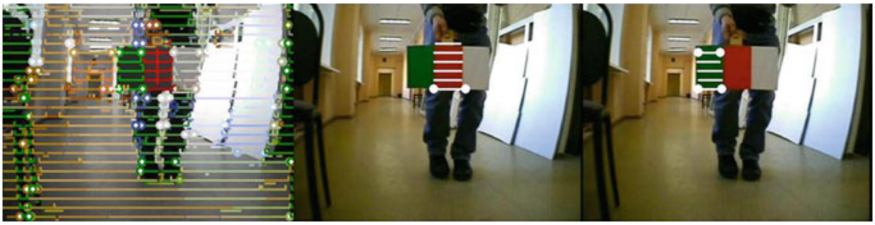


Fig. 5.4 Results of frame processing, when the STG is put on the grayscale component, and two selected *rectangles*

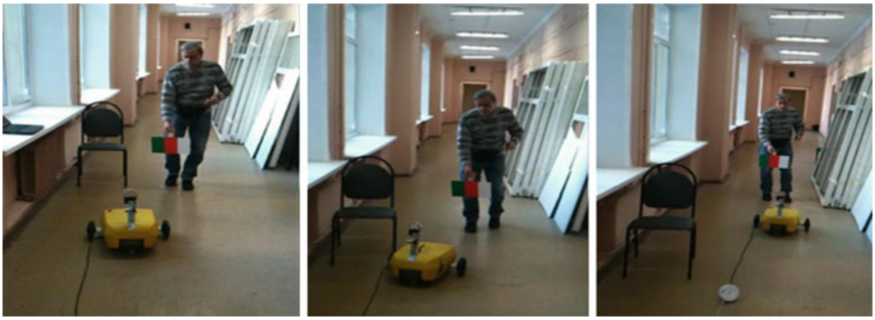


Fig. 5.5 Video record of an experiment

allowed that the human is able to maneuver leftward or rightward relative to the direction of motion in an arbitrary way. The robot should stop, if a certain small distance to the label is reached. The control is based on pure visual feedback and is performed using only video information received from the camera, without application of any other sensors (no dead reckoning, radio beacons, etc.). Note that it does not mean that the author is not going to use these useful sensors in the navigation of the robot. This restriction is considered in order to underline the capabilities of visual feedback. There are no conditions of observation imposed in advance. The label is visible in the image, and its colors have to be distinguishable, but may significantly change because of different types of illumination. The robot tries to find this visual landmark without any information on its size in the image. The size may vary from several pixels (at the stage of finding the label) to dimensions comparable with the dimension of the whole image. Also the author does not confine himself to solving the problem under stable illumination conditions. It is supposed only that the color characteristics of the rectangles constituting the label vary in wide ranges, containing the specified colors of the rectangles. For example, when the robot tries to find a label consisting of green, red, and white rectangles, it is supposed only that the green rectangle can have color from yellow to cyan, and the red rectangle may have color from violet to yellow. In addition, the difference in hue grades between the red and green rectangles has to be greater than a certain number (not big), and the white rectangle has to have a smaller saturation than the previous two rectangles and a greater intensity. This means that the conditions imposed are mainly of qualitative nature. Information about the background, against which the label is found, is absent. This background may be very complex (e.g. bookshelves). The illumination may change sharply. For example, in tests the electric lighting was switched on and off, and a robot traveled in corridors with changing (because of the distance to the sources of electric lighting) and low illumination in order to prove the stability of vision algorithms. The rectangles' dimensions of the label and the fact, that the orientation of the vertical axis of the label does not differ much from the orientation of the local vertical axis, are known.

Let us describe briefly the algorithms for solving the posed problem. These algorithms and experiments with the robot will be described in detail in a particular publication. To find and follow the label, the partition of the image into parallel horizontal strips of width about ten pixels is used. Consider for definiteness the case of a label constituted of three rectangles—green, red, and white ones. In the set of partially ordered color bunches of each strip, the algorithm finds three sequences of color bunches. The conditions, imposed on the bunches in each sequence, are as follows:

- There are no contrasts between bunches in each sequence.
- The color characteristics of bunches within each sequence belong to the corresponding color range.
- The boundary bunches of each sequence have contrasts with the adjacent bunches of the adjacent sequence.
- The total sizes of the sequences in pixels are approximately the same.

For each boundary bunch, the left and right germs of global contrast objects are detected, and the shape of objects obtained using the segmentation with the LRG is analyzed. Since very weak constraints on the triples of sequences of color bunches are imposed, it is possible that the STG may contain false candidates for the label. Sometimes the appearance may be corrupted by highlights, caused by bright illumination or by occlusion. In this case, the connectedness of the label may be violated. The situation can be complicated by the presence of false labels (triples of sequences of bunches satisfying the color conditions). False labels occasionally arise in large halls with many objects, including crowds of people (e.g., on big presentations of robots). To determine the direction of motion towards the label, it is required to find the center of the horizontal projection of the central rectangle. The sequences of color bunches of each of the candidates were combined into three integrated color bunches (b_1, b_2, b_3). Assume that n triples of integrated bunches $Tr^i = \left(b_1^i, b_2^i, b_3^i\right)$, $i = 1, \ldots, n$, are situated in the STG located in different strips and satisfying the color conditions. These triples may belong to the same label, if the label is sufficiently big and is located in several strips, into which the image was divided. Among these triples, those that belong to false candidates may be detected. Let $d_2^i \in \mathrm{Os}$ be the center of the interval $\left[\mathrm{beg}_2^i,\ \mathrm{end}_2^i\right] \subset \mathrm{Os}$ of the color bunch b_2^i. A clustering procedure is applied to the points d_2^i and intervals $\left[\mathrm{beg}_2^i,\ \mathrm{end}_2^i\right]$ and triples into hypothetical labels are assembled. In the clustering procedure, the proximity of d_2^i as integers and the proximity of intervals $\left[\mathrm{beg}_2^i,\ \mathrm{end}_2^i\right]$ in the sense of measures (Eq. 5.1) are used. As a result, several clusters of triples in the vertical direction can be obtained. In each cluster Cl_j, the corresponding intervals $\left[\mathrm{beg}_2^i,\ \mathrm{end}_2^i\right]$ have strong intersections. One can assign to each of the clusters the value of the cost function $Cost(Cl_j) = \sum_k Col(Tr^k) + \sum_k Sim(b_2^k) + Sh(Cl_j)$. Here the first term evaluates, how each triple fits the color condition, the second term estimates, whether these intervals generate a rectangle, and the third term tests whether the obtained rectangle has the desired ratio of lengths of its sides.

As a result, the candidate that has the biggest value of the cost function is chosen, and the color and shape conditions are fitted in the best way. In this way, false candidates are eliminated. In the case, when the label was found reliably in the previous frames of the video sequence, its position, color characteristics, and the value of its cost function can be used to eliminate false labels.

To control the robot in its motion towards the label, an intelligent controller was used with a set of control parameters. The control parameters are as follows:

- An array of 32 last positions of the center of the central rectangle.
- The previous and current positions of the left and right boundary sides of the central rectangle.
- Several last positions of the lowest and highest triples of bunches of the label.
- The approximate distance to the label, calculated based on the pixel dimensions of the label and the results of camera calibration.

There are several modes of robot motion:

- The label search.
- The direct motion in the chosen direction.
- Several modes of left turn with different angular velocities.
- Several modes of right turn with different angular velocities.
- The stopping at a small distance in front of the label.

The control is performed in order to keep the horizontal projection of the center of the central rectangle on the axis Os in a neighborhood of the middle of the projection of the screen on this axis. Also it is better to keep the label within the screen. If the label goes out of the screen (e.g., because of intentional fast actions of the human), the controller should catch it again based on the analysis of its trajectory. At the *k*th step of control, each state of the robot is determined by the array of parameters (j^k, d^k, $dist^k$, [l^k, r^k], [tp^k, bt^k]), where j^k is the state of the controller (search, direct motion with a certain speed, left turn of a certain type, right turn of a certain type, termination), d^k is the position of the center of the central rectangle, $dist^k$ is the estimate of the distance to the label, [l^k, r^k] is the projection onto the axis Os of the central rectangle, and [tp^k, bt^k] is the projection on the vertical axis of the central rectangle. Using these parameters, control rules are formed. Let c be the horizontal coordinates of the center of the screen, $c = dim\text{X}/2$, where dimX is the horizontal dimension of the image array in pixels. The deviation of the robot from the label is $dev = d - c$, $-c \le dev \le c$. Depending on the sign of dev and its absolute value $|dev|$, the mode of forward motion and of left (right) turn with an appropriate angular velocity is chosen. A certain insensitivity zone of small deviations from the center, within which the forward motion is chosen, is selected. The rest of the range of variations of $|dev|$ is divided into several intervals. For each of them, a certain left (right) turn with an appropriate angular velocity (a certain control of the left and right electric motors of the leading robot wheels) is chosen. If $dist^k < \alpha_{\text{stop}}$, then the smallest possible distance is reached, and the robot is stopped. Using this, one can stop the robot at any moment, if the label to the robot is in close vicinity in front of it. If the label leaves the frame through the top boundary of the screen, the robot goes backward. If the label goes out of the screen (due to its sharp movements) through the left or right side, its motion is approximated based on the record of the trajectory. Using this prediction, the control j^{k+1} is chosen to provide the quick return of the label to the screen. The application of the developed control is illustrated by video sequences with the record of a robot rides that can be found in [9].

The video sequence shows the sharp reactions of the robot to variations of the label motion. Figures 5.4 and 5.5 present results of processing in one of the experiments. The left frame in Fig. 5.4 shows the STG overlaid on the grayscale component of the color image. Contrast color bunches have small vertical segments at the ends. The generating color bunches are overlaid on the middle lines of the corresponding strips. Other color bunches are put a bit lower. The central rectangle is labeled by vertical segments demonstrating that this triple was identified as belonging to the label.

In conclusion, here are few remarks on present and future applications in the field of control of the proposed approach. At present, the robot is trained in finding closed and open doors in corridors and rooms. For this purpose, the technique provided by the graph LRG in the STG is used. In this work, the methods for estimating the shape of left (right) contrast curves (boundaries of left (right) germs of global contrast objects in the STG were developed. Particularly, the criteria for finding straight segments in contrast boundary curves stable against segmentation errors were proposed. Another application will be connected with autonomous road following of a "Niva" off-road vehicle. The developed technique will be applied to finding roads and various objects on it, as well as in its close vicinity. These works will be the subject of the next publications.

5.6 Conclusion

In this chapter, a new method for contextual image description and segmentation of color images was presented. Based on this vector description, a new technique that makes it possible to formalize the concepts of contrast boundary curves and contrast objects in the image was proposed. The developed method provides the rich techniques for real-time solution of recognition problems in order to find objects in images with given shape and color characteristics. These objects can be found by using the proposed vector description only without additional operations on the image array. This immensely increases the performance of the method and makes it possible to solve in real time various tasks, arising in navigation problems of robots. The developed technique has been implemented in a vision system that was installed on a mobile robot. Experiments in autonomous navigation of the robot have been conducted, and first successful results have been obtained. The problem of finding artificial visual landmarks by the robot vision system was addressed. A control algorithm with visual feedback for a robot following a moving landmark consisting of colored geometric figures has been presented, and the results of its application were discussed. Future work for application of the developed technique in the indoor and outdoor navigation of robots was outlined.

References

1. Bonin-Font F, Ortiz A, Oliver G (2008) Visual navigation for mobile robots: a survey. J Intell Rob Syst 53(1):263–296
2. Mishra AK, Aloimonos Y (2009) Active segmentation. Int J Humanoid Rob 6(3):361–386
3. Hedau V, Arora H, Ahuja N (2008) Matching images under unstable segmentation results. IEEE Int Conf Comp Vis Pattern Recogn 44:1614–1628
4. Kiy KI (2010) A new real-time method for description and segmentation of color images. Pattern recognition and image analysis. Adv Math Theor Appl 20(2):169–176

5. Kii KI (1992) Topologically geometric method of processing images in large and its application to analysis of road scenes. Techn Kibern 5:244–248 (in Russian)
6. Kiy KI, Klimontovich AV, Buyvolov GA (1995) Vision-based system for road following in real time. Int Conf Adv Rob 1:115–124
7. Kiy KI, Dickmanns ED (2004) A color vision system for analysis of road scenes. Proceedings of IEEE intelligent vehicle'2004 symposium, pp 54–59
8. Color Vision (2014). http://sites.google.com/site/colorvisionkikiy/. Accessed 15 June 2014
9. My Videos (2014). http://video.mail.ru/mail/kikip_46/_myvideo/. Accessed 15 June 2014
10. Forsyth DA, Ponce J (2003) Computer vision. A modern approach. Prentice Hall, New York
11. Meger D, Muja M, Helmer S, Gupta A et al (2010) Curious George: an integrated visual search platform. Canadian conference on computer and robot vision (CRV'2010), pp 107–114

Chapter 6
Perception of Audio Visual Information for Mobile Robot Motion Control Systems

Snejana Pleshkova, Alexander Bekiarski, Shima Sehati Dehkharghani and Kalina Peeva

Abstract Motion is the main characteristic of intelligent mobile robots. There exist a lot of methods and algorithms for mobile robots motion control. These methods are based on different principles, but the results from these methods must leads to one final goal—to provide a precise mobile robot motion control with clear orientation in the area of robot perception and observation. First, in the proposed chapter the mobile robot audio and visual systems with the corresponding audio (microphone array) and video (mono, stereo or thermo cameras) sensors, accompanied with laser rangefinder sensor, are outlined. The audio and video information captured from the sensors is used in the perception audio visual model proposed to perform joint processing of audio visual information and to determine the current mobile robot position (current space coordinates) in the area of robot perception and observation. The captured from audio visual sensors information is estimated with the suitable algorithms developed for speech and image quality estimation to apply the preprocessing methods for increasing the quality and to minimizing the errors of mobile robot position calculations. The current space coordinates determined from laser rangefinder are used as supplementary information of mobile robot position, for error calculation and for comparison with the results from audio visual mobile robot motion control. In the development of the mobile robot perception audio visual model, some methods are used: method RANdom SAmple Consensus (RANSAC) for estimation of parameters of a mathematical model from a set of

S. Pleshkova · A. Bekiarski (✉)
Department of Telecommunications, Technical University of Sofia, 8 Bld. Kliment Ohridski
Sofia, 1000 Sofia, Bulgaria
e-mail: aabbv@tu-sofia.bg

S. Pleshkova
e-mail: snegpl@tu-sofia.bg

S.S. Dehkharghani · K. Peeva
French Language Faculty of Electrical Engineering, Technical University of Sofia, 8 Bld.
Kliment Ohridski Sofia, 1000 Sofia, Bulgaria
e-mail: sh.sehati@gmail.com

K. Peeva
e-mail: kala_peeva@yahoo.com

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_6

observed audio visual coordinate data; method Direction Of Arrival (DOA) for sound source direction localization with microphone array of speaker sending voice commands to the mobile robot; method for speech recognition of the voice command sending from the speaker to the robot. The current mobile robot position calculated from joint usage of perceived audio visual information is used in appropriate algorithms for mobile robot navigation, motion control, and objects tracking: map based or map less methods, path planning and obstacle avoidance, Simultaneous Localization And Mapping (SLAM), data fusion, etc. The error, accuracy, and precision of the proposed mobile robot motion control with perception of audio visual information are analyzed and estimated from the results of the numerous experimental tests presented at the end of this chapter. The experiments are carried out mainly with simulations of the algorithms listed above, but are trying also parallel computing methods in implementation of the developed algorithms to reach real time robot navigation and motion control using perceived audio visual information from the mobile robot audio visual sensors.

Keywords Audio visual perception · Human perception · Intelligent robot · Mobile robot · Motion control · Audio visual object tracking

6.1 Introduction

Robot perception is an important characteristic of all modern intelligent robots [1] closely related to the human perception [2]. Although the robot perception tries to copy a human perception system, there are significant differences between human and robot perception. These differences are not only in the hardware and software robot perception system realization. Generally, these differences are in understanding and in precision of modeling the human perception using mathematical interpretation or heuristic interpretation of visual information from the environments around the mobile robots. There are a lot of scientific publications and articles trying to solve the general robot perception problem mostly related and compared them with the same characteristics of the human perceptions [3–6]. The main advantages from these articles are the conclusions that it is necessary to have a general representative robot perception model containing all existing human perception characteristics and applying this general model for solving concrete more often practical robot application tasks. There exist a large number of examples of robot applications, where the robot perception models help to develop the effective algorithms in wide range of robot applications from robot manipulators [7–9] to mobile robots [10–13] and humanoid like robots [14, 15]. In this chapter, the attention is focused to mobile robot perception and especially for mobile robot motion control. The task of mobile robot motion control is well presented in scientific literature [16–18], and also there are the applications strictly directed only to audio robot perception [19] and only to visual robot perception [20]. From the presented above short analysis of wide spread basic publications in area of mobile robot perception is formulated the proposition of an audio visual mobile robot perception system and algorithms for mobile robot motion control. Based on the condition to perform joint processing of audio visual information, it was proposed to determine the current wheel mobile robot position (space coordinates) in an indoor area of a robot perception and observation (tests are carried out in a corridor with strongly existence of vertical objects parts).

The presented in this chapter audio-visual system is proposed for mobile robot motion control based on audio-visual and range information perceived from robot sensors (microphone array, video camera, and laser rangefinder). A robot motion is controlled through speech commands, and the Extended Kalman Filter for Simultaneous Localization And Mapping (EKF-SLAM) applied for robot navigation. In the EKF-SLAM, the environment landmarks are vertical edges, perceived from the camera image, associated to corners, perceived from the 2D laser rangefinder. This way of modeling of the environment has the advantage that because there is not high feature clutter, the problem of quadratic complexity of the EKF-SLAM is solved to a great extent. On the other hand, such assumption constraints the proposed system to be applicable only in structured indoor environment, which contains enough vertical edges. The proposed system for mobile robot motion control through the speech commands (Sects. 6.2–6.4) is based on parts of the researches done towards the PhD thesis "Development of Methods and Algorithms for Audio-Visual Mobile Robot Motion Control", conducted at the French Language Faculty of Electrical Engineering, Technical University of Sofia, Bulgaria [21]. The general system is presented in Sect. 6.2. Sensor calibration and robot navigation based on the EKF-SLAM are explained in detail in Sects. 6.3–6.4, respectively. In Sect. 6.5, an algorithm is presented for quality estimation of perceived speech information. Experimental results in Sect. 6.6 show the functionality of the proposed system. The conclusion in Sect. 6.7 puts an end to this chapter.

6.2 Mobile Robot Audio and Visual Perception System

The general system for mobile robot navigation based on its perceptions from its environment is presented in Fig. 6.1. It is assumed that the mobile robot perceives audio-visual information from its environment through a microphone and a video camera and range information through a laser rangefinder. The path of the robot is planned based on the perceived audio information. In other words, it is possible to command the robot to navigate within its environment through the speech commands. It is described in detail in Sects. 6.4.2 and 6.4.3. Robot navigation is based on the EKF-SLAM, in which the environment is modeled based on perceived visual and range information from the camera and the laser rangefinder. Therefore, the sensors are calibrated in order to compensate the systematic errors and also to calculate the relative position of sensors to be able to associate visual perceptions with range information.

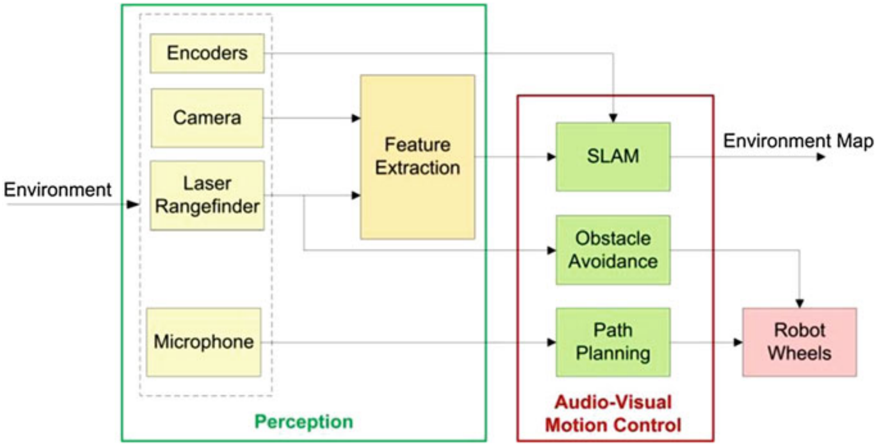


Fig. 6.1 General system for mobile robot navigation within its environment based on perceived audio-visual and range information

6.3 Sensor Calibration Using Mobile Robot Visual and Range Perceptions

The first preparatory step for any system, which consists of several sensors, is the calibration step [22]. Sensor calibration is performed in order to compensate the systematic errors of sensor measurements and being able to transform object coordinates from the world reference frame to the local frame of the sensor and vice versa. By geometric calibration of the camera, its intrinsic parameters are obtained, which are used for compensation of lens distortions. Intrinsic camera calibration method is discussed in Sect. 6.3.1.

After intrinsic calibration of the camera, the laser rangefinder is calibrated with it extrinsically. In this way, the relative position of the sensors are obtained. Subsequently, it is easy to find correspondence between the data provided by each of them.

Thus, in order to achieve precise results from the proposed system, the camera parameters are first computed by geometric calibration of the camera. Then, the 2D laser rangefinder is calibrated extrinsically with the camera (see Sect. 6.3.2). In this way, a compensation of systematic errors is ensured, and measurements can be modeled by a Gaussian distribution containing uncertainty (three standard deviation) caused by random errors. It is also possible to find correspondence between vertical edges in the image received from the camera and corners in laser data. Thus, each feature is presented by its bearing extracted from visual data and its corresponding range extracted from laser data.

6.3.1 Geometric Video Camera Calibration from Perceived Visual Information of Mobile Robot

A geometric camera calibration method provides the camera parameters used for the transformation of the object coordinates from the 3D world reference frame to 2D image frame and vice versa based on a set of images captured by the camera from a test object with a unique pattern from different positions (with varying angle and depth). The most popular pattern is a printed chessboard pattern. It is important that the pattern produces distinct and well defined corners in the set of images used for camera calibration.

The intrinsic and extrinsic parameters of the camera are determined based on an ideal pinhole camera model. The intrinsic camera parameters include the camera's focal length, the principal point, the lens distortions (tangential and radial), and the scaling factors (for transformation from 3D metric world reference frame to 2D metric image frame and from metric units to pixels). The extrinsic parameters are the rotation matrix and the translation vector of the object reference frame with respect to the camera reference frame. Pixels are assumed to be rectangular (zero skew).

The coordinate systems used for camera calibration procedure is presented in Fig. 6.2 [22], where $\mathbf{P}_O = (x_o, y_o, z_o)$ and $\mathbf{P}_C = (x_c, y_c, z_c)$ are object coordinates with respect to its local frame and camera frame, respectively. The parameter $\mathbf{P}_I = (u_i, v_i)$ represents object coordinates in the image plane in pixels.

Assuming that the object local frame with respect to the camera reference frame is represented by a 3 × 3 rotation matrix **R** and a 3 × 1 translation vector **t**, object coordinates in the camera frame are provided by Eq. 6.1.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{6.1}$$

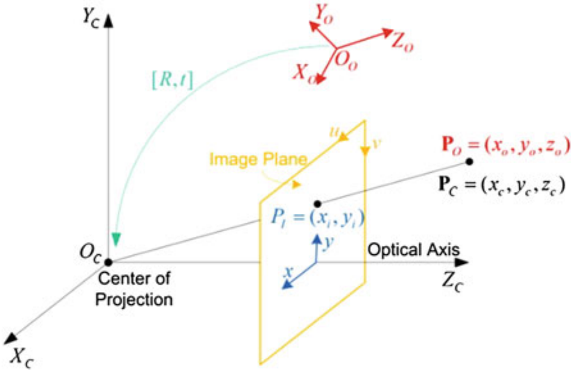


Fig. 6.2 Coordinate systems used in geometric camera calibration procedure

The coordinates of the corresponding point projected to the image plane are calculated using Eq. 6.2, where f is the focal length of the camera.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{6.2}$$

Then, the projected point in the image plane is represented in pixels $\left(u_i^{'}, v_i^{'}\right)$ by Eq. 6.3, where s_u is the scale factor, D_u, D_v are coefficients for conversion from metric units to pixels, $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$ is a principal point.

$$\begin{bmatrix} u_i^{'} \\ v_i^{'} \end{bmatrix} = \begin{bmatrix} D_u s_u x_i \\ D_v y_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{6.3}$$

The pinhole camera assumption is an ideal assumption. The radial and tangential lens distortions are added to the ideal model in order to correct this assumption. Here, only two coefficients are considered for each distortion. The radial and tangential distortions are modeled by Eqs. 6.4 and 6.5, respectively, where k_1, k_2 are radial distortion coefficients, and $r_i = \sqrt{x_i^2 + y_i^2}$, p_1, p_2 are tangential distortion coefficients.

$$\begin{bmatrix} \Delta u_i^{(r)} \\ \Delta v_i^{(r)} \end{bmatrix} = \begin{bmatrix} x_i\left(k_1 r_i^2 + k_2 r_i^4\right) \\ y_i\left(k_1 r_i^2 + k_2 r_i^4\right) \end{bmatrix} \tag{6.4}$$

$$\begin{bmatrix} \Delta u_i^{(t)} \\ \Delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 x_i y_i + p_2\left(r_i^2 + 2x_i^2\right) \\ p_1\left(r_i^2 + 2y_i^2\right) + 2p_2 x_i y_i \end{bmatrix} \tag{6.5}$$

Therefore, the general camera calibration model is obtained by correcting the pinhole model by combining the pinhole model and the radial and tangential distortions describing by Eq. 6.6, where $(\tilde{u}_i, \tilde{v}_i)$ are distorted coordinates.

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \alpha_u \tilde{u}_i \\ \alpha_v \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} D_u s_u \left(x_i + \Delta u_i^{(r)} + \Delta u_i^{(t)}\right) \\ D_v \left(y_i + \Delta v_i^{(r)} + \Delta v_i^{(t)}\right) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{6.6}$$

The camera calibration parameters can be estimated linearly using Direct Linear Transform (DLT) method. In this approach, the nonlinear radial and tangential distortions are ignored and the transformation from object local frame to image frame is assumed to be linear using the homogeneous 3 × 4 matrix **M** [23] represented in Eq. 6.7.

$$\begin{bmatrix} u_i w_i \\ v_i w_i \\ w_i \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{6.7}$$

By eliminating the depth value, w_i, for each control point $\left(x_j, y_j, z_j\right)$, $j = 1, 2, \ldots, N$, Eq. 6.8 is valid

$$\mathbf{L}_j \mathbf{m} = 0,\ j = 1, 2, \ldots N, \tag{6.8}$$

where $\mathbf{m} = [m_{11}, m_{12}, m_{13}, m_{14}, m_{21}, m_{22}, m_{23}, m_{24}, m_{31}, m_{32}, m_{33}, m_{34}]^T$ and $\mathbf{L}_j = \begin{bmatrix} x_j & y_j & z_j & 1 & 0 & 0 & 0 & -x_j u_j & -y_j u_j & -z_j u_j & -u_j \\ 0 & 0 & 0 & x_j & y_j & z_j & 1 & -x_j v_j & -y_j v_j & -z_j v_j & -v_j \end{bmatrix}$.

By replacing $\left(u_j, v_j\right)$ with the coordinates of the observed points $\left(U_j, V_j\right)$, the values of $m_{11}, \ldots, m_{34}$ can be estimated using the least squares method. In order to avoid singularities, the constraint $m_{31}^2 + m_{32}^2 + m_{33}^2 = 1$ is proposed to use in [24].

The main steps of the geometric camera calibration algorithm are depicted in Fig. 6.3 [22]. In the proposed method a sequence of images of the chessboard pattern is captured by the camera from different positions with varying depth and angle. Then, the user selects the extreme grid corners for each image and inputs some geometrical information about the dimensions of the grid cells. These values are used for corner extraction initialization. The coordinates of the points P_i in the image plane are the location of all corners detected in each observed image.

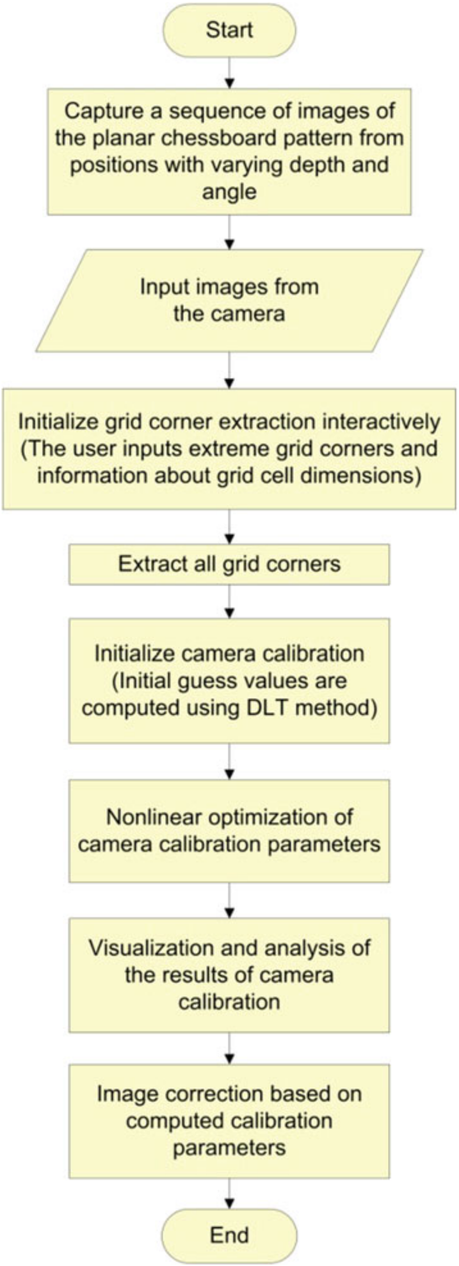
In this stage of the calibration procedure, it is assumed that during image observation only Gaussian noise is present and a systematic measurement noise is compensated. Therefore, the camera calibration parameters are computed by minimizing the least square error between the observed coordinates and the coordinates computed based on the calibration model presented in Eq. 6.6. Considering N corner observations$\{(U_1, V_1), \ldots, (U_N, V_N)\}$, the least squares method is used to minimize Eq. 6.9.

$$E^2 = \sum_{i=1}^{N} (U_i - u_i)^2 + \sum_{i=1}^{N} (V_i - v_i)^2 \tag{6.9}$$

Because the calibration model is a nonlinear, the calibration parameters are estimated iteratively by minimizing Eq. 6.9 using the Levenberg–Marquardt Algorithm (LMA) [25]. In order to avoid the local minimum problem during the iterative optimization process, the initial values of the parameters are computed using the DLT method.

The computed camera calibration parameters are used for image correction. Table 6.1 presents the image correction algorithm. First, a 40 × 40 grid with tie-points (x_i, y_i) is generated, covering the entire image. Distorted coordinates $(\tilde{u}_i, \tilde{v}_i)$ of the corresponding tie-points are calculated. Then, parameters $a_1, \ldots,\ a_8$ of

Fig. 6.3 Geometric camera calibration algorithm


Start
Capture a sequence of images of the planar chessboard pattern from positions with varying depth and angle
Input images from the camera
Initialize grid corner extraction interactively (The user inputs extreme grid corners and information about grid cell dimensions)
Extract all grid corners
Initialize camera calibration (Initial guess values are computed using DLT method)
Nonlinear optimization of camera calibration parameters
Visualization and analysis of the results of camera calibration
Image correction based on computed calibration parameters
End

Eq. 6.10 are estimated iteratively using the least squares method in order to calculate the undistorted coordinates [23], where $N = \left(a_5 r_i^2 + a_6 \tilde{u}_i + a_7 \tilde{v}_i + a_8\right) r_i^2$ and $r_i^2 = \tilde{u}_i^2 + \tilde{v}_i^2$.

Table 6.1 Image correction algorithm

Image correction algorithm
1. Generate a 40 × 40 grid with distorted and undistorted tie-points (x_i, y_i) and $(\widetilde{u}_i, \widetilde{v}_i)$, covering the entire image
2. Calculate the corresponding distorted coordinates $(\widetilde{u}_i, \widetilde{v}_i)$
3. Estimate parameters (a_1, …, a_8) for calculation of the undistorted coordinates iteratively using the least squares method
4. Compute the corrected undistorted coordinates, (x_i, y_i) based on the estimated parameters
5. Calculate all actual coordinates of the image by interpolation based on $(\widetilde{u}_i, \widetilde{v}_i)$ and new (x_i, y_i)

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{1}{N}\begin{bmatrix} \tilde{u}_i\left(1 + a_1 r_i^2 + a_2 r_i^4\right) + 2a_3\tilde{u}_i\tilde{v}_i + a_4\left(r_i^2 + 2\tilde{u}_i^2\right) \\ \tilde{v}_i\left(1 + a_1 r_i^2 + a_2 r_i^4\right) + a_3\left(r_i^2 + 2\tilde{v}_i^2\right) + 2a_4\tilde{u}_i\tilde{v}_i \end{bmatrix} \tag{6.10}$$

Once the parameters are estimated, Eq. 6.10 can be employed for the computation of the corresponding undistorted coordinates. Actual coordinates of the points of the image are calculated by interpolating the computed distorted and corresponding undistorted results.

6.3.2 Camera-Laser Rangefinder Extrinsic Calibration

After intrinsic camera calibration, the extrinsic calibration of the 2D laser rangefinder and the camera is performed in order to calculate the relative position of the camera local frame with regard to the laser local frame by providing the translation vector, $\mathbf{t}_{ci}$ and the rotation matrix, $\mathbf{R}_{ci}$. As a result, point $\mathbf{P}_c$ in the camera frame can be corresponded with point $\mathbf{P}_l$ in the laser frame using Eq. 6.11.

$$\mathbf{P}_l = \mathbf{R}_{ci}\mathbf{P}_c + \mathbf{t}_{ci} \tag{6.11}$$

The calibration is based on observing the same test pattern by both of the sensors from different positions. The position of the test pattern in each observation is obtained based on parameters achieved in camera calibration in previous step, and the line corresponding to the board in laser data is extracted iteratively by minimizing the re-projection error. In order to be able to extract the planar chessboard pattern from laser data in each observation, the planar chessboard has to be moved in an almost static environment for each observation. Therefore, one of the constraints of extracting the board line in each image is that it is a straight line, which changes position in each observation. The other constraint comes from considering the fact that the points belonging to the board line in laser data must lie on the calibration plane, extracted from camera calibration. In other words, assuming that the calibration plane $\mathbf{N}$ is on the plane $z = 0$ and is presented by the translation vector **t** and the rotation matrix **R** provided by camera calibration (Eq. 6.12), the coordinates of each point $\mathbf{P}_l$ of the board line in laser data must be on plane **N** defined by Eq. 6.13, where $\mathbf{R}_3$ is the third column of the rotation matrix **R** and $\mathbf{t}_0$ is the center of the camera in the world frame.

$$\mathbf{N} = -\mathbf{R}_3\left(\mathbf{R}_3^T.\mathbf{t}_0\right) \tag{6.12}$$

$$\mathbf{N} \cdot \mathbf{R}_{cl}^{-1}(\mathbf{P}_l - \mathbf{t}_{cl}) = |\mathbf{N}|^2 \tag{6.13}$$

It is evident from Eq. 6.13 that **N** is normal to the calibration board and its magnitude is equal to the distance from the center of the camera to the calibration board.

Assuming that all board lines in laser data are on the plane $y = 0$, the matrices $\mathbf{R}_{cl}$ and $\mathbf{t}_{cl}$ can be estimated by minimizing iteratively the error, which is defined by the sum of Euclidian distance of laser points from the calibration plane, using Levenberg-Marquardt method. The outliers can also be removed considering the first constraint. Therefore, assuming two described constraints, the translation vector and the rotation matrix are computed iteratively by minimizing the error in the re-projection of the board line in the camera image.

6.4 Navigation of Mobile Robot from Perception of Audio Visual Information

As it is already mentioned in Sect. 6.1, the robot is going to follow speech commands in structured unknown indoor environments. Therefore, the robot navigation within its environment is based on the EKF-SLAM. This is described briefly in Sect. 6.4.1. The robot path planning, presented in Sect. 6.4.2, depends on the recognized speech command. The audio sensor model, sound source localization, and speech recognition are explores in Sect. 6.4.3.

6.4.1 Robot Navigation Based on EKF-SLAM

A robot placed in an unknown environment can concurrently build the map of its surrounding environment while localizing itself by performing the SLAM method. In general, the probabilistic definition of the SLAM is as follows. At each time t, given the control input (obtained from encoders) $\mathbf{u}_t$, and a set of landmark observations (sensor measurements) $\mathbf{z}_t$, the joint posterior density of the robot state $\mathbf{x}_t$ and the landmark locations **M** has the distribution described by Eq. 6.14.

$$P_{post} = P(\mathbf{x}_t\mathbf{M}|\mathbf{z}_t\mathbf{u}_t) \tag{6.14}$$

Using Bayes rule, the posterior probability can be written by Eq. 6.15.

$$P(\mathbf{x}_t, \mathbf{M}|\mathbf{z}_t, \mathbf{u}_t) = \eta P(\mathbf{z}_t|\mathbf{z}_t, \mathbf{M})P(\mathbf{x}_t, \mathbf{M}|\mathbf{z}_{t-1}, \mathbf{u}_t) \tag{6.15}$$

By applying the Theorem of total probability [26] and then the definition of the conditional probability to Eq. 6.15, the posterior probability is described by Eq. 6.16.

$$\begin{aligned} P(\mathbf{x}_t, \mathbf{M}|\mathbf{z}_t, \mathbf{u}_t) &= \eta P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M}) \int P(\mathbf{x}_t, \mathbf{M}|\mathbf{x}_{t-1}, \mathbf{z}_{t-1}, \mathbf{u}_t)P(\mathbf{x}_{t-1}|\mathbf{z}_{t-1}, \mathbf{u}_t)d\mathbf{x}_{t-1} \\ &= \eta P(\mathbf{z}_t|\mathbf{x}_t, \mathrm{M}) \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u})P(\mathrm{M}|\mathbf{x}_{t-1},\mathbf{z}_{t-1}, \mathbf{u}_t)P(\mathbf{x}_{t-1}|\mathbf{z}_{t-1}, \mathbf{u}_\mathrm{t})d\mathbf{x}_{t-1} \\ &= \eta P(\mathbf{z}_t|\mathbf{x}_t, \mathrm{M}) \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_\mathrm{t})P(\mathrm{M}|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}\mathbf{M}|\mathbf{z}_{t-1}, \mathbf{u}_\mathrm{t})d\mathbf{x}_{t-1} \end{aligned} \tag{6.16}$$

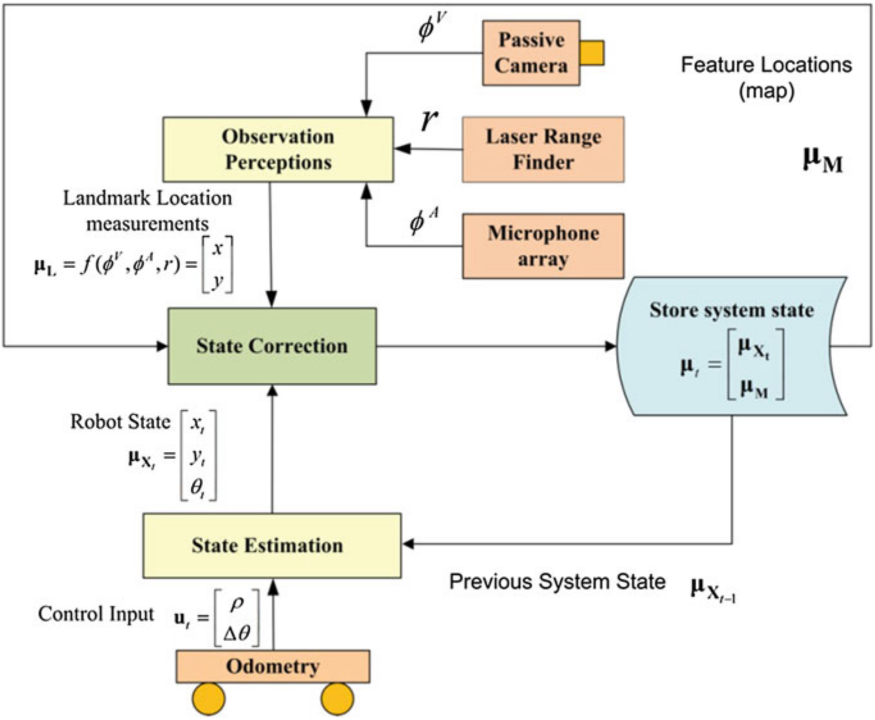
The resultant recursive equation shows that the SLAM posterior probability is a function of the measurement model $P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M})$, the motion model $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$, and the SLAM posterior probability at time (t–1).

The EKF is the most common estimation of the SLAM posterior probability, which represents it as a high-dimensional, multivariate Gaussian parameterized by a mean $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ [27, 28] as it is shown in Eq. 6.17.

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_\mathrm{X} \\ \boldsymbol{\mu}_\mathrm{M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{x_r} \\ \boldsymbol{\mu}_{y_r} \\ \boldsymbol{\mu}_{\theta_r} \\ \boldsymbol{\mu}_{L_1} \\ \vdots \\ \boldsymbol{\mu}_{L_N} \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_\mathbf{X} & \boldsymbol{\Sigma}_\mathbf{XM} \\ \boldsymbol{\Sigma}_\mathbf{MX} & \boldsymbol{\Sigma}_\mathbf{M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_\mathbf{X} & \boldsymbol{\Sigma}_\mathbf{XL_1} & \cdots & \boldsymbol{\Sigma}_\mathbf{XL_N} \\ \boldsymbol{\Sigma}_\mathbf{L_1X} & \boldsymbol{\Sigma}_\mathbf{L_1L_1} & \cdots & \boldsymbol{\Sigma}_\mathbf{L_1L_N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_\mathbf{L_NX} & \boldsymbol{\Sigma}_\mathbf{L_NL_1} & \cdots & \boldsymbol{\Sigma}_\mathbf{L_NL_N} \end{bmatrix} \tag{6.17}$$

The system state consists of the set of landmark locations **M** (where the coordinates are with respect to the world reference frame) and the robot state. The construction of the system state in brief is demonstrated in Fig. 6.4 [21].

The non-linear motion model $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t$ ($\mathbf{w}_t$ is the zero-mean Gaussian system noise with the covariance $\mathbf{Q}_t$) and the measurement model $\mathbf{z}_t = h(\mathbf{x}_{t-1}) + \mathbf{r}_t$ ($\mathbf{r}_t$ is the Gaussian measurement error with the covariance $\mathbf{R}_t = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{bmatrix}$) are linear $\mathbf{z}_t$ around the most likely system state $\boldsymbol{\mu}_{t-1}$, using Taylor



ϕ^V
Passive Camera
Feature Locations (map)
r
Observation Perceptions
Laser Range Finder
μ_M
Landmark Location measurements
ϕ^A
Microphone array
$\mu_L = f(\phi^V, \phi^A, r) = \begin{bmatrix} x \\ y \end{bmatrix}$
Store system state
$\mu_t = \begin{bmatrix} \mu_{X_t} \\ \mu_M \end{bmatrix}$
State Correction
Robot State
$\mu_{X_t} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$
State Estimation
Previous System State $\mu_{X_{t-1}}$
Control Input $u_t = \begin{bmatrix} \rho \\ \Delta\theta \end{bmatrix}$
Odometry

Fig. 6.4 Developed scheme for system state construction in the EKF-SLAM

expansion: $g \approx g + g^I$, where g^I is the Jacobian of the function with respect to its variables.

Considering the above assumptions, the EKF-SLAM is a recursive algorithm that can be divided into two main steps: the state estimation (prediction) and the state update (correction).

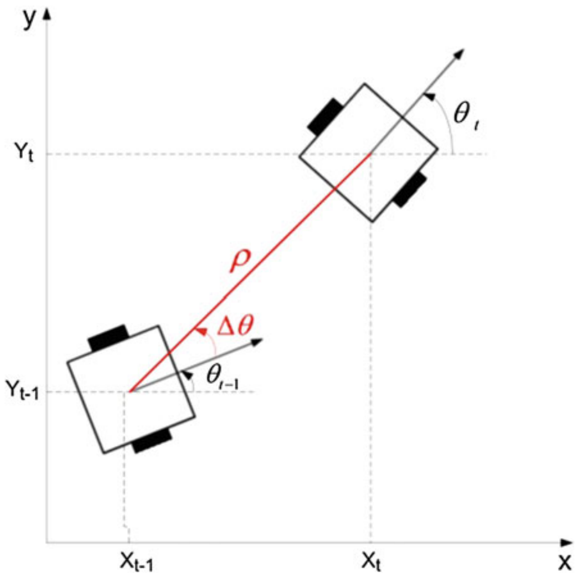
1. State estimation (Prediction).

The estimated state vector and the covariance matrix are calculated from the previous state and covariance, and the control input by Eq. 6.18, where $\mathbf{F}_{x,t} = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\boldsymbol{\mu}_{t-1}}$ is the Jacobian of the state transition function f with respect to the robot state.

$$\bar{\mathbf{x}}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) \ \bar{\boldsymbol{\Sigma}}_t = \mathbf{F}_{x,t}\boldsymbol{\Sigma}_{t-1}\mathbf{F}_{x,t}^T + \mathbf{F}_{u,t}\mathbf{Q}_{t-1}\mathbf{F}_{u,t}^T \tag{6.18}$$

The motion model of the two-wheeled nonholonomic mobile robot is situated in Fig. 6.5. The predicted system state and covariance, considering the odometry model $\mathbf{u}_t = \boldsymbol{\mu}_{\mathbf{u}_t} + \mathbf{w}_t$, are given by Eq. 6.19.

Fig. 6.5 Motion model of a two-wheeled nonholonomic mobile robot



$$\bar{\boldsymbol{\mu}} = \begin{bmatrix} \bar{\boldsymbol{\mu}}_{\mathbf{X}} \\ \boldsymbol{\mu}_{\mathbf{M}} \end{bmatrix} \bar{\boldsymbol{\Sigma}} = \begin{bmatrix} \bar{\boldsymbol{\Sigma}}_{\mathbf{X}} & \bar{\boldsymbol{\Sigma}}_{\mathbf{XM}} \\ \bar{\boldsymbol{\Sigma}}_{\mathbf{MX}} & \boldsymbol{\Sigma}_{\mathbf{M}} \end{bmatrix} \tag{6.19}$$

The only time variant part of the system state is the robot state. Therefore, $\boldsymbol{\Sigma}_M = \bar{\boldsymbol{\Sigma}}_M$ and $\boldsymbol{\mu}_M = \bar{\boldsymbol{\mu}}_M$ (the estimated and the updated map features with regard to the world reference frame are the same).

The robot state and system covariance are predicted by Eqs. 6.20–6.21, where $\mathbf{F_x}$ and $\mathbf{F_u}$ are the Jacobians of the state transition function with respect to the robot state and control input, defined by Eqs. 6.22–6.23, respectively.

$$\boldsymbol{\mu}_{\mathbf{X}_t} = \boldsymbol{\mu}_{\mathbf{X}_{t-1}} + \boldsymbol{\mu}_{\mathbf{u}_t} = \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{\theta}_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x_{t-1} + \rho\cos(\theta_{t-1} + \Delta\theta) \\ y_{t-1} + \rho\sin(\theta_{t-1} + \Delta\theta) \\ \theta_{t-1} + \Delta\theta \end{bmatrix} \tag{6.20}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{X}_t} = \mathbf{F_x}\boldsymbol{\Sigma}_{\mathbf{X}_{t-1}}\mathbf{F}_{\mathbf{x}}^T + \mathbf{F_u}\mathbf{Q}_t\mathbf{F}_{\mathbf{u}}^T, \bar{\boldsymbol{\Sigma}}_{\mathbf{X}_t} = \mathbf{F_x}\boldsymbol{\Sigma}_{\mathbf{X}_{t-1}\mathbf{M}}, \bar{\boldsymbol{\Sigma}}_{\mathbf{MX}_t} = \bar{\boldsymbol{\Sigma}}_{\mathbf{X}_t\mathbf{M}}^T \tag{6.21}$$

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 & -\rho\sin(\theta_{t-1} + \Delta\theta) \\ 0 & 1 & \rho\cos(\theta_{t-1} + \Delta\theta) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta y \\ 0 & 1 & \Delta x \\ 0 & 0 & 1 \end{bmatrix} \tag{6.22}$$

$$\mathbf{F_u} = \begin{bmatrix} \cos(\theta_{t-1} + \Delta\theta) & -\rho\sin(\theta_{t-1} + \Delta\theta) \\ \sin(\theta_{t-1} + \Delta\theta) & \rho\cos(\theta_{t-1} + \Delta\theta) \end{bmatrix} \tag{6.23}$$

2. State update (Correction).

For each keypoint observation $\mathbf{z}_t = \begin{bmatrix} r_t \\ \phi_t \end{bmatrix}$, data association is performed. The extracted keypoint can be associated to a landmark in the database or will be considered to be added to the database, if it is not associated to any known landmark. In the latter case, if the landmark is observed at least a specific number of times, then it will be added to the database. Assuming that the location of the observed landmark in the map is $\bar{\boldsymbol{\mu}}_{L_i} = \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix}$, the expected measurement is obtained by Eqs. 6.24–6.26.

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} \bar{x}_t - \bar{x}_i \\ \bar{y}_t - \bar{y}_i \end{bmatrix} \tag{6.24}$$

$$\lambda = \delta^T \delta \tag{6.25}$$

$$h(\bar{\boldsymbol{\mu}}_t) = \begin{bmatrix} \sqrt{\lambda} \\ \arctan\left(\frac{\delta_y}{\delta_x}\right) - \bar{\theta}_t \end{bmatrix} \tag{6.26}$$

The Jacobian of the measurement model with respect to robot state $\mathbf{H}_{\mathbf{x},t}$ is obtained by Eq. 6.27.

$$H_{x,t} = \frac{1}{\lambda} \begin{bmatrix} -\sqrt{\lambda}\delta_x & -\sqrt{\lambda}\delta_y & 0 & \underbrace{0 \; \ldots \; 0}_{2i-2} & \sqrt{\lambda}\delta_x & -\sqrt{\lambda}\delta_y & \underbrace{0 \; \ldots \; 0}_{2N-2i} \\ \delta_y & -\delta_x & -\lambda & 0 \; \ldots \; 0 & -\delta_y & \delta_x & 0 \; \ldots \; 0 \end{bmatrix} \tag{6.27}$$

The Kalman gain, and the system state and covariance are updated by Eqs. 6.28–6.30.

$$\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_{\mathbf{x},t}^T \underbrace{\left(\mathbf{H}_{\mathbf{x},t} \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_{\mathbf{x},t}^T + \mathbf{R}_t\right)}_{\text{Innovation Covariance}}{}^{-1} \tag{6.28}$$

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - h(\bar{\boldsymbol{\mu}}_t)) \tag{6.29}$$

$$\boldsymbol{\Sigma}_t = \left(I - \mathbf{K}_t \mathbf{H}_{\mathbf{x},t}\right) \bar{\boldsymbol{\Sigma}}_t \tag{6.30}$$

Landmark update (Augmentation). If a keypoint is stable enough to be added to the map, the state vector and the covariance matrix are updated to contain the new landmark according to Eq. 6.31, where $\mathbf{J}_{\mathbf{X}}$ and $\mathbf{J}_{\mathbf{z}}$ represent the Jacobians of landmark prediction with respect to robot state and measurement variables provided by Eqs. 6.32–6.33, respectively.

$$\boldsymbol{\mu}_t = \begin{bmatrix} \boldsymbol{\mu}_t \\ x_{N+1} \\ y_{N+1} \end{bmatrix}, \boldsymbol{\Sigma}_t = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{X}} & \boldsymbol{\Sigma}_{\mathbf{XM}} & \boldsymbol{\Sigma}_{\mathbf{X}}^T \mathbf{J}_{\mathbf{X}}^T \\ \boldsymbol{\Sigma}_{\mathbf{XM}}^T & \boldsymbol{\Sigma}_{\mathbf{M}} & \boldsymbol{\Sigma}_{\mathbf{XM}}^T \mathbf{J}_{\mathbf{X}}^T \\ \mathbf{J}_{\mathbf{X}} \boldsymbol{\Sigma}_{\mathbf{X}} & \mathbf{J}_{\mathbf{X}} \boldsymbol{\Sigma}_{\mathbf{XM}} & \mathbf{J}_{\mathbf{X}} \boldsymbol{\Sigma}_{\mathbf{X}}^T \mathbf{J}_{\mathbf{X}}^T + \mathbf{J}_{\mathbf{z}} \mathbf{R}_t \mathbf{J}_{\mathbf{z}}^T \end{bmatrix} \tag{6.31}$$

$$\mathbf{J}_{\mathbf{x}} = \begin{bmatrix} 1 & 0 & -\rho \sin(\theta_{t-1} + \Delta\theta) \\ 0 & 1 & \rho \sin(\theta_{t-1} + \Delta\theta) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta y \\ 0 & 1 & \Delta x \end{bmatrix} \tag{6.32}$$

$$\mathbf{J}_{\mathbf{z}} = \begin{bmatrix} \cos(\theta_{t-1} + \Delta\theta) & -\rho \sin(\theta_{t-1} + \Delta\theta) \\ \sin(\theta_{t-1} + \Delta\theta) & \rho \cos(\theta_{t-1} + \Delta\theta) \end{bmatrix} \tag{6.33}$$

Landmark extraction. A movement of the robot is considered to be two-dimensional. A good map of the environment can be obtained using both visual and laser data. The corners in laser data associated with vertical edges in the camera image are landmarks of the environment. The laser rangefinder provides accurate distance data to the edges, while accurate bearing information is computed from the images acquired by the camera. In order to extract vertical edges in the images provided by a visual sensor, the Hough transform [29] is employed to the resultant binary image of Canny edge detector [30]. The corners are extracted from raw laser information, and only the ones that can be corresponded to a vertical edge in the image are considered as map features. Since the laser rangefinder and camera are calibrated extrinsically, the vertical edges in image can be easily corresponded to the corner points in laser data.

Data Association. The data association finds correspondence between the current landmark database and the new observations. In this chapter, an observation is based on gated nearest-neighbor approach, in which each matching between sensor observations and map features is considered independently. In this approach, a correlation between measurement prediction errors is ignored. This will cause problems by accepting bad data associations in high clutter or when robot error increases. Because the map features are based on perceptions from two sensors and only corners in the laser data, which correspond to vertical edges in the camera image, are considered as the map features, it is assumed that there is not high feature clutter. This also deals with the problem of quadratic complexity of the EKF-SLAM.

