Studies in Systems, Decision and Control 33

Aleksander Nawrat
Karol Jędrasiak *Editors*

Innovative Simulation Systems

Springer

Studies in Systems, Decision and Control

Volume 33

About this Series

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control- quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/13304

Aleksander Nawrat · Karol Jędrasiak
Editors

Innovative Simulation Systems

Springer

Editors

Aleksander Nawrat
Institute of Automatic Control
Silesian University of Technology
Gliwice
Poland

Karol Jędrasiak
Institute of Automatic Control
Silesian University of Technology
Gliwice
Poland

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-319-21117-6 ISBN 978-3-319-21118-3 (eBook)
DOI 10.1007/978-3-319-21118-3

Library of Congress Control Number: 2015943839

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Be nice to people on your way up
because you meet them on your way down.

Jimmy Durante

Preface

The book has been motivated by rapid development of technology allowing us to create innovative simulation systems designed to deliver broader content; at the same time enhancing educational opportunities and reducing the economic cost of teaching, in particular, in areas such as national defense, medicine, or sport. The concept of a simulation system is extensive and contains high-growth areas such as augmented reality or design and construction of unmanned vehicles.

Augmented reality (AR) is a term used to describe the process of enriching the perception of the real world with digital elements, generated by a computer. "Augmentation" of reality is typically performed in real time and in a contextual conformity with the observed real world. Mainly, the information added is a multimedia extension of the senses of the observer. Many research groups are working on projects and implementation of simulation systems containing elements of AR, which are used in many fields such as navigation, rehabilitation, entertainment, and video surveillance of the city.

The main aim of design and construction of UAVs is their practical application. However, their design is a lengthy process and requires intensive evaluation using both simulation and experimental data. Human perception is limited to tracking in full concentration only seven elements simultaneously. Control of unmanned vehicle and analysis of data recorded by the payload of unmanned vehicles is a significant excess of information for human and negatively impacts its analytical capabilities. Therefore, it is often necessary to apply amendments based on the actual real life experience of the application.

The goal of this monograph is to provide comprehensive guidelines on the current and future trends of innovative simulation systems; in particular, their important components, such as augmented reality and unmanned vehicles.

The book consists of three parts: Vision-based Information for Innovative Simulation Systems, Design, Construction and Analysis for Purpose of Innovative Systems and Design and Evaluation of Control Algorithms. Each part presents good practices, new methods, concepts of systems, and new algorithms. Presented challenges and solutions are the results of research conducted by the contributing authors.

The monograph can be a valuable source of information for those in contact with the design and creation of innovative simulation systems as researchers, programmers, designers, engineers, and users. The book is addressed to a wide audience: academic staff, representatives of research institutions, employees of companies and government agencies as well as students and graduates of technical universities in the country and abroad. We took an effort to present multiple solutions and challenges in the fields above. The authors encourage to actively explore the following chapters of the book and, in case of questions or concerns, invite you to an open and frank discussion of these matters.

In conclusion, the main features of the book are the following:

1. The book describes and evaluates the current state of knowledge in the field of innovative simulation systems.
2. Throughout the chapters there are presented current issues and concepts of systems, technology, equipment, tools, research challenges and current, past, and future applications of simulation systems.

We would like to thank the authors for their contributions. Without their knowledge, experience, and commitment this monograph would have never appeared. We also thank the reviewers, editors, and staff of Springer for their support during the creation of this monograph.

Gliwice
April 2015

Aleksander Nawrat
Karol Jędrasiak

Acknowledgment

The work was financially supported by NCBiR grant POIG 02.03.01-24-099 GeCONiI—Upper Silesian Centre for Computational Science and Engineering.

Contents

Part I Vision-Based Information for Innovative Simulation Systems

Minimizing the Image Resolution in Order to Increase the Computing Speed Without Losing the Separation of the Recognised Patterns 3
Zygmunt Kuś and Aleksander Nawrat

The Method of Guaranteeing the Separation Between the Recognised Object and Background 19
Zygmunt Kuś and Aleksander Nawrat

The Method of Developing the Invariant Functions Vector for Objects Recognition from a Given Objects Set. 33
Zygmunt Kuś and Aleksander Nawrat

Adjusting the Thresholds to the Recognised Pattern in Order to Improve the Separation Between the Recognised Patterns 47
Zygmunt Kuś and Aleksander Nawrat

The Concept of an Active Thermal Camouflage for Friend-Foe Identification System 67
Paulina Wilk, Tomasz Targiel, Dawid Sobel, Jan Kwiatkowski, Karol Jędrasiak and Aleksander Nawrat

Augmented Reality in UAVs Applications. 77
Paweł Iwaneczko, Karol Jędrasiak and Aleksander Nawrat

Real Time Dense Motion Estimation Using FPGA Based Omnidirectional Video Acquisition Device 87
Jan Kwiatkowski, Dawid Sobel, Artur Ryt, Mariusz Domżał, Karol Jędrasiak and Aleksander Nawrat

GPU-Based Parameters Estimation for Anisotropic Diffusion 109
Mariusz Domżał, Artur Ryt, Dawid Sobel, Jan Kwiatkowski, Karol Jędrasiak and Aleksander Nawrat

An Evaluation of SETh—The Method for Long-Term Feature Tracking. 119
Karol Jędrasiak and Aleksander Nawrat

The System for Augmented Reality Motion Measurements Visualization 189
Dawid Sobel, Jan Kwiatkowski, Artur Ryt, Mariusz Domżał, Karol Jędrasiak, Łukasz Janik and Aleksander Nawrat

Real-Time Multiple Laser Points Tracking 201
Artur Ryt, Dawid Sobel, Jan Kwiatkowski, Mariusz Domżał, Karol Jędrasiak and Aleksander Nawrat

Part II Design, Construction and Analysis for Purpose of Innovative Systems

Numerical and Experimental Analysis of a Truck Frame 217
Eugeniusz Rusiński, Artur Iluk and Mariusz Stańco

Designing and Implementing Elements of a Vehicle Model in the VBS2 Virtual Simulation Environment 233
Roman Wantoch-Rekowski and Konrad Szumiec

Semi-active Suspension System for 2S1 Tracked Platform in Drive Comfort Improvement Application 259
Tomasz Nabagło, Andrzej Jurkiewicz and Janusz Kowal

Experimental Mobile Robot—Hardware. 277
Krzysztof Jaskot and Tomasz Łakota

Conception of a Diagnostic System for Evaluating a Technique Correctness and Effectiveness of Running 291
Krzysztof Skrzypczyk

Part III Design and Evaluation of Control Algorithms

Experimental Mobile Robot—Software 303
Krzysztof Jaskot and Tomasz Łakota

Application of Genetic Algorithms for Identification of Simulated Systems 317
Martyna Ulinowicz and Janusz Narkiewicz

Quadrotor Dynamics and Control for Precise Handling 335
Przemysław Bibik, Janusz Narkiewicz, Maciej Zasuwa and Marcin Żugaj

The Application of the Modified BLT Method for the Synthesis of UAV's Control System 353
Zygmunt Kuś and Aleksander Nawrat

The Modified BLT Method for Multivariable Control Systems 363
Zygmunt Kuś and Aleksander Nawrat

Analysis of Human Arm Nonlinear and Linear Mathematical Models 373
Robert Bieda and Krzysztof Jaskot

Naive Kalman Filtering for 3D Object Orientation 399
Robert Bieda, Rafal Grygiel and Adam Galuszka

Non-GPS Navigation System for Criminalistic Investigation on Mobile Robot 411
Robert Głębocki, Paweł Kicman and Janusz Narkiewicz

Path Planning Algorithms for Autonomous Mobile Platform 423
Krzysztof Jaskot and Radomir Mysliwiec

Part I
Vision-Based Information for Innovative Simulation Systems

Progress of miniaturization and increase in resolution of widely available video cameras allowed a rapid progress of vision-based innovative simulation systems. With the advent of high-quality smartphones enabling not only the acquisition, but at the same time processing and displaying of data, it became possible practical application of the concept of augmented reality (AR).

The part presents an example of Attitude Indicator Augmented Reality (AR) control for the unmanned aerial vehicles (UAV) ground control station. The article describes perspective heads-up display overlaid on the image, which is acquired from the UAV rotatable camera. Article shows the mechanisms of aircraft camera calibration with the AR artificial horizon. The whole algorithm of the render instructions is presented. Elements used in the augmented reality are as follows: 3D artificial horizon, latitude and longitude, GPS info, executed command, time to command end, percent of command accomplish, fuel and battery level, height and speed vertical scale, landing field direction arrow, unmanned vehicle marks.

Another discussed application of AR is an attempt to improve existing methods of measuring the range of motion of people during rehabilitation. Range of motion measurements are used by physicians, physical therapists and veterinarians. There is a need for a solution that would increase the accuracy and repeatability of the measurements obtained. Commonly used methods based on tests using a classical goniometer are often inaccurate and depend on the individual interpretation of result by the therapist. The use of motion capture data to RoM measurements results in an accuracy of order of tens of micrometers, which is mainly due to the precision of the Motion Capture system. During the test, patient is not constrained by any device and has a total freedom of movement. It allows to carry out a medical examination not only during flexion and extension of the one limb but more limbs during dynamic movement such as walking.

However, one should keep in mind that to a large extent the effectiveness of augmented reality systems depends on use for its implementation video acquisition components and tracking algorithms.

One of the main problems of image acquisition by the UAV is the need for the use of opto-electronic gimbals which tend to be expensive, and the moving parts are subject to frequent breakdowns. An alternative may be a multi objective solution devoid of moving parts. One of the solutions designed in such a way is *Omnidirectional Video Acquisition Device* (OVAD), a device, which was implemented using analog cameras and STM32 microcontroller, has proven its validity during testing process. The device may be found useful in many applications, both civilian and military. However, size and weight of the device made it difficult to implement OVAD on small unmanned vehicles. Proposed solution is FPGA-based OVAD, which is the result of OVAD further development.

Due to the significant differences in the acquired image from thermal imaging cameras and visible light cameras it is often necessary to develop independent algorithms for filtering and tracking the objects of interest pointed by the human operator. Within the part there is presented a novel long-term object tracking method called SETh. It is an adaptive tracking by detection method which allows near real-time tracking within challenging sequences. The algorithm consists of three stages: detection, verification and learning. In order to measure the performance of the method, video data set consisting of more than a hundred videos was created and manually labeled by a human. Quality of the tracking by SETh was compared against five state-of-the-art methods. The presented method achieved results comparable and mostly exceeding the existing methods, which proved its capability for real life applications like vision-based control of UAVs.

Another study was to examine the influence of image thresholding on the correctness of pattern recognition in grey scale images. The method based on moment invariants, which were the elements of feature vectors defining the features of the recognized object, was used by authors in order to recognize the objects. The article presents the influence of image thresholding with histogram equalization for exemplary images on the distribution of the distance between pattern vector and feature vector for every pixel of an image. The authors have paid great attention to the fact that proper selection of the thresholds is significant for distinguishing given object classes. The aim of the following study was to develop a procedure which guarantees the separation between the part of an image where we have the recognized object and the part of the image which corresponds to the terrain where the object moves. The authors have presented the method which uses moment invariants for creating feature vectors which define the features of the recognized object and the features of the background. The presented method is based on calculating the distance between the values of invariant functions calculated for an object and the background. On top of that, the study presents the example of recognizing the object moving in various types of a terrains.

Concluding, the part includes a number of important challenges in the fields mentioned above. At the same time valuable suggestions and conclusions from authors are presented and discussed in detail.

Minimizing the Image Resolution in Order to Increase the Computing Speed Without Losing the Separation of the Recognised Patterns

Zygmunt Kuś and Aleksander Nawrat

Abstract The aim of the following study was to examine the influence of image resolution in the pattern recognition in the grey scale images. In order to recognise the pattern, the authors used the method based on moment invariants which were the elements of feature vectors defining the features of the recognised object. The paper presents the influence of image resolution for exemplary images on both: the values of moment invariants and distances between feature vectors. The authors have paid a great attention to the fact that these distances are significant for distinguishing given object classes. One can conclude from the results that for a significant decrease in resolution there occur problems in pattern recognition. It results from the influence of image resolution on the value of moment invariants and at the same time on the value of the distance between the feature vectors defining the recognised objects. In this way, the paper shows that in order to recognise objects correctly, it is necessary to retain some necessary minimum resolution. It is indispensable despite the fact that we usually aim at decreasing the amount of processed data which is on the hand crucial because of short processing times in many practical applications. It is therefore essential because of the fact that we need to guarantee short times of image processing in many practical applications. Moreover, this study presents the examples of the algorithms in use.

Keywords Object recognition · Moment invariants · Pattern vector · Image resolution

Z. Kuś (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Akademicka 16 Street,
Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_1

1 Introduction

Image processing encounters multiple problems in the preliminary stages of image processing. One of the first stages in image processing is to decide about the image resolution. It lets us obtain a less amount of data to process; however, it should not result in deteriorating pattern recognition.

There are some problems during the acquisition of the image which influence the required image resolution and consequently they impact on the quality of the pattern recognition.

Firstly, each measurement in the real world is burdened with some error and can be disturbed by a some kind of noise [1–3]. When the camera works in various atmospheric conditions, then rain, snow or the reflections of the sunlight can result in an image with noise.

Secondly, the amount of data which can be stored and processed is usually smaller than the amount of data acquired by the camera system [4, 5].

Therefore, on the one hand, we would like to have the highest resolution to distinguish artefacts in the image, which are the effect of disturbance, from the significant elements of the image. On the other hand, we would like to have the smallest amount of data to store and process.

The following study will focus on developing a method of calculating the resolution for the acquisition of the grey scale images. Changing the resolution will be conducted in order to decrease the amount of data defining the image and it will be carried out to speed up the calculations during the recognition of the pattern (object) location in the image. Decreasing the data included in the image after decreasing the resolution must not decrease the effectiveness of pattern recognition in the image [6–8].

Many solutions of a pattern recognition methods can be found in literature [9–15] and a lot of them is based on the feature vectors which elements are moment invariants [16, 17].

This paper will study the influence of decreasing the image resolution on the values of the moment invariants which are used to recognise an object. We will analyse the images of 256 grey scale levels.

2 Formulating the Problem of the Resolution Change for Grey Scale Images

The problem of the image resizing in image processing is discussed at greater length in the literature. In the body of literature, one can find various techniques for example: nearest neighbour, bilinear interpolation, or bicubic interpolation.

The image resizing, either upscaling or downscaling resulting from the need for increased or decreased image size is presented in [18]. The image resizing method presented in [18] is based on the analysis of the sum of primary implicants representation of image data, as generated by a logical transform. The algorithm used in the aforementioned study emulates the nearest neighbour technique, while subsequent variations build on this to provide more accuracy and output image comparable to the other traditional methods.

The other approach to this problem is discussed in [19]. This paper presents an adaptive interpolation algorithm which is based on the Newton Polynomial in order to reduce the limitation of the traditional algorithm for image resizing.

In the next paper [20], a comprehensive study and comparison of different image scaling algorithms was presented. In [20], the authors compared the major methods of image resizing. These methods comprised of nearest neighbour image scaling, bilinear image scaling, bicubic image scaling, Lanczos image scaling and modified bicubic image scaling.

We will use the method based on calculating the mean value to change the image resolution. Moreover, we are going to use the moment invariants as invariant functions for pattern recognition. These moment invariants $(J_1 - J_8)$, which are presented in (1)–(3), will create the feature vector.

$$\begin{aligned}
&J_1 = \frac{I_1}{m_{00}^2}; J_2 = \frac{I_2}{m_{00}^4}; J_3 = \frac{I_3}{m_{00}^5}; J_4 = \frac{I_4}{m_{00}^5};\\
&J_5 = \frac{I_5}{m_{00}^{10}}; J_6 = \frac{I_6}{m_{00}^7}; J_7 = \frac{I_7}{m_{00}^4}; J_8 = \frac{I_8}{m_{00}^5};
\end{aligned} \tag{1}$$

$$\begin{aligned}
I_1 &= M_{20} + M_{02}\\
I_2 &= (M_{20} - M_{02})^2 + 4M_{11}^2\\
I_3 &= (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2\\
I_4 &= (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2\\
I_5 &= (M_{30} - 3M_{12})(M_{30} + M_{12})((M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2)\\
&\quad + (3M_{21} - M_{03})(M_{21} + M_{03})(3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2);\\
I_6 &= (M_{20} - M_{02})((M_{30} + M_{12})^2 - (M_{21} + M_{03})^2)\\
&\quad + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03})\\
I_7 &= M_{20}M_{02} - M_{11}^2\\
I_8 &= M_{30}M_{12} + M_{21}M_{03} - M_{12}^2 - M_{21}^2
\end{aligned} \tag{2}$$

$$\begin{gathered}
\text{Central moments of order } (p+q):\\
M_{pq} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(i - x_s)^p (j - y_s)^q f(i,j)\\
\text{The co-ordinates of the region's centre of gravity (centroid):}\\
x_s = \tfrac{m_{10}}{m_{00}}; \quad y_s = \tfrac{m_{01}}{m_{00}}\\
\text{Moments of order } (p+q):\\
m_{pq} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} i^p j^q f(i,j)
\end{gathered} \tag{3}$$

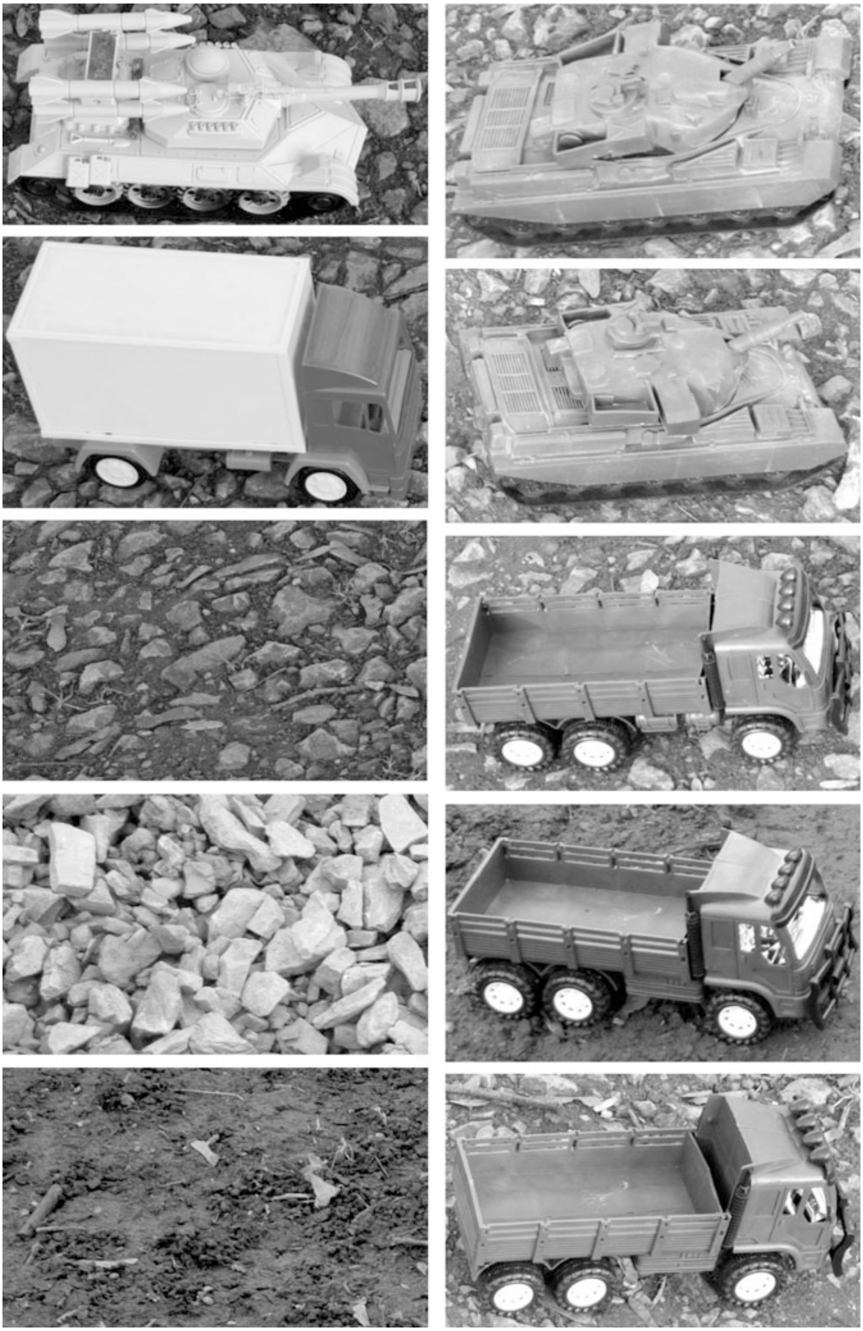


Fig. 1 The set of ten 'objects' which are going to be recognised

where

$f(i, j)$ image pixel,

M the height of the image,

N the width of the image.

Ten pattern images, cf. Fig. 1, (seven images of the vehicles and three images of various terrains) define ten patterns which should be distinguished from each other. The resolution will be changed in this way that in every next step we will obtain the image of four times smaller area.

It means that we will examine the cases for eight values of resolutions: original image—not downscaled and image linearly downscaled in each axis: 2, 4, 8, 16, 32, 64 and 128 times.

Figure 2 presents the examples of the various image resolution on the example of the selected object.

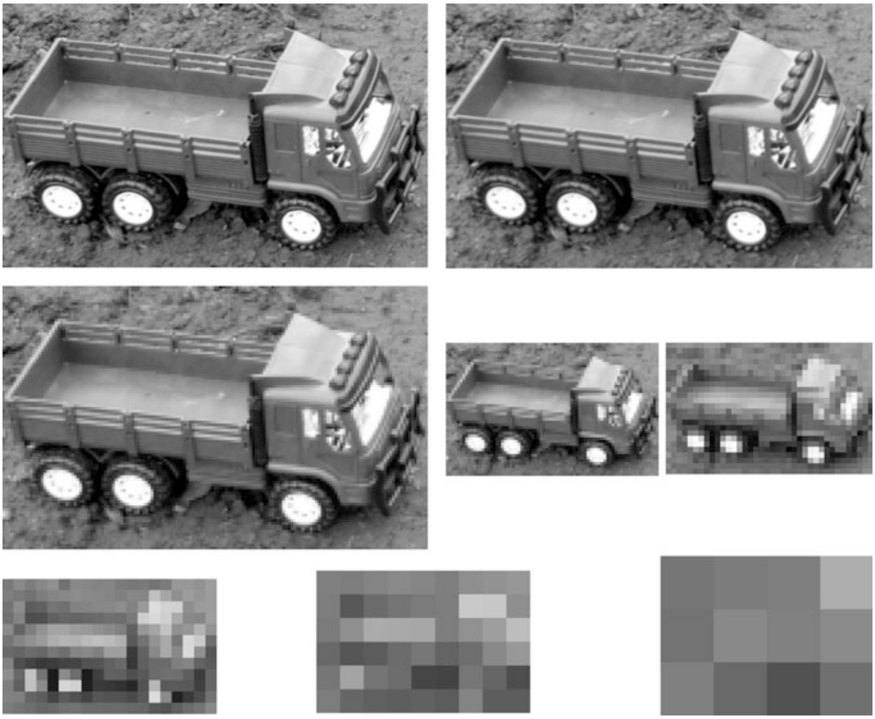


Fig. 2 The examples of the various image resolution on the example of the selected objects

3 Examining the Influence of the Image Resolution on the Moment Invariants Values for Grey Scale Images

To examine the influence of image resolution on pattern recognition, we will use the pattern images presented in Fig. 1. All the images will be resized for the following linear scale rates: 1/1, 1/2, 1/8, 1/16, 1/32, 1/64, 1/128. The values of the moment invariants J_1 to J_8 for all scale rates and exemplary images are presented in Figs. 3, 4, 5, 6, 7, 8, 9 and 10.

Firstly, we will examine the influence of the resolution on the values of the moment invariants. Figures 3, 4, 5, 6, 7, 8, 9 and 10 present the values of J_1 to J_8 for all examined patterns and resolutions.

According to Fig. 3, we can see that the values of J_1 are almost unaltered as the resolution decreases.

According to Fig. 4, we can see that the values of J_2 are slightly variable for the resolutions steps which are lower than 6. When the step of resolution is higher than 5, we can observe that J_2 increases. This takes place for almost all objects.

According to Fig. 5, we can see that the values of J_3 are almost constant for resolution steps lower than 5 whereas the resolution steps are variable for higher ones.

According to Fig. 6, we can see that the values of J_4 are almost constant for the step of resolution lower than 7 whereas the values of J_4 change for steps of resolution equal 7 and 8.

According to Fig. 7, we can see that the values of J_5 are variable when the resolution changes for all objects.

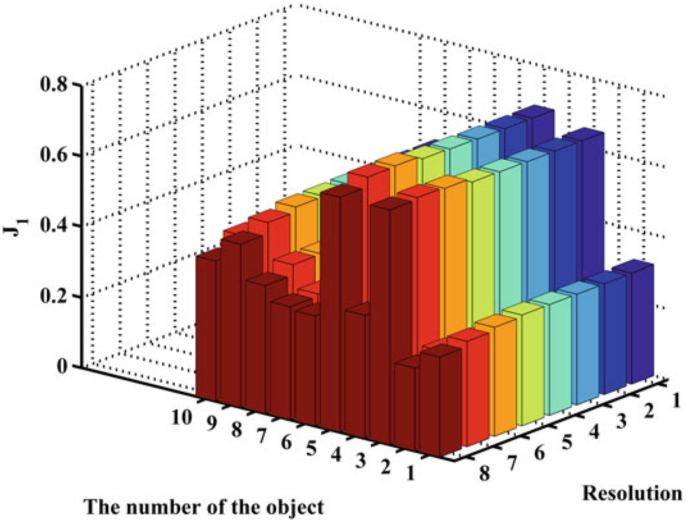


Fig. 3 The graphs of the invariant function J_1 for ten objects and eight resolutions

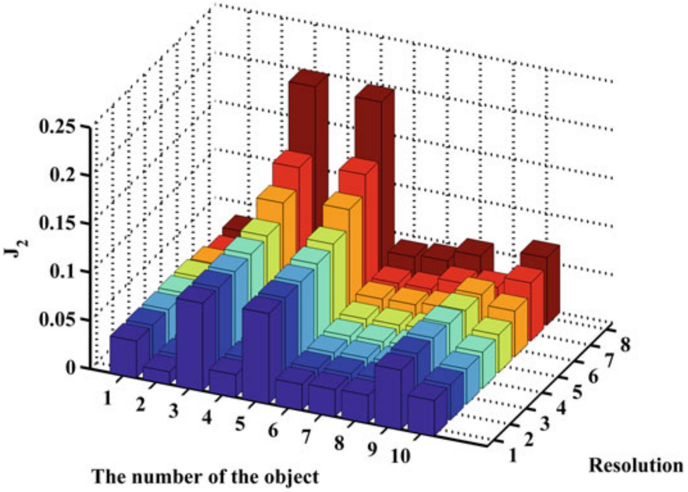


Fig. 4 The graphs of the invariant function J_2 for ten objects and eight resolutions

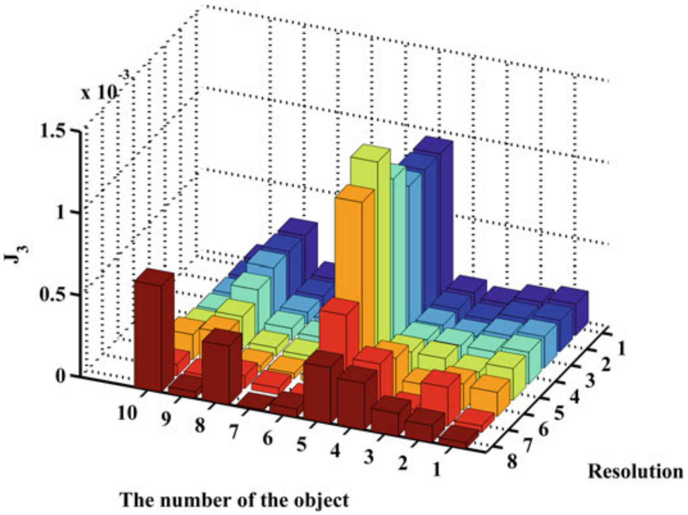


Fig. 5 The graphs of the invariant function J_3 for ten objects and eight resolutions

According to Fig. 8, we can see that the values of J_6 are almost unaltered for the step of resolution lower than 5 whereas for higher steps the value of J_6 increases, decreases or is variable. Consequently, for the higher steps the behaviour of J_6 depends on the number of the object.

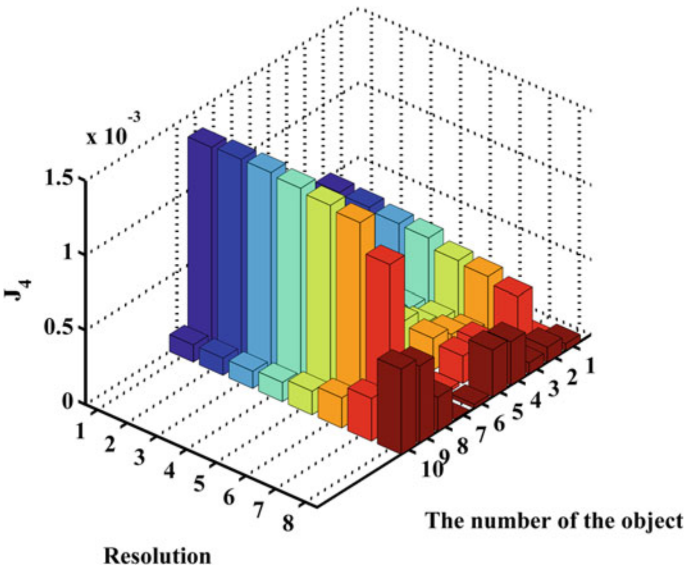


Fig. 6 The graphs of the invariant function J_4 for ten objects and eight resolutions

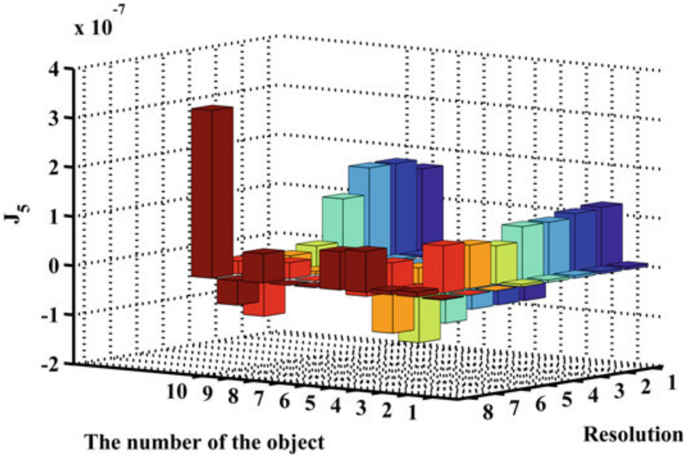


Fig. 7 The graphs of the invariant function J_5 for ten objects and eight resolutions

According to Fig. 9, we can see that the values of J_7 are constant as the resolution decreases to the step number 6. We can see that the values of J_7 decrease as the resolution decreases for the steps 7 and 8.

According to Fig. 10, we can see that the values of J_8 are slightly variable when the step of resolution is lower than some limiting value. We can see the changes of

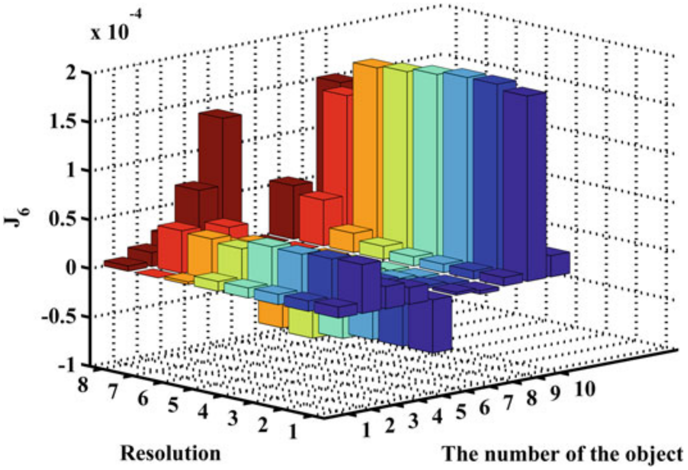


Fig. 8 The graphs of the invariant function J_6 for ten objects and eight resolutions

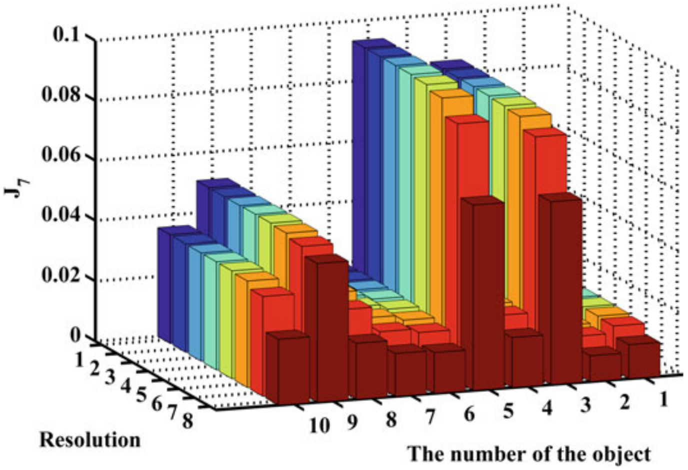


Fig. 9 The graphs of the invariant function J_7 for ten objects and eight resolutions

J_8 for the steps of resolution which exceed this limiting value. The value of this limiting step depends on the number of the object. To sum up, we can state on the basis of Figs. 3, 4, 5, 6, 7, 8, 9 and 10 that invariant functions can assume values considerably differing from high and lower resolutions when the resolution is decreased unduly. One has to consider whether it will have influence or not on image recognition. In order to do this, we will examine how the distances between feature vectors for different objects change along with the resolution change.