The feature database is a table, in which each landmark is defined as $\mathbf{L}_i = (x_i, y_i, h_i, m_i)$, where x_i and y_i are landmark locations, parameters h_i and m_i show the number of hits and misses of each keypoint during data association. The first time a keypoint is extracted, $h_i = 1$ and $m_i = 0$. If the minimum probability is above some fixed threshold value, then the observation is considered for addition as a new landmark (h_i will be incremented). On the other hand, if a landmark is predicted to be in the field of view of the camera but is not associated with any observation, then it is missed (m_i is incremented). A keypoint that is hit more than a specific number of times will be added to the map, and a landmark in the map that is missed a specific number of times is suppressed from the map.

6.4.2 Path Planning Based on Perceived Audio Information

The robot is controlled by speech commands [21]. Each speech command is a set of isolated pre-trained words, like: "STOP MOVING, COME HERE, TURN LEFT, TURN RIGHT, STOP AT, FOLLOW ME, …" and some numbers. All of these commands are saved in a database in the memory of the computer, and for each of the commands there is a function for interpreting the corresponding command for the robot. Table 6.2 shows the list of the commands and the corresponding robot actions.

In the proposed system for audio-visual mobile robot motion control, the program for Windows Speech Recognition application is used, which is based on a Hidden Markov Model (HMM). Once a speech signal is received by the microphones, the speech-to-text program, which is being run asynchronously in parallel to the proposed system, writes the detected speech signal in a specific predefined text file and the sound source direction in another text file. This program is independent from the main system, is executed in parallel, and uses a Speech Application Programming Interface (SAPI) to write the received speech signals in a text file that is going to be read in the program of the main system. It also writes the calculated sound source location (described in Sect. 6.4.3) in another text file.

Table 6.2 List of commands and the corresponding robot actions

The command	The action
TURN LEFT	The robot turns left 90°
TURN RIGHT	The robot turns right 90°
STOP MOVING	The robot stops moving
CONTINUE x[a] METERS	The robot continues *x* m straight with its current direction
STOP AT x[a] AND y[a]	The robot moves towards the goal point (*x*,*y*) and stops there
FOLLOW WALL	The robot keeps moving parallel to the wall on its right side while keeping a distance of 0.5 m from it
COME HERE	The robot moves towards the calculated location of sound source and stops at a distance of 0.5 m from it
FOLLOW ME	The robot rotates towards the calculated sound source direction and tracks the displacements of the human detected in that direction in the camera image

[a] *x* and *y* are in m

The text file that contains information about the sound source location is only used (read) in the main program, if it is detected that the received command is “COME” or “FOLLOW ME”. Otherwise, the robot follows the received command, even if no human is present in the environment.

As is demonstrated in Fig. 6.6, the robot waits for speech input by reading the text file, in which the received command is written [21]. If the text file is not empty, the speech command-based path planner reads the text and compares it with the command words in the database. If a match is not achieved, data in the text file is deleted and the program restarts waiting for a speech input, otherwise, the function corresponding to the detected command is called in order to interpret the command for the robot. And the robot paths (waypoints) are changed so that the robot follows the received command. The new waypoints are inputted to the SLAM layer, causing the control input signals to be changed. Thus, the robot starts following the command while it is performing the SLAM. At the end, the data in the text file is

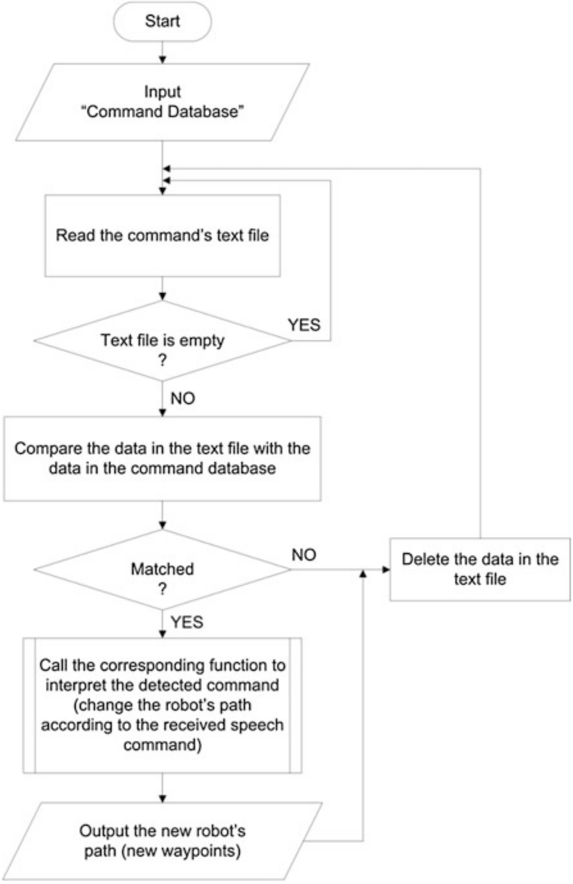

Start
Input
“Command Database”
Read the command's text file
Text file is empty
?
YES
NO
Compare the data in the text file with the data in the command database
Matched
?
NO
YES
Delete the data in the text file
Call the corresponding function to interpret the detected command (change the robot's path according to the received speech command)
Output the new robot's path (new waypoints)

Fig. 6.6 Flow chart of the proposed speech command-based path planning

deleted. In this way, always the last command is the one, which will be followed even, if the previous one is not performed completely.

In case the received command is "COME" or "FOLLOW ME", if a human is visible in the direction of the sound source in the field of view of the camera, the distance to the human is obtained from laser data, and the next control inputs (robot's new path) will be changed. Thus, the robot moves towards the sound source and stops at a distance of 50 cm from it, if the command is "COME", or tracks the detected human, if the detected command is "FOLLOW ME". If a specific person is going to command the robot, an adaptation of the method described in [31] is going to be applied. In this case, the system is trained to be able to detect a specific person by different gestures. The human body is considered to be composed of three parts: head, torso, and feet.

It is expected a human body with the specifications close to the trained ones be detected in the direction obtained from sound source localization. In order to minimize the computational complexity, only rectangles, in which the detected sound source falls in the head part of the body model, are considered as areas of interest. For each possible rectangle, a constant value is assigned, which is a function of the maximum likelihood of the three parts of that rectangle with the pre-trained model parts. If the constant value for a rectangle is greater than a threshold, that rectangle is assumed to contain a match to the speaker. The centroid of that rectangle is considered to be the center of mass of the speaker.

The decomposition of the main rectangle into three parts based on body contour model considering the pre-trained model is shown in Fig. 6.7 [31]. For each part, a value is assigned that shows the likelihood of the part to the corresponding part of the trained model. The weighted sum of the values of the three parts is the constant value assigned to the rectangle containing the detected human model (Eq. 6.34). Mobile parts of the body like hands and legs are excluded from the model.

$$K = 0.3K_1 + 0.6K_2 + 0.1K_3 \tag{6.34}$$

Once the human is detected, the robot moves towards him or follows him (depending on the received command).

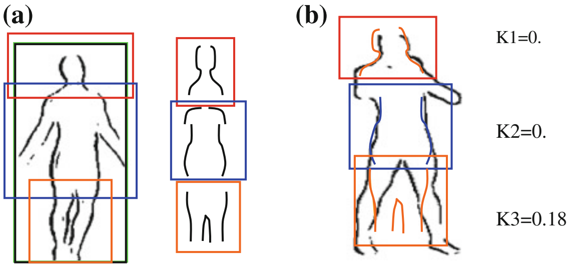


Fig. 6.7 Decomposing the rectangle around the body contour model: **a** into three parts based on the trained model, **b** assigning likelihood coefficients to each part

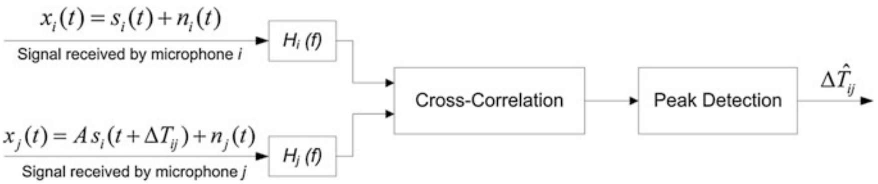

$x_i(t) = s_i(t) + n_i(t)$
Signal received by microphone i
$H_i(f)$
$x_j(t) = A s_i(t + \Delta T_{ij}) + n_j(t)$
Signal received by microphone j
$H_j(f)$
Cross-Correlation
Peak Detection
$\Delta\hat{T}_{ij}$

Fig. 6.8 The TDOA estimation

6.4.3 Audio Sensor Model, Sound Source Localization, and Speech Recognition

The sound source localization method employed for audio-visual motion control of mobile robot is based on a Time Delay Of Arrival (TDOA) estimation and is presented in Fig. 6.8. The lag time between the receptions of the signals by each of the microphones is obtained by finding the maximum of the cross-correlation of two signals received by the corresponding microphones. Then, considering that the relative geometrical position of the microphones is known, the sound source can be located.

Direction localization using microphone array. Assuming far field conditions, the TDOA approach is used to localize the direction of the sound source. The lag time between each pair of microphone is obtained by finding the peak in the cross correlation of the signals received by them [21]. With n microphones, there are (n–1) independent cross correlations. Therefore, by finding the lag time between the first microphone and all other microphones, all lag time can be calculated by Eq. 6.35.

$$\Delta T_{ij} = \Delta T_{1j} - \Delta T_{1i} \tag{6.35}$$

Value of parameter ΔT_{1j} (for $j = 2,3,\ldots,n$) is calculated using Eq. 6.36, where R_{1j} is the cross correlation between the signals received at the first microphone and the rest of the microphones, assuming that the microphones are not all positioned in the same plane (for the stability of the system of equations mentioned in Eq. 6.40).

$$\Delta T_{1j} = \arg\max_{\tau} R_{1j}(\tau) \tag{6.36}$$

Since the objective of the sound source localization in the proposed system is to localize the source of the speech command and considering that the voice signal is generally low-pass, the peaks of the cross-correlations can be very wide. This problem is solved by normalizing (whitening) the spectrum of the signals prior to computing the cross-correlation [32, 33]. Also, in order to increase the robustness of the signal to noise, more weight is given to the regions in the spectrum with higher Signal-to-Noise Ratio (SNR). Assuming that $X(k)$ is the mean power spectral density for all the microphones at a given time and that $X_n(k)$ is a noise estimate based on the time average of previous X(k), the noise masking weight is calculated by Eq. 6.37.

$$w(k) = \max\left(0.1, \frac{X(k) - \alpha X_n(k)}{X(k)}\right) \quad \alpha < 1 \tag{6.37}$$

In tonal regions of the spectrum, the SNR is very high. Thus, the contribution of the signal in tonal regions is increased using the weight function Eq. 6.38.

$$w(k) = \begin{cases} w_1(k) & \text{if } X(k) \leq X_n(k) \\ w_1(k)\left(\frac{X(k)}{X_n(k)}\right)^{\gamma} & \text{if } X(k) \leq X_n(k) \quad 0 < \gamma < 1 \end{cases} \tag{6.38}$$

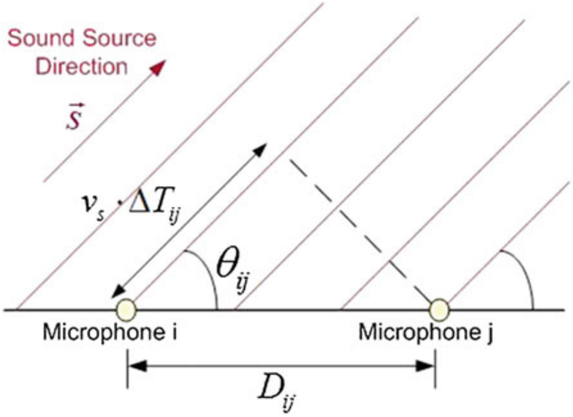
Therefore, the resulting weighted cross-correlation has a view (Eq. 6.39).

$$R_{mj}(\tau) = \sum_{k=1}^{n-1} \frac{w^2(k) X_m(k) X_j^*(k)}{|X_m(k)||X_j(k)|} e^{i2\pi k\tau/n} \tag{6.39}$$

After obtaining the lag times between all microphones, the sound source localization is achieved based on the relative geometrical positions of microphones. As is illustrated in Fig. 6.9, let $\vec{\mathbf{s}} = (u, v, w)^T$ be the sound source direction. For n microphones, each with location (x_i, y_i, z_i), one can obtain Eq. 6.40.

$$\begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ \vdots & \vdots & \vdots \\ (x_n - x_1) & (y_n - y_1) & (z_n - z_1) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} v_s.\Delta T_{12} \\ v_s.\Delta T_{13} \\ \vdots \\ v_s.\Delta T_{1n} \end{bmatrix} \tag{6.40}$$

Fig. 6.9 Sound source localization



6.5 Algorithms for Quality Estimation of Perceived Speech Information

The importance of quality estimation of perceived from the robot speech and image information can be motivated as one of the possible effective methods to increase the precision of mobile robot motion control. There are a lot of methods [34–37] and standards [38–40] for audio quality estimations. Most of them are based on subjective tests, others are objective methods and algorithms trying to obtain the precision of subjective methods. For the mobile robot speech perception quality estimation, only the objective methods and algorithms can be applied.

An important condition to choose an adequate objective algorithm is the closeness to the precision of the subjective methods widespread for speech quality estimation in communication systems [41–44]. The flow chart of algorithm corresponding to this condition is presented in Fig. 6.10 and is based on the proposed method of objective speech quality estimation replacing person as estimator in subjective methods with text to speech and speech to text methods as criterion in speech quality estimation [45, 46].

The main advantages of this proposition are the elimination of the person subjective factor in speech quality estimation process and the approach of the precision of objective speech quality methods to the higher precision of subjective methods. In Fig. 6.11, the corresponding simulation model of the mobile robot speech perception quality estimation is presented.

At the beginning of the algorithm, it is used an original text (marked as block “Original text”) from printed document, which is converted into a speech signal from a microphone connected to the mobile robot perception system (marked as block “Robot perception system”). The input speech signal is recorded as audio file (marked as block “Audio Record 1”) in the mobile robot computer perception system and simultaneously is converted into a digital text file (referred as block “Speech to Text Conversion 1”). The converted speech signal into a digital text file must be interpreted from the mobile robot as a speech command from a person speaking to the robot. In the same time the perceived from the mobile robot speech signal is reproduced by loudspeaker device (presented as “Speaker” in Fig. 6.10) and is recorded on the computer as audio file (marked as block “Audio Record 2”). In front of and nearby the loudspeaker device is placed another microphone, which receives the speech signal for conversion into a new text file (referred as block “Speech to Text 2”).

In the simulation model on Fig. 6.11, two types of possibilities to choose the source of the speech signal are presented:

- The real speech signal perceived direct from mobile robot microphone (From Audio Device).
- The speech signal converted from a speech to text system (Data Type Conversion).

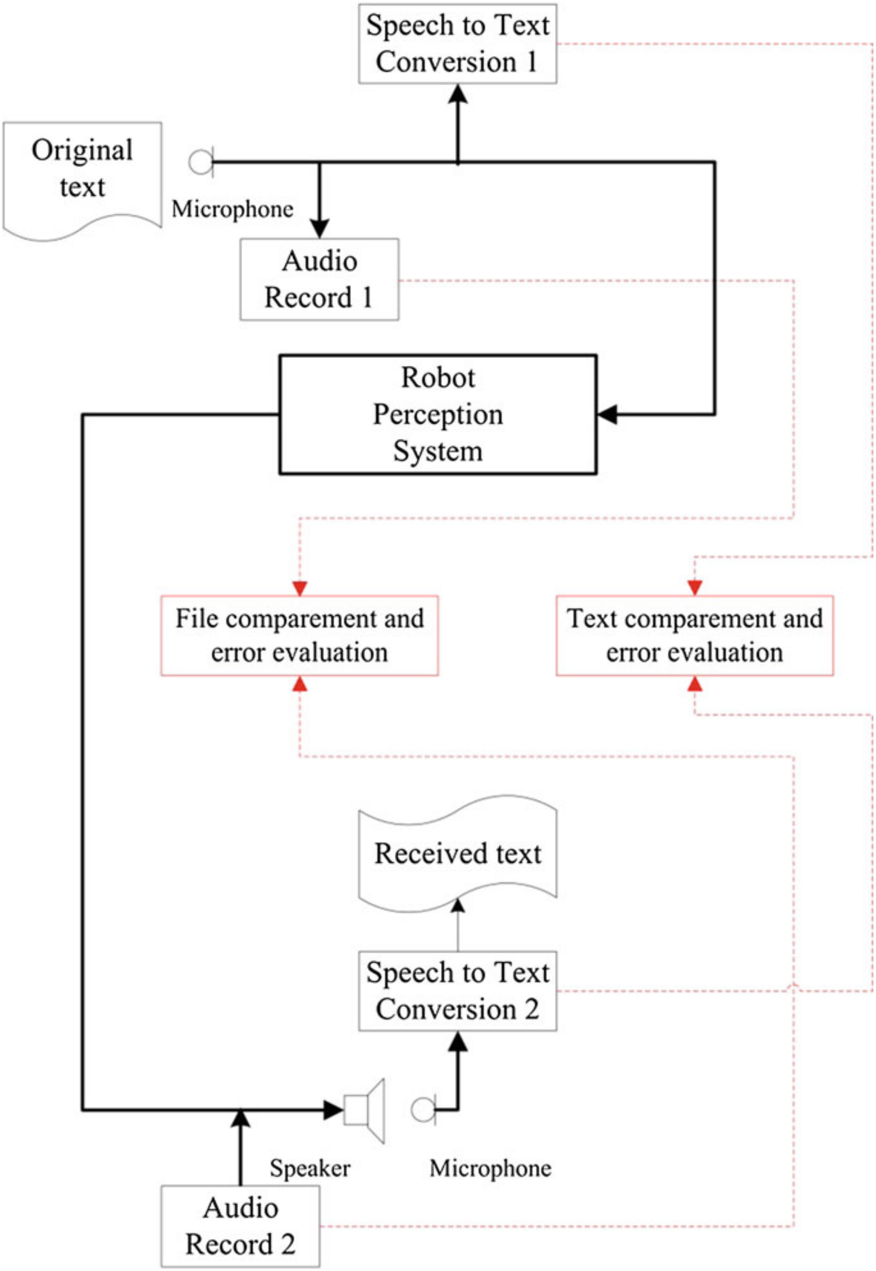

Speech to Text Conversion 1
Original text
Microphone
Audio Record 1
Robot Perception System
File comparement and error evaluation
Text comparement and error evaluation
Received text
Speech to Text Conversion 2
Speaker
Microphone
Audio Record 2

Fig. 6.10 Flow chart of objective quality estimation of speech robot perception

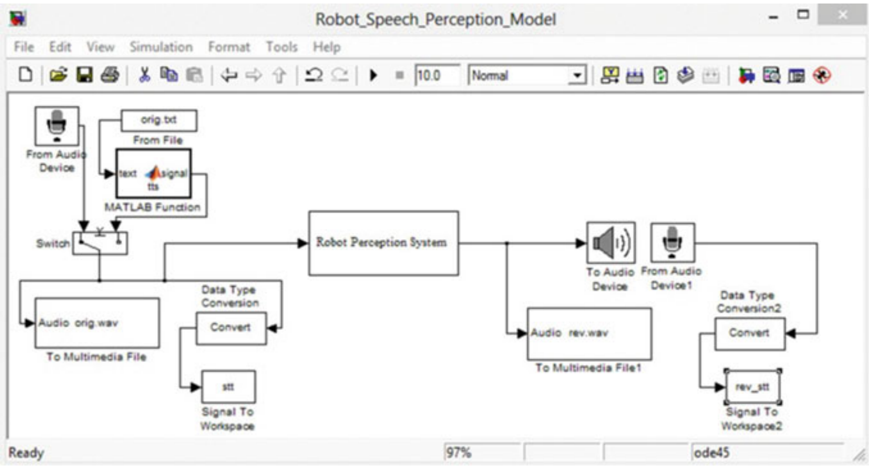


Fig. 6.11 Simulation model of audio input part with text to speech and audio output part with speech to text

The simulation model from Fig. 6.11 executes algorithm presented in Fig. 6.10 in situations, when the mobile robot receives the person speech commands. The quality estimation of speech command perception from mobile robot is prepared as the comparison (marked as block "Text compartment and error evaluation" in Fig. 6.10) and calculation of the number of incorrect received words between two text files (Fig. 6.11):

- The text document is created after direct speech to text converssion of spoken from the person words as speech comands to the robot and saved as text file "stt.txt".
- The text document is created after speech to text converssion of perceved from the robot words as speech comands and saved as text file "rev_stt.txt".

As a result of comparison, the error evaluations are used to define two types of quality assessment for mobile robot objective speech perception as speech command from a person provided by Eqs. 6.41–6.42, where $OSQE_D$ *and* $OSQE_R$ are the objective quality estimations of mobile robot speech perception defined as difference *(DNErW)* or as ratio *(RNErW)* between the number of erroneous words ($NErW_{robor}$) after speech to text converssion of perceved from the robot words as speech comand and the number of erroneous words $(NErW_{person})$ after direct speech to text converssion of spoken from person words as speech comand to the robot.

$$OSQE_D = DNErW = NErW_{person} - NErW_{robot} \tag{6.41}$$

$$OSQE_R = RNErW = NErW_{person} \big/ NErW_{robot} \tag{6.42}$$

An additional to the proposed above objective quality estimations of mobile robot speech perception is the possibility to prepare an extra or additional estimation function for more precise objective speech quality assessment of the method and algorithm presented in Fig. 6.10. This additional estimation is proposed to prepared as a possible comparison (marked as block “File compartment and error evaluation” in Fig. 6.10) between two audio records “Audio Record 1” and “Audio Record 2”:

- The original speech signal is saved as speech file “orig.wav”.
- The received speech signal is saved as speech file “rev.wav”.

6.6 Experimental Results and Discussions

Experimental results represented in this section are obtained by the robot Surveyor SRV-1 [47], which is equipped with a platform of sensors consisting of a Blackfin camera [48] and a Hokuyo URG-04LX-UG01 scanning laser rangefinder [49] (Fig. 6.12). The camera uses the OV9655 CMOS sensor [50]. The SNR of the sensor and its dynamic range (ratio between the maximum and minimum measurable light intensities) are 42 dB and 50 dB, respectively. The proposed control system for audio-visual mobile robot motion control is assumed to be applied in structured indoor environments. The main system is demonstrated by simulations in Matlab, which are based on real sensor data. In parallel with the main algorithm, Microsoft Speech API is used by a program, written in Visual Basic, for writing the received speech signals in a text file. The speech commands are detected in the main program by reading this text file and comparing it with the command database. The commands are followed by employing SLAM [21].

The control noise **Q** and the measurement noise **R** are assumed to be pointed in Eq. 6.43.

$$\mathbf{Q} = \begin{bmatrix} (0.2)^2 & 0 \\ 0 & (1^\circ)^2 \end{bmatrix} \mathbf{R} = \begin{bmatrix} (0.08)^2 & 0 \\ 0 & (1^\circ)^2 \end{bmatrix} \tag{6.43}$$

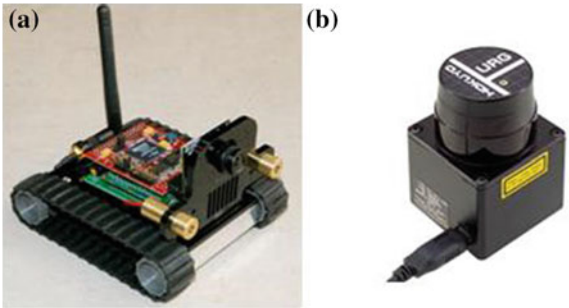


Fig. 6.12 Experimental hardware: **a** robot Surveyor SRV-1 equipped with a Blackfin Camera, **b** Hokuyo URG-04LX-UG01 Laser Range Finder

The environment of the experiments is the telecommunication laboratory (No. 1258). Measurements provide range-bearing information about the landmarks (corners) of the environment. In simulations, the frequency of control updates is 40 Hz and observations are obtained with a frequency of 5 Hz, and the robot speed and wheelbase are assumed to be 0.1 m/s and 10 cm, respectively.

The practical issues of sensor calibration are discussed in Sect. 6.6.1. Section 6.6.2 provides the data about robot navigation based on the EKF-SLAM. The experimental results from simulations of the proposed objective speech quality estimation are given in Sect. 6.6.3.

6.6.1 Sensor Calibration

In the proposed system for audio-visual motion control of mobile robots, the camera and the laser rangefinder must be calibrated extrinsically so that their relative position could be computed. This is useful for modeling the environment by corners, detected from laser data and associated with their corresponding vertical edges in the field of view of the camera.

The intrinsic parameters of the camera are obtained by geometric camera calibration. These parameters are used for compensation of lens distortions. The camera is calibrated practically using Bouguet's camera calibration toolbox [51]. A sequence of images of the chessboard pattern, captured by the camera from different positions with varying depth and angle and used for camera calibration, is shown in Fig. 6.13.

After selecting the extreme grid corners for each image and providing geometrical information about the dimensions of the grid cells (30 × 30, mm), corner extraction is performed and camera calibration parameters are computed by minimizing the least square error between the observed corner coordinates and the

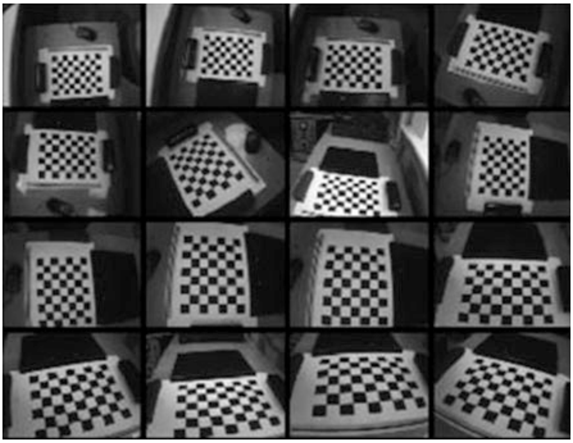


Fig. 6.13 The sequence of images of the test pattern used for geometric camera calibration

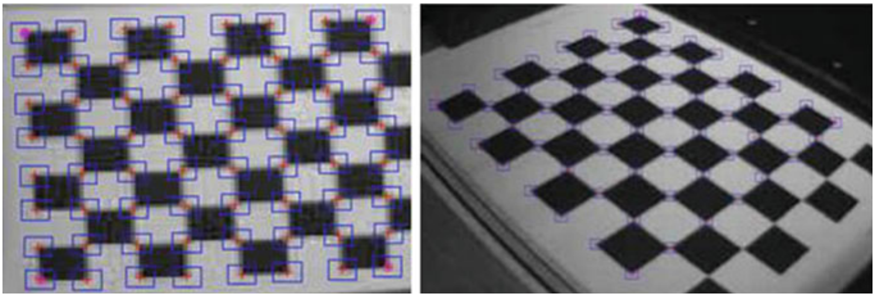


Fig. 6.14 Corner extraction performed in camera calibration

coordinates computed based on the calibration model. Figure 6.14 demonstrates the extracted corners from two of the images.

The intrinsic parameters of the camera are presented in Table 6.3 and determined by Eq. 6.44, where (f_u,f_v) is the focal length in pixels (it includes the scaling factors, too), u_0, v_0 are the coordinates of the principal point in pixels, $k_1^{(r)}, k_2^{(r)}$ present radial lens distortion coefficients and are related to the radial distortion coefficients by $k_1^{(r)} = f^3 k_1$ and $k_2^{(r)} = f^5 k_2$, $k_1^{(t)}, k_2^{(t)}$ present tangential lens distortion coefficients and are related to the tangential distortion coefficients by $k_1^{(t)} = f^2 p_1$ and $k_2^{(t)} = f^2 p_2$.

$$\left\{(f_u,f_v), (u_0, v_0), \left(k_1^{(r)}, k_2^{(r)}\right), \left(k_1^{(t)}, k_2^{(t)}\right)\right\} \tag{6.44}$$

It is visible in Table 6.3, that a lens tangential distortion is negligible. On the other hand, there is a considerable radial lens distortion, and an image correction is performed on the images captured by the camera based on coefficients of radial lens distortion.

Other useful information provided by the geometric camera calibration is that the relative position of each of the images from the sequence with regard to the local frame of the camera is obtained as extrinsic parameters of each image (Fig. 6.15).

Table 6.3 Results of geometric calibration of the camera

Parameter	Value	Standard deviation
f_u (pixels)	301.49221	1.71848
f_v (pixels)	303.21885	1.57392
u_0 (pixels)	155.45581	1.18310
v_0 (pixels)	139.29446	1.82011
$k_1^{(r)}$	−0.43039	0.00763
$k_2^{(r)}$	0.24063	0.01870
$k_1^{(t)}$	−0.00382	0.00195
$k_2^{(t)}$	−0.00297	0.00052

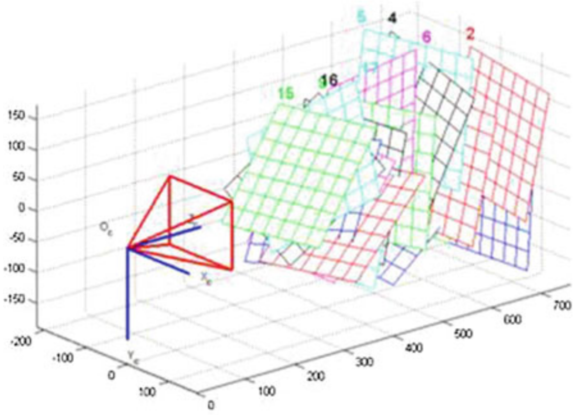


Fig. 6.15 Extrinsic parameters (camera-centered)

Therefore, considering that the position of all corners is now known with respect to the camera frame, by finding the board corners in laser scan data for each chessboard position, the laser rangefinder's relative position with respect to the camera is iteratively calculated minimizing the squared error.

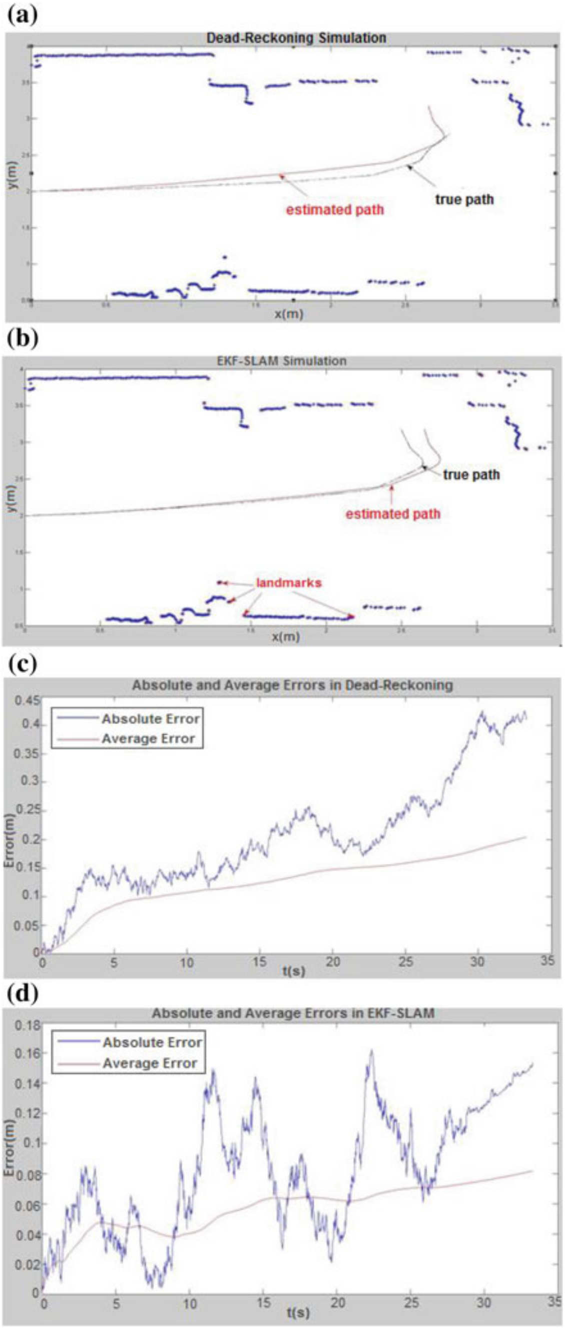
After calibration of the sensors, their systematic errors are compensated and measurements can be modeled by Gaussian distribution containing uncertainty caused by random errors. Also, assuming measurements to have Gaussian distribution in the SLAM generates results with good precision. Additionally, because of extrinsic calibration of laser rangefinder and the camera, a correspondence is found between vertical edges in the image received from the camera and corners in laser data.

6.6.2 Robot Navigation Based on EKF-SLAM

In the following simulations, the accuracy of robot localization in the proposed method, which is based on the EKF-SLAM, is compared to robot localization of a similar method, in which robot navigation is based on dead-reckoning. It is assumed that the same path is planned within the same environment, and that the robot starts from the same point in all the experiments. Each of the algorithms is applied 40 times under the same conditions, and the result with the highest average localization error (the worst result) among the 40 experiments represents each of the algorithms. Figure 6.16 demonstrates simulation results and robot localization absolute and average errors for each of the algorithms [21].

It is obvious that in mobile robot positioning based on dead-reckoning, the accuracy decreases over time. Figure 6.16c shows that positioning error is accumulated over time because random errors of proprioceptive sensor measurements accumulate and lead to incremental uncertainties in the robot position estimation over time. In about 3.2 m of robot navigation based on dead-reckoning, the average

Fig. 6.16 Accumulated positioning error: **a** dead-reckoning simulation, **b** the EKF-SLAM simulation, **c** absolute and average errors in dead-reckoning, **d** absolute and average errors in the EKF-SLAM



error relative to the traveled path is more than 6 %. The SLAM algorithms are applied to correct the high localization error in dead-reckoning based on environmental perceptions acquired from exteroceptive sensors. In the EKF-SLAM, the error is decreased to a great extent. The average error from start point until the robot reaches the goal point is reduced from 0.2 m in dead-reckoning to 0.08 m in the EKF-SLAM. In robot navigation based on the EKF-SLAM after 3.2 m of robot displacement, the average error relative to the traveled path is more than 2.5 %.

6.6.3 Experimental Results from Simulations of the Proposed Objective Speech Quality Estimation

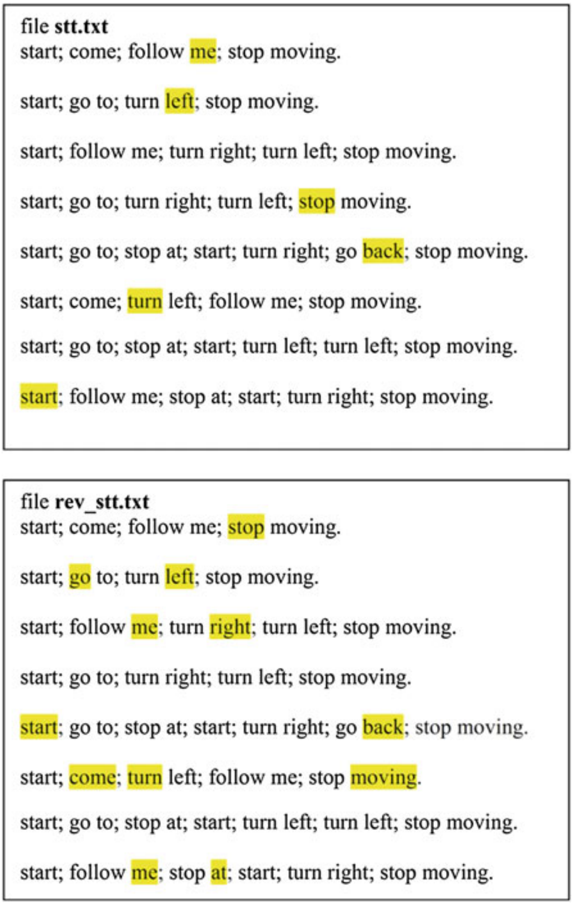
After running the simulation model presented in Fig. 6.11, the defined two types of quality assessment for mobile robot objective speech perception as speech command from a person, using Eqs. 6.41–6.42, are calculated. Some of the important results from the simulations are shown in Fig. 6.17, where are presented the results from a simple example of one of the simulations:

- A part of the text document (file "stt.txt") after direct speech to text converssion of spoken from person words as speech comands to the robot.
- A part text document (file "rev_stt.txt") after speech to text converssion of perceved from the robot words as speech comands.
- It can be seen from Fig. 6.17, that there are differences between the number of erroneous words (marked in yellow color) as speech comands to the robot in the text documents "stt.txt" and "rev_stt.txt". With this difference, the values the objective speech quality estimation for the robot speech perception are calculated by Eqs. 6.41–6.42.
- For the example in Fig. 6.17, the concrete values of *NErWrobot* and *NErWperson* are: *NErWrobot* = 12 and *NErWperson* = 6. Then with Eqs. 6.41–6.42, the following values of the objective speech quality estimation for the robot speech perception are calculated (Eqs. 6.45–6.46).

$$OSQE_D = DNErW = NErW_{person} - NErW_{robot} = 6 - 12 = -6 \tag{6.45}$$

$$OSQE_R = RNErW = \frac{NErW_{person}}{NErW_{robot}} = \frac{6}{12} = 0.5 \tag{6.46}$$

Thus, from Eqs. 6.45–6.46 it can be concluded that the calculated values of two types of the objective speech quality estimation give a quantitative objective notion for the robot speech perception useful for estimation the precision of mobile robot motion control and guidance with speech commands from a person.



file **stt.txt**

start; come; follow me; stop moving.

start; go to; turn left; stop moving.

start; follow me; turn right; turn left; stop moving.

start; go to; turn right; turn left; stop moving.

start; go to; stop at; start; turn right; go back; stop moving.

start; come; turn left; follow me; stop moving.

start; go to; stop at; start; turn left; turn left; stop moving.

start; follow me; stop at; start; turn right; stop moving.

file **rev_stt.txt**

start; come; follow me; stop moving.

start; go to; turn left; stop moving.

start; follow me; turn right; turn left; stop moving.

start; go to; turn right; turn left; stop moving.

start; go to; stop at; start; turn right; go back; stop moving.

start; come; turn left; follow me; stop moving.

start; go to; stop at; start; turn left; turn left; stop moving.

start; follow me; stop at; start; turn right; stop moving.

Fig. 6.17 Parts of the text documents stt.txt and rev_stt.txt after speech to text in transformation and the receiving parts with erroneous spoken words marked in *yellow color*

6.7 Conclusion

The proposed audio-visual perception system is used for joined audio visual mobile robot motion control employing audio-visual and range information perceived from robot sensors (microphone array, video camera, and laser rangefinder). The algorithms developed for robot motion control through speech commands and robot navigation by performing the EKF-SLAM that assumes the vertical edges as the environment landmarks are based on perceived audio information from microphone and visual and range information of camera images and 2D laser rangefinder, respectively. The way of modeling the environment as vertical edges in the camera image associated to corners in range information from the laser rangefinder has the advantage that because there is not high feature clutter, the problem of quadratic complexity of the EKF-SLAM is solved to a great extent. The navigation based on the SLAM algorithms corrects the high localization error in robot navigation based on dead-reckoning employing the environmental perceptions acquired from exteroceptive sensors. The average error relative to the traveled path is decreased from 6 to 2.5 %.

The mobile robot audio visual perception to achieve a defined motion control precision is estimated with the proposed in this chapter algorithm of objective quality estimation of speech robot perception. It is shown that the application of well known Microsoft Speech to Text and inverse Text to Speech algorithms allow to replace a person as an estimator of speech quality and bring closer the precision of objective speech quality estimation methods of mobile robot audio perception to corresponding subjective methods. It is necessary to mention that all presented in this chapter results are the subject of researches done towards two PhD thesis's: "Development of Methods and Algorithms for Audio-Visual Mobile Robot Motion Control" and "Development of Methods and Algorithms of Audio and Video Quality Estimation and Increasing in Multimedia Communication Systems" conducted at the French Language Faculty of Electrical Engineering, Technical University of Sofia, Bulgaria.

References

1. Suh II. H. Editor-in-Chief. Intelligent Service Robotics. Springer, ISSN Print: 1861-2776, ISSN Online: 1861-2784
2. Jarvis R (2008) Intelligent robotics: past, present and future. Int J Comput Sci Appl Technomathematics Res Found 5(3):23–35
3. Bittermann MS, Sariyildiz IS, Ciftcioglu Ö (2007) Visual perception in design and robotics. J Int Comput Aided Eng Inform Control Autom Robot 14(1):73–91
4. Bigun J (2006) Vision with direction. Springer, Berlin
5. Adams B, Breazeal C, Brooks RA, Scassellati B (2000) Humanoid robots: a new kind of tool. Int Syst Appl IEEE 15(4):25–31
6. Eckmiller R, Baruth O, Neumann D (2006) On human factors for interactive man-machine vision: requirements of the neural visual system to transform objects into percepts. In: IEEE world congress on computational intelligence—international joint conference on neural networks (WCCI 2006), pp 99–703
7. Demmel J, Lafferriere G, Schwartz J, Sharir M (1988) Theoretical and experimental studies using a multifinger planar manipulator. In: IEEE international conference on robotics and automation, pp 390–395
8. Murray RM, Sastry SS (1990) Grasping and manipulation using multi fingered robot hands. In: Brockett RW (ed) Robotics: proceedings of symposia in applied mathematics, American Mathematical Society, Providence, Rhode Island
9. Manocha D, Canny JF (1992) Real time inverse kinematics for general 6R manipulators. Technical report ESRC 92-2, University of California, Berkeley
10. Jarvis R, Ho N, Byrne JB (2007) Autonomous robot navigation in cyber and real worlds. In: International conference on cyberworlds (CW'2007), pp 66–73
11. Siegwart R, Nourbakhsh IR (2004) Introduction to autonomous mobile robots. a bradford book. The MIT Press Cambridge, Massachusetts
12. Adams M (1999) Sensor modelling, design and data processing for autonomous navigation. World scientific series in robotics and intelligent systems. World Scientific Publishing Co Ltd, Singapore

13. Borenstein J, Everet HR, Feng L (1996) Navigating mobile robots, systems and techniques. AK Peters Ltd, Wellesley
14. de Pina Filho AC (ed) (2010) Humanoid Robots. New Developments. Advanced Robotic Systems International and I-Tech, Vienna
15. Kuffner JJ, Nishiwaki K, Kagami S, Inaba M (2001) Motion planning for humanoid robots under obstacle and dynamic balance constraints. IEEE Int Conf Robot Autom 1:692–698
16. Favorskaya M (2012) Recognition of dynamic visual images based on group transformations. J Pattern Recognit Image Anal 22(1):180–187
17. Favorskaya M, Pyankov D, Popov A (2013) Motion Estimations based on invariant moments for frames interpolation in stereovision. Procedia Comput Sci 22:1102–1111
18. Wit C (1998) Trends in mobile robot and vehicle control. In: Siciliano B, Valavanis KP (eds) Control problems in robotics. Springer, London
19. Bekiarski A, Pleshkova S (2009) Microphone array beamforming for mobile robot. In: 8th WSEAS International Conference on Circuits, systems, electronics, control and signal processing (CSECS'2009), pp 146–149
20. Favorskaya M (2009) Detection of moving objects by using local 3D structural tensors. Vestnik of Siberian State Aerospace Univ 2:141–146 (in Russian)
21. Dehkharghani SS (2014) Development of methods and algorithms for audio-visual mobile robot motion control (Doctoral dissertation) Technical University of Sofia, Bulgaria
22. Dehkharghani SS, Pleshkova S (2014) Geometric thermal infrared camera calibration for target tracking by a mobile robot. Comptes rendus de l'Academie bulgare des Sciences 67 (1):109–114
23. Heikkila J, Olli S (1997) A four-step camera calibration procedure with implicit image correction. In: IEEE computer society conference on computer vision and pattern recognition, pp 1106–1112
24. Abdel-Aziz YI, Karara HM (1971) Direct linear transformation into object space coordinates in close-range photogrammetry. In: Symposium on close-range photogrammetry, pp 1–18
25. Levenberg K (1994) A method for the solution of certain nonlinear problems in least squares. Q Appl Math 2(2):164–168
26. Benjamin JR, Cornell CA (1970) Probability, statistics, and decision for civil engineers. McGraw-Hill, New York
27. Dehkharghani SS, Bekiarski A, Pleshkova S (2012) Application of probabilistic methods in mobile robots audio visual motion control combined with laser range finder distance measurements. In: Advances in circuits, systems, automation and mechanics, pp 91–98
28. Dehkharghani SS, Bekiarski A, Pleshkova S (2013) Method and algorithm for precise estimation of joined audio visual robot control. In: Iran's 3rd international conference on industrial automation
29. Hough PVC (1962) Method and means for recognizing complex patterns. U.S. Patent No. 3,069,654. 18.12.1962
30. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI 8(6):679–698
31. Venkov P, Bekiarski A, Dehkharghani SS, Pleshkova S (2010) Search and tracking of targets with mobile robot by using audio-visual information. In: International conference on automation and informatics (CAI'2010), pp 463–469
32. Omologo M, Svaizer P (1994) Acoustic event localization using a crosspower-spectrum phase based technique. IEEE Int Conf Acoust Speech Signal Process 2:273–276
33. Valin JM, Michaud F, Rouat J, Létourneau D (2003) Robust sound source localization using a microphone array on a mobile robot. IEEE/RSJ Int Conf Intell Robot Systs (IROS 2003) 2:1228–1233
34. Ji P, Benyuan L, Towsley D, Kurose J (2002) Modeling frame-level errors in gsm wireless channels. IEEE Globecom Internet Perform Symp 3:2483–2487
35. Mohamed S, Rubino G, Varela M (2004) A method for quantitative evolution of audio quality over packet networks and its comparison with existing techniques. In: Measurement of speech and audio quality in networks (MESAQIN'2004)

36. Hu Y, Loizou P (2006) Subjective comparison of speech enhancement algorithms. IEEE Int Conf Acoust Speech Signal Process 1:153–156
37. Kondo K (2012) Subjective quality measurement of speech: its evaluation, estimation and applications. Springer, Berlin
38. ITU-T Rec. P.862 (2001) Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs. International Telecommunication Union, Geneva, Switzerland
39. ITU-T Rec. P.835 (2003) Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm. International Telecommunication Union, Geneva, Switzerland
40. Malfait L, Berger J, Kastner M (2006) P.563-the ITU-T standard for single-ended speech quality assessment. IEEE Trans Audio Speech Lang Process 14(6):1924–1934
41. ITU-T Rec. P.800 (1996) Methods for subjective determination of transmission quality. International Telecommunication Union, Geneva, Switzerland
42. ITU-R Rec. BS.1116 (1994) Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems. International Telecommunication Union, Geneva, Switzerland
43. Thorpe L (1999) Subjective evaluation of speech compression codes and other non-linear voice-path devices for telephony applications. Int J Speech Technol 2:273–288
44. Pleshkova S, Peeva K (2013) Simulation of different types of voice communication systems used for speech quality estimation with applying speech to text as objective criterion of audio quality. Int J of Emerging Technol Comput Appl Sci Issue 2(6):107–111
45. Pleshkova S, Peeva K (2013) Application of speech to text as criterion of audio quality estimation in multimedia communication systems, Sofia, (CEMA'2013), pp 92–95
46. Pleshkova-Bekjarska S (2013) Simulation analysis of speech quality dependence from communication channel type and channel coding methods. Int J Emerging Technol Comput Appl Sci 1(6):8–12
47. Surveyor SRV-1 Blackfin Robot Surveyor Corporation. http://www.surveyor.com/SRV_info.html. Accessed 15 June 2014
48. Surveyor SRV-1 Blackfin Camera Surveyor Corporation. http://www.surveyor.com/blackfin/. Accessed 15 June 2014
49. Scanning range finder (SOKUIKI sensor): URG-04LX-UG01 Hokuyo Corporation. http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html. Accessed 15 June 2014
50. Omni Vision. http://www.surveyor.com/blackfin/OV9655-datasheet.pdf. Accessed 15 June 2014
51. Bouguet J. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed 15 June 2014

Chapter 7
Adaptive Surveillance Algorithms Based on the Situation Analysis

Nikolay Kim and Nikolay Bodunkov

Abstract One of the major trends in the development of robotic systems is the designing of methods ensuring their autonomous functioning and action planning in complex variable environment. This chapter examines the organization aspects of surveillance task calculations for automatic robotic systems such as visual navigation or search in variable and uncertain conditions. The application of methods based on the pixel-by-pixel or particular attribute comparison is hampered by their high computational load. Besides, it becomes impossible to design the reference images for uncertain surveillance conditions. The novel approach based on the complex adaptive algorithm is discussed in this chapter. The adaptive algorithm involves a range of particular surveillance algorithms, situation description, and situation analysis. The technique for generation of the descriptions of the observed scene uses the predicates and is based on the statistical methods of object recognition. Examples demonstrate the implementation of our approach in visual navigation and search of on-land mobile objects.

Keywords Computer vision · Surveillance · Situation analysis · Visual navigation · Robotics · Object recognition

7.1 Introduction

Nowadays, Autonomous Robotic Systems (ARSs), which operate in an uncertain variable environment, are becoming more common. In these conditions, the ARSs are required to plan their actions and manage the available own resources according to a Mission Task (MT). Due to the actions planning and possible strategic or

N. Kim (✉) · N. Bodunkov
Moscow Aviation Institute, 4 Volokolamskoe Shosse, Moscow 12599, Russian Federation
e-mail: nkim2011@list.ru

N. Bodunkov
e-mail: boduncov63@hotmail.com

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_7

tactical decisions, the ARS can "understand" the observed situation, i.e. establish the links (relations) between the Objects of Interest (OIs) for implementation of the MT [1–3]. For example, if a land vehicle is an object of observation, then such OIs include the roads, on which the object of surveillance may move, or obstructions on the road. The possibility of a vehicle moving on the road is the relation "vehicle—road". If the ARS task performs an observation, then the objects essential for implementation of the MT may include the objects obstructing the field of view or impacting the illumination intensity, etc.

The understanding of situation is executed through a situation analysis. A situation analysis includes the generation of the descriptions of the observed scene [2]: the presence of the OIs and relations between the objects (spatial, temporal, casual, etc.). Such descriptions will enable not only to expand the scope of the ARS-solved tasks, but also enhance the effectiveness of observations in the adverse, uncertain, and variable conditions.

In literature, one can find some papers devoting to solve the situation control problems [4, 5], which are based on the situation descriptions. However, this area of research has developed insufficiently. At present, hardware and software solutions in modern Synthetic Vision Systems (SVSs) including in the ARSs enable to solve many sub-surveillance tasks of search, detection, and identification of various OIs [6–16]. However, these papers almost fully omit the approaches for generation of scene descriptions, which restricts the use of such SVS in the ARS. Therefore, one of the important tasks of the ARS SVS is to develop the methods for understanding the observed situations in scene that will improve the efficiency of search, identification reliability of the OIs, etc.

The chapter is organized as follows. Some problems of automatic surveillance are discussed in Sect. 7.2. Section 7.3 provides a description of complex adaptive surveillance algorithm. Analysis of the observed situation is located in Sect. 7.4. Conclusion is drawn in the last Sect. 7.5.

7.2 Problems of Automatic Surveillance in Autonomous Robotic Systems

The main problems for automation of surveillance in the ARS SVS are following:

- The diversity of surveillance processes including detection, identification, and tracking.
- The difficulties in determining of operating attribute space (operating attribute vocabulary) while solving the specific surveillance tasks.
- The complexity of efficiency evaluating for various surveillance tasks.

One of the ways to solve these problems is to search for common procedural approaches for implementation of surveillance. The identification in surveillance tasks is situated in Sect. 7.2.1. The decision making based on statistical methods of identification is discussed in Sect. 7.2.2. The information description of surveillance process is provides by Sect. 7.2.3 while Sect. 7.2.4 describes a decrease of initial entropy of the OIs observation.

7.2.1 Identification in Surveillance Tasks

Let us consider how the basic surveillance can be viewed as the identification process. The decision, which is made by the SVS, is the decision about identification (a decision on whether the OI being identified belongs to a specific class of objects or classification of phenomena) or about detection/non-detection of the OI. For example, the search consists in detection of OIs through analyzing the certain space segments. It should be noted that the detection is a preliminary step for identification with only two possible outcomes being evaluated: the object is present (within the concerned area of search) or the object is absent (the dual-alternative identification). A tracking of the OI can be perceived as the series of detections of the object at each subsequent step during the surveillance.

The decision-making during identification is accomplished based on the selection and evaluation of the OI attributes and by using statistical criteria with account of possible errors and losses. The useful information capacity of the attributes (irrespective of surveillance tasks being solved), which enables to generate the operating attribute vocabulary, is evaluated using a notion of entropy.

Let us define the classes of identified object by Eq. (7.1), where M is a number of the identified classes (in partial case, with two outcomes $M = 2$ being studied for detection).

$$X = (x_1, x_2, \ldots, x_M) \tag{7.1}$$

One of the tasks solved by the identification systems is to formulate the appropriate principle for objects classification. This requires a creation of an apriori alphabet of object classes.

A set of vectors $Y = (y_1, y_2, \ldots, y_n, \ldots, y_N)$ in the attribute space are referred as vectors-implementations. The attributes are the basis of communication carrying the relevant information. There are the simple (non-derivative) and the complex attributes. The simple attributes include such characteristics as amplitude, phase, intensity, etc. in a separate digital image element. The complex attributes consist from values of criteria used in correlation-extreme systems such as an edge (outline) of an image, corners, lines, etc. In particular, this group of attributes includes structural (linguistic, syntactic) attributes. The structural attributes are based on the non-derivative elements (terminals, symbols, constants). These elements form the sentences, which describe visual objects, e.g. the sentences, which include the separate letters of an alphabet and can be presented as a set of straight-line segments with angles between them.

It should be noted, that in many applications the identification has a hierarchic nature: the simple attributes provide a possibility to identify (to distinguish) the complex attributes, the complex attributes enable to identify visual objects, and the identified objects generate the attributes for a scene description. The efficiency of the developed identification systems is considerably connected with design of the attribute vocabulary (the attribute space) necessary for description of corresponding classes of objects by using the language of these attributes. Such vocabulary includes the attributes of different types. The main idea is that the information about attributes shall be made physically available and its informative capacity shall solve the assigned tasks. Selection and evaluation of various attributes are accomplished by using the applicable algorithms.

7.2.2 Decision Making Using Statistical Methods of Identification

The most flexible approach for decision making is to evaluate the possible risks (losses, penalties). For this reason, such approach is a basic one under real conditions of the ARS operation. Risks of correct and faulty decisions are usually represented as elements of a cost matrix given in Eq. (7.2).

$$R = \left\| \begin{array}{cccc} R_{11} & R_{12} & \dots & R_{1m} \\ R_{21} & R_{22} & \dots & R_{2m} \\ \dots & \dots & \dots & \dots \\ R_{m1} & R_{m2} & \dots & R_{mm} \end{array} \right\| \tag{7.2}$$

The elements in the main diagonal of this matrix are corresponded to the identification risks obtained in the case of correct decisions. The rest of the elements define the risks of faulty decisions: the element R_{mn} determines a risk from an identification error (the object x_n is identified as the object x_m).