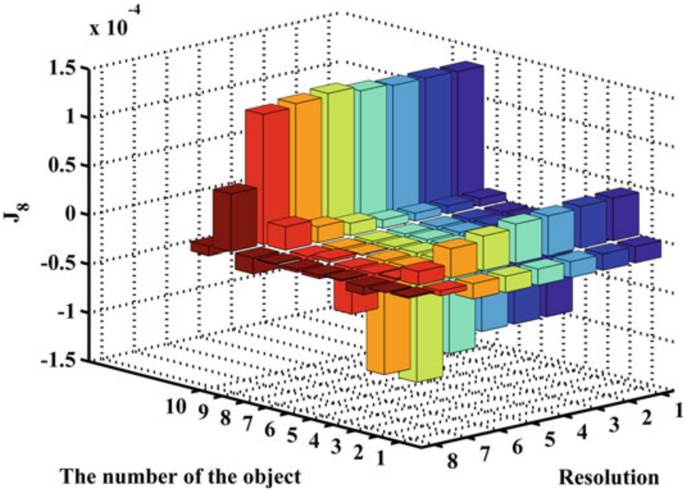
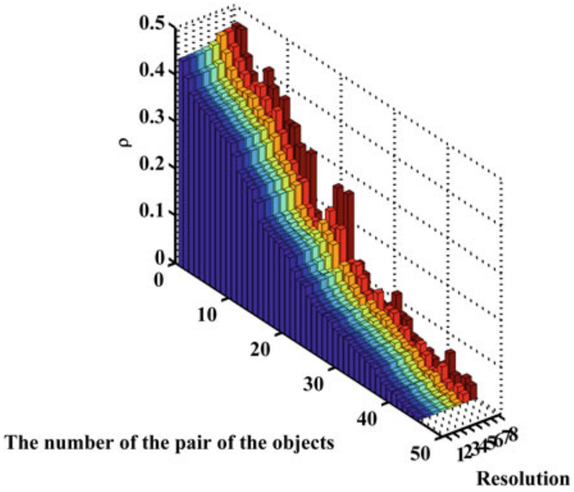


Fig. 10 The graphs of the invariant function J_8 for ten objects and eight resolutions

Therefore the next step of examining the influence of the resolution is to compare the distances between the pattern vectors for the various objects. We expect that the distance will be big enough to recognise the objects correctly. We assume that the pattern vector for the object number k is denoted as $\bar{J}_k = [J_1, \ldots, J_8]$. The distance (difference) between objects ith and jth is defined as $\rho_{ij} = ||J_i - J_j||$. We use the metric for computing the distance between the vectors. The values of these distances for changing resolution and for all combinations of the pairs of the recognised patterns are presented in Fig. 11.

Fig. 11 The graphs of the invariant function J_8 for ten objects and eight resolutions



According to Fig. 11, we can see that for too low resolution the distances between feature vectors for particular objects are becoming smaller. It seems that such a situation may have an adverse effect on the recognition (differentiation) of the objects.

4 Example

We will present as example a case of two objects moving in the terrain which we will have to locate. Figures with the highest resolution have the following dimensions: the image in which we will recognise objects has dimensions 3840 × 2160 pixels whereas object pattern images have dimensions (625 × 325) pixels (Object 1) and (587 × 375) pixels (Object 2). We will examine the cases for

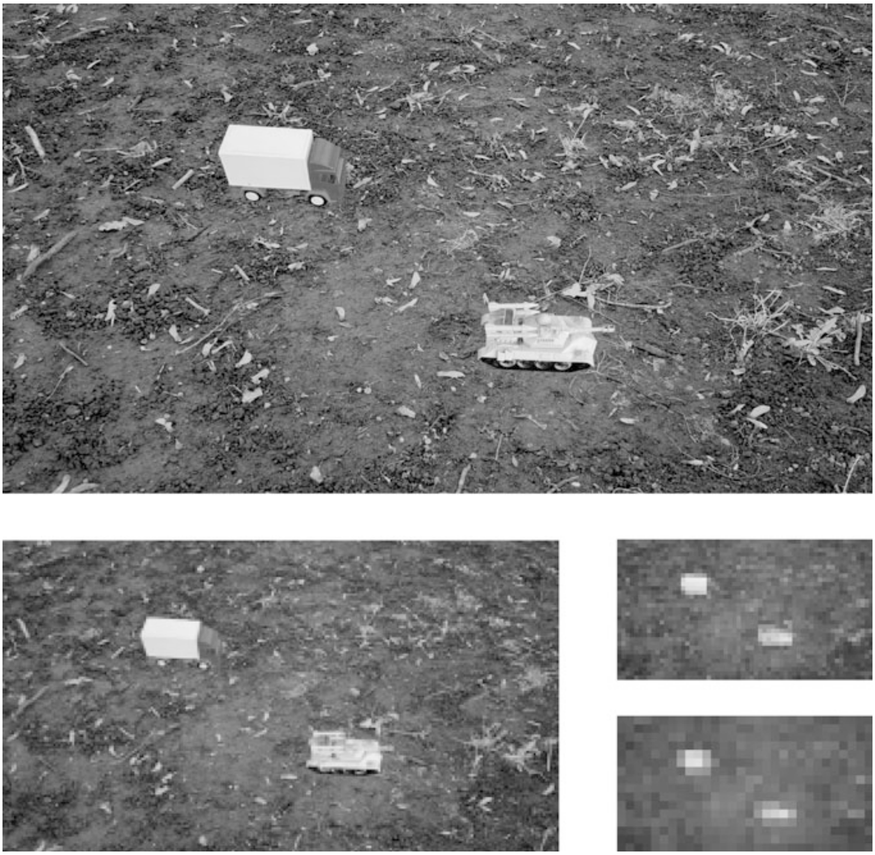


Fig. 12 Four scales of the image in which we recognise the objects

four values of resolutions: (a) original image—not downscaled, (b) image linearly downscaled in each axis 16 times, (c) image linearly downscaled in each axis 64 times and (d) image linearly downscaled in each axis 128 times.

Figure 12 presents the image in which we will recognise the objects for four examined scales.

Figure 13 presents four scales of pattern images of the recognised objects.

Figure 14 presents results of the objects recognition for the original image. The particular colours denote isolines corresponding to the minimum distance to the pattern of a given object.

Figure 15 presents results of the objects recognition for the image linearly downscaled in each axis with the scale factor equal 1/16.

Figure 16 presents results of the objects recognition for the image linearly downscaled in each axis with the scale factor equal 1/64.

Figure 17 presents results of the objects recognition for the image linearly downscaled in each axis with the scale factor equal 1/128.

We can see that for the cases presented in Figs. 14, 15 and 16 the results of recognition are correct and coincide with each other whereas in the case presented

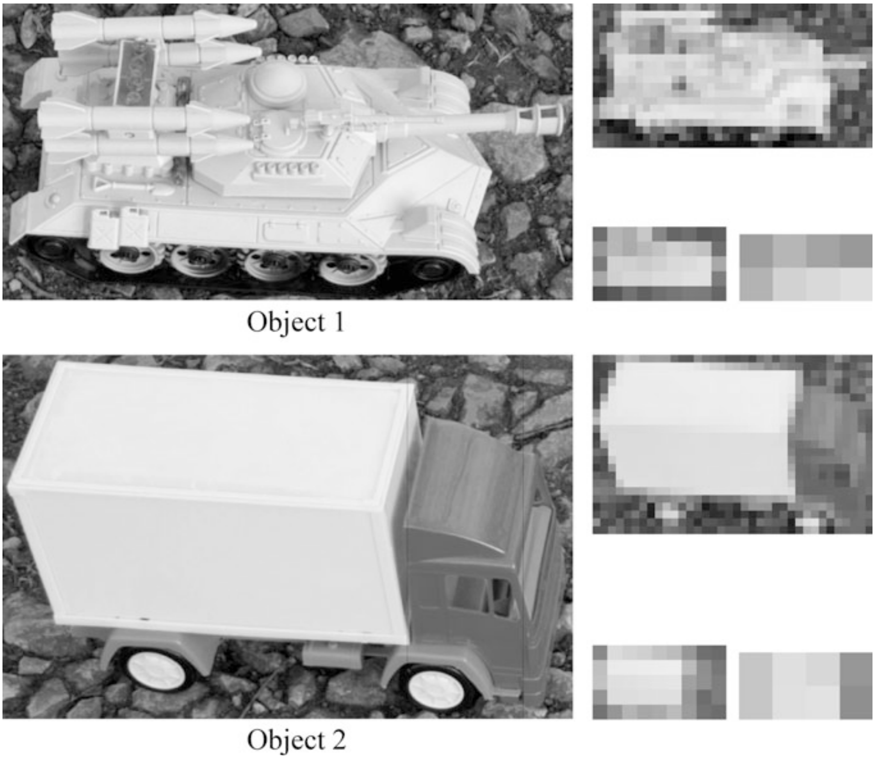


Fig. 13 Four scales of pattern images of the recognised objects

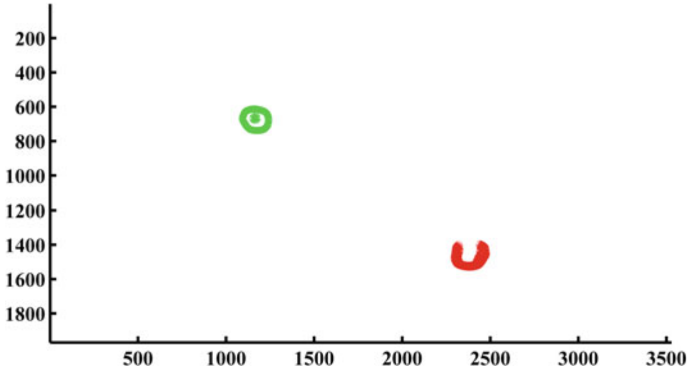


Fig. 14 The places of object 2 (*green*) and object 1 (*red*) recognition for the original image—not downscaled (Color figure online)

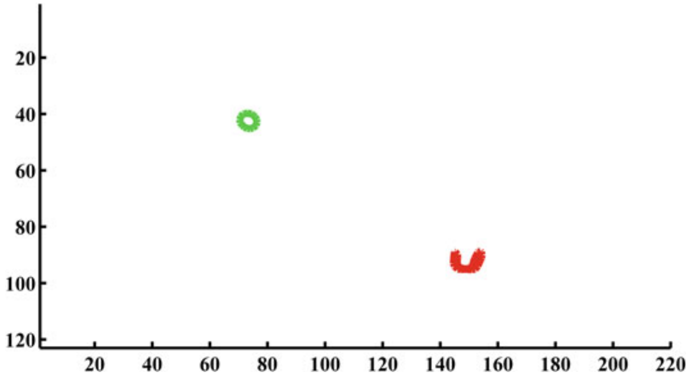


Fig. 15 The places of object 2 (*green*) and object 1 (*red*) recognition for the image linearly downscaled in each axis with the scale factor equal 1/16 (Color figure online)

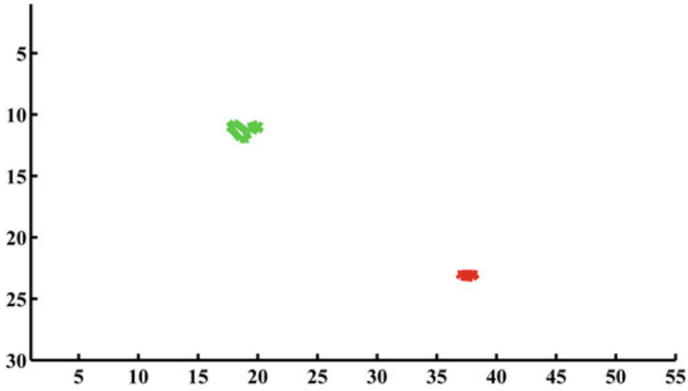


Fig. 16 The places of object 2 (*green*) and object 1 (*red*) recognition for the image linearly downscaled in each axis with the scale factor equal 1/64 (Color figure online)

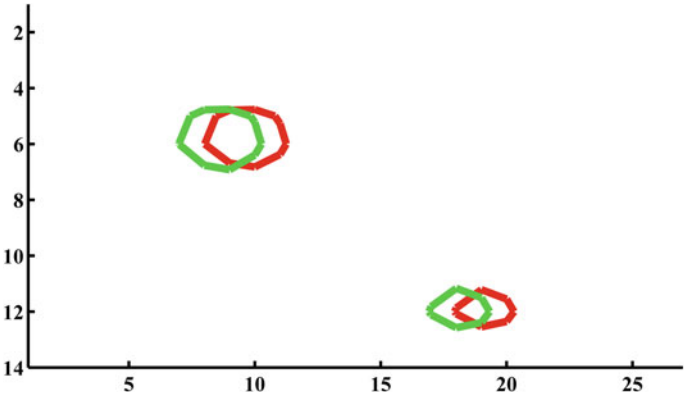


Fig. 17 The places of object 2 (*green*) and object 1 (*red*) recognition for the image linearly downscaled in each axis with the scale factor equal 1/128 (Color figure online)

in Fig. 17 the places of the objects' location were computed correctly; however, we cannot state which object is in which place.

The presented examples illustrate that we can obtain short time of image processing thanks to downscaling but there is some limit of the scale. When we exceed this limit the speed is achieved at the cost of accuracy. In this case it is not possible to recognise correctly.

5 Conclusions

The following study examined the influence of image resolution in the pattern recognition for the grey scale images. In order to recognise the pattern, the authors used the method based on moment invariants which were the elements of the feature vectors.

The examples of the terrain images, where the tracked objects can move [21–23], and the examples of various vehicles images were used as the pattern images for which the pattern vectors were calculated [24–27].

In order to examine the influence of image resolution, the values of moment invariants for various objects and various image scales were presented herein. The obtained values of moment invariants and distances between feature vectors allowed to conclude that the problem of image resolution have to be taken into consideration as the previous stage of pattern recognition.

The authors presented results which showed that for a significant decrease in resolution the problems in pattern recognition may occur. It results from the influence of image resolution on the values of moment invariants and, at the same time, on the value of the distance between the feature vectors defining the recognised objects [28, 29].

There were presented examples which indicate that in order to recognise objects correctly, it is necessary to retain some necessary minimum resolution [30]. The short time of numerical calculations can be achieved owing to the downscaling of the image, however, this operation can results in recognising an object incorrectly.

To conclude, this paper shows that a simple comparison of the values of the moment invariants for various image resolutions allows to assess what is the lowest acceptable value of an image size. Examples presented in this study illustrate the proposed solution of this problem.

References

1. Bibik, P., Gradolewski, S., Zawislak, W., Zbudniewek, J., Darakchiev, R., Krczel, J., Michalski, M., Strzelczyk, K.: Problems of detecting unauthorized satellite transmissions from the VSAT terminals. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–4. ISBN: 978-1-4673-1422-0, 8–9 Oct 2012
2. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guid. Control Dyn. **35**, 1354–1362 (2012)
3. Bibik, P., Narkiewicz, J.: Helicopter modeling and optimal control in autorotation. Ann. Proc. Am. Helicopter Soc. **64**(2), 986 (2008)
4. Bieda, R., Grygiel, R.: Wyznaczanie Orientacji Obiektu w Przestrzeni z Wykorzystaniem Naiwnego Filtru Kalmana. Przeglad Elektrotechniczny **90**, 34–41 (2014)
5. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. Appl. Mech. Mater. **249**, 1234–1246 (2013)
6. Babiarz, A., Bieda, R., Jaskot, K.: Vision system for group of mobile robots, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 139–156 (2013). ISBN: 978-3-319-00368-9
7. Ryt, A., Sobel, D., Kwiatkowski, J., Domzal, M., Jedrasiak, K., Nawrat, A.: Real-time laser point tracking. Comput. Vision Graph. **8671**, 542–551 (2014). (Lecture Notes in Computer Science)
8. Nawrat, A., Jedrasiak, K.: Fast colour recognition algorithm for robotics, Problemy Eksploatacji, pp. 69–76 (2008)
9. D. Davies, P.L. Palmer, M. Mirmehdi: Detection and tracking of very small low contrast objects. In: Proceedings of the 9th British Machine Vision Conference (1998)
10. Zhang, S., Karim, M.A.: Automatic target tracking for video annotation. Op. Eng. **43**, 1867–1873 (2004)
11. Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP: Graph Models Image Process. **53**, 231–239 (1991)
12. Chesnaud, C., Refegier, P., Boulet, V.: Statistical region snake-based segmentation adapted to different physical noise models. IEEE Trans. Patt. Anal. Mach. Intell. **21**, 1145–1157 (1999)
13. Gordon, N., Ristic, B., Arulampalam, S.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)
14. Sharp, C., Shakernia, O., Sastry, S.: A vision system for landing an unmanned aerial vehicle. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1720–1727. IEEE, Los Alamitos (2001)
15. Casbeer, D., Li, S., Beard, R., Mehra, R., McLain, T.: Forest Fire Monitoring With Multiple Small UAVs, Porland, OR (2005)
16. Papoulis, A.: Probability, Random Variables, and Stochastic Processes, 3rd edn. McGraw-Hill, New York (1991)

17. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. Thompson, Boston (2008)
18. Danahy, E.E., Agaian, S.S., Panetta, K.A.: Algorithms for the resizing of binary and grayscale images using a logical transform. In: Proceedings SPIE 6497, Image Processing: Algorithms and Systems V, 64970Z (February 27, 2007). doi:10.1117/12.704477 (2007)
19. Xiao, J., Zou, X., Liu, Z., Guo, X.: Adaptive interpolation algorithm for real-time image resizing. In: Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06), IEEE (2006)
20. Suresh, C., Singh, S., Saini, R., Saini, A.K.: I.J. Image, Graphics and Signal Processing, **5**, 55–62. Published Online April 2013 in MECS (http://www.mecs-press.org/) A Comparative Analysis of Image Scaling Algorithms (2013)
21. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. Comput. Vision Graph. **7594**, 423–432 (2012). (Lecture Notes in Computer Science)
22. Jedrasiak, K., Bereska, D., Nawrat, A.: The prototype of gyro-stabilized UAV gimbal for day-night surveillance. Adv. Technol. Intell. Syst. Nat. Border Secur. **440**, 107–115 (2013). (Studies in Computational Intelligence)
23. Galuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych. Przeglad Elektrotechniczny **86**, 278–283 (2010)
24. Bereska, D., Daniec, K., Fras, S., Jedrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 117–189 (2013). ISBN: 978-3-319-00368-9
25. Bereska, D., Daniec, K., Jedrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 115–121 (2013). ISBN: 978-3-319-00368-9
26. Sroka, M., Sciegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzalogowego pojazdu podwodnego—uklad stabilizacji i utrzymania zadanego kursu. Przeglad Elektrotechniczny **89**, 205–217 (2013)
27. Jaskot, K., Babiarz, A., Sroka, M., Sciegienka, P.: Prototyp bezzalogowego pojazdu podwodnego—konstrukcja mechaniczna, panel operatora. Przeglad Elektrotechniczny **89**, 52–67 (2013)
28. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Chen, S., (ed.) Proceedings of WSEAS international conference on mathematics and computers in science and engineering, no. 9. World Scientific and Engineering Academy and Society (2009)
29. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. Comput. Vis. Graph. **8671**, 302–315 (2014). (Lecture Notes in Computer Science)
30. Babiarz, A., Bieda, R., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 3–25 (2013). ISBN: 978-3-319-00368-9

The Method of Guaranteeing the Separation Between the Recognised Object and Background

Zygmunt Kuś and Aleksander Nawrat

Abstract The aim of the following study was to develop a procedure which guarantees the separation between the part of an image where we have the recognised object and the part of the image which corresponds to the terrain where the object moves. This research is conducted for grey scale images. The authors have presented the method which uses moment invariants for creating feature vectors which define the features of the recognised object and the features of the background. The presented method is based on calculating the distance between the values of invariant functions calculated for an object and the background. The distances were calculated for all moment invariants. These moment invariants were elements of the feature vector. In the next step the elements of the feature vector were ordered according to the values of these distances—from lowest to highest. Finally, the moment invariants, for which the distances were highest, were chosen as elements of a new—shorter feature vector. Furthermore, the algorithm of creating features vector was presented in the following paper. The developed algorithm allows to assess if a given invariant function is useful for the classification of the elements of a given set of classes. Owing to this approach, it was possible to choose properly the invariant functions which constitute the features vector. On the one hand, we can decrease the size of the features vector by choosing the invariant functions which separate particular classes in the best way. On the other hand, we know which function is the most proper to be added to the features vector when the size of the features vector is too small. On top of that, this study presents the example of recognising the object moving in some kind of a terrain.

Keywords Object recognition · Moment invariants · Pattern vector

Z. Kuś (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16 Street, Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_2

1 Introduction

Human visual perception provides a wealth of information allowing to function in a real environment. Therefore a computer's role is to provide the visual perception (electronic acquisition and understanding) so that devices are able to function in the same environment [1, 2].

However, computer visual perception is limited by the limited amount of information which can be processed by a computer. Thus, image analysis algorithms narrow down to taking into consideration only a local neighbourhood of the central point of the camera field of view (image of the small part of the environment). By the central point of the camera field of view we mean the point of the environment which is visible in the image centre. It is very difficult to interpret an image when we do not consider environment from outside of the image; worse, we only consider a part of the image which is in the aperture. In the literature of the subject [3–9] one can find a great number of solutions of a pattern recognition problem which are based on image processing methods. One of these solutions are methods based on region moment representations [10, 11].

2 Formulating the Problem of Distinguishing an Object and Background

The basic problem of this study is to recognise an object which is moving in a terrain in the case when we assume the involvement of an operator in the first stage of tracking. Therefore in the preliminary stage of tracking after the tracked object appears in the camera field of view [12–14], the operator marks on a screen the area of the image which constitutes the tracked object. Such an approach allows to conduct the examination of the tracked object characteristics (setting the vector of features); moreover, it allows to treat an unmarked area as the background.

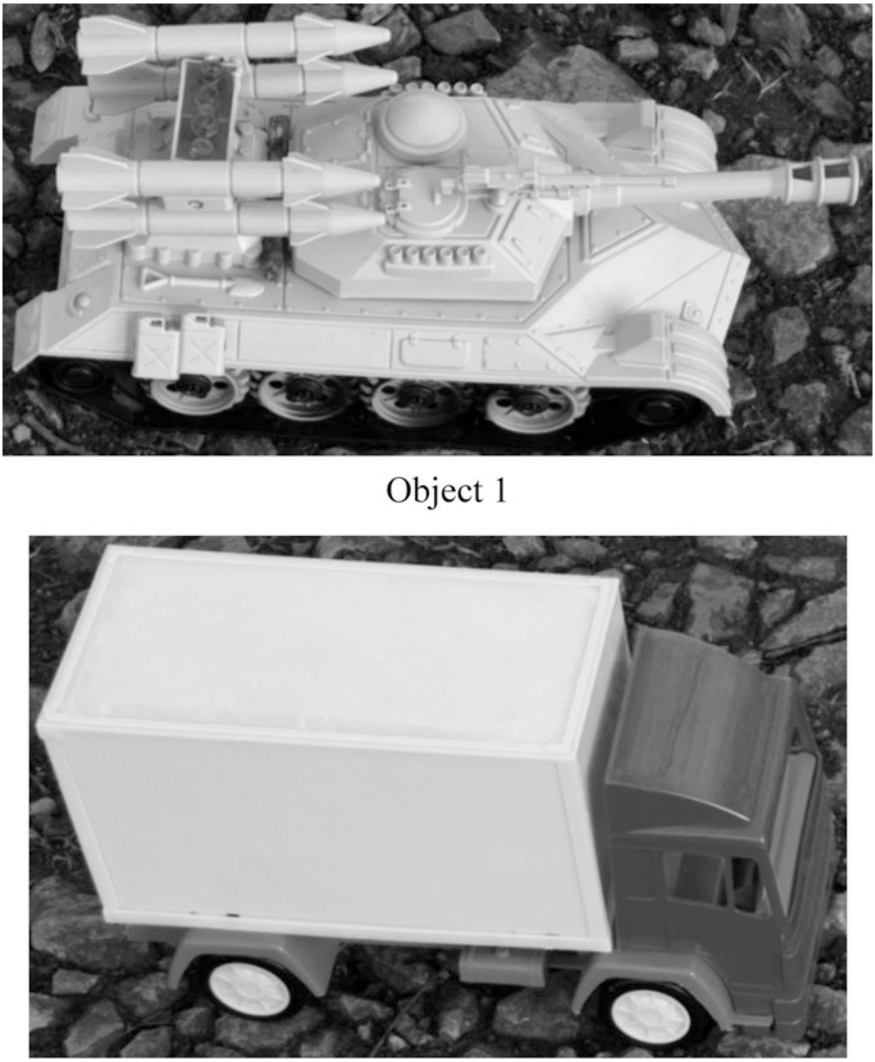
Undoubtedly, in a general case, the background—terrain where the object is moving might be inhomogeneous, therefore the analysis of the image could be hindered if we did not have any rationale—indication how to choose the aperture size. Thanks to the operator showing in a starting point the size and location of the object, it is possible to select the aperture size so that it covers the size of the object [15–19]. For such an aperture we will also find the characteristics of the background—terrain.

During object tracking and the appearance of a different background in the camera field of view, we can correct the background features vector on the basis of the marked object image and the new image of the terrain. One of the fundamental assumptions is the support of the image processing system by a rangefinder which allows to measure the distance between the camera and the tracked object in the starting point. Thanks to the knowledge of the focal length of the camera we can compute the size of an object [20–22].

In this way, during the processing of the images which were acquired at different distances between the camera and the observed part of the terrain, we can accordingly resize the aperture. It must be conducted in such a way that the area in the image covered by the aperture should be approximate to the area covered by an object (if the object was in the observed part of the terrain).

The examples of the objects, which we are going to recognise, are presented in Fig. 1.

The examples of the terrain images where the tracked objects can move are presented in Fig. 2.



Object 1

Object 2

Fig. 1 The examples of the objects

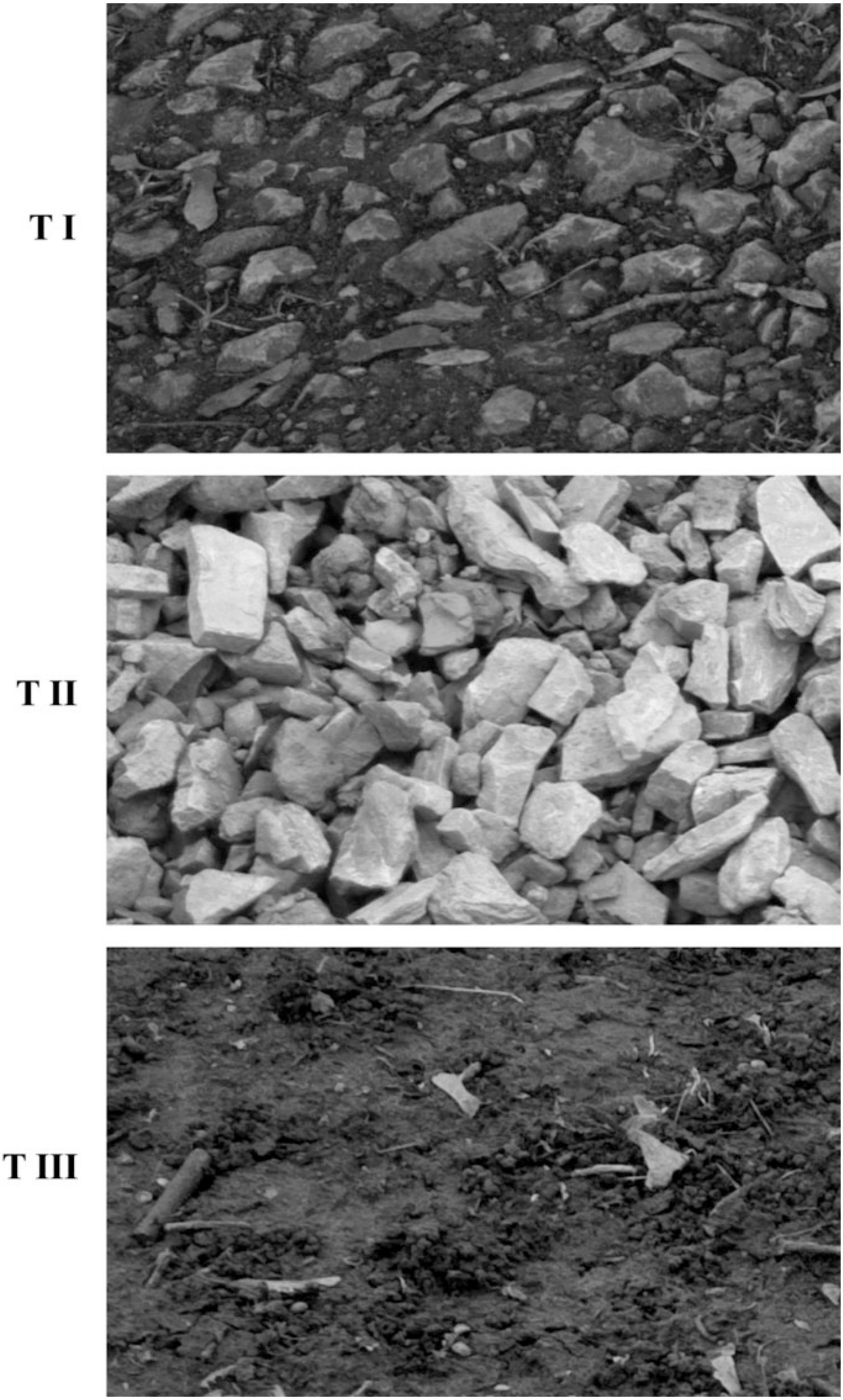


Fig. 2 The examples of the terrain images

3 Invariant Functions Used for a Description of an Object and Background Features

Examining the usefulness for recognising the objects from a given test set required that we chose invariant functions (1) (moment invariants) as it was presented in literature:

$$\begin{aligned} &J_1 = \frac{I_1}{m_{00}^2};\quad J_2 = \frac{I_2}{m_{00}^4};\quad J_3 = \frac{I_3}{m_{00}^5};\quad J_4 = \frac{I_4}{m_{00}^5};\\ &J_5 = \frac{I_5}{m_{00}^{10}};\quad J_6 = \frac{I_6}{m_{00}^7};\quad J_7 = \frac{I_7}{m_{00}^4};\quad J_8 = \frac{I_8}{m_{00}^5}; \end{aligned} \tag{1}$$

where

$$\begin{aligned} I_1 &= M_{20} + M_{02}\\ I_2 &= (M_{20} - M_{02})^2 + 4M_{11}^2\\ I_3 &= (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2\\ I_4 &= (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2\\ I_5 &= (M_{30} - 3M_{12})(M_{30} + M_{12})((M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2)\\ &\quad + (3M_{21} - M_{03})(M_{21} + M_{03})(3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2)\\ I_6 &= (M_{20} - M_{02})((M_{30} + M_{12})^2 - (M_{21} + M_{03})^2)\\ &\quad + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03})\\ I_7 &= M_{20}M_{02} - M_{11}^2\\ I_8 &= M_{30}M_{12} + M_{21}M_{03} - M_{12}^2 - M_{21}^2 \end{aligned} \tag{2}$$

Central moments of order $(p + q)$:

$$M_{pq} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (i - x_s)^p (j - y_s)^q f(i,j)$$

The co-ordinates of the region's centre of gravity (centroid):

$$x_s = \frac{m_{10}}{m_{00}};\quad y_s = \frac{m_{01}}{m_{00}} \tag{3}$$

Moments of order $(p\ +\ q)$:

$$m_{pq} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} i^p j^q f(i,j)$$

where $f(i, j)$—image pixel; M—the height of the image, N—the width of the image.

The abovementioned equations, usually formulated for black and white images, were herein modified in such a way that their values can be calculated for the grey scale images.

A great number of the abovementioned definitions of the IF were presented in [10]. In [11] the use of these IF in image recognition was discussed.

The abovementioned IFs should be invariant regarding rotation, resizing and the shift. It concerns transformations which are made in the image plane on the parts of the image which are the very object. In the case of 3D objects, which change their location in relation to the camera, we obtain different images for the same object [23–25]. In this case, it is necessary to treat each view of the object as a separate subclass and to create the class of the object consisting of the subclasses corresponding to different views of a given object. This paper, for the sake of clarity, will narrow down to one view for each object in the presented example.

4 The Method of Creating Features Vector Which Guarantees the Separation of an Object and Background

The aim of the following study is to develop a method which will enable to select the invariant function F_k (calculated for the part of the picture taken from the aperture). This method should guarantee the best distinction between two situations: (a) the object occurs in the aperture and (b) the background occurs in the aperture. It is one of the problems which appear in the pattern recognition task which is realised by the image processing system. We assume that the images are grey scale images. Furthermore, we assume that the size of the aperture is adjusted to the size of the object (pattern) which we are looking for. It also needs to be assumed that the picture of the terrain where the object is located and various pictures of the object are available. We will examine the invariant functions presented in Sect. 3.

This paper will aim at achieving the results which are adjusted to the particular object and particular background; therefore the research will be conducted for an exemplary terrain image and exemplary object pattern which we are looking for. F_k(aperture) is the value of the kth invariant function calculated for the part of the picture taken from the aperture. For each kth invariant function we define the difference of the function calculated for the background and object as:

$$\Phi_k(\text{background},\ \text{object}) = F_k(\text{background}) - F_k(\text{object}) \tag{4}$$

On the basis of this difference calculated for different exemplary object and background images, we are going to assess the usefulness of the kth invariant function in order to look for a given object which is located in the given background.

The algorithm

A. We take the photos of an object and select the set of images which will define a desirable pattern vector $\bar{P}_o = \{P_o(i)\}$ where $P_o(i)$ is the *i*th image of the object.

B. We take the photos of a terrain corresponding to different kinds of the backgrounds in which the recognised object can occur. Next we select the set of images which will define a backgrounds vector $\bar{P}_T = \{P_T(j)\}$ where $P_T(j)$ is the *j*th image of the background.

C. We create a set of all possible pairs $P_{oT} = \{P_o(i), P_T(j)\}$.

D. We calculate $\Phi_{kl}(i,j) = |F_k(P_T(j)) - F_k(P_o(i))|$ (in our case: k = 1, …, 8 and l = 1, …, 6) and create the set $\Phi_k = \{\Phi_{kl}(i,j)\}$ for each i and j.

E. For each invariant function (a vector of the invariant functions) F_k we order the elements of the set Φ_k from the lowest to the highest. We select half of the elements of the set—elements placed on the left of the median—lower than the median. Next we calculate the mean value Φ_{kmean} of the selected elements.

F. We compare the values Φ_{kmean} and select this function F_k, for which Φ_{kmean} is the highest. In order to make the comparison of different invariant functions possible.

5 Examples

We examine the images of the objects presented in Fig. 1 as a set of the images of the objects, whereas the set of the terrain images was shown in Fig. 2. In this way we obtain 6 pairs 'object type'—'terrain type' as presented in (5).

$$\begin{array}{llllll} 1. & \text{'ob1'—TI,} & 3. & \text{'ob1'—TII,} & 5. & \text{'ob1'—TIII,} \\ 2. & \text{'ob2'—TI,} & 4. & \text{'ob2'—TII} & 6. & \text{'ob2'—TIII} \end{array} \tag{5}$$

For each object and terrain type, we calculate the invariant functions J_1 to J_8. These values are shown in Table 1.

Table 2 illustrates the values of the differences modules of the IFs for all possible pairs 'object type'—'terrain type' for the subsequent invariant functions J_1 to J_8. Owing to this fact, we will be able to assess how much a given function J distinguishes an object from the background.

Table 1 The values of the invariant functions for the tested objects and terrain types

	J_1	J_2	J_3	J_4	yJ_5	J_6	J_7	J_8
Object 1	0.3150	0.0380	0.0002	0.0001	0.0000	0.0000	0.0153	0.0000
Object 2	0.2599	0.0128	0.0002	0.0005	0.0000	0.0001	0.0137	0.0000
Terrain I	0.6545	0.0906	0.0001	0.0001	0.0000	0.0000	0.0844	0.0000
Terrain II	0.3336	0.0233	0.0001	0.0001	0.0000	0.0000	0.0220	0.0000
Terrain III	0.6834	0.0933	0.0010	0.0002	0.0000	−0.0001	0.0934	−0.0001

Table 2 The values of the differences modules of the IFs for an object and terrain type

	|J(TI) − J(O1)|	|J(TI) − J(O2)|	|J(TII) − J(O1)|	|J(TII) − J(O2)|	|J(TIII) − J(O1)|	|J(TIII) − J(O1)|
J_1	0.3394	0.3945	0.0186	0.0737	0.3684	0.4235
J_2	0.0526	0.0778	0.0148	0.0105	0.0553	0.0805
J_3	0.0001	0.0000	0.0000	0.0000	0.0008	0.0008
J_4	0.0000	0.0004	0.0001	0.0004	0.0001	0.0003
J_5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
J_6	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001
J_7	0.0691	0.0707	0.0067	0.0083	0.0781	0.0797
J_8	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001

In the next step, as shown in Table 3, for each invariant function J we order the values of the function $\Phi(k, l)$ from the lowest to the highest.

Next, we will calculate the mean value for three lowest values of $\Phi(k, l)$. They are shown in Table 4.

According to Table 4, we are able to assess which invariant function J suits the best for distinguishing an object from the background. The best one is the one for which the mean of $\Phi(k, l)$ is the highest. In Table 4 the best functions J are in the

Table 3 The ordered values of the differences $\Phi(k, l)$ of the invariant functions

	|J(Tj) − J(Oi)|					
J_1	0.0186	0.0737	0.3394	0.3684	0.3945	0.4235
J_2	0.0105	0.0148	0.0526	0.0553	0.0778	0.0805
J_3	0.0000	0.0000	0.0000	0.0001	0.0008	0.0008
J_4	0.0000	0.0001	0.0001	0.0003	0.0004	0.0004
J_5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
J_6	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001
J_7	0.0067	0.0083	0.0691	0.0707	0.0781	0.0797
J_8	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001

Table 4 The mean value for three lowest values of $\Phi(k, l)$

	|J(Tj) − J(Oi)|			Mean value
J_4	0.0186	0.0737	0.3394	0.1439
J_7	0.0067	0.0083	0.0691	0.028033
J_5	0.0105	0.0148	0.0526	0.025966
J_1	0.0000	0.0001	0.0001	0.00007
J_6	0.0000	0.0000	0.0000	0.0000
J_3	0.0000	0.0000	0.0000	0.0000
J_2	0.0000	0.0000	0.0000	0.0000
J_8	0.0000	0.0000	0.0000	0.0000

top half of the table. If we create the features vector using a few functions J, it will guarantee a better separation of an object from the background. After choosing four functions J_4, J_5, J_7 and J_1, we will conduct the recognition of a given object on a given background.

Figures 3, 4 and 5 presents the selected objects on the backgrounds TI, TII and TIII.

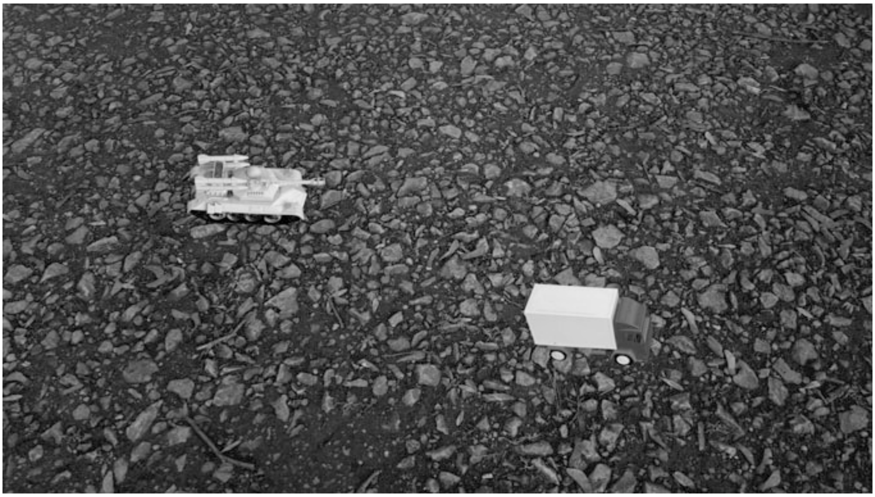


Fig. 3 The example of the objects ‘1’ and ‘2’ on the background I

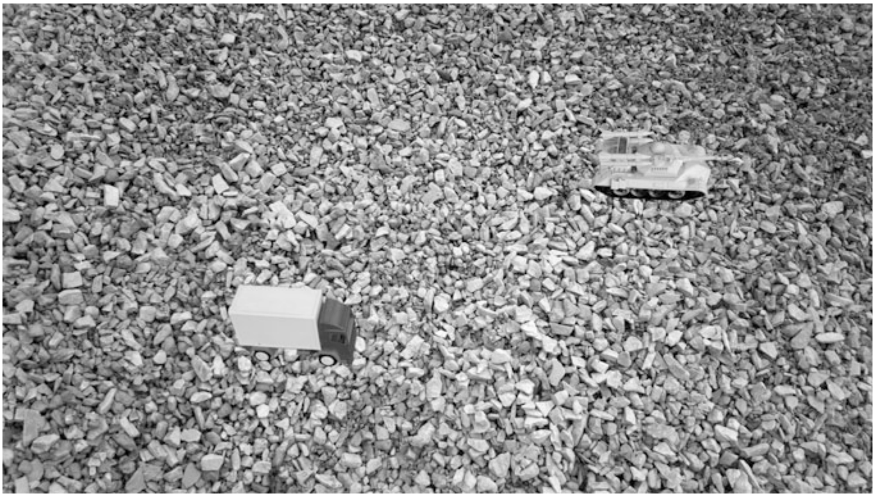


Fig. 4 The example of the objects ‘1’ and ‘2’ on the background II

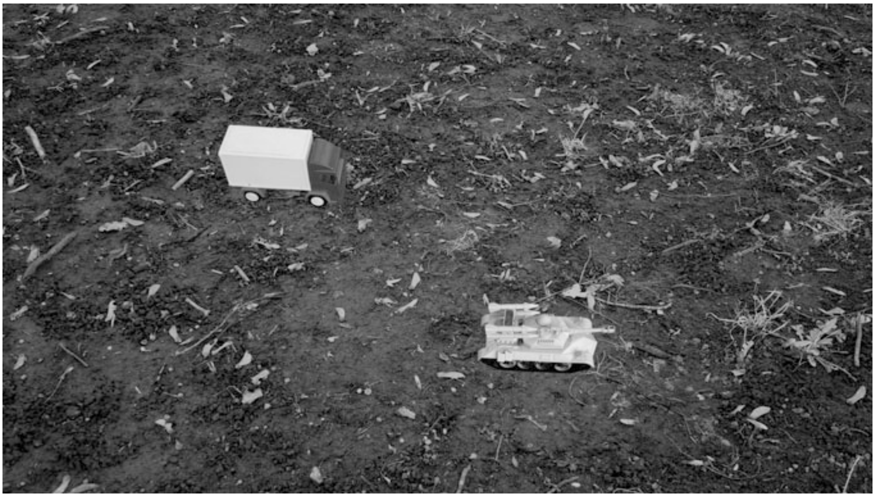


Fig. 5 The example of the objects ‘1’ and ‘2’ on the background III

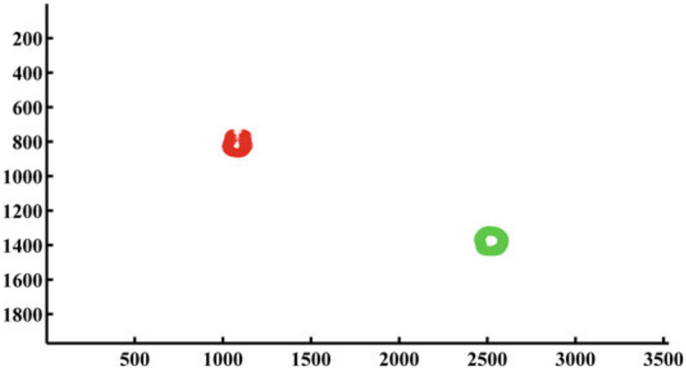


Fig. 6 The location of the object ‘2’ (*green*) and object ‘1’ (*red*) in the Fig. 3 (Color figure online)

Having the values of the pattern vector of the invariant functions [J_4, J_5, J_7, J_1] both for an object and the background, we can carry out the recognition of the object on the background by assigning each point of the image to the background or the object. In order to assess the distance between the pattern vectors and current vectors for the object and background, we will use Euclidian metric. Figures 6, 7 and 8 presents the calculated location of the recognised object.

The areas, which were marked by appropriate colours, correspond to the minimal values of the distances between the model pattern vector for a given object and the pattern vector for the current location of the aperture. When we compare Figs. 3, 4 and 5, we can observe that the location of the object was correctly determined.

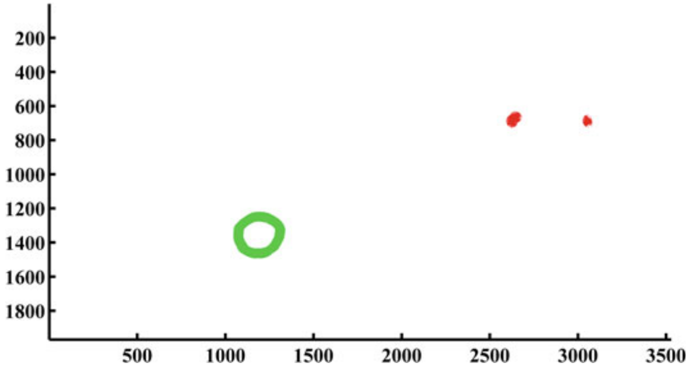


Fig. 7 The location of the object ‘2’ (*green*) and object ‘1’ (*red*) in the Fig. 4 (Color figure online)

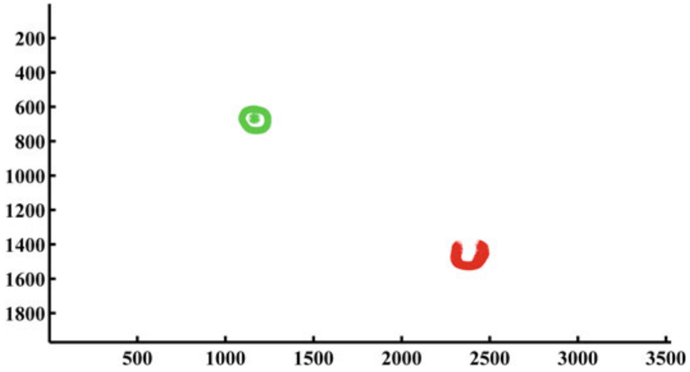


Fig. 8 The location of the object ‘1’ (*green*) and object ‘2’ (*red*) in the Fig. 5 (Color figure online)

6 Conclusions

This paper has presented an algorithm of creating the invariant functions vector which guarantees the best separation of the given object classes. In this very case, these classes consisted of different objects and different terrain types where these objects were moving. This algorithm allows to assess the usefulness different invariant functions used for classifying the given set of classes.

At the same time, owing to the ordering of the invariant functions from ‘the best’ to ‘the worst’, we can when necessary increase the size of the features vector if this size is too small to classify properly the elements belonging to the particular classes. The presented example illustrates that the proposed method works properly.

References

1. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guid. Control Dyn. **35**, 1354–1362 (2012)
2. Bibik, P., Narkiewicz, J.: Helicopter modeling and optimal control in autorotation. In: Annual Proceedings—American Helicopter Society, vol. 64, no. 2, pp. 986 (2008)
3. Davies, D., Palmer, P.L., Mirmehdi, M.: Detection and tracking of very small low contrast objects. In: Proceedings of the 9th British Machine Vision Conference, Sept 1998
4. Zhang, S., Karim, M.A.: Automatic target tracking for video annotation. Op. Eng. **43**, 1867–1873 (2004)
5. Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP Graph Models Image Process. **53**, 231–239 (1991)
6. Chesnaud, C., Refegier, P., Boulet, V.: Statistical region snake-based segmentation adapted to different physical noise models. IEEE Trans. Patt. Anal. Mach. Intell. **21**, 1145–1157 (1999)
7. Gordon, N., Ristic, B., Arulampalam, S.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)
8. Sharp, C., Shakernia, O., Sastry, S.: A vision system for landing an unmanned aerial vehicle. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1720–1727. IEEE, Los Alamitos (2001)
9. Casbeer, D., Li, S., Beard, R., Mehra, R., McLain, T.: Forest fire monitoring with multiple small UAVs, Portland, OR, Apr 2005
10. Papoulis, A.: Probability, Random Variables, and Stochastic Processes, 3rd edn. McGraw-Hill, New York (1991)
11. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. Thompson, Stamford (2008)
12. Babiarz, A., Bieda, R., Jaskot, K.: Vision system for group of mobile robots. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 139–156 (2013). ISBN: 978-3-319-00368-9
13. Ryt, A., Sobel, D., Kwiatkowski, J., Domzal, M., Jedrasiak, K., Nawrat, A.: Real-time laser point tracking. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 8671. pp. 542–551 (2014)
14. Nawrat, A., Jedrasiak, K.: Fast colour recognition algorithm for robotics. In: Problemy Eksploatacji, pp. 69–76 (2008)
15. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 423–432 (2012)
16. Bieda, R., Grygiel, R.: Wyznaczanie Orientacji Obiektu w Przestrzeni z Wykorzystaniem Naiwnego Filtru Kalmana. Przeglad Elektrotechniczny **90**, 34–41 (2014)
17. Galuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych. Przeglad Elektrotechniczny **86**, 278–283 (2010)
18. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. Appl. Mech. Mater. **249**, 1234–1246 (2013)
19. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorbtion during the progressive fracture process. WIT Trans. Model. Simul. **51**, 625–633 (2011)
20. Bereska, D., Daniec, K., Fras, S., Jedrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera. Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 117–189 (2013). ISBN: 978-3-319-00368-9, 2013

21. Sroka, M., Sciegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzalogowego pojazdu podwodnego - uklad stabilizacji i utrzymania zadanego kursu. Przeglad Elektrotechniczny **89**, 205–217 (2013)
22. Jaskot, K., Babiarz, A., Sroka, M., Sciegienka, P.: Prototyp bezzalogowego pojazdu podwodnego - konstrukcja mechaniczna, panel operatora. Przeglad Elektrotechniczny **89**, 52–67 (2013)
23. Babiarz, A., Bieda, R., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 3–25 (2013). ISBN: 978-3-319-00368-9
24. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 407–413 (2012)
25. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)

The Method of Developing the Invariant Functions Vector for Objects Recognition from a Given Objects Set

Zygmunt Kuś and Aleksander Nawrat

Abstract The goal of the following paper was to develop the methodology of recognising a given object out of other objects in the grey scale images. We have presented the method which utilises moment invariants for creating model vectors which define the features of the recognised object. Moreover, we have discussed the rules of creating the model vectors which guarantee differentiation of the given object classes. We have put forward the method of recognising an object in the image. This method is based on searching the points in the image for which there is minimal distance between the model vector and current vector—calculated for each point of the image. On top of that, this study presents the examples of the object recognising by means of the developed method.

Keywords Object recognition · Moment invariants · Pattern vector · Minimum distance principle

1 Introduction

Images constitute the basic source of information about the environment for a human being, however, in many cases this information is surplus. At the same time, the ambiguity of information contained in the image prevents from easy simplification—the reduction of surplus information. There are particular problems, especially when a computer makes decisions without the human being's participation on the basis of the images—wrong decisions cannot be corrected by an operator of the vision system.

Z. Kuś (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Akademicka 16 Street, Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_3

The whole problem is more complicated when the image processing system makes decisions in real time system—in the interaction with the environment on which we do not have an influence. It results in the necessity of carrying out an analysis of the image and making a decision in a limited time—this time is usually too short for the image processing system [1, 2].

The fundamental stage of the image processing is the improvement of the image quality. Our goal is not to increase the amount of information in the image, but to emphasise the features essential from a particular point of view for a given application. Moreover, we aim at removing from the image some elements which are irrelevant to a given application [3, 4]. In this scope there are among others such operations as noise reduction, removing of the background, edges sharpening and contouring.

There are many solutions of a pattern recognition problem which are based on image processing methods [5–11]. One of these solutions are methods based on region moment representations [12, 13].

We have to deal with this situation in the case of the application of object recognition to object tracking [14–18]. One of the methods, which allows to reduce the amount of information, is to use grey scale image instead of colour image. In many cases, such an image is thresholded in order to obtain a black and white image. Such an approach, which allows to reduce the amount of information to a greater extent, is at the same time used to provide segmentation—distinguishing a recognised object in the image. It is possible when the object's and background's intensities are significantly different. This difference have to be known or possible to find a priori. The authors have focused on the analysis of grey scale images using moment invariants for recognising what kind of object appears in the image.

2 Formulating the Problem of the Recognition for Grey Scale Images

In order to recognise an object in the image, one can usually use not only one Invariant Function (IF), but a certain set of IFs—the vector of IFs. The selection of the set of IFs is proper when it guarantees meeting the following conditions:

- for a given object the IF should have constant value regardless of the properties of the particular image of this object (lighting—diffused or directional, rotation, shift or the change of the scale);
- for various objects the IF should have significantly different values.

Selecting one function, which would have such properties, is usually impossible, therefore one can use a set of IFs ordered in a vector. The amount of these IFs determines a size of a vector and a size of a features space in which the classification of the objects takes place.

The main goal of the paper will be to develop the method—algorithm which allows to choose IFs in such a way that these IFs should create the features space adjusted to a given set of objects which are to be recognised.

The following paper assumes that the set of objects which are going to be recognised consists of four elements. Figure 1 presents these elements.

For each of these objects we took a set photographs in various lighting, various object orientation and various angles between the camera axis and the vertical direction. The diffused lighting gives an image without shadows whereas the directional lighting results in an image with shadows. The orientation of the objects was changed by turning manually each object which resulted in appearing various perspective views of the object in the images. In a similar way, we obtained the changes of the perspective view of the objects by changing the camera position in the vertical axis. It corresponds to the situation when the objects moving on the ground are watched by the camera mounted on the UAV [19–21].

Figure 2 shows the examples of various orientation, lighting and the angle between the camera direction and the vertical axis.

In order to examine the usefulness for recognising the objects from a given test set, we selected the following (1)–(8) IFs (moment invariants) discussed in the literature:

$$J_1 = \frac{I_1}{m_{00}^2} \tag{1}$$

$$J_2 = \frac{I_2}{m_{00}^4} \tag{2}$$

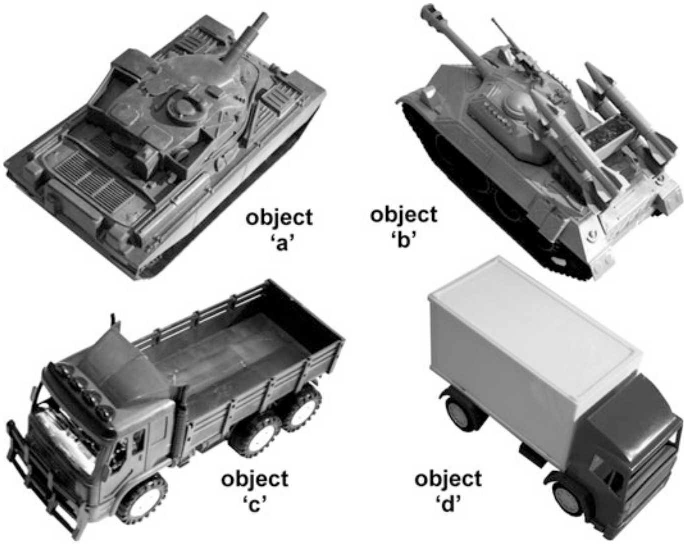
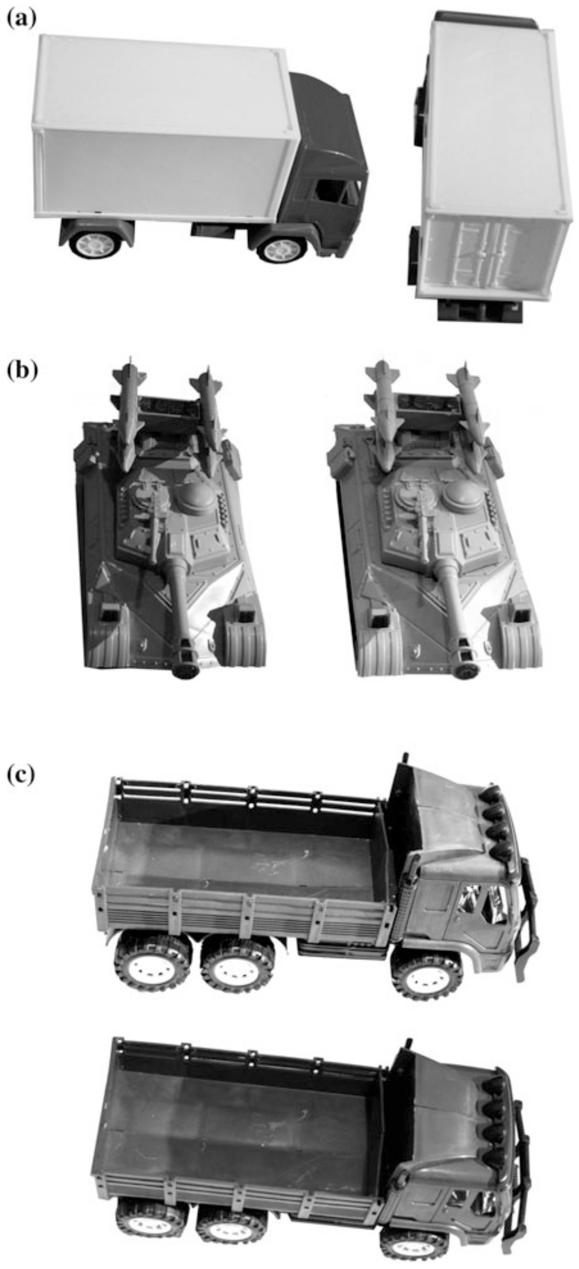


Fig. 1 The set of four objects which are going to be recognised

Fig. 2 The examples of the various orientation (**a**), lighting (directional, diffused) (**b**), and the angle between the camera direction and the vertical axis (**c**)



$$J_3 = \frac{I_3}{m_{00}^5} \tag{3}$$

$$J_4 = \frac{I_4}{m_{00}^5} \tag{4}$$

$$J_5 = \frac{I_5}{m_{00}^{10}} \tag{5}$$

$$J_6 = \frac{I_6}{m_{00}^7} \tag{6}$$

$$J_7 = \frac{I_7}{m_{00}^4} \tag{7}$$

$$J_8 = \frac{I_8}{m_{00}^5} \tag{8}$$

where:

$$I_1 = M_{20} + M_{02} \tag{9}$$

$$I_2 = (M_{20} - M_{02})^2 + 4M_{11}^2 \tag{10}$$

$$I_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \tag{11}$$

$$I_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \tag{12}$$

$$\begin{aligned} I_5 = {} & (M_{30} - 3M_{12})(M_{30} + M_{12})((M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2) \\ & + (3M_{21} - M_{03})(M_{21} + M_{03})(3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2); \end{aligned} \tag{13}$$

$$\begin{aligned} I_6 = {} & (M_{20} - M_{02})((M_{30} + M_{12})^2 - (M_{21} + M_{03})^2) \\ & + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03}); \end{aligned} \tag{14}$$

$$I_7 = M_{20}M_{02} - M_{11}^2 \tag{15}$$

$$I_8 = M_{30}M_{12} + M_{21}M_{03} - M_{12}^2 - M_{21}^2 \tag{16}$$

Central moments of order ($p + q$):

$$M_{pq} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (i - x_s)^p (j - y_s)^q f(i,j) \tag{17}$$

The coordinates of the region's centre of gravity (centroid):

$$x_s = \frac{m_{10}}{m_{00}}; \quad y_s = \frac{m_{01}}{m_{00}} \tag{18}$$

Moments of order (p + q):

$$\text{Moments of order } (p+q): \\ m_{pq} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} i^p j^q f(i,j) \tag{19}$$

where $f(i, j)$—image pixel; M—the height of the image; N—the width of the image.

The above mentioned IFs were modified in such a way that their values can be calculated for the grey scale images. There were used approximately 30 images for each object.

The majority of the abovementioned definitions of the IF were presented in [12]. In [13] the use of these IF in image recognition was discussed.

Each of the examined four objects was photographed in positions corresponding eight rotations of the object on the ground. Figure 3 presents the denotation of these rotation direction. We need to emphasise that the rotations were done in the ground plane on which there were the objects, but not in the image plane. It poses yet another problem, namely, we obtain different images (not the same images rotated by a certain angle) for different rotation angles on the ground plane. It results from perspective distortions.

Figures 4, 5, 6 and 7 present the graphs of the eight IFs. Each figure presents graphs for one out of the four tested objects. On the vertical axes there are the values of the functions J, whereas on the horizontal axes we have the numbers of the consecutive images obtained for the objects in the successive rotations.

According to the Figs. 4, 5, 6 and 7, part of the functions J is correlated with each other. At the same time, we can observe that some functions J change when the object rotation angle changes.

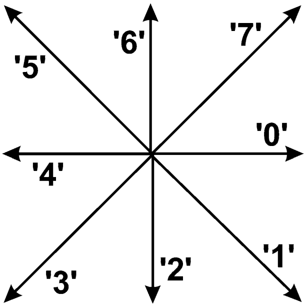


Fig. 3 The rotation directions of the spatial object on the ground plane

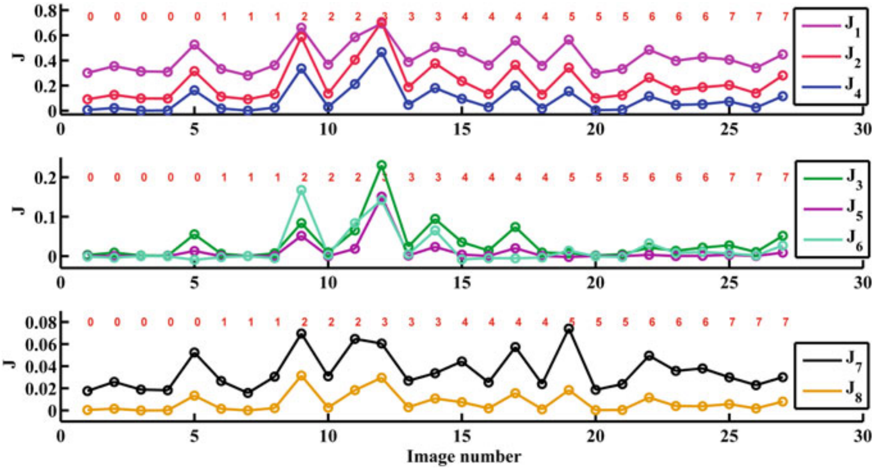


Fig. 4 The graphs of the invariant functions J_1–J_8 for the object ‘a’

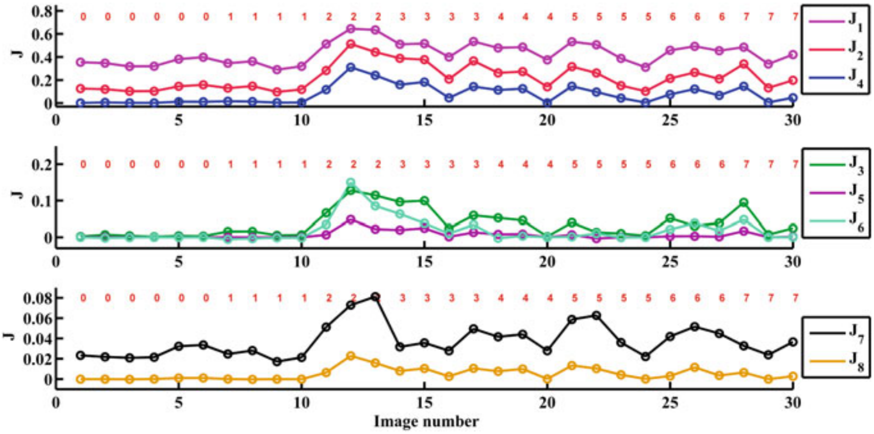


Fig. 5 The graphs of the invariant functions J_1–J_8 for the object ‘b’

3 Creating Model Vectors of the Features and Assessing the Distance Between a Current Vector and Model Vector of the Features

When comparing the images and the values of the functions J, we can see that the image recognition problem for the rotated 3D object is far more complex than the same problem for flat objects rotated in the image plane.

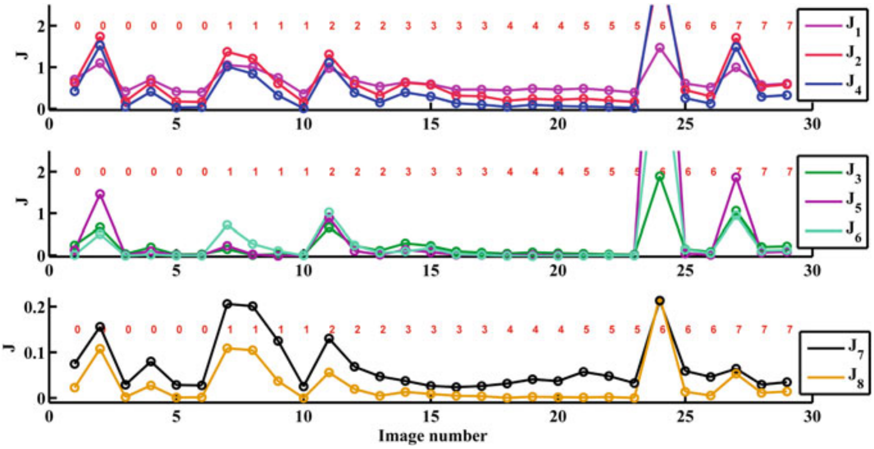


Fig. 6 The graphs of the invariant functions J_1–J_8 for the object 'c'

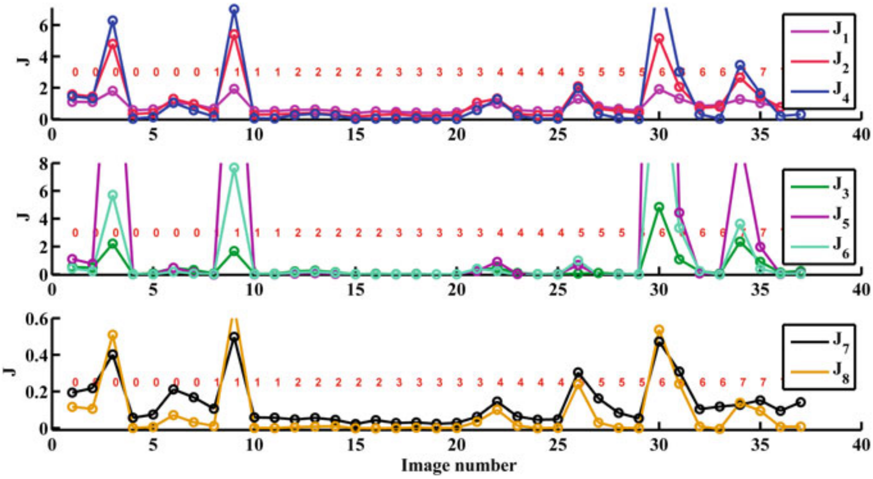


Fig. 7 The graphs of the invariant functions J_1–J_8 for the object 'd'

Therefore the images of the same test object for different rotations have to be treated as different objects for which we need to define the vector of the IFs. This on the hand results in the fact that an analysis must be conducted in the following steps. Firstly, we compare the IFs vector for the image of the recognised object with 32 model vectors (the subset for each of the 4 objects consists of the images for 8 rotation angles). Secondly, we choose the vector which is closest to the vector corresponding to the recognised image from 32 model vectors of the IFs. Finally, we test to which subset the chosen vector belongs—it defines the object number recognised in the image. The standard deviation will be used in order to assess whether IFs obtained for the exemplary images of a given object are really invariant.