In statistical methods of identification, if the object is identified as x_m, the total conditional risk will make up by Eq. (7.3), and the average risk of identification is determined by Eq. (7.4), where $P(x_m/Y_n)$ is a posteriori probability of the object x_m surveillance by a condition that the attribute Y_n is obtained (observed, identified), $p(Y/x_m)$ is a conditional density of the values of the attribute Y with the object x_m as a source.

$$R_m = \sum_{m=1}^{M} P(x_n/Y_m) R_{mn} \quad n \neq m \tag{7.3}$$

$$R = \sum_{m=1}^{M} P(Y_n) \sum_{m=1}^{M} P(x_m/Y_n) R_{nm} = \sum_{m=1}^{M} P(x_m) \sum_{m=1}^{M} R_{nm} \int_{Y_n} p(Y/x_m) dY \tag{7.4}$$

$$n \neq m$$

The decisions may differ depending on the current situation, in particular, the conditions of observations, which define the limitations of the allowable risks. Thus, when various decision-making algorithms are implemented in the ARS, it is necessary to consider the peculiarities of the mission implementation and conditions of surveillance, which substantially complicate these algorithms.

Let us study some problems of decision making, when statistical methods of identification are applied [7]. These methods are universal by nature and can be used in cases with the deterministic attributes. However, this chapter does not cover the specific attributes, but only procedures of decision making at different levels of information awareness.

The decision-making for OIs detection has two states, which are possibly estimated by using the probabilities $P(x_1)$ and $P(x_2)$: the search object is situated in the observed scene (image is received) (x_1) or the object is absent (in particular cases, a background is observed) (x_2), with $P(x_1) + P(x_2) = 1$. The possible errors of detection are connected with a non-detection of the existing object or making a decision as to detection though the object is absent. The probability of the error relating to decision making as to the presence of the object herewith it being actually absent is called the probability of first order error or false alarm probability and is denoted as α. The probability of the error of target (object of detection) acquisition failure, i.e. a non-detection of the object, which is actually present, is called a second order error and is denoted as β.

The availability of two outcomes allows to simplify the procedure of decision making as compared with the multiple-choice variant of identification. When objects are detected, the decision surface divides the space of attributes Y into two sub-spaces Y_1 and Y_2, which correspond to the presence of (x_1) or absence of (x_2) of the OI in a region of search. A decision is made by defining the region, to which the attributes of the obtained images are related. The statistical connection between the obtained attribute values (vector-implementation $Y = (Y_1, Y_2)$) in the space of conditions Y and, in general case, the states x_1, x_2 are determined with the conditional densities $p(Y/x_m)$ as multivariate ones. Thus, values of parameters of the decision surface are determinative for taking a reasonable decision.

Let us consider so-called parametric method of evaluation, if the analytical form of probability density function is known. Figure 7.1 shows the probability density plots of values of object attributes $p(Y/x_1)$ and background $p(Y/x_2)$. A shape and a relative position of these plots may vary depending on surveillance conditions (dot-and-dash and solid curves $p(Y/x_1)$). The selected threshold Y_0 divides the attribute space into two parts. If the measured value Y gets into the left part, then the decision in favor of outcome x_1 is made, otherwise, the decision x_2 is selected.

In general case, the larger the distance between the densities $p(Y/x_1)$ and $p(Y/x_2)$ provides a less significant of the detection errors. In case of complete overlap of the densities, the detection (identification) cannot be implemented. The area A (under a curve $p(Y/x_1)$, at $Y > Y_0$, and above the axis Y) defines a "Target acquisition failure" error, which occurs, when the decision "Object is not detected"

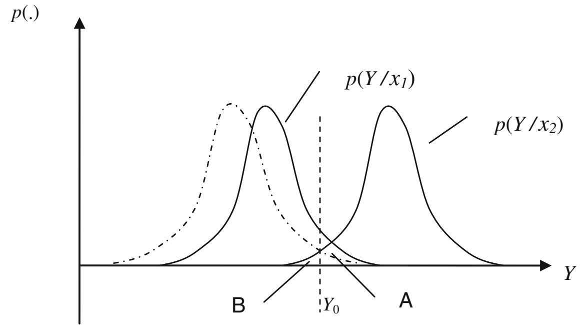


Fig. 7.1 Probability density plots of values of object attributes $p(Y/x_1)$ and background $p(Y/x_2)$

is taken. The area $B(Y < Y_0)$ (under a curve $p(Y/x_2), Y > Y_0$), corresponds to the error "False alarm" in the case of decision "Object is detected". Change of Y_0 position allows to change values of the above errors.

Depending on the accomplished mission, the requirements for limitation of detection errors are set, which determine the selection of a surveillance strategy. If it is assumed that the probability distributions are known in advance, then the methods of parametric image identification can be implemented. Let us briefly consider such methods.

Bayes Criterion of Minimum Risk. When the Bayes criterion of minimum risk is used, the losses due to detection errors are minimum. However, this approach requires maximum apriori information:

- The conditional probability density functions $p(Y/x_1)$, $p(Y/x_2)$.
- The conditional losses R_{21}, R_{12} (usually, it is assumed that R_{11} = R_{22} = 0).
- The apriori probabilities of states $P(x_1)$, $P(x_2)$, i.e. probabilities of the object presence/absent.

The minimal average risk of detection is estimated by Eq. (7.4).

The threshold value of so-called likelihood factor λ_0 is defined by Eq. (7.5).

$$\lambda_0 = \frac{R_{21}P(x_1)}{R_{12}P(x_2)} = \frac{p(Y/x_1)}{p(Y/x_2)} \tag{7.5}$$

The procedure for identification according to Bayes minimum risk criterion is as follows:

- On the basis of the received Y, the values of probability density functions $p(Y/x_1)$ and $p(\mathrm{Y}/x_2)$ are determined, and the likelihood factor is estimated by Eq. (7.6).

$$\lambda = \frac{p(Y/x_1)}{p(Y/x_2)} \tag{7.6}$$

- The parameter λ is compared with the threshold value determined in advance by Eq. (7.7). At > λ_0, the identified object relates to class 2 or to class 1.

$$\lambda_0 = \frac{R_{21}P(x_1)p(Y/x_1)}{R_{12}P(x_2)p(Y/x_2)} \tag{7.7}$$

Siegert-Kotelnikov Ideal Observer Criterion. If it is difficult to assign the values of losses, then it can be assumed that they are equal, R_{12} = R_{21}. Then the average risk of detection shall be determined by Eq. (7.8).

$$R = R_{12}\left[P(x_1)\int_{Y_1} p(Y/x_1)dY + P(x_2)\int_{Y_2} p(Y/x_2)dY\right] \tag{7.8}$$

The threshold value is calculated by Eq. (7.9), and the decision rule is executed such as at $\lambda > \lambda_0$ the object relates to class 1.

$$\lambda_0 = \frac{P(x_1)}{P(x_2)} \tag{7.9}$$

Thus, in this method the decision-making device selects the hypothesis corresponding to the maximum posterior probability.

Fisher Criterion of Maximum Likelihood. In the case, when a priori probabilities of objects presence in the observed scene $P(x_1)$, $P(x_2)$ are not known, and $R_{12} = R_{21}$, the Fisher criterion may be used. The condition $P(x_1) = P(x_2)$ is assumed, and Eq. (7.10) is executed. Thus, the decision rule has the following view: the object relates to class x_1 at $\lambda > 1$.

$$\lambda_0 = 1 \tag{7.10}$$

Minimax Criterion. This criterion is safer respect to the Fisher criterion. The first stage of implementation consists in determination of the worst distribution $P(x_1)$, at which the average risk is maximum for the fixed values of the likelihood factor λ^*. Then the value, which minimizes the risk λ_0, is sought among these distributions (Eq. 7.11).

$$R_0 = \min_{\lambda^*} \max_{P(x_1)} R[\lambda, p(x_1)] = R[\lambda_0, P_o(x_1)] \tag{7.11}$$

Neumann-Pearson Criterion. If one of the detection errors leads to significant (e.g. catastrophic) losses, the Neumann-Pearson criterion should be used. In accordance with this criterion, the requirements for the detection are represented by Eq. (7.12), where α_0, β_0 are the specified (maximum allowable) values of errors.

$$\min\alpha\,(\text{or }\min\beta),\quad \text{where}\quad \beta a \leq \beta_0\,(\text{or }\alpha \leq \alpha_0) \tag{7.12}$$

The value λ_0 is taken from Eq. (7.13) or (7.14).

$$\alpha_0 = \int_{\lambda_0}^{+\infty} p(\lambda/x_1)d\lambda \tag{7.13}$$

$$\beta_0 = \int_{-\infty}^{\lambda_0} p(\lambda/x_2)d\lambda \tag{7.14}$$

Wald Criterion of Minimum Duration of Experiment. To ensure the specified limitations of error values, a corresponding increase of vector Y length or increase of a number in the incoming surveillance readings is required. Wald has suggested a procedure, which ensures a minimum average amount of readings m to reach the specified probability of correct detection. This method is called a sequential analysis. The axis Λ is divided into three zones:

$$\begin{aligned} \Lambda_1 &\quad \text{includes all} \quad \lambda \leq \lambda_1 \\ \Lambda_2 &\quad \text{includes all} \quad \lambda \geq \lambda_1 \\ \Lambda_0 &\quad \text{includes all} \quad \lambda_1 < \lambda < \lambda_2 \end{aligned}$$

The areas Λ_1, Λ_2 are the determinate zones, Λ_0 is the domain of uncertainty.

At each ith step of the received information (readings) processing, the likelihood ratio is calculated by Eq. (7.15), which is compared with the thresholds λ_1, λ_2.

$$\lambda_i = \frac{p(Y_i/x_2)}{p(Y_i/x_1)} \tag{7.15}$$

When λ_i gets into the zone Λ_0, the decision is not made and the following $(i+1)$th reading is taken, otherwise an appropriate decision is made. This is a finite procedure, which leads to minimum average duration of an experiment, provided that the threshold values were selected in accordance with the relationships from Eq. (7.16).

$$\lambda_1 = \frac{1-\beta}{\alpha},\quad \lambda_2 = \frac{\beta}{1-\alpha} \tag{7.16}$$

While identification, if the number of identified objects' classes is $M > 2$, then the different criteria are used, in particular, criteria of minimum risk, ideal observer, and specified exceedance.

Criterion of Minimum Risk. In accordance with this criterion, the alternative hypotheses of decision making are compared considering the apriori values of losses. The difference between the obtained losses, if a decision is made to identify the k object as the object j and object i, is evaluated by Eq. (7.17).

$$\Delta R_{ij} = \sum_{k=1}^{M} \frac{\left(R_{jk} - R_{ik}\right) P(x_k) p(Y/x_k)}{\left[P(x_m) p(Y/x_m)\right]} \tag{7.17}$$

Ideal Observer Criterion. The method, which uses the ideal observer criterion, is based on the selection of the hypothesis, which corresponds to the maximum posterior probability $P(x_m/Y)$. In this case, a condition that all losses are equal is assumed, if there are errors of identification $(\Delta R_{ij} = 0)$.

Specified Exceedance Criterion. If the posterior probabilities of some hypotheses are close to the maximum one, then the ideal observer criterion may lead to a large amount of faulty decisions. In this case, the specified exceedance criterion gives better results. According to this criterion, the decision of the state of x_m is made, when Eq. (7.18) is fulfilled, where C is a specified factor of exceedance, $P(x_i/Y)$ is a probability closest to the maximum one.

$$P(x_m/Y) > CP(x_i/Y) \tag{7.18}$$

If the mentioned above condition is not fulfilled, the identification shall be continued.

7.2.3 Information Description of Surveillance Process

The complexity or even the impossibility of automatic solution in particular cases of surveillance based on the conventional methods of video processing and analysis can be connected with the following factors:

- The uncertainty of position and motion variables of the OIs.
- The uncertainty and the variability of characteristics of the observed scene.
- The distorting optical effects and the effects of abrupt change in lighting.
- The partial or the complete occlusion of the OIs.
- The variability of attributes of the OIs and other objects.

Under variable conditions of the ARS operation, the mentioned above factors can change the useful informative capacity of the various attributes. Let us notice that the use of less informative attributes reduces the efficiency of observations. In order to evaluate the useful informative capacity of attributes, the methods of information theory are applied [17]. It is assumed that investigation of the surveillance within the specified tasks will be performed on the basis of statistical theory of information. The main concepts of the information theory are as follows: information, entropy and system capacity.

In this work, a measure of information proposed by Shannon has been taken as a basis. It is estimated using Eq. (7.19), where $I(X/Y)$ is the amount of information containing in the message Y about the event X, $H(X)$ is an entropy of the event X, $H(X/Y)$ is a conditional entropy of the event X, when receiving the message Y.

$$I = I(X/Y) = H(X) - H(X/Y) \tag{7.19}$$

The priori entropy $H(X)$ is a measure of uncertainty of some event X prior to receiving the message Y. The entropy $H(X/Y)$ corresponds to the uncertainty of the event after receiving the message Y and is the posterior entropy. For informational description of continuous processes, Eq. (7.20) is used, where $p(x)$ is a density function of a random variable x, e_x is a sampling interval.

$$H(x) = -\int_{-\infty}^{+\infty} P(x)\log_2[p(x)e_x]dx \tag{7.20}$$

The sampling interval value can be determined, e.g. in accordance with the requirements for the system accuracy. For the processes with the finite number of outcomes, the entropy shall be written as Eq. (7.21), where P_m is a probability of the mth event (outcome) from the M ones possible.

$$H(x) = -\sum_{m=1}^{M} P_m \log_2 P_m \tag{7.21}$$

In the case of equally probable distribution of outcomes, the entropy value is maximum and equal to presented in Eq. (7.22), which corresponds to Hartley's measure of information.

$$H = \log_2(M) \tag{7.22}$$

It is assumed that the apriori probability P_m of its presence in each of the M cells of the search space is known prior to beginning of the OI search. Then the initial entropy of the search H_0 is calculated by Eq. (7.21) and the anticipated final entropy of the process (after completion of observations) will be obtained by Eq. (7.23), where I is a possible amount of the received useful information.

$$H_k = H_0 - I \tag{7.23}$$

It can be considered that the initial entropy H_0 gives a univocal characteristic of the amount of initial uncertainty or the volume of the task being solved. Indeed, let the anticipated distribution of the OI location over the region of search be equally probable, then Eq. (7.22) can be used. Therefore, if $M = 1000$, then $H_0 \approx 9.97$ bit, if $M = 106$, then $H_0 \approx 19.9$ bit, and if $M = 109$, then $H_0 \approx 29.9$ bit.

The final entropy H_k characterizes the uncertainty, which has been left after completion of observations, e.g. if the OI has been found in the q cell and the probability of its presence is $P_q = 1$, then according to Eq. (7.21), Eq. (7.24) is obtained.

$$H_k = 0 \tag{7.24}$$

The quality of the surveillance can be determined with the required final entropy. If the H_k exceeds the preset threshold of the H_{thr}, then the system does not fulfill the prescribed functions, e.g., when coordinates of an object are evaluated with insufficient accuracy and reliability.

If duration of observations T is known, then so-called system capacity can be calculated by Eq. (7.25), which is referred to as the informational criterion of efficiency.

$$C = \frac{(H_0 - H_k)}{T} \tag{7.25}$$

The system capacity allows to calculate not only the information capacity, which characterizes the extent of the received useful information I, but also the time required for its processing and analysis. It is evident that algorithms, which give higher values of the capacity C in comparison with other algorithms under identical conditions of observations, are more efficient. Regardless of a physical nature of message carriers and the structure of attributes, their information capacity is evaluated identically. Let us denote U_{nk} the discrete attribute, where $k = 1, 2, \ldots, K_n$ is the index of attribute value n, $n = 1, 2\ldots N$ is the index of attribute. The attribute U_{nk} may be interpreted as, e.g. a signal intensity in the nth cell of an image at K levels of quantization or as a discretized value of some criterion function.

During the identifying of M objects, if all the values of the attribute U_n are measured, in accordance with Eqs. (7.19), (7.21), and (7.23), the total of useful obtained information is equal to Eq. (7.26), where $P(x_m)$, $P(x_l)$ are the probabilities of presence of objects with m and l indices, respectively, $P(U_{nk}/x_l)$ is a conditional probability of appearance of the kth value of the nth attribute of U_{nk} in case of presence of lth object.

$$\begin{aligned} I_n &= H(x_m) - H(x_m/U_n) = H(x_m) - [H(x_m, U_n) - H(U_n)] \\ &= -\sum_{m=1}^{M} P(x_m)\log_2 P(x_m) \\ &\quad + \sum_{l=1}^{M} P(x_l) \sum_{m=1}^{M} \sum_{k=1}^{K} P(U_{nk}/x_l) P(x_m/U_{nk}) \log_2 P(x_m/U_{nk}) \end{aligned} \tag{7.26}$$

In accordance with Eq. (7.26), the capacity for the algorithm, which ensures processing of the nth attribute, can be defined by Eq. (7.27), where T_n is a time of algorithm implementation.

$$C = \frac{I_n}{T_n} \tag{7.27}$$

The time T_n may be calculated in advance or determined experimentally, taking into account the peculiarities of specific on-board computer.

In general, the information capacity of attributes is a value, which may vary in the course of surveillance. The amount of useful information I_n cannot be calculated in advance. This value is determined in the course of calculations because it depends on the values, which are taken by the measured attributes. This fact is of paramount importance as it serves as a basis for information management in adaptive algorithms. Equations (7.21), (7.23), (7.26–7.27) are the basis in the informational description of processes. It should be noted that these formulae do not define the essence of the decisions to be made. Instead, this is done using applicable detection and identification criteria described in Sect. 7.2.2.

7.2.4 Decrease of Initial Entropy of the OI Observation

The complexity of surveillance is determined to a large extent by the degree and reliability of knowledge about parameters of the conditional probability density function $p(Y/x_m)$. The uncertainties increase the initial entropy of processes and complicate a decision-making process. If available information $p(Y/x_m)$ is enough for classification of the identified objects into classes, then a creation of apriori dictionary of attributes, the determination of relations between the classes of objects and attributes, and unsupervised identification systems are used. If the analytical form of the probability density functions with some unknown but definable parameters is known, then the parametric methods are used for evaluating and decision making. In particular, under these conditions the evaluation of information capacity of attributes and statistical criteria described in Sect. 7.2.2 can be estimated by Eq. (7.26).

However, in identification practice the nature of probability density functions is often unknown in advance. Type and parameters of density depend on different factors, particularly, lighting intensity, texture, and reflective properties of a surface, etc. The methods, which provide an appropriate assessment of the probability density functions with unknown $p(Y)$, are called nonparametric.

Generally, the nonparametric methods are based on solving the problem of evaluation of probability densities, e.g. with the help of histogram method, Parzen window method, or k-nearest neighbor method, etc. These methods require preliminary investigation of an area of search. Such investigations shall provide statistical data for evaluation of the density $p(Y/x_m)$, moreover, for various classes of objects M, which cannot be done in most cases.

In real-life conditions of the ARS operation, many of surveillance tasks are solved by using nonparametric methods. Therefore, the critical task is to evaluate the conditional probability density functions $p(Y/x_m)$ on the basis of automatic analysis of situations without preliminary examination of the observed scenes.

Let a set $Q = (q_1, q_2, \ldots, q_r, \ldots, q_R)$ be a vector of surveillance conditions. Particularly, if the index r is the time of a year, then q_{rk} is the kth time of year, and $q_{(r+1)p}$ is the pth time of day, etc. Conditions of lighting, opacity, and others may be included in the list of conditions. On the set basis of the conditions of surveillance $p(Yx_m, Q_v)$, Eq. (7.28) is executed, where the index v designates some area of the observed space.

$$p(Y|x_m, Q_v) = p_T(Y|x_m, Q_v) \tag{7.28}$$

Let us study three methods of design of conditional probability density functions:

- The values of densities (for each from M objects) are determined under some averaged conditions of surveillance Q_v, e.g. the values of lighting in the morning, afternoon, evening, and night, etc. are defined four times of a year. For this purpose, the attributes of objects, which determine values of object features Y, are selected from the corresponding data and knowledge bases. Each of the relations $p_T(Y|x_m, Q_v)$ can be used in a sufficiently wide range of observing conditions, however, the accuracy of calculations will be relatively low.
- The values of densities $p(Y|x_m, Q_v) = p_p(Y|x_m, Q_v)$ are determined using general physical laws, knowledge of properties of various underlying surfaces, etc., e.g. the laws of light reflection. Theoretically, such method of determination of densities gives a rather high accuracy "in the large", i.e. in a wide range of observing conditions. On the other hand, it is difficult to take into account the specific peculiarities of variation in attributes. Thus, "in the small" the significant differences are possible between actual and calculated values of conditional probability density functions.
- The values of densities $p(Y|x_m, Q_v) = p_E(Y|x_m, Q_v)$ are defined by using the methods of artificial intelligence, in particular, neuron networks taught in advance, neuro-fuzzy systems, or production rules.

When using any of mentioned above methods, the analysis of a situation should be performed, which will help to create a description of current conditions Q_v and select (or calculate) from the knowledge data base the respective probability density function $p(Y/x_m, Q_v)$. Now the obtained value may be used to calculate information capacity according to Eq. (7.26), which ensures a selection of the most efficient attributes of the OIs. In addition, the density function is used to calculate the criteria of detection Eqs. (7.5–7.16) and identification. For implementation of algorithm, the attributes are selected on the basis of analysis of the current observing conditions.

7.3 Complex Adaptive Surveillance Algorithm

Generally, the surveillance efficiency depends on the attributes that are used for the OIs identification in specific surveillance conditions. For example, if the attributes have the insufficient information capacity or if the SVS cannot select and evaluate them within a temporal domain, then such surveillance cannot be implemented physically.

It is assumed that a distinguishing assessment of values of certain attribute types is implemented with the help of respective particular algorithms. These algorithms applied for the parameter control in the surveillance system are a crucial part of the ARS SVS search resources. The strategy of search resources management is determined with the efficiency criteria W or the quality factors of mission accomplishment.

The structure of complex adaptive surveillance algorithm is represented in Sect. 7.3.1. The correlation algorithms of information processing are situated in Sect. 7.3.2. Section 7.3.3 involves the pair criterion functions. The search of characteristic points of images is discussed in Sect. 7.3.4 briefly.

7.3.1 Structure of Complex Adaptive Surveillance Algorithm

One of the approaches that will improve the efficiency of solving surveillance tasks in an uncertain variable environment is the implementation of the adaptive image processing algorithms. The particular algorithms within such resources are selected with account of current surveillance conditions. Such particular algorithms ensure the selection and evaluation of attributes from the operating attribute vocabulary. Within this approach, if surveillance conditions change, then it is necessary to define information capacity of all attributes from the initial vocabulary. With this, it is possible to select, as an operating attribute, the most informative attribute or the attribute allowing for a maximum capacity of applicable algorithm.

Depending on conditions of the system operation and missions, the following informational factors of quality may be used:

- Minimization of final entropy is calculated by Eq. (7.29), where T is a process time, T_{rq} is a required value of the process time.

$$W_1 = \min H_k(Q) \quad \text{if} \quad T < T_{rq} \tag{7.29}$$

In special cases, this factor corresponds, for example, to maximization of probability of the correct identification or detection, minimization of the identification errors, or detection, to minimize of the identification errors or the mean-root-square error of coordinate measurement, etc., in a time-limited surveillance.

If several algorithms provide the equal time limitation, then the algorithm, which allows to obtain the maximum amount of useful information according to Eq. (7.26), shall be selected as the operating algorithm. With this, the duration T of the process of selection and evaluation of operating attributes in the algorithms being used should ensure compliance under the condition in Eq. (7.29).

- Minimization of the process time with the limited final (or current) entropy is computed by Eq. (7.30).

$$W_2 = \min T \quad \text{if} \quad T_f < T_{f.rq} \tag{7.30}$$

If the information capacity of several algorithms is enough for providing the condition according to the final entropy, then the algorithm, which can be implemented in minimum time T, is selected. The final entropy, as provided by Eq. (7.21), can define the acceptable errors in the OI coordinate evaluation, the detection errors, or the identification errors, among others.

- Maximization of the process capacity is determined by Eq. (7.31) with different limitations of the entropy or a process time.

$$W_3 = \max C \tag{7.31}$$

In this case, the algorithm with max$C$ (Eq. 7.25) is selected from the entire set of algorithms. In accordance with criteria from Eqs. (7.29–7.31), the implementation of various surveillance strategies is possible.

The algorithm, which is selected for use, should:

- Correspond to the desired efficiency criterion W.
- Ensure in the decision-making process with account of the MTs fulfillment conditions, in particular with account of acceptable risks.

As it was shown above, a compliance with criterion W is ensured through computation of information capacity of the attributes and the capacity of algorithms.

Therefore, the complex adaptive surveillance algorithm includes:

- The Database (DB) with the initial attribute vocabulary.
- The DB with the set alphabet of OIs classes.
- The DB with a set of particular attribute selectors and evaluations.
- The DB with a set of the decision-making algorithms.
- Knowledge base (KB) and/or the DB containing data for implementation of situation analysis procedure.

The complex adaptive surveillance algorithm in the ARS SVS executes the following procedures:

- The MT obtaining and the defining of the process quality factor W and the OIs (alphabet of classes).
- The analysis of current situation and the evaluation of probability density functions $p(Y|x_m, Q_v)$ (for the attributes included in the initial vocabulary).
- The evaluation information capacity of the attributes, the generation of operating attribute vocabulary, and the selection of applicable particular algorithms based on the set factor W.
- The receiving and the processing of current video information for the purpose of evaluation of attribute values.
- The making decisions about identification of the OIs [18–22].

In dependence of a decision making, the observations may be continued or ceased that makes possible to change the MT, alphabet of classes, operating attribute vocabulary, and the decision-making criteria.

Let us review some complex attribute identifiers and evaluations that can be used as particular algorithms.

7.3.2 Correlation Algorithms of Information Processing

The composition of particular algorithms may vary depending on the ARS tasks and environment [10, 12, 13]. Ones of the most effective algorithms, which provide design and assessment of complex attributes of the images observed, are the correlation algorithms.

The correlation algorithms are efficiently used for evaluation of coordinates of the OI (including navigation references), tracking and identification in different observing systems. The correlation algorithms are based on the principle of comparison of the Current Image (CI) with the Reference Image (RI), generated preliminary. The initial misalignments of the CI and the RI (due to non-coincidence of coordinates, different scale, etc.) can be defined by means of evaluation of the correlation function extremum, which corresponds to the best alignment of the CI and the RI. For images processing, it is very important to select such correlation algorithm, which defines the system noise immunity and volumes of the required calculations. If it is selected correctly, the parameters of correlation algorithm can ensure effective detection of the OIs, e.g. the landmarks with the signal-to-noise ratio.

Depending on the problems to be solved, not only cross-correlation function may be used in such correlation algorithms, but also other functions, which differ by the rules of image proximity measure calculation. These functions are called the correlation criterion functions [6]. The algorithms, which help to find the extremum of correlation criterion functions at initial displacement, not exceeding the interval (radius) of image correlation are called the algorithm with a less search.

In such algorithms, the required displacements of the CI and the RI are relatively small. Therefore, the applied correlation criterion function has one extremum, which can be searched by the gradient methods. Such surveillance conditions are encountered, for example in mobile objects tracking systems, in "secondary search" systems, etc. In this respect, it is required that one of the images will have bigger sizes. While a search of small-sized objects, the RI has smaller sizes generally, and in the case of navigation fixing of an aircraft to ground references, the CI is smaller.

If according to the observing conditions the initial displacements of the CI and the RI may exceed correlation intervals of images, then the correlation search algorithms are used. In this case, it is very difficult to evaluate displacements of the RI and the CI. When values of the correlation functions are calculated, the false extremums may appear that leads to incorrect fixing of images.

Let us assume that if the OI is detected or identified, then the attribute Y_n corresponds to the extremum of some nth correlation criterion function $K_n(di = 0, dj = 0)$, where di, dj represent misalignments of the RI and the CI. In this case, in order to make the decision in line with the criteria reviewed in Sect. 7.2.1, it is necessary to define functions $p(K_n \;/\; x_k)$. In particular, to enable detection, it is necessary to have distribution densities $p(K_n \;/\; x_1)$ for the correlation function corresponding to the OI, and $p(K_n \;/\; x_2)$ corresponding to the background (underlying surface).

Time T_n of computations K_n depends on the quantity of elementary operations R_n, that have to be accomplished for computation of the function, and the sizes of the CI and the RI being compared. For example, for searching of the landmarks by using the reference terrain map, Eq. (7.32) is obtained, where N, M are sizes of the reference image, $i_{\max}$, $j_{\max}$ are sizes of the current image (or the image fragments being compared).

$$T_n = i_{\max} j_{\max} (N - i_{\max})(M - j_{\max}) \tag{7.32}$$

Let us study the most widespread types of correlation criterion functions.

The cross-correlation function is defined by Eq. (7.33), where $S(i,j)$ is the CI, $R(i,j)$ is the RI, $M[\cdot]$ is the symbol of expectation function, i,j are coordinates of the image elements, $i_{\max}, j_{\max}$ are sizes of the compared fragment of the image.

$$K_{C_1}(di, dj) = \frac{1}{i_{\max} j_{\max}} \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} \{S(i,j) - M[S]\}\{R(i + di, j + dj) - M[R]\} \tag{7.33}$$

The normalized cross-correlation function is determined by Eq. (7.34), where $\sigma[\cdot]$ is the root-mean-square difference statement.

$$K_{C_2}(di, dj) = \frac{K_{C_1}(di, dj)}{\sigma[S] \cdot \sigma[R]} \tag{7.34}$$

In the case of complete alignment of the CI and the RI, these functions have maximum values. In the case of complete alignment of the CI and the RI, the difference criterion functions in contrast to the maximized correlation ones have the minimum value. The following functions have become the most widespread.

Mean square difference function is calculated by Eq. (7.35), where d is an index of difference criterion functions.

$$K_{d_1}(di, dj) = M\left[(S(i,j) - R(i + di, j + dj))^2\right] \tag{7.35}$$

Average difference modulus function is computed by Eq. (7.36).

$$K_{d_2}(di, dj) = M[|S(i,j) - R(i + di, j + dj)|] \tag{7.36}$$

The advantage of the difference functions as compared with the correlation ones is the absence of multiplication operations, that allows to increase the computational processes by a factor of 4 to 10. At the same time, the algorithms based on these functions have lower noise immunity.

In the case of complete alignment of the RI and the CI ($di = dj = 0$), the cross-correlation function has a maximum, and the difference function has a minimum. The difference function takes a zero value only in the case of full identity of images and absence of noise. Errors, which occur during evaluating the CI and the RI displacements, depend on the accuracy for determination of values of correlation functions and their correlation radius.

7.3.3 Pair Criterion Functions

The implementation of pair-criterion functions is connected with calculation of the amount of the CI and the RI elements, which coincide or do not coincide with respect to their intensity. In comparison to the correlation criterion functions, the paired functions demonstrate much worse interference immunity, on the one hand, and better computational cost efficiency, on the other hand. In the conditions of high contrast ratio of the OI, at $\beta \geq 10$, the paired function algorithms show the best capacity values.

Particularly, for binary images some pair criterion functions are represented in Eq. (7.37), where a, e are the number of coincident elements with intensities $(1,\ 1)$ and $(0,\ 0)$, respectively, b, c are the number of non-coincident elements with intensities $(1, 0)$ and $(0,\ 1)$, respectively, d_1 and d_2 are the number “1” in the current image and the reference image.

$$\begin{array}{ll}
\text{Rao} & Kn_1 = \dfrac{a}{(a+b+c+e)} \\[2ex]
\text{Djekard} & Kn_2 = \dfrac{a}{(a+b+c)} \\[2ex]
\text{Dake} & Kn_3 = \dfrac{2a}{(2a+b+c)} \\[2ex]
\text{Kuizinsky} & Kn_4 = \dfrac{a}{(a+b)} \\[2ex]
\text{Rogers} - \text{Tanimoto} & Kn_5 = \dfrac{a}{(d_1+d_2-a)} \\[2ex]
\text{Hamman} & Kn_6 = \dfrac{(a+e-b-c)}{(a+b+c+e)}
\end{array} \tag{7.37}$$

Separate functions provide the different weights to coincident and non-coincident elements. In particular, the Dake function gives large weight to coincident unit elements.

7.3.4 Characteristic Points of Images

The characteristic points [6, 12] or the aggregate of such points called as "constellations" can be used as one of the attributes during the OI search. The characteristic points are the points, which considerably increase the information capacity of the whole constellation. These elements are the angular points or the points with maximum values of gradient.

There are many works devoted to the procedures for determining of the characteristic points. Thus, such algorithms as Förstner and Harris algorithms, Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) detectors, and others are widely known. Thus, the discussion of methods based on the characteristic points is not actually. Their use in the OI search tasks will be considered. Figure 7.2 gives a general algorithm for design of a constellation of the characteristic points taking into account their information capacity.

The constellation is re-arranged at each cycle of the algorithm operation. In the process of constellation design, the "false" points may come as informative ones (i. e. points, which do not relate to the object itself (e.g., blinks)), which may lead to errors, when determining the object coordinates subsequently. With the reference

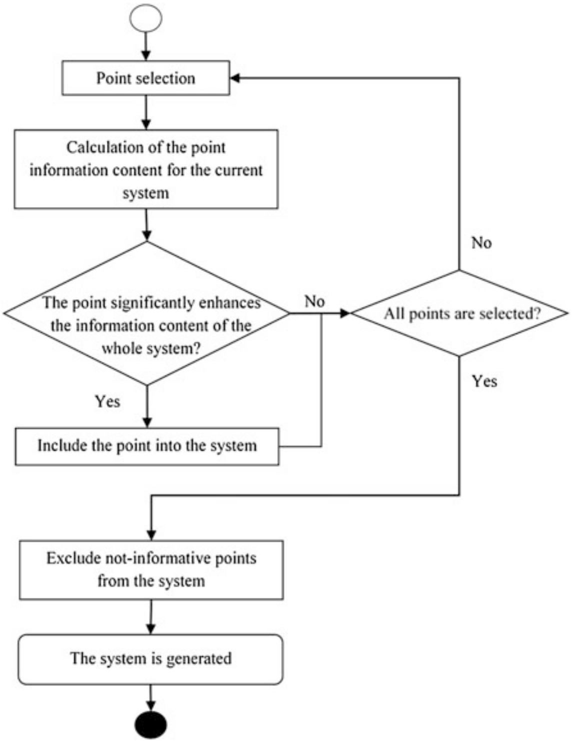

Point selection
Calculation of the point information content for the current system
The point significantly enhances the information content of the whole system?
No
Yes
Include the point into the system
All points are selected?
No
Yes
Exclude not-informative points from the system
The system is generated

Fig. 7.2 A general algorithm of a constellation design from the characteristic points

being constantly re-arranged, a filtration of the "false" points is performed. As a result, a stable constellation is formed.

The search is performed in two stages. First, the values of gradients corresponding to one of the characteristic points of the reference are searched. Second, the relative position of other points is examined. As the aspect and scale of the object may change, a "blurring" of the constellation points is performed during the search. That is why, a search of the characteristic point is performed in the vicinity of the respective reference point (Fig. 7.3).

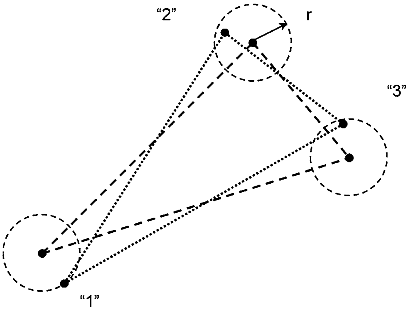


Fig. 7.3 Search of the characteristic point

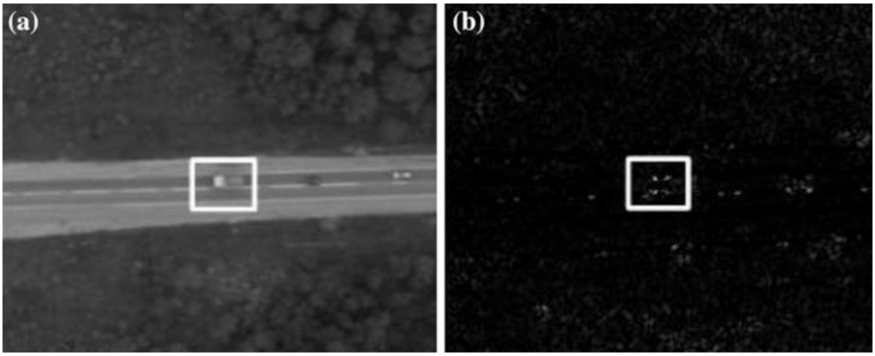


Fig. 7.4 Feature points: **a** initial image, **b** image with characteristic points

The determination of characteristic points in the car image is shown in Fig. 7.4a, b.

7.4 Analysis of the Observed Situation

The descriptions of the observed situation are created on the basis of the received (e. g. video) and available apriori information (e.g. digital map or database). For this purpose, the descriptions of various degrees of complexity depending on the mission to be accomplished are used. A creation of descriptions is preformed in two stages. At the first stage, the algorithms ensuring local detection and identification of objects are implemented [6, 7, 13]. At the next stage, the attributes of objects and the relations between them (in particular, shape, geometry, and space relations) are determined [23–27]. On the basis of the obtained information, the current description of the scene (situation) being observed is created.

In the terminology of situation control [4, 5], such description represents an enumeration of the objects observed with their attributes and relations. Such objects, their attributes, and relations in a view of predicates connected by symbols $\ni$ (utensils), $\cap$ (logical AND membership), and $\cup$ (logical OR memberships). Depending on the problem solved, the descriptions of the same scene may be different. Further, the examples of descriptions for the navigation (Sect. 7.4.1) and search tasks (Sect. 7.4.2) are given.

7.4.1 Creation of Descriptions for Navigation Tasks

In the current study, the task is to determine the ARS own position based on the analysis of the underlying surface image. Let us assume that the Unmanned Automatic Vehicle (UAV) is equipped with the SVS, whose memory contains the reference terrain map. Suppose that the evaluation precision of the UAV position

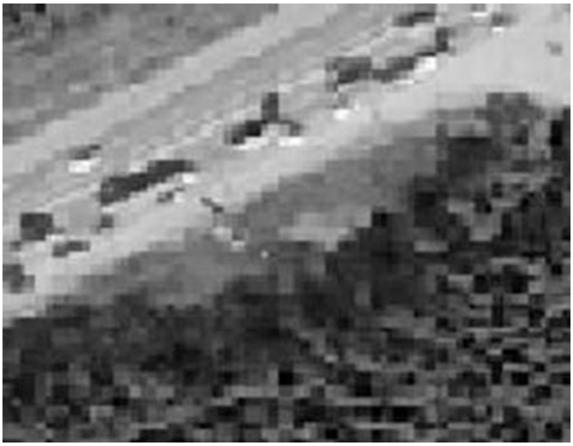


Fig. 7.5 Underlying surface image

using the airborne Navigation System (NS) is insufficient for solving the mission tasks. The NS errors may be explained by navigation sensor errors, absence of satellite navigation signals, etc. In such cases, the map-marching navigation methods are used, which are based on referencing the UAV position to the navigation landmarks through use of the SVS. Figure 7.5 shows a fragment of the underlying surface image.

In order to evaluate the UAV own position using the map-matching methods, it is necessary to do the following:

- Select and identify the OIs in the observed scene.
- Select the navigation landmarks among the identified OIs.
- Determine the coordinates of the landmarks, e.g. in geographical reference system.
- Determine the UAV coordinates.

The OIs are selected and identified using the known SVS methods. In particular, it is possible to select stationary objects that can be used as the landmarks. The road, the roadside, and the forest are identified based on their texture through selection and evaluation of the following attributes: color, brightness, and root-mean-square deviation of brightness.

For the further comparison of the reference and the current descriptions, it is necessary to design the Observed scene Description (OD). At the beginning, the identification process is implemented. Let us use the statistical method for this task. Three OIs were selected for current example ($m = 3$): wayside, road, and forest. Let us use brightness as the identification attribute. As it was shown, the conditional probability density distributions may differ for different surveillance conditions. Therefore, they should be defined for current surveillance conditions. Assume that there is a Knowledge Base (KB), containing rules to design the density distributions on the basis of the prior information about the OIs and various surveillance conditions. The density distributions for every OI are determined using the KB

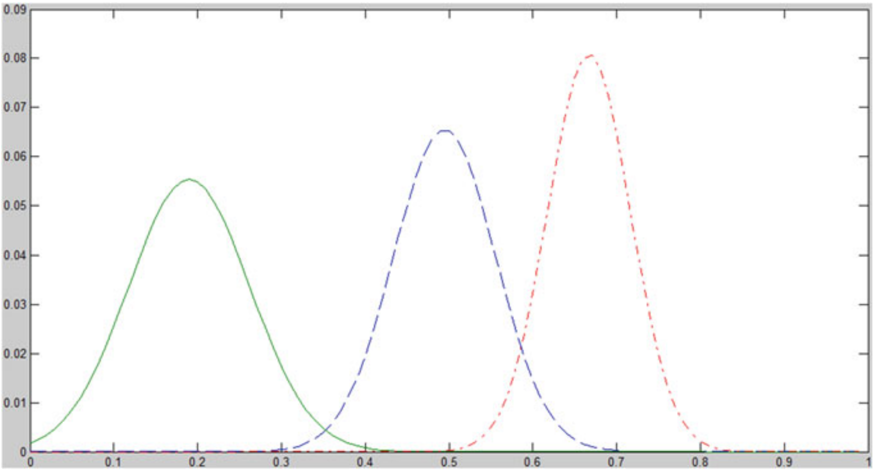


Fig. 7.6 Probability density plots of brightness for the forest (*green*), wayside (*blue*), and road (*red*)

according to current conditions, for example, clear sky, sunny, afternoon, summer in Central Russia. Figure 7.6 shows the probability density plots for objects: wayside $p(Y/x_1)$ (blue line), road $p(Y/x_2)$ (red), and forest $p(Y/x_3)$ (green).

For convenience, the recognition process will be implemented in pairs: forest-wayside, wayside-road. The comparison forest-road is not required, because their density distributions do not intersect and have no common areas of signs.

Consider the recognition task in example of pair class wayside-forest recognition. This task for two objects becomes the detection task. For the decision making let us select the Bayes criterion of minimum risk (Eq. 7.7). The threshold value is calculated by Eq. (7.38).

$$l_0 = \frac{R_{31}P(x_1)p(Y/x_1)}{R_{13}P(x_3)p(Y/x_3)} \tag{7.38}$$

The priori probabilities $P(x_m)$ and risks R_{mn} required for the criterion calculation are chosen on the basis of a prior map analysis. For example, assuming that the appearance of all landmarks on the stage has equal probability, the prior probability calculation may use the ratio of area occupied by the object (S_m) to whole map square S_0 by Eq. (7.39).

$$P(x_m) = \frac{S_m}{S_0} \tag{7.39}$$

The recognition result and, therefore, the scene description result depend on the values of the cost matrix that are selected according to the OIs informative capacity and the conditions of current mission solution. For example, Fig. 7.7 shows density distribution plots and thresholds (red line) of the forest and wayside for different cost matrix values R_{a} (case *a*) and R_{b} (case *b*) (Eq. 7.40).

$$R_a = \left\| \begin{matrix} 0 & 0.1 & 0.1 \\ 0.9 & 0 & 0.1 \\ 0.9 & 0.9 & 0 \end{matrix} \right\| \quad R_b = \left\| \begin{matrix} 0 & 0.7 & 0.1 \\ 0.3 & 0 & 0.65 \\ 0.9 & 0.35 & 0 \end{matrix} \right\| \tag{7.40}$$

Figure 7.7 shows that with the same attribute value (Y = 0.35—blue line) but different risk values the probabilities of identification result "skipping object" and "false alarm" probabilities are different. The current value of the attribute indicates the forest presence (Fig. 7.7a) and the wayside (Fig. 7.7b).

More careful selection of risk values allows more accurate identification. Figure 7.8a, b shows the scene recognition results for the proposed cost matrices R_{a} and R_{b}, respectively.

The boundaries of recognized objects in Fig. 7.8a are similar to their actual boundaries. The "wayside" object in Fig. 7.8b is poorly discerned, and its boundaries are distorted. Values of risks from "skipping object" errors grow with increasing the information capacity of the recognizing object. For example, the losses from "skipping object" errors may become critical in the case, where a "wayside" is the informative object, because its identification can significantly reduce entropy of the navigation task solution. For the selected thresholds and probability density distribution functions, there is a segment of the road with vehicles in the obtained image, and below the wayside there is a segment of forest. This scene does not allow for a highly precise determination of the UAV coordinates as the OI boundaries are linear and run parallel to each other. Therefore, the UAV position with respect to the road is uncertain. As a result, the landmarks found in the observed scene are insufficiently informative.

The action plan in such situations is as follows:

- Determine the variants of possible positions of the UAV on the map.
- Select sufficiently informative landmarks being closest to the possible positions of the UAV.
- Determine the UAV flight direction allowing detection of the landmark with the maximum probability.

The determination of the UAV position using the correlation-based comparison of the obtained current image with the reference image may be impeded in view of difference in illumination intensity, scale, and aspect angle of the images. This is why, the evaluation procedure of the UAV position is accomplished by comparing the descriptions rather than the images.

Let us assume that the description of the reference terrain map was generated in advance. The description of this scene is provided by Eq. (7.41), where:

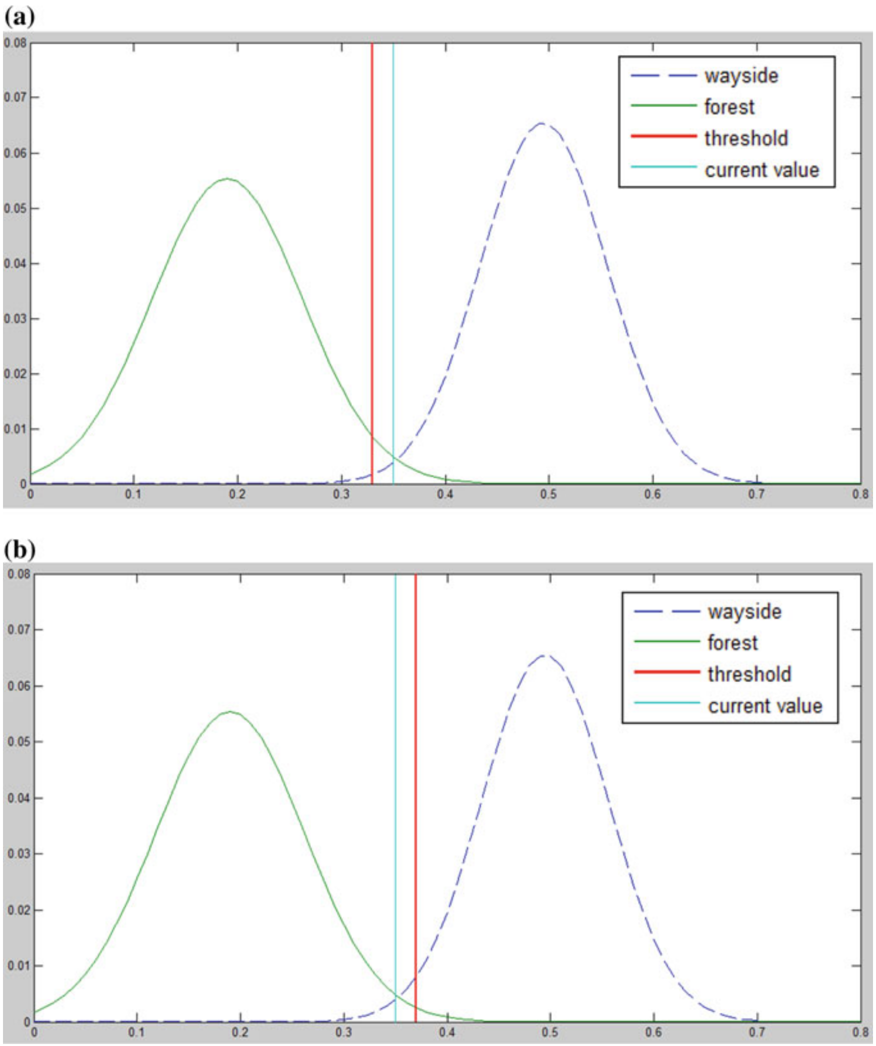


Fig. 7.7 Density distribution plots and thresholds for different cost matrix values: **a** for matrix R_a, **b** for matrix R_b

- «1» is an index of object in the description.
- O is a class of the object.
- S is a type of the surface.
- G is a geometric description of the object.
- K is the coordinates of the object's center of mass.
- M is the inertia moments of the object.
- L is a number of lines.

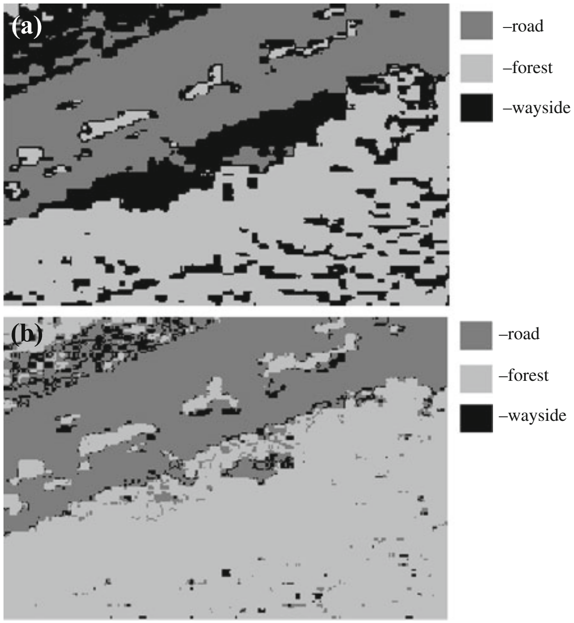


Fig. 7.8 Identification results using different cost matrix: **a** for matrix R_a, **b** for matrix R_b

$$
\begin{array}{l}
H(\\
\quad ``1''(O(4)\cap S(3)\cap G(K[x126y109]\cap M[Jx:7041281Jy:24019394]))\cap\\
\quad ``2''(O(3)\cap S(4)\cap G(K[x157y100]\cap M[Jx:5963281Jy:21718900]))\cap\\
\quad ``3''(O(4)\cap S(3)\cap G(K[x124y73]\cap M[Jx:11503574Jy:36472907]))\cap\\
\quad ``4''(O(1)\cap S(1)\cap G([X_1Y_1,..X_nY_n],L(2)))\cap\\
\quad ``5''(O(1)\cap S(1)\cap G([X_1Y_1,..X_nY_n],L(3)))\cap\\
\quad ``6''(O(3)\cap S(3)\cap G(K[x135y140]\cap M[Jx:2827345Jy:6941276]))\\
)
\end{array}
\tag{7.41}
$$

The predicate $O(\cdot)$ indicates the class of stationary object. Only stationary objects are studied in this description. The class of the object is given in brackets: 1 is a road, 2 is a roadside, 3 is a meadow, 4 is a forest.

The geometrical descriptions of objects are specified with the help of predicate G. For various classes of objects the descriptions may be different, e.g. the image of a road may be described with an aggregate of nodal points ($[X_1Y_1,..X_nY_n]$) and a number of lines $L(2)$ (two lines), and for such objects as field or forest, coordinates of mass center $K[X\,Y]$ and inertia moments $M[Jx\,Jy]$ of their images can be specified. For

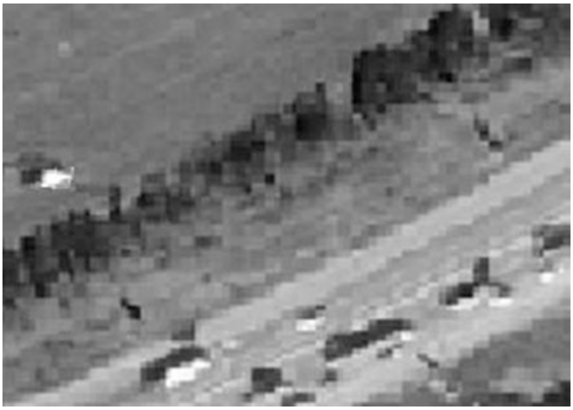


Fig. 7.9 Next frame of the surface

stationary objects, the type of surface should be also specified, e.g. the predicate value S(·) for the road may be: 1 is the asphalt, 2 is the concrete, 3 is the ground, etc.

Let us assume that a comparison of the current scene and the reference description of the general map of the zone of interest stored in the memory of the UAV has shown that there are several areas (r = 1, 2,.., R) similar to the received image. In the terminology of situation control, this sentence may be written down as a location predicate in Eq. (7.42), where P_r values are the probabilities of presence in the rth area.

$$((\ni [x_1, y_1] \cap P_1) \cup (\ni [x_2, y_2] \cap P_2) \cup \ldots) \tag{7.42}$$

Then according to the descriptions of each R area, the possible location of objects, which ensure more precise determination of the coordinates, is defined, and a further strategy of the ARS actions is determined (which objects to search, which algorithms of image processing to use, which type of control to implement, etc.).

Figure 7.9 shows a segment of a terrain, when the UAV is flying north-west. The left part of the Fig. 7.9 shows an isolated stationary object, which allows for a sufficiently precise determination of the UAV coordinates. This kind of landmarks provides the positioning of the small UAVs with the root-mean-square error σ ≤ 1 m.

Thus, the use of the description, generated thought the analysis of the observed scene, enables to solve the navigation task with insufficiently informative landmarks.

7.4.2 Creation of Descriptions for Search Tasks

Another task, which may require a creation of situation description, is a search of mobile objects. This description shall help in defining the area, in which the UAV is most likely to meet with the OI. In this case, the mentioned above rules, which describe only spatial and geometric relations between the objects, are not enough. It is desirable to introduce the resources, which allow to consider the target function of the object, its dynamic possibilities, peculiarities of behavior, etc.

The example of a scene description with a mobile object is provides by Eq. (7.43), where:

- $O(\cdot)$ is a class of a stationary object (road, field, forest, car, etc.).
- *OM* is a class of a mobile object.
- *Phf* is a description of the object physical properties.
- *Mf* is the properties of mobile objects.
- *Pass* is a passibility.
- *Vmax/min* is maximum or minimum speed of the object.
- *Tf* is a description of the object's target function (*Vav* is the average speed).
- *Tar* is a target object (direction, object, city).

$$\begin{aligned}
&H(\\
&\quad {}^{``}1^{''}(O(1)\cap G([X_1Y_1,..X_nY_n],L(2))\cap S(1)\cap Phf(..))\cap\\
&\quad {}^{``}2^{''}(O(1)\cap G([X_1Y_1,..X_nY_n],L(3))\cap S(1)\cap Phf(..))\cap\\
&\quad {}^{``}3^{''}(OM(2)\cap G([X_1Y_1,..X_nY_n])\cap Mf(Pass(1)^{\wedge}V\max(..)\\
&\qquad\qquad\qquad \cap V\min(..)..)\cap Tf[1](V(..)\cap T\arg({}^{``}N^{''})))\\
&\quad)
\end{aligned}\tag{7.43}$$

The predicate $OM(\cdot)$ indicates the class of mobile objects. Similar to the predicate $OM(\cdot)$, the object class is assigned with the index in brackets after the predicate: 1 is a person, 2 is a car, etc.