We build the IFs vector by choosing from all IFs these IF for which the distribution of values is low.

By conducting this procedure for each IF and each image set, we obtain the values of the IFs characteristic for each object (for each object rotation angle) cf. Tables 1, 2, 3 and 4.

In Tables 1, 2, 3 and 4 we have only those functions which had the low value of distribution even for one object. The IFs for the model vector will be chosen on the basis of the Tables 1, 2, 3, and 4. This vector should allow us to distinguish 32 classes of the images (8 for each object). The selected IFs have to assume different values for at least 2 classes. It means that in the vertical column there have to be at least 2 different values.

The chosen functions create a model vector. For each object the vector has different values at particular positions. Thus, we obtain 32 model vectors for 4 objects and 8 rotation angles, cf. Table 5. In Table 5 we have only focused on the objects which are rotated at an angle denoted ‘0’ in order to limit the amount of recognised object classes and to increase the clarity of the results.

The task of assigning an unknown object to one of the classes is conducted in the following steps: (a) we calculate the values of IFs for the examined image in such a

Table 1 The values of the IFs characteristic for the object ‘a’ in particular rotations

Rotation angle	0	1	2	3	4	5	6	7
J_1	0.3543	0.3318	0.5853	0.5057	0.4681	0.3314	0.4253	0.4058
J_2	0.1257	0.1123	0.4051	0.3759	0.2354	0.1231	0.1866	0.2035
J_3	0.0013	0.0003	0.0651	0.0944	0.0354	0.0015	0.0215	0.0273
J_4	0.0209	0.0171	0.2115	0.1801	0.0945	0.0082	0.0513	0.0726
J_5	0.0000	0.0000	0.0187	0.0234	0.0037	0.0000	0.0008	0.0032
J_6	0.0001	0.0000	0.0835	0.0650	−0.0081	−0.0006	0.0105	0.0068
J_7	0.0258	0.0265	0.0647	0.0336	0.0443	0.0237	0.0357	0.0301
J_8	0.0015	0.0014	0.0183	0.0107	0.0074	0.0004	0.0040	0.0080

Table 2 The values of the IFs characteristic for the object ‘b’ in particular rotations

Rotation angle	0	1	2	3	4	5	6	7
J_1	0.3467	0.3457	0.6344	0.5099	0.4852	0.4500	0.4577	0.4205
J_2	0.1204	0.1300	0.4425	0.3885	0.2730	0.2000	0.2135	0.1978
J_3	0.0070	0.0054	0.1145	0.0850	0.0458	0.0102	0.0300	0.0245
J_4	0.0060	0.0159	0.2409	0.1601	0.1251	0.0700	0.0756	0.0465
J_5	0.0000	0.0000	0.0220	0.0191	0.0086	0.0000	0.0030	0.0015
J_6	−0.0019	−0.0013	0.0853	0.0450	0.0033	−0.0008	0.0382	0.0003
J_7	0.0215	0.0247	0.0728	0.0356	0.0416	0.0450	0.0449	0.0327
J_8	0.0000	0.0000	0.0158	0.0104	0.0076	0.0070	0.0035	0.0027

Table 3 The values of the IFs characteristic for the object ‘c’ in particular rotations

Rotation angle	0	1	2	3	4	5	6	7
J_1	0.7050	1.3750	0.6794	0.5907	0.4827	0.4425	0.6066	0.6039
J_2	0.6294	0.9000	0.5925	0.5754	0.2412	0.1999	0.4518	0.5853
J_3	0.1911	0.0070	0.2313	0.0933	0.0723	0.0413	0.1474	0.1991
J_4	0.4100	0.5800	0.3867	0.2955	0.0937	0.0402	0.2554	0.3270
J_5	0.1037	0.0227	0.1063	0.0144	0.0061	−0.0022	0.0435	0.0689
J_6	0.0165	0.2735	0.2410	0.0239	−0.0202	−0.0075	0.1357	0.1075
J_7	0.0745	0.1248	0.0684	0.0263	0.0370	0.0325	0.0460	0.0346
J_8	0.0227	0.0369	0.0194	0.0087	0.0017	0.0001	0.0053	0.0141

Table 4 The values of the IFs characteristic for the object ‘d’ in particular rotations

Rotation angle	0	1	2	3	4	5	6	7
J_1	1.0606	0.5067	0.5158	0.3908	0.5684	0.6501	1.0000	0.9800
J_2	1.2899	0.2782	0.2904	0.2048	0.3325	0.5106	0.7078	1.0666
J_3	0.4685	0.0414	0.1501	0.0032	0.0992	0.0377	1.0806	0.9012
J_4	1.0337	0.0466	0.2305	0.0009	0.2029	0.0483	0.3026	0.7000
J_5	0.4740	−0.0016	0.0394	0.0000	0.0214	−0.0020	4.4321	1.9762
J_6	0.1897	0.0106	0.1103	0.0001	−0.0616	−0.0167	3.3372	0.3982
J_7	0.1680	0.4980	0.0448	0.0309	0.0632	0.0830	0.1033	0.1516
J_8	0.0321	0.6671	0.0101	0.0027	0.0130	0.0013	0.0089	0.0450

Table 5 The model vectors characterising 4 examined objects

	Pattern vector for ‘a’	Pattern vector for ‘b’	Pattern vector for ‘c’	Pattern vector for ‘d’
J_1	0.3543	0.3467	0.7050	1.0606
J_2	0.1257	0.1204	0.6294	1.2899
J_3	0.0013	0.0070	0.1911	0.4685
J_4	0.0209	0.0060	0.4100	1.0337
J_5	0.0000	0.0000	0.1037	0.4740
J_6	0.0001	−0.0019	0.0165	0.1897
J_7	0.0258	0.0215	0.0745	0.1680
J_8	0.0015	0.0000	0.0227	0.0321

way that we obtain a pattern vector; (b) we compare the calculated vector with pattern vector—we calculate the distance of the calculated vector to each of 32 pattern vectors; (c) we assume that there is an object in the image for which the distance from the calculated vector to pattern vector is the shortest.

The method presented in this paper uses a minimum distance classifier [13] which is based on calculation of the closest distance between current pattern vector and model pattern vector.

To define the distance between vectors, we can assume various metrics (20)–(23). For example these are:

$$\text{Minkowski metric: } \rho_n(x,y) = \sqrt[n]{\sum_{k=1}^{k=L} |x_k - y_k|^n} \tag{20}$$

$$\text{City-block metric: } \rho_1(x,y) = \sum_{k=1}^{k=L} |x_k - y_k| \tag{21}$$

$$\text{Euclidean metric: } \rho_n(x,y) = \sqrt{\sum_{k=1}^{k=L} (x_k - y_k)^2} \tag{22}$$

$$\text{Chebyshev metric: } \rho_\infty(x,y) = \max_k(|x_k - y_k|) \tag{23}$$

where: L—the dimension of space, x_k and y_k are the corresponding each other elements of vectors between which we calculate the distance. The following paper uses Minkowski's metric.

4 Examples

As an example of the developed object recognition algorithm we will present, for the clarity's sake, a simplified problem. Let us assume that we have to recognise in given test images (Fig. 8) an object 'a' for a rotation angle denoted as '0'. The set of four objects 'a', 'b', 'c' and 'd' will be examined as a set of images where we need to recognise the object 'a'. We consider a rotation angle denoted as '0' for all objects.

Therefore the set of examined images will consist of four objects. Using the Figs. 4, 5, 6 and 7 and Tables 1, 2, 3 and 4, we obtain the values of invariant functions characteristic for each object, cf. Table 5.

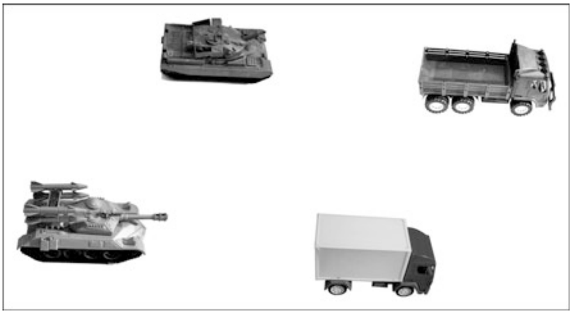


Fig. 8 The test image containing recognised objects 'a', 'b', 'c' and 'd' located in random places

The invariant functions vector was selected in such a way that the assumptions for the proper differentiation of objects are met. In the next step we conduct the image processing which consists of calculating J_1–J_8 for each image point $f(i, j)$ inside the assumed aperture. Further on, we calculate the differences $\rho_{`\mathrm{a}'}(i,j), \ldots, \rho_{`\mathrm{d}'}(i,j)$ (Minkowski's metric) between the model vectors for the recognised objects: ‘a’, ‘b’, ‘c’ and ‘d’ and the current feature vectors calculated for a given aperture location.

We treat that a given object (e.g. ‘a’) is located in this place in the image where the abovementioned difference (e.g. $\rho_{`\mathrm{a}'}$) for a given object (e.g. ‘a’) assumes the lowest value. The areas where the values of these differences are lowest, for the objects from the Fig. 8, are presented in Fig. 10. For the sake of figure's clarity, we presented in the Fig. 10 the isolines of the areas where the values of differences $\rho_{`\mathrm{a}'}, \ldots, \rho_{`\mathrm{d}'}$ are lowest. Therefore in these points of image where the values of $\rho_{`\mathrm{a}'}, \ldots, \rho_{`\mathrm{d}'}$ are lowest, we obtain in the Fig. 10 the isolines in colours which denote the objects. We assume that the recognised object is in the same point of the image where these lowest values occur. Figure 9 presents the location of the objects in the image in coordinate system which correspond to the coordinate system in Fig. 10.

Comparing the Fig. 10 with Figs. 8 and 9, we can observe that the system has correctly recognised all objects.

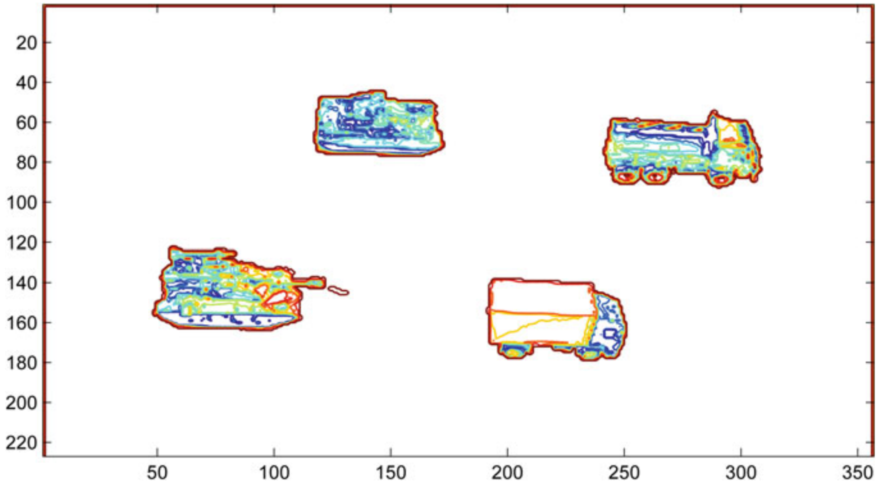


Fig. 9 The location of the objects in the image in coordinate system which correspond to the coordinate system in Fig. 10

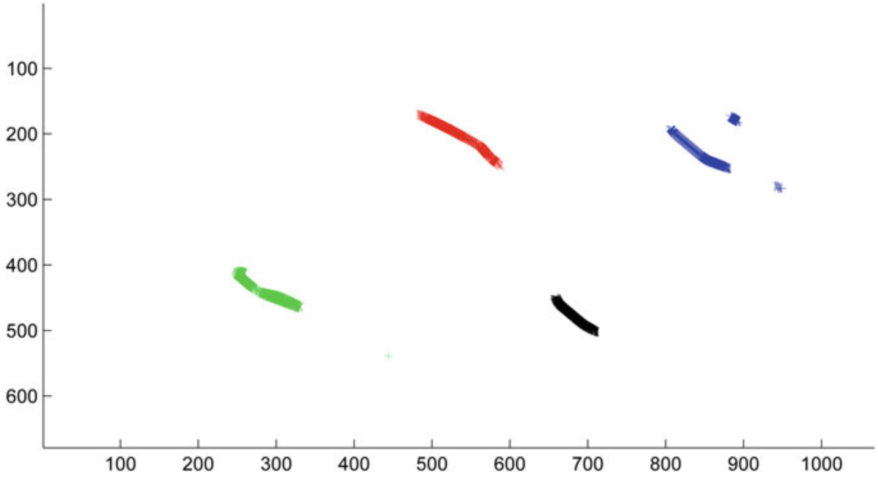


Fig. 10 The areas of the image where the values of $\grave{\rho}_{\mathrm{a}'}$ (*red*), $\grave{\rho}_{\mathrm{b}'}$ (*green*), $\grave{\rho}_{\mathrm{c}'}$ (*blue*) and $\grave{\rho}_{\mathrm{d}'}$ (*black*) are lowest

5 Conclusions

This study has focused on utilising moment invariants for object recognition in grey scale images without thresholding which is often used to obtain black and white images. The objects which were examined were 3D objects photographed at different object rotation angles on the ground, in different lighting and different angles between camera axis and the vertical direction.

The change of the abovementioned parameters influenced on the obtained values of the invariant functions. The parameter which caused the greatest changes was the angle of object rotation on the ground. These changes forced the multiple increase of number of the recognised object classes.

The authors presented a methodology used in recognising a given object within the group of other objects [22, 23]. The examples presented in the paper confirmed the correctness of this methodology.

References

1. Bieda, R., Jaskot, K., Jedrasiak, K., Nawrat, A.: Vision system for group of mobile robots, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 27–45 (2013). ISBN: 978-3-319-00368-9
2. Nawrat, A., Jedrasiak, K.: Fast colour recognition algorithm for robotics. Problemy Eksploatacji, 69–76 (2008)
3. Babiarz, A., Bieda, R., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images, vision based systems for UAV applications. Stud. Comput. Intell. **481**, 3–25 (2013). ISBN: 978-3-319-00368-9

4. Jedrasiak, K., Andrzejczak, M., Nawrat A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)
5. Davies, D., Palmer, P.L., Mirmehdi, M.: Detection and tracking of very small low contrast objects. In: Proceedings of the 9th British Machine Vision Conference, Sept 1998
6. Zhang, S., Karim, M.A.: Automatic target tracking for video annotation. Opt. Eng. **43**, 1867–1873 (2004)
7. Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP Graph Models and Image Process. **53**, 231–239 (1991)
8. Chesnaud, C., Refegier, P., Boulet, V.: Statistical region snake-based segmentation adapted to different physical noise models. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 1145–1157 (1999)
9. Gordon, N., Ristic, B., Arulampalam, S.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)
10. Sharp, C., Shakernia, O., Sastry, S.: A vision system for landing an unmanned aerial vehicle. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1720–1727. IEEE, Los Alamitos (2001)
11. Casbeer, D., Li, S., Beard, R., Mehra, R., McLain, T.: Forest Fire Monitoring With Multiple Small UAVs, Porland, OR, April (2005)
12. Papoulis, A.: Probability, Random Variables, and Stochastic Processes, 3rd edn. McGraw-Hill, New York (1991)
13. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision: Thompson Learning, Toronto (2008)
14. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guid. Control Dyn. **35**, 1354–1362 (2012)
15. Bibik, P., Narkiewicz, J.: Helicopter modeling and optimal control in autorotation. Ann. Proc. Am. Helicopter Soc. **64**(2), 986 (2008)
16. Bieda, R., Grygiel, R.: Wyznaczanie Orientacji Obiektu w Przestrzeni z Wykorzystaniem Naiwnego Filtru Kalmana. Przeglad Elektrotechniczny **90**, 34–41 (2014)
17. Galuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych. Przeglad Elektrotechniczny **86**, 278–283 (2010)
18. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. Appl. Mech. Mater. **249**, 1234–1246 (2013)
19. Sroka, M., Sciegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzalogowego pojazdu podwodnego - uklad stabilizacji i utrzymania zadanego kursu. Przeglad Elektrotechniczny **89**, 205–217 (2013)
20. Jaskot, K., Babiarz, A., Sroka, M., Sciegienka, P.: Prototyp bezzalogowego pojazdu podwodnego - konstrukcja mechaniczna, panel operatora. Przeglad Elektrotechniczny **89**, 52–67 (2013)
21. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 423–432 (2012)
22. Ulinowicz, M., Narkiewicz, J.: Modeling and identification of actuator for flap deflection. J. Autom. Mob. Robot. Intell. Syst. **5**, 35–40 (2011)
23. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorbtion during the progressive fracture process. WIT Trans. Model. Simul. **51**, 625–633 (2011)

Adjusting the Thresholds to the Recognised Pattern in Order to Improve the Separation Between the Recognised Patterns

Zygmunt Kuś and Aleksander Nawrat

Abstract The aim of the following study was to examine the influence of image thresholding on the correctness of the pattern recognition in the grey scale images. The method based on moment invariants, which were the elements of feature vectors defining the features of the recognised object, was used by authors in order to recognise the objects. The paper presents the influence of image thresholding with histogram equalisation for exemplary images on the distribution of the distance between pattern vector and feature vector for every pixel of an image. The authors have paid a great attention to the fact that proper selection of the thresholds is significant for distinguishing given object classes. One could conclude from the results that adjusting the value of the thresholds to grey levels, which were in a searched object, could considerably improve object recognition. The proposed method was based on the analysis of the part of the image obtained from a camera. We assumed that the camera was mounted on the UAV and it watched the objects moving on the ground. This part of the image was selected in this way that it contained only a tracked object. We computed the histogram of this part of the image, equalised this histogram and computed new thresholds on the basis of the equalised histogram. Next step was to threshold the abovementioned part of an image and then to compute a pattern vector for this thresholded image of the object. In this way we obtained pattern vector which was used to recognize the object in the whole image. Since we wanted to use the pattern vector obtained on the basis of thresholded object image, we had to threshold the whole image (a terrain with moving objects) with the same thresholds. The examples presented in the paper showed that image pre-processing based on thresholding might improve the accuracy of the pattern recognition. It was achieved thanks to the thresholding which was conducted in a manner that guaranteed to distinguish features of the recognised object. In order to calculate the distance (ρ) between an object pattern

Z. Kuś (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16 Street, Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_4

vector and pattern vector obtained for a given point of the whole image, we used the Euclidean metric. We compared the values of the ρ obtained during recognition process which was conducted for both: the thresholded and original—not-thresholded image. The lower values of the ρ for the thresholded image meant that in this case the pattern vector described the features of the object better.

Keywords Object recognition · Moment invariants · Pattern vector · Thresholding · Histogram equalisation

1 Introduction

The image processing is always numerically complex task owing to the fact that one has to deal with a large number of data which are contained in the image. The first step of reducing the amount of data is to convert a colour image to a grey scale image. In the next step, it is necessary to take into account what number of the threshold levels is indispensable to appropriately conduct the task for the image processing system. Owing to the fact that for some tasks it is not possible to reduce a grey scale image to a black and white image, it poses the problem of the amount and values of grey scale levels.

This paper will discuss the selection of the method for calculating the values of the thresholds during thresholding grey scale images. Thresholding will be carried out in order to decrease the amount of data defining the image and in order to decrease the calculation time during the recognition of the pattern (object) location in the image [1–3]. The chosen method has to guarantee the proper visualisation of the recognised pattern in the image. Decreasing the levels of greyness in the image after thresholding must be done in such a manner that it does not worsen the separation of the pattern from the background in the image.

Decreasing the number of thresholds, if it is done properly, may even emphasise the features of the object (pattern) which we are looking for and it may increase the effectiveness of pattern recognition.

Thresholding is used to separate the sought object from the background when there are differences in lighting or the colour of the object and background. In this case, despite decreasing the amount of the grey scale levels, it is possible to achieve better effectiveness of pattern recognition in the image.

We will put forward the method of calculating the value of the thresholds which is based on the histogram of the object and background.

Many solutions of a pattern recognition methods can be found in literature [4–10] and a lot of them is based on the feature vectors which elements are moment invariants [11, 12].

2 Formulating the Problem of Thresholding for Grey Scale Images

Gray-level thresholding is commonly used as the simplest segmentation process. In many cases, owing to the fact that the brightness of an object and background is different, we obtain a black and white image where the background is white and the object is black. However, in some cases, despite the differences in brightness, both a background and object have to be described by means of a greater number of grey levels than one. Adjusting the distribution of the threshold levels to the brightness of the object allows to distinguish the object from the background. Thresholding is worth carrying out because it is computationally inexpensive and relatively fast. What is more, thresholding can easily be done in real time using specialized hardware [13, 14].

The image thresholding resulting from the need for image segmentation is presented in [15–20].

Thresholding which is used for segmentation can be described in the following manner:

(a) we assume that $f(i, j)$ and $g(i, j)$ are input and output images;
(b) if $f(i, j) \geq T$ then we assume $g(i, j) = 1$ and treat as an object where T is the threshold;
(c) if $f(i,j) < T$ then we assume $g(i,j) = 0$ and treat as a background.

When a particular grey level interval represents the object whereas the background is lighter, and the recognised object darker, then we can use this thresholding.

In certain cases, it is necessary to use thresholding which does not result in a binary image; however, we obtain a significant reduction of grey levels amount.

The method of the choice of the thresholds set can be more efficient when it is possible to use the human-assisted analysis [21–23]. We have to cope with this situation when it is possible for an operator to mark in the image the region of the image where the recognised object appears. The set of grey levels in the marked region can indicate the thresholds which should be used.

The effective methods of the thresholds calculation are based on a histogram shape analysis.

This analysis is very simple for an image consisting of the object's pixels which have the same grey-level which is different from the grey-level of the background. In this case, we can observe in the histogram two peaks. The threshold can be calculated from the value of threshold for which there is a valley between these two peaks.

However, we have to deal with different situation when both the object and background have identical or similar grey scale levels. In such a case, we cannot use the difference in grey scale levels to differentiate an object from the background.

In the real cases, we will most often have in-between situations. What is meant by in-between situations is the situation when there is certain differentiation between grey shades in the fragments of the image which constitute the background and grey shades in the fragments of the image containing searched object. Our aim is to threshold an image in such a way that we can stress the difference between the background and the object.

One can distinguish three thresholding methods.

The first method we are going to discuss is even thresholding. In this method we obtain the reduction of grey scale levels by means of gathering a few adjacent grey scale levels and then replacing them with one grey scale level. The amount of the gathered levels denotes a degree of data amount reduction. The gathered levels are selected by defining the amount of adjacent levels which are going to be gathered.

The second method is called the histogram equalisation method [12]. We will use this method with a certain modification. We will not compute the histogram on the basis of whole image in which we search the object. On the contrary, we will compute the histogram only on the basis of the part of the image where the object is located. Therefore the histogram equalisation will only concern this very fragment of the image. It should result in a correct visualisation of an object which may be helpful for increasing the difference between the object and background.

The third method is also based on the analysis of the histogram for the part of the image containing the object. In this case, our goal is to take into consideration only these grey levels which constitute the greatest areas of the object image. This operation allows to obtain the image of the object without smaller details. It is a kind of an object segmentation. Not taking the background into consideration may cause that the shapes constituting the background will be deformed.

3 Examining the Influence of the Thresholding Method on the Usefulness of the Moment Invariants for Pattern Recognition

The thresholding method which will be proposed in this paper is based on a particular transformation of grey levels. This transformation will be based on histogram equalisation of the part of the image. The image consists of some objects moving in the terrain. We do not use the whole image, but only the part of this image which contains a recognised object. In the next step, we threshold the whole image (the objects in the terrain) on the basis of this equalised histogram. This method can be presented in the form the following algorithm:

(a) we choose a fragment of the whole image which contains the recognised object and calculate the table of grey levels transitions on the basis of the histogram equalisation method;

(b) we analyse the obtained equalised histogram and determine the thresholds according to (1);

$$t_1 = t : \left([H(t)]' = 0 \text{ and } [H(t)]'' > 0\right) \tag{1}$$

where t_1 is a grey level after thresholding, t is a grey level in an image before thresholding and $H(t)$ denotes equalised histogram for the image.

(c) we transform the grey levels of a whole image according to the thresholds computed in point (b);
(d) we calculate the pattern vector for the object using the image of the object after histogram equalisation and image thresholding;
(e) we carry out recognition of a described object in the whole image.

The abovementioned algorithm has to be conducted separately for each recognised object [24, 25]. It results in the necessity to carry out the calculation four times in the case of recognising four objects, as it will be presented in Sect. 4. It seemingly causes that the amount of the calculations increases; however, we have to notice that the histogram equalisation and thresholding are fast operations which can be supported by a dedicated signal processor. At the same time, thanks to these operations we obtain a significant decrease of the amount of data which is processed during pattern vector calculation and pattern recognition.

The feature vectors which consist of moment invariants will be used to describe features of the searched objects. The equations which define the used moment invariants are presented in (2)–(4).

$$\begin{aligned} J_1 &= \frac{I_1}{m_{00}^2};\ J_2 = \frac{I_2}{m_{00}^4};\ J_3 = \frac{I_3}{m_{00}^5};\ J_4 = \frac{I_4}{m_{00}^5};\\ J_5 &= \frac{I_5}{m_{00}^{10}};\ J_6 = \frac{I_6}{m_{00}^7};\ J_7 = \frac{I_7}{m_{00}^4};\ J_8 = \frac{I_8}{m_{00}^5}; \end{aligned} \tag{2}$$

$$\begin{aligned} I_1 &= M_{20} + M_{02}\\ I_2 &= (M_{20} - M_{02})^2 + 4M_{11}^2\\ I_3 &= (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2\\ I_4 &= (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2\\ I_5 &= (M_{30} - 3M_{12})(M_{30} + M_{12})((M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2)\\ &\quad + (3M_{21} - M_{03})(M_{21} + M_{03})(3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2);\\ I_6 &= (M_{20} - M_{02})((M_{30} + M_{12})^2 - (M_{21} + M_{03})^2)\\ &\quad + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03})\\ I_7 &= M_{20}M_{02} - M_{11}^2\\ I_8 &= M_{30}M_{12} + M_{21}M_{03} - M_{12}^2 - M_{21}^2 \end{aligned} \tag{3}$$

$$\begin{gathered} \text{Central moments of order } (p+q):\\ M_{pq} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(i - x_s)^p(j - y_s)^q f(i,j)\\ \text{The co-ordinates of the region's centre of gravity (centroid):}\\ x_s = \tfrac{m_{10}}{m_{00}};\ y_s = \tfrac{m_{01}}{m_{00}}\\ \text{Moments of order } (p+q):\\ m_{pq} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} i^p j^q f(i,j) \end{gathered} \tag{4}$$

where $f(i, j)$—image pixel; M—the height of the image; N—the width of the image.

We expect that adjusting grey levels in the whole image to the grey levels depicting the object will distinguish object's features.

At the same time, it is possible that the details which are placed beyond object pixels region will be lost or distorted. It depends on the grey levels which constitute the part of the image containing the image of the terrain where the object is moving. We predict that this fact not only cannot worsen, but it can even improve the image recognition quality.

4 Example

We will present as example a case of four objects, which we will have to locate, moving in the terrain.

In the subsequent examples, we will calculate the thresholds, which are used to threshold the whole image, on the basis of this object image which is to be searched.

We will calculate the thresholds adjusted to the object image according to the example presented in Fig. 2 and in Eq. (1).

Firstly, we are going to present the case in which the images of objects and terrain are not processed. The pattern recognition will be conducted for the original images which will only be resized in order to decrease the time of calculations.

Figure 1 presents the whole image of the terrain where the recognised objects are moving. There are four various vehicles moving on the stony ground.

The objects which will be searched are shown in Fig. 2. The presented vehicles are denoted as objects 1–4.

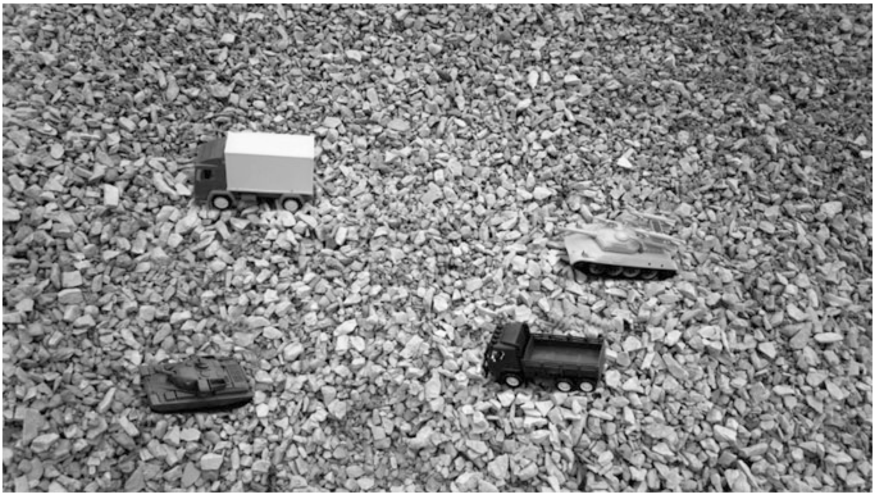


Fig. 1 The whole image of the terrain where the recognised objects are moving

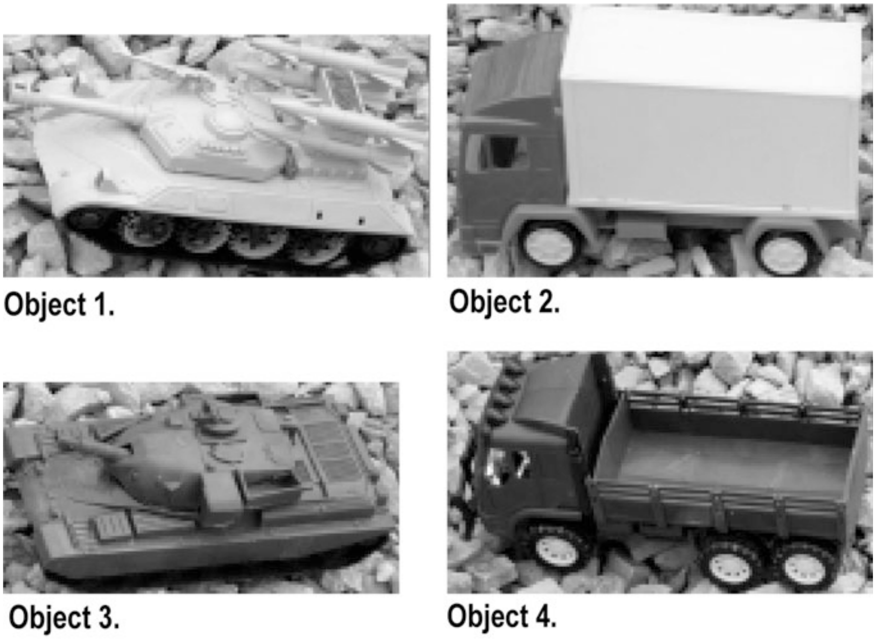

Object 1.
Object 2.
Object 3.
Object 4.

Fig. 2 The images of the recognised objects

Figure 3 presents the locations of the objects calculated for Fig. 1. On the basis of the objects' patterns representing the objects' images showed in Fig. 2.

According to Fig. 3, we can state that: (a) the object 1 (red) was recognised correctly; (b) the object 2 (green) was recognised in few places—one of them is correct, but we cannot state which of these places is the correct location; (c) the object 3 (blue) was recognised in two places where only one is correct (blue point on the left); (d) the object 4 (pink) was recognised correctly.

Secondly, we are going to transform both the object images and the image of the terrain. We will conduct four transformations of these images—each of them will be adjusted to the features of the subsequent object. The first stage of this transformation consist of histogram equalisation. It is carried out separately for each object image as it is presented in Fig. 4.

As a result of this operation, we obtain for each object a new set of grey levels—adjusted to a given object. We use separately each of these sets for transforming the terrain image so in this way we obtain four images of the terrain presented in Figs. 5, 6, 7 and 8.