Depending from the mission to be accomplished, the description of objects (stationary or mobile) may be extended by descriptions of object physical properties with main physical characteristics and correlations required for accomplishment of a mission being recorded within the predicate $Phf(\cdot)$. It should be noted that physical description of object may change and extend with the course of time. The physical characteristic of the object "road" may be a friction coefficient, which changes depending on the environment temperature and precipitates.

For description of specific properties of mobile objects, the predicates $Mf(\cdot)$ and $Tf(\cdot)$ are used. The predicate $Mf(\cdot)$ indicates general properties of a mobile object, e.g. minimum and maximum speed and acceleration (*V*min/max) or passibility (*Pass*). The predicate $Tf(\cdot)$ describes the supposed target functions of the object. The type of the target function is given in the square brackets after the predicate $Tf(\cdot)$, e.g. *Tf*[1] (*Targ* (${}^{``}N^{''}$)) will indicate the minimum time, and *Tf*[2] will show the maximum of stealthiness. Several target functions may be assigned to one object: "*Go to city N*" ($Tf[1](V(..)\cap T\arg({}^{``}N^{''}))$) and "*Travel as invisible as possible*" ($Tf[2]\ldots(..)$).

A convenient tool for description of behavioral attributes or target functions are the production systems. In these systems, the knowledge (e.g. experts' knowledge or obtained knowledge as a result of simulation) about probable results and preferable strategies of solving problem is represented in a view of production rules Eq. (7.44), where j is an individual number of the production, which distinguishes it from other productions of the system, S is a description of the situation class, in which this production is used, L is a condition of actualization of the production, A, B are left and right parts of the production, respectively, Q is an instruction introduced after implementation of this production.

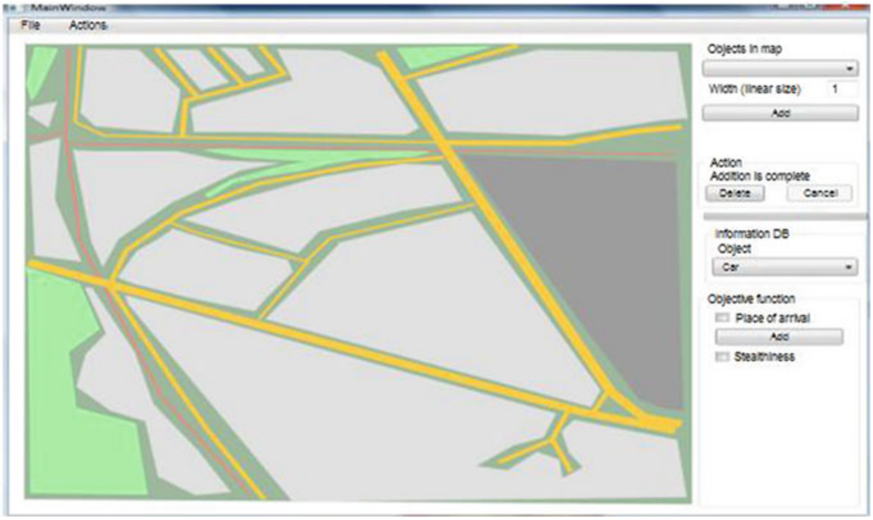
$$(j)S; L; A \Rightarrow B; Q \tag{7.44}$$

For example, let A be "the object moves from the point (x_1, y_1) to the point (x_2, y_2)" and B—"the object must travel "northward". This is described by Eqs. (7.45–7.46).

$$([(1)S; L1; A1 \Rightarrow B1; Q1]O \supset [x_1, y_1]^{P1}) \tag{7.45}$$

$$(\cup([(2)S; L2; A2 \Rightarrow B2; Q1]O \ni [x_2, y_2] \cap P2) \cup \ldots) \tag{7.46}$$

The scope of the problem being solved is characterized with the initial entropy of a process. Regards the tasks relating to search of objects, a decrease of the initial entropy corresponds to the reduction of the area of search and, consequently decrease of the process duration, e.g. if there is some apriori information about the OI location or peculiarities of its movement, then the probability of its presence within the area of search may be defined more exactly.



Width (linear size) 1

Action
Addition is complete

Object
Car

□ Place of arrival

□ Stealthiness

Fig. 7.10 Fragment of the map

Figure 7.10 shows a fragment of the map, which represents a space of search for a ground mobile OI.

It is assumed that the apriori distribution of the OI appearance in the search zone is equally probable, and the OI is moving from north to south. The initial area of search is approximately 1 sq. km or 10^6 m^2, which corresponds to (with the required accuracy of 1 m^2) the initial entropy calculated by Eq. (7.21), $H_0 = 19.9$ bit. For decrease of the initial entropy of search on the basis of situation analysis, the possible strategies of the OI behavior were studied. Particularly, a decrease of the initial zone of search may be reached by analyzing the following issues:

- Supposed directions or terminal point of movement.
- Possible requirements for speed rate of the OI.
- Possible requirements for movement security (stealthiness).
- Behavior attributes of the OI driver, etc.

Figure 7.11 shows the probabilities of the object appearance in different parts of the road within the zone of interest. The area of the road intervals, where the OI may appear made up about 2 × 10^4 m^2, which was approximately 50 times less than the total area of search, and the maximum initial entropy became $H_0 = 14.3$ bit.

The probabilities of the object appearance, in the case, when a “stealthiness” of movement is required, are shown in Fig. 7.12.

In this case, it is supposed that the OI can:

- Select the road intervals with heavy traffic to hide itself in a stream of other vehicles.
- Select the roads surveillance, of which is impaired, etc.

The total area of such road intervals made up 3 × 10^4 m^2, and the maximum initial entropy increased up to $H_0 = 15$ bit. Therefore, according to the calculations, the data obtained with the help of analysis of situations allow to reduce the area of search considerably and increase the management efficiency of search resources.

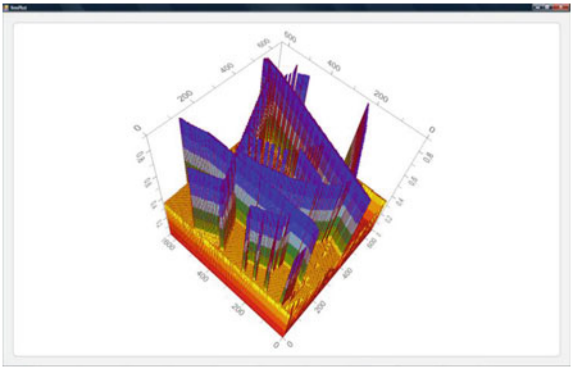


Fig. 7.11 Probabilities of the object appearance in different parts of the road

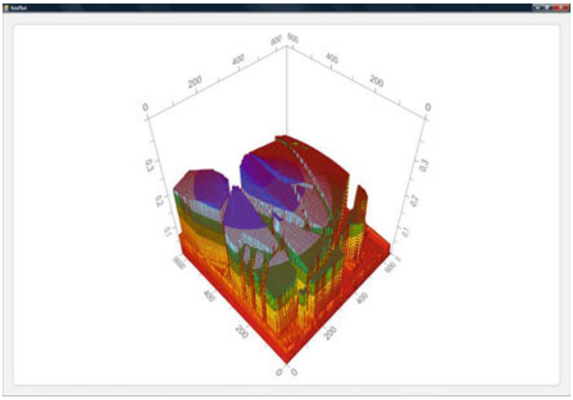


Fig. 7.12 Probabilities of the object appearance, in the case, when a "stealthiness" of movement is required

7.5 Conclusion

The chapter provides a novel approach for increasing of the computational performance and the reliability of the reference image and the current image comparison during visual navigation and search under uncertain and variable conditions. The approach is based on using the descriptions of the observed scenes rather than the pixel-by-pixel image comparison and provides that different tasks may employ different descriptions, reflecting the relevant properties of the described objects. The proposed integrated adaptive image processing and analysis algorithm involves a set of particular algorithms of observation, situation analysis, and knowledge bases, which helps to design and to compare the scene's descriptions. It is shown that depending on the mission task and the current situation, the particular surveillance algorithms and attributes have different efficiency. The proposed approach ensures the adaptation of surveillance algorithm to variable conditions. Examples of using the adaptive algorithm for the purposes of navigation and search of mobile objects are also provided.

The further task of the research is the development of methods for estimation the probability density functions of attributes during the nonparametric detection and recognition of objects of interest under various surveillance conditions.

References

1. Tulum K, Durak U, Yder SK (2009) Situation aware UAV mission route planning. IEEE Aerospace conference, pp 1–12
2. Kim N (2012) Using methods of situation analysis solving the target tasks of unmanned aerial. In: Conference of technical vision in control systems (TVCS'2012), pp 10–14 (in Russian)
3. Kim N, Kuzneczov A, Krylov I (2010) Application of machine vision systems for unmanned aerial vehicles in the problems of terrain orientation. Vestnik MAI 17(3) (in Russian)
4. Osipov GS, Smirnov IV, Tikhomirov IA (2012) Formal methods of situational analysis: experience from their use. Autom Doc Math Linguist 46(5):183–194. ACM Press

5. Klimova SG, Mikheyenkova MA (2012) Formal methods of situational analysis: experience from their use. Autom Doc Mathe Linguist 46(5):183–194
6. Vizilter U, Zheltov S, Bondarenko A, Ososkov M, Morshin A (2010) Image processing and analysis in machine vision tasks. Fizmat, Moscow (In Russian)
7. Forsyth D, Ponce J (2004) Computer vision: a modern approach, 2nd edn. Prentice Hall, New Jersey
8. Lin F, Lum KY, Chen BM, Lee TH (2009) Development of a vision-based ground target detection and tracking system for a small unmanned helicopter. Sci China Series F: Inf Sci 52 (11):2201–2215
9. Bethke B, Valenti M, How J (2007) Cooperative vision based estimation and tracking using multiple UAVs. In: Pardalos PM, Murphey R, Grundel D, Hirsch MJ (eds) Advances in cooperative control and optimization LNCIS 369:179–189
10. Sigal L, Zhu Y, Comaniciu D, Black M (2007) Tracking complex objects using graphical object models. In: Jähne B, Mester R, Barth E, Scharr H (eds) Complex motion. LNCS 3417 223–234
11. Cesetti A, Frontoni E, Mancini A, Zingaretti P, Longhi S (2010) A vision-based guidance system for UAV navigation and safe landing using natural landmarks. J Intell Robot Syst 57 (1–4):233–257
12. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. J ACM Comput Surv CSUR 38 (4):article 13
13. Li X, Hu W, Shen C, Zhang Z, Dick A, Hengel AVD (2013) A survey of appearance models in visual object tracking. J ACM Trans Intell Syst Technol 4(4):article 58
14. Ulman S (1996) High-level vision: object recognition and visual cognition. MIT Press, Bradford
15. Ulman S, Basri R (1991) Recognition by linear combination of models. IEEE Trans Pattern Anal Mach Intell 14(2):157–173
16. Shum H, Ikeuchi K, Reddy R (1995) Principal component analysis with missing data and its application to polyhedral object modeling. IEEE Trans Pattern Anal Mach Intell 17 (9):854–867
17. Temnikov F (1979) Fundamentals of information technology. Energya, Moscow (In Russian)
18. Valavanis KP, Kokkinaki AI, Tzafestas SG (1994) Knowledge-based (expert) systems in engineering applications: a survey. J Intell Robot Syst 10(2):113–145
19. Dietterich TG, Bariki G (1995) Solving multiclass learning problems via error correcting output codes. J Artif Intell Res 2:263–286
20. Dietterich TG (2002) Ensemble learning. The handbook of brain theory and natural network. MIT Press, Cambridge
21. Forman G (2006) Tackling concept drift by temporal inductive transfer. In: 29th annual Int ACM SIGIR conference on research and development in information retrieval, pp 252–259
22. Forman G (2004) A pitfall and solution in multi-class feature selection for text classification. In: 21st international conference on machine learning (ICML'2004), pp 38–45
23. Browne GJ, Pitts MG, Wetherbe JC (2007) Cognitive stopping rules for terminating information search in online tasks. MIS Q ACM Press 31(1):89–104
24. Anagnostopoulos A, Broder AZ, Punera K (2006) Effective and efficient classification on a search-engine model. 15th Conf on Information and Knowledge Management (CIKM'2006) 208–217
25. Callahan SM, Voland G (1995) Extracting knowledge from examples: Induction of heuristic rules for wheelchair prescription. J Intell Robot Syst 14(2):133–153
26. Leishman RC, McLain TW, Beard RW (2014) Relative navigation approach for vision-based aerial GPS-denied navigation. J Intell Robot Syst 74(1–2):97–111
27. Basili M (2004) Complex linguistic features for text classification: a comprehensive study. In: 26th European conference on information retrieval (ECIR'2004), pp 181–196

Chapter 8
Enhanced, Synthetic and Combined Vision Technologies for Civil Aviation

Oleg Vygolov and Sergey Zheltov

Abstract The perspective of aviation safety improvement is closely tied with the development of novel avionics solutions, aimed to enhance a flight visibility and a situation awareness of a flight crew. Such solutions include Enhanced Vision System (EVS), Synthetic Vision System (SVS), and Combined Vision System (CVS). These systems provide a supplemental view of external cabin space for a flight crew using technical vision, computer graphics, and augmented reality. The chapter addresses the general principles of the EVS/SVS/CVS development and proposes a number of original methods and algorithms for image enhancement, TV and infrared (IR) image fusion, vision based runway and obstacle detection, the SVS image creation, the EVS/SVS image fusion.

Keywords Enhanced vision · Synthetic vision · Combined vision · Image fusion · Mathematical morphology · Image processing · 3D image · Digital terrain model · Augmented reality

8.1 Introduction

Poor visibility leading to a lack of visual contact with key topographic features is one of the major contributors to worldwide civil aviation fatal accidents during approach and landing procedures [1]. At recent years, due to the increasing performance of the on-board computers in 2D and 3D visual data processing, the perspective for improvement of aviation safety is closely tied with the development of novel avionics solutions aimed to enhance a flight visibility and a situation

O. Vygolov (✉) · S. Zheltov
The Federal State Unitary Enterprise "State Research Institute of Aviation Systems",
7, Viktorenko Str., Moscow 125319, Russian Federation
e-mail: o.vygolov@gosniias.ru

S. Zheltov
e-mail: zhl@gosniias.ru

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_8

awareness of a flight crew using technical vision, computer graphics, and augmented reality. Such solutions include in the EVS/SVS/CVS, which provide a supplemental out the window view based on information from optical sensors and geospatial databases.

The chapter describes the current results of a Research and Development (R&D) project aimed to create an advanced prototype of an Enhanced and Synthetic Vision System (ESVS). It addresses such aspects of the ESVS prototype development as experimental data acquisition, mathematical and computer simulation, development of methods, algorithms, and software tools.

The rest of the chapter is organized as follows. Section 8.2 gives an EVS/SVS/CVS survey including the main regulatory documents, generic functional requirements, and general architecture. In Sect. 8.3, a short overview of commercial EVS/SVS/CVS systems and references to R&D projects are represented. The general principles of the ESVS prototype development are proposed in Sect. 8.4. Section 8.5 briefly describes the ESVS hardware components and platform. In Sect. 8.6, the following image processing algorithms for enhanced and synthetic vision support are proposed such as an image enhancement, the TV and IR images fusion, a vision based runway and obstacle detection. Section 8.7 describes the prototype of the SVS function. In Sect. 8.8, the combined vision algorithm based on photogrammetric approach is proposed. Conclusion is given in Sect. 8.9.

8.2 EVS/SVS/CVS Survey

The list of main regulatory documents for the EVS/SVS/CVS is discussed in Sect. 8.2.1. The overview for enhanced vision, synthetic vision, and combined vision system systems is given in Sects. 8.2.2–8.2.4, respectively.

8.2.1 The Main Regulatory Documents

The development of technical requirements to the minimum specifications of the certified EVS/SVS/CVS is the responsibility of Radio Technical Commission for Aeronautics (RTCA Inc., USA) [2] and The EURopean Organization for Civil Aviation Equipment commission (EUROCAE, EU) [3]. Basic RTCA regulatory documents for the design of the EVS/SVS/CVS are mentioned below:

- RTCA DO-315. Minimum Aviation System Performance Standard (MASPS) for Enhanced Vision Systems, Synthetic Vision Systems, Combine Vision Systems and Enhanced Flight Vision Systems [4].
- RTCA DO-341. Minimum Aviation System Performance Standards (MASPS) for an Enhanced Flight Vision System to Enable All-Weather Approach, Landing and Roll-Out to a Safe Taxi Speed [5].

RTCA DO-315 was initially approved in December, 2008 and has two revisions: "A" (July, 2010) and "B" (June, 2011). RTCA DO-341 was approved in September, 2012. European analogues of RTCA MASPS for EVS/SVS/CVS are ED-179 (December, 2008) and ED-179B (September, 2011) [6].

Also the following documents are used in developing requirements for certified systems:

- DO-160/ED-14. Environmental Conditions and Test Procedures for Airborne Equipment.
- DO-254/ED-80. Design Assurance Guidance for Airborne Electronic Hardware.
- ARINC 653. Avionics Application Software Standard Interface.
- ARINC 818. Avionics Digital Video Interface.
- DO-178/ED-12. Software Considerations in Airborne Systems and Equipment Certification.
- DO-276. User Requirements for Terrain and Obstacle Data.
- DO-200. Standards for Processing Aeronautical Data.
- ARINC 762-1. Terrain Awareness and Warning System (TAWS).

The next three sections give a functional description and general system architecture of the EVS/SVS/CVS based on analysis of regulatory documents mentioned above.

8.2.2 Enhanced Vision System Overview

The EVS is an electronic means, which uses optical sensors and (optionally) image processing algorithms to expand a flight crew capability to see the most important visual keys of aerodrome structure (e.g. runway lights).

According to DO-315/ED-179, the approved types of sensors for the EVS are IR-sensors, millimeter wave radar, low-level TV sensor. Also the EVS should include the following hardware components:

- On-board computer.
- Display element.
- Display interface with ARINC-818 support.
- Aircraft interface for navigation, flight, and command data.
- Aircraft installation elements: sensor window with anti-icing function, multispectral transparent random.

The EVS performs the following generic tasks:

- Acquire image from the sensor unit.
- Enhance acquired image (optional).
- Obtain information about coordinates, altitude, and orientation of the aircraft from on-board systems.

- Combine the image and flight symbology represented in the form of text and vector graphics.
- Display the EVS images on a Head Down Display (HDD) and/or optionally Head Up Display (HUD).

It is important to mention that the EVS with the HDD is assumed by regulators as a supported function only and gives no additional operational credit to a crew. Meanwhile, the approved EVS with the HUD (so called Enhanced Flight Vision System (EFVS)) allows crew to continue descent below decision height for CAT I down to 100 feet, even when the actual flight visibility is less than required for the landing procedure, but the image displayed on the HUD EFVS clearly visible, and visual landmarks are recognized as defined in [7].

8.2.3 Synthetic Vision System Overview

The SVS is an electronic means, which provides to a crew a computer-generated 3D image of the out-of-cabin view through the use of navigation data, terrain, obstacles, and textures databases. According to DO-315/ED-179, the SVS includes the following elements:

- Terrain database.
- Airports database.
- Obstacles databases.
- Textures database.
- Display element: the HDD and/or optionally the HUD.
- Display interface with ARINC-818 support.
- Data processing board with 3D graphics processing mezzanine.
- Aircraft interface for navigation, flight, and command data.

The SVS performs the following generic tasks:

- Obtain information from on-board systems about coordinates, altitude, and orientation of the aircraft.
- Extract terrain and obstacles data from on-board database according for actual navigation information and including potentially dangerous aerodrome objects in the current phase of flight.
- Generate 3D image of the out-of-cabin view by combining navigation data and on-board database information about terrain, obstacles, and aerodrome objects.
- Generate vector graphics flight symbology including potentially dangerous situations such as controlled flight in terrain (in the case of using with TAWS data), collision with obstacles, and aerodrome objects rolling off runway.
- Create the SVS image by combining vector graphics flight symbology and 3D image of the out-of-cabin view.
- Display the SVS image on the HUD and/or the HDD.

8.2.4 Combined Vision System Overview

Another promising function of out-of-cabin view representation is the CVS. The CVS consists of integral representation (mixture) of real (from the EVS optical sensors) and synthesized (from the SVS) images.

In the CVS mode, the basis is taken from the SVS image, on which typical texture of key topographic features (e.g. runway) are replaced with the corresponding fragments of the current image obtained from the EVS. Thus, the CVS has both advantages of the EVS and the SVS: the EVS realism and flight visibility of unmapped objects (moving obstacles, runway lights, etc.), as well as the SVS ideal visibility conditions of the terrain and mapped obstacles. Moreover, the CVS provides a convenient means of visual inspection of information reliability of the EVS and the SVS.

The CVS performs the following generic tasks:

- Obtain information about coordinates, altitude, and orientation of the aircraft from on-board systems.
- Obtain video information and data from the on-board SVS/EVS.
- Combine the SVS/EVS images.
- Generate the unified information layer of the CVS image.
- Display the CVS image on the HUD and/or the HDD.

8.3 Commercial EVS/SVS/CVS Systems and R&D Projects

Nowadays, there are a number of commercial EVS/SVS/CVS systems that have successfully passed the certification stage. A very short list of companies, whose systems are already available on aircrafts and helicopters, includes “CMC Electronics Inc.”, “Rockwell Collins Inc.”, “Max-Viz Inc.”, “Kollsman Inc.”, “Thales”.

The “CMC Electronics Inc.” offers three types of enhanced vision systems: CMA-2600i I-Series ™, more compact and cheaper version CMA-2600 I-Series ™, and new system CMA-2700 I-Series ™ (Fig. 8.1). In comparison with the CMA-2600, the CMA-2600i has enhanced image processing and improved optics. The CMA-2700 uses the highest resolution dual-band (Short Wavelength InfraRed (SWIR) and Middle Wavelength InfraRed (MWIR)) IR-sensor: 640 × 512 pixels, whereas previous generation uses 256 × 256 pixels. Also, the “CMC Electronics Inc.” is being developing new system, which would be based on millimeter-wave radar. The SVS might also be mounted, which would provide information about the terrain.

The “Rockwell Collins Inc.” represents the Head-up Vision System (HVS), which integrates the EVS, the SVS, and special Head-up Guidance System (HGS™). The EVS uses uncooled IR-sensor and new generation HGS software to create and display image for a pilot (Fig. 8.2).

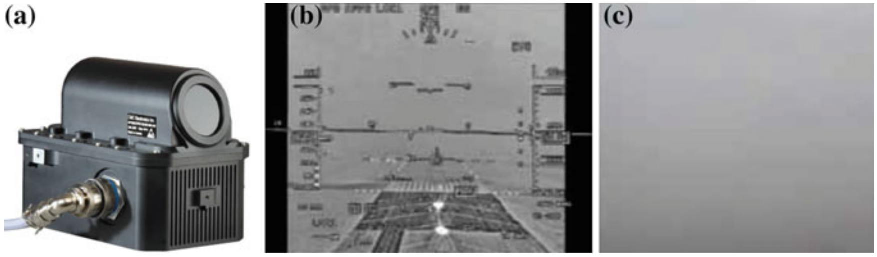


Fig. 8.1 SureSight® I-Series™: **a** CMA-2700 sensor, **b** the EVS image, **c** visual scene. *Source* CMC Electronics EFVS brochures

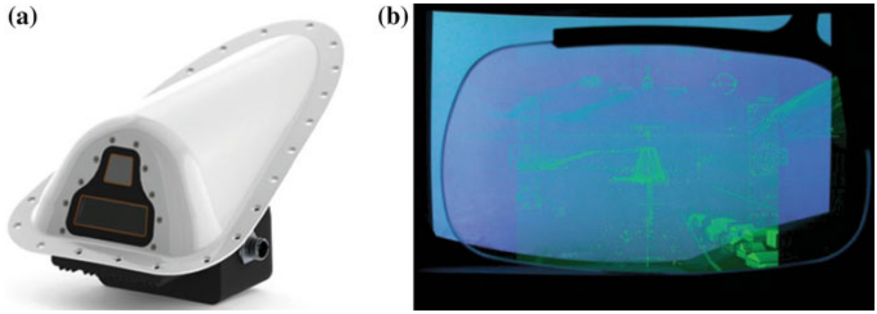


Fig. 8.2 Head-up vision system: **a** Rockwell Collins EVS-3000 sensor, **b** example of the HVS image. *Source* Rockwell Collins HVS brochure

The “Max-Viz Inc.” provides a choice to its customers between two types of the EVS: Max-Viz 600 and Max-Viz 1500. Both products use uncooled solid state microbolometers with 320 × 240 pixels resolution. Max-Viz 600 additionally includes low-light CMOS camera and software to fuse Long Wavelength InfraRed (LWIR) and TV images. Both systems have compact sizes and low weight and can be mounted not only on aircrafts but on helicopters as well. Furthermore, the absence of cryogenic cooling significantly reduces the cost of products.

According to the Max-Viz 1500 specifications, it has pilot selectable dual field of view 53° × 40° or 30° × 22.5° with the optical zoom, which has no loss in resolution, while Max-Viz 600 has only one field of view 40° × 30° (Fig. 8.3). As it is said in product’s brochure “The wide angle gives maximum peripheral visibility during ground operations and the zoom provides early runway acquisition and detection of incursions during takeoff, approach and landing.”

The “Kollsman Inc.” offers its new system EVS II. This system allows pilots to identify runway lights at night and under low visibility as well as ground features. The EVS II includes three elements: IR sensor of 320 × 240 pixels InSb FPA and IR spectrum from 1 to 5 micron, processor unit and IR window (Fig. 8.4). The field of view of this product is 30° × 22.5°. The EVS II is installable in both fixed wing and rotary wing aircraft.

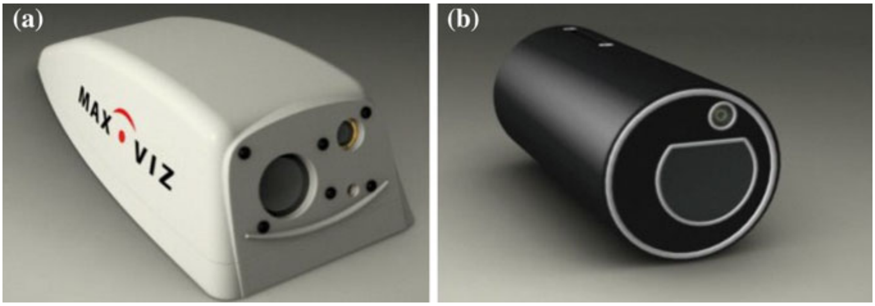


Fig. 8.3 Enhanced vision system: **a** Max-Viz 600 sensor, **b** Max-Viz 1500 sensor. *Source* Max-Viz EVS brochure

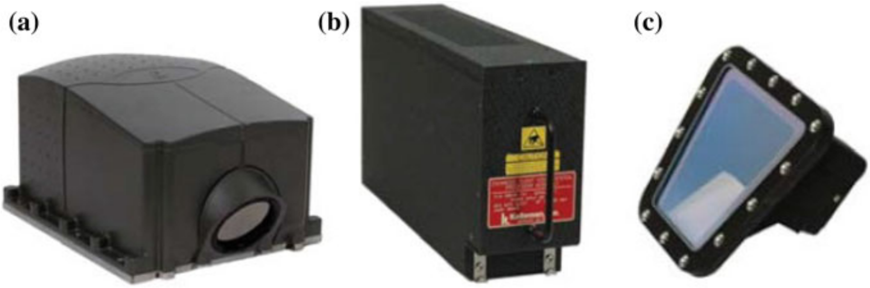
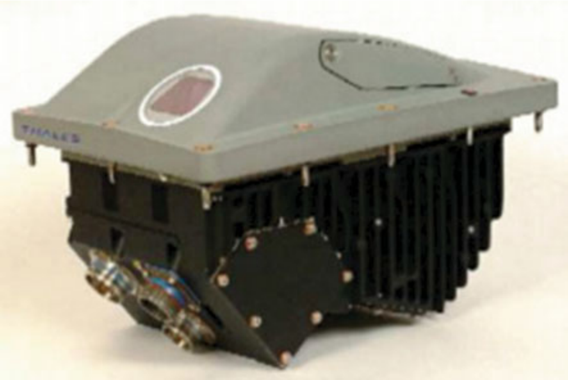


Fig. 8.4 Kollsman EVS II elements: **a** the IR sensor, **b** processor unit, **c** the IR window. *Source* Kollsman EVS II brochure

Fig. 8.5 Thales EVS. *Source* Thales EVS brochure



The “Thales” offers the EVS with the highest IR-image resolution flying today—1,024 × 768 pixels and four fibre optic video outputs to provide pilots with a real-time image. The EVS includes the following components: sensor, processor, and built-in sensor window with anti-icing technology. All components are integrated in a single line replaceable unit (Fig. 8.5).

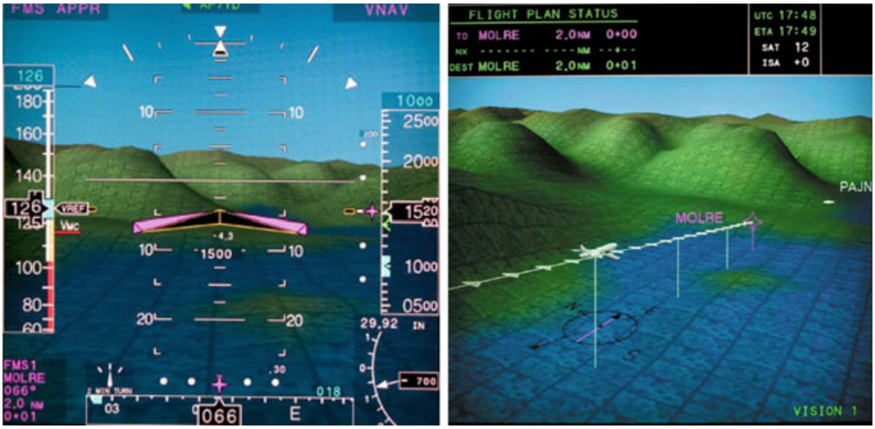


Fig. 8.6 Universal Avionics, Vision-1 SVS: **a** the egocentric view, **b** the exocentric view. *Source* Universal Avionics, Vision-1 brochure

There is also a long list of companies offering the SVS: “Honeywell”, “Rockwell Collins”, “Chelton Flight Systems”, “Blue Mountain Avionics”, “Universal Avionics”, “Avidyne Corp.”, “L-3 Avionics Systems”, and “Garmin”. In this short overview noted the following: “Chelton Flight Systems”, “Universal Avionics”, and “Honeywell”.

The “Chelton Flight System” with the SVS function provides 3D-image of surround space in real time, based on the GPS and preloaded geospatial database of terrain. The system was approved by the Federal Aviation Administration (FAA) in 2003 and available for mounting on private and small commercial aircrafts.

The “Universal Avionics” has introduced Vision-1 SVS, which was certified by the FAA in 2002. Vision-1 supports egocentric and exocentric types of SVS indication (Fig. 8.6).

The “Honeywell” is testing advanced CVS solution, which provides combination of EVS and its SmartView SVS images (Fig. 8.7). The CVS image is displayed on primary flight display. Tests of this system showed a positive result in terms of reducing the weather minimum requirements during the instrumental approach procedure in poor visibility conditions.

Also, government agencies, industry, and universities in leading aviation countries are pursuing the numerous R&D programs to the further development of the EVS/SVS/CVS technologies. Advisory Council for Aeronautics Research in Europe (ACARE) has included the EVS/SVS technologies in its “Strategic Research and Innovation Agenda” up to 2035 [8]. These technologies also are in the FAA and NASA investment plans under such programs as NASA’s Aviation Safety Program (AvSP), Vehicle Systems Safety Technologies project, and the FAA’s Human Factors R&D Project for NextGen [9, 10]. The Defense Advanced Research Projects Agency (DARPA) has the SVS project for helicopters, which is conducted by Honeywell Aerospace [11].

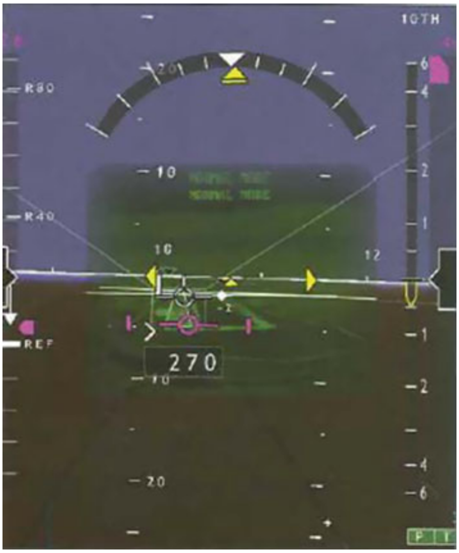


Fig. 8.7 Honeywell SmartView SVS and EVS combination. *Source* Universal Avionics, SmartView brochure

In 2010, the Russian State Research Institute of Aviation Systems (FGUP "GosNIIAS") initiated the R&D project aimed to create scientific and technological bases for the development of the advanced prototype of Enhanced and Synthetic Vision System [12, 13]. This is a part of more complex R&D program for Integrated Modular Avionics (IMA) systems development involving leading enterprises in Russian aviation industry.

The ESVS prototype should have such distinctive features as image fusion, intelligent video processing, sensory and synthetic data fusion, implementation on the IMA platform. These features will provide the following benefits compared with the first generation of the EVS/SVS:

- The improved information content and more realistic view of generated images.
- Partially automation of some functions of a pilot on landing and taxiing stages (runway detection, obstacle detection).
- A scalable architecture based on the IMA.

The sections below describe the current results of the ESVS R&D project.

8.4 The Main Principles of ESVS Prototype Development

In addition to the generic functions of the EVS and the SVS (Sects. 8.2.2–8.2.3, respectively), there are the advanced tasks being studied specifically with the ESVS prototype:

- Synchronize the video input of multi-spectral images with the on-board computer.
- Process and fuse multi-spectral image for visual enhancement.
- Visually detect typical aerodrome objects (e.g. runway, obstacles).
- Combine the enhanced image and the results of automatic detection of typical aerodrome objects, represented in the form of graphic primitives.
- Combine the enhanced image and elements of the SVS image.

The development and testing of such a complex and multipart system as the ESVS is a long and difficult process that involves many subtasks and requires a significant number of additional directions of research. This section briefly addresses several fundamentally important principles of the ESVS development. The issues of computer simulation are discussed in Sect. 8.4.1. The approach based on visual programming language is presented in Sect. 8.4.2. Section 8.4.3 provides multi-spectral data acquisition, and, at last, the hardware of the ESVS is shown in Sect. 8.4.4.

8.4.1 Computer Simulation and Its Role in the Development Process

The ESVS operates in an extremely wide range of conditions determined by many factors, all combinations of which are extremely difficult and economically impractical attempt to register in real flight experiments. Such factors can include different weather and visibility conditions, different terrains, various airports, different types of runways, finally, various situations due to the presence of obstacles on a runway.

Obviously, all mentioned factors should be simulated and the program of simulation experiments should be extensive enough to ensure the reliable ESVS operation in real conditions. The simulation can also be used to explore the range of operating conditions. The quality of simulation should be constantly verified on conformity to data of the flight experiments. The ESVS development by using simulated data only is impossible in principle. Within the R&D project, the simulator is developed for imitation of TV and IR sensor outputs and for the ESVS functions testing through the use of generated imagery by applying computer graphics.

The main component of the simulator is an External Data Simulation System (EDSS). The EDSS includes 3D models of terrain and infrastructure, dynamic models of ground objects and provides a realistic view of out-of-cabin environment with imitation of weather conditions, time of day, atmosphere conditions, sky textures (Fig. 8.8).

The considerable amount of the EDSS generated video was acquired for testing of the ESVS functions in different conditions.

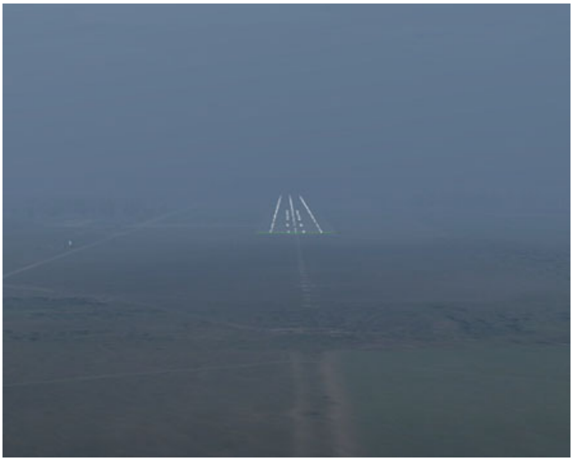


Fig. 8.8 Out-of-cabin view simulation with fog

8.4.2 Visual Programming Language Approach for Algorithms Development

In the hardware-software systems such the ESVS, which include a large number of sensors and other sources of information as well as a large number of processing units that should work jointly in synchronous or asynchronous mode, the important role is given to a special software platform that allows consideration of all these modules as a toolbox of visual components with input and/or output links for data flow. In this case, the developers are able to change easily the configuration of the system, activate and deactivate various sensors "on the fly" to modify and try out different processing algorithms, quickly generate the necessary reports of algorithms testing. In simultaneous processing of multi-spectral images, it is very useful to represent not only the original images and the processing results, but various processing steps as well, providing all the necessary tools for viewing.

For these purposes, the special integrated image processing environment was developed. It is based on "frame-oriented" programming technology and provides the possibility to form interactively any processing schemes from available blocks (frames) without using programming languages (Fig. 8.9). The environment has a user friendly, problem-oriented navigation in multi-window interface and supports such functions as:

- The automatic processing of updated data ("frames-on-line" mode).
- The adaptive adjustment of operational menu accordingly to current processing mode and active window.
- The adaptive adjustment of input/output mode of interactive graphical information (markers, lines, curves, rectangles, etc.) in all appropriate windows simultaneously.

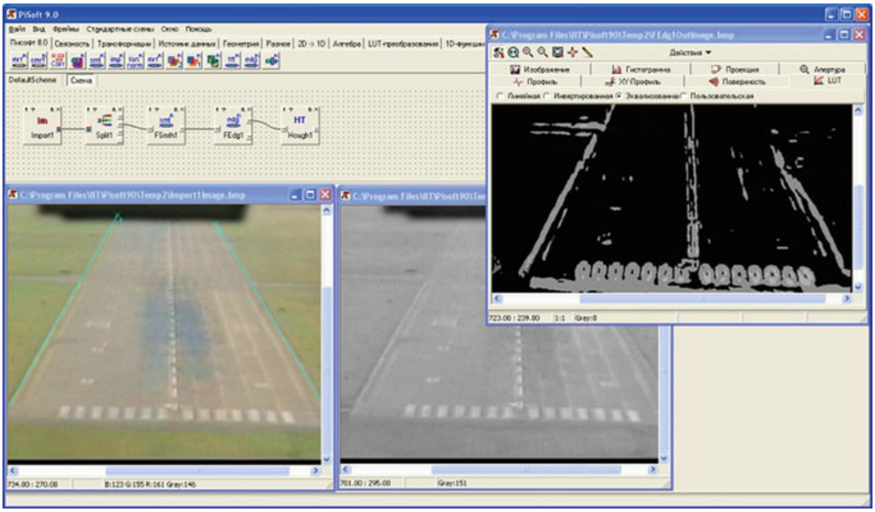

PiSoft 9.0
Файл Вид Фреймы Стандартные схемы Окно Помощь
Писофт 8.0 | Связность | Трансформации | Источник данных | Геометрия | Разное | 2D -> 1D | Алгебра | LUT-преобразования | 1D-функции
DefaultScheme
Схема
Im
Import1
Split1
FSmth1
FEdg1
HT
Hough1
C:\Program Files\IIT\Pisoft90\Temp2\Import1Image.bmp
C:\Program Files\IIT\Pisoft90\Temp2\FEdg1OutImage.bmp
Действия
Изображение
Гистограмма
Проекция
Апертура
Профиль
XY Профиль
Поверхность
LUT
Линейная
Инвертированная
Эквализованная
Пользовательская
734.00 : 270.00
B:123 G:155 R:161 Grey:146
701.00 : 295.00
Grey:151
723.00 : 239.00
1:1
Grey:0

Fig. 8.9 Tool for the ESVS algorithms development

The employing of such environment has significantly reduced the development and testing cycle of the ESVS computer vision algorithms and software.

8.4.3 Multi-spectral Data Acquisition Using Real Sensors

An important element of the ESVS development is a database of multi-spectral images of objects specific to airport runway area. Without such systematic database, the development of robust computer vision algorithms seems impossible. To create this database, the multi-spectral data acquisition system is used. The system consists of compact electro-optical airborne gimbal FILR UltraForce 350 LRF with MWIR and TV channels and FLIR P640 with LWIR camera. The database contains sensor characteristics, multi-spectral image sequences, navigation and flight control data, date and place of registration, and weather conditions.

To estimate the informativity of database images in different conditions of runway visibility, the following parameters are used: linear resolution of runway area in image, edges strength in runway area, brightness separability of runway area, and background description. Also the database is applied to evaluate informativity of spectral ranges in different weather and time conditions and select the most appropriate set of the EVS sensors for the ESVS prototype.

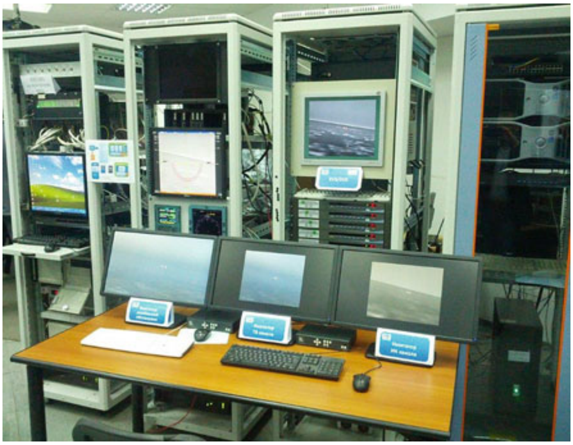


Fig. 8.10 Integrated the IMA bench with the ESVS

8.4.4 Testing ESVS Interaction with On-Board Systems

Another crucial task is the integration of the ESVS with other sensors and on-board systems. Without solving this task, it is impossible to speak about reliability of the system as a whole because unforeseen errors usually occur exactly at the junction of several information, software, and hardware units. Since on the ground position, there is no possibility to fully duplicate the operation of all on-board devices. It is necessary to emulate some real devices by using actual records of their signals or corresponding mathematical models.

The ESVS interaction with other on-board systems is tested at the integrated IMA bench (Fig. 8.10).

In addition to the actual on-board equipment, the bench includes a flight deck simulator with real controls and the aircraft flight simulation system [14]. The bench components use a local network to exchange video sequences, flight and navigation data, and control signals.

8.5 Overview of ESVS Hardware Components and Platform

The EVS functions of the ESVS prototype will be provided by multi-spectral optical-electronic system with three types of sensors (Fig. 8.11): the visible (TV), the SWIR, and the uncooled LWIR. When choosing this set of sensors, not only the energy and spectral characteristics were taken into account, but also the final weight, size, and cost of optical unit as well.

The presence of the TV sensor is largely explained by the necessity to provide a familiar view of out-cabin space in good visibility conditions, when it cannot be observed through the window (for example, in the case of icing). The SWIR

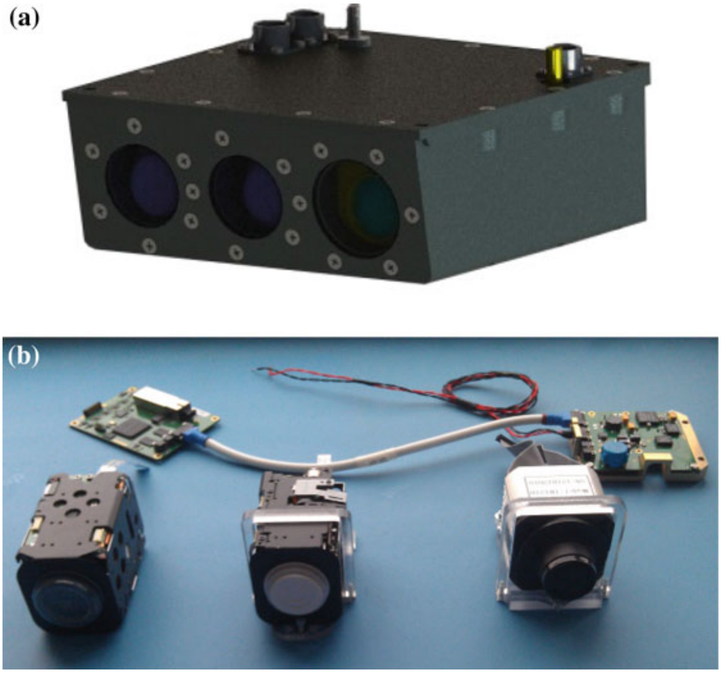


Fig. 8.11 Multi-spectral optical-electronic system: **a** model of the ESVS optical unit, **b** the ESVS sensors

channel provides a good visibility of an important airfield feature—runway lights, in bad weather and at night. Motivation for the uncooled LWIR sensor is a good sensitivity in fog. Our experiments on informational estimation of the IR-channels showed that there are no decisive advantages between the cooled MWIR/LWIR sensors based on HgCdTe and InSb and the uncooled LWIR, however, the latter sensor has a much lower cost and size.

Specification of the optical-electronic system includes:

- Optical unit size—250 mm (D) × 200 mm (W) × 100 mm (H).
- Optical unit weight—not exceeding 3 kg.
- Image dimensions—at least 640 × 512 pixels.
- Field of view (FOV)—at least 40 (H) × 30 (V).
- ARINC 818 output support.
- Video and service data recording.

The prototype of the optical-electronic system is scheduled to be manufactured at the beginning of 2014 by Quantum Optical Systems Co. Ltd. (Russian Federation). The ESVS computer system is developing by the Joint Stock Company “Scientific Design Bureau of Computer Systems” (Russian Federation). This is an open distributed system based on VPX 3U IMA modules [15, 16]. Structurally, the ESVS computational resource will consist of one Digital Image Processing Module with

ARINC-818 video input mezzanine for the EVS tasks and Central Processing Module with Graphic Processor Mezzanine for the SVS tasks. The ESVS modules will occupy two slots in the IMA computational platform.

8.6 Image Processing Algorithms for Enhanced and Synthetic Vision Support

The first generation of the EVS systems, which have passed certification and are available on the market, is mostly a type of "sensor-to-display" systems. It means that the IR imagery is usually displayed directly from a sensor or after some brightness and/or contrast adjustment. Very few systems implement an image fusion (e.g. Max-Viz 600). Ongoing research efforts to expand the EVS, the SVS and the CVS capabilities are mainly focused on the improvement of image enhancement algorithms, the TV, the IR and the Millimeter Wave radar data fusion, objects detection from geospatial database (e.g. a runway), real and synthetic images fusion (see for example [10, 17, 18–20]).

To ensure competitiveness of the future ESVS prototype, the researches ought to be conducted for all mentioned tasks during the R&D project. Up to the date the original algorithms for image enhancement, the TV and the IR images fusion, the runway and obstacles detection were developed for the ESVS and are presented in this section. A set of ground and flight tests have shown that the developed algorithms are robust enough and provide a real-time data processing using computational resources of the IMA platform. It is difficult to compare these algorithms with the ones of competitors due to lack of public datasets such as PETS, ETISEO, CANDELA for tests of video surveillance solutions or YTCelebrity for tests of biometric solutions. Therefore, nowadays verification and validation of the enhanced, synthetic, and combined vision algorithms should be based on the extensive experimental work using simulated and real flight data.

Section 8.6.1 provides the modified of multi-scale Retinex algorithm. Image fusion based on the morphological approach is discussed in Sect. 8.6.2. The algorithms for vision-based runway detection and vision-based detection of obstacle on a runway are described in Sects. 8.6.3–8.6.4, respectively.

8.6.1 Image Enhancement

For image enhancement, the modification of Multi-Scale Retinex [21] for the ESVS (MSR-ESVS) was developed. The MSR-ESVS has the following distinctive features unique to the ESVS prototype:

- A new method of iterative autoregressional smoothing is used to obtain multi-scale brightness estimation in more computationally efficient way as compared to existing techniques (e.g. convolution or summed area table algorithms).

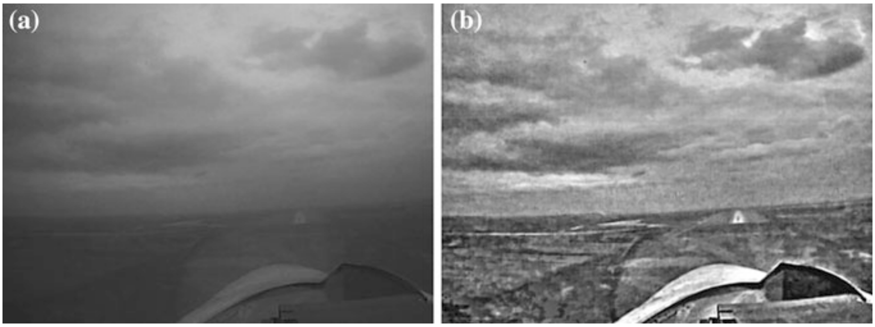


Fig. 8.12 Examples of images: **a** the TV image, **b** the MSR-ESVS image

- A new image normalization procedure is developed for an automatic adjustment of brightness and removing “flicker” effects on result video sequences that are common on today’s displays.

Example of the MSR-ESVS image is shown on Fig. 8.12.

8.6.2 Image Fusion Based on Morphological Approach

The ESVS image fusion algorithm is based on Pyt’ev’s morphological approach [22–24], which provides a convenient formal description of multi-spectral images. In the framework of Pyt’ev’s morphology approach, an image is considered as a piecewise constant function described by Eq. 8.1, where n is a number of non-intersected connected regions of tessellation $\mathbf{F} = \{F_1,\ldots, F_n\}$ of the frame Ω, $\mathbf{f} = \{f_1, \ldots, f_n\}$ is a corresponding vector of real-valued region intensities, $\chi_{F_i}(x, y) \in \{0, 1\}$ is a characteristic (support) function of ith region defined by Eq. 8.2.

$$f(x, y) = \sum_{i=1}^{n} f_i \chi_{F_i}(x, y) \tag{8.1}$$

$$\chi_{F_i}(x, y) = \begin{cases} 1 & \text{if } (x, y) \in F_i \\ 0 & \text{otherwise} \end{cases} \tag{8.2}$$

Set of images with the same tessellation $\mathbf{F}$ is a convex and close subspace $\mathbf{F} \subseteq L^2(\Omega)$ called a shape-tessellation or simply shape (Eq. 8.3).

$$\mathbf{F} = \left\{ f(x, y) = \sum_{i=1}^{n} f_i \chi_{F_i}(x, y) \;\; \mathbf{f} = \{f_1, \ldots, f_n\} \;\; \mathbf{f} \in R^n \right\} \tag{8.3}$$

For any image $g_F(x, y) \in L^2(\Omega)$, the projection onto the shape **F** is determined by Eq. 8.4.

$$\begin{gathered} g_F(x,y) = P_F g(x,y) = \sum_{i=1}^{n} g_{F_i} \chi_{F_i}(x,y) \\ g_{F_i} = \left(\chi_{F_i}, g\right) / \left\|\chi_{F_i}\right\|^2 \quad i = 1, \ldots, n \end{gathered} \tag{8.4}$$

Input data for the ESVS image fusion algorithm is represented by three geometrically matched grayscale multi-spectral images called as the TV, the IR1, and the IR2. The main steps of the algorithm are described as follows:

- Obtain the morphological shape (labeling of connected regions).
- Calculate the morphological projection.
- Merge source images and the morphological projection.

The morphological shapes are obtained for the IR1 images using a histogram-based segmentation. The histogram modes are found by optimization of global separability criterion for $n > 1$ modes. The $(n + 1)$-dimensional vector $\mathbf{t} = (t_0, \ldots, t_n)$ is introduced, where $t_0 = 0$, $t_n = 255$, $t_1, \ldots, t_{n-1}$ are thresholds that separate histogram modes. The mean-square criterion for selecting the segmentation threshold is provided by Eq. 8.5, where $DISP(\cdot)$ is the measure of statistical dispersion (variance).

$$\sum_{i=0}^{n-1} DISP(t_i, t_{i+1}) \rightarrow \min(t_1, \ldots, t_{n-1}) \tag{8.5}$$

To resolve this task, the dynamic programming method is used. If n is unknown, then the task is incorrect and requires additional regularization. The following criterion represented in Eq. 8.6 is used.

$$\sum_{i=0}^{n-1} DISP(t_i, t_{i+1}) + \alpha n \rightarrow \min(n, t_1, \ldots, t_{n-1}) \tag{8.6}$$

For a fixed value of α, the obtained operator of histogram-based segmentation is also a criterial morphological projector. After histogram segmentation, a noise reduction and a labeling of connected regions the IR1 shape (called as IR1’) are considered. The next step is the calculation of morphological projections for the TV and the IR2 images on the IR1’ image as follows: for the each connected region in IR1’ image the mean intensity values of corresponding areas in the TV and the IR2 images are calculated. The fused image is formed by weighted sum of the TV and the IR2 morphological projections.

Example of image fusion obtained in ground experiments is shown on Fig. 8.13.

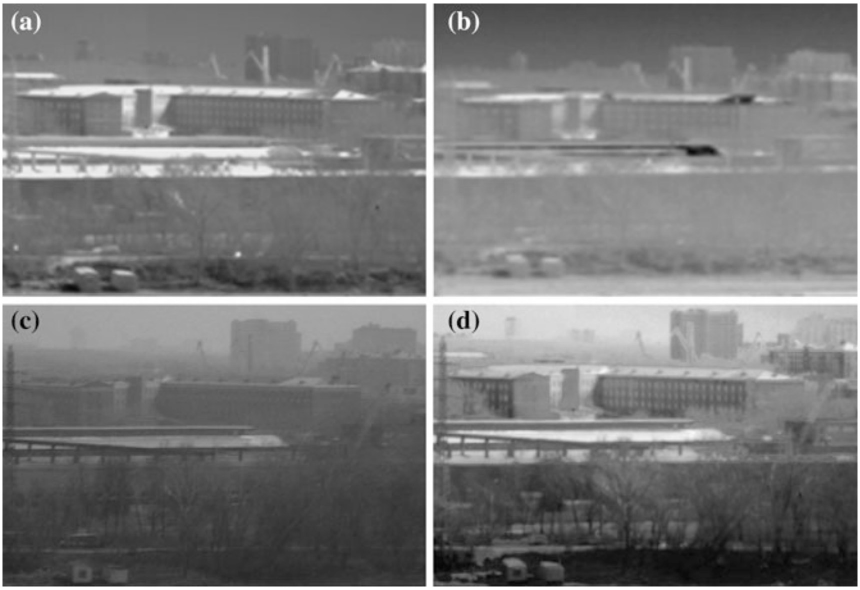


Fig. 8.13 Examples of images: **a** the A MWIR image, **b** the LWIR image, **c** the TV Image, **d** the fused image

8.6.3 Vision-Based Runway Detection

A robust and computationally effective algorithm for automatic detection of runway in video sequences was developed for the ESVS [25].