According to Fig. 5, we can state that the whole image is not altered too much. According to Fig. 6, we can state that the whole image is altered in such a way that the grey scale for the object 2 is extended; however, the rest of the image is slightly altered. According to Fig. 7, we can state that the whole image is considerably brightened. It is caused by the dark image of the object 3. However, the object 3 is

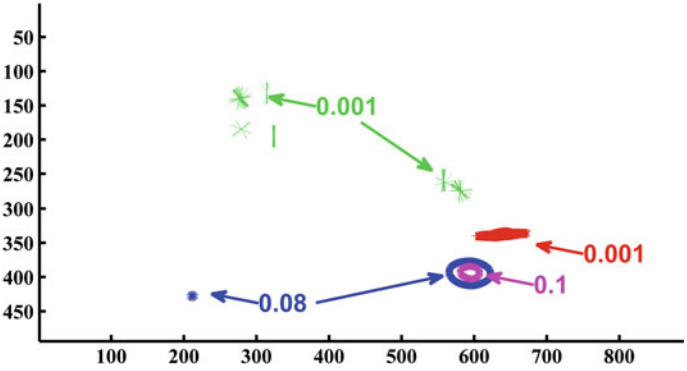


Fig. 3 The locations of the recognised objects in the image presented in Fig. 1 with marked minimum values of the distance $\rho_{\min}$. Object 1—*red*, Object 2—*green*, Object 3—*blue*, Object 4—*pink* (Color figure online)

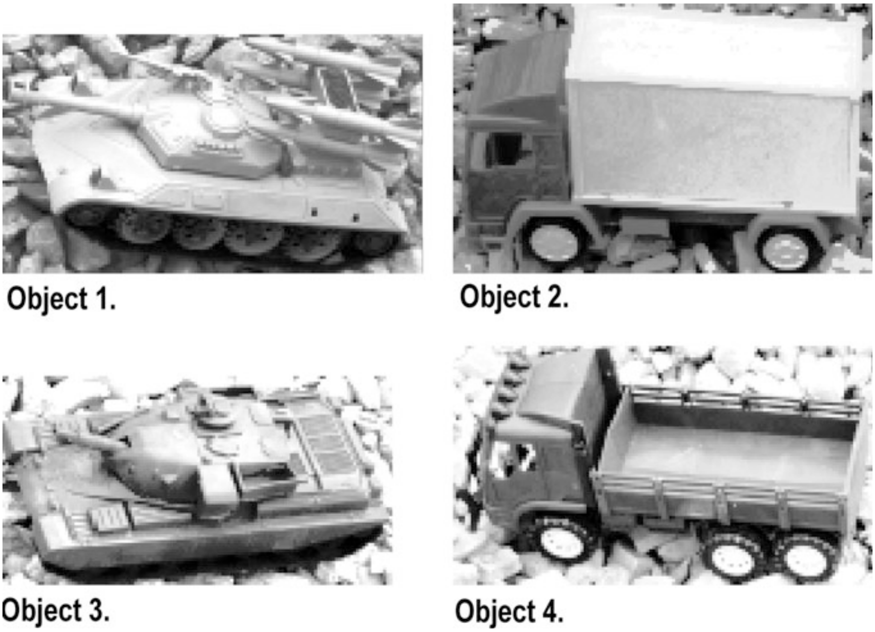


Fig. 4 The images of the recognised objects after the operation of histogram equalization

clearly visible in Fig. 7. According to Fig. 8, we can state that the image is brightened even more than in the case of Fig. 7. The dark grey levels defining the object 4 in the original image cause this effect. Nevertheless, the object 4 is clearly visible in Fig. 8.

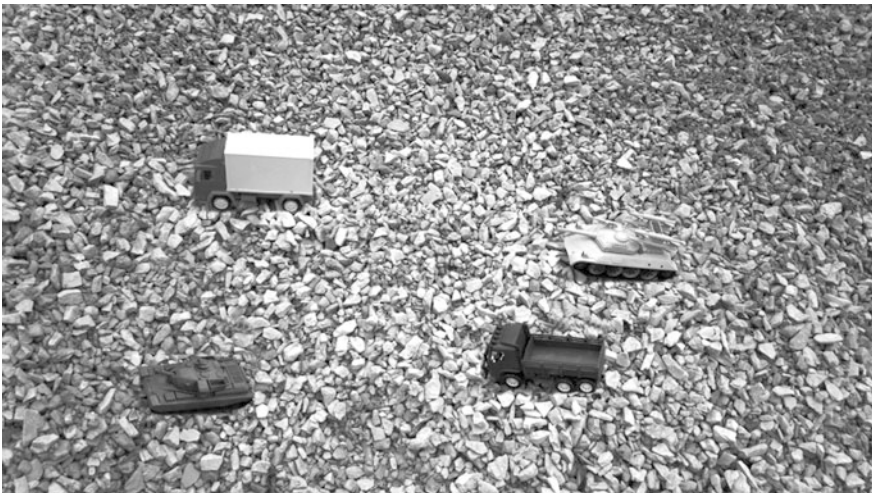


Fig. 5 The whole image of the terrain where the recognised objects are moving obtained after histogram equalisation according to grey levels obtained for the object 1

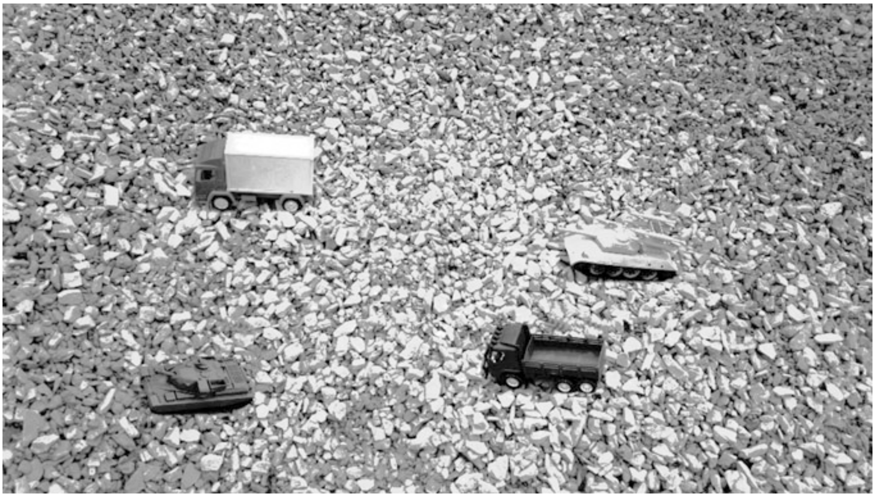


Fig. 6 The whole image of the terrain where the recognised objects are moving obtained after histogram equalisation according to grey levels obtained for the object 2

The abovementioned effects can be illustrated by comparing the histograms before and after equalisation. The histograms for all objects are presented in Figs. 8, 9, 10, 11 and 12. We can see that for all objects the histogram equalisation resulted in the more even distribution of the grey scale levels. Each presented histogram corresponds to one object's image. The application of the grey levels set obtained

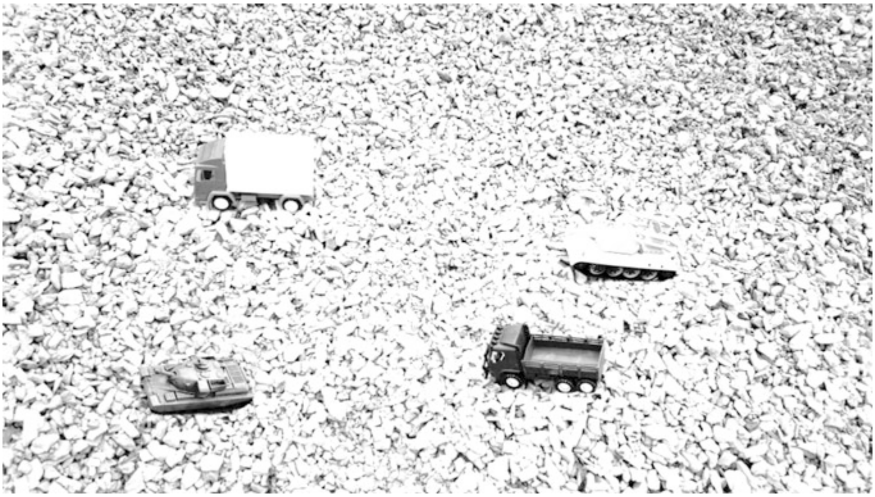


Fig. 7 The whole image of the terrain where the recognised objects are moving obtained after histogram equalisation according to grey levels obtained for the object 3

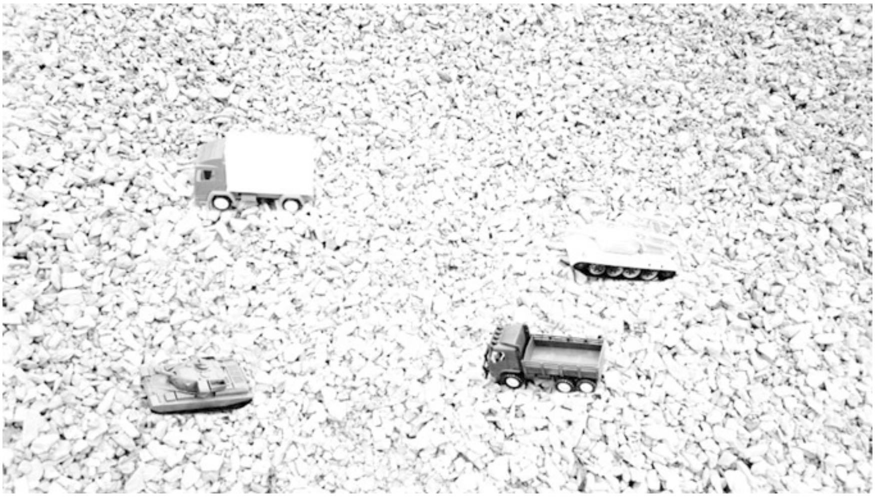


Fig. 8 The whole image of the terrain where the recognised objects are moving obtained after histogram equalisation according to grey levels obtained for the object 4

after histogram equalisation conducted for a particular object improves a visibility of this particular object. However, the part of the terrain image beyond this object can be 'destroyed'—including the parts of terrain images which consist of other objects. The abovementioned effect can be observed by comparing Figs. 5, 6, 7 and 8 to the original terrain image presented in Fig. 1.

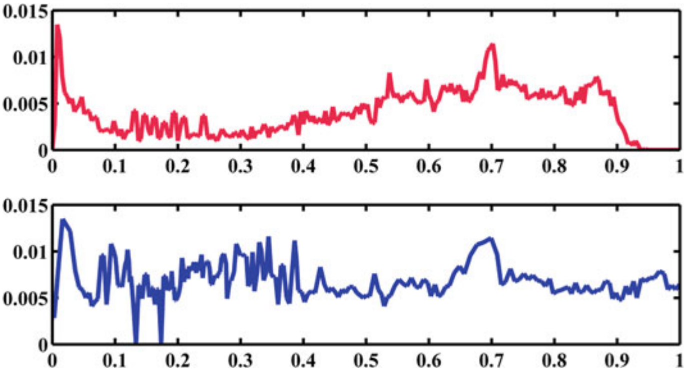


Fig. 9 The histogram of the object 1 before (*red*) and after (*blue*) equalisation (Color figure online)

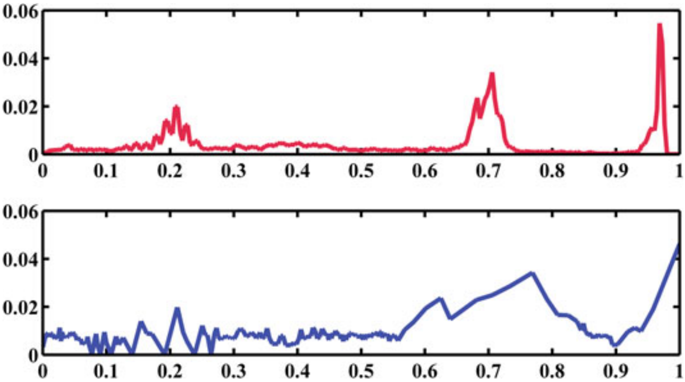


Fig. 10 The histogram of the object 2 before (*red*) and after (*blue*) equalisation (Color figure online)

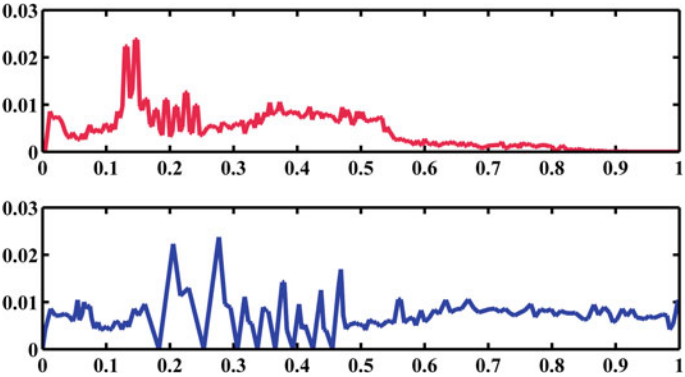


Fig. 11 The histogram of the object 3 before (*red*) and after (*blue*) equalisation (Color figure online)

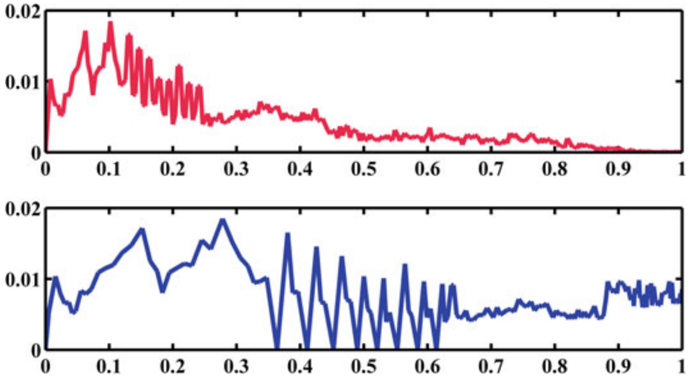


Fig. 12 The histogram of the object 4 before (*red*) and after (*blue*) equalisation (Color figure online)

The next stage of the pre-processing of the image will be image thresholding. In order to calculate the thresholds for each object, we will use object images obtained after histogram equalisation. The results of these thresholds application for object images were presented in Fig. 17.

Figures 13, 14, 15 and 16 present the histograms for the objects images after histogram equalisation (black) and the same histograms averaged with marked thresholds calculated according to formula (1).

According to Figs. 13, 14, 15 and 16, after thresholding we obtain far lower number of the thresholds in the image description.

Figure 17 shows objects' images after histogram equalisation and thresholding with thresholds (red rings) obtained on the basis of the averaged histograms (blue line) presented in Figs. 13, 14, 15 and 16.

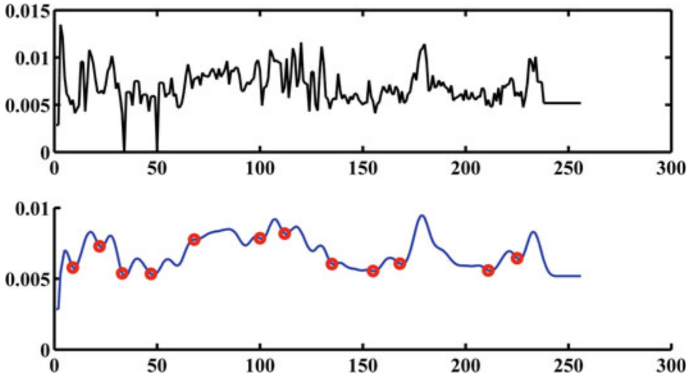


Fig. 13 The histogram for the image of the object 1 after histogram equalisation (*black*) and the same histogram averaged with marked thresholds calculated according to formula (1) (Color figure online)

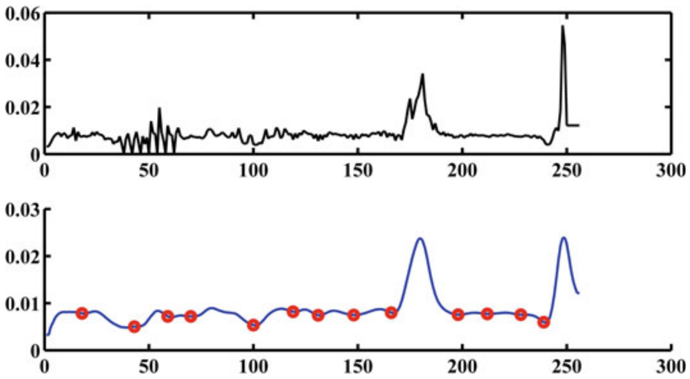


Fig. 14 The histogram for the image of the object 2 after histogram equalisation (*black*) and the same histogram averaged with marked thresholds calculated according to formula (1) (Color figure online)

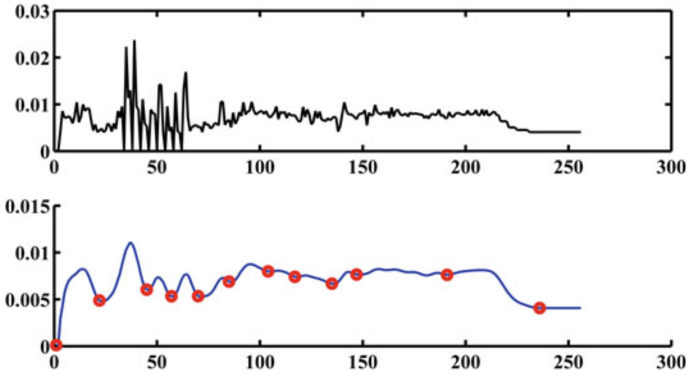


Fig. 15 The histogram for the image of the object 3 after histogram equalisation (*black*) and the same histogram averaged with marked thresholds calculated according to formula (1) (Color figure online)

The next stage consists of thresholding the whole terrain image. In order to do so, we will use, as the source images, images after histogram equalisation (Figs. 5, 6, 7 and 8). We present the results of such terrain image thresholding in Figs. 18, 19, 20 and 21. It can be seen that the effect of the vivid representation of the searched object was not lost despite considerable decrease in the number of the grey levels.

The results of object recognitions are presented in Figs. 22, 23, 24 and 25.

There are marked the minimum values of the distance $\rho_{\min}$. Each value is marked in the place in the image where this value occurs in the image. In order to calculate the distance (ρ) between an object pattern vector and pattern vector obtained for a given point of the whole image, we used the Euclidean metric. According to

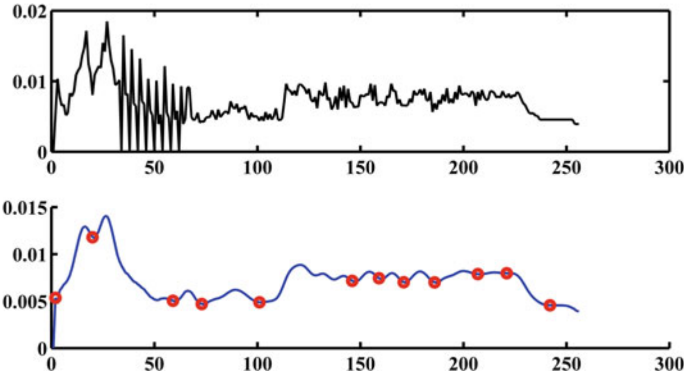
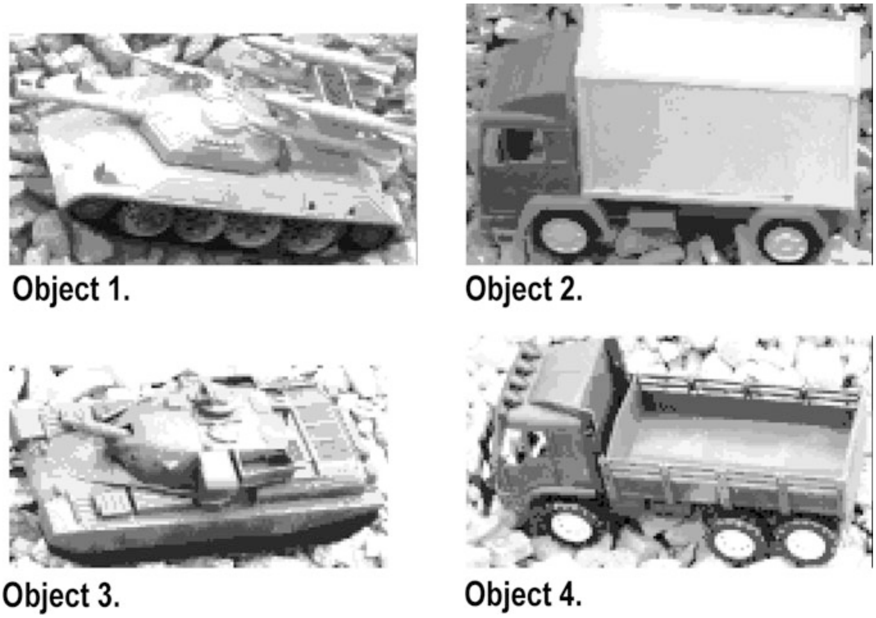


Fig. 16 The histogram for the image of the object 4 after histogram equalisation (*black*) and the same histogram averaged with marked thresholds calculated according to formula (1) (Color figure online)


Object 1.

Object 2.

Object 3.

Object 4.

Fig. 17 The objects' images after histogram equalisation and thresholding with the thresholds from Figs. 13, 14, 15 and 16

Fig. 22, we can observe that the value of $\rho_{\min}$ is the lowest for the object 1. It means that in this figure the recognition of the object 1 is easier than the recognition of the other objects in this figure.

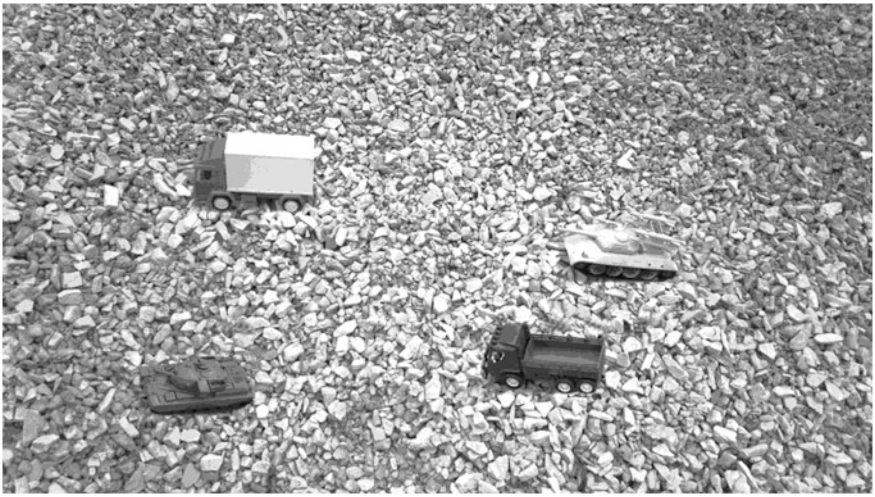


Fig. 18 The terrain image after histogram equalisation and thresholding operations adjusted to the features of the object 1

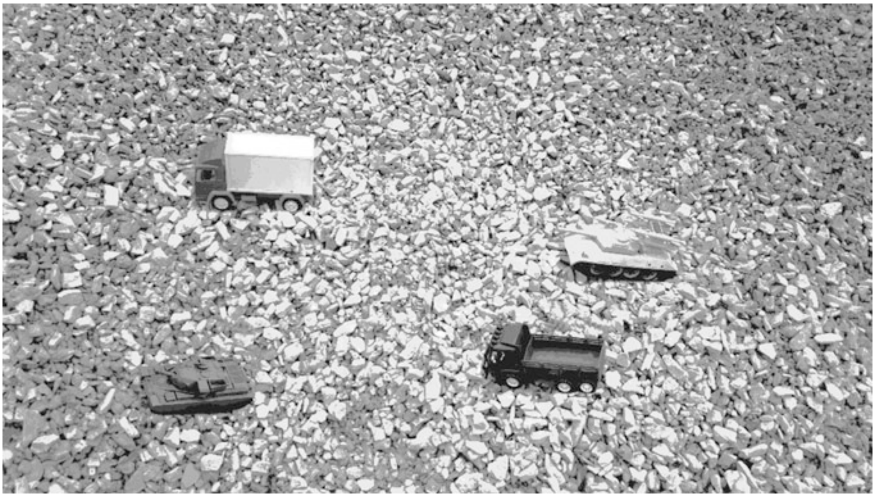


Fig. 19 The terrain image after histogram equalisation and thresholding operations adjusted to the features of the object 2

The comparison of the minimum distance value $\rho_{\min}$ to this value in original terrain image, presented in Fig. 1, let us conclude that after histogram equalisation and thresholding we obtain lower values of the $\rho_{\min}$.

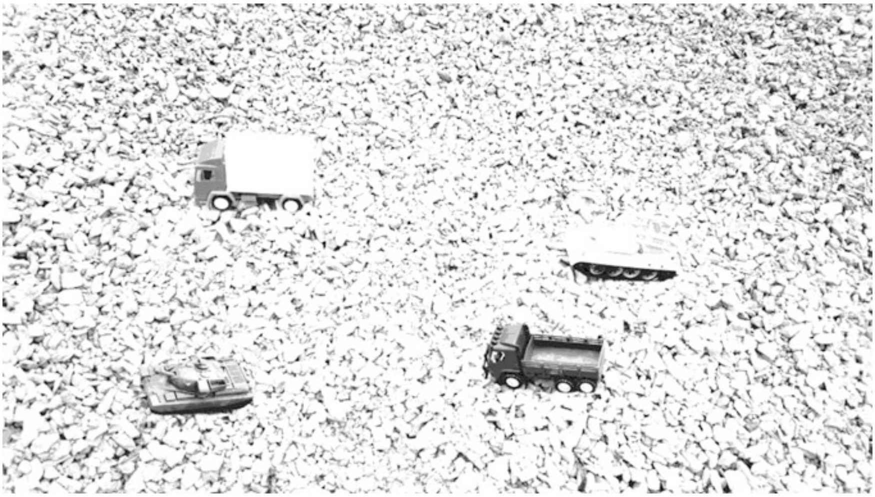


Fig. 20 The terrain image after histogram equalisation and thresholding operations adjusted to the features of the object 3

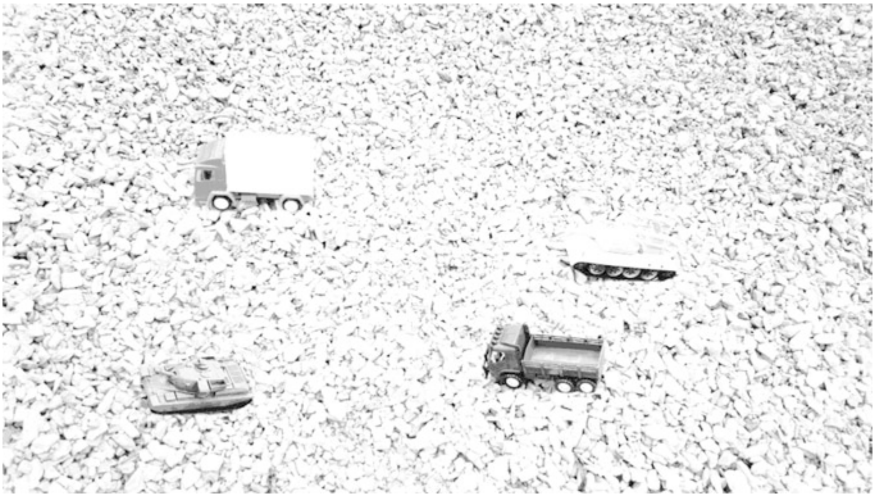


Fig. 21 The terrain image after histogram equalisation and thresholding operations adjusted to the features of the object 4

Analogically, we can observe the same effect in Figs. 23, 24 and 25 for the objects 2–4.

Comparing the distribution of ρ for the original image with the distribution of ρ for the image after thresholding, we can see that for the thresholded image ρ

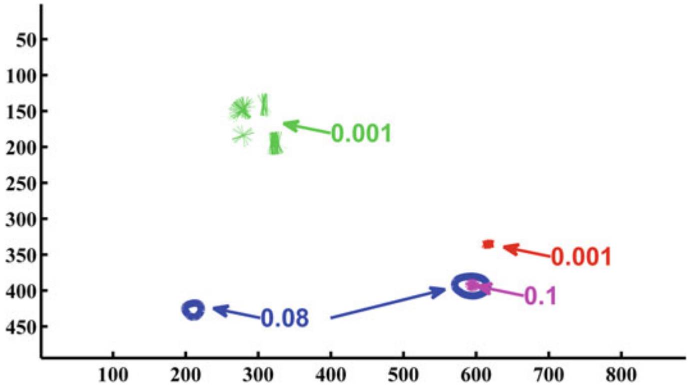


Fig. 22 The locations of the recognised objects in the image presented in Fig. 18 with marked minimum values of the distance $\rho_{\min}$. Object 1—*red*, Object 2—*green*, Object 3—*blue*, Object 4—*pink* (Color figure online)

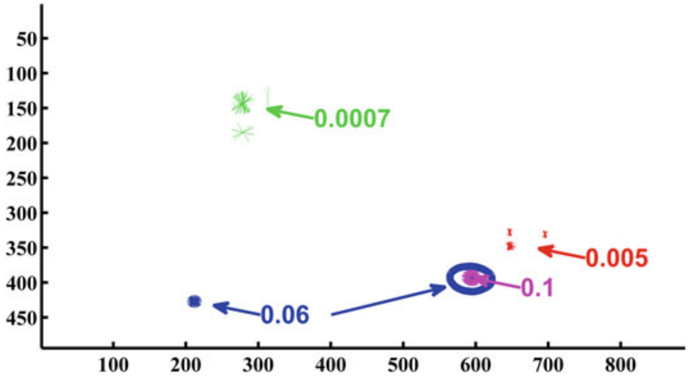


Fig. 23 The locations of the recognised objects in the image presented in Fig. 19 with marked minimum values of the distance $\rho_{\min}$. Object 1—*red*, Object 2—*green*, Object 3—*blue*, Object 4—*pink* (Color figure online)

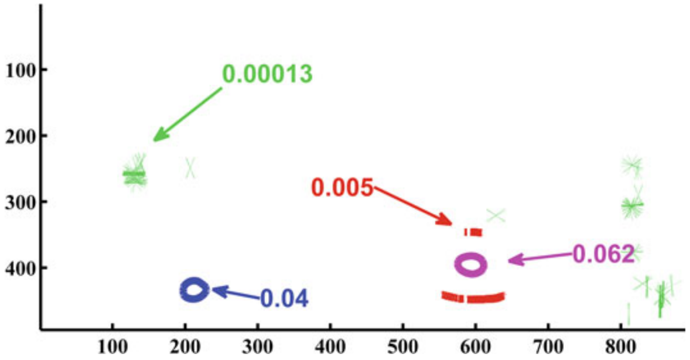


Fig. 24 The locations of the recognised objects in the image presented in Fig. 20 with marked minimum values of the distance $\rho_{\min}$. Object 1—*red*, Object 2—*green*, Object 3—*blue*, Object 4—*pink* (Color figure online)

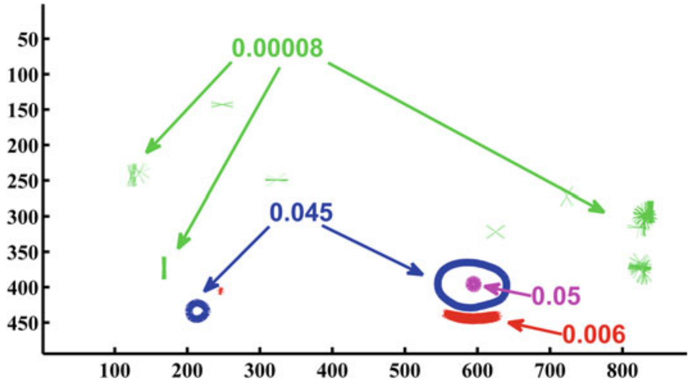


Fig. 25 The locations of the recognised objects in the image presented in Fig. 21 with marked minimum values of the distance $\rho_{\min}$. Object 1—*red*, Object 2—*green*, Object 3—*blue*, Object 4—*pink* (Color figure online)

achieves lower values in the place of the searched object location. This fact allows to conduct correct object localisation easier.

Since the histogram equalisation was adjusted to a given object, the other objects are recognised less effectively. This problem can be solved by repeating all the equalisation, thresholding and recognising processes for each object.

5 Conclusions

To sum up, the paper spelt out the methodology of the object recognition improvement. This effect was obtained thanks to some previous image transformations which allowed to improve the visibility of the objects in the terrain image.

The presented method consisted of two stages. The first one, was histogram equalisation—conducted for each object separately. The second one, was thresholding of the terrain image with the thresholds obtained on the basis of the equalised histogram—for each object separately. Finally, we obtained the terrain image transformed in this way that the object is clearly visible—the object for which equalisation and thresholding were conducted.

The most significant result of the presented methodology is to enhance the correctness of object recognition. What is more, thanks to a lower number of grey scale levels we obtain the shorter time of the calculations. These are the calculations which are necessary to provide recognition process. The presented approach allows to improve the quality and speed of the recognition process [26, 27].