The algorithm has the following main steps:

- The detection of horizon and relative position of land and sky.
- A preliminary detection of runway area.
- A check for runway marker.
- The detection of runway longitudinal boundaries.
- The spatio-temporal filtering of the runway position.

The Hough Transform (HT) [26] is used to detect a horizon. It is assumed that the horizon line corresponds to the local maxima in the HT accumulator such that, on the one hand, there is a free space (sky), and, on the other hand, an informative region (land).

To check for a runway marker, the horizontal projection of source image vertical edges is calculated by Eq. 8.7, where Im[x, y] is an edge image intensity in a point (x, y), $DimX$, $DimY$ are width and height of the edge image, respectively.

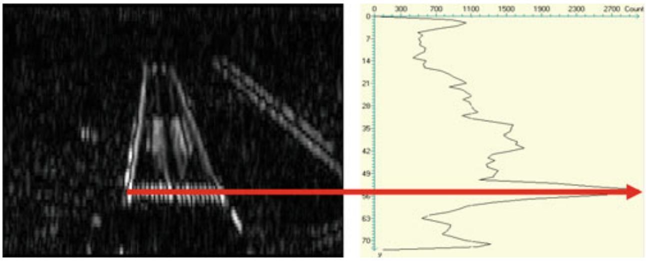


Fig. 8.14 Runway marker detection

$$\mathrm{Proj}_{\mathrm{Y}}[\mathrm{y}] = \sum_{\mathrm{x}=0}^{DimX-1} \mathrm{Im}[\mathrm{x,y}] \quad y = 0\ldots DimY - 1 \tag{8.7}$$

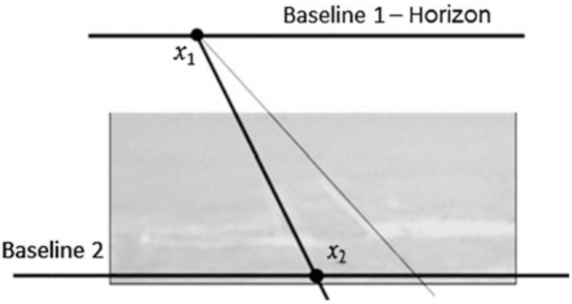
The marker corresponds to global maxima on this projection is depicted in Fig. 8.14.

The detection of runway longitudinal boundaries is based on the original modification of Hough transform—Projection Hough Transform (PHT). The idea of the PHT is to project source image edges onto a horizontal plane in object space (runway plane) and then to calculate the intensity projections of edge pixels on different directions of intensity gradient.

The PHT is calculated using parametrization (x_1, x_2), where x_1 is the vanishing point of runway boundaries, x_2 is the intersection of runway middle line and bottom line of the image. Then the rotation of runway middle line in the runway plane corresponds to moving the vanishing point along the horizon. The shift of the midline line to the left or to the right in the runway plane corresponds to a shift of x_2 along the bottom line of the image (see Fig. 8.15).

The problem of the runway longitudinal bounds detection is reduced to the problem of horizontal brightness profile detection in the PHT differentiated accumulator so that this profile has symmetrical local extremums of maximal amplitude (see Fig. 8.16). These extremums correspond to the left and right bounds of the runway (Fig. 8.17a). To determine the position of the top and bottom runway bounds, the assumption is used that the runway image area is brighter than the surrounding

Fig. 8.15 Parametrization of the PHT



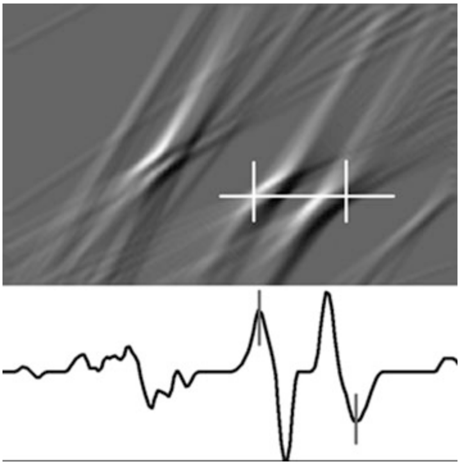


Fig. 8.16 Extremums of the PHT accumulator

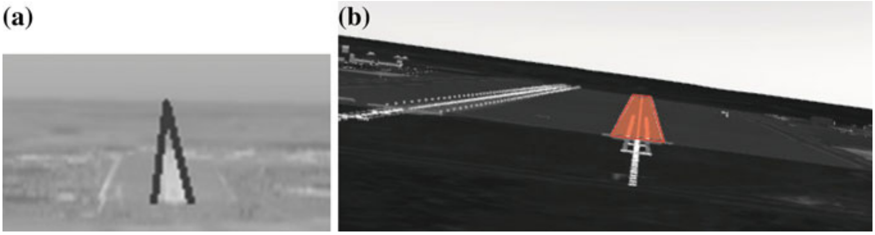


Fig. 8.17 The examples of runway bounds detection: **a** the longitudinal bounds, **b** the transverse bounds

background. The bounds correspond to the most significant value changes on brightness projection taken inside the longitudinal line triangle (runway area).

At the final step, the spatio-temporal autoregression filtering of the following runway parameters is performed: *A* is a horizontal image coordinate of vanishing point, *B* is a distance from the end of the runway to the horizon, *C* is a distance from the beginning of the runway to the horizon, α is an angle between the vertical line and the center line of the runway, β is an angle between left and right boundaries of the runway (Fig. 8.18). Thus in the case of interruption of data flow from the

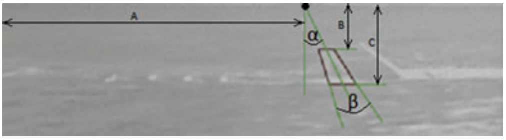


Fig. 8.18 Filtering of runway parameters

primary detection block, the approximation of runway position based on current estimates of geometric model parameters of runway is used.

Currently the runway detection is performed only by computer vision algorithms. In future, work the navigation data, the flight control data, and information about airport structure will be used to increase robustness and computational efficiency of the detection algorithm.

8.6.4 Vision-Based Detection of Obstacle on a Runway

Another innovative function of the ESVS prototype is the algorithm for detection of obstacles on runways. In the context of this work, the "obstacle" means 3D object rising above the runway and making unsafe or impossible to use the runway by approaching aircraft. For example, the obstacle can be an airport service car or another aircraft.

The problem of detecting obstacles located on the surface of known analytical model often occurs in various machine vision applications, e.g. obstacle detection for car safety systems [27, 28]. One of the proven approaches to obstacle detection problem is a stereo vision, which allows retrieve 3D information from a scene. Thus, obstacles can be distinguished from other contrast objects on the surface.

From the design reasons, the ESVS prototype cannot be equipped by a stereo system. The special algorithm was developed, which preserves the benefits of stereo vision but uses the on-board monocular video system [29]. The required stereopair

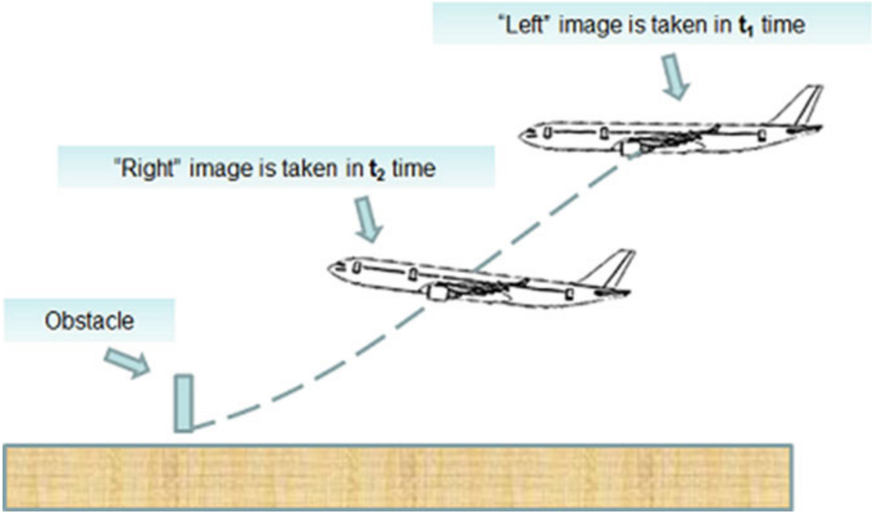

"Left" image is taken in t_1 time
"Right" image is taken in t_2 time
Obstacle

Fig. 8.19 Stereopair from motion

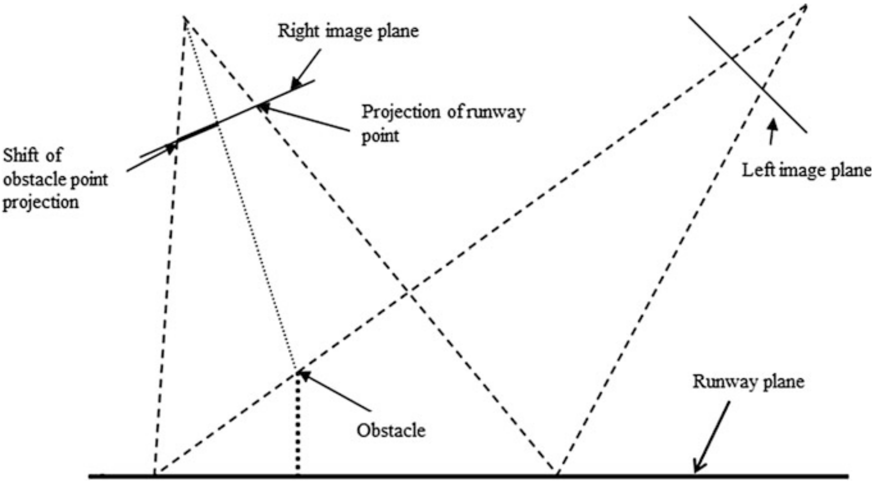

Right image plane
Projection of runway point
Shift of obstacle point projection
Left image plane
Runway plane
Obstacle

Fig. 8.20 Obstacle detection principle

is formed as a result of camera movement ("stereo from motion" method), where the "left" image of stereopair is the image obtained at time t_1 and "right" image obtained at time t_2, $t_1 < t_2$ (Fig. 8.19).

The main idea of the detection algorithm is as follows. The transformation of the left image to the right image plane is performed using a projective camera model. For runway points, the right and transformed left points are the same, but for obstacle points there is a shift between the corresponding right and transformed left points (Fig. 8.20). This shift leads to an intensity deviation on the right and transformed left images difference, which is used as the main obstacle feature

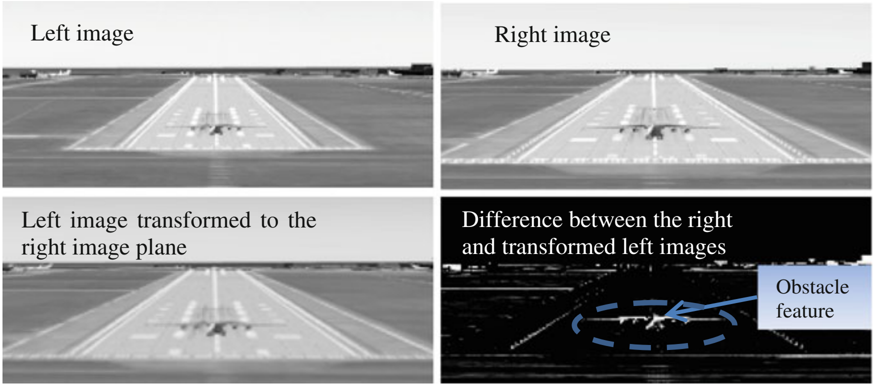

Left image
Right image
Left image transformed to the right image plane
Difference between the right and transformed left images
Obstacle feature

Fig. 8.21 Processing results in the case of an obstacle

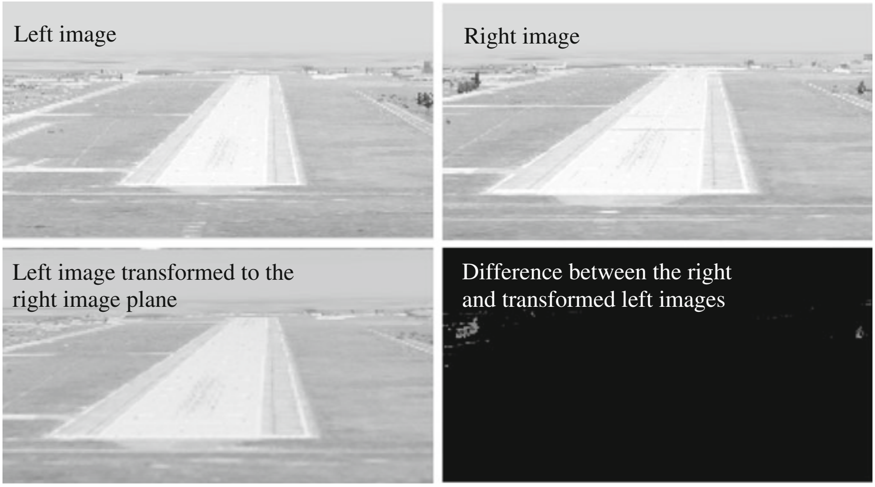

Left image
Right image
Left image transformed to the right image plane
Difference between the right and transformed left images

Fig. 8.22 Processing results in the case of a runway

(Figs. 8.21, 8.22). Parameters of the left image transformation are found by the method of least squares using two sets of matched runway points detected by the computer vision system.

8.7 Prototype of Synthetic Vision Function

The prototype of the synthetic vision function of the ESVS provides the 3D rendering of terrain and display of 3D obstacles in real-time mode (Fig. 8.23). A natural view of generated 3D image is based on the following innovative approaches [30–32]:

- Both texturing and hypsometric methods are used for 3D rendering of terrain.
- Textured geometric shapes are used for the each type of displayed obstacles.
- Sky image rendering is based on a sphere model with use of interpolated texture and fog-effect.
- 3D-symbols of aircraft navigation objects are included in the SVS image.

The SVS obtains data from terrain, airports, and obstacles databases. The source data can be stored in various formats such as Digital Elevation Model (DEM), Digital Terrain Elevation Data (DTED), SHaPefile (SHP), Keyhole Markup Language (KML), etc., or in Storage and eXchange Format (SXF) used in Russian Federation. To provide a real-time processing of 3D data, the source data is converted into the special on-board format.

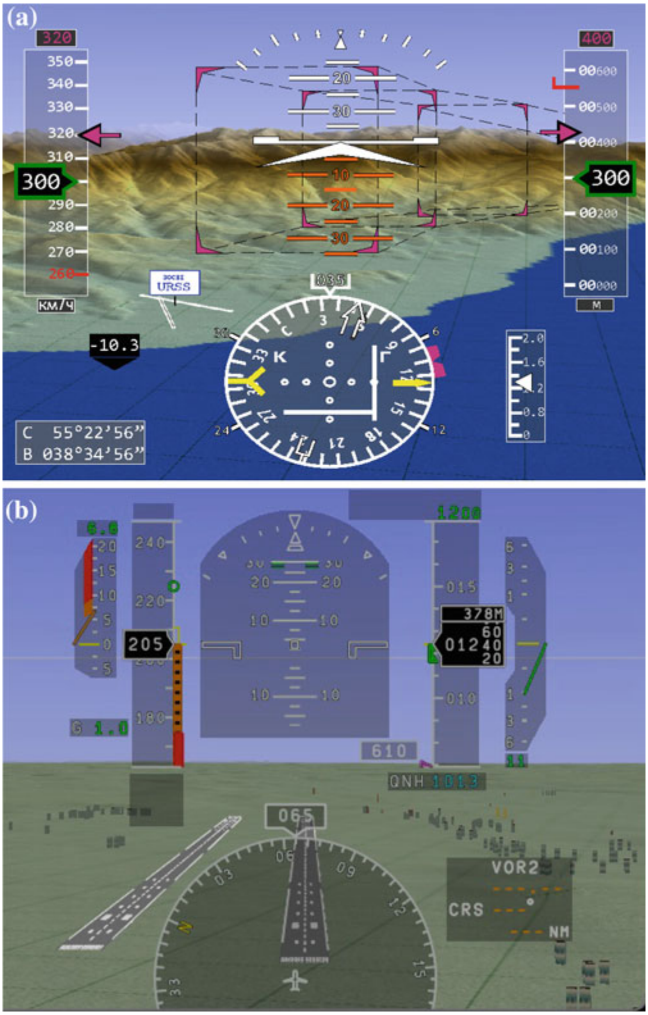


Fig. 8.23 Examples of the SVS images: **a** mountain landscape, **b** 2D “flat” landscape

The SVS software uses hardware-supported implementation of OpenGL. A high performance (25–30 Hz) of the SVS image synthesis is ensured by optimization of the on-board databases and implementation of different levels of image details depending on the distance to a viewpoint.

8.8 Combined Vision Algorithm Based on Photogrammetric Approach

To create the CVS image, it is necessary to solve the algorithmic problem of real (from optical sensor) and synthetic (from the SVS) images matching, since the latter one is formed relying on the information of current aircraft position, which is measured with errors. One possible way to solve this problem is based on the photogrammetric approach and includes the following crucial procedures:

- Receive input data: real image, flight navigation information, and digital terrain model.
- Detect points of known geospatial coordinates on the real image (reference points).
- Make exterior orientation with use of reference points and navigation information as an initial approximation.
- Specify the exact position of the "virtual" camera in a 3D scene using the exterior orientation results. Create a new SVS image corresponding to the real image.
- Combine the real and synthesized images.

This section describes an algorithm for automatic combining of real and synthesized images based on the procedure of camera exterior orientation using a runway points. The formal statement is situated in Sect. 8.8.1. Section 8.8.2 presents the algorithm for exterior orientation using the runway points. Experimental results are drawn in Sect. 8.8.3.

8.8.1 The Formal Statement of the Problem

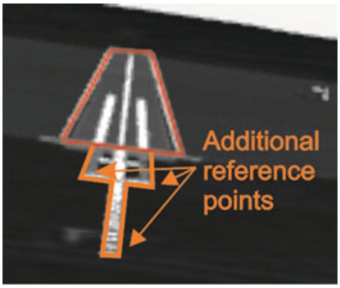
The exact position of the on-board camera is described by unknown vector of exterior orientation in Eq. 8.8, where (x_r, y_r, z_r) is a vector of coordinates, (φ_r, ψ_r, θ_r) is an orientation vector of the aircraft.

$$v_r = (x_r, y_r, z_r, \varphi_r, \psi_r, \theta_r) \tag{8.8}$$

Let the current camera position, measured with errors, be described by vector represented in Eq. 8.9, where (x_e, y_e, z_e) is a vector in geodetic coordinate system obtained from the satellite navigation system, (φ_e, ψ_e, θ_e) is a vector including heading angle, pitch angle, and angle roll of the aircraft.

$$v_e = (x_e, y_e, z_e, \varphi_e, \psi_e, \theta_e) \tag{8.9}$$

Fig. 8.24 Additional reference points



Suppose further that $I_r = I(\mathrm{v_r})$ is the image generated by the on-board camera and $I_e = I(\mathrm{v_e})$ is the synthesized image based on the current values v_e and existing 3D model M. The task is to calculate v_r using parameters v_e, I_e, I_r, and M.

8.8.2 Exterior Orientation Using the Runway Points

Since the CVS system is applied generally on approach and landing stages, then for exterior orientation, it is convenient to use the reference points from airfield infrastructure objects. An automatic detection of such objects in the real image may be performed by various technical vision methods [33–36], in particular, on the basis of 3D models [37, 38]. In the current version of the algorithm, the corner points of the runway are used as the main reference points, which are found by the PHT transform (see Sect. 8.6.3) and marker lights at the start of a runway (Fig. 8.24) as additional points, which are found by Harris detector [39].

An exterior orientation is based on the minimization of residual of collinearity equations taken for the projection center, the runway points, and the corresponding points in the image. The residual of collinearity equations are calculated by Eqs. 8.10–8.12, where b_x, b_y are coordinates of the projection center on the image, x_p, y_p, z_p are coordinates of reference point, x_f, y_f, z_f are coordinates of aircraft, **A** (θ, γ, ψ) is a rotation matrix (Eq. 8.13).

$$\begin{aligned} e_x &= D_x - \frac{x_{pxl}}{f} \cdot D_y \\ e_y &= D_z - \frac{x_{pxl}}{f} \cdot D_y \end{aligned} \tag{8.10}$$

$$\begin{aligned} x_{pxl} &= f \cdot \frac{D_x}{D_y} + b_x \\ y_{pxl} &= -f \cdot \frac{D_z}{D_y} + b_y \end{aligned} \tag{8.11}$$

$$\begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} = \mathbf{A}(\theta, \gamma, \psi) \cdot \begin{pmatrix} x_p - x_f \\ y_p - y_f \\ z_p - z_f \end{pmatrix} \tag{8.12}$$

$$\begin{pmatrix} \cos(\gamma)\cdot\cos(\psi) - \sin(\gamma)\cdot\sin(\psi)\cdot\sin(\theta) & \cos(\gamma)\cdot\sin(\psi) + \sin(\gamma)\cdot\cos(\psi)\cdot\sin(\theta) & -\sin(\gamma)\cdot\cos(\theta) \\ -\cos(\theta)\cdot\sin(\psi) & \cos(\theta)\cdot\cos(\psi) & \sin(\theta) \\ \sin(\gamma)\cdot\cos(\psi) + \cos(\gamma)\cdot\sin(\psi)\cdot\sin(\theta) & \sin(\gamma)\cdot\sin(\psi) - \cos(\gamma)\cdot\cos(\psi)\cdot\sin(\theta) & \cos(\gamma)\cdot\cos(\theta) \end{pmatrix} \tag{8.13}$$

The problem might be solved by the method of least squares. Since the modern onboard satellite navigation systems have a precision that does not affect the accuracy of the synthesized and real images combination on considered distances, the external orientation is performed only for the heading angle, pitch angle, and roll angle.

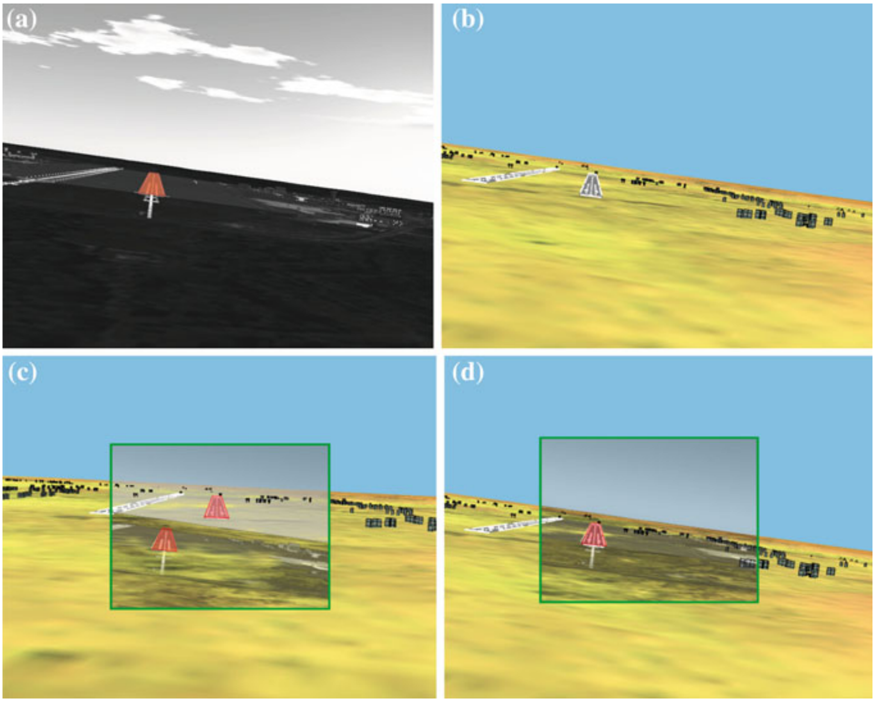


Fig. 8.25 The CVS algorithm testing by the means of computer simulation: **a** modeled TV image and the result of the runway detection, **b** the SVS image, **c** combination of a TV and a SVS images without exterior orientation, **d** combination of a TV and a SVS images after exterior orientation (in *bottom* row the runway images are shown in *red* color to highlight its difference **c** and matches **d)**

8.8.3 *Experimental Results*

The developed algorithm was tested using an External Data Simulation System (EDSS) (see Sect. 4.1). During the experiments the aircraft landing at the glidepath was simulated. The CVS algorithm was started within the flight visibility of the runway. The algorithm receives the TV image modeled by the EDSS (Fig. 8.25a), the coordinates of runway points in the TV image (Fig. 8.25a), the coordinates of runway points in the geospatial coordinate system, and the aircraft orientation received from the dynamic model of the aircraft flight with simulated errors.

The SVS image (see Fig. 8.25b) is formed by the SVS function prototype described in Sect. 8.7. Figure 8.25c shows an example of combining the TV and the SVS images without exterior orientation, just as simple overlay of images. When the simulated errors in pitch angle is equal 3° and in roll and heading angles are equal 5°, the difference between corresponding runway fragments is about 100 pixels. After exterior orientation procedure, the new accurate values of aircraft orientation angles are available for the SVS function, and new SVS image is generated. Result of its combination with corresponding TV image is shown on Fig. 8.25d. The experiments showed that on the simulated trajectories the maximum error of combining the TV and the SVS images in the runway area has not exceeded one pixel and does not have any negative effect on the perception of the CVS image by an operator.

Further the CVS research will be aimed at improving the robustness and accuracy of the CVS algorithm by applying a larger set of reference points in the exterior orientation procedure obtained by the automatic detection and identification of not only the runway but also other types of airfield infrastructure objects.

8.9 Conclusion

The chapter describes current results of the R&D project aimed to create an advanced prototype of the ESVS. The ESVS distinctive features include a multispectral enhanced vision (TV, SWIR, LWIR channels), a computational platform based on VPX 3U IMA modules, an advanced image processing. The algorithmic solutions have been developed for image enhancement, image fusion, vision-based runway and obstacles detection, synthetic and combined vision. Different hardware and software elements of the ESVS prototype were tested in flight experiments, performed by the Federal State Unitary Enterprise “Pilot-Research Center” (Russian Federation).

The further work includes the development of the ESVS optical-electronic system prototype (scheduled to be manufactured at the beginning of 2014), its integration with the IMA platform, and ground and flight tests of the ESVS prototype.

Acknowledgements This work was funded by the Ministry of Industry and Trade of the Russian Federation within the R&D program for IMA systems development. This work was supported by RFBR grants 13-08-01071-a, 11-08-01039-a, and 12-07-31186-mol_a.

This chapter describes the results of the joint work of a large number of people, and the authors wishes to thank all colleagues, who are working on the ESVS project and IMA R&D program in the following companies: State Research Institute of Aviation Systems, Quantum Optical Systems Co Ltd., Scientific Design Bureau of Computer Systems, Ramenskoye Design Company, JSC, Pilot-Research Center.

References

1. CAP1036: Global Fatal Accidents Review 2002 to 2011 (2013) UK: TSO civil aviation authority (CAA)
2. The Aviation Gold Standard (2014) http://www.rtca.org/. Accessed 15 June 2014
3. Standards for Future Aviation (2014) http://www.eurocae.net/. Accessed 15 June 2014
4. RTCA DO-315 Minimum Aviation System Performance Standard (MASPS) for Enhanced Vision Systems, Synthetic Vision Systems, Combine Vision Systems and Enhanced Flight Vision Systems, RTCA, Inc (2008–2011)
5. RTCA DO-341 Minimum Aviation System Performance Standards (MASPS) for an Enhanced Flight Vision System to Enable All-Weather Approach, Landing and Roll-Out to a Safe Taxi Speed, RTCA, Inc (2012)
6. ED-179B Minimum Aviation System Performance Standard (MASPS) for Enhanced Vision Systems, Synthetic Vision Systems, Combine Vision Systems and Enhanced Flight Vision Systems, EUROCAE (2011)
7. FAA 14 CFR 91.175 Takeoff and Landing under Instrument Flight Rules (2007)
8. Advisory Council for Aviation Research and Innovation in Europe (2014) http://www.acare4europe.org/sria/exec-summary/volume-2. Accessed 15 June 2014
9. Shelton KJ, Kramer LJ, Ellis K, Rehfeld SA (2012) Synthetic and enhanced vision systems for nextgen (sevs) simulation and flight test performance evaluation. IEEE/AIAA 31st digital avionics systems conference (DASC’2012):2D5-1–2D5-12
10. Bailey RE (2012) Awareness and detection of traffic and obstacles using synthetic and enhanced vision systems. Technical report NASA/TM-2012-217324, L-20100, NF1676L-12443
11. Honeywell improves synthetic vision avionics (2014) http://www.flightglobal.com/news/articles/honeywell-improves-synthetic-vision-avionics-392013/. Accessed 15 June 2014
12. Vygolov OV (2013) Enhanced and synthetic vision systems development based on integrated modular avionics for civil aviation. 32nd digital avionics systems conference (DASC’2013):2B5.1–2B5.13
13. Zheltov S, Vizilter YV, Vygolov OV (2013) Enhanced and synthetic vision systems for civil aviation. Polet J 1:33–39 (in Russian)
14. Kosyanchuk V (2012) Prospects of development of onboard avionics suite on the basis of IMA. International conference on condition and prospects of development of integrated modular avionics
15. VPX Baseline Standard (2007) American national standards institute (ANSI/VITA)
16. Kulikov D, Tarandevich K (2012) Base technological solutions of « Basis.5 » IMA platform. International conference condition and prospects of development of integrated modular avionics
17. Thurber M (2011) Honeywell moves forward on head-down EVS/SVS combo. NBAA convention news, pp 30–32

18. Actronics (2014) http://max-viz.com/2009/07/max-viz-awarded-patent-for-ir-image-processing/. Accessed 5 June 2014
19. Tandra S, Rahman Z (2008) Robust Edge-detection algorithm for runway-edge detection. SPIE Image Process Mach Vision Appl. doi:10.1117/12.766643
20. Kumar SV, Kashyap SK, Kumar NS (2014) Detection of runway and obstacles using electro-optical and infrared sensors before landing. Defense Sci J 64(1):67–76
21. Jobson D, Rahman Z, Woodell GA (1997) A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Trans Image Process 6(7):965–976
22. Pyt'ev Yu (1993) Morphological image analysis. Pattern Recogn Image Anal 3(1):19–28
23. Vizilter Yu, Vygolov OV, Rubis AY (2012) Morphological correlation coefficients of image forms for multi-spectral image fusion tasks. Bull Comput Inf Technol 3:14–20 (in Russian)
24. Vizilter Yu, Zheltov SY (2012) Geometrical correlation and matching of 2D image shapes. ISPRS Ann Photogrammetry, Remote Sens Spat Inf Sci 1–3:191–196
25. Komarov D, Vizilter YV, Vygolov OV, Knyaz VA (2012) Vision based runway detection for aviation enhanced vision system. International conference on intelligent information processing IIP-9, pp 350–354 (in Russian)
26. Hough PVC (1962) Method and means for recognizing complex patterns. U.S. Patent 3,069,654
27. Zehang S, Bebis G, Miller R (2004) On-road vehicle detection using optical sensors: a review. 7th IEEE international conference on intelligent transportation systems (ITSC'2004), pp 585–590
28. Zheltov S, Sybiryakov AV, Vygolov OV (2002) Car collision avoidance system based on orthophoto transformation. Int Arch Photogrammetry Remote Sens Spat Inf Sci 34(5):125–130
29. Stepanyanc D, Komarov DV, Vygolov OV (2012) The development of vision based algorithm for automatic runway obstacle detection for aviation enhanced vision system. International conference on intelligent information processing IIP-9, pp 402–405 (in Russian)
30. Djandjgava GI, Sazonova TV, Shcherbunov GI (2010) Issues of cartographical support of modern navigation systems for aerial vehicles. Issues Defense Technol Mag 9:33–39 (in Russian)
31. Djandjgava GI, Sazonova TV, Leshchuk OG, Shelagurova MS (2011) Integrated usage of digital cartographic information for navigational and displaying tasks during all flight stages of modern aerial vehicles. Aerosp Instrum-Making Mag 3:11–20 (in Russian)
32. Djandjgava GI, Sazonova TV, Shelagurova MS (2012) Intellectual support of flight crew during the landing of an aerial vehicle. Issues Defense Technol Mag 9(5):4–14 (in Russian)
33. Vizilter Yu, Zheltov S, Stepanov A (1996) Object detection and recognition using events-based image analysis. SPIE Proc 2823:184–195
34. Vizilter Yu, Zheltov S, Stepanov A (1996) Events-based image analysis for machine vision and digital photogrammetry. ISPRS proceedings. Int Arch Photogrammetry Remote Sens 31 (B3):898–902
35. Vizilter Yu (2007) Applying morphological events-based analysis in machine vision tasks. Bull Comput Inf Technol 9:11–18 (in Russian)
36. Vizilter Yu, Zheltov S (2009) Projection morphology for image based objects detection and identification. Int J Comput Syst Sci 2:125–138 (in Russian)
37. Boguslawski P, Gold CM, Rahman AA (2012) CAD construction method of 3D building models for GIS analysis. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 1–2:93–98
38. Zhang S, Sullivan GD, Baker KD (1992) using automatically constructed view-independent relational models in 3D object recognition. In: Sandini G (ed) 2th European conference on computer vision (ECCV'1992), Springer-Verlag, Berlin, Heidelberg
39. Harris C, Stephens M (1988) A combined corner and edge detector. 4th Alvey vision conference, pp 147–151

Chapter 9
Navigation of Autonomous Underwater Vehicles Using Acoustic and Visual Data Processing

Igor Burdinsky and Anton Myagotin

Abstract A navigation model for an Autonomous Underwater Vehicle (AUV) combines acoustic and vision-based navigation principles. The acoustic guidance is based on the Time-Of-Flight (TOF) measurements carried out in a one-way asynchronous mode. Vision-based positioning employs a digital image processing approach using the log-polar transformations for a temporal series of on-board camera images. A proportional-integral-derivative controller is used to change a vehicle's position and course. The corresponding control and error functions are provided. The model is implemented and tested numerically. The experiments confirmed a high reliability of the developed algorithms, which can be further applied in autonomous vehicle navigation and docking systems.

Keywords Autonomous underwater vehicle · Navigation · Acoustics · Time-of-flight measurements · Homing · Target capturing · Log-polar transform · Pattern detection

9.1 Introduction

It is a well known fact that the World Ocean is the richest and at the same time the least explored source of natural resources on the Earth. This idea inspires the sea power nations to invest substantial funds in marine researches and especially in the development of general- and specific-purpose underwater vehicles. During the last

I. Burdinsky (✉)
Pacific National University, 136 Tikhookeanskaya Street, Khabarovsk 680035,
Russian Federation
e-mail: igor_burdinsky@mail.ru

A. Myagotin
Saint Petersburg State University of Civil Aviation, 38 Pilotov Street,
St. Petersburg 196210, Russian Federation
e-mail: anton.myagotin@gmail.com

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_9

decades, the intensive studies in this field led to novel technological solutions and appearance of unique devices. A quite large family of up-to-date underwater vehicles and robots are usually classified as mentioned below [1]:

- The Deep Submersible Vehicle *(*DSV), which is a small diving manned submarine.
- The Remotely Operated Vehicle (ROV), which is controlled by an operator from a supporting ship.
- The Autonomous Underwater Vehicle (AUV), which is a self-floating robot operating without a manual control.

Due to autonomy, the AUVs are obviously preferable for a broad range of scientific, military, and commercial applications such as seafloor observations [2], mine detection [3], sub-sea pipeline inspections [4], among others. At present, a dominant challenge to reach the real operational autonomy promotes the development of a reliable navigation system performing positioning, guidance, and docking of an underwater vehicle. The existing navigation systems use one of the concepts described below.

The modern Global Positioning System (GPS) provides the accurate coordinate estimations for any type of tracking objects. Since the penetration depth for radiowaves into sea water does not exceed 25–80 cm [5], an underwater vehicle equipped with a GPS receiver should float near the sea surface. In this case, the AUV is able to identify its location precisely and in real time [6]. However, the above mentioned requirement reduces significantly the range of possible robot's applications.

An alternative approach for the fully 3D navigation capability provided by GPS is a relative positioning system consisting of a Doppler velocity log measuring vehicle velocity relative to the seafloor and an inertial measurement unit providing information regarding the linear acceleration and bearing [7]. Having initial vehicle coordinates, the measured data are used to compute the current position. It is worth mentioning that with the course of the time the estimation error increases. Therefore, the inertial systems are typically combined with a GPS unit.

A terrain-based navigation uses a given map of a sea region, where the AUV mission is planned. Referencing to a particular vehicle location, a terrain-based navigation is performed by the inspection of visual data. Either conventional photo images or images produced by the side-scan sonar are used as the input data [8]. It is clear that the seabed map is not always available beforehand. In this case, it can be reconstructed on-line using simultaneous localization and mapping algorithms [9]. The acoustic navigation systems are based on the concept of the TOF measurements for ultra-sound waves propagating between a stationary buoy (or a set of buoys) and a hydrophone (passive sonar) installed on the vehicle [10]. Nowadays due to simplicity, accuracy, and robustness, the acoustic navigation systems dominate over other solutions.

Recently, thanks to advances in the development of high-performance embedded hardware (such as Field Programmable Gate Array (FPGA) and Digital Signal Processing (DSP)) the on-board vision-based guidance has become possible [11]. The idea is almost straightforward: an optical camera attached to an underwater vehicle produces a sequence of frames, which is processed afterwards by a digital unit executing the complex image processing algorithms.

Analyzing the existing approaches for the underwater navigation, one might notice that an acoustic operating distance (i.e., a maximal distance to the closest buoy) must be in the range of $O(10\text{ m}–10\text{ km})$, while a vision-based guidance is limited only by several meters (due to strong attenuation and scattering of optical light in sea water). In this chapter, the combination of acoustic- and vision-based principles is considered. This approach approved by numerical simulations allows for the robust near- and far-distant vehicle navigation, thus it is profitable for the implementation in a real navigation system.

This chapter is organized as follows. Section 9.2 describes formally the navigation problem. A detailed description of acoustic vehicle guidance is given in Sect. 9.3. The vision-based positioning algorithm is outlined in Sect. 9.4. Section 9.5 summarizes the simulation results. Finally, the chapter concludes in Sect. 9.6.

9.2 Problem Statement

Consider a torpedo-shaped autonomous underwater vehicle equipped with engines, which are able to change its direction (course, bearing) $\phi \in [-\pi, \pi]$ and velocity $v \in [-v_{\max}, v_{\max}]$. Let us assume that $v_{\max}$ is much smaller than sound propagation speed v_s in sea water ($v_{\max} \ll v_s$). The vehicle has a hydrophone and an optical camera that is oriented downwards. The sketch of the model configuration is shown in Fig. 9.1.

The considered AUV mission corresponds to a so-called homing and docking problem. Namely, starting from an arbitrary position the AUV should reach a certain location and take a requested orientation in space. Both parameters are defined unambiguously by an underwater target that is a two-dimensional billboard fixed on the seabed and equipped with an acoustic buoy. The buoy operates as a transmitter while the hydrophone registers only acoustic signals. Both the transmitter and

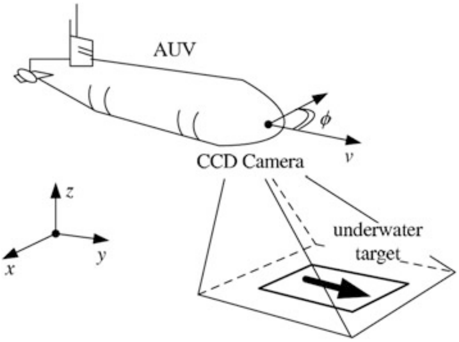


Fig. 9.1 Model configuration

transducer are assumed to be synchronized so that synchronization moments for navigation signals and their period T are known a priori. Consequently, the time span t between the synchronization and registration moments defines the AUV-buoy distance. It is worth to mention that the AUV-buoy distance estimations obviously require accurate clocking on both transmitter-transducer sides. The described approach requires a low synchronization error (e.g. 10^{-8} ppm), this is why either an accurate clocking device should be used on-site or sophisticated algorithms for on-going synchronization must be applied. Additionally, let us assume that a signal transmission channel agrees well with the Rician channel model, where the dominant multi-path components contain almost all the energy of the received signal.

In our model the navigation consists of two logical stages, which are called a long-distant and near-distant guidance, respectively [12]. During the first stage, a vehicle performs a series of the TOF measurements and uses them to attain an approximate target location. Capturing the target billboard by a vision system, it switches to a near-distant guidance (the second stage) in order to improve its location and change the bearing in accordance with the underwater target.

9.3 Acoustic Navigation

Conventional acoustic navigation systems use the concept of the TOF measurements for the ultra-sound waves propagating in sea water. For a measured TOF t and known sound speed v_s, the distance between the buoy and vehicle r is determined by Eq. 9.1.

$$r \cong c \cdot t \cdot v_s \tag{9.1}$$

Depending on the constant c, two strategies for signal exchanging can be distinguished (Fig. 9.2). In the case of the two-way transmission the AUV emits a pilot signal, which is replicated by the buoy after a fixed delay Δ_d, and it is registered back on the AUV side. The estimation t is proportional to the doubled AUV-buoy distance r (c = 1/2). The one-way transmission assumes that an acoustic buoy transmits navigation signals, which are registered by an AUV hydrophone in a so-called salient mode (c = 1).

Comparing pros and cons of the above strategies, one usually notices that the two-way transmission is simpler in implementation. On the other hand, it takes twice longer time for a single TOF measurement. Moreover, it uses the capacity of a channel inefficiently. Additional challenges appear for multiple underwater vehicles operating in the same sea region. One has to apply either a time- or code-division schemes to share access to the common channel [13]. Consequently, among two mentioned modes the more efficient one-way transmission is selected.

In general, the acoustic guidance relies on the solutions for two sub-problems: the implementation of an accurate TOF and the measurements v_s and translation of the TOFs into engine control signals changing the vehicle position and course.

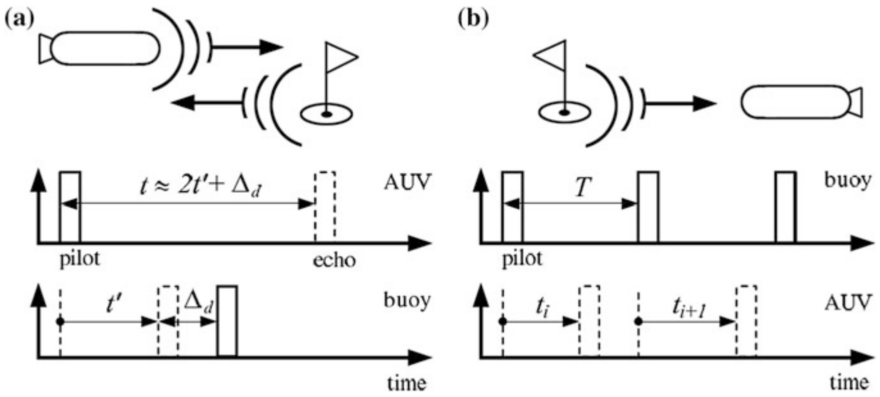


Fig. 9.2 Signal exchanging modes: **a** two-way, **b** one-way

The sound speed in sea water is a complex function of temperature T (°C), salinity S (psu), and depth D (m). A simplified formula adapted from Wilson's equation for the computation of the sound speed in seawater [14] has a view provided by Eq. 9.2, where v_s is given in m/s.

$$v_s = 1449 + 4.6T - 0.055T^2 + 0.0003T^3 + 1.39(S - 35) + 0.017D \tag{9.2}$$

The temperature and salinity measurements of salt water by an AUV are a nontrivial task. Therefore, in practical applications the speed is assumed to be constant and is of 1,500 m/s.

An accurate estimation of the TOF is a challenging task as well. Major sources of erroneous TOF estimations can be classified as follows:

- The transmission loss. Due to wave front propagation, the amplitude of a sound wave reduces. The intensity attenuation is expressed as $I = I_0 \exp(-\alpha r)$, where I and I_0 are emitted and transmitted intensity, respectively, r is a transmitter-receiver distance, and α is an attenuation coefficient.
- The multi-path propagation introduces the amplitude-frequency distortions and the multiple echoes in the original signal, which lead to so-called inter-carrier and inter-symbol interferences.
- The Doppler effect appears due to a relative motion of a sound source with respect to a target. The effect changes a carrier frequency that is a serious problem in acoustic systems using frequency modulations.
- The acoustic noise. There are different kinds of noise sources in sea environment: hydrodynamical (wind, surface waves, rain), technological (resulting from human activity), thermal (chaotic motion of molecules), biological, etc.
- The synchronization error may appear due to imperfect clocking of receiver and transmitter hardware.

It is clear that a robust acoustic guidance demands a reliable registration of navigation signals by an underwater vehicle even in the presence of acoustic noises. A conventional detection method for the transmitted signal in a noisy channel is the computation of a correlation function of an input signal and a given mask (being a series of discrete samples of expected signal). In this work, the phase-manipulated M-sequences were used as transmitting signals [15]. Digital bits (zeros and ones) are randomly placed in the sequence, i.e. they are uniformly distributed over a binary string. From this reason, the M-sequences are often called pseudo-random or pseudo-noise codes. Let C_S denote the bit length for an M-sequence, a uniform probability for symbol occurrence implies that the side lobe level of correlation function is proportional to $1/\sqrt{C}_S$. Consequently, the increasing of the length of the sequences leads to the decreasing of the magnitude of side lobes. This characteristic property of the M-sequences guarantees a high reliability for the detection of a transmitting signal even in the presence of a significant noise level.

Vehicle motion relative to an acoustic source changes the duration of an M-sequence. Thus, the comparison of a reference signal with a Doppler-distorted received signal may cause errors during the detection stage. In order to avoid this problem, it was suggested to use the short M-sequences having bit lengths much longer than possible time axis contractions induced by the Doppler effect.

For the detection of a signal that represents a bit sequences, the so-called character-stepped correlation function was computed. A digitalized signal is divided into a series of chunks. Every chunk corresponds to a single bit (character) in a sequence. Further, the correlation function $\langle R \rangle_k$ is computed by Eq. 9.3, where M_j is the jth bit in a bit sequence, s_j is an estimated value of the jth bit ($s_j = 1$, if $\sum_{i=0}^{N_s-1} m_i \cdot u_{k+i+j \cdot N_s} < 0$, and $s_j = 1$ otherwise), u represents input samples, m_j is an array of mask samples, C_S is a total number of bits, N_S is a length of a single character measured in discrete samples, $\oplus$ denotes a logical XOR function, and $\bar{a}$ is a logical inversion of a binary string a.

$$\langle R \rangle_k = \sum_{j=0}^{C_S-1} \overline{M_j \oplus s_j} \tag{9.3}$$

Top level schematics for a correlation unit implementing above mentioned Eq. 9.3 is shown in Fig. 9.3. The basic element of the unit is an n-parallel four stage pipeline allowing for simultaneous computation of n values for the correlation function (outputs values R_{k+j} with $j \in [i, i+n-1]$). The peculiarity of the circuit is two clock frequencies: samples u_k are loaded into the input register D (organized as a First Input First Output (FIFO) queue) with the frequency f_d, and the remaining functional units are clocked with a system frequency $f_F \gg f_d$.

The correlation unit works as follows. After the register file D is filled with ($N + n$) samples, first n input samples are written into the shift register RG_d. Simultaneously, the current mask value m_i is recorded into the mask register RG_m. Afterwards, the circuit performs multiplications of the samples in RG_d with m_i in all n pipeline levels in parallel. On the next system clock the intermediate results of the

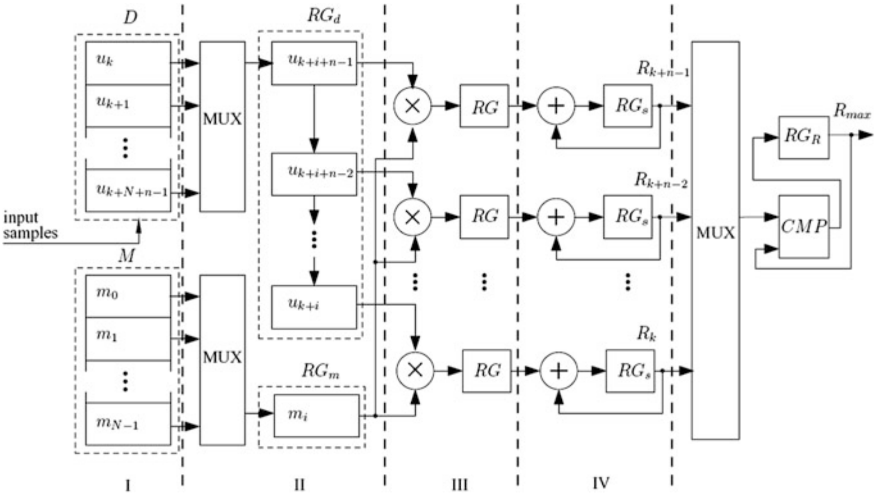


Fig. 9.3 Top level schematics of a correlation unit

correlation function are written into registers RG_S, the multiplexer fetches the next sample from D shifting the values of RG_d, and the next mask value m_i + 1 is selected. After ($N + n$) system clocks n values of the computed correlation function are stored in the registers RG_S. The values are transferred sequentially into the input of the comparator *CMP*, which records the current maximal value of the correlation function into the register RG_R. In addition, it stores the sample number that corresponds to the maximal value. After the computation of the first n values of the correlation function the registers RG_S and RG_d are cleared and new processing cycle begins.

The schematics were implemented on a Xilinx Virtex-4 XC4VLX25 device [16]. The parameters of the resulting signal processing system are f_S = 12 kHz, f_d = 48 kHz, f_F = 100 kHz, N = 4,064, and n = 2. The implementation of a single correlation unit required less than 10 % of FPGA recourses. Thus, this device allows one to implement up to 10 correlation units. Each unit can be tuned to the signals detection of a different carrier frequency and/or to the detection of different M-sequences, i.e. it is possible to organize navigation of multiple vehicles working in a common bandwidth.

In the following let us focus on the second sub-problem, which is the translation of the TOF estimates into engine control signals. Our solution is based on an extremum search controller [17] forcing the vehicle to turn toward the buoy. The decision for the current bearing is made on a difference of two successive TOF measurements. Let p_1 and τ_1 be an AUV position, and the estimate of the TOF measurements at p_1, respectively. After a period T, the vehicle that is moving with a constant velocity v attains a location p_2, where it registers the next pilot signal. Basing on the heading φ one can distinguish three extreme cases:

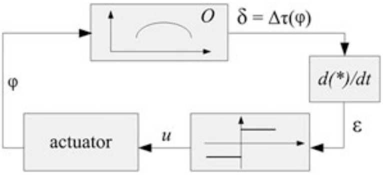


Fig. 9.4 Flowchart of the extremum search controller

- φ = 0: the AUV floats toward the buoy, i.e. $\tau_1 > \tau_2$ and $\tau_1 - \tau_2 \rightarrow \max$.
- φ = ± π/2: the AUV moves equidistantly around the buoy, i.e. $\tau_1 = \tau_2$.
- φ = ± π: the AUV moves in the opposite direction from the buoy, i.e. $\tau_1 < \tau_2$ and $\tau_1 - \tau_2 \rightarrow \min$.

The difference $\Delta\tau = (\tau_{i\,-\,1} - \tau_i)$ is a function of φ and it has a well-defined extremum point. Consequently, the homing problem can be reduced to the maximization of the difference Δτ.

Figure 9.4 depicts a flowchart of the Extreme Search (ES)-controller. The difference of two successive TOF measurements δ = Δτ(φ) from the signal registration block *O* is passed to the differentiating unit. The derivative ε = dδ/d*t* is computed and transferred further to a signum relay. The latter is a logical device that changes the sign of a control/impact function *u*, when the controller passes through the extremum point of the object characteristics δ. The impact is chosen to be proportional to the magnitude of the derivative ε. The actuator in our case is an engine control mechanism that forces the AUV to turn toward the source of acoustic signals. For the sake of simplicity, let us assume that the controller is an inertialess system, i.e. hydrodynamical parameters of the environment and mechanics-related delays are not taken into consideration.

Let us consider a typical operation cycle of the ES-controller. The unknown object characteristic δ is a periodical function of a heading φ as it is shown in Fig. 9.5a. The angle φ_1 denotes an initial heading corresponding to the point M_1 on the δ-curve. One can assume that after initialization at time t_1 the parameter φ increases, i.e. the AUV is turning toward the buoy. Moving from the point M_1 to M_2,

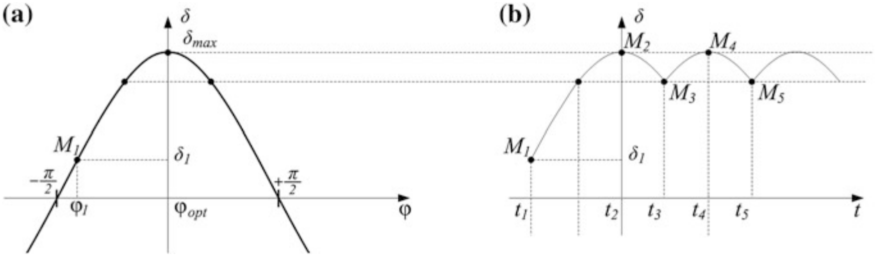


Fig. 9.5 Characteristics of the object and the typical operating cycle of the ES-controller: **a** object characteristic δ(φ), **b** characteristic of the operation cycle in the ES-controller

the characteristic δ increases with a simultaneous decrease of the derivative ε. At time t_2 the controller reaches the maximal value $\delta_{\max}$ and the derivative ε converges to zero. Due to an intentionally introduced insensitivity to a small magnitude of the derivative, the controller continues to generate a positively signed impact so that it passes through the extremum point. At time t_3, the derivative ε attains a minimum allowed negative value ε_{in}, signum relay the flips the sign of the impact u, thus the AUV is turning in a backward direction. At time t_4 the system crosses the extremum again (moving from the point M_3 to M_4), and the operation cycle repeats.

In the above example, the characteristic δ is assumed to be positive, consequently, the AUV always floats toward the buoy. However, if the AUV is oriented originally in the opposite direction, the homing procedure fails. The δ-curve has extremum points at $\varphi = \pm\,\pi$ as well, thus the controller forces the AUV to turn in the exact opposite from the desired direction. The solution for the problem could be the following. The ES-controller is switched off, when the characteristic δ becomes negative-valued. The AUV begins the motion on a circular path until δ becomes well above zero. Thus, the ES-controller can be started again and further navigation proceeds in a way described above.

The authors analyzed the boundary operation conditions for the ES-method in [18]. The resulting criterion is formulated by Eq. 9.4.

$$\rho < 1/(2 \cdot e_{\max}) \tag{9.4}$$

In the formula, the auxiliary measure $\rho = r/(v \cdot T)$ aggregates such model parameters as AUV-buoy distance r, vehicle velocity v, and navigation period T. The variable $e_{\max}$ expresses a maximal value for the TOF estimation error. Consequently, fixing T, $v_{\max}$, and $e_{\max}$, a maximal (theoretical) radius for the region, where acoustic navigation with the ES-controller succeeds, can be estimated numerically.

The acoustic guidance finishes, when a vision system captures an underwater target and the further positioning is continued basing on a series of images produced by the on-board Charge-Coupled Device (CCD) camera.

9.4 Vision-Based Homing

A near-distant guidance aims to improve the position of the AUV and justify its course in accordance with an underwater target. In the literature, different solutions for the problem were already presented earlier. The positioning of an underwater vehicle via image mosaicking is discussed in [19]. The estimations of vehicle motion parameters from an image series during a pipe inspection are shown in [20]. An underwater docking procedure for the AUV equipped with the CCD camera was outlined in [11]. Visually augmented navigation via multi-sensor framework employing a camera and strap-down sensor suite was described in [21]. Alternatively, one can use acoustic sonar for the object recognition [22].

The core of any vision-based navigation system is an image processing algorithm, which extracts unknown parameters of camera (vehicle) motion from a single frame or an entire image series. In computer vision, the problem is referred to as pattern matching. More formally, for a given picture I obtained by a digital camera and target pattern I_p stored in a memory, the algorithm has to determine a linear translation (Δx, Δy), an angular difference θ, and a scaling factor σ of I with respect to I_p. A conventional approach applies a cross-correlation technique for the original and pattern images [23]. If there is no rotation and scaling, then the correlation maximum corresponds to the lateral shift between two images. More sophisticated algorithms use the polar Fourier transform [24]. The methods are based on the properties of the Fourier space:

- The amplitude spectrum is invariant to the translation in the spatial domain.
- The rotation in the spatial domain corresponds to the translation in the polar Fourier domain.

Another large group of the algorithms deals with a set of interest points extracted from the original and pattern images [25, 26]. The point extraction is followed by the computation of an optimal affine transformation.

In general, the enumerated methods have a high computational complexity. Consequently, they are not suited well for the implementation by portable hardware making online data processing on-board as an expensive task for the moving vehicle. In our work for the parameter extraction, the modified log-polar transformations were used originally introduced in [27]. The developed algorithm includes several computational steps [28] (Fig. 9.6).

Step 1. The camera image I is convolved with a 2D Gaussian kernel (Eq. 9.5) in order to reduce the image noise and filter out insignificant image details.

$$G_\omega(x,y) = g_\omega(x)g_\omega(y), \text{ where } g_\omega(z) = \frac{1}{\sqrt{2\pi\omega^2}}\exp\left(-\frac{z^2}{2\omega^2}\right), z \in \{x,y\} \quad (9.5)$$

Increasing the scale parameter ω the image becomes more blurred, therefore, a greater number of soft and small image features vanish. In order to perform the computations efficiently, the separability of the 2D Gaussian function can be used

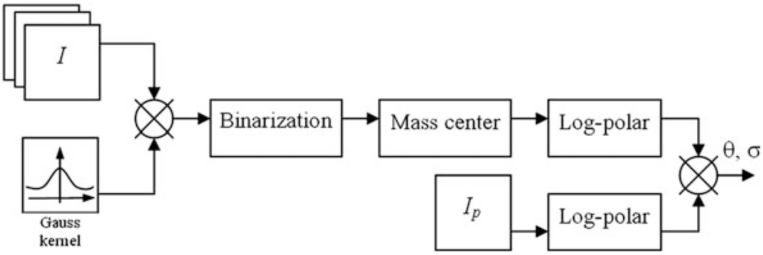


Fig. 9.6 Image processing algorithm

(Eq. 9.6), i.e. the 2D convolution converges to two successive convolutions with the 1D Gaussian functions along the *x*- and *y*-axes, respectively.

$$(I * G_{\omega}) = (I * g_{\omega}(x)) * g_{\omega}(y) \tag{9.6}$$

Step 2. A gradient image is computed and binarized in accordance with a predefined threshold T_b (Eq. 9.7).

$$I_b = \begin{cases} 1 & \text{if } (\partial I/\partial x)^2 + (\partial I/\partial y)^2 > T_b \\ 0 & \text{else} \end{cases} \tag{9.7}$$

The image derivatives are computed with the well-known Sobel operator [29]. It produces an averaged value for the horizontal and vertical derivatives by sequential image convolution with smoothing and difference masks represented by Eq. 9.8.

$$\begin{aligned} \frac{\partial I}{\partial x} &\approx I * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \\ \frac{\partial I}{\partial y} &\approx I * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}] = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \end{aligned} \tag{9.8}$$

Step 3. Let us assume that an underwater target is captured by the AUV camera. In the binary image I_b, a center of the target is found estimating a physical center of mass by Eq. 9.9.

$$x_0 = \sum_{x \in I_b} x/N \quad y_0 = \sum_{y \in I_b} y/N \tag{9.9}$$

The resulting (x_0, y_0) coordinates are used in the next step as an origin for a log-polar transform.

Step 4. In order to determine an angular difference θ and scale factor σ, the log-polar transformations are applied [27]. The transformation maps a point (*x, y*) in the Cartesian plane to a point (θ, σ) in the log-polar plane. New coordinates are computed by Eq. 9.10, where (x_0, y_0) is the origin calculated at step 3.

$$\begin{aligned} \sigma &= \log\left(\sqrt{(x - x_0)^2 + (y - y_0)^2}\right) \\ \theta &= \arctan\left((y - y_0)/(x - x_0)\right) \end{aligned} \tag{9.10}$$

The relation between the Cartesian and log-polar grid is demonstrated in Fig. 9.7. Our algorithm computes a cross-correlation of the camera I and pattern I_p images converted into (θ, σ) space. A maximum in the cross-correlation gives the estimate for parameters θ and σ. A lateral offset between the images is computed by Eq. 9.11, where (x_{p0}, y_{p0}) is the center of mass computed for the pattern image.

$$(\Delta x, \Delta y) = (x_0 - x_{p0}, y_0 - y_{p0}) \tag{9.11}$$

The outlined algorithm (steps 1–4) is computationally inexpensive: for input images of size $M \times N$ pixels the total computational complexity is required $O(MN \log(\max(M, N)))$ operations.

A translation of extracted image parameters into engine control signals is implemented by a Proportional-Integral-Derivative (PID) controller [30] given by Eq. 9.12.

$$u(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(\tau) d\tau + K_d \frac{d}{dt} \varepsilon(t) \tag{9.12}$$

Here K_p, K_i, K_d are the scale parameters for proportional, integral, and derivative parts, $u(t)$ is a control/impact function, error $\varepsilon(t)$ is a difference between measured and requested values. The PID controller works as follows (Fig. 9.8). The input for the controller is the difference between the measured and ideal characteristic of a chosen process. Changing the control function $u(t)$ according to the above equation,

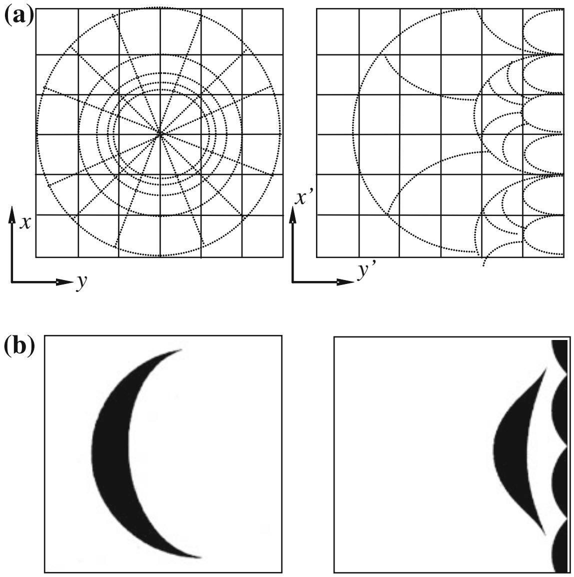


Fig. 9.7 The Cartesian and log-polar grid: **a** relationship between Cartesian and log-polar planes, **b** example of log-polar transformations for a given pattern

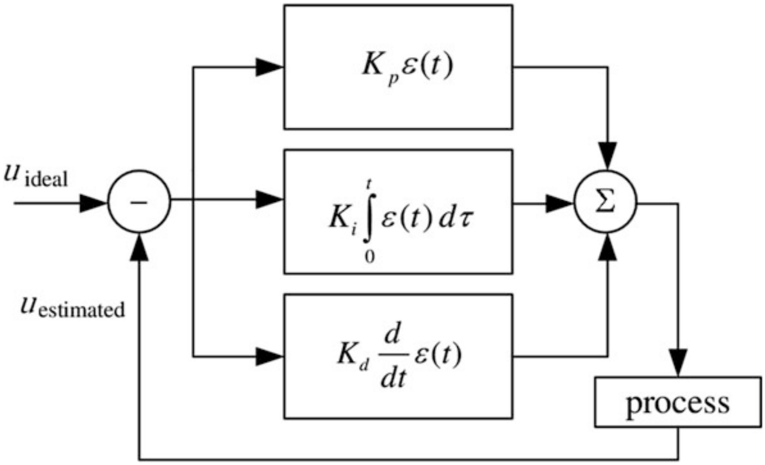


Fig. 9.8 Conventional PID controller

one minimizes the difference denoted as ε(*t*). In our model, the vehicle motion is controlled by the PID control functions $u_\varphi(t)$ and $u_v(t)$, which adjust its course φ and velocity *v*, respectively. The corresponding orientation ε_φ velocity ε_v error functions, which depend on the image parameters, are defined by Eq. 9.13.

$$\varepsilon_\varphi = \theta \,\text{and}\, \varepsilon_v = \cos(\theta)\sqrt{\Delta x^2 + \Delta y^2} \tag{9.13}$$

The error functions are defined in a way that the corresponding control function forces the vehicle to adjust its course with respect to the underwater billboard. In addition, parameter σ can be used to justify a vertical distance between the AUV and the underwater target.

9.5 Numerical Experiments

In order to examine the developed navigation methods, a series of numerical tests were carried out. A continuous time line is represented by a set of discrete moments t_i. Let us denote the position and the course of the AUV at t_i by $[x_i\ y_i]^T$ and φ_i, respectively. A program loop is organized in such a manner that on each iteration the parameters are recomputed in accordance with derived control functions $u_\varphi(t)$ and $u_v(t)$ as pointed in Eq. 9.14.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + u_v(t_i)\begin{bmatrix} \cos(\varphi_i) \\ \sin(\varphi_i) \end{bmatrix} \,\text{and}\, \varphi_{i+1} = \varphi_i + u_\varphi(t_i) \tag{9.14}$$

Starting the AUV mission, the acoustic guidance is performed first. The virtual AUV is operated by the ES-controller described in Sect. 9.3. As the AUV approaches the underwater target, the model switches to the vision-based navigation algorithm (Sect. 9.4). In our tests, a short AUV-buoy distance was used as a switch criterion between the modes. Alternatively, a cross correlation of camera image and pattern image can be applied. As soon as the convolution maximum exceeds a predefined threshold, the camera is seemed to capture the pattern. The vision-based positioning runs until the angular difference θ and lateral offset (Δx, Δy) minimize.

The simulation results are demonstrated for the acoustic navigation stage in Fig. 9.9. In our first experiment, there are neither errors in the TOF measurements nor external impacts (e.g., undercurrents). The resulting AUV path and scaled regions at the beginning and end of its trajectory are presented in Fig. 9.10. Initially the AUV is oriented in the opposite direction ($\varphi = -\pi$) from the acoustic buoy, which is located at point (0, 0). In the beginning, the vehicle moves on a circular path until the difference $\Delta\tau_i$ becomes positive. The ES-controller is turned on at the point marked by a circle on the AUV trajectory. As soon as the vehicle arrives to the buoy, it starts to make butterfly-like motions due to a delayed reaction of a signum relay.

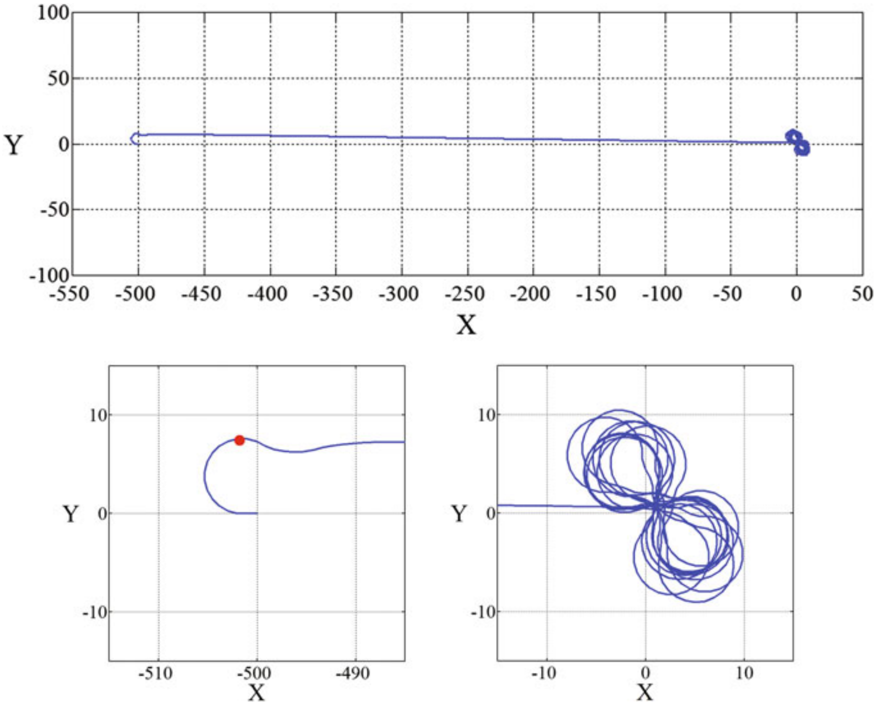


Fig. 9.9 The AUV motions controlled by the extremum search algorithm (X and Y axes are presenting in m)

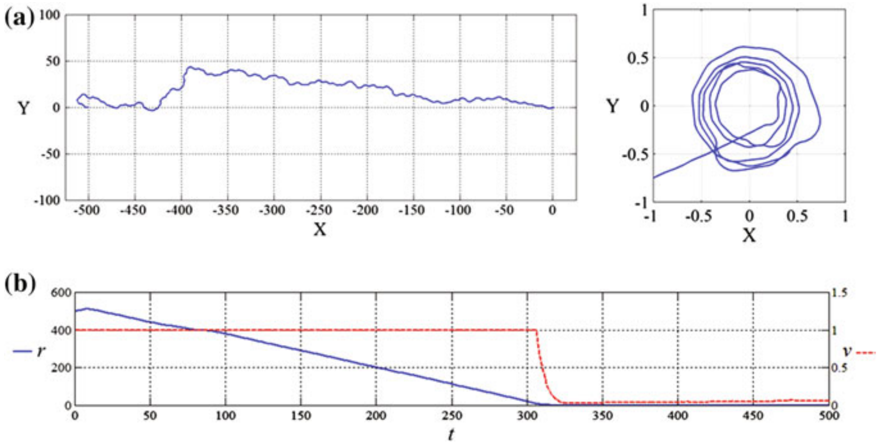


Fig. 9.10 Acoustic guidance in the presence of additive noise in ranging estimations: **a** resulting trajectory, **b** corresponding AUV-buoy distance $r(t)$ and the vehicle velocity $v(t)$ (X- and Y-axes are presenting in m, vehicle velocity is given in m per s)

In the second test, the extremum search algorithm was examined under unfavorable conditions. Additive noise in the TOF measurements was introduced (Fig. 9.11). Instrumental error e was chosen to be a uniformly distributed random variable with $|e_{\max}| = 0.002$.

Further, an undercurrent oriented parallel to the y-axis was put (see short arrows in Fig. 9.12). A velocity of the current was set to 0.1 (while $v_{\max}$ was chosen to be 1.0). In order to qualify the proposed ES-algorithm, the characteristic represented

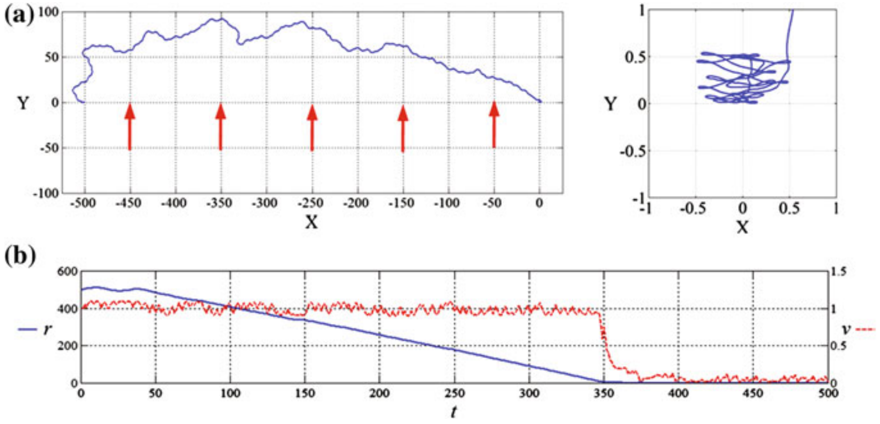


Fig. 9.11 Acoustic guidance in the presence of additive noise in ranging estimations and directional current: **a** resulting trajectory, **b** corresponding AUV-buoy distance $r(t)$ and vehicle velocity $v(t)$ (X- and Y-axes are presenting in m, vehicle velocity is given in m per s)

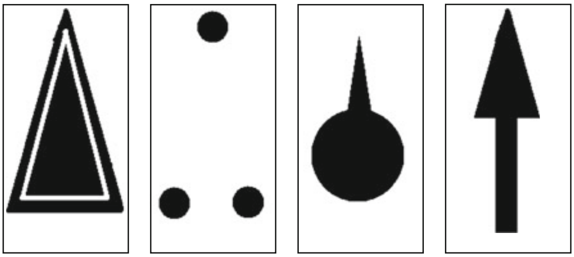


Fig. 9.12 Different target patterns

by Eq. 9.15 was introduced with r_0 is an initial AUV-buoy distance and S denotes a resulting path length.

$$Q = r_0/S \tag{9.15}$$

A simulation is stopped, if $r \leq 1$. About 300 experiments were carried out for each of the mentioned above scenarios. The averaged values of the measure Q are summarized in Table 9.1. The obtained data allows one to conclude that the ES-controller is an appropriate choice for the homing of an underwater vehicle, even in unfavorable circumstances.

Simulations for the vision-based positioning stage confirmed that the choice of the target billboard is of high importance. The pattern should define a certain orientation in space unambiguously, thus any type of symmetry should clearly be avoided. Some of the proper patterns are collected in Fig. 9.12.

In Fig. 9.13, a frame obtained by a real AUV camera and corresponding cross-correlation function are shown. The unknown image parameters are well defined by the correlation maximum (bottom plot in Fig. 9.13).

A typical operation cycle for vision-based vehicle positioning is shown in Fig. 9.14. The plot depicts three curves: φ_{current} and φ_{ideal} correspond to the current and requested vehicle bearing, $u(t)$ defines the amplitude of an impact generated by the PID controller. According to Fig. 9.14, the AUV meets requested orientation in space already on the 7th iteration. Repeating the numerical experiments more than several hundred times, a high accuracy and robustness of the presented algorithms were validated (angular error didn't exceed 5^{o} and lateral offset was in the range of centimeters).

The acoustic ranging and vision-based navigation algorithms were implemented and combined in a self-written 3D simulator (Fig. 9.15). A 3D virtual scene includes an underwater vehicle (which is equipped with a hydrophone and the CCD camera), a target, and an artificial seabed. The outer shape of the virtual AUV was borrowed from a real underwater vehicle MMT-3000 constructed in the Institute of Marine Technology Problems [31].

Table 9.1 Quality of the ES-algorithm for different simulation scenarios

Parameter	Ideal case	Noise in τ	Noise in τ and current
Q	0.95	0.76	0.71

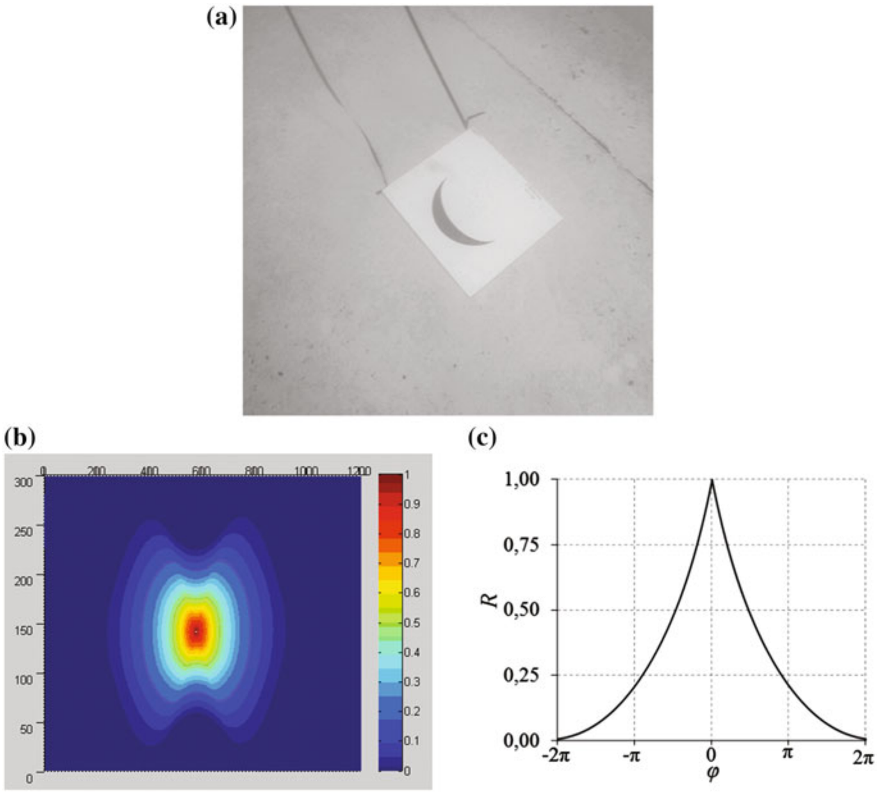


Fig. 9.13 The obtained frame by a real AUV camera: **a** typical snapshot of a real AUV camera, **b** cross-correlation function, **c** its slice of a pattern depicted in Fig. 9.7

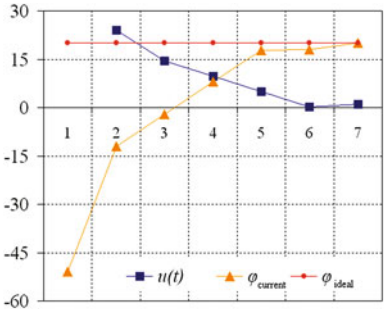


Fig. 9.14 Adjustment of vehicle course φ_{ideal} (deg) in accordance with the underwater target φ_{current} (deg) during vision-based operation

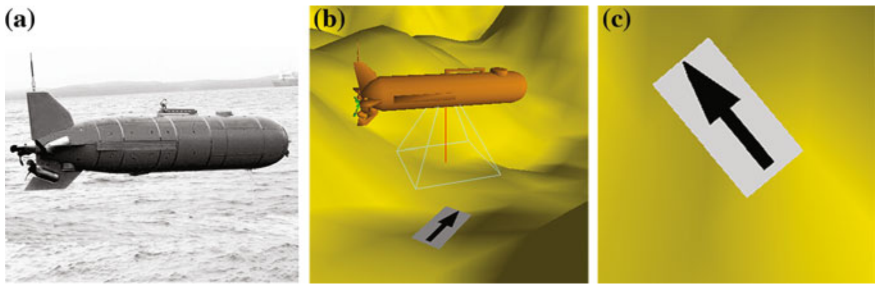
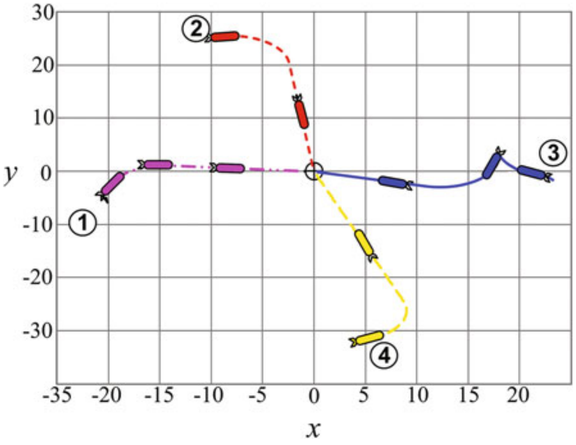


Fig. 9.15 Real and 3D virtual scenes: **a** autonomous robot MMT-3000; **b** virtual scene; **c** frame from the on-board AUV camera

Fig. 9.16 Numeric simulations of the long- and near-distant guidance combined



The results of combined numeric simulations of the long- and near-distant guidance are shown in Fig. 9.16.

9.6 Conclusion

The navigation method is proposed, which combines acoustic and vision-based guidance principles. The former is used on large AUV-target distances, the latter allows one to justify the vehicle course and improve its location with respect to an underwater target. The acoustic guidance is based on the TOF measurements from a single buoy. Among different signal exchanging modes, the one-way transmission fits the requirements of autonomy best. The vision-based positioning is performed using fast log-polar transformations. The PID controller is applied to change the vehicle position and course. The corresponding control and error functions are given. A series of numerical experiments confirms the reliability and accuracy of the developed algorithms. Several non-trivial challenges remain for the future work. So far the vehicle motion is considered in two dimensions only. Consequently, the model has to be extended for the third (depth) dimension. An open question remains as to how a vehicle should operate, if it does not discover a requested target. A billboard construction needs to be clarified as well, i.e. the question as to how to make the pattern visible in sea water is of high importance. Finally, the proposed navigation algorithms need to be carefully evaluated for the robustness in real marine environment.

Acknowledgments The authors wish to thank the staff of the Laboratory for intellectual technologies and systems at Pacific National University, namely Ivan Karabanov, Mikhail Linnik, and Fedor Bezruchko, for the fruitful discussions and suggestions during the chapter preparation stage.

References

1. National Research Council (1996) Underwater vehicles and national needs. National Academies Press, Washington DC
2. Sangekar M, Thornton B, Ura T (2012) Wide area seafloor observation using an autonomous landing vehicle with adaptive resolution capability. In: MTS/IEEE OCEANS International Conference, pp 1–9
3. Wyber RJ (2010) Overview of Australian R&D in mine imaging. In: MTS/IEEE OCEANS International Conference, pp 1–9
4. Paim PK, Jouvencel B, Lapierre (2005) A reactive control approach for pipeline inspection with an AUV. In: MTS/IEEE OCEANS International Conference 1:201–206
5. Butler L (1987) Underwater radio communication. Amateur Radio
6. Desset S, Damus R, Morash J. Bechaz, C (2003) Use of GIBs in AUVs for underwater archaeology. Sea Technol 44(12):22–27
7. Stokey R, Roup A, von Alt C, Allen B et al (2005) Development of the REMUS 600 autonomous underwater vehicle. In: MTS/IEEE OCEANS International Conference 2:1301–1304
8. Williams S, Dissanayake G, Durrant-Whyte H (2001) Towards terrain-aided navigation for underwater robotics. Adv Robot 15(5):533–549
9. Newman P, Leonard J, Rikoski R (2003) Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In: international symposium on robotics research, pp 409–420
10. Eustice RM, Whitcomb LL, Singh H, Grund M (2006) Recent advances in synchronous-clock one-way-travel-time acoustic navigation. In: MTS/IEEE OCEANS international conference, pp 1–6
11. Park JY, Jun BH, Lee PM, Oh J (2009) Experiments on vision guided docking of an autonomous underwater vehicle using one camera. Ocean Eng 36(1):48–61
12. Burdinsky IN (2012) Guidance Algorithm for an Autonomous Unmanned Underwater vehicle to a Given Target. Optoelectronics Instrum Data Process 48(1):69–74
13. Rice J (2005) Seaweb acoustic communication and navigation networks. In: International conference underwater acoustic measurements: technologies and results, pp 1–7
14. Christ RD, Wernli RL (2007) The ROV Manual. A user guide for observation class remotely operating vehicles. Elsevier Ltd, Butterworth-Heinemann, MA
15. Burdinskiy IN, Karabanov IV, Linnik MA (2010) Threshold methods of sonar pseudonoise phase-shift signal detection. In: 1st Russia and Pacific conference on computer technology and applications (Russia Pacific Computer 2010), pp 404–408

16. Virtex-4 Family Overview. http://xilinx.com/support/documentation/data_sheets/ds112.pdf. Accessed 15 June 2014
17. Burdinkiy IN, Mironov AS, Myagotin AV (2009) A multichannel correlational detector of pseudonoise hydroacoustic signals. In: 16th St Petersburg international conference on integrated navigation systems, pp 218–219
18. Bezruchko F, Burdinky I, Myagotin A (2011) Global extremum searching algorithm for the AUV guidance toward an acoustic buoy. In: MTS/IEEE OCEANS international conference, pp 1–7
19. Garcia R, Batlle J, Cufi X, Amat J (2001) Positioning an underwater vehicle through image mosaicking. In: IEEE international conference on robotics and automation, vol. 3. Seoul, Korea, pp 2779–2784
20. Branca A, Stella E, Distante A (1998) Autonomous navigation of underwater vehicles. In: MTS/IEEE OCEANS international conference 1:61–65
21. Eustice RM, Pizarro O, Singh H (2008) Visually augmented navigation for autonomous underwater vehicles. IEEE J of Oceanic Engineering 33(2):103–122
22. Yu SC, Kim TW, Asada A, Weatherwax S, Collins B, Yuh J (2006) Development of high-resolution acoustic camera based real-time object recognition system by using autonomous underwater vehicles. In: MTS/IEEE OCEANS international conference, pp 1–6
23. Cole L, Austin D, Cole L (2004) Visual object recognition using template matching. Indian J Comput Sci Eng 1(2):98–107
24. Keller Y, Shkolnisky Y, Averbuch A (2005) The angular difference function and its application to image registration. IEEE Trans Pattern Anal Mach Intell 27(6):969–976
25. Schmid C, Mohr R, Bauckhage C (2000) Evaluation of interest point detectors. Int J Comput Vision 37(2):151–172
26. Dufournaud Y, Schmid C, Horaud R (2004) Image matching with scale adjustment. Comput Vis Image Underst 93(2):175–194
27. Young D (2000) Straight lines and circles in the log-polar image. In: 11th british machine vision conference, pp 426–435
28. Myagotin A, Burdinsky I (2010) AUV positioning model employing acoustic and visual data processing. In: MTS/IEEE OCEANS international conference, pp 1–6
29. Pringle KK (1969) Visual perception by a computer. Automatic interpretation and classification of images. Academic Press, New York
30. Silva GJ, Datta A, Bhattacharyya SP (2005) PID controllers for time-delay systems. Birkhauser Boston, New York
31. Gornak VE, Inzartsev AV, Lvov OYu, Matvienko YV, Scherbatyuk APh (2006) MMT 3000—Small AUV of new series of IMTP FEB RAS. In: MTS/IEEE OCEANS international conference, pp 1–6

Chapter 10
Efficient Denoising Algorithms for Intelligent Recognition Systems

Andrey Priorov, Kirill Tumanov and Vladimir Volokhov

Abstract A great variety of electronic devices nowadays provide images with different quality, resolution, and color depth parameters. Despite of the differences, all intelligent recognition systems built from the basic image capturing components inevitably involve image preprocessing blocks, which in addition to other tasks perform the "raw" image filtration. This chapter presents the algorithms of modern filtration techniques dealing with the Principal Component Analysis (PCA) and non-local processing based on the denoising algorithms including multiple examples. Some attention is devoted to the modeling of noise on raw images. Finally, the charter ends with a discussion on the applicability of the specific filtration algorithms to modern tasks such as pattern recognition and object tracking.

Keywords Image filtration · Principal component analysis · Non-local processing · Denoising algorithm · Image reconstruction · Image recognition

10.1 Introduction

At present, the digital imaging has numerous applications in science, technology, medicine, as well as in everyday life—in digital cameras and mobile phones. The latter is due to the known fact that visual sensors provide humans with the most information about the external world. That is why methods of digital image processing are extensively studied. In general, the typical devices, which form digital

A. Priorov · V. Volokhov
P.G. Demidov Yaroslavl State University, 14 Sovetskaya Str., Yaroslavl 150000,
Russian Federation
e-mail: andcat@yandex.ru

V. Volokhov
e-mail: volokhov@piclab.ru

K. Tumanov (✉)
Maastricht University, Minderbroedersberg 4-6, 6211 LK Maastricht, The Netherlands
e-mail: tumanov@susqu.edu

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_10

images, contain of lenses and semiconductor sensors to capture a projected scene. These components cause distortions such as simple geometrical distortion, degradation, and noise. Therefore, a sophisticated denoising [1–3], a sharpening, and the color correction algorithms are crucial for obtaining the high-quality digital images.

These algorithms are used for various noises cancellation in the majority of cases of an Additive White Gaussian Noise (AWGN). According to [2], the approaches for the AWGN cancellation in digital images include a local processing, a non-local processing, a point-wise processing, and a multi-point processing. This classification is not acknowledged by all scholars, but it is used here to ease the analysis of the discussed topics. Each of the given filtration methods has its specific pros and cons mainly related to the quality of the reconstructed images and computational costs of their processing algorithms. An analysis of the contemporary scientific and technical literature shows that the PCA [4, 5], a non-local processing [6, 7], and their combinations are among the perspective approaches applied to the task of digital image reconstruction. Their filtration efficiency leads to implement these algorithms in modern recognition systems.

The chapter is organized in follow manner. Section 10.2 involves two-stage PCA filtration scheme. The sequential and parallel filtration schemes based on the PCA with a non-local processing are discussed in Sect. 10.3. The image filtration using a non-local PCA is explained in Sect. 10.4. Section 10.5 provides Bayer patterns filtration based on a non-local PCA. The application of denoising algorithms to the task of license plate recognition is situated in Sect. 10.6. Conclusion is drawn in Sect. 10.7.

10.2 Two-Stage PCA Filtration Scheme

The spatially adaptive PCA based image denoising scheme was proposed by Muresan and Parks in 2003 [8]. The authors of the chapter proposed a modification of this research [9]. A scheme of the modified filtration algorithm is shown in Fig. 10.1.

Assume that an analyzed digital image **x** is affected by the AWGN **n** with zero mean and variance σ^2. Note that during the AWGN modeling a noised digital image passed through the preliminary processing block with a characteristics of a quantization with saturation. The scheme includes two stages of image estimation called as a primary "coarse" evaluation and a second "fine" evaluation. The first stage of the PCA filtration algorithm involves the following steps:

1. Assume the AWGN variance σ^2 to be known.
2. Divide the input noisy image into a set of overlapping blocks. Each of them contains the training, denoise, and overlap regions. Dimensions of these regions may vary.
3. In the training region, select all possible blocks size of $l^{\mathrm{I}} \times l^{\mathrm{I}}$ (training vectors). The last ones are column vectors each $\left(l^{\mathrm{I}}\right)^2$ in length. They allow us to form a selective matrix $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$ with a size of $\left(l^{\mathrm{I}}\right)^2 \times n^{\mathrm{I}}$, which contains the mentioned column vectors. Here n^{I} is a number of training vectors found in the training region.

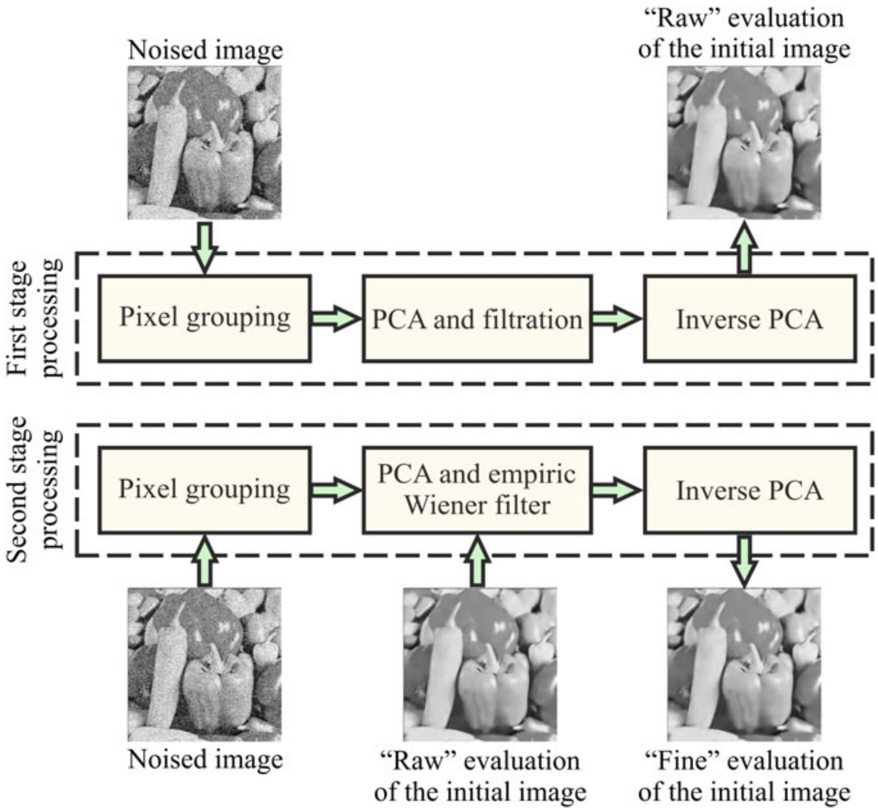

"Raw" evaluation of the initial image
Noised image
First stage processing
Pixel grouping
PCA and filtration
Inverse PCA
Second stage processing
Pixel grouping
PCA and empiric Wiener filter
Inverse PCA
Noised image
"Raw" evaluation of the initial image
"Fine" evaluation of the initial image

Fig. 10.1 Two-stage PCA-based filtration

4. Based on the preliminary centered $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$ matrix, create a covariance matrix $\mathbf{Q}^{\mathrm{I}}_{\bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}}$, in which $\bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}$ is a centered selective matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$. Then, for the $\mathbf{Q}^{\mathrm{I}}_{\bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}}$ matrix, find eigenvalues and corresponding eigenvectors (principal components of data comprised in the $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$ matrix). Finally, create an orthogonal transform matrix $\mathbf{P}^{\mathrm{I}}_{\mathbf{y}}$.
5. For each $i = 1, 2, \ldots, (l^{\mathrm{I}})^2$ and $j = 1, 2, \ldots, n^{\mathrm{I}}$ find projections (transform coefficients) $(\bar{Y}^{\mathrm{I}})_i^j$ of vectors contained in the matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$, on eigenvectors found in the previous step by using Eq. 10.1, where $(\bar{Y}^{\mathrm{I}})_i^j = (\bar{X}^{\mathrm{I}})_i^j + (N^{\mathrm{I}})_i^j$ $((\bar{Y}^{\mathrm{I}})_i^j$ as an ith projection of vector with index j from the matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$ on eigenvectors of the matrix $\mathbf{Q}^{\mathrm{I}}_{\bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}}$ equals a sum of an ith projection of undistorted data vector with index j $\left(\hat{\bar{X}}^{\mathrm{I}}\right)_i^j$ and an ith projection of noise vector with index j $(N^{\mathrm{I}})_i^j$).

$$\bar{\mathbf{Y}}^{\mathrm{I}} = \mathbf{P}_{\mathbf{y}}^{\mathrm{I}}\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}} = \begin{bmatrix} (\bar{Y}^{\mathrm{I}})_1^1 & (\bar{Y}^{\mathrm{I}})_1^2 & \cdots & (\bar{Y}^{\mathrm{I}})_1^n \\ (\bar{Y}^{\mathrm{I}})_2^1 & (\bar{Y}^{\mathrm{I}})_2^2 & \cdots & (\bar{Y}^{\mathrm{I}})_2^n \\ \vdots & \vdots & \ddots & \vdots \\ (\bar{Y}^{I})_{(l^{\mathrm{I}})^2}^1 & (\bar{Y}^{\mathrm{I}})_{(l^{\mathrm{I}})^2}^2 & \cdots & (\bar{Y}^{\mathrm{I}})_{(l^{\mathrm{I}})^2}^n \end{bmatrix} \tag{10.1}$$

Note, that there is no line above the $(N^{\mathrm{I}})_i^j$ component. The reason for this is that the centered and non-centered noising matrices have the same projections $(N^{\mathrm{I}})_i^j$ because the used AWGN model has a zero mean.

6. Evaluate the received projections with optimal Linear Minimum Mean-Square Error (LMMSE) estimator [8] calculated by Eq. 10.2, where σ^2 is a noise variance and σ_i^2 is a variance of ith projection of undistorted vectors $j = 1, 2, \ldots, n^{\mathrm{I}}$.

$$\left(\hat{\bar{X}}^{\mathrm{I}}\right)_i^j = \frac{\sigma_i^2}{\sigma_i^2 + \sigma^2} \cdot (\bar{Y}^{\mathrm{I}})_i^j \tag{10.2}$$

Here a variance σ_i^2 can be found using a maximum likelihood estimator [8] determined by Eq. 10.3.

$$\hat{\sigma}_i^2 = \max\left[0, \frac{1}{n}\sum_{j=1}^{n}\left((\bar{Y}^{\mathrm{I}})_i^j\right)^2 - \sigma^2\right] \tag{10.3}$$

7. Based on the processed data $\left(\hat{\bar{X}}^{\mathrm{I}}\right)_i^j$, reconstruct an evaluation $\hat{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}}$ of non-noised data matrix $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$. Then reconstruct a separate processed image area. In this case, first of all, a training region is reconstructed by inserting the training vectors into their spatial positions considering the overlaps. The training vectors kept as column vectors in the matrix $\hat{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}}$ are again transformed into blocks size of $l^{\mathrm{I}} \times l^{\mathrm{I}}$ prior the insertion into the training region. Note, that an overlapping region is averaged using simple arithmetic averaging. Then, after the reconstruction of the training region extract the smaller denoise region from it.

Repetition of the similar operations for the rest denoise regions considering the overlaps allows the processing of the whole image and the obtaining of a primary "coarse" evaluation $\hat{\mathbf{x}}^{\mathrm{I}}$ of the non-noised image $\mathbf{x}$. While doing this, the processed denoise regions are inserted into their spatial positions of the image $\hat{\mathbf{x}}^{\mathrm{I}}$, and the overlap region is arithmetically averaged.

The second stage of the PCA filtration algorithm includes the steps mentioned below:

1. Using the noisy image **y**, repeat steps 2–5, discussed in the first stage. Sizes of the training, denoise, and overlap regions as well as training vectors change accordingly.
2. Process the received projections by using Eq. 10.4, where $(\bar{Y}^{\mathrm{II}})_i^j = (\bar{X}^{\mathrm{II}})_i^j + (N^{\mathrm{II}})_i^j$ (an *i*th projection of vector with index *j* from a matrix $\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{II}}$ on eigenvectors of a matrix $\mathbf{Q}_{\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{II}}}^{\mathrm{II}}$) is a sum of an *i*th projection of undistorted data vector with index *j* and an *i*th projection of noise vector with index *j*, $\left(\left(\hat{\bar{X}}^{\mathrm{I}}\right)^{\mathrm{II}}\right)_i^j = (\bar{X}^{\mathrm{II}})_i^j + \left(\left(\hat{N}^{\mathrm{I}}\right)^{\mathrm{II}}\right)_i^j$ (an *i*th projection of vector with index *j* from matrix $\bar{\mathbf{S}}_{\hat{\mathbf{x}}^{\mathrm{I}}}^{\mathrm{II}}$ on eigenvectors of a matrix $\mathbf{Q}_{\bar{\mathbf{S}}_{\hat{\mathbf{x}}^{\mathrm{I}}}^{\mathrm{II}}}^{\mathrm{II}}$) is a sum of an *i*th projection of undistorted data vector with index *j* and *i*th projection of residual noise vector with index *j*.

$$\left(\hat{\bar{X}}^{\mathrm{II}}\right)_i^j = \frac{\left|\left(\left(\hat{\bar{X}}^{\mathrm{I}}\right)^{\mathrm{II}}\right)_i^j\right|^2}{\left|\left(\left(\hat{\bar{X}}^{\mathrm{I}}\right)^{\mathrm{II}}\right)_i^j\right|^2 + \sigma^2} \cdot \left(\bar{Y}^{\mathrm{II}}\right)_i^j \tag{10.4}$$

   Equation 10.4 is a formula of an empirical Wiener filter. Note, that in early works in digital image processing [10] Yaroslavsky showed a great potential of empirical Wiener filter as an operator for transform coefficients reduction.
3. After application the same operations discussed in step 7 of the first stage of processing, a second "fine" evaluation $\hat{\mathbf{x}}^{\mathrm{II}}$ of the non-noised image will be received.

The researches show that the main advantages of the described algorithm are:

- A versatility of data processing due to two-stage structure.
- The absence of residual noise on the reconstructed images in comparison to the "classic" implementation.

On the other hand, the both "classic" and modified implementations of the filtration algorithm based on the PCA suffer from the Gibbs effect on the edges of high-contrast objects in a digital image as well as from a more noticeable distortion of the main object contours as opposed to some modern denoising algorithms.

10.3 Sequential and Parallel Filtration Schemes Based on PCA and Non-local Processing

Among the strategies aimed for decrease of the Gibbs effect and distortion of the object edges on a processed digital image, notable two strategies are popular:

- The sequential filtration scheme, where an input noisy image is processed and then its “raw” estimation is filtered.
- The parallel filtration scheme, which is based on the simultaneous processing of an input noisy image by a series of denoising algorithms. Then a final estimation of the image is formed by a mixing technique.

In the presented chapter the first and the second processing stages completely resemble those in the aforementioned modification of the PCA-based digital image filtration algorithm, which forms an input for non-local denoising algorithm in both sequential and parallel filtration schemes. Namely it provides a first estimation $\hat{\mathbf{x}}^{\mathrm{I}}$ of the initial image $\mathbf{x}$, which is used for calculation of the second estimation $\hat{\mathbf{x}}^{\mathrm{II}}$ of an non-noised image $\mathbf{x}$.

Let us discuss the sequential and parallel filtration schemes in Sects. 10.3.1 and 10.3.2, respectively. Some applications of the filtration methods are located into Sect. 10.3.3.

10.3.1 Sequential Filtration Scheme

A scheme of a sequential filtration [9] is shown in Fig. 10.2.

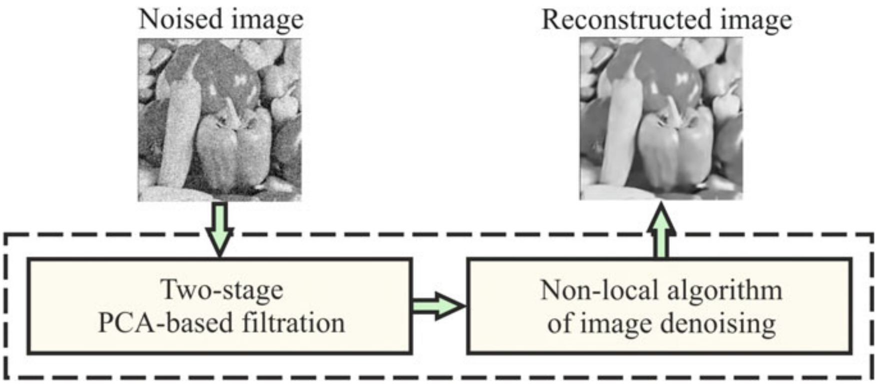


Fig. 10.2 Sequential digital image filtration

The implementation of third stage is based on the approach proposed by Buades et al. in 2005–2008 [6, 11–13]. The following procedure provides a specification of the steps taken to calculate non-local means and shown in Fig. 10.2.

1. For each processed pixel $\hat{x}^{\mathrm{II}}(i,j)$ of an image $\hat{\mathbf{x}}^{\mathrm{II}}$, a square neighborhood (similarity) area centered on a processed pixel with fixed dimensions $l^{\mathrm{III}} \times l^{\mathrm{III}}$ is considered.
2. Similarity of the processed pixel $\hat{x}^{\mathrm{II}}(i,j)$ of an image $\hat{\mathbf{x}}^{\mathrm{II}}$ and a pixel $\hat{x}^{\mathrm{II}}(k,l)$ (around which a square neighborhood is also formed) of the same image is evaluated, using a weighted Euclidean distance $\sum_{m,n\in N} g_a(m,\, n)\cdot [\hat{x}^{\mathrm{II}}(i+m,\, j+n) - \hat{x}^{\mathrm{II}}(k+m,\, l+n)]^2$, where N is a square similarity area centered in a pixel with coordinates (0, 0), $g_a(m,\, n)$ are additional weighting coefficients defined as a Gaussian kernel coefficients with a standard deviation a.
3. A weight of a pixel $\hat{x}^{\mathrm{II}}(k,l)$ similar to $\hat{x}^{\mathrm{II}}(i,j)$ is evaluated for the calculation a final estimation of the pixel $\hat{x}^{\mathrm{III}}(i,j)$, using the Eq. 10.5, where h^{III} is a filtration parameter, which affects a digital image filtration degree.

$$w_{h^{\mathrm{III}}}(i,\, j,\, k,\, l) = e^{-\frac{\sum_{m,n\in N} g_a(m,\, n)\cdot\left[\hat{x}^{\mathrm{II}}(i+m,\, j+n) - \hat{x}^{\mathrm{II}}(k+m,\, l+n)\right]^2}{\left(h^{\mathrm{III}}\right)^2}} \tag{10.5}$$

It can be calculated as shown in Eq. 10.6, where c^{III} is an empirically determined positive constant with a value <1, σ^2 is a variance of an AWGN added to the initial image $\mathbf{x}$, $\sigma^2_{\mathbf{y}-\hat{\mathbf{x}}^{\mathrm{II}}}$ is a variance of a differential signal between a noisy image $\mathbf{y}$ and a second estimation $\hat{\mathbf{x}}^{\mathrm{II}}$ of the image $\mathbf{x}$.

$$h^{\mathrm{III}} = C^{\mathrm{III}} \cdot \sqrt{\sigma^2 - \sigma^2_{\mathbf{y}-\hat{x}^{\mathrm{II}}}} \tag{10.6}$$

4. The final non-local estimation of the processed pixel $\hat{x}^{\mathrm{II}}(i,j)$ is formed by using Eq. 10.7, where coefficients g(·) are calculated by Eq. 10.8.

$$\hat{x}^{\mathrm{III}}(i,j) = \sum_{k,l} g_{h^{\mathrm{III}}}(i,\, j,\, k,\, l)\hat{x}^{\mathrm{II}}(k,\, l) \tag{10.7}$$

$$g_{h^{\mathrm{III}}}(i,\, j,\, k,\, l) = \frac{w_{h^{\mathrm{III}}}(i,\, j,\, k,\, l)}{\sum_{k,l} w_{h^{\mathrm{III}}}(i,\, j,\, k,\, l)} \tag{10.8}$$

Repeating the listed steps for the remaining pixels of an image $\hat{\mathbf{x}}^{\mathrm{II}}$ a third, “fine” estimation $\hat{\mathbf{x}}^{\mathrm{III}}$ of an initial image $\mathbf{x}$ will be obtained.

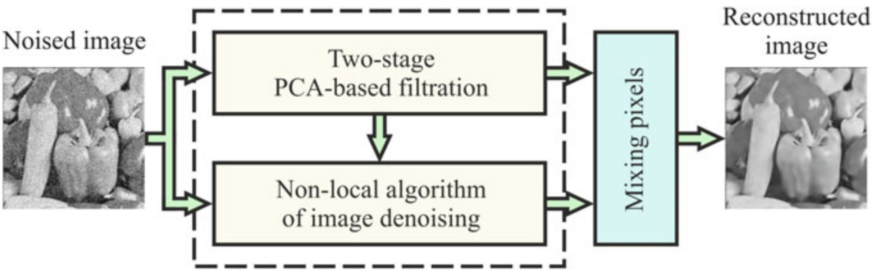


Fig. 10.3 Parallel digital image filtration

10.3.2 Parallel Filtration Scheme

A scheme of a parallel filtration [9, 14] is shown in Fig. 10.3.

For implementation of the third stage, the described non-local processing technique was used. Note that, the block “Non-local algorithm of image denoising” shown in Fig. 10.3 on contrary to the analogous block in Fig. 10.2 processes a noisy image $\mathbf{y}$, not a second estimation $\hat{\mathbf{x}}^{\mathrm{II}}$ of an non-noised image $\mathbf{x}$, as it is done in Eq. 10.7. Wherein weight of a pixel $y(k,l)$ similar to a processed pixel $y(i,j)$ in a final estimation $\hat{\mathbf{x}}^{\mathrm{II}}$ of the received non-noised image $\mathbf{x}$ as an output of the block is calculated by using Eq. 10.5. In calculations of the parameter h^{III}, a signal’s variance component $\mathbf{y} - \hat{\mathbf{x}}^{\mathrm{II}}$ in Eq. 10.6 was not considered.

The fourth stage provides a final “fine” evaluation $\hat{\mathbf{x}}^{\mathrm{IV}}$ of a non-noised image $\mathbf{x}$ by using a “Mixing pixels” procedure shown as a separate block in Fig. 10.3. In the present chapter, this procedure was implemented in the simple form, described by Eq. 10.9, where d^{II} and d^{III} are weighting constants with values less than 1. In this research, both of weighting constants were supposed to be equal 0.5.

$$\hat{\mathbf{x}}^{\mathrm{IV}} = d^{\mathrm{II}} \cdot \hat{\mathbf{x}}^{\mathrm{II}} + d^{\mathrm{III}} \cdot \hat{\mathbf{x}}^{\mathrm{III}} \tag{10.9}$$

Modeling results show the following advantages of the discussed filtration schemes over the modification of the PCA-based digital image filtration algorithm:

- The sequential scheme—a significant Gibbs effect reduction in the resulting images while object edges distortion remains at the same level or increases.
- The parallel scheme—both Gibbs effect and distortion of the main object edges are significantly reduced in a processed image.

The numerical results of the described algorithms are shown in Fig. 10.4 on an example of a test image “Baboon” [15]. Specific values of Peak Signal-to-Noise Ratio (PSNR) and Mean Structural Similarity Index Map (MSSIM) are shown for each algorithm. Hereinafter, the best image reconstruction results based on the criteria of the PSNR [16] and the MSSIM [17] are marked in bold.