References

1. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics, vol. 7594, pp. 407–413. Lecture Notes in Computer Science (2012)
2. Bereska, D., Daniec, K., Jedrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 115–121, (2013). ISBN: 978-3-319-00368-9
3. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, vol. 8671, pp. 302–315. Lecture Notes in Computer Science (2014)
4. Davies, D., Palmer, P.L., Mirmehdi, M.: Detection and tracking of very small low contrast objects. In: Proceedings of the 9th British Machine Vision Conference, Sept 1998
5. Zhang, S., Karim, M.A.: Automatic target tracking for video annotation. Op. Eng. **43**, 1867–1873 (2004)
6. Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP Graph Models Image Process. **53**, 231–239 (1991)
7. Chesnaud, C., Refegier, P., Boulet, V.: Statistical region snake-based segmentation adapted to different physical noise models. IEEE Trans. Patt. Anal. Mach. Intell. **21**, 1145–1157 (1999)
8. Gordon, N., Ristic, B., Arulampalam, S.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)
9. Sharp, C., Shakernia, O., Sastry, S.: A vision system for landing an unmanned aerial vehicle. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1720–1727. IEEE, Los Alamitos (2001)
10. Casbeer, D., Li, S., Beard, R., Mehra, R., McLain, T.: Forest Fire Monitoring With Multiple Small UAVs, Porland, OR, April 2005
11. Papoulis, A.: Probability, Random Variables, and Stochastic Processes, 3rd edn. McGraw-Hill, New York (1991)
12. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machina Vision. Thompson (2008)
13. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. In: Computer Vision and Graphics, vol. 7594, pp. 423–432. Lecture Notes in Computer Science (2012)
14. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics, vol. 7594, pp. 87–94. Lecture Notes in Computer Science (2012)
15. Adollah, R., Mashor, M.Y., Francis, E.U., Harun, N.H.: Bone marrow image segmentation based on multilevel thresholding. In: International Conference on Biomedical Engineering (ICoBE), Penang 27–28 Feb 2012
16. Yan, F., Zhang, H., Kube, C.R.: A multistage adaptive thresholding method. Pattern Recogn. Lett. **26**(8), 1183–1191 (2005)
17. Davies, E.R.: Chapter 4—Thresholding Techniques Computer and Machine Vision, 4th edn., pp. 82–110 (2012)
18. Chang, J.-S., Mark Liao, H.-Y., Hor, M.-K., Hsieh, J.-W., Chern, M.-Y.: New automatic multi-level thresholding technique for segmentation of thermal images. Image Vision Comput. **15**(1), 23–34 (1997)
19. Ma, J., Wen, D., Yang, S., Wang, L., Zhan, J.: Hierarchical segmentation based on a multilevel thresholding. In: 2010 3rd International Congress on Image and Signal Processing (CISP) (2010)
20. Zhang, X., Liu, D.: A multilevel thresholding method based on improved chaos optimization for image segmentation. In: International Conference on Information Engineering and Computer Science, ICIECS (2009)

21. Bibik, P., Gradolewski, S., Zawislak, W., Zbudniewek, J., Darakchiev, R., Krczel, J., Michalski, M., Strzelczyk, K.: Problems of detecting unauthorized satellite transmissions from the VSAT terminals. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–4, 8–9 Oct 2012. ISBN: 978-1-4673-1422-0
22. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical analysis of IED detonation effect on steel plate. Acta Mechanica et Automatica **6**, 10–12 (2012)
23. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorbtion during the progressive fracture process. WIT Trans. Model. Simul. vo. **51**, 625–633 (2011)
24. Bieda, R., Grygiel, R.: Wyznaczanie Orientacji Obiektu w Przestrzeni z Wykorzystaniem Naiwnego Filtru Kalmana. Przeglad Elektrotechniczny **90**, 34–41 (2014)
25. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. Appl. Mech. Mater. **249**, 1234–1246 (2013)
26. Bieda, R., Jaskot, K., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 3–25 (2013). ISBN: 978-3-319-00368-9
27. Babiarz, A., Bieda, R., Jaskot, K.: Vision system for group of mobile robots. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 139–156 (2013). ISBN: 978-3-319-00368-9

The Concept of an Active Thermal Camouflage for Friend-Foe Identification System

Paulina Wilk, Tomasz Targiel, Dawid Sobel, Jan Kwiatkowski, Karol Jędrasiak and Aleksander Nawrat

Abstract The chapter presents a concept of utilization of active thermal camouflage combined with friend or foe recognition system. The main purpose of the system is automatization of decision-making, whether utilization of thermal camouflage is ought to be used. Moreover, the described system should react properly, when an enemy is detected. Implementation of mentioned purposes. As a way of accomplishing of the described functionalities, an active thermal camouflage, combined with vision system, which is able to distinguish, whether an observed object is friend or foe, is proposed. Furthermore, an example application of the system, supported by a net of sensors and decision-making model, was described. For testing purposes, the system was simplified by utilization of a single camera and an adaptive thermo camouflage module. The obtained results confirmed that such a solution is possible to be implemented in the future.

1 Introduction

Camouflage is one of multiple ways of combat security, known and used since many years. The main purpose of camouflage is confusion of enemies by making allies invisible to them. The simplest form of camouflage is making an allies similar to their environment, in a way which makes them impossible to be identified through optical way. The described functionality is achieved by utilization of fragments of locally existing plants, or by utilization of appropriate colors and camouflage patterns. These methods are described as passive camouflage, and are strictly connected to environmental and weather conditions. In others words, there is no existing pattern, which could be used in every terrain.

P. Wilk (✉) · T. Targiel · D. Sobel · J. Kwiatkowski · K. Jędrasiak · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Akademicka, 16, 44-100 Gliwice, Poland
e-mail: paulina.wilk@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_5

Nowadays, there exists a well-known possibility of identification of an passive camouflage covered enemy. The solution, known as thermography, is based on measurement of emitted thermal radiation [1–3]. The higher is the temperature of the object, the greater is the intensity of the thermal radiation. Thermal imaging cameras operate in the wavelength range of infrared radiation, outside of visible spectrum, therefore the passive camouflage, designed for visible spectrum, does not meet the requirements [4].

Therefore occurs a need of providing of methods, which would create a possibility to deceive thermal video systems of an enemy. The simplest solution is utilization of materials, which reduce the emission of heat (passive thermal camouflage). However, the more interesting solution is the handling of the emission of heat in a way, which provides not only possibility to remain invisible, but also possibility to deceive the enemy. That solution is often referred as an active thermal camouflage. In example, a tank, equipped with an active thermal camouflage system, could reshape itself in an image, acquired by thermal camera of an enemy, to be seen as a farm tractor.

The current level of technical advancement does not create a possibility to develop an active thermal camouflage system, which could be carried by a human, due to the heaviness of system elements and efficient sources of energy. Therefore, the article is focused on development of a system which can be used as a vehicle cover. Vehicles are mainly equipped with combustion engines, which are generating high amount of heat. In order to prevent the heat from being emitted, a lot of energy is needed. Minimization of energy consumption requires an active thermo camouflage to be used only when it is necessary—when an enemy is detected within a vehicle surroundings. Environmental conditions, weather, human factor—may lead to incorrect identification of the approaching object [5–7]. Therefore, it is important to create an effective friend or foe recognition system.

The basic assumption of such a system would be utilization of active thermal camouflage cooperating with a set of cameras, which task would be the identification of the enemy. Moreover, the system would be supported by a human independent decision-making model [8]. It should be noted that the response time of such a system must be very short.

2 Existing Solutions

Cases of fratricidal fire during wars resulted in the development of the so-called recognition technology. Combat Identification (CID)—is a process of classification of detected objects as members of one of the groups: friend, enemy, neutral, or unknown. Fratricidal fire is bombardment of an allied forces and resources. The main cause of that occurrence is incorrect identification of units on the battlefield. Article [9] presents known fratricide cases from the eighteenth century. Only since the Gulf War in 1990–1991, the Americans began to pay a special attention to this issue. Despite utilization of the latest identification systems, due to fratricidal fire, 35 American soldiers were killed and 72 were wounded. Fratricide is not a new phenomenon, such cases have been occurring and probably will occur, because, despite of the use of the latest technology, it is not possible to completely eliminate human error.

Currently, every army is equipped with friend or foe recognition system, which consists of several different technologies working together. Due to the fact that these are military technologies, much of information regarding them is confidential. In [10] the Combat Identification systems available today have been grouped and described in detail. Of these three groups (passive signaling devices, active signaling devices and interrogation/response system) are characterized by the possibility of a very rapid response—and for this reason only those will be given under consideration later in the article.

Technologies that create possibility to label people and vehicles are part of the group of passive signaling devices. Among others, those are: IR fluorescent paints, identification panels and smoke markers. Tagged objects using the first two technologies can be seen in an ordinary or thermal imaging camera. IR paints are characterized by low cost of implementation. Installation of Identification panels (flat rectangular components coated with low emissivity thermal tape) requires special procedures. There are two types of such panels: combat identification panels are typically used for determination and subsequent recognition of objects on the ground, while the thermal identification panels are designed for flying objects. The last specified technology among described group are smoke markers, which are used to mask the position of people and equipment, or to confuse the enemy.

The second group of technologies used to identify objects on the battlefield are active signaling devices, which emit an electromagnetic signal. The most widely used vehicle-mounted device of this group is the infrared beacon, which periodically sends pulses of radiation in the near-infrared, invisible to the naked eye or thermal imaging camera [11]. Therefore, this device is only used during military operations carried at night.

Interrogation/response systems allow object recognition through a process of queries and responses. Among these technologies the oldest and the best known system is IFF (identification, friend or foe), which was first used during World War II [12]. As one of the means of defense of the UK, an early detection radar network was built (code name: Chain Home). This system could detect aircraft at high altitudes [13], but it was prone to interference—radar echoes caused distinguishing between the Allied and enemy aircraft impossible to perform. The IFF Mark I system was patented and introduced in 1939. The general idea of this system is implementation of a special onboard device—a transmitter, which, under the influence of the radar signal, generates an additional signal received by the radar on the ground. Today, the United States and NATO countries use the system Mark XII—it is a standardized system of IFF, in a way which provides, that all befriended individuals can be identified correctly.

The above solutions are focused on the correct identification of objects—in most cases they are able to detect only a friendly unit and thereby reduce probability of accidental firing at their own units. In addition, the use of signaling devices requires appropriate training of soldiers in correct identification of tagged objects. However, there are no solutions regarding systems allowing automatic enemy identification and hiding using active thermal camouflage. Such a system could be used if the unit cannot attack or carry out a recon mission.

There are many publications regarding active systems for thermal camouflage. The only presented to the public and working technology is Adaptive, an active thermal camouflage developed by an international company BAE Systems AB [14]. The solution consist of modules based on Peltier cooler. They allow heating or cooling of its surface—and thus the opportunity to deceive an optical infrared sensor. Camouflage creates a possibility to dynamically change the temperature in a range, which can be recorded by thermal imaging camera [15]. The solution problem is a long time of temperature change (basing on footage dynamic changes in temperature are estimated at approx. 5 °C per second).

3 System Architecture

How was mentioned earlier there is no a solution which uses recognition system to detection enemy units to attack them but in order to hide, confuse the enemy active thermal camouflage is used. Will be described below the main assumptions of the architecture of the system [16, 17].

The first and the most important assumption is very short time of reaction. We can assume that this time does not exceed 30 ms. This mean that the recognition and total process of camouflage themselves must be performed within the specified time. If video from thermovision is refreshed 30 times per second, the activation of the thermo camouflage must take place between one frame and another. Due to the short time of response it should be ruled out aspect of making decisions by human. It should be implemented a system of automatic recognition which will be fast enough.

This particular problem is related with the second assumption of the system. The algorithm of recognition allied and enemy units must be sufficiently fast and effective in order to use of the thermal camouflage was sensible. In case when there is not possible to classify an object into one of three groups (friend, enemy, neutral) the system should immediately turn on thermal camouflage system in hide mode and adapts to ambient temperature [18, 19].

Another condition working of the system are assumptions regarding of active thermo camouflage. This should be a flat shell, which allows to cover the whole vehicle, entire buildings, which is solid, easy to conservation and repair. One of the proposals is to use a modular solution. This will allow for quick replacement of defective parts, adjusting to the surfaces of various shapes and facilitate mass production. In addition, such a system should allow to long action of the thermo camouflage, up to several hours, what mean that it cannot charge too much energy.

Example scheme of the application of the system is shown in Fig. 1. Around the base was distributed network of sensors (distance sensors, cameras). Information

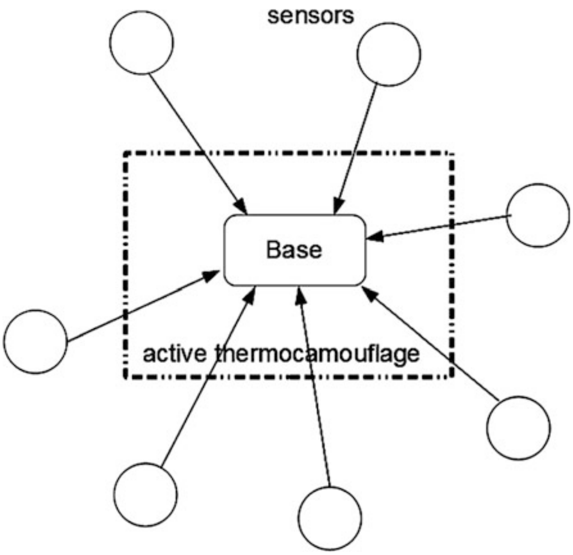


Fig. 1 Schematic diagram of the application

from the network are collected in the main computer in base, which is responsible for identifying object in operating area of sensors. By using automatic decision model a signal is generated for the thermal camouflage, which is applied on the entire base or vehicles located near. The whole procedure, from detection of approaching object by classifying it and turning off thermal camouflage takes less than 30 ms.

Another example of the use of thermal camouflage is to install it on a single vehicle with a digital cameras system. On the basis of the image from the camera system detects and classifies objects and then controls the thermal camouflage.

4 Experimental Verification of Concept of Friend or Foe Recognition System

For the tests, due to very extensive system architecture, concept of active thermal camouflage with the friend or foe recognition system has been simplified.

In order to demonstrate the work of the system there was created a simple system consisting of a USB camera Microsoft HD-3000, on the mobile platform, connected to the computer Raspberry Pi and Peltier cooler. System together with the aluminum plate and a prototype driver of the Peltier cooler is a single module of the thermal camouflage. System diagram is shown in Fig. 2.

For the purpose of the tests was built movable camera base consist of two connected servos. The construction is shown in Fig. 3. Raspberry Pi platform has

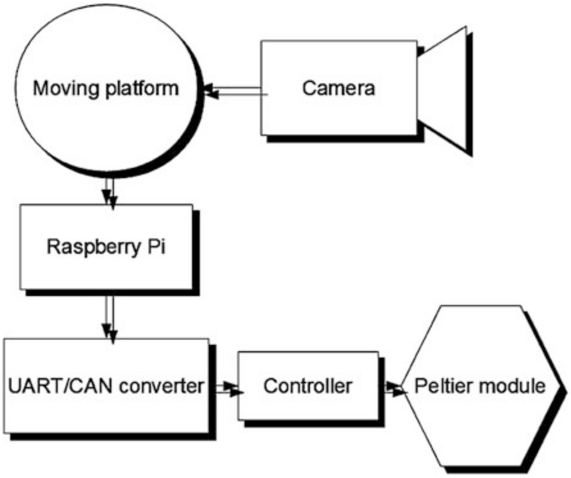


Fig. 2 Demonstrator scheme

Fig. 3 The camera module with raspberry pi

only one hardware PWM so controls of the servos is realized by using Linux kernel module—ServoBlaster developed by Richard Hirst [20].

The main unit of computing in the system is the Raspberry Pi computer with the operating system Raspbian. After image acquisition from the camera, follows his processing. In this case, the object detection algorithm was based on the color detection. To the image processing OpenCV library was used.

On the beginning was created a simple model of the object classification. A fast-moving object in red color is classified as an enemy, while the object in the different color is classified as a friend. In addition, an “enemy” object is tracked—through appropriate setting of the servo in the mobile platform with camera. Moving the camera, results in wider field of view.

When an object is classified, a signal is sent to the module of thermal camouflage. Data for the Peltier coolers are sent through RS232 protocol to the appropriate system, which converts the signal to CAN and UART sends to the driver of the cell.

5 Tests

Testing of the demonstrator was based on the recording by camera the red objects. When the object has appeared in the field of view and has been classified as an enemy, active thermal camouflage has been turned on and module with Peltier cooler has been cooled. Figures 4, 5, 6 and 7 show the result of the tests. The difference between the initial value of the temperature and value in 35 s is 20.69 °C, which gives the dynamics of changes in the surface of the thermal camouflage module around 177 °C per 3 s.

Fig. 4 Tracked object

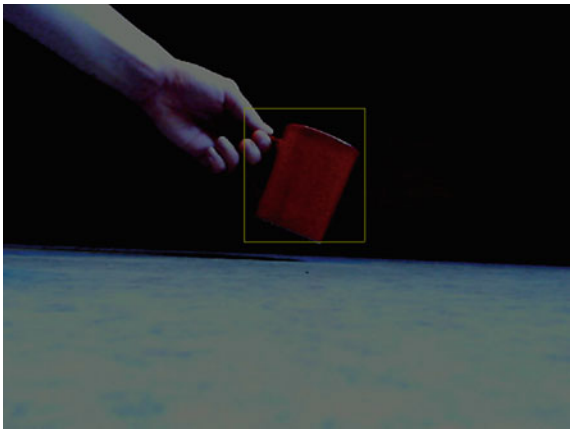
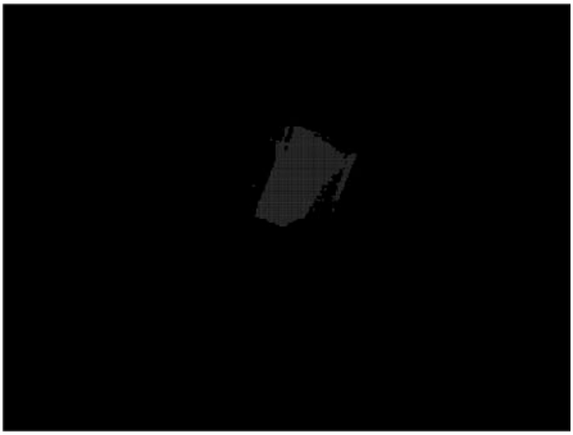


Fig. 5 Camera image after the thresholding operation



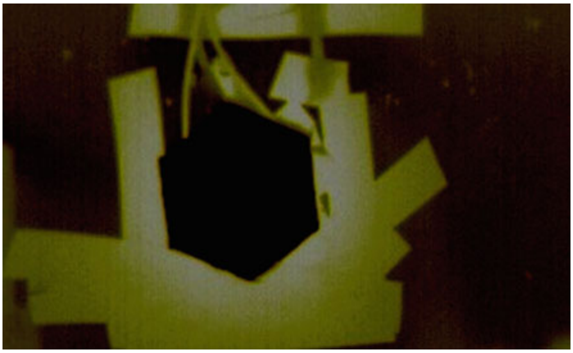


Fig. 6 Thermo camouflage, cooling of a single cell

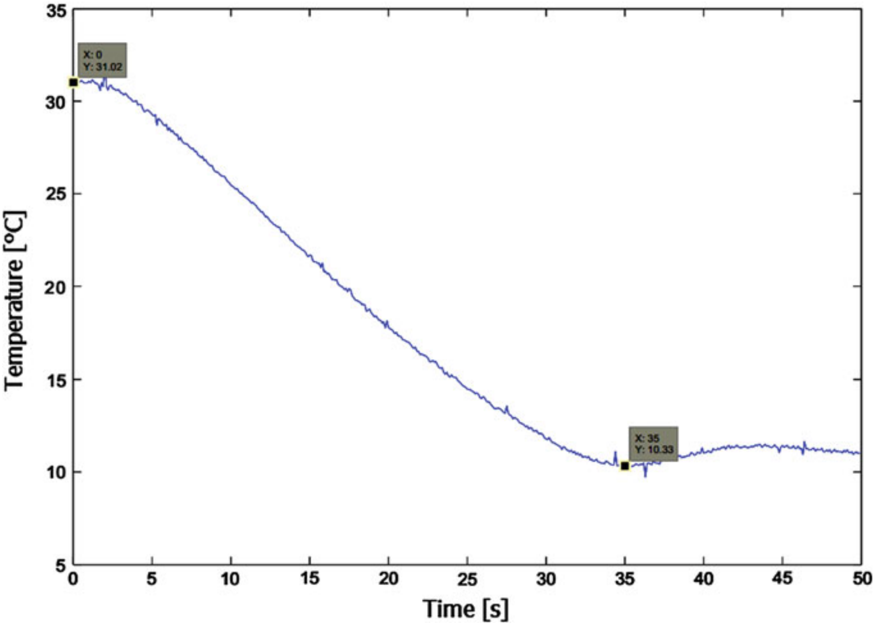


Fig. 7 Dynamics of change of temperature (cooling)

6 Conclusions

The paper presents the concept of using active thermal camouflage with the friend or foe recognition system. To demonstrate the possibilities there was built a simple system with a camera and a Peltier cooler. After testing the system, it turned out that it is not devoid of drawbacks.

The first disadvantage is the delay occurring during image acquisition from the camera. The reason is the low efficiency of computing Raspberry Pi platforms. In order to reduce these delays the camera resolution was reduced to 320 × 240 pixels. Poor performance Raspberry Pi platform also contributed to the extension of time of calculation. Therefore, the detection and tracking object algorithms require optimization.

Module of active thermal camouflage also requires improvements. The system response time is too long and module can't be used in real system. An additional limitation is large power consumption of the Peltier cooler.

Further work on the development of this project will focus on the use of the module in real-time system, improving the recognition system and optimizing tracking algorithms.

References

1. Daniec, K., Iwaneczko, P., Jędrasiak, K., Nawrat, A.: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 219–232 (2013). ISBN: 978-3-319-00368-9
2. Bereska, D., Daniec, K., Jędrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 115–121 (2013). ISBN: 978-3-319-00368-9
3. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)
4. Jędrasiak, K., Bereska, D., Nawrat, A.: The prototype of gyro-stabilized UAV gimbal for day-night surveillance. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 107–115 (2013)
5. Nawrat, A., Jędrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Chen, S. (ed.) Proceedings of WSEAS International Conference on Mathematics and Computers in Science and Engineering, no. 9. World Scientific and Engineering Academy and Society (2009)
6. Jędrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 5337, pp. 391–399 (2009)
7. Grzejszczak, T., Mikulski, M., Szkodny, T., Jędrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 407–413 (2012)
8. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: Design and implementation of mobile ground base station for UGV. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 57–71 (2014)
9. Doton, L.: Integrating Technology to reduce fratricide. Acquisition Rev. Q. (1996)
10. Boyd, C.S., Collyer, R.S., Skinner, D.J., Smeaton, A.E., Wilson, S.A., Krause, D.W., Dexter, R.M., Perry, A.R., Godfrey, J.: Characterisation of Combat Identification Technologies, Lord Bowden of Chesterfield, The story of IFF (identification friend of foe) (1985)
11. Jędrasiak, K., Nawrat, A., Wydmańska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)

12. Internet information: http://www.baesystems.com/image/BAES_019603/innovation-adaptiv-car-signature
13. Ulinowicz, M., Narkiewicz, J.: Modeling and identification of actuator for flap deflection. J. Autom. Mob. Robot. Intell. Syst. **5**, 35–40 (2011)
14. ServoBlaster source code: https://github.com/richardghirst/PiBits/tree/master/ServoBlaster
15. Nawrat, A., Jędrasiak, K.: Fast colour recognition algorithm for robotics. Problemy Eksploatacji, pp. 69–76 (2008)
16. Wochlik, I., Bułka, J., Folwarczny, Ł., Daniec, K., Jędrasiak, K., Koteras, R., Nawrat, A.: Application of telemedical technologies in remote evaluation of soldiers' vital signs during training and in combat conditions. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 189–202 (2014)
17. Antosz, P., Bereska, D., Gatys, K., Niedziela, T., Szota, P.: Multiłącznik IMPRESJA IQ-element instalacji inteligentnego budynku wykorzystującej magistralę CAN, Szybkobieżne, Pojazdy Gąsienicowe, vol. 1, pp. 99–106 (2011)
18. Sobel, D., Kwiatkowski, J., Ryt, A., Domżał, M., Jędrasiak, K., Janik, Ł., Nawrat, A.: Range of motion measurements using motion capture data and augmented reality visualization. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 594–601 (2014)
19. Jedrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 403–408, 19–21 June 2013. ISBN: 978-1-4673-6320-4
20. Nawrat, A., Jędrasiak, K., Daniec, K., Koteras, R.: Inertial Navigation Systems and Its Practical Applications (2012)

Augmented Reality in UAVs Applications

Paweł Iwaneczko, Karol Jędrasiak and Aleksander Nawrat

Abstract We present the example of Attitude Indicator Augmented Reality (AR) control for the unmanned aerial vehicles (UAV) ground control station. The article describes perspective heads-up display overlaid on the image, which is acquired from the UAV rotatable camera. Article shows the mechanisms of aircraft camera calibration with the AR artificial horizon. The whole algorithm of the render instructions is presented. Elements used in the augmented reality are as follows: 3D artificial horizon, latitude and longitude, GPS info, executed command, time to command end, percent of command accomplish, fuel and battery level, height and speed vertical scale, landing field direction arrow, unmanned vehicles marks.

1 Introduction

Nowadays, unmanned aerial vehicles (UAVs) are commonly used by civil and defense industries. More and more applications of UAVs are using cameras to observe terrain in order to allow patrolling and monitoring of the specific areas [1, 2]. Cameras with various parameters and sizes are mounted generally under the airplanes or under the rotorcrafts. These cameras are mainly used for video streaming and obstacles avoidance algorithms. Algorithms such as: obstacles avoidance and collisions avoidance, are widely developed by many organizations, and this means that these algorithms are a key aspect of unmanned aviation [3–6]. Obstacles and collisions avoidance algorithms and problems are described in many articles e.g. [7–11]. As mentioned in [8], the obstacle avoidance of unmanned aerial vehicle in urban environment is the most complex, difficult and essential part of the autonomous flight problems. Collision avoidance is also one of the regulations from ICAO Circular 328 [12].

P. Iwaneczko · K. Jędrasiak (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: Karol.Jedrasiak@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_6

The pilot-in-command of a manned aircraft is responsible for detecting and avoiding potential collisions and other hazards. The same requirement will exist for the remote pilot of an RPA (remotely-piloted aircraft) [13–16]. Technology to provide the remote pilot with sufficient knowledge of the aircraft's environment to fulfil the responsibility has to be incorporated into the aircraft with counterpart components located at the remote pilot station [12]. Remote control is often called —attitude control and it is the one of the basic functions of the mobile ground control station (MGCS) [17]. The MGCS has to be equipped in a specific tools to support pilots during a remote control of the unmanned objects. This paper presents one of the solutions, that can improve the UAV manual control performance, using perspective artificial horizon, telemetry data labels and other UAV objects overlay.

2 Literature Review

In industry, there are already many well-known and commonly used solutions to support pilots during the attitude control of the UAV [18, 19]. One of this solution can be attitude indicator control also known as gyro horizon or artificial horizon control, as illustrated in Fig. 1.

Another solution can be head-up display (HUD), which is a transparent display that presents data without requiring users to look away from their usual viewpoints. As it is described in [23] the origin of the name HUD—stems from a pilot being able to view information with the head positioned "up" and looking forward, instead of angled down looking at lower instruments. It is also called on-screen display (OSD) and usually it is drawn on the camera image. Most of published results of the OSD are a combination of the artificial horizon, generated from non-perspective lines and telemetry data labels. Two examples of OSD are illustrated in Fig. 2. Unfortunately, these solutions are often blurred or even unreadable and sometimes they are not very intuitive. This is because the elements of augmented reality (lines, labels and textures) are usually overlaid on the analog image,

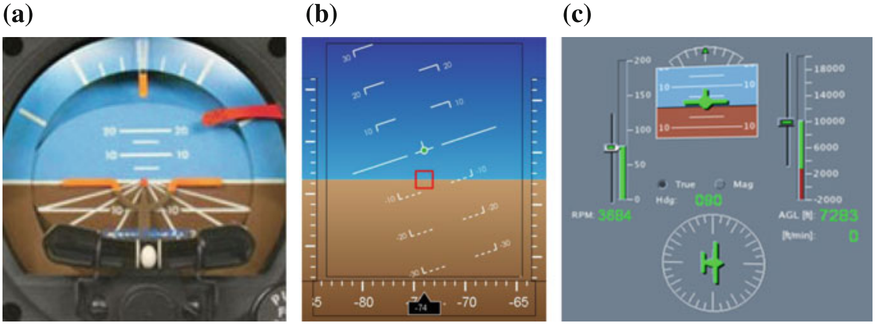


Fig. 1 Examples of artificial horizons. **a** Artificial horizon [20]. **b** Tajfun GCS [21]. **c** VCS-4586 [22]

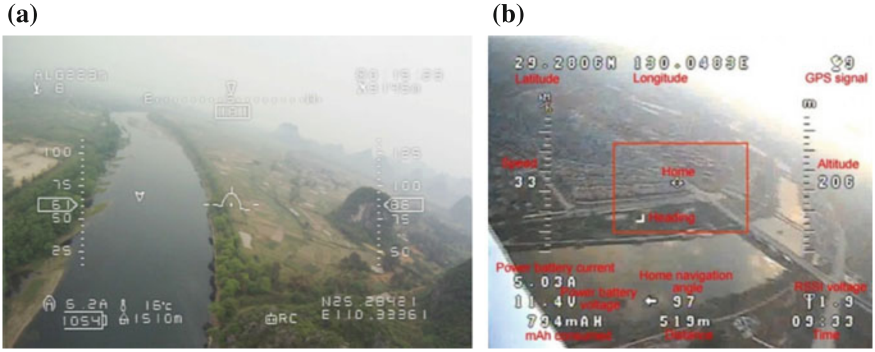


Fig. 2 Examples of camera OSD applications in UAV. **a** FY-DOS OSD [28]. **b** Skylark Tiny OSD III [29]

acquired directly from the camera source, at the aircraft electronics level, and sent to the ground station as a finally generated image [24–27].

Nowadays more and more popular are digital cameras with high resolution, and image processing is currently performed on the ground control station. In this article we present the solution of perspective heads-up display overlaid on the image, that is received from the UAV camera without any changes [30, 31]. A similar solution, where the augmented reality elements are a perspective trajectory tips, is described in [32].

3 Perspective Controls Implementation

Perspective attitude indicator (artificial horizon) is the combination of the three angular scales which are created from three Euler aircraft angles. Aircraft Euler angles are a part of the airplane telemetry, which is received from the UAV by using wireless interfaces. Augmented reality elements, that are used in the proposed AR system, are marked in the Fig. 3 and are described as follows:

1. 3D artificial horizon (3 scales created from yaw, pitch and roll angles)
2. Aircraft telemetry data:
    - Latitude and Longitude (degrees),
    - GPS information (number of satellites and GPS Fix information),
    - Executed command (waypoint, taking off, landing, attitude flight or loiter),
    - TTE—time to end of the executed command (hours, minutes and seconds),
    - POA—percent of accomplish of the executed command,
    - Battery level (percent),
3. Height vertical scale (meters),
4. Speed vertical scale (meters per second),
5. Landing field direction arrow,
6. Other unmanned aerial vehicles marks and icons.

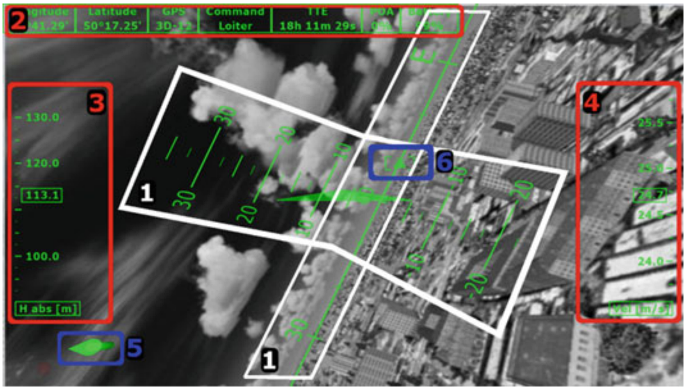


Fig. 3 Parts of the augmented reality system—marked with numbers *1–6*

To implement augmented reality 3D controls overlaid on the camera image, first of all we have to set our 3D environment as a normalized device (set projection and model matrix to identity). After doing it, we have to set the perspective model view to the same parameters (vertical field of view and ratio) as in the camera specification. Then we have to rotate local coordinate system of graphical environment to airplane coordinate system (Eq. 1).

$$M_{GL} = M_{GL} * Rot_Y(90^\circ) * Ror_X(-90^\circ) \tag{1}$$

where: M_{GL}—graphical model matrix (load after each substitution), Rot_Y, Rot_X—rotation around x and y axis.

At the second step, it will be rendered landing field direction arrow (Eq. 2). To make this possible we have to calculate ENU navigation coordinates from the difference between two geographical $\varphi\lambda h$ position. Full algorithm of the conversion between the GPS coordinates and ENU navigation coordinates is presented in [33].

$$\begin{aligned}
M_{temp} &= M_{GL} \\
x_b &= \mathfrak{B} * \cos(\gamma * 0.35) * \cos(\beta * 0.35) \\
y_b &= \mathfrak{B} * \cos(\gamma * 0.35) * \sin(\beta * 0.35) \\
z_b &= \mathfrak{B} * \cos(\gamma * 0.35) * \sin(-\gamma * 0.35) \\
\varphi_b &= \tan^{-1}\frac{e_b}{n_b}, \theta_b = \tan^{-1}\frac{u_b}{\sqrt{e_b^2 + n_b^2}} \\
M_{GL} &= M_{GL} * Trans(x_b, y_b, z_b) * Rot_X(\psi) \\
&\quad * Ror_Y(\theta + \theta_b) * Ror_z(\varphi - \varphi_b - 90) \\
&\text{“\textit{landing field direction arrow rendering instructions}”} \\
M_{GL} &= M_{temp}
\end{aligned} \tag{2}$$

where: M_{temp}—temporary matrix, x_b, y_b, z_b—calculated arrow position, $\mathfrak{B}$—Arrow distance constant, β, γ—camera horizontal and vertical field of view, φ_b, θ_b—landing field yaw and pitch angles values [rad], e_b, n_b, u_b—landing field navigation coordinates [m], φ, θ, ψ—unmanned aerial vehicle yaw, pitch and roll angles values [rad].

Now we have to undo the airplane camera rotation, and rotate the matrix model using the UAV yaw, pitch and roll angles (Eq. 3). When there will be also added some translation in Z axis our system will be almost calibrated with the camera video stream.

$$\begin{aligned}
M_{GL} &= M_{GL} * Rot_Z(-\varphi_{cam}) * Ror_Y(-\theta_{cam}) * Ror_X(\psi_{cam}) \\
M_{temp} &= M_{GL} \\
h_{offs} &= -\mathfrak{H} * \tan\left(\cos^{-1}\frac{\Re}{\Re + h}\right) \\
M_{GL} &= M_{GL} * Rot_X(-\psi) * Ror_Y(\theta) * Ror_Z(\varphi) * Trans\big(0, 0, h_{offs}\big)
\end{aligned} \tag{3}$$

where: $\varphi_{cam}, \theta_{cam}, \psi_{cam}$—UAV camera orientation [rad], h,h_{offs}—UAV absolute height and height correction value, $\mathfrak{H}$—horizon distance constant, $\Re$—earth radius constant. After this, we can render other UAVs connected to the MGBS (Eq. 4).

$$\begin{aligned}
&M_{temp1} = M_{GL} \\
&M_{GL} = M_{GL} * Trans(n_o, -e_o, u_o) * Rot_Z(-\varphi) * Ror_Y(-\theta) * Ror_X(\psi) \\
&\text{“}\textit{UAV object rendering instructions}\text{”} \\
&M_{GL} = M_{temp1}
\end{aligned} \tag{4}$$

where: e_o, n_o, u_o—UAV navigation coordinates [m], M_{temp1}—another temporary matrix. At this point we can render horizon line and pitch scale lines. Horizon line is drawn using 36 lines, which are rendered on a plane tangent to the XY axis as a circle with a radius of $\mathfrak{H}$. Depending on the visible horizontal field of view of the aircraft camera, there are rendered textural labels such as: North, East, South and West letters and the middle numerical values. Lines of the pitch scale are rendered from the center of the aircraft beak with the saturation between the $\theta_{\min}$ and $\theta_{\max}$ values (Eq. 5).