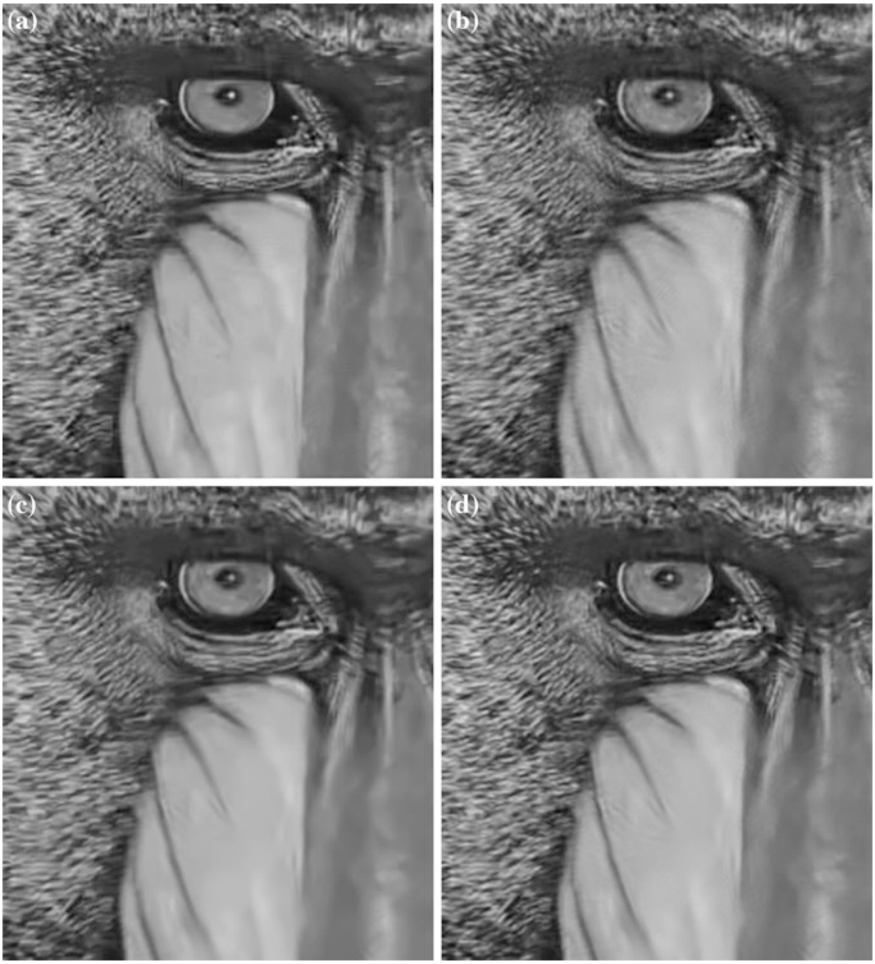


Fig. 10.4 Fragments of AWGN-affected “Baboon” ($\sigma = 25$) test image reconstruction results (in brackets PSNR, dB and MSSIM): **a** block matching and 3D filtering (BM3D) [18] (25.46 dB, 0.745), **b** modification of PCA-based filtering (25.52 dB, 0.748), **c** sequential scheme (24.90 dB, 0.714), **d** parallel scheme (**25.57 dB, 0.750**)

10.3.3 Applications of Filtration Methods

The modern AWGN filtration methods applied to grayscale images may be additionally used in a series of other digital image processing tasks. Examples of such tasks are: an AWGN-affected color image filtration, a mixed noise filtration from grayscale and color images, and a removal of blocking artifacts caused by JPEG compression. Let us briefly consider these tasks.

A filtration of color images is an issue of the day for various practical applications. That is why there are numerous solutions to it. As a possible approach, in

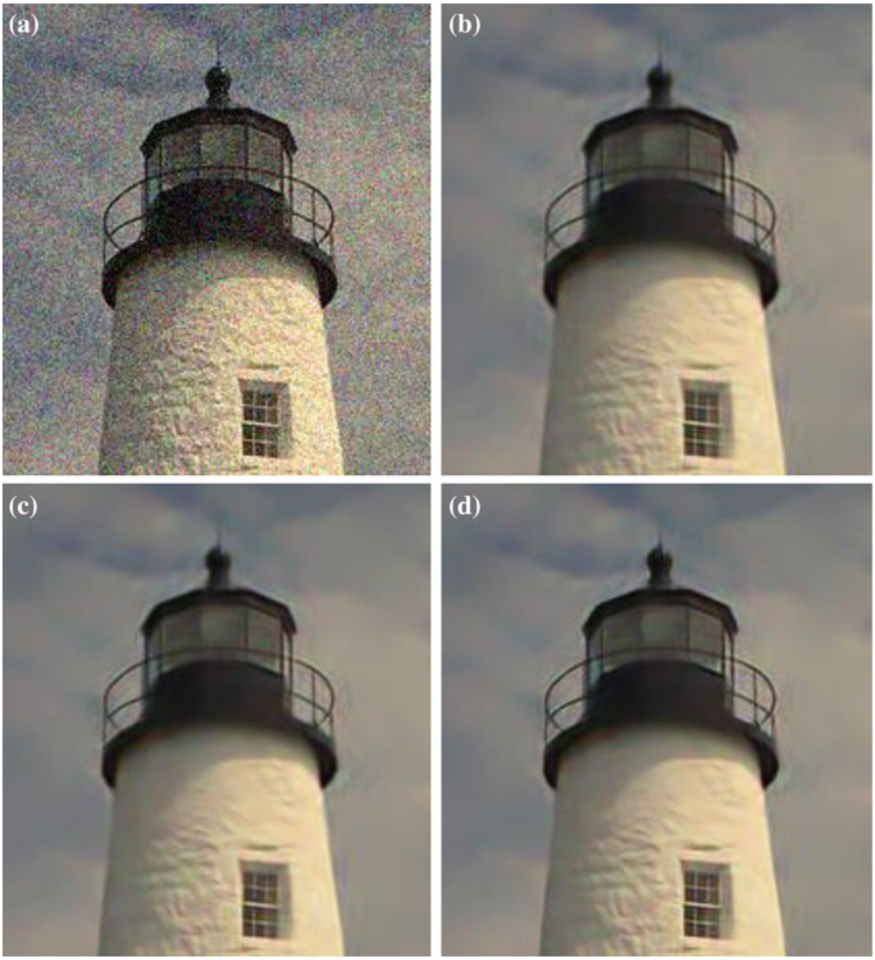


Fig. 10.5 Fragments of AWGN-affected “Lighthouse” (σ = 25) color test image reconstruction results (in brackets PSNR, dB and MSSIM): **a** noised image (20.31 dB, 0.468), **b** modification of PCA-based filtering (29.34 dB, **0.808**), **c** sequential scheme (29.20 dB, 0.792), **d** parallel scheme (**29.63** dB, 0.805)

this chapter no transition from RGB image to an image with separated brightness and color information was performed, and an AWGN was added separately to each channel with the same characteristics. The described algorithms processing results are shown in Fig. 10.5 on an example of a color test image “Lighthouse” from the Kodak image database [19].

The discussed AWGN model may be complicated using the mixed noise model. An example of such model was proposed by Hirakawa and Parks in 2006 [20] to characterize noise of Complementary Metal-Oxide Semiconductor (CMOS) matrices. The model is described by Eq. 10.10, where σ_1 and σ_2 are the constants, which determine a noisiness degree, and $\mathbf{n}$ is an AWGN with zero mean and $\sigma = 1$.

$$\mathbf{y} = \mathbf{x} + (\sigma_1 + \sigma_2 \mathbf{x})\mathbf{n} \tag{10.10}$$

Because of the irregular character of noise variance in the mixed noise model, which is explained by the dependency of noise from the initial signal (as shown in Eq. 10.10), a direct application of the described schemes is impossible. For this reason, a generalized homomorphic filtration method [21] proposed by Ding and Venetsanopoulos in 1987 was used. The idea of this method is in using a logarithm-type transform to interpret noised data $\mathbf{y}$ as a sum of an initial non-noised signal and the AWGN, process them with described filtration schemes and then reconstruct the data with the inverse transform. The processing results are shown on an example of a test image "Boats" [15] in Fig. 10.6.

Consider the situation, when the noise model $\mathbf{y} = \mathbf{x} + \mathbf{n}$ depicts a result of the image compression by JPEG algorithm (Fig. 10.7a [22, 23]). The task was formulated as a situation, where an image compression using JPEG algorithm is used as a noise model [22, 23]. In this case, a noise component $\mathbf{n}$ may be treated as a result of distortion connected with blocking artifacts in a digital image. Then a solution to this task may be found as variance σ^2 of a noise component $\mathbf{n}$. A possible way of finding σ^2, using a priori knowledge about a quantization matrix of JPEG standard coefficients, is shown, for example, in [24]. The processing results are shown on an example of a test image "Cameraman" for a case of high image compression (compression coefficient $Q = 6$) in Fig. 10.7.

10.4 Image Filtration Using Non-local PCA

One of the major drawbacks of the filtration algorithms discussed in the previous section is that procedures of the PCA basis and non-local search of similar blocks in fact are performed independently from each other. The present section describes a filtering algorithm, which integrates those two procedures in an efficient way.

Assume that an analyzed digital image $\mathbf{x}$ is affected by the AWGN $\mathbf{n}$ with zero expectation and σ^2. The following are processing stages comprising the mentioned efficient filtration algorithm.

First stage.

1. Noise variance σ^2 of an input image $\mathbf{y} = \mathbf{x} + \mathbf{n}$ is assumed to be known.
2. A wavelet preprocessing [25] of the image $\mathbf{y}$ is performed. It provides a preliminary estimation of a non-noised image $\mathbf{x}$. The use of this estimation allows to perform a higher quality statistical data search, which is needed for digital image reconstruction.
3. The preliminary estimation of a non-noised image is split in a series of overlapping reference blocks. In each iteration of the algorithm, the search and overlapping

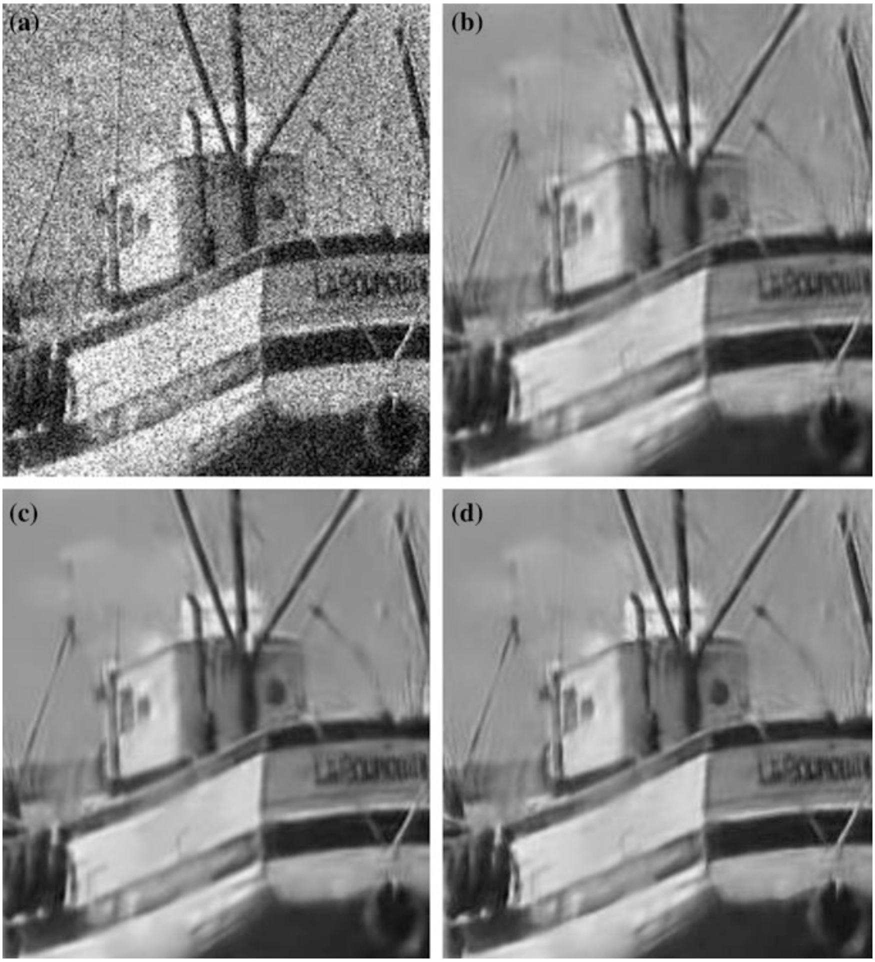


Fig. 10.6 Fragments of mixed noise affected “Boats” (σ_1 = 25, σ_2 = 0.01) test image reconstruction results (in brackets PSNR, dB and MSSIM): **a** noised image (16.68 dB, 0.229), **b** modification of PCA-based filtering (27.14 dB, 0.723), **c** sequential scheme (27.13 dB, 0.721), **d** parallel scheme (**27.52** dB, **0.730**)

regions are defined as well as candidate-blocks, which are equal in size with a reference block. Sizes of the search and the overlapping regions may vary.

4. For a search region, selected on the preliminary estimation of a non-noised image, a block-matching procedure is performed in order to determine the coordinates of similar blocks of size $l^{\mathrm{I}} \times l^{\mathrm{I}}$, which are later taken from the input image $\mathbf{y}$. These similar blocks, depicted as column vectors, $(l^{\mathrm{I}})^2$ in length, allow to form a selective matrix $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$ with a size of $(l^{\mathrm{I}})^2 \times n^{\mathrm{I}}$, which contains the mentioned column vectors. Here n^{I} is a number of training vectors found in the train region.

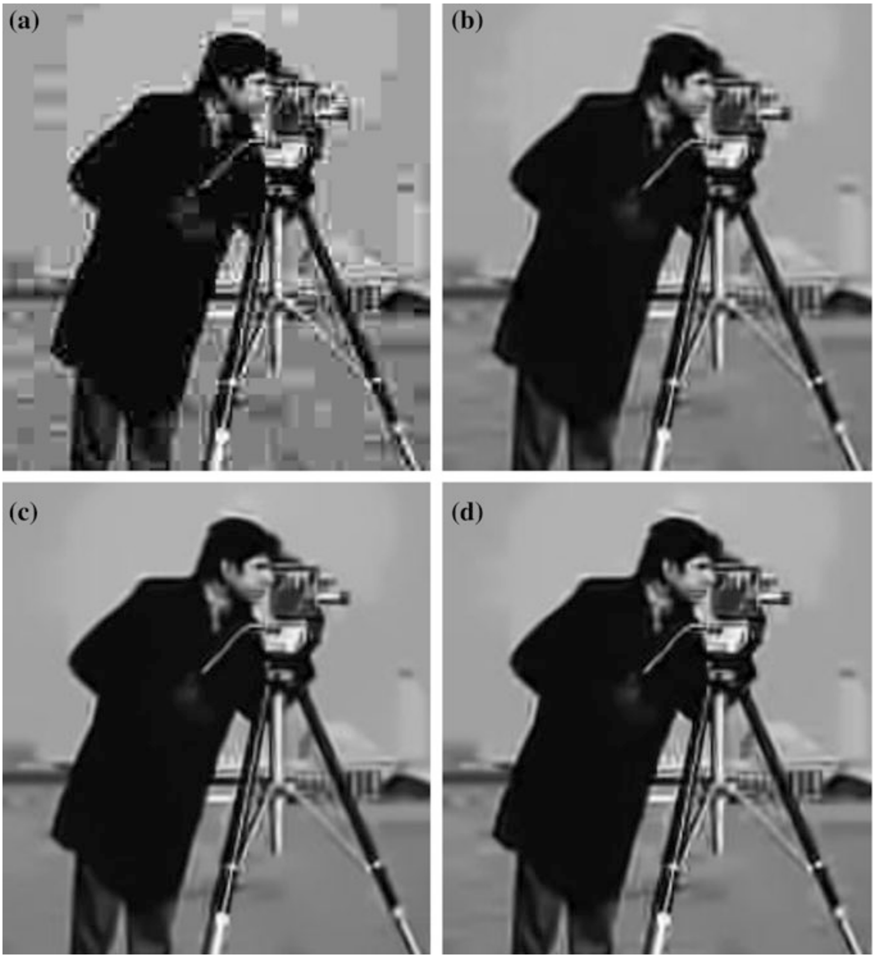


Fig. 10.7 Fragments of JPEG-compressed “Cameraman” ($Q = 6$) test image reconstruction results (in brackets PSNR, dB and MSSIM): **a** image compressed with JPEG (25.03 dB, 0.756), **b** modification of PCA-based filtering (25.79 dB, 0.783), **c** sequential scheme (25.76 dB, 0.782), **d** parallel scheme (**25.92** dB, **0.788**)

5. Based on the preliminary centered $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$ matrix, create a covariance matrix $\mathbf{Q}_{\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}}}^{\mathrm{I}}$, in which $\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}}$ is a centered selective matrix $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$. Then, for the $\mathbf{Q}_{\bar{\mathbf{S}}_{\mathbf{y}}^{\mathrm{I}}}^{\mathrm{I}}$ matrix find eigenvalues and corresponding eigenvectors (principal components of data comprised in the $\mathbf{S}_{\mathbf{y}}^{\mathrm{I}}$ matrix). In the present algorithm, the use of a truncated basis of eigenvectors is proposed. While computing an eigenvectors’ basis, each eigenvector matches its own eigenvalue—a variance along the vector’s direction. According to the PCA properties, these values are aligned in a decreasing order. Based on that, it is possible to introduce a variance threshold—an

indication of the lowest acceptable signal value. After the threshold filtering, the remaining values are $\lambda : \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{p^{\mathrm{I}}}$, where $p^{\mathrm{I}} \leq (l^{\mathrm{I}})^2$. These values correspond to the eigenvectors $\varphi : \varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_{p^{\mathrm{I}}}$.

For each $i = 1, 2, \ldots, (l^{\mathrm{I}})^2$ and $j = 1, 2, \ldots, n^{\mathrm{I}}$, find the projections (the transform coefficients) $(\bar{Y}^{\mathrm{I}})_i^j$ of vectors contained in the matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$ on eigenvectors found in the previous step by using Eq. 10.11, where $(\bar{Y}^{\mathrm{I}})_i^j = (\bar{X}^{\mathrm{I}})_i^j + (N^{\mathrm{I}})_i^j$ (an ith projection of vector with index j from the matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$on eigenvectors of the matrix $\mathbf{Q}^{\mathrm{I}}_{\bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}}$ is a sum of an ith projection of undistorted data vector with index j and an ith projection of noise vector with index j.

$$\bar{\mathbf{Y}}^{\mathrm{I}} = \mathbf{P}^{\mathrm{I}}_{\mathbf{y}} \bar{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}} = \begin{bmatrix} (\bar{Y}^{\mathrm{I}})_1^1 & (\bar{Y}^{\mathrm{I}})_1^2 & \cdots & (\bar{Y}^{\mathrm{I}})_1^n \\ (\bar{Y}^{\mathrm{I}})_2^1 & (\bar{Y}^{\mathrm{I}})_2^2 & \cdots & (\bar{Y}^{\mathrm{I}})_2^n \\ \vdots & \vdots & \vdots & \vdots \\ (\bar{Y}^{\mathrm{I}})_{p^{\mathrm{I}}}^1 & (\bar{Y}^{\mathrm{I}})_{p^{\mathrm{I}}}^2 & \cdots & (\bar{Y}^{\mathrm{I}})_{p^{\mathrm{I}}}^n \end{bmatrix} \tag{10.11}$$

6. Evaluate the received projections with the LMMSE estimator in Eq. 10.3.

Based on the processed data $\left(\hat{\bar{X}}^{\mathrm{I}}\right)_i^j$, an estimation $\hat{\mathbf{S}}^{\mathrm{I}}_{\mathbf{y}}$ of a non-noised data matrix $\mathbf{S}^{\mathrm{I}}_{\mathbf{y}}$ is reconstructed. Then each separate processing region of the image is reconstructed by using Eq. 10.12 for weighting a number of block estimations, where (i,j) are coordinates of the reconstructed pixel, $\chi^{\mathrm{I}}_{k,l}(i,j)$ is a coefficient equal to 0 or 1, showing the attachment of the pixel to a given reference block.

$$\hat{x}^{\mathrm{I}}(i,j) = \frac{\sum_k \sum_l \hat{x}^{\mathrm{I}}_{k,l}(i,j)\chi^{\mathrm{I}}_{k,l}(i,j)}{\sum_k \sum_l \chi^{\mathrm{I}}_{k,l}(i,j)} \tag{10.12}$$

As a result, a first estimation of the input image is obtained.

Second stage.

1. Repeat steps 3–8 from the first stage. The search of similar blocks' coordinates is performed based on the first image estimation, and the blocks themselves are taken from the input image **y**. However, the algorithm's input parameters—search, overlapping regions, and the reference block's sizes, are changed. The truncation of the PCA vectors' basis is not performed.
2. The processing of the resulting number of projections is done using Eq. 10.4. As an output, a second estimation of the input image is formed.

Third stage (post-processing).

1. Based on the received second estimation of an input image, a non-local averaging is performed. The post-processing algorithm searches a set of blocks for each processed block, averaging over which results in a given block. Then this

average replaces an initial processing block in an input image. The higher is a number of blocks in the set the lower is a residual noise after averaging but, on the other hand, the less is an averaging precision.

2. In the AWGN modeling, it is assumed that a noised digital image passed through the preliminary processing block with characteristics of a quantization with saturation. In this case, a noise does not possess a zero mean characteristics, and strictly speaking is no longer Gaussian. It is desired to minimize a standard deviation of the reconstructed image. This may be done using a histogram correction of an output image. The result of the stage is a final estimation of a non-noised image **x**.

The processing results are shown in Fig. 10.8 on an example of a test image "Penelope".

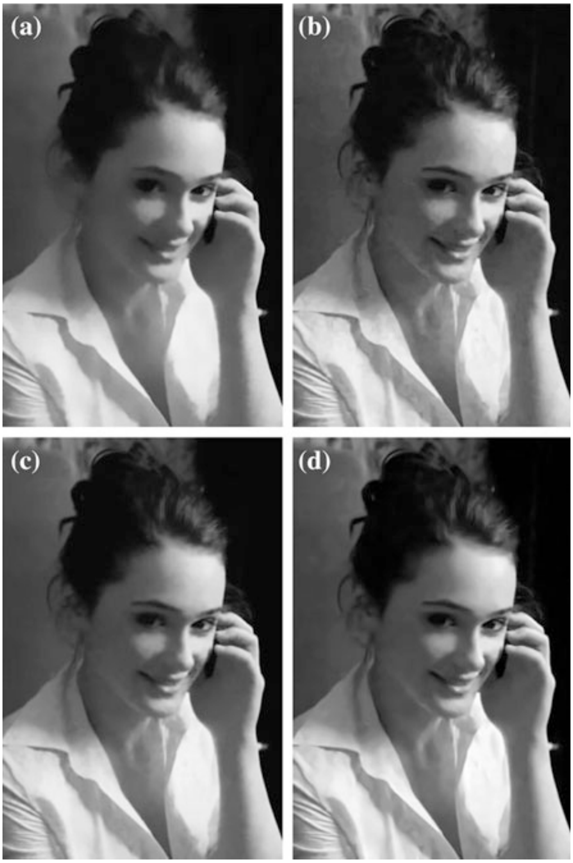


Fig. 10.8 Fragments of the AWGN-affected "Penelope" ($\sigma = 25$) test image reconstruction results (in brackets PSNR, dB and MSSIM): **a** non-local means [6, 11–13] (30.70 dB, 0.754), **b** the Block Matching and 3D filtering (BM3D) [18] (32.11 dB, 0.813), **c** the Local Pixel Grouping (LPG)-PCA [5] (32.00 dB, 0.813), **d** parallel scheme (**34.03** dB, **0.814**)

10.5 Bayer Patterns Filtration Based on Non-local PCA

This section is devoted to the possible application of the denoising algorithm based on the non-local PCA for "raw" image filtration (images formed by using the Bayer patterns). The AWGN is again used as a noise model. Most of the modern filtration algorithms designed for grayscale and color image processing cannot be directly applied to the mosaic structure of the "raw" image because they do not encounter the correlation dependences between its color components. The adaptation of the discussed denoising algorithm to the specific structure of the "raw" image is addressed here. Two processing strategies are possible as mentioned below:

- The channel-wise processing (four channels in this case). This approach was considered irrational, and thus it is not described.
- The simultaneous processing of all channels treating the "raw" image as an integral structure. In this case, to ensure that colors are not mixed in a training set, a block-matching algorithm should be modified. It was shown that this approach is the most efficient.

Let us consider the following main modification stages of the described denoising algorithm based on the non-local PCA:

1. A noise variance σ^2 of the input image $\mathbf{y} = \mathbf{x} + \mathbf{n}$ is assumed to be known. Here, $\mathbf{y}$ and $\mathbf{x}$ are the "raw" images.
2. A wavelet preprocessing [25] of the image $\mathbf{y}$ is performed. It provides a preliminary estimation of a non-noised image $\mathbf{x}$.
3. Similar to the step 3 of the first processing stage of the described denoising algorithm, an image is split into a series of blocks, and the coordinates of similar blocks are searched. In the search, the regions only containing blocks with a color orientation analogous to the reference block are considered.
4. In the implementation, it is proposed to use the odd-sized blocks for overlapping of the adjacent Bayer patterns inside each of the reference blocks: 3 × 3, 5 × 5, 7 × 7, etc. A block-matching will result in a stack of the similar odd-sized blocks with the analogous color orientation.
5. Repeat the steps 4–8 of the first processing stage of the described denoising algorithm based on the non-local PCA. A selective matrix is formed from the stack of the similar blocks. It consists of the input data required for the PCA. The basis truncation and the transform coefficients processing are performed in the similar fashion as shown for the described denoising algorithm. As a result, a primary estimation of the "raw" image is formed.
6. The second processing stage resembles the one of the described denoising algorithm based on the non-local PCA. An empiric Wiener filter is formed again [8]. The stage results in a second estimation of the "raw" image.
7. The present algorithm does not utilize the whole description of the post-processing stage. This is again due to the mosaic nature of "raw" images. However, it is possible to apply a histogram correction. This stage forms a final estimation of the input non-noised "raw" image $\mathbf{x}$.

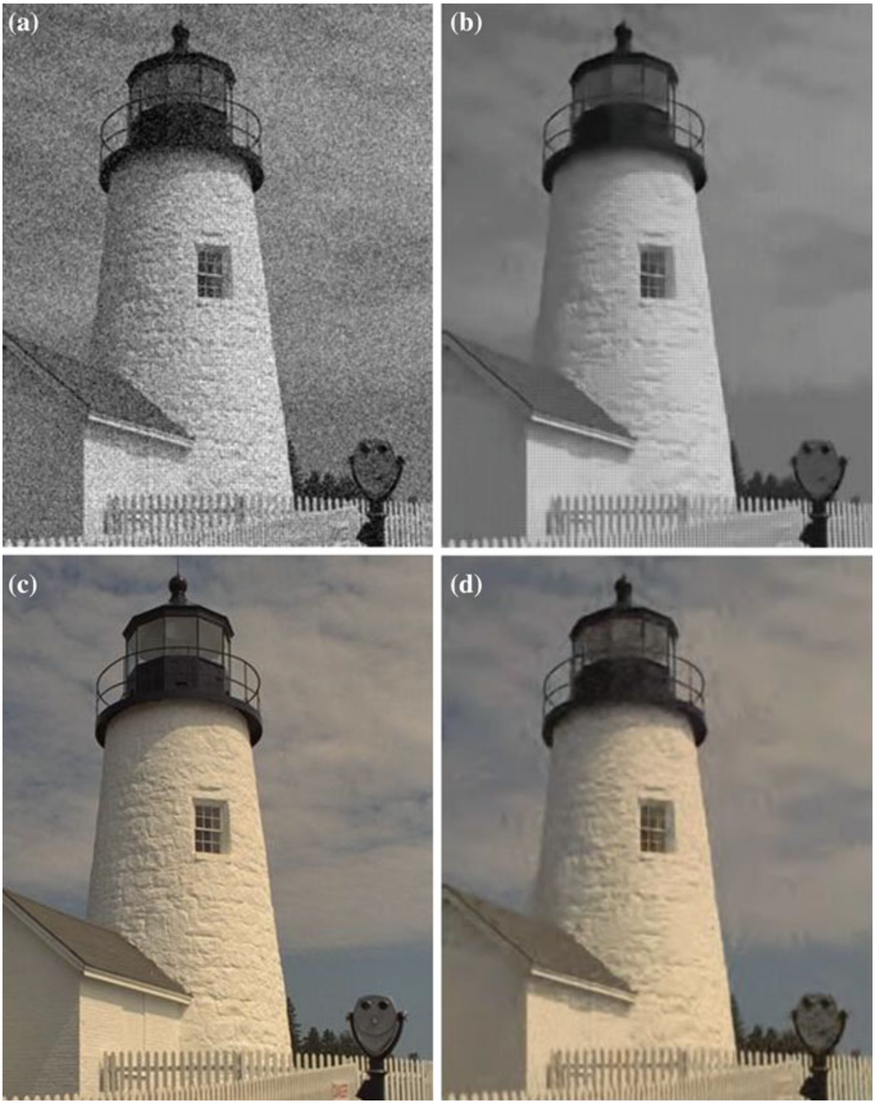


Fig. 10.9 Fragments of test image “Lighthouse” (in brackets PSNR, dB and MSSIM): **a** noised “raw“ image (20.25 dB, 0.422), **b** non-local PCA (29.98 dB; 0.905), **c** non-noised true color image (∞ dB, 1), **d** non-local PCA + demosaicking (R: 29.63 dB, 0.865; G: 29.91 dB, 0.868; B: 29.10 dB, 0.857)

The processing results are visualized in Figs. 10.9 and 10.10 on the example of images “Lighthouse” (#19) and “Houses” (#8) of Kodak database [19] for the denoising algorithm based on the non-local PCA and the described demosaicking procedure.

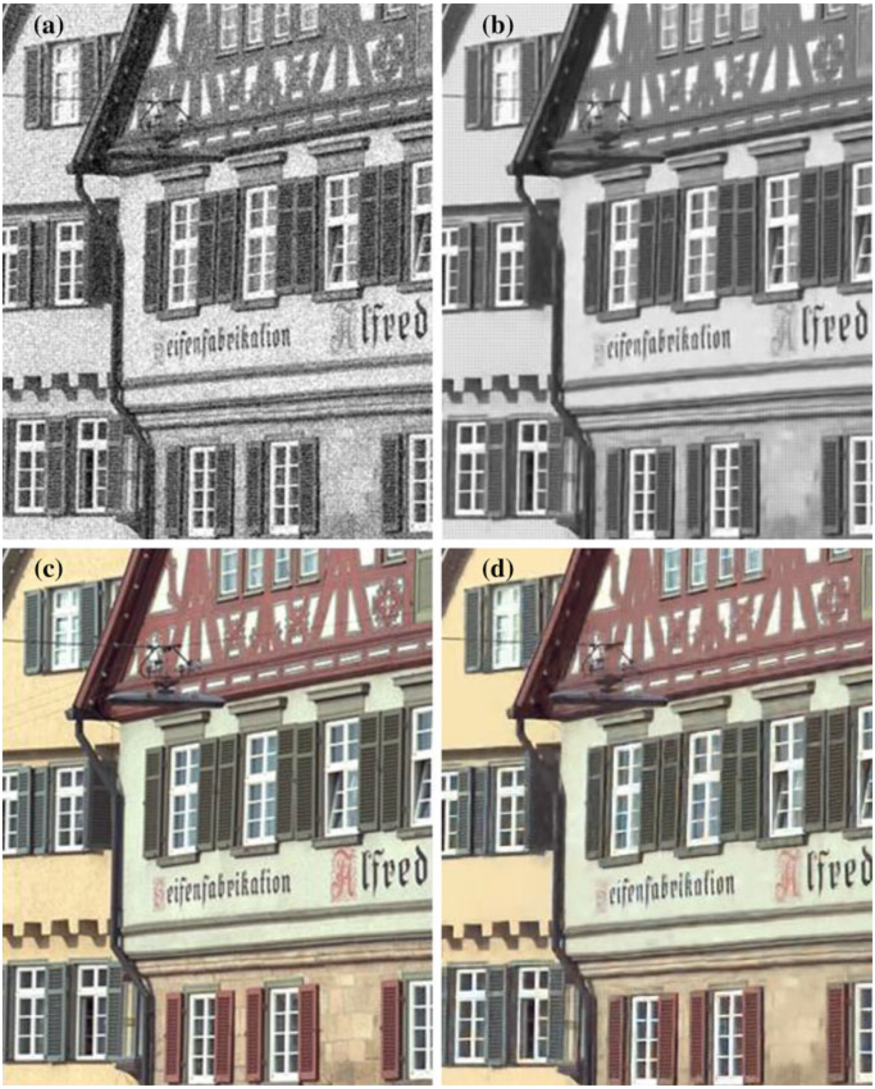


Fig. 10.10 Fragments of test image “Houses” (in brackets PSNR, dB and MSSIM): **a** noised “raw” image (20.46 dB, 0.637), **b** non-local PCA (26.62 dB, 0.869), **c** non-noised true color image (∞ dB, 1), **d** non-local PCA + demosaicking (R: 25.98 dB, 0.807; G: 26.52 dB, 0.816; B: 26.31 dB, 0.806)

10.6 Application of Denoising Algorithms to the Task of License Plate Recognition

The last decade was marked by a global breakthrough in the information technology field, which allowed a broad application of intelligent information systems to many scopes of modern life. Among those perspective and dynamically developing systems in the past years, the computer vision systems capable of solving a multitude of important tasks are excelled. The examples of such tasks are the pattern recognition issues. At present, in many practical problems it is necessary to make some assumptions about the content of the input image or alternatively classify the objects present in the image. This section briefly describes the main stages of car license plate recognition in a digital image [26].

The importance of ensuring traffic security is very high everywhere in the world. Improvement of the road situation is possible by using the intelligent transport systems. They involve a set of interconnected functional blocks such as information collection system, which utilizes the road sensors, detectors, and cameras, meteorological monitoring system, traffic lights, controlled road signs, etc. All these elements are operated from the information analysis center. In the last years, the development of such systems has received an increased attention [27].

One of the integral components of the system is a subsystem of Automatic car Number Plate Recognition (ANPR) [28]. It allows not only recognize and count the moving vehicles but also distinguish between them by identifying their unique state registration numbers. The main processing stages of such system are the preliminary image processing (Sect. 10.6.1), the license plate detection (Sect. 10.6.2), its segmentation (Sect. 10.6.3), and further recognition of the symbols on the plate (Sect. 10.6.4).

10.6.1 Preliminary Image Processing

The preliminary image processing is necessary for contrasting of the edge points, a noise cancellation, and a correct binarization of the input image from the traffic camera. Typical examples of input images used for license plate recognition are shown in Fig. 10.11.

At the preliminary processing stage, it is possible to work with various image types, for example, with grayscale. A noise cancellation can be performed using one of the algorithms described in the preceding sections. Examples of the processing of license plates' fragments with the parallel filtration scheme (Fig. 10.3) are shown in Fig. 10.12. Here for illustrative purposes, the AWGN was artificially added to a detected fragment of the license plate. The PSNR and the MSSIM were calculated in relation to a non-noised plate fragment.

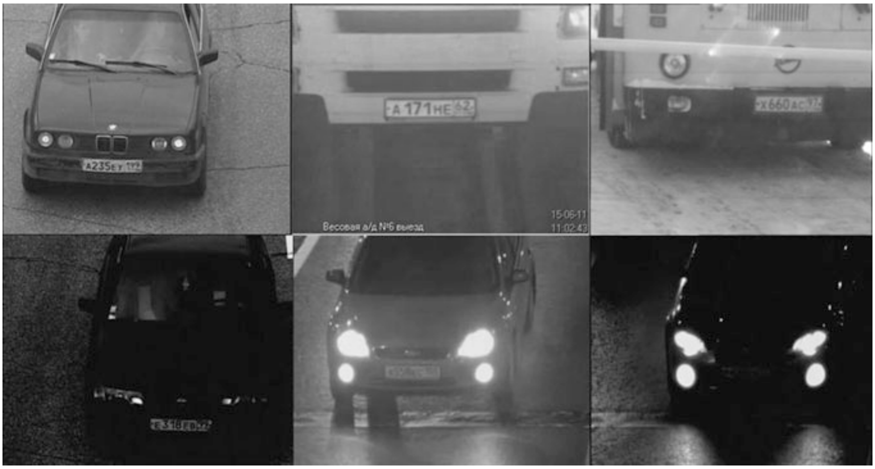


Fig. 10.11 Example of typical images processed by automatic license plate recognition systems

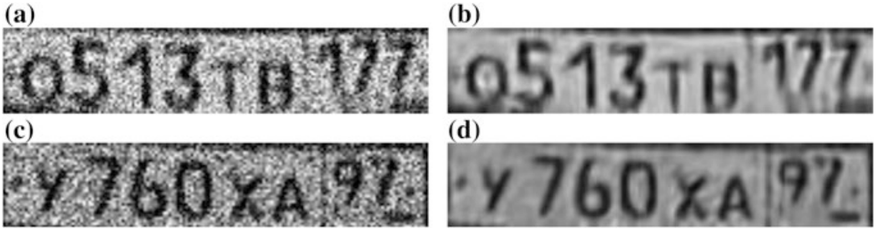


Fig. 10.12 Detected license plates' fragments before and after preliminary processing (in brackets PSNR, dB and MSSIM): **a** the AWGN-noised (σ = 50) image (14.77 dB, 0.492), **b** parallel filtration scheme (24.09 dB, 0.889), **c** the AWGN-noised (σ = 50) image (14.64 dB, 0.465), **d** parallel filtration scheme (23.88 dB, 0.849)

10.6.2 License Plate Detection

The design of the algorithm capable of correct detection of the license plate area in the input image should require the least possible number of a priori assumptions of the plates' properties and features: its size, aspect ratio, and symbols' fill patterns. However, the use of additional assumptions significantly limits the versatility of the algorithm, and thus does not guarantee the correct work in all settings.

The processing stages required for the license plate detection include:

- A preliminary search of the regions of interest, where a license plate may be located. Note that among those may be wrongly found regions (e.g. letterings, radiator grill, etc.).
- A description of the obtained preliminary processing results. Humans denote a license plate in a "common" language as "a metal or plastic plate attached to a

vehicle for its official identification". Hence, it is needed to form a definition of the license plate based on initially known to the system set of descriptors.

- A specific decision rule or a set of those, based on which the system will answer the question of an area's belonging to the definition of the license plate.

For the task of localization of the license plate in the processed image, it is possible to use a search for key features, by which every license plate possess, called as corner points. Here a Harris corner detector may be successfully applied [29]. The algorithm decides, if a current point is the corner point or not based on the intensity gradient values, then the statistics are gathered from the current pixel neighborhood. This algorithm provides a map of the corner features of the given input image. Further, the Harris corner map is binarized according to a set threshold [30], for example, based on the Otsu method [31]. Then, the extraction of the connected regions found at the previous stage is performed. One of these regions refers to a license plate.

The search of this region of interest includes a task of classification of the whole set of regions. The task is decomposed into a description of the found regions using a given set of features and the classification procedure itself including the machine learning techniques [32]. This algorithm provides essential information, based on which the system decides, if the given region of the input image is a license plate or not. As a result, the algorithm formulates the coordinates of the license plate in a given image.

A corner detector or in a more general terminology—a feature point detector is an approach, which is used in computer vision systems for location of specific digital image features. Literature introduces a number of point feature detectors including: Moravec, Shi and Tomasi detectors, etc. One of the major disadvantages of many modern algorithms is in their high computational complexity and not very high accuracy.

After the binarization of the Harris corner map, a resulting (binary) image contains several connected regions. They match the regions of the initial input image. Among these regions, a license plate may be found. To determine, which of them is the one searched for, a classification of these regions using the anomalies detection algorithm is performed. For each of the regions, its characteristic features are computed; here it is proposed to use Histograms of Oriented Gradients (HOG) descriptors [33]. In the construction of such type of descriptors specification of their parameters, called as the number of cells used in the histograms, plays an important role. A normalization of the histograms inside the cells ensures its invariance to photometric transformations such as contrast and brightness correction. The final stage of the HOG-descriptors computation is to obtain a feature vector. It is calculated as a unity of all the elements of normalized histogram blocks.

Currently many machine learning techniques are capable of quite accurate separation of the classes. An anomalies detection algorithm [34] was used for the task of the binary classification. Its key feature is learning only on the "positive" cases. Thereat, all its positive responses have a significant learning impact, whereas the remaining cases are treated at the random fluctuations level.

Table 10.1 A percentage of the test images with a correctly detected license plate

AWGN variance σ	Without a parallel filtration scheme (%)	With a parallel filtration scheme (%)
5	96	95
10	95	95
15	93	95
20	86	92
25	67	89
30	56	86
35	44	75

An algorithm operation should perform a stable detection of the license plate at various PSNR levels. The omission and the false detection as the most critical errors for the whole ANPR system should be minimized. The efficiency of the parallel filtration scheme described in the previous sections is verified by the experimental results. Tests were conducted using the video tape of one of the traffic cameras placed in a city street. The frames containing one car with the fully visible license plate, similar to the ones shown in Fig. 10.11, were considered. The tape encountered in total 190 such frames. A test basis was marked for these images each with the size of 720 × 576 pixels (a standard television resolution of 576i). The algorithm itself was implemented in MATLAB. The AWGN with various σ values was artificially added to the previously created basis images. Further, a number of images with fully detected license plate was counted. The comparison results for the work of the detector with and without the parallel filtration scheme are shown in Table 10.1.

Notice that the application of the parallel filtration scheme allows to detect a license plate on a higher number of images even on highly noised input images. Coincidence of the results for low σ values demonstrates the useful property of the parallel filtration scheme in preserving of characteristic features of an input image. The detection results using the described above license plate algorithm are visualized in Fig. 10.13.

10.6.3 License Plate Segmentation

The next stage of analysis in the ANPR is commonly a segmentation of separate characters in the detected region of the license plate. A possible solution may be a construction of the horizontal projection of the binarized image [35] as shown in Fig. 10.14. This approach requires fairly low computational time. However, the change of the camera position relative to the captured vehicle leads to perspective distortions of an input image, which vertical and horizontal axes are no longer parallel to the axis of the license plate. Similarly, an input image noise leads to significant errors while using the projective methods.

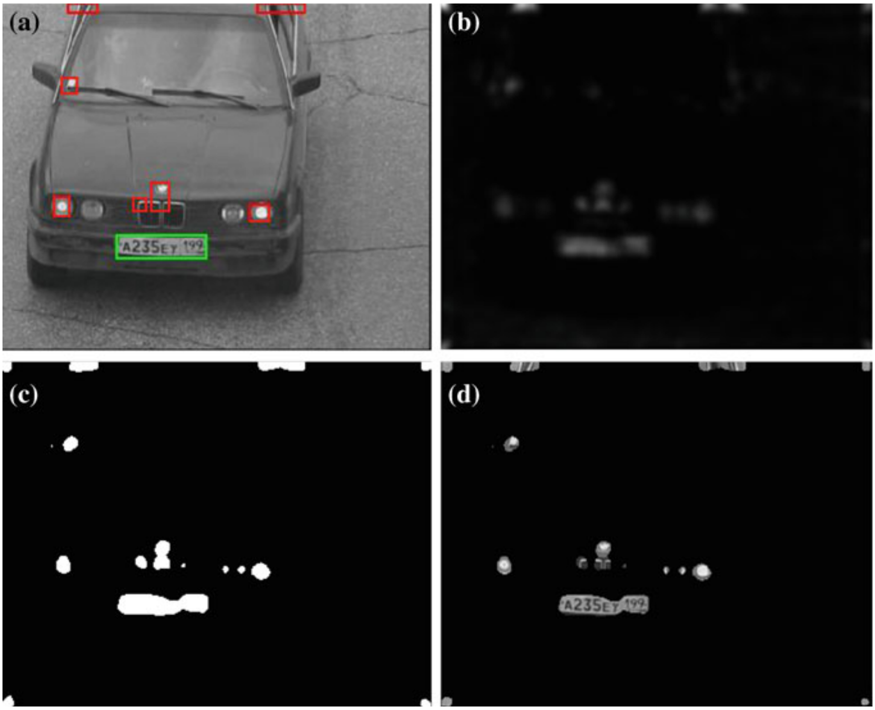


Fig. 10.13 Visualization of the license plate detection algorithm processing stages: **a** initial image with marked license plate candidate regions, **b** the Harris corner map, **c** the Otsu method-binarized Harris corner map, **d** regions of the initial image found using the binarized Harris corner map

10.6.4 Symbols Recognition

The recognition of text characters on the car license plates is based on the classification of the previously found and segmented objects of the input image over the alphabetical characters. The existed classification methods use either pre-built patterns, or use learning. In the first approach for each possible character the corresponding patterns are built and placed to the database. During the recognition process, an incoming image is compared to all the existed patterns. In this case, it is also needed to bring all the compared characters to the same size. However, this method results in significant errors even with minor alterations of color or luminance of the objects.

Among the notable recognition methods, there are learning-based multi-layered neural networks [36], decision trees, support vector machines, and committee methods based on the simultaneous use of the several classifiers. The use of the mentioned above methods allows to achieve a higher accuracy of recognition in the ANPR systems.

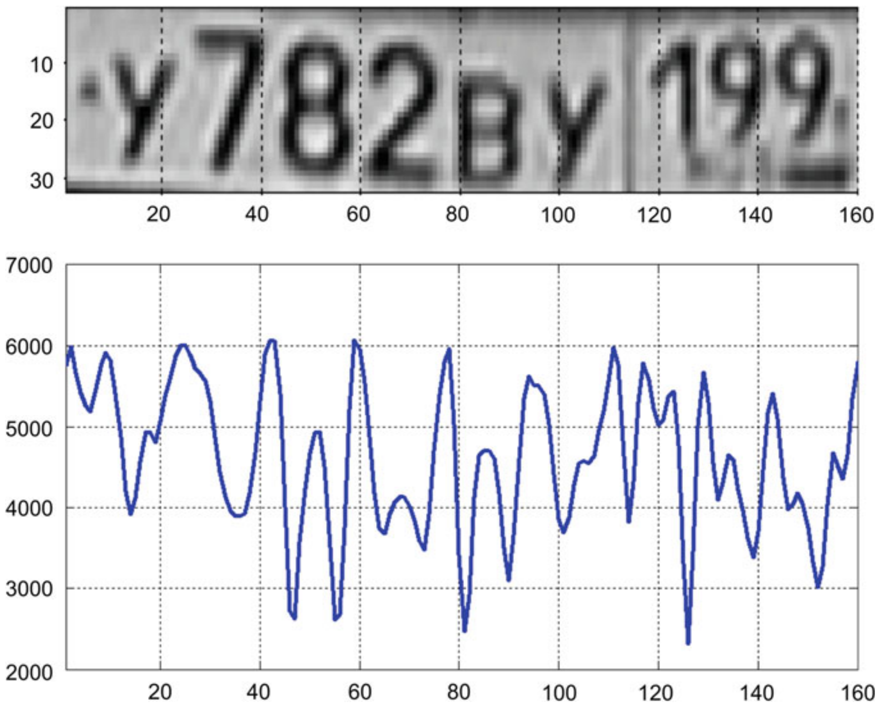


Fig. 10.14 Example of typical images processed by automatic license plate recognition systems

10.7 Conclusion

This chapter includes several denoising algorithms for grayscale image based on the PCA, the optimal Wiener filtering, and a non-local processing idea, which allow effective reconstruction of the AWGN-affected digital images. Several applications of the described algorithms were illustrated including color image and the AWGN-affected “raw” image filtration, mixed noise filtering, and removal of the blocking artifacts appearing after the JPEG compression. A brief overview of the car license plate recognition system with its major processing stages was presented. An applicability of the described filtration algorithms was indicated for such recognition systems.

References

1. Aharon M, Elad M, Bruckstein A, Katz Y (2006) The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54 (11):4311–4322
2. Chatterjee P, Milanfar P (2010) Is denoising dead? IEEE Trans Image Process 19(4):895–911

3. Katkovnik V, Foi A, Egiazarian K, Astola J (2010) From local kernel to nonlocal multiple-model image denoising. Int J Comput Vision 86(8):1–32
4. Deledalle CA, Salmon J, Dalalyan A (2011) Image denoising with patch based PCA: local versus global. Br Mach Vision Conf (BMVC'2011) 25:1–10
5. Zhang L, Dong W, Zhang D, Shi G (2010) Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recogn 43(8):1531–1549
6. Buades A (2005) Image and film denoising by non-local means. Ph.D. thesis, Uni. de les Illes Balears
7. Tasdizen T (2008) Principal components for non-local means image denoising. In: 15th IEEE international conference on image processing (ICIP'2008), pp 1728–1731
8. Muresan DD, Parks TW (2003) Adaptive principal components and image denoising. IEEE Image Proc 1:101–104
9. Priorov A, Tumanov K, Volokhov V, Sergeev E, Mochalov I (2013) Applications of image filtration based on principal component analysis and nonlocal image processing. IAENG Int J Comput Sci 40(2):62–80
10. Yaroslavsky L (1985) Digital picture processing. an introduction. Springer, Berlin
11. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. IEEE Comp Soc Conf Comput Vision Pattern Recognit (CVPR'2005) 2:60–65
12. Buades A, Coll B, Morel JM (2005) A review of image denoising algorithms, with a new one. Multiscale Model Simul 4(2):490–530
13. Buades A, Coll B, Morel JM (2008) Nonlocal image and movie denoising. Int J Comput Vision 76(2):123–139
14. Priorov A, Volokhov V, Sergeev E, Mochalov I, Tumanov K (2013) Parallel filtration based on principle component analysis and nonlocal image processing. Int MultiConf Eng Comput Scientists (IMECS'2013) 1:430–435
15. University of Granada Computer Vision Group test images database. http://decsai.ugr.es/cvg/dbimagenes. Accessed 1 Sept 2014
16. Salomon D (2004) Data, image and audio compression. Technoshere, Moscow
17. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
18. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3D transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
19. The RPI-CIPR Kodak image database. http://www.cipr.rpi.edu/resource/stills/kodak.html. Accessed 15 July 2014
20. Hirakawa K, Parks TW (2006) Image denoising using total least squares. IEEE Trans Image Process 15(9):2730–2742
21. Ding R, Venetsanopoulos AN (1987) Generalized homomorphic and adaptive order statistic filters for the removal of impulsive and signal-dependent noise. IEEE Trans Circuits Syst CAS 34(8):948–955
22. Gonzalez R, Woods R (2008) Digital image processing, 3rd edn. Prentice Hall, NJ
23. Salomon D (2002) A guide to data compression methods. Springer, NY
24. Foi A, Katkovnik V, Egiazarian K (2007) Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. IEEE Trans Image Process 16 (5):1395–1411
25. Lang M, Guo H, Odegard J, Burrus CS (1995) Noise reduction using an undecimated discrete wavelet transform. IEEE SP Lett 3(1):10–12
26. Shapiro V, Dimov D, Velichkov V, Gluhche G (2004) Adaptive license plate image extraction. Int Conf Comp Syst Tech IIIA.2 1–6
27. Szeliski R (2010) Computer vision: algorithms and applications, 1st edn. Springer, London
28. Lukyanitsa AA, Shishkin AG (2009) Digital processing of video images. ISS Press, Moscow
29. Harris C, Stephens M (1988) A combined corner and edge detection. In: 4th alvey vision conference pp 147–151
30. Singh TR, Roy S, Singh OI, Sinam T, Singh KM (2011) A new local adaptive thresholding technique in binarization. Int J Comput Sci 8(6):271–277

31. Shapiro LG, Stockman G (2001) Computer vision. Prentice-Hall, NJ
32. Marsland S (2009) Machine learning: an algorithmic perspective. Chapman & Hall/CRC, Boca Raton
33. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110
34. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):article 15
35. Martinsky O (2007) Algorithmic and mathematical principles of automatic number plate recognition systems. B.Sc. thesis, BRNO University of Technology
36. Nagare AP (2011) License plate character recognition system using neural network. Int J Comput Appl 25:article 2

Chapter 11
Image Segmentation Based on Two-Dimensional Markov Chains

Elena Medvedeva and Ekaterina Kurbatova

Abstract The image segmentation is a crucial problem in many tasks of computer vision. In this chapter, it is proposed the mathematical theory of conditional Markov processes, the representation of halftone *g*-digit images as a collection of *g* binary images, and the entropy approach to segment the objects of interest and texture areas in images, particular in the satellite images. The proposed approach develops the novel efficient methods for contour and texture segmentation in real world noisy images provided a high accuracy and less computational resources in comparison with the conventional methods (Canny, Laplacian of Gaussian, Roberts, Prewitt, and Sobel). The performed mathematical models and experimental researches confirm that the developed segmentation algorithms are effective in terms of quality and processing speed.

Keywords Segmentation · Digital halftone images · Markov process · Information content · Nonlinear filtering · Texture segmentation

11.1 Introduction

Methods of segmentation can be concern to a middle stage of image processing in many tasks including the remote sensing of the Earth, the disease diagnosis, and other various tasks of video analysis. The great contribution to the development of image preprocessing methods was made by such scientists as Gonzalez and Woods [1], Zhuravlev and Gurevich [2], Jahne [3], Soifer [4], Canny [5], Pratt [6], Prewitt [7], Sobel [8], Roberts [9], Kirsch [10].

E. Medvedeva (✉) · E. Kurbatova
Vyatka State University, 36 Moskovskaya Str, Kirov 610000, Russian Federation
e-mail: emedv@mail.ru

E. Kurbatova
e-mail: kurbatovae@gmail.com

M.N. Favorskaya and L.C. Jain (eds.), *Computer Vision in Control Systems-2*,
Intelligent Systems Reference Library 75, DOI 10.1007/978-3-319-11430-9_11

A variety of existing segmentation methods is realized by two approaches. The first approach detects the contours of objects in image and subsequently fills the selected segments, which are related to objects of interest (contour segmentation methods) [1–15]. Most segmentation methods of this type require the additional changes, the contours tracing, the increasing quantity or dimension of masks, the calculation of threshold, and etc. to detect accurately the contours of objects of interest in a Digital Halftone Image (DHI). The second approach for segmentation analyzes the neighbor elements according to their homogeneity (methods are based on a search of homogeneous or texture areas) [1, 16]. Many segmentation methods of this type take into account a spatial arrangement of image elements. For this goal, the properties of some area are analyzed. It is necessary to process each image element many times. Methods based on search of homogeneous areas are more reliable to noise than contour methods but require the high computing resources.

The detection of object of interest is complicated even more, if it is necessary to quickly process the image transmitted over a noisy radio channel. A complex non-homogeneous background in image and sometimes a low signal-to-noise ratio do not allow the application of the simple solutions. At the same time, the steady growth of video data and the necessity of data processing in real time require the novel, more efficient algorithms.