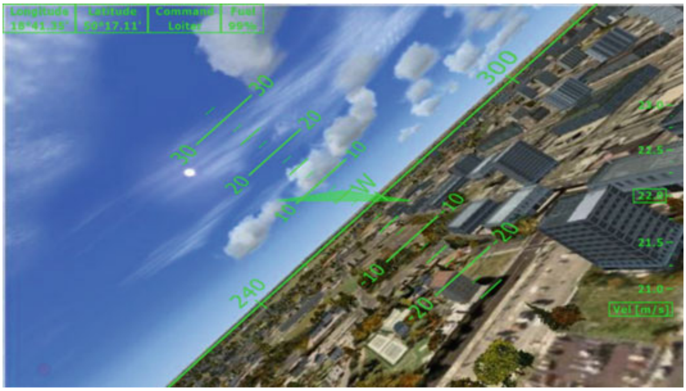
$$\begin{aligned}
&\theta_{\min} = \theta - \theta_{cam} - \gamma * 0.4 \\
&\theta_{\max} = \theta - \theta_{cam} + \gamma * 0.4 \\
&M_{GL} = M_{GL} * Rot_Z(-\varphi) \\
&\left\{\begin{array}{c}
for(i = -9; i \leq 9; i++) \\
\left\{\begin{array}{c}
if(i \neq 0 \cap i * 10 \geq \theta_{\min} \cap i * 10 \leq \theta_{\max}) \\
\left\{\begin{array}{c}
M_{temp1} = M_{GL} \\
M_{GL} = M_{GL} * Rot_Y(0, -i * 10, 0) \\
\text{“}\textit{pitch line and labels rendering instructions}\text{”} \\
M_{GL} = M_{temp1}
\end{array}\right.
\end{array}\right.
\end{array}\right.
\end{aligned} \tag{5}$$

where: $\theta_{\min}$, $\theta_{\max}$—minimum and maximum pitch saturation values, i—iterator variable. At the end of the perspective artificial horizon render algorithm we have to draw the plane triangle marker (Eq. 6).

$$\begin{gathered} M_{GL} = M_{temp} \\ \text{``}\textit{airplane model rendering instructions}\text{''} \end{gathered} \tag{6}$$

4 Project Results

This paper shows the solution of attitude indicator AR control for the UAV GCS. It was implemented using OpenGL in C++ language. We were using Windows 8.1 hardware platform. All of telemetry data and video stream come from Lockheed's Martin Prepar3D® simulation environment and it was tested only in SIL (Software In the Loop) simulation [34]. System is configurable from the user interface, as it is presented in the Fig. 4. User can choose which part of the control will be visible overlaid on the image, including the image type (infrared or colored image). GCS operator can toggle all of the elements that are specified in Sect. 3. points 2–6. This article shows two examples of screenshot sequences of ground control station (GCS). Analysing the images in the Fig. 5, it can be concluded, that green horizon line is synchronized with the Prepar3D® simulation earth/sky horizon line. You may also notice a large motion dynamic of the unmanned object and stability of the AR attitude indicator control. The first sequence describes the situation, when the aircraft is preparing to execute the loiter command. The next sequence of images (Fig. 6), also shows, that artificial horizon is correctly calibrated with the camera image. It shows that presented solution it is very stable, while the UAV is executing the loiter command.



Longitude	Latitude	Command	Fuel
18°41.35'	50°17.11'	Loiter	99%

Fig. 4 Possible configuration of the augmented camera graphical control

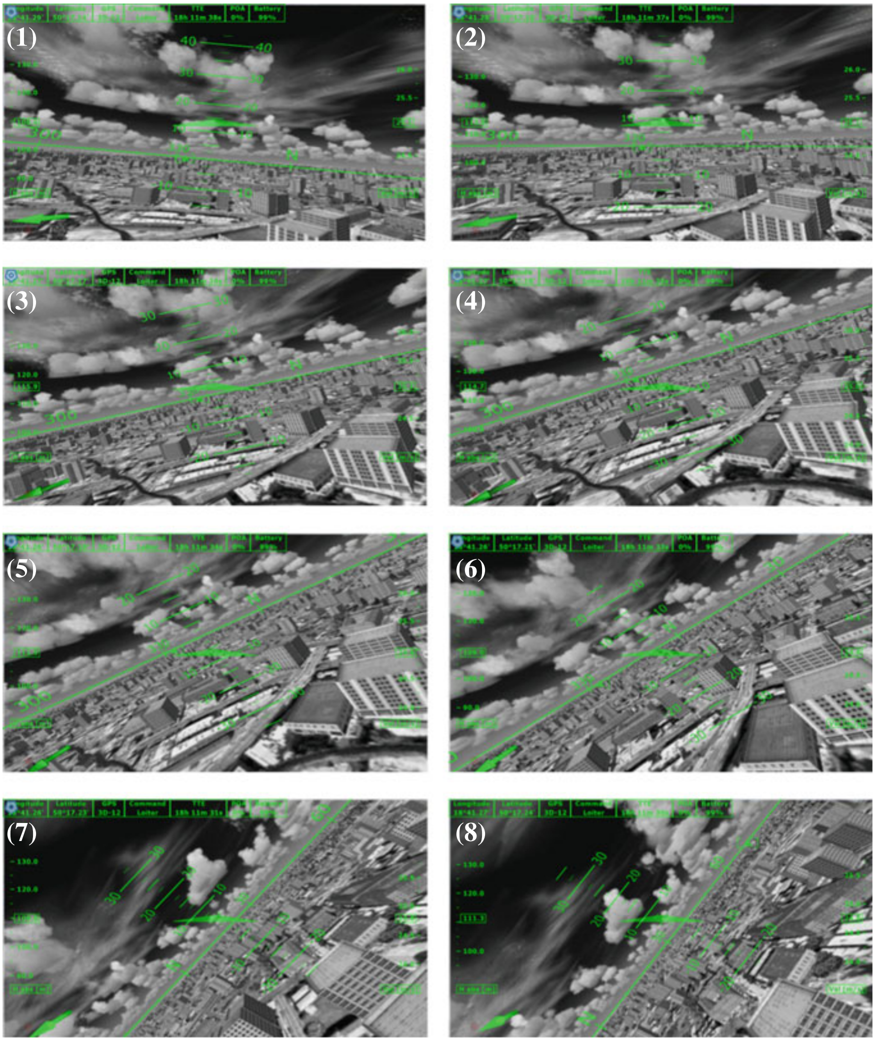


Fig. 5 A first images sequence of augmented reality in UAV. *1* Frame no. 1142 (Time: 19 s), *2* Frame no. 1202 (Time: 20 s), *3* Frame no. 1263 (Time: 21 s), *4* Frame no. 1321 (Time: 22 s), *5* Frame no. 1383 (Time: 23 s), *6* Frame no. 1442 (Time: 24 s), *7* Frame no. 1564 (Time: 26 s), *8* Frame no. 1623 (Time: 27 s)

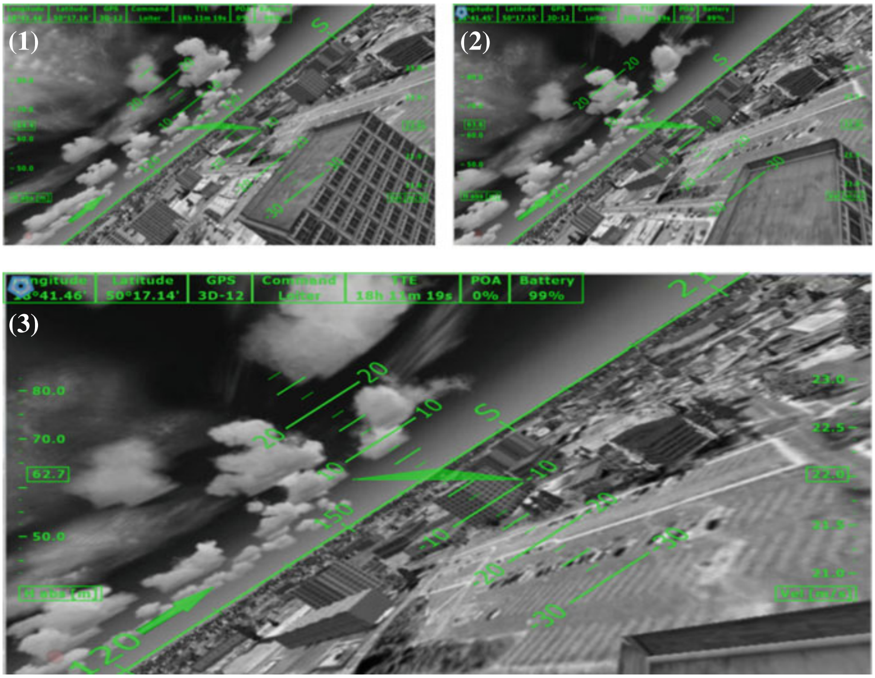


Fig. 6 A second images sequence of augmented reality in UAV. *1* Frame no. 3301 (Time: 55 s), *2* Frame no. 3361 (Time: 56 s), *3* Frame no. 3422 (Time: 57 s)

5 Conclusions

It has been shown, that the aircraft camera video stream can be calibrated with the 3D viewport, which can be used as augmented reality. Perspective heads-up display, which is overlaid on the image acquired from the unmanned aerial vehicle gimbal camera, can be used in UAV ground control stations to improve the UAV manual control performance.

In the future we are planning to implement an user interface, which will provide possibility of the camera orientation control from the MGBS application. Another of the ideas and development of this system is to perform full test of augmented reality artificial horizon with real unmanned aerial vehicles in urban and difficult terrain.

Acknowledgment This work has been supported by Applied Research Programme of the National Centre for Research and Development as a project ID 178438 path A—Costume for acquisition of human movement based on IMU sensors with collection, visualization and data analysis software.

References

1. Jędrasiak, K., Bereska, D., Nawrat, A.: The prototype of gyro-stabilized uav gimbal for day-night surveillance. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 107–115 (2013)
2. Gałuszka, A., Skrzypczyk, K., Bereska, D., Pacholczyk, M.: Re-handling operations in small container terminal operated by reach stackers. World Acad. Sci. Eng. Technol. **70**, 674–677 (2010)
3. Niezgoda, T., Panowicz, R., Sybilski, K., Barnat, W.: Numerical analysis of missile impact being shot by rocket propelled grenades with rod amour. WIT Trans. Model. Simul. **51**, 625–633 (2011)
4. Kuś, Z., Nawrat, A.M.: The limitation for the angular velocity of the camera head during object tracking with the use of the UAV. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 127–145 (2014)
5. Kuś, Z., Nawrat, A.M.: Camera head control system with a changeable gain in a proportional regulator for object tracking. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 105–125 (2014)
6. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 157–175 (2013) (ISBN: 978-3-319-00368-9)
7. Wang, X., Vivek, Y., Balakrishnan, S.N.: Cooperative UAV formation flying with obstacle/collision avoidance. IEEE Trans. Control Syst. Technol. **15**(4) (July 2007)
8. Kownacki, C.: Obstacle avoidance strategy for micro aerial vehicle. In: Advances in Aerospace Guidance, Navigation and Control, pp. 117–135 (2011)
9. Kownacki, C.: Control algorithm of micro aerial vehicle flight in streets' canyons based on vision system, Faculty of Mechanical Engineering, Białystok Technical University, ul. Wiejska 45 C, 15–351 Białystok
10. McGee, T.G.: Obstacle detection for small autonomous aircraft using sky segmentation. In: Robotics and Automation ICRA 2005 (2005)
11. Frew, E.: Vision-based road following using a small autonomous aircraft. In: Aerospace Conference (2004)
12. International Civil Aviation Organization Cir 328, "Unmanned Aircraft Systems" (UAS), http://www.icao.int/Meetings/UAS/Documents/Circular%20328_en.pdf, 04.2014
13. Niezgoda, T., Barnat, W.: Analysis of protective structures made of various composite materials subjected to impact. Mater. Sci. Eng., A **483**, 705–707 (2008)
14. Bibik, P., Zasuwa, M., Zugaj, M.: Research and training symulator of unmanned quadrotor. In: 18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 403–407 (2013)
15. Bereska, D., Balcewicz, R., Garczyński, M.: Implementacja magistrali CAN i protokołu CANopen w robocie edukacyjnym. Szybkobieżne Pojazdy Gąsienicowe **1**, 157–162 (2008)
16. Czapla, T., Wrona, J.: Technology development of military applications of unmanned ground vehicles. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 293–309 (2013) (ISBN: 978-3-319-00368-9)
17. Zeng, J.: System framework and standards of ground control station of unmanned aircraft system. Electr. Eng. Control, LNEE 98, 327–334
18. Nawrat, A., Jędrasiak, K., Daniec, K., Koteras, R.: Inertial Navigation Systems and Its Practical Applications (2012)
19. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: Design and Implementation of mobile ground base station for UGV. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 57–71 (2014)
20. Artificial Horizons from Castleberry Instruments, Falcon Gauge, RC Allen and Trutrak Flight Systems. http://sarasotaavionics.com/category/flight-instruments/artificial-horizon, 04.2014

21. ESC Aerospace, RPAS/UAS GCS (GROUND CONTROL SYSTEM) “TAJFUN”, http://www.esc-aerospace.com/eshop/index.php?controller=attachment&id_attachment=3, 04.2014
22. Lockheed Martin, VCS-4586 Ground Control Operator Software for Unmanned Vehicle Systems, http://www.lockheedmartin.com/content/dam/lockheed/data/ms2/documents/cdl-systems/VCS-4586%20CAPABILITIES%20GUIDE-August2013.pdf 04.2014
23. Stevenson, A.: Oxford Dictionary of English. In: Heads-Up Display. Oxford University Press, Oxford, page 809 (2010)
24. Jędrasiak, K., Daniec, K., Nawrat, A., Koteras, R.: Wykorzystanie kamer termowizyjnych w systemach dozoru wizyjnego infrastruktury krytycznej sieci dystrybucyjnych gazu. Przegląd Elektrotechniczny **88**, 90–97 (2012)
25. Ryt, A., Sobel, D., Kwiatkowski, J., Domzal, M., Jedrasiak, K., Nawrat, A.: Real-time laser point tracking. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671. pp. 542–551 (2014)
26. Daniec, K., Iwaneczko, P., Jędrasiak, K., Nawrat, A.: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 219–232 (2013) (ISBN: 978-3-319-00368-9)
27. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: The Method for Long-Term Object Tracking. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)
28. FY-DOS OSD, est_fpv_and_uav_set_for_rc_air_plane_FY_OSD.html, 04.2014
29. Skylark Tiny OSD III, http://www.nem.gr/skylark-tiny-osd-iii-set-p33842.html
30. Gałuszka, A., Pacholczyk, M., Bereska, D., Skrzypczyk, K.: Planning as artificial intelligence problem—short introduction and overview. In: Advanced Technologies for Intelligent Systems for National Border Security, pp. 95–103 (2013)
31. Antosz, P., Bereska, D., Gatys, K., Niedziela, T., Szota, P.: Multiłącznik IMPRESJA IQ-element instalacji inteligentnego budynku wykorzystującej magistralę CAN. Szybkobieżne Pojazdy Gąsienicowe **1**, 99–106 (2011)
32. Tadema, J., Theunissena, E., Koeners, J.: Using perspective guidance overlay to improve UAV manual control performance. In: Enhanced and Synthetic Vision 2007, vol. 6559, Orlando, Florida, USA, 09 Apr 2007
33. Drake, S.P.: “Converting GPS Coordinates (φλh) to Navigation Coordinates (ENU)”, http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/3538/1/DSTO-TN-0432.pdf, Surveillance Systems Division Electronics and Surveillance Research Laboratory
34. Iwaneczko, P., Jędrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics Lecture Notes in Computer Science, vol. 7594, pp. 87–94 (2012)

Real Time Dense Motion Estimation Using FPGA Based Omnidirectional Video Acquisition Device

Jan Kwiatkowski, Dawid Sobel, Artur Ryt, Mariusz Domżał, Karol Jędrasiak and Aleksander Nawrat

Abstract OVAD (Fraś et al. in Vision based systems for UAV applications, Springer International Publishing, pp. 123–136 [1]; Kwiatkowski et al. in Recent advances in electrical engineering and computer science, pp. 58–61 [2]) is a device, which is purposed for multi directional video acquisition. Further development of the device includes hardware implementation of time-absorbing calculations, connected with video stream processing, with utilization of FPGA (Field Programmable Gate Array). The paper presents the results of research on the problem of real time dense optical flow estimation. Furthermore, the described results are connected to the problem of parallel processing and hardware acceleration using FPGA. Proposed solution may be found as useful in many applications, both civilian and military and proves utility of FPGA based solutions in real time video processing.

Keywords Omnidirectional camera · Image processing · FPGA · Optical flow · Dense motion estimation · Hardware acceleration

1 Introduction

Nowadays, video surveillance is often a very important part of a lot of facilities, such as petrol stations, banks or houses. There are many places, which, due to the need of ensuring an adequate level of security, have to be surveilled, to a greater or lesser extent. When observation of an area is impossible to be conducted by a man or a need of continuous observation exists, video cameras are used. Acquired video stream can be processed in many ways, from simple compression and transmission over long distances to advanced analysis of acquired video stream, such as motion estimation or object recognition and tracking [3–5]. Moreover, provision of effec-

J. Kwiatkowski (✉) · D. Sobel · A. Ryt · M. Domżał · K. Jędrasiak · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland
e-mail: jan.kwiatkowski@polsl.pl

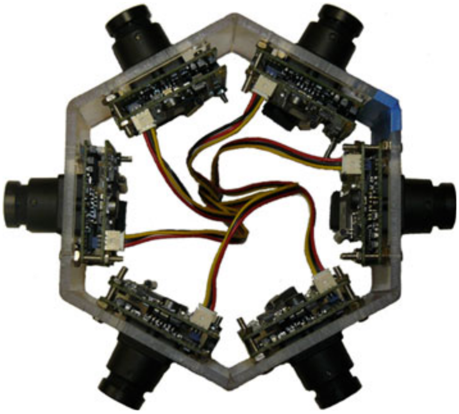
A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_7

tive, advanced and robust vision systems is important not only in civilian applications, but also in military solutions. Among purposes of those applications two main groups can be distinguished: surveillance of facilities and observation of surroundings of vehicles, especially unmanned vehicles and unmanned aerial vehicles. The last group of vehicles imposes an another assumption, which forces solutions to be small-sized and computer independent.

Another problem, connected with vision based surveillance systems, is provision of wide field of view of the system, to cover the entire surveilled area. There are three ways of increasing of the area covered by the system: dedicated lenses, such as fisheye lenses, servomotors, as in PTZ cameras or multiplication of cameras, as in FPGA based OVAD [2]. The Omnidirectional Video Acquisition Device (Fig. 1) consists of multiple cameras placed in a way, that proves 360° of horizontal field of view. The advantages of the device over fisheye lenses and PTZ cameras are: less distortions than in fisheye cameras, no moving elements, which are susceptible to environmental conditions, multiplication of resolution of acquired video stream and redundancy of video recorders, which improves robustness of the solution [6].

OVAD consists of multiple digital cameras and FPGA, which task is to maintain the process of parallel conditioning and recording of video signal, acquired from multiple cameras. Through utilization of FPGA, the device is computer-independent, while maintaining small size, which allows OVAD to be used in mobile applications such as unmanned aerial vehicles. The use of FPGA enabled another possibility for further development, which is implementation and hardware acceleration of algorithms used in video processing [7–10]. FPGA programmable logic device can be used in preprocessing of the acquired data, before it is transmitted to receiver. In a lot of human independent, autonomous applications, the video system's ability of real time data analysis is strongly required. However, processing such a lot of data in real time, determined by the frame rate and resolution of used camera, is most of the times hard to maintain, even with utilization of highly advanced computer systems. Solution to that problem is parallel processing and hardware acceleration through utilization of dedicated FPGA implementation [11].

Fig. 1 Omnidirectional video acquisition device [1]



Ability of motion detection is often desired in video surveillance. Taking under consideration, that there are no known implementations of real-time, large resolution dense motion estimation algorithms, with utilization of FPGA [12], the first step of further improvement of FPGA based OVAD is hardware implementation of real time dense optical flow estimation algorithm, which is used to specify the movement of all objects in acquired video stream [13]. Moreover, the information regarding movement in acquired video stream is often more important, than acquired video itself, and can be used by many algorithms dedicated for object recognition or tracking of the captured objects [14, 15].

2 Field Programmable Gate Array (FPGA)

Described device is based on Field Programmable Gate Array. FPGAs are capable of parallel processing, creating a possibility of processing of large amounts of data, while whole system is being clocked at lower frequencies. The basic units, of which FPGA consists, are CLBs (ang. Configurable Logic Block). Their task is to realize simple functions such as flip-flops, multiplexers or logic gates. Each CLB is based on Look-Up Table (LUT). CLB block is configured via filling LUTs. LUT operation is based on the selection of one of predefined output for each of the possible combination of inputs. This simplifies calculations, which can be performed in one clock cycle. Sample LUT configurations are shown in Fig. 2. The Configurable Logic Blocks are connected through a network of buses operated by switching circuits. Adjacent blocks are connected directly, while some of them are also connected to the global, high-speed interconnection buses. Thanks to that solution, CLB and switching systems can be configured in such a way, that they realize

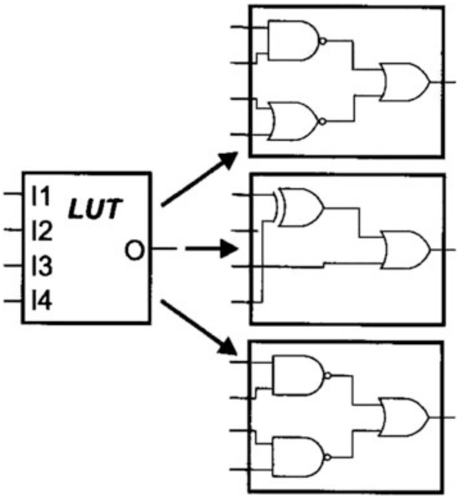


Fig. 2 Examples of LUT configuration [17]

completely independent tasks, in parallel, as shown in Fig. 3. In addition, each Configurable Logic Block can be triggered by any of FPGA inputs, which means that FPGA supports asynchronous processing [16].

What is worth mentioning, is that, despite the advantages associated with asynchronous and parallel processing, FPGAs are not devoid of drawbacks. The first of these is the use of Static Random Access Memory (SRAM) as the configuration memory. This type of memory resets every time it is switched off and back on, which means FPGAs need to be configured every time its power is switched on. That creates necessity of utilization of external memory, to store the configuration data. Moreover, because of being based on Look-Up Tables, FPGA programmable logic devices are highly inefficient in terms of memory storage. A workaround to

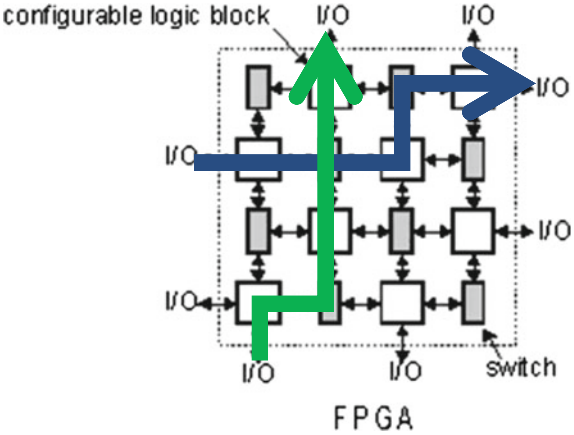


Fig. 3 Example of parallel CLB routing [18]

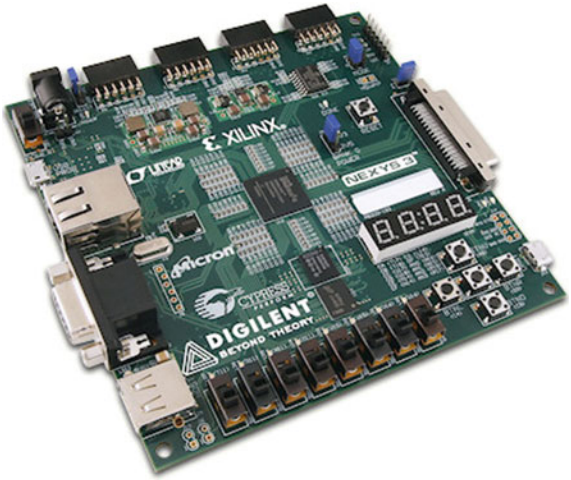


Fig. 4 Digilent Nexys3 FPGA Board [20]

that disadvantage is utilization of dedicated dual-ported memory, called BlockRAM, which is still not capable of storing the entire VGA video frame [17].

Used FPGA device is Xilinx Spartan-6 [19], available on Digilent Nexys3 FPGA board (Fig. 4) [20]. The FPGA unit is supplied by 100 MHz oscillator, however, the frequency can be increased by using DCM and PLL, up to frequencies greater than 500 MHz. Nexys3 also includes 3 external memories, of which one is used as external storage for sequence of video frames, used to compute dense optical flow [21].

3 Dense Optical Flow

The main objective of conducted research was implementation of dense optical flow computing algorithm. The concept behind the determination optical flow, is computation of vector field, containing information on apparent motion of light patterns in consecutive frames of video stream. The concept is related to human way of perception of their surroundings. Human perception of moving objects is based on analysis of the entire registered image, precisely on determination of displacement of all individual objects, captured by an observer's eye. Thanks to the eye-brain system, a person is able to capture, extract and recognize multiple objects moving in different directions.

The history of optical flow dates back to the 1970s. Since then, many papers, considering the subject, have been published, and a number of methods, designed for the problem solution, have been proposed. Taking under consideration the general principle, on which each method is based, considered solutions can be divided into two basic groups: correlation methods, gradient methods [22].

Every considered dense optical flow estimation algorithm is embodiment of an attempt to answer the question: "If there is a change in between consecutive frames of captured video stream, then what is the value of vector describing movement of every spot in the image?". The answer to this question is field of velocity vectors, binding consecutive frames, and that information can be used to transform one frame into another, subsequent frame (Fig. 5) [23].

The variety of methods were examined for their suitability to be used in the described implementation. The most important (and the most difficult to realize) assumption of the implementation, is achieving real-time performance of implemented method. This assumption makes global and gradient methods practically impossible to be implemented, taking under consideration the quality of electronics used in the solution. This is due to the fact, that for effective implementation of gradient methods, such as Lucas-Kanade [22], more than three video frames have to be analyzed, to accurately calculate the partial derivative with respect to time. The external memory, used in the project, does not provide appropriately fast access to

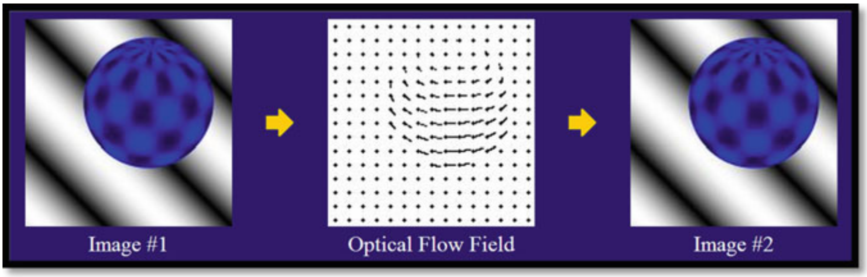


Fig. 5 An illustration of definition of dense optical flow [24]

the selectively chosen data, and such fast response is a must, given the inability of FPGA, to store large blocks of data, such as video frame.

3.1 Selection of Hardware Implemented Methods

The solution had to rely on relatively simple methodology of optical flow determination. Such a low computational complexity is provided by the correlation methods, a local methods, based on searching for similarities between fragments of consecutive frames. Due to its insensitivity to changes in brightness between subsequent frames, preferred method seems to be the normalized cross-correlation method (1) [22], which bases on estimation of Pearson's correlation coefficient (2) between two series of data (2).

$$\boldsymbol{U} = arg\max_{\boldsymbol{u}}(E(\boldsymbol{u})) = \left\{\boldsymbol{u}|\bigwedge_{\boldsymbol{y}\in A} E_{NCC}(\boldsymbol{u}) \le E_{NCC}(\boldsymbol{y})\right\}. \tag{1}$$

$$E_{NCC}(\boldsymbol{u}) = \frac{\sum_i [I_0(x_i) - \bar{I}_0][I_1(x_i + \boldsymbol{u}) - \bar{I}_1]}{\sqrt{\sum_i [I_0(x_i) - \bar{I}_0]^2}\sqrt{\sum_i [I_1(x_i + \boldsymbol{u}) - \bar{I}_1]^2}}, \tag{2}$$

where:$\bar{I}_0 = \frac{1}{N}\sum_i I_0(x_i), \bar{I}_1 = \frac{1}{N}\sum_i I_1(x_i + \boldsymbol{u})$, **U**—set of solutions, $\mathbf{x_i}$—coordinates of examined pixel, **u** = (**u**, **v**)—shift of pixel coordinates, $\mathbf{I_0}$—previous frame, $\mathbf{I_1}$—present frame, **i**—pixel index.

However, the RAM, available on Nexys3 development board is incapable of achieving the appropriate speed of downloading of two compared blocks of data, from both the current and the previous frame image. Taking under consideration the hardware limitations, perfect method for real-time determination of dense optical flow would be a method, that determines the optical flow of the appearing new pixel, obtained by the camera or image generator. Therefore, it has been decided, that implemented method will not be based on a comparison of two blocks, but a comparison of the new pixel value and block around the corresponding pixel, acquired from the previous frame, containing the equivalent of the analyzed pixel environment. This solution increases the susceptibility to noise, but, given the equipment incapability, has proven to be achievable. Under these assumptions two methods of determining optical flow were proposed:

- M1 method—if change of the value of examined pixel is detected (in comparison to previous value), a pixels, which have the same value, are searched for in a block (3), (4), formed around the pixel in the previous frame and located on the same coordinates. If at least one pixel, which has the same value as the tested pixel, is found in the neighborhood, the smallest difference between pixels' coordinates is information about the optical flow.

$$\boldsymbol{U} = \left\{ \boldsymbol{u} | \bigwedge_{\boldsymbol{y} \in A} E_{M1}(\boldsymbol{u}) = 0 \right\} \tag{3}$$

$$E_{M1}(\boldsymbol{u}) = I_0(\boldsymbol{x} + \boldsymbol{u}) - I_1(\boldsymbol{x})(14), \tag{4}$$

where:**U**—set of solutions, $\mathbf{x_i}$—coordinates of examined pixel, **u = (u, v)**—shift of pixel coordinates, $\mathbf{I_0}$—previous frame, $\mathbf{I_1}$—present frame.

- M2 method—if change of the value of examined pixel is detected (in comparison to previous value), a pixels, which have the most similar value, are searched for in a block (5), (6), formed around the pixel in the previous frame and located on the same coordinates. The smallest difference between pixels' coordinates is information about the optical flow. The criterion, by which the pixels are compared, is the sum of the differences between the R, G and B components of each pixel value.

$$\boldsymbol{U} = arg \min_{\boldsymbol{u}}(E_{M2}(\boldsymbol{u})) = \left\{ \boldsymbol{u} | \bigwedge_{\boldsymbol{y} \in A} E_{M2}(\boldsymbol{u}) \le E_{M2}(\boldsymbol{y}) \right\} \tag{5}$$

$$\begin{aligned} E_{M2}(\boldsymbol{u}) = I_{0R}(\boldsymbol{x} + \boldsymbol{u}) - I_{1R}(\boldsymbol{x}) + I_{0G}(\boldsymbol{x} + \boldsymbol{u}) - I_{1G}(\boldsymbol{x}) \\ + I_{0B}(\boldsymbol{x} + \boldsymbol{u}) - I_{1B}(\boldsymbol{x}) \end{aligned} \tag{6}$$

U—set of solutions, $\mathbf{x_i}$—coordinates of examined pixel, **u** = (**u**, **v**)—shift of pixel coordinates, $\mathbf{I_0}$—previous frame, $\mathbf{I_1}$—present frame.

3.2 Lucas-Kanade Method

For comparison purposes, during testing of selected methods with utilization of a computer, the Lucas-Kanade method was implemented. Choice of the method is indicated by its popularity in known implementations, low complexity, and the fact that it is a local method (as well as proposed ones). Assumptions made by the authors of the method are defined as follows: in two consecutive frames light pattern movement is so slight that it is not bigger than assumed window size. Furthermore, the second assumption of the method is locally uniform, linear movement of all the dots within the window. That assumptions are fulfilled when time delay between consecutive frames is relatively small. The Lucas-Kanade method is described as follows:

Let $\boldsymbol{u} = [uv]^T$ will be the velocity vector determining shift of the spots in image. The pixel value I, which is considered as a function of time and image coordinates, should be identical to the pixel from second frame, after time and coordinates shift:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{7}$$

To approximate (7) with linear function, Taylor series first order expansion can be used as follows:

$$\begin{aligned} I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x \\ + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T. \end{aligned} \tag{8}$$

In linear approximation high order terms can be assumed as equal to 0. Combination of (7) and (8) is:

$$0 = \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{9}$$

In this case (9), assuming a locally uniform shift of all the pixels [3]:

$$I_x(q_i) * u + I_y(q_i) * v = -I_t(q_i), \tag{10}$$

where: $\mathbf{I_x}(\mathbf{q_i})$—directional gradient along the x-axis in point q_i, $\mathbf{I_y}(\mathbf{q_i})$—directional gradient along the y-axis in point q_i, $\mathbf{I_t}(\mathbf{q_i})$—directional gradient along the t-axis in point q_i, $\mathbf{q_i}$—coordinates of the examined pixel.