The using of multi-dimensional Markov chains with several states [17–19] as mathematical model is a promising solution for image processing. However, there are some difficulties with storing and handling of the transition probability matrices with dimension of $2^g \times 2^g$ in processing of digital halftone (or color) images with numbers of intensity levels equal 2^g. Such way of image processing requires the high computational resources. The representation of g-bit halftone images as a set of g Bit Binary Images (BBI) made it possible to reduce the computational resources due to handling the transition probability matrices with size of 2×2.

The chapter is organized as follows. The image segmentation method based on contours detection is developed in Sect. 11.2. The combined segmentation method for noisy images is described in Sect. 11.3. Section 11.4 provides a description of the method for texture image segmentation. Conclusion is situated in Sect. 11.5.

11.2 Image Segmentation Method Based on Contours Detection

Let the DHI be represented as a set of g BBIs. Each BBI is a 2D random Markov field with a separable correlation function provided by Eq. 11.1 [20], where σ_μ^2 is a dispersion of two-dimensional discrete-valued Markov process, α_1, α_2 are the multipliers depending on the width of the power spectral density of random processes in two dimensions, f and s are the horizontal and vertical correlation steps, respectively.

$$r_{f,s} = \sigma_{\mu}^{2} exp\{-\alpha_1|f| - \alpha_2|s|\} \tag{11.1}$$

In this case, the BBI may be represented as the superposition of two orthogonal 1D horizontal and vertical Markov chains with two equal probable $\left(p_1^{(l)} = p_2^{(l)}\right)$ states $M_1^{(l)}$ and $M_2^{(l)}$ with the transition probability matrices for the horizontal and vertical directions of an image, respectively (Eq. 11.2).

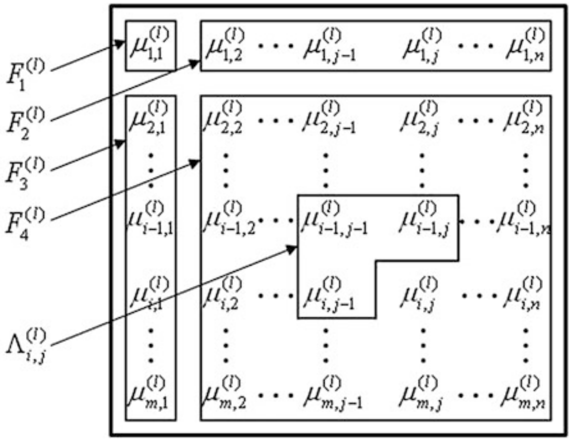
$$^{1}\Pi^{(l)} = \begin{Vmatrix} ^{1}\pi_{11}^{(l)} & ^{1}\pi_{12}^{(l)} \\ ^{1}\pi_{21}^{(l)} & ^{1}\pi_{22}^{(l)} \end{Vmatrix} \quad ^{2}\Pi^{(l)} = \begin{Vmatrix} ^{2}\pi_{11}^{(l)} & ^{2}\pi_{12}^{(l)} \\ ^{2}\pi_{21}^{(l)} & ^{2}\pi_{22}^{(l)} \end{Vmatrix} \quad l = \overline{1, g} \tag{11.2}$$

The 2D lth BBI divided in the areas $F_i^{(l)}\left(i = \overline{1, 4}\right)$, whose elements represent a Markov chain of different dimensions, is shown in Fig. 11.1.

The 2D lth BBI fragment corresponding to the random Markov field area $F_4^{(l)}$ (Fig. 11.1) is shown in Fig. 11.2, where some notations are accepted.

The entropy approach for calculation of probabilities of the elements states is applied. In accordance with this approach, the information content in the element $v_3^{(l)}$ relative to the elements of the nearest neighborhood $v_1^{(l)}$, $v_2^{(l)}$ is determined as difference between own information in an element $v_3^{(l)}$ and the mutual information between elements $v_1^{(l)}$, $v_2^{(l)}$, $v_3^{(l)}$ [18]. This is expressed by Eq. 11.3, where

Fig. 11.1 Areas of the lth BBI


$F_1^{(l)}$
$F_2^{(l)}$
$F_3^{(l)}$
$F_4^{(l)}$
$\Lambda_{i,j}^{(l)}$
$\mu_{1,1}^{(l)}$
$\mu_{1,2}^{(l)} \cdots \mu_{1,j-1}^{(l)} \quad \mu_{1,j}^{(l)} \cdots \mu_{1,n}^{(l)}$
$\mu_{2,1}^{(l)}$
$\mu_{2,2}^{(l)} \cdots \mu_{2,j-1}^{(l)} \quad \mu_{2,j}^{(l)} \cdots \mu_{2,n}^{(l)}$
$\mu_{i-1,1}^{(l)}$
$\mu_{i-1,2}^{(l)} \cdots \mu_{i-1,j-1}^{(l)} \quad \mu_{i-1,j}^{(l)} \cdots \mu_{i-1,n}^{(l)}$
$\mu_{i,1}^{(l)}$
$\mu_{i,2}^{(l)} \cdots \mu_{i,j-1}^{(l)} \quad \mu_{i,j}^{(l)} \cdots \mu_{i,n}^{(l)}$
$\mu_{m,1}^{(l)}$
$\mu_{m,2}^{(l)} \cdots \mu_{m,j-1}^{(l)} \quad \mu_{m,j}^{(l)} \cdots \mu_{m,n}^{(l)}$

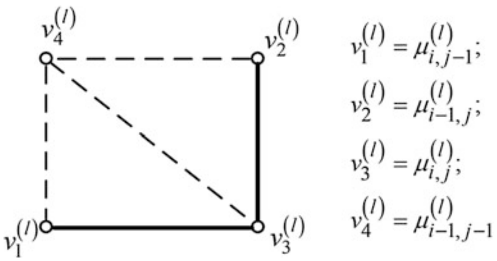

$v_4^{(l)}$
$v_2^{(l)}$
$v_1^{(l)}$
$v_3^{(l)}$
$v_1^{(l)} = \mu_{i,j-1}^{(l)};$
$v_2^{(l)} = \mu_{i-1,j}^{(l)};$
$v_3^{(l)} = \mu_{i,j}^{(l)};$
$v_4^{(l)} = \mu_{i-1,j-1}^{(l)}$

Fig. 11.2 Image fragment with the neighborhood $\Lambda_{i,j}^{(l)}$

$p\left(v_i^{(l)}\right)$, $i = \overline{1,3}$ is a priori probability density of elements values, $p\left(v_1^{(l)}, v_2^{(l)}, v_3^{(l)}\right)$, $p\left(v_i^{(l)}, v_j^{(l)}\right)$, $i \neq j = \overline{1,3}$ are joint probability densities of elements values, $w\left(v_3^{(l)}\middle|v_1^{(l)}\right)$, $w\left(v_3^{(l)}\middle|v_2^{(l)}\right)$ are the 1D probability densities for the transition between neighbor elements, $w\left(v_3^{(l)}\middle|v_2^{(l)}, v_1^{(l)}\right)$ is the probability density for the transition in a 2D Markov chain.

$$\begin{aligned} I\left(v_3^{(l)}\middle|v_2^{(l)}, v_1^{(l)}\right) &= I\left(v_3^{(l)}\right) - I\left(v_1^{(l)}, v_2^{(l)}, v_3^{(l)}\right) \\ &= -\left[\log p\left(v_3^{(l)}\right) + \log \frac{p\left(v_3^{(l)}, v_1^{(l)}\right)p\left(v_1^{(l)}, v_2^{(l)}\right)p\left(v_2^{(l)}, v_3^{(l)}\right)}{p\left(v_1^{(l)}\right)p\left(v_2^{(l)}\right)p\left(v_3^{(l)}\right)p\left(v_3^{(l)}, v_2^{(l)}, v_1^{(l)}\right)}\right] \\ &= -\log \frac{w\left(v_3^{(l)}\middle|v_1^{(l)}\right)w\left(v_3^{(l)}|v_2^{(l)}\right)}{w\left(v_3^{(l)}\middle|v_2^{(l)}, v_1^{(l)}\right)} \end{aligned} \tag{11.3}$$

Let us write the probability density in a two-dimensional Markov chain $w\left(v_3^{(l)}\middle|v_2^{(l)}, v_1^{(l)}\right)$ by Eq. 11.4, where $\delta(\cdot)$ is the delta-function.

$$\begin{aligned} w\left(v_3^{(l)}\middle|v_1^{(l)}, v_2^{(l)}\right) = \sum_{j,q=1}^{2} \pi\left(v_3^{(l)} = M_i^{(l)}\middle|v_1^{(l)} = M_j^{(l)}; v_2^{(l)} = M_q^{(l)}\right) \\ \times \delta\left(v_1^{(l)} - M_j^{(l)}\right)\delta\left(v_2^{(l)} - M_q^{(l)}\right) \end{aligned} \tag{11.4}$$

For the elements of the lth BBI, the information content between the element $v_3^{(l)}$ and various combinations of neighbor elements $\Lambda_{ij}^{(l)}$ is determined by the Eq. 11.5 [18], where ${}^r\pi_{ii}^{(l)}$, $\left(i,j = \overline{1,2}; r = \overline{1,3}\right)$ are the elements of transition probability matrices in 1D Markov chain with two states ${}^1\Pi^{(l)}$, ${}^2\Pi^{(l)}$, ${}^3\Pi^{(l)} = {}^1\Pi^{(l)} \cdot {}^2\Pi^{(l)}$.

$$\begin{aligned}
I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_i^{(l)}; v_2^{(l)} = M_i^{(l)}\right) &= -\log \frac{{}^1\pi_{ii}^{(l)}\, {}^2\pi_{ii}^{(l)}}{{}^3\pi_{ii}^{(l)}} \\
I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_i^{(l)}; v_2^{(l)} = M_j^{(l)}\right) &= -\log \frac{{}^1\pi_{ii}^{(l)}\, {}^2\pi_{ij}^{(l)}}{{}^3\pi_{ij}^{(l)}} \\
I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_j^{(l)}; v_2^{(l)} = M_i^{(l)}\right) &= -\log \frac{{}^1\pi_{ij}^{(l)}\, {}^2\pi_{ii}^{(l)}}{{}^3\pi_{ij}^{(l)}} \\
I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_j^{(l)}; v_2^{(l)} = M_j^{(l)}\right) &= -\log \frac{{}^1\pi_{ij}^{(l)}\, {}^2\pi_{ij}^{(l)}}{{}^3\pi_{ii}^{(l)}}
\end{aligned} \tag{11.5}$$

The information content in an image element will be minimal, if the neighboring elements $v_1^{(l)}$ and $v_2^{(l)}$ have states identical to $v_3^{(l)}$.

When areas of different brightness appear in image, the state of one or two neighbor elements at the boundary of an area will differ from that of $v_3^{(l)}$ state, and the information content in the element $v_3^{(l)}$ is increased. Comparing the calculated information content in the image element $v_3^{(l)}$ with the threshold h, one can determine, whether the given point belongs to the contour or not.

The threshold value h is calculated for each BBI with consideration for the calculated minimum information content and the information content corresponding to the case, when any of the neighboring elements accepts a different state (Eq. 11.6).

$$h = \frac{I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_i^{(l)}, v_2^{(l)} = M_i^{(l)}\right) + I\left(v_3^{(l)} = M_i^{(l)} \middle| v_1^{(l)} = M_i^{(l)}, v_2^{(l)} = M_j^{(l)}\right)}{2} \tag{11.6}$$

It is assumed that the elements of transition probabilities matrices are known apriori, when the information content is calculated.

In the case of the 8-bit DHI represented by 256 brightness levels, the upper (eighth) BBI is characterized by 128 brightness levels. Therefore, using the upper BBI, one can distinguish all light areas with brightness from 128 to 255 in a dark background, or, on the contrary, all dark objects in a background with brightness above 128. To distinguish less contrasting areas or objects with fuzzy edges, it is necessary to outline the contours on the following bin BBI (seventh, sixth, or fifth). In this case a contour image will represent the sum of contour images of several BBIs.

As a result, two types of errors may occur. A point can be mark as a contour point but it does not concern to a contour. Also a point can be not mark as a contour point, but it is a contour point in the ideal image. Thus, two criteria are proposed for estimation the accuracy of contour detection: the criterion Figure of Merit (FOM) showing a level of similarity between the segmented and the ideal image and the criterion Root Mean Squared (RMS) error showing a level of their difference.

The FOM criterion corresponds to empirical distance between the ideal contour image *f* and the contours obtained as a result of segmentation *g*. It is provided by Eq. 11.7, where *card*(*f*) is a number of pixels in the image *f*, *card*(*g*) is a number of pixels in the image *g*, *d*(*i*) is distance between the *i*th pixel *f* and the nearest to it pixel *g*.

$$FOM(f,g) = \frac{1}{max\{card(f), card(g)\}} \cdot \sum_{i=1}^{card(g)} \frac{1}{1 + d^2(i)} \tag{11.7}$$

The RMS criterion represents an averaged square error calculated by Eq. 11.8, where *f*(*x*), *g*(*x*) are the intensity values of pixels x in f_i and g_i, *X* is a set of pixels in the segmented image.

$$RMS(f,g) = \left[\frac{1}{card(X)} \cdot \sum_{x \in X} (f(x) - g(x))^2\right]^{\frac{1}{2}} \tag{11.8}$$

Another important requirement is to provide a closed contour. The criterion of the relative contour incoherence *E* is used to estimate a cliquishness of contour by Eq. 11.9, where *N* is a number of incoherent contour components and elements with coherence equal 1, *card(g)* is a total number of contour elements.

$$E = \frac{N}{card(g)} \tag{11.9}$$

Table 11.1 shows the estimations of the FOM, the RMS, and the E criteria averaged over the hundred test images for the developed and the well-known segmentation methods. For experiments, the images from Berkeley Segmentation Dataset were used [21]. The algorithms were modeled in Matlab. All contour images were obtained by the developed method for upper BBI of the DHI. For methods Roberts, Prewitt, and Sobel, the threshold is selected equal to 0.1. For method Canny, the thresholds are equal 0.03 and 0.015, for Laplacian of Gaussian, the threshold is equal 0.002.

The developed method according to the FOM criterion exceeds the known methods by 6–38 %, and by RMS criterion slightly inferior to Sobel and Prewitt methods. In addition, the developed method, in contrast to the known methods provides the closed contours. It is important for the next step of segmentation—filling the objects of interest.

Table 11.1 Estimations of FOM, RMS, and E criteria for various segmentation methods

Criterion	Developed method	Roberts	Prewitt	Sobel	Canny	LoG
FOM	0.1574	0.1488	0.1390	0.1387	0.1141	0.1200
RMS	0.2877	0.2895	0.2806	0.2850	0.4105	0.3791
E	0	0.3285	0.2245	0.2173	0.1143	0.0637

In the developed method, a contour point is calculated only by using two neighbor BBI elements, whereas the other known techniques (gradient, contrast, second derivative, Prewitt-Kirsch, and Marr-Hilderth methods) use the processing with a slicing window (or a series of slicing windows) with the total number of coefficients of no more than 4. This means that the suggested algorithm reduces the time resources required for implementation in comparison with the known algorithms.

The curves of time dependence from various sizes of image by different segmentation methods are represented in Fig. 11.3. The modeling was performed in Matlab environment by using Windows XP on processor IntelCore2 Duo, 2.66 GHz. For calculations, sixty real halftone images were used.

The developed method has the highest speed of image processing. In the case of big sizes of a processed image, our method gives a gain of 5.7–19.7 times in comparison with the known methods. Let us notice that other contours are outlined, the area corresponding to a distinguished object is determined. For this purpose, the interval $[Y_{min}, Y_{max}]$ of brightness values is specified, which can be accepted by an object, and the average brightness value Y_{av} is assigned to all the elements inside the area of the object.

The seed row-by-row filling algorithm is the most efficient method for the filling of areas [22, 23]. It provides a considerable gain in the memory size and the processing time at the expense of storing only a single seed element for each filled area. For the implementation of this algorithm, the initial element (seed) is specified in the internal area of a segment. If the filled area is an object, then one may accept a value of the average brightness Y_{av}, otherwise a value corresponds to white color (background). All elements along the current row are filled on the right and left of the seed until the contour points X_{lt} and X_{rt} are encountered. Then the rows above and below of the current one are checked into the interval $[X_{lt}; X_{rt}]$. If they have unfilled elements, then the rightmost element of each interval is marked as a seed. As a result, the segments that have the brightness fitting in the specified interval are filled by the color of an object, and all the remaining segments are filled by the color

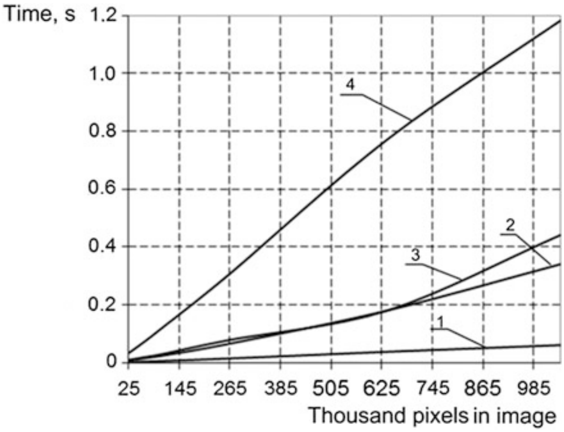


Fig. 11.3 Curves of time from sizes of image processed by different methods: *1* the proposed method, *2* the LoG, *3* methods of Roberts, Prewitt, and Sobel, *4* Canny method

of background. Specifying several brightness intervals, one can distinguish several different segments filled by various colors in the outlined image. For typical images, in which objects of interest have close brightness values, the number of BBIs with outlined contours and the interval of seed values will be constant. Therefore, the developed method can be applied to the automatic segmentation of images.

The example of detection of oil spill is shown in Fig. 11.4. The contour image (Fig. 11.4b) is obtained by using two upper BBIs of DHI (Fig. 11.4a). The detection of oil spill in contour image is situated in Fig. 11.4c, and the detection of oil spill in contour image with additional boundaries of islands, which can track the movement of oil slicks on water surface, is presented in Fig. 11.4d.

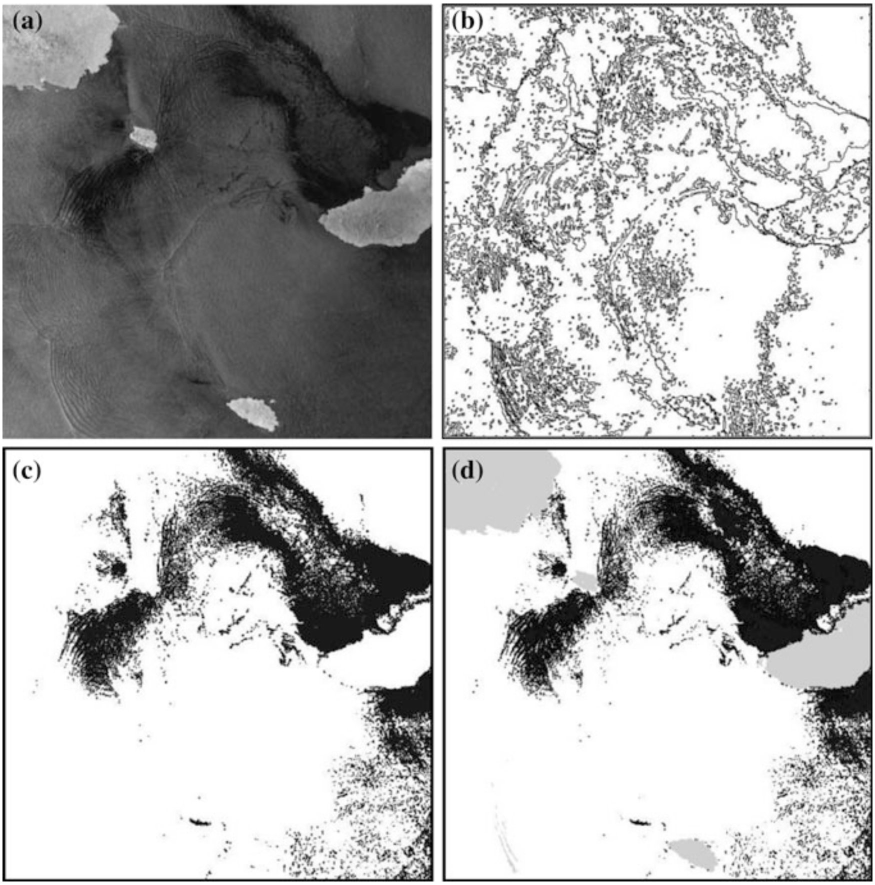


Fig. 11.4 Example of halftone image segmentation: **a** initial DHI, **b** contour image, **c** detection of oil spill on contour image, **d** detection of oil spill and islands on contour image

A color image can be represented as a set of three halftone images, each of which is processed by the developed method of contours detection. Then three contour images are combined into a single contour image, and filling segments is performed. To improve a quality of segmentation, a stage of color image normalization is added in order to equalize the brightness of color components.

The example of color satellite image segmentation is shown in Fig. 11.5. The initial image is shown in Fig. 11.5a. The segmentation results by developed method and known k-means method are represented in Fig. 11.5b, c, respectively. The objects detection of different classes is illustrated by Fig. 11.5d–f. This example shows that the developed method effectively allocates objects of interest, for example, shown by red color burned areas and is not inferior in quality to k-means method.

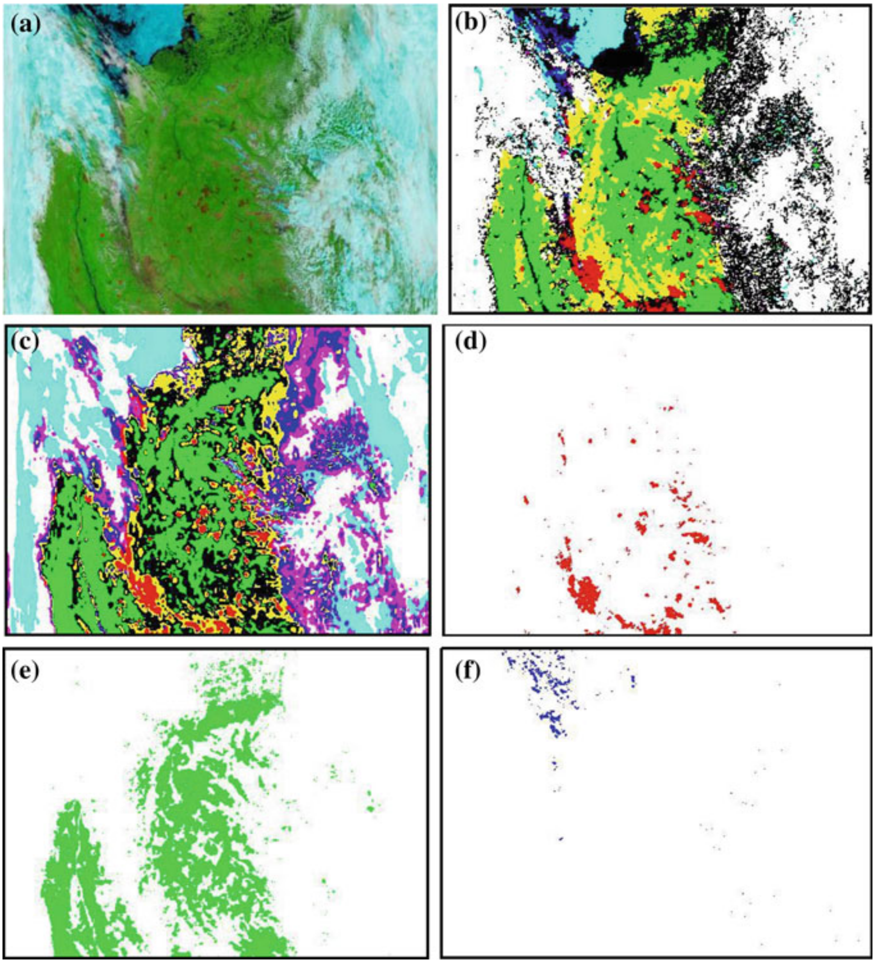


Fig. 11.5 Example of color image segmentation: **a** color image, **b** result of segmentation by developed method, **c** result of segmentation by k-means method, **d** allocated *red* objects, **e** allocated *green* objects, **f** allocated *blue* objects

Thus, the developed segmentation method based on contour detection allocates the objects of interest on halftone in the color images by small computational resources.

11.3 Combined Segmentation Method for Noisy Images

Usually, it is necessary to filter efficiently the images transmitted over a noisy radio channel. The own statistical redundancy of image is a large reserve to restore the noisy images. One of approaches of statistical redundancy is the use of multidimensional conditional Markov processes for synthesis of image filtering algorithms.

Using the approximation of the BBIs by 2D Markov chain with two equiprobable states $M_1^{(l)}$ and $M_2^{(l)}$ with the horizontal and vertical transition probability matrices ${}^1\Pi = \left\| {}^1\pi_{ii}^{(l)} \right\|_{2\times 2}$ and ${}^2\Pi = \left\| {}^2\pi_{ii}^{(l)} \right\|_{2\times 2}$ [24, 25], the 2D nonlinear filtration of the lth BBI is developed by Eq. 11.10, where $u\left(v_3^{(l)}\right) = \ln \frac{p\left(M_1\left(v_3^{(l)}\right)\right)}{p\left(M_2\left(v_3^{(l)}\right)\right)}$ is the log ratio of the posterior probabilities of the filtered element $v_3^{(l)}$ of lth BBI (Fig. 11.2).

$$\begin{aligned} u\left(v_3^{(l)}\right) = {} & \left[f\left(M_1\left(v_3^{(l)}\right)\right) - f\left(M_2\left(v_3^{(l)}\right)\right)\right] + u\left(v_1^{(l)}\right) + z_1\left(u\left(v_1^{(l)}\right), {}^1\pi_{ij}^{(l)}\right) \\ & + u\left(v_2^{(l)}\right) + z_2\left(u\left(v_2^{(l)}\right), {}^2\pi_{ij}^{(l)}\right) - u\left(v_4^{(l)}\right) - z_3\left(u\left(v_4^{(l)}\right), {}^3\pi_{ij}^{(l)}\right) \end{aligned} \tag{11.10}$$

The variables $z_r(\cdot)$ are determined by Eq. 11.11, where ${}^r\pi_{ij}^{(l)}$ $\left(i,j = \overline{1,2}; r = \overline{1,3}\right)$ are the transition probability matrices elements in 1D Markov chains with two states ${}^1\Pi^{(l)}, {}^2\Pi^{(l)}$ and ${}^3\Pi^{(l)} = {}^1\Pi^{(l)} \cdot {}^2\Pi^{(l)}$.

$$z_r(\cdot) = ln \frac{{}^r\pi_{ii}^{(l)} + {}^r\pi_{ji}^{(l)} \exp\left(-u\left(v_r^{(l)}\right)\right)}{{}^r\pi_{jj}^{(l)} + {}^r\pi_{ij}^{(l)} \exp\left(u\left(v_r^{(l)}\right)\right)} \tag{11.11}$$

The DHI filtration algorithm contains g filtration algorithms (Eq. 11.10).

At the first stage, the decision about the presence in the received signal realization $s\left(\mu_{i,j,k}^{(l)}\right)$ of the lth BBI element $\mu_{i,j,k}^{(l)}$, having the value $M_1^{(l)}$ or $M_2^{(l)}$, is performed on the basis of comparison of logarithm of a posteriori probability ratio with some threshold H [26] selected in compliance with the ideal observer criterion in a view of Eq. 11.12.

$$u\left(v_3^{(l)}\right) \geq H \tag{11.12}$$

For forming the logarithm of the a posteriori probability of the BBI element $u\left(v_3^{(l)}\right)$, the input data defined by the first term in the Eq. 11.10 are added with the previously calculated data about elements of neighborhood $\Lambda_{i,j} = \left\{v_1^{(l)}, v_2^{(l)}, v_4^{(l)}\right\}$ and with calculated values of nonlinear function $z_r(\cdot)$, which contains apriori data about the filtering process. The implementation of a priori data in receiver leads to increase the signal to noise ratio.

In the case of statistical independence of pulse signals $\left({}^r\pi_{ij}^{(l)} = 0,5;\right.$ $i,j = \overline{1,2}; r = \overline{1,3})$, the function $z_r(\cdot)$ has a view of Eq. 11.13.

$$z_r(\cdot) = -u\left(v_q^{(l)}\right) \tag{11.13}$$

The logarithm of a posteriori probabilities ratio will be determined only by the difference of logarithms of the likelihood functions expressed by Eq. 11.14.

$$u\left(v_3^{(l)}\right) = \left[f\left(M_1\left(v_3^{(l)}\right)\right) - f\left(M_2\left(v_3^{(l)}\right)\right)\right] \tag{11.14}$$

In the case of ${}^r\pi_{ij}^{(l)} = 0.5$, the decision about presence in the received signal of one or other state of a discrete signal parameter $\mu_{ij}^{(l)}$ is adopted at each time step based on a single measurement.

If ${}^r\pi_{ij}^{(l)} = 1$, when the pulses with the same sign $M_1^{(l)}$ or $M_2^{(l)}$ are transmitted, the function $z(\cdot) = 0$. In this case, a "clean" accumulation of data, arriving to the receiver input, is performed in the adder of a nonlinear filter.

At the second stage, having more accurate estimates of state of lth BBI elements, taking into account states of neighboring elements from the nearest neighborhood $\Lambda_{i,j}^{(l)}$ of the element $v_3^{(l)}$ and the transitions probabilities between elements, it is possible to calculate the information content in the element $v_3^{(l)}$ (Eq. 11.5). Comparing the calculated information content with the threshold h (Eq. 11.6), one can determine, whether the given element belongs to the contour.

The device realizing the combined contour allocation method of noisy images includes a nonlinear filter $\left(NF^{(l)}\right)$ and a device of contour detection. The structure of device (Fig. 11.6) contains discriminators of binary signals $\left(D^{(l)}, \ldots, D^{(g)}\right)$ realizing calculating operations for the logarithm difference of the likelihood functions of lth channel $\left[f\left(M_1^{(l)}(v_3) - M_2^{(l)}(v_3)\right)\right]$, nonlinear bit filters $(NF^1, \ldots, NF^g)$, memory units $\left(\mathrm{SD}_1^{(l)}\right)$ for delay of image elements in horizontal, vertical and diagonal lines, and $\left(\mathrm{SD}_2^{(l)}\right)$ for storage of transitions probabilities matrices element values $\left({}^1\Pi^{(l)}, {}^2\Pi^{(l)}, {}^3\Pi^{(l)}\right)$, summers $\Sigma_1^{(l)}$, computational units for nonlinear function (Eq. 11.11), the same threshold devices $\left(TD^{(1)}, \ldots, TD^{(g)}\right)$, units

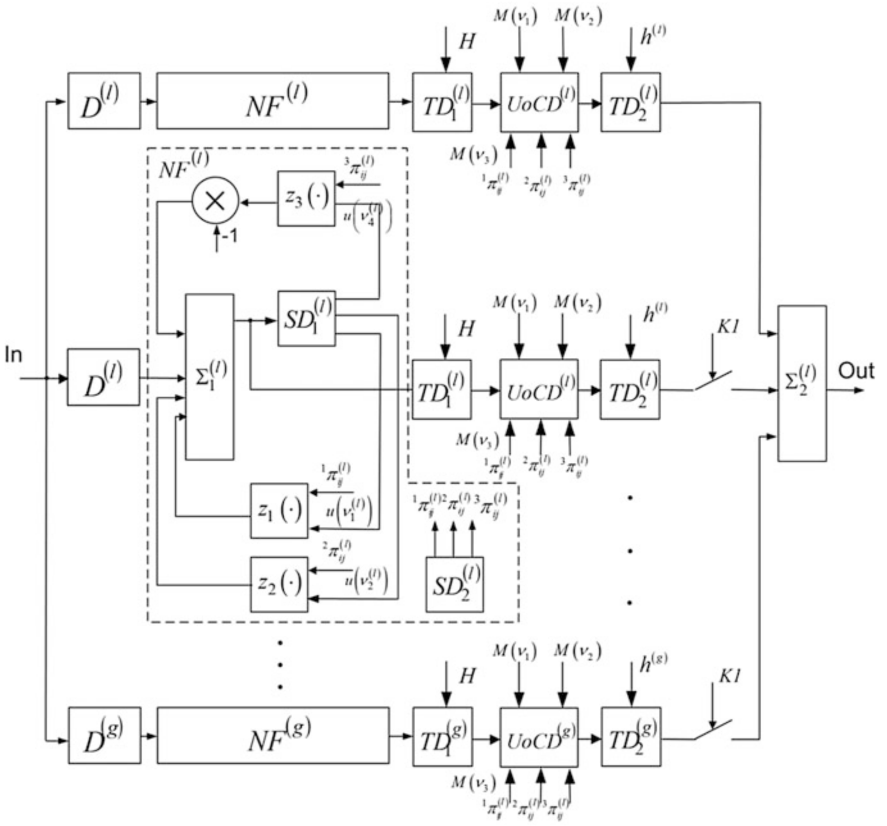


Fig. 11.6 The device of combined method of contours detection

of contours detection ($UoCD^{(l)}$), which using states of neighboring elements from the nearest neighborhood $\Lambda_{i,j}^{(l)}$ of element $v_3^{(l)}$ and transitions probabilities from memory units $SD_2^{(l)}$ to calculate the information content in the lth BBI element. Then the calculated values $I\left(v_3^{(l)}\right)$ are compared with the threshold values $h^{(l)}$, which have different values for each BBI.

The number of channels connected to the adder Σ_2 is determined by brightness of allocated areas and degree of noise in an image, and usually is equal 1–2. Lower BBIs constitute the background. Therefore, to detect contours it is sufficient $g/2$ number of channels, corresponding to the upper (8–5) BBIs. To connect the lth channel to the adder Σ_2, it is necessary to close the key K_1.

If the apriori data about statistical characteristics are unknown, it is necessary to use the adaptive filtering algorithm [27, 28]. The adaptation reduces the computational costs of the estimates for elements of transitions probabilities matrices (Eq. 11.2) and substitutes them into the Eq. 11.10. In this case, the adaptation unit that calculates the estimations ${}^{1}\widehat{\pi}_{ii}^{(l)}$, ${}^{2}\widehat{\pi}_{ii}^{(l)}$, and ${}^{3}\widehat{\pi}_{ii}^{(l)}$ of the received signal are added into the device for nonlinear filtering.

Consider the example of noisy image segmentation of oil spill with $\rho_{in}^{2} = -3\,\text{dB}$ represented in Fig. 11.7. Figure 11.7a shows the initial image. Figure 11.7b is a contour image. Figure 11.7c shows a noisy image with white Gaussian noise, and Fig. 11.7d is a filtered image. Figure 11.7e shows the segmentation of initial image, and Fig. 11.7f provides the result of noisy image segmentation. The segmentation is performed using two upper BBIs of DHI.

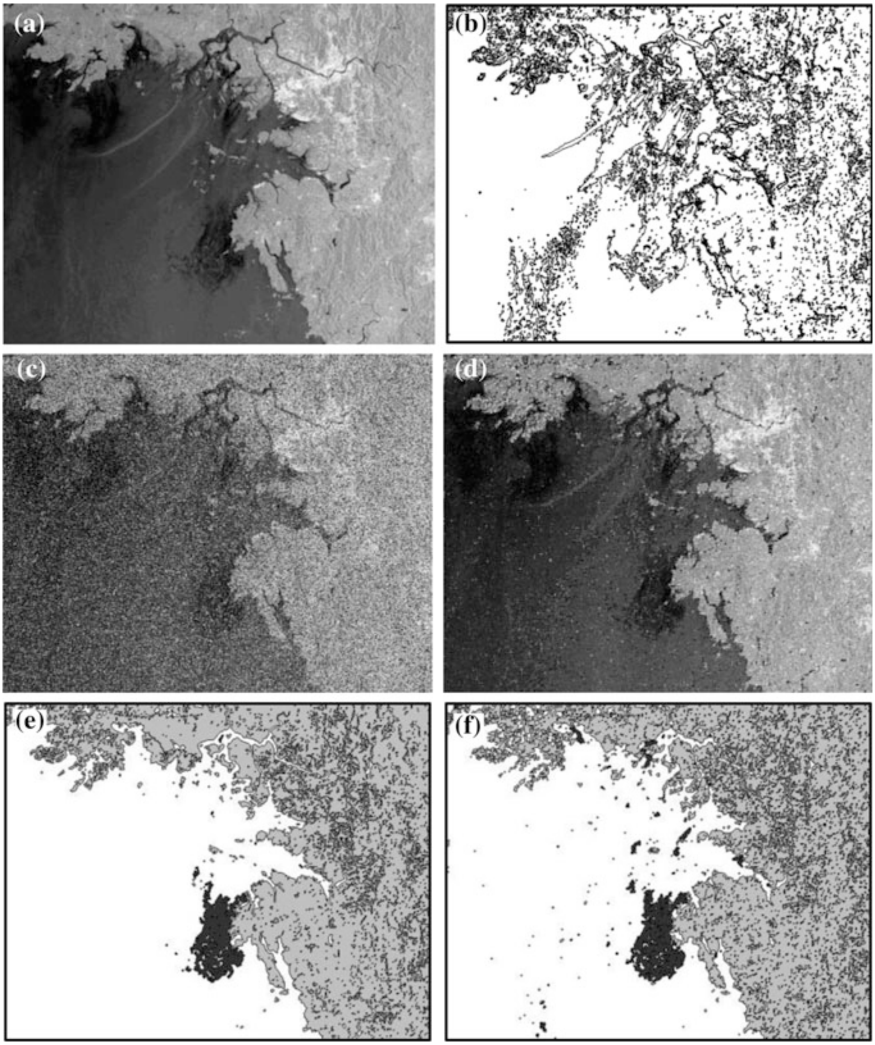


Fig. 11.7 The example of noisy DHI segmentation: **a** initial image, **b** contour image, **c** noisy image ($\rho_{in}^{2} = -3\text{dB}$), **d** filtered image, **e** the result of segmentation of initial image, **f** result of segmentation of noisy image

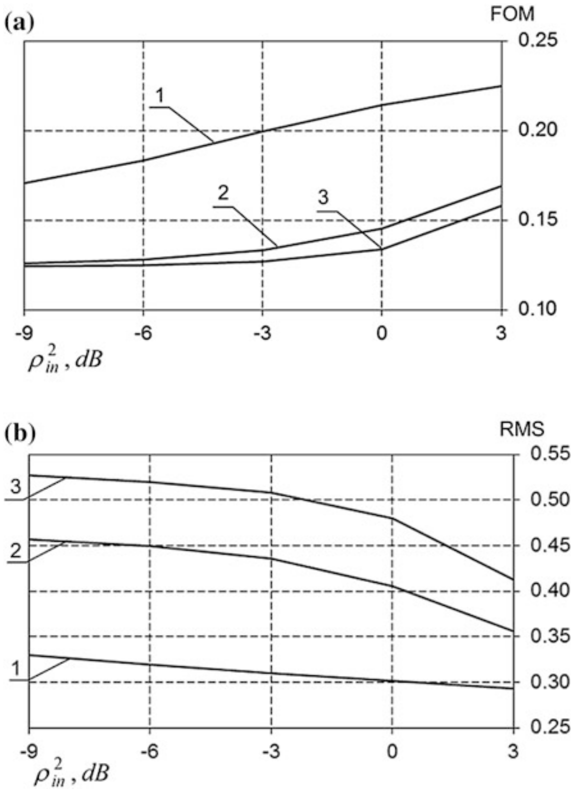


Fig. 11.8 Criteria FOM and RMS for different methods of contour allocation at different signal to noise ratio (*1* the developed method, *2* Laplacian of Gaussian; *3* Canny method): **a** the FOM criterion; **b** the RMS criterion

In known methods of contours detection, such as Canny or Laplacian of Gaussian, the image smoothing with Gaussian filter is used at the first stage to improve noise immunity. The criteria FOM and RMS for different methods of contour allocation at different signal to noise ratio ρ_{in}^2 are situated in Fig. 11.8.

The developed method by FOM criterion at $-9\,\text{dB} \le \rho_{in}^2 \le 3\,\text{dB}$ exceeds the known methods by 33–49 % and by RMS criterion exceeds by 18–29 %. Criterion E for developed method for all signals to noise ratios is 0.

The results of our research show that the combined method of contours detection is effective to allocate objects of interest in noisy images under the signal to noise ratio at the receiver input to –9 dB.

11.4 Method for Texture Image Segmentation

Often in the images of Earth's surface, there are extensive areas with similar statistical characteristics that do not have clearly determined borders and significant details such as the area of the forest, the area of urban developments, etc. In this case, it is necessary to analyze the texture of these areas for their identification.

It is proposed to use the estimate of transition probabilities in 2D Markov chain as a textural characteristic. Assume that the textural characteristic does not vary appreciably over an area, but possesses considerably different values in different areas. Consider that main detailed areas are clear in upper BBI of DHI. It is necessary to process the upper BBI with the most pronounced textural characteristics to locate those areas.

The 2D mathematical model of image (Eq. 11.3) can be used to determine a transition probability $\pi_{iii}^{(l)}$ in two-dimensional Markov chain. The estimation of horizontal transition probability ${}^{1}\widehat{\pi}_{ii}^{(l)}$ depends on average length of sequence of like-sign elements [27, 28] in Eq. 11.15, where $\widehat{\chi}^{(l,r)}$ is the estimated value of the average length of the sequence of equal elements in the lth BBI on the rth step of the estimate adjustment, $p_1^{(l)}$ is the initial probability $\left(p_1^{(l)} = 0.5\right)$.

$$ {}^{1}\widehat{\pi}_{ii}^{(l)} = 1 - \frac{2p_1^{(l)}}{\widehat{\chi}^{(l,r)}} \tag{11.15}$$

From the second line of the BBI by using the set of elements $\Psi = \left\{v_1^{(l)}, v_2^{(l)}, v_3^{(l)}, v_4^{(l)}\right\}$ and the previously calculated estimations of horizontal transition probability ${}^{1}\widehat{\pi}_{ii}^{(l)}$, one can calculate the vertical transition probability ${}^{2}\widehat{\pi}_{ii}^{(l)}$ and estimate of transition probability in 2D Markov chain $\widehat{\pi}_{iii}^{(l)}$ by Eq 11.16, where ${}^{3}\widehat{\pi}_{ii}^{(l)} = {}^{1}\widehat{\pi}_{ii}^{(l)2}\widehat{\pi}_{ii}^{(l)} + {}^{1}\widehat{\pi}_{ij}^{(l)2}\widehat{\pi}_{ij}^{(l)}$.

$$ \widehat{\pi}_{iii}^{(l)} = \frac{{}^{1}\widehat{\pi}_{ii}^{(l)} \cdot {}^{2}\widehat{\pi}_{ii}^{(l)}}{{}^{3}\widehat{\pi}_{ii}^{(l)}} \tag{11.16}$$

The method of "scanning frame" calculates the local changes of statistical characteristics. A value of frame sizes depends on the specified accuracy and the requirement to minimize computational costs.

During the processing of the first rows and columns, the frame size increases according to the processing element index until it reaches the adjusted value. On each step of processing, the frame sizes are determined by Eq. 11.17, where i, j are the coordinates of processed element, M and N are a width and a height of the scanning frame, respectively.

$$ \begin{gathered} m = \overline{1, (2i-1)} \quad i = \overline{2, (M-1)/2} \\ n = \overline{1, (2j-1)} \quad j = \overline{2, (N-1)/2} \end{gathered} \tag{11.17}$$

The average transition probability $\widetilde{\pi}_{iii}^{(l)}$ for the central element of the frame is estimated by counting the average transition probability within the frame as shown in Eq. 11.18.

$$\widetilde{\pi}_{iii}^{(l,r,k)} = \frac{1}{m \times n} \sum_{r=1}^{m} \sum_{k=1}^{n} \widehat{\pi}_{iii}^{(l,r,k)} \tag{11.18}$$

In the next step, the frame is shifted by one element to the right and downwards, and the average transition probability is estimated for every element of the BBI.

To allocate the areas with different textures, it is necessary to compare the calculated estimation $\widetilde{\pi}_{iii}^{(l)}$ with a threshold value. The threshold value h between different textures is the estimation of $\widetilde{\pi}_{iii}^{(l)}$, the choice of which being based on the analysis of image histogram [29]. If the image contains areas of two textures with different statistical characteristics, two labels (0 and 1) will be enough to enumerate them. All elements, for which estimate exceeds the threshold value, are labelled with 1, all the rest ones are labelled with 0. If the DHI contains several textures, then for each texture there should be a unique label. In this case, several threshold values corresponding to different textures are used.

To estimate a quality of texture areas allocation method, a number of wrongly segmented elements is calculated. The segmented image is compared with the ideal mapping to determine a number of wrongly segmented elements by Eq. 11.19, where N and M are a height and a width of image, respectively, $F(i, j)$ is a variable that takes a value 0, when the element of image is segmented correctly, and a value 1, otherwise.

$$ESE = \frac{1}{N \cdot M} \sum_{i=1}^{h} \sum_{j=1}^{w} F(i,j) \tag{11.19}$$

The quality estimation of texture segmentation is performed using a set of artificial images, which are formed by specified mapping. They contain such statistical characteristics that have constant value within the area and different values in other areas. The borders of mapping are generated by 2D mathematical model with constant transitions probabilities matrix [18, 19]. The quality of texture areas allocation depends on a scanning frame size and a threshold value.

The Exonic Splicing Enhancers (ESE) dependence from the frame size at different transitions probabilities values of segmented textures is shown in Fig. 11.9. The frame sizes 21 × 21 pixels are acceptable for the majority of textures and the most efficient in terms of the ratio segmentation quality to processing time. If the frame size 21 × 21 pixels is used, then the developed method allows the separation of image in texture areas, for which the difference between transition probabilities is equal 0.15. In this case, a segmentation error does not exceed 6 %. A number of histogram peaks equals a number of textures in texture image. Therefore, a threshold value, which is used to separate the textures, can be chosen from histogram analysis like a minimum value between two adjacent peaks of the histogram.

The modeling used the real and artificial images formed by means of 2D mathematical model, and the algorithm is developed by Petrov et al. [18, 19].

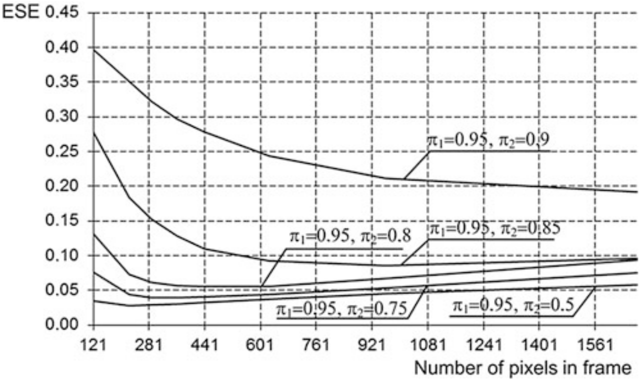
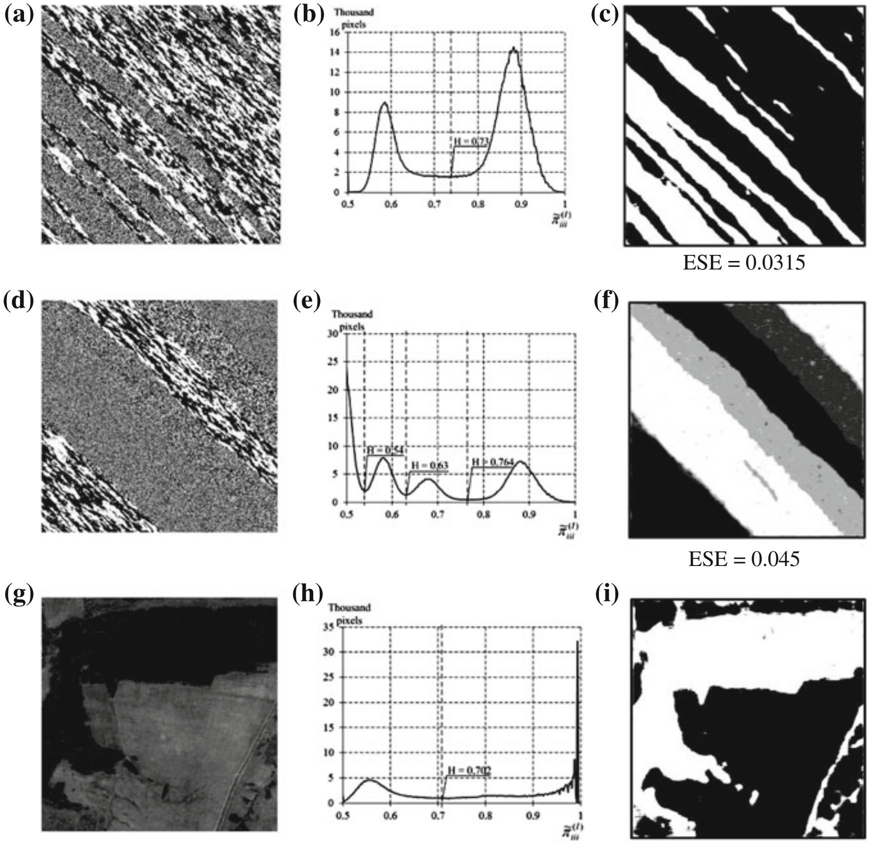


Fig. 11.9 The ESE dependence from the frame size



ESE = 0.0315

ESE = 0.045

Fig. 11.10 Segmentation of texture images: **a**, **d**, **g** artificial images with two and four textures and satellite image, **b**, **e**, **h** histograms of texture characteristic image, **c**, **f**, **i** results of segmentation

Table 11.2 The ESE criterion for artificial and satellite images

Type of images	Developed method	Canny	Sobel	LoG
Artificial test image	0.0082	0.3523	0.2466	0.3073
Artificial image based on real images	0.0079	0.7665	0.4811	0.5956
Satellite image	0.0388	0.7268	0.4707	0.5969

The example of segmentation of artificial images with two and four textures and also real satellite image is represented in Fig. 11.10. In the case of satellite image, the segmentation was performed using the sixth BBI. The threshold value is selected from a histogram. The frame size is 21 × 21 pixels. The artificial images with two and four textures and satellite image are situated in Fig. 11.10a, d, g. The histograms of texture characteristic image are located in Fig. 11.10b, e, h. The segmentation results one can see in Fig. 11.10c, f, i.

The calculated results of the ESE criterion by the developed method and also known contour methods are listed in Table 11.2.

11.5 Conclusion

The proposed method of texture segmentation based on the random Markov fields allocates the areas with different textures efficiently. Thus, the segmentation error is reduced more than in 10 times in comparison with known methods based on a gradient calculation. If the developed method is applied to each color components, then a texture segmentation of color images can be performed. The results of texture segmentation of color components are combined to one color segmented image, on which areas with different textures are allocated by the different colors.

References

1. Gonzalez RC, Woods RE (2008) Digital image processing. Prentice Hall, Upper Saddle River
2. Zhuravlev JI, Gurevich IB (1989) Pattern recognition and image recognition. Recogn Classif Prognosis 2(2):5–72 (in Russian)
3. Jahne B (2005) Digital image processing: concepts, algorithms, and scientific applications. Springer, Berlin
4. Soifer VA (2003) Methods of computer processing of images. Fizmatlit, Moscow (in Russian)
5. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8(6):679–698
6. Pratt WK (2001) Digital image processing. Wiley Interscience, New York
7. Prewitt JMS (1970) Object enhancement and extraction. In: Lipkin BS, Rosenfeld A (eds) Picture processing and psychopictorics. Academic Press, New York
8. Sobel IE (1970) Camera models and machine perception. PHD dissertation, Stanford University

9. Roberts LG (1965) Machine perception of three-dimensional solids. In: Tippett JT (ed) Optical and electro-optical information processing, MIT Press, Cambridge
10. Kirsch R (1971) Computer determination of the constituent structure of biological images. Comput Biomed 4:315–328
11. Deriche R (1990) Fast algorithms for low-level vision. IEEE Trans Pattern Anal Mach Intell 12(1):78–87
12. Seise M, McKenna SJ, Ricketts IW, Wigderowitz CA (2007) Learning active shape models for bifurcating contours. IEEE Trans Med Imaging 26(5):666–677
13. Seise M, McKenna SJ, Ricketts IW, Wigderowitz CA (2005) Double contour active shape models. In: British machine vision conference (BMVC), vol 2, pp 159–168
14. Martin D, Fowlkes C, Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 26(5):530–549
15. Meer P, Georgescu B (2001) Edge detection with embedded confidence. IEEE Trans Pattern Anal Mach Intell 23(12):1351–1365
16. Wang H, Dong Y (2008) An improved image segmentation algorithm based on Otsu method. In: International symposium on photoelectronic detection and imaging: related technologies and applications, SPIE, vol 6625, pp 1–8
17. Derin H, Kelly P (1989) Random processes of Markov type with discrete arguments. TIEEE 77(10):42–71
18. Petrov EP, Medvedeva EV, Metelyov AP (2011) Method of synthesis of video images mathematical models based on multidimensional Markov chains. Nonlinear World 4:213–231 (in Russian)
19. Petrov EP, Trubin IS, Medvedeva EV, Smolskiy SM (2013) Mathematical models of video-sequences of digital half-tone images. In: Atayero AA, Sheluhin OI (eds) Integrated models for information communication system and networks: design and development. IGI Global, Hershey
20. Akasi A (1981) Recovering of Gaussian images with the help of two-dimension maximal a posteriori estimation. J Densi Tsusin Gakkai Rombusini A-64(11):908–915
21. Martin D, Fowlkes C. The berkeley segmentation dataset and benchmark. http://www.cs.berkeley.edu/projects/vision/grouping/segbench/. Accessed 15 June 2014
22. Rogers D (1998) Procedural elements for computer graphics. WCB/McGraw-Hill Inc, New York
23. Pavlidis T (1986) Algorithms for graphics and image processing. Radio and Communication, Moscow (in Russian)
24. Petrov EP, Medvedeva EV (2010) Nonlinear filtering of statistically connected video sequences based on hidden markov chains. J Commu Technol Electron 55(3):307–315
25. Petrov EP, Trubin IS, Medvedeva EV, Smolskiy SM (2013) Development of nonlinear filtering algorithms of digital half-tone images. In: Atayero AA, Sheluhin OI (eds) Integrated models for information communication system and networks: design and development. IGI Global, Hershey
26. Tikhonov VI (1966) Statistical radio engineering. Sov. Radio, Moscow (in Russian)
27. Medvedeva EV, Kurbatova EE (2011) A two-stage image preprocessing algorithm. Pattern Recogn Image Anal 21(2):297–301
28. Trubin IS, Medvedeva EV (2008) The adaptive nonlinear filtering of static and dynamic gray scale images. In: 9th international conference on pattern recognition and image analysis: new information technologies (PRIA-9-2008): conference proceeding, vol 2, pp 222–225
29. Shapiro LG, Stockman GC (2001) Computer vision. Prentice-Hall, Upper Saddle River