Obtained linear system (10) is overdefined in most of the cases and can be solved as follows:

$$A\boldsymbol{v} = b, \text{ where: } A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ \ldots & \ldots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, b = \begin{bmatrix} -I_t(q_1) \\ \ldots \\ -I_t(q_n) \end{bmatrix} \tag{11}$$

$$A^T A\boldsymbol{u} = A^T b \tag{12}$$

$$\boldsymbol{u} = (A^T A)^{-1} A^T b \tag{13}$$

The resulting vector (13) is information about the speed of movement of light patterns in direction of both image axes, and thus information on the optical flow in examined video sequence.

The disadvantage of gradient methods is the need for accurate computation of partial derivatives of a frame, which imposes performing calculations after acquisition of a sequence of several images is completed. Only then it is possible to compute the gradient with respect to all three parameters of pixel function and numerically computed partial derivatives with respect to time are sufficiently accurate, and noise-independent.

Further development of the method is computing and merging optical flow of each of subsequent levels of Gaussian pyramid (Fig. 6) of frame, to improve the quality of gained results gained from Lucas-Kanade method. The reduction of image resolution allows inclusion of a larger, averaged area within a window and thus prevents the algorithm results from focusing around the high frequencies in the image (Fig. 7). The disadvantage of this method is significant extension of the

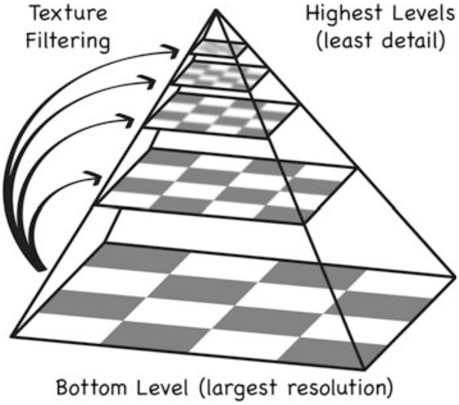


Fig. 6 Gaussian pyramid [25]

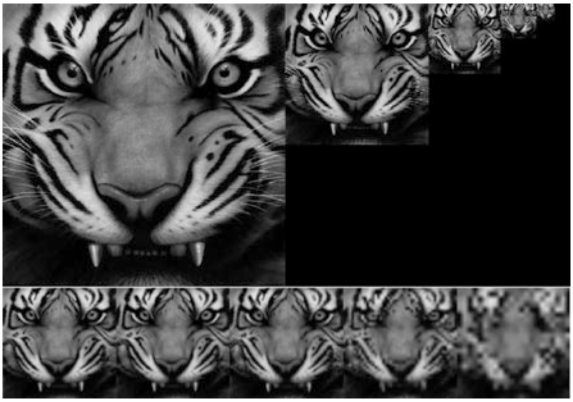


Fig. 7 Gaussian pyramid of an image [26]

computation time. The method, called Hierarchical Lucas-Kanade, has also been implemented as part of the method comparison application, as a mean of determination of the quality of other methods.

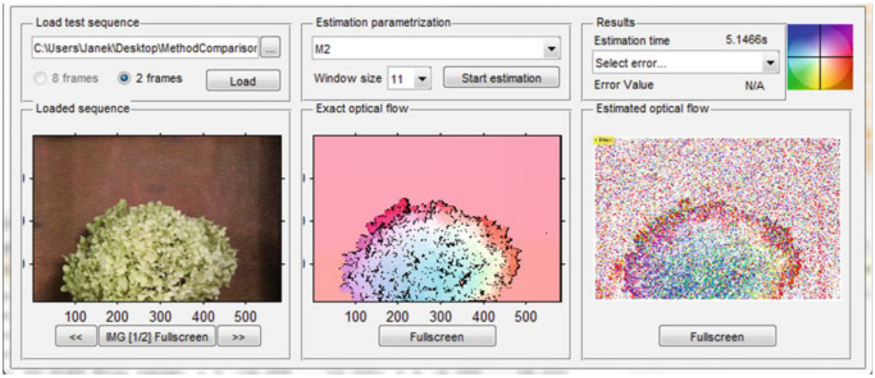
4 Implementation

4.1 Computer Program—Method Comparison

For the purpose of comparison of considered methods, an computer application (Fig. 8), designed for multi-method dense optical flow estimation, was created. Calculations were performed on a computer and obtained results allowed to compare effectiveness of the described and proposed methods, by comparison of numerical and visual results, obtained in the form of optical flow. The results of the comparison are presented in Sect. 5.

4.2 Configuration of FPGA

Both proposed methods were implemented in FPGA. However, the whole System on a Chip implementation includes not only a module used to compute dense optical flow. Proper application required hardware implementation of additional modules, such as image acquisition module, camera configuration module and memory control module, which is used to store image and determined dense optical flow. The use of external ram was necessary, due to the FPGAs incapability of storing of large amounts of data. This means that utilized FPGA is unable to store even a single frame of VGA video sequence. All of modules are interconnected through FIFO queues and thanks to that solution every module can operate at different clocking


Load test sequence

C:\Users\Janek\Desktop\MethodComparisor

○ 8 frames ● 2 frames

Estimation parametrization

M2

Window size 11

Results

Estimation time 5.1466s

Select error...

Error Value N/A

Fig. 8 Application purposed for comparison of dense optical flow estimation algorithms

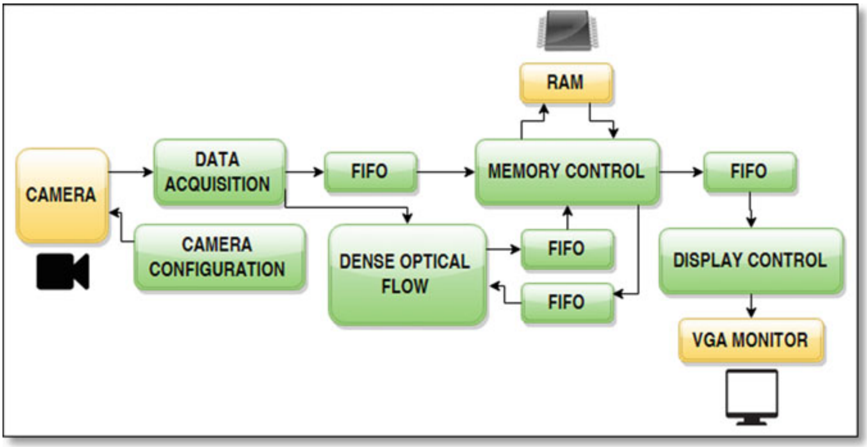


Fig. 9 Simplified block diagram of hardware implementation

speed (forced by utilization of different external devices), independent from each other. Block diagram in Fig. 9 presents simplified hardware configuration of FPGA. Green marked elements are implemented in the FPGA, while the yellow colour is used to mark external devices supported by the system. What is worth mentioning, is that both of the proposed algorithms are computed with utilization of parallel processing—dense optical flow estimation module computes both of the methods at the same time, and user can choose which one is currently displayed.

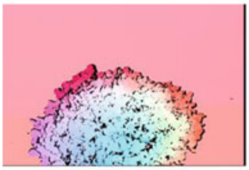
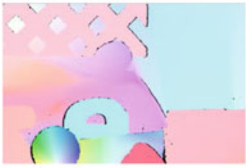
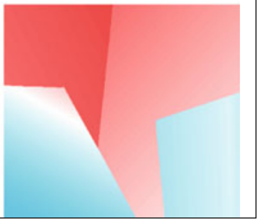
5 Test Results—Predefined Sequences

The section presents the results of comparison of described algorithms of dense motion estimation. Test set, including frames sequences and ground truth optical flow, used in testing process is a collection of sequences, shared by University of Middlebury [27]. All of used sequences are available on the Internet.

5.1 Test Sequences

Table 1 presents used sequences with adequate ground truth optical flow. The optical flow is encoded as follows: the colour of a pixel indicates the angle of estimated movement, while saturation of the colour indicates the value of the apparent movement of light patterns (Fig. 10). Selected from the online base (University of Middlebury) sequences are: Hydrangea, RubberWhale and Venus.

Table 1 Test sequences with adequate ground truth optical flow [27]

	Hydrangea	RubberWhale	Venus
Example frame			
Optical flow			

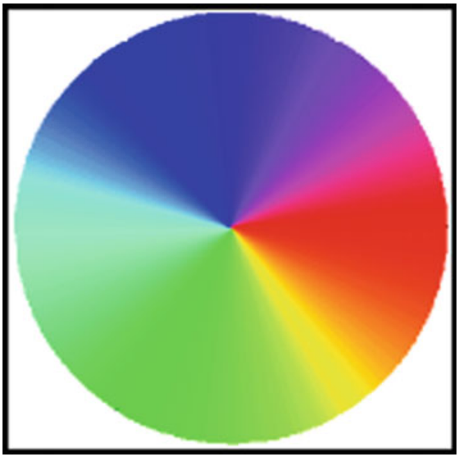


Fig. 10 Color encoding of computed optical flow

5.2 *Error Measurement*

Testing of all researched algorithms required using objective ways of their classification, apart of visual comparison of acquired results. To achieve that, two quality coefficients, which are present in literature, are used [28]. The indicators are: endpoint error (14), which is the average error between estimated and ground truth optical flow values and angular error (15, 16), which stands for the average error between estimated and real angle of movement. Moreover, the third very important factor, from the point of view of maintaining real-time processing, is time of single frame conversion.

- Endpoint error

$$e_E = \frac{\sum_{i=1}^{N} |F(i) - F_{GT}(i)|}{N} \tag{14}$$

- Angular error

$$e_A = \frac{\sum_{i=1}^{N} \theta(i)}{N} \tag{15}$$

$$\theta(i) = \arccos\left(\frac{F(i)^T F_{GT}(i)}{|F(i)||F_{GT}(i)|}\right) \tag{16}$$

F(i)—computed optical flow vector of i-th pixel, $\mathbf{F_{GT}(i)}$—ground truth dense optical flow vector of i-th pixel, **θ(i)**—angle between i-th vectors of computed and ground truth optical flow.

5.3 *Test Results*

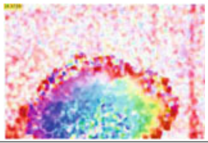
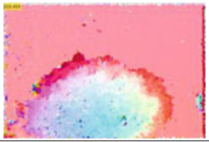
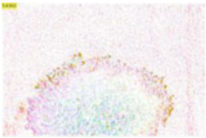
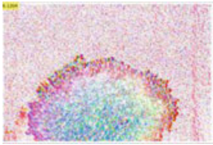
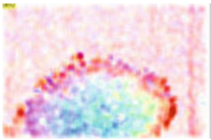
The results of comparison process are presented in Tables 2, 3 and 4. Quality coefficients, taken under consideration in testing process, are angular error, endpoint error and processing time. Moreover, color encoded optical flow is presented in the tables.

5.4 *Summary*

The proposed methods are working properly. In most cases, numerical quality coefficients, characterizing computed dense optical flow, shown Lucas-Kanade and Hierarchical Lucas-Kanade methods as superior, in terms of quality of achieved results. However, achieving greater quality is associated with a significant increase in duration of calculations, most significant for Hierarchical Lucas-Kanade method.

An important feature of the second proposed method, is higher density of the detected optical flow, in comparison with the first method (M1). This is connected with the small changes of brightness between successive frames, which affects pixel values. Because of that method M1 is less effective. Researched methods are

Table 2 Method comparison results—Hydrangea

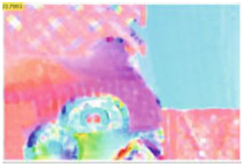
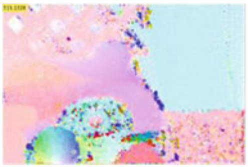
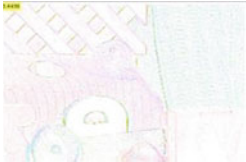
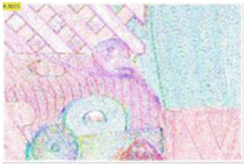
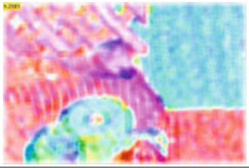
Hydrangea				
Method	Angular error (rad)	Endpoint error (pixel)	Processing time (s)	Visual comparison
LK	0.7007	1.4128	17.5109	
HLK	0.24654	0.3892	237.3942	
M1	1.3910	2.1108	3.6505	
M2	1.3496	1.7161	5.061	
M2 with moving average filter	0.5874	1.5220	5.206	

characterized by higher density of measurements located around high frequencies in images (edges), which is a feature of local methods.

A good way of improvement of the quality of received results can be utilization of moving average filter. Quality coefficients for M2 improved considerably, to the extent, that the results are comparable with both LK and HLK methods. Short time of calculations makes proposed M1 and M2 methods as superior, from the point of view of real-time hardware implementation.

What is more, processing time is measured for a program running on computer. As shown in further tests, hardware implementation using FPGA is characterized by much smaller delays.

Table 3 Method comparison results—RubberWhale

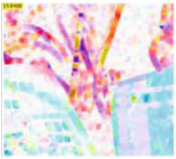
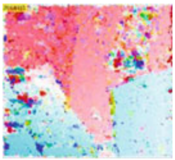
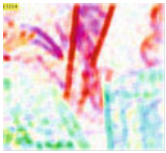
RubberWhale				
Method	Angular error (rad)	Endpoint error (pixel)	Processing time (s)	Visual comparison
LK	0.3172	0.2086	17.6118	
HLK	0.3454	0.3880	241.1111	
M1	1.3021	0.9003	3.4745	
M2	1.1588	0.6650	5.1286	
M2 with moving average filter	0.3966	0.4318	5.2358	

6 Test Results—Generated Video Stream

Another type of conducted tests was verification of the quality of hardware, FPGA implementation of proposed algorithms. In this case, the camera is substituted by video generator, implemented in FPGA, which task is to generate and analyze video stream, with known optical flow. A sample screenshot of the acquired image is shown in Fig. 11. Utilization of video generator creates a possibility to test the impact of external factors, such as:

- multiple objects moving in different directions, both uniformly and differentially structured,

Table 4 Method comparison results—Venus

Venus				
Method	Angular error (rad)	Endpoint error (pixel)	Processing time (s)	Visual comparison
LK	0.8664	1.6621	13.8488	
HLK	0.5890	1.2681	167.1799	
M1	1.3926	2.4605	2.5123	
M2	1.3668	1.9095	3.5656	
M2 with moving average filter	0.8030	1.6884	3.6424	

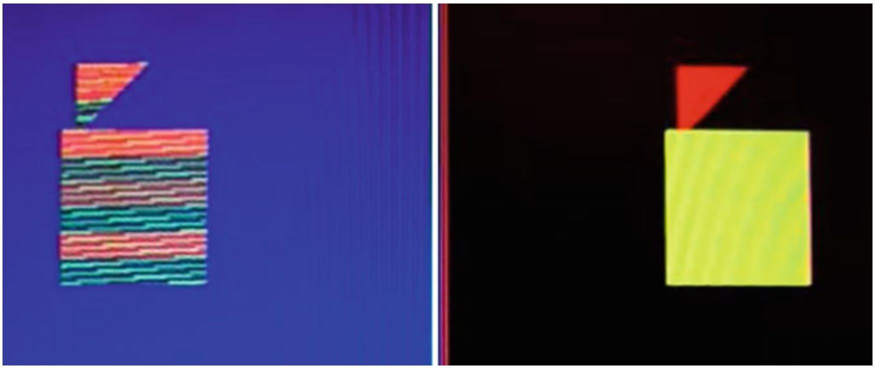


Fig. 11 Example of generated data with real-time computed dense optical flow

Table 5 Test results

<table>
<tr><th>Test</th><th>M1 method</th><th>M2 method</th></tr>
<tr><td>Detection of movement of uniformly structured objects</td><td colspan="2">Detection of movement located only around objects’ edges, proper distinction between objects moving in different directions</td></tr>
<tr><td>Detection of movement of differentially structured objects</td><td colspan="2">Dense motion detection of whole moving objects (Fig. 11), proper distinction between objects moving in different directions</td></tr>
<tr><td>Influence of small brightness variations</td><td>Loss of the optical flow data</td><td>Insignificant errors</td></tr>
<tr><td>Influence of Salt-and-pepper noise</td><td>Noise is influencing detected optical flow</td><td>Noise is influencing detected optical flow, but method 2 is more robust than method 1</td></tr>
<tr><td>Speed efficiency</td><td colspan="2">830,000 pixel of computed optical flow per second 2.7 VGA standard frame per second</td></tr>
</table>

- small variations of brightness between consecutive frames,
- salt-and-pepper noise.

In addition, an important functionality of the generator is control over frequency of pixel clock, which creates possibility to determine the efficiency and speed of the implementation, defined by maximum number of pixels per second, for which the implementation is able to maintain real-time dense optical flow estimation.

The results of the testing process are described in Table 5.

Through conducted series of tests, it was found that used external memory is the bottleneck of the implementation, and prevents dense optical flow from being determined in real time (for VGA standard video). Maximum operating frequency of the memory is 80 MHz, while its data bus has a width of 16 bits. When the generator sends pixels at a frequency greater than 830,000 pixel per second, the picture displayed on monitor starts to flow, due to the loss of lines of the projected image. However, this is only a hardware problem, which can be solved through the use of faster RAM, and does not rule out the possibility of achieving real-time performance for high resolution video.

7 Test Results—Real-Time Dense Motion Estimation in Video Stream Captured by a Camera

The last series of tests involved real-time determination of dense optical flow in video stream acquired by a camera. However, as resulted from previous tests, described in Sect. 6, real-time dense optical flow estimation in VGA standard

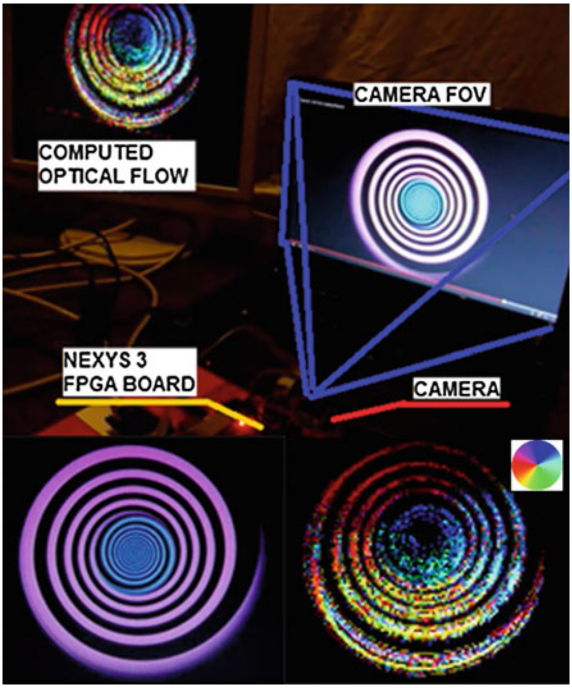


Fig. 12 Test results. Rotation of spirals indicates apparent movement from the center to the edges of an image. Dense optical flow is estimated properly

(640 × 480 pixels, 30 frames per second), has been proved as impossible to realize, due to the insufficient speed of utilized Random Access Memory. The solution to that problem is reduction of resolution of processed video stream. It has been reduced to a value of 160 × 120 pixels, which means that implementation must work at achievable frequency of 576,000 pixel per second. The results of data processing are shown in Fig. 12.

The next step was testing implementation's utility in analysis of real video. During testing process, the film was displayed on the monitor, and was captured by the camera. In order to compare the quality of the estimated dense optical flow, computed results were compared with the results obtained via utilization of Lucas-Kanade method. As can be seen in Fig. 13, the results of processing with use of Lucas-Kanade algorithm are better, but the calculations are time consuming, and cannot be conducted in real-time. In addition, a problem, that occurred during testing process is low quality of the image obtained by the camera. Despite that, estimated information can be considered as useful. The appropriate filtering can be an effective way to improve evaluation of the motion in the image.

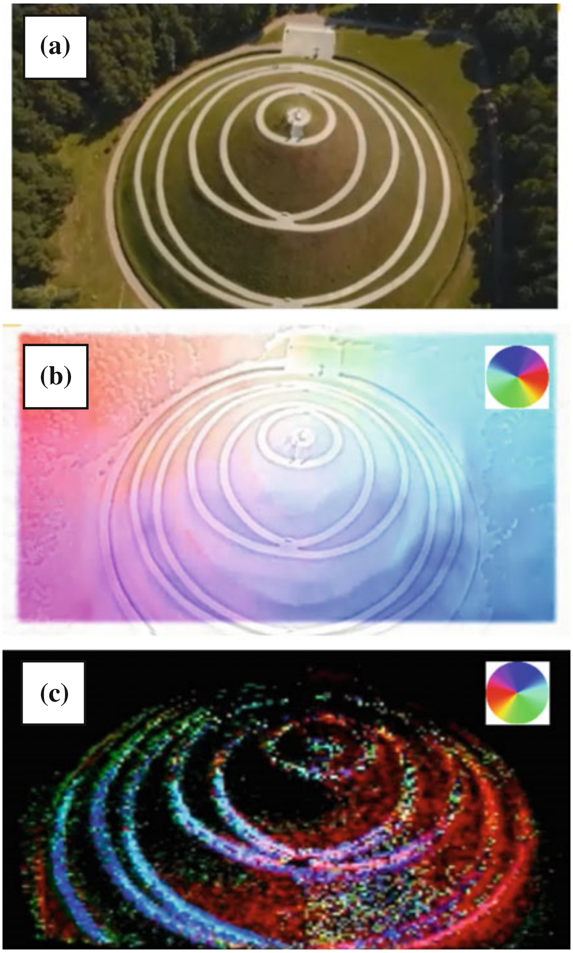


Fig. 13 Comparison of results of dense motion estimation. **a** Source video, **b** Lucas-Kanade, **c** real-time implementation of M2 method. The observer is moving away from captured object, so all of the movement is directed towards center of the image

8 Summary

The presented solution has significant potential for further development. The first suggestion is to use faster external memory, which would allow implementation of advanced algorithms. In addition, good way of further improvement would be the functionality of communicating via the Ethernet interface. This would allow communication with a computer, which would create an opportunity to implement better testing methods.

The most important advantage of the implemented solution is its compactness. The whole system is based on FPGA, cameras, and external memory and is completely independent of the computer, which creates a possibility of using the solution in many branches of contemporary industry, from close circuit surveillance, to advanced military applications [29–32]. Computing of dense optical flow creates possibility to distinguish moving objects from the background, by masking of the image acquired by a camera with estimated optical flow. Thanks to that, objects such as a car or a person may be easily tracked, isolated and identified, with the support of identification algorithm. Moreover, the determined dense optical flow can be found as useful in visual odometry [33], which stands for determination of displacement of an observer, based on visual information. An example of military application of presented solution may be use of optical flow as a support for infrared camera systems of UAVs. Thermography [34] allows detection of objects based on the heat emitted by the objects, in form of infrared radiation. However, nowadays a state-of-the-art solutions to reduce the emitted radiation are being researched. The solutions are known as thermal camouflage. An example of a solution which uses that type of camouflage can be the direct fire support vehicle developed by OBRUM, in partnership with the British BAE Systems. Determination of dense optical flow allows the isolation of such an object on the basis of the image obtained by the casual camera.

In conclusion, the problem of real-time hardware implementation of dense optical flow algorithm, has proven to be extremely difficult and challenging task. The problems encountered during implementation are hardware limitations, namely FPGAs low capability of storing large blocks of data, which resulted in the need of utilization of external memory. External memory, which is available on the Nexys3 FPGA board, is incapable of maintaining rapid switching between read and write operations. This limited the possibility of using advanced dense motion estimation algorithms. An effect of that limitation is implementation of proposed methods, which are not free of drawbacks. The implemented methods have shown their susceptibility to a number of external factors, such as noise, or, as in the case of the first proposed method, changes of global brightness between consecutive frames. The achieved result is the ability of processing maximum 830,000 pixels per second. This number determines the implementations capability of real-time processing. However, testing process of the hardware implementation allowed to confirm the validity of the assumption, that it is possible to implement real-time dense optical flow estimation, based on FPGA programmable logic device, and obtained information is useful for the observer. Implemented algorithms are able to detect changes in successive frames of video sequences, and determine direction of detected movement.

References

1. Fraś, S., Jędrasiak, K., Kwiatkowski, J., Nawrat, A., Sobel, D.: Omnidirectional video acquisition device (OVAD). In: Vision Based Systems for UAV Applications, pp. 123–136. Springer International Publishing (2013)
2. Kwiatkowski, J., Sobel, D., Jędrasiak, K., Nawrat, A.: FPGA based omnidirectional video acquisition device (OVAD). In: Recent Advances in Electrical Engineering and Computer Science, pp. 58–61

3. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control, computer vision and graphics. Lecture Notes in Computer Science, Vol. 7594, pp. 407–413. Springer (2013)
4. Jedrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles, computer vision and graphics. Lecture Notes in Computer Science, Vol. 5337, pp. 391–399. Springer (2009)
5. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features, In: 9th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China (2009)
6. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. Int. J. Comput. Vis. **12**(1), 43–77 (1994)
7. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A Prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion, computer vision and graphics. Lecture Notes in Computer Science, Vol. 7594, pp. 423–432. Springer (2013)
8. Jędrasiak, K., Nawrat, A.: The comparison of capabilities of low light camera, thermal imaging camera and depth map camera for night time surveillance applications. Advanced Technologies for Intelligent Systems of National Border Security, pp. 117–128 (2013)
9. Sobel, D., Jędrasiak, K., Daniec, K., Wrona, J., Jurgaś, P., Nawrat, A.: Camera calibration for tracked vehicles augmented reality applications. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 147–162 (2014)
10. Kwiatkowski, J., Sobel, D., Jedrasiak, K.: FPGA based omnidirectional video acquisition device (OVAD). In: Recent Advances in Electrical Engineering and Computer Science, pp. 58–61. ISBN: 978-1-61804-249-1, 2014
11. Jin, S., Cho, J., Dai Pham, X., Lee, K.M., Park, S.K., Kim, M., Jeon, J.W.: FPGA design and implementation of a realtime stereo vision system. Circ. Syst. Video Technol. IEEE Trans. **20** (1), 15–26 (2010)
12. Díaz, J., Ros, E., Pelayo, F., Ortigosa, E.M., Mota, S.: FPGA-based real-time optical-flow system. Circ. Syst. Video Technol. IEEE Trans. **16**(2), 274–279 (2006)
13. Bailey, D.G.: Design for Embedded Image Processing on FPGAs. Wiley, Hoboken (2011)
14. Gałuszka, A., Skrzypczyk, K., Bereska, D., Pacholczyk, M.: Re-handling operations in small container terminal operated by reach stackers. World Acad. Sci. Eng. Technol. **70**, 674–677 (2010)
15. Czapla, T., Wrona, J.: Technology development of military applications of unmanned ground vehicles. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, Vol. 481, pp. 293–309. ISBN: 978-3-319-00368-9, 2013
16. Johnston, C.T., Gribbon, K.T., Bailey, D.G.: Implementing image processing algorithms on FPGAs. In: Proceedings of the Eleventh Electronics New Zealand Conference, ENZCon'04, pp. 118–123 (2004)
17. Majewski, J., Zbysiński, P.: Układy FPGA w przykładach. BTC (2007)
18. Understanding FPGA and CPLD (online), (access: Dec 2014), access via Internet: http://www.circuitstoday.com/understanding-fpga-and-cpld/
19. Official Xilinx website, (access: Mar 2015), access via Internet: http://www.xilinx.com/
20. Official Digilent website, (access: Dec 2015), access via Internet: http://www.digilentinc.com/
21. Micron, MT45W8MW16BGX datasheet (online), (access: Mar 2014), access via Internet: http://www.micron.com/
22. Szeliski, R.: Computer vision: algorithms and applications, Springer (2010)
23. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow. Comput. Vis. Image Understand. **84**(1), 126–143 (2001)
24. Optical Flow Algorithm Evaluation (online), (access: Dec 2014), access via Internet: http://of-eval.sourceforge.net/
25. Texture Publishing to mental ray (online), (access: Dec 2014), access via Internet: https://elementalray.wordpress.com/

26. Wu, H.: Assignment 0: hybrid images [online], [access: Dec 2014], access via Internet: http://vis.berkeley.edu/
27. Optical Flow—Middlebury Computer Vision, (access: Dec 2014), access via Internet: http://vision.middlebury.edu/flow/data/
28. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**(1), 1–31 (2011)
29. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition, computer vision and graphics. Lecture Notes in Computer Science, Vol. 7594, pp. 87–94. Springer (2013)
30. Antosz, P., Bereska, D., Gatys, K., Niedziela, T., Szota, P.: Multiłącznik IMPRESJA IQ-element instalacji inteligentnego budynku wykorzystującej magistralę CAN. Szybkobieżne Pojazdy Gąsienicowe **1**, 99–106 (2011)
31. Babiarz, A., Jaskot, K., Koralewicz, P.: The control system for autonomous mobile platform. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 15–28 (2013)
32. Bereska, D., Balcewicz, R., Garczyński, M.: Implementacja magistrali CAN i protokołu CANopen w robocie edukacyjnym. Szybkobieżne Pojazdy Gąsienicowe **1**, 157–162 (2008)
33. Campbell, J., Sukthankar, R., Illah, N.: Techniques for evaluating optical flow for visual odometry in extreme terrain. In: Proceedings 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), Vol. 4. IEEE (2004)
34. Bieszczad, G., Krupiński, M., Madura, H., & Sosnowski, T.: Thermal camera for autonomous mobile platforms. In: Vision Based Systems for UAV Applications, pp. 95–114. Springer International Publishing (2013)

GPU-Based Parameters Estimation for Anisotropic Diffusion

Mariusz Domżał, Artur Ryt, Dawid Sobel, Jan Kwiatkowski, Karol Jędrasiak and Aleksander Nawrat

Abstract Anisotropic diffusion is well-known edge-preserving noise removing filter. Unfortunately, estimating parameters of this method can cause some problems. We aimed to find efficient solution to resolve this issue and decided to use GPU compute ability, because of that almost every operation is performed on GPU in parallel. It let us to estimate each parameter in short relatively short time. That algorithm may in future find use in everything that require noise reduction in image.

Keywords Anisotropic diffusion · GPU-based · Parameters estimation

1 Introduction

Nowadays many algorithm that are concerned in image processing require way to eliminate noise from the image. Anisotropic diffusion can remove noise, if right parameters are chosen. Wrong ones can leave image with not removed noise or distorted and one set of parameters is not universal, because of that there is need to estimate them for each image. Unfortunately it is highly time-consuming in case of CPU, this leads to use of GPU, as its computational power is even hundreds times higher [1–5]. Anisotropic diffusion can serve also to identify textures or objects and segment image. Parameter estimation using GPU computational power is established practice [6–9].

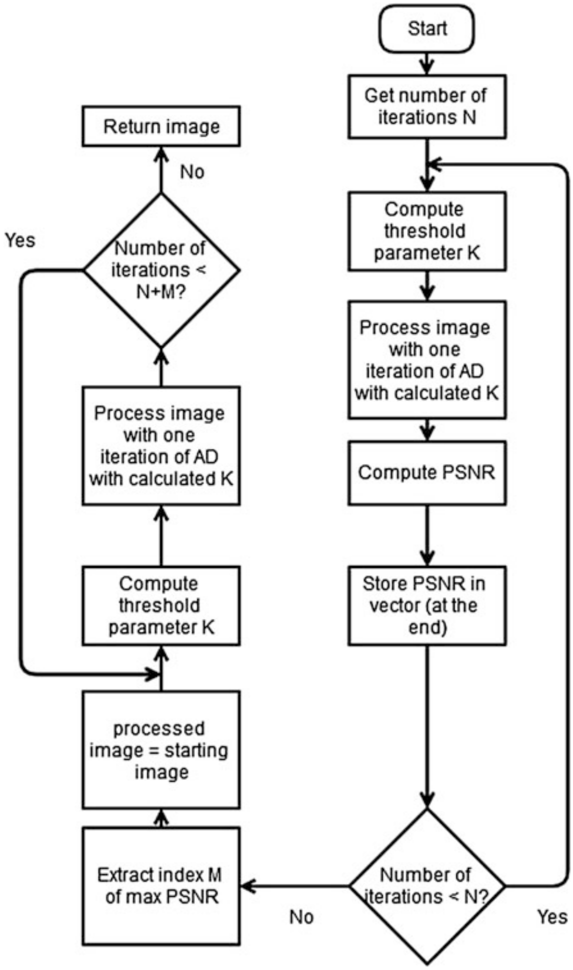
M. Domżał (✉) · A. Ryt · D. Sobel · J. Kwiatkowski · K. Jędrasiak · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland
e-mail: mariusz.domzal@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_8

2 Algorithm

Algorithm is based on modified original paper from Perona and Malik [10]. Values of Laplacian were calculated for 8 neighbours of each pixel. Additionally, algorithm estimates gradient threshold parameter K as 90 % of Canny noise estimator (as Perona and Malik suggested) [11]. Quality of image denoising is measured by PSNR (*Peak Signal-to-Noise Ratio*) [12], which is computed in every iteration. Moreover, PSNR is used by the algorithm to choose number of iterations, which will restore image to the best quality. Algorithm is split into 3 steps: firstly, it processes image for chosen number of iterations (*N*) and computes PSNRs of obtained images, which are stored in a vector. Secondly, after the Nth iteration, index of highest PSNR value is extracted from the vector, assume that is M. In the last, third part, algorithm processes starting image again in M iterations (Fig. 1).

Fig. 1 Flowchart of algorithm


Start
Get number of iterations N
Compute threshold parameter K
Process image with one iteration of AD with calculated K
Compute PSNR
Store PSNR in vector (at the end)
Number of iterations < N?
Yes
No
Extract index M of max PSNR
processed image = starting image
Compute threshold parameter K
Process image with one iteration of AD with calculated K
Number of iterations < N+M?
Yes
No
Return image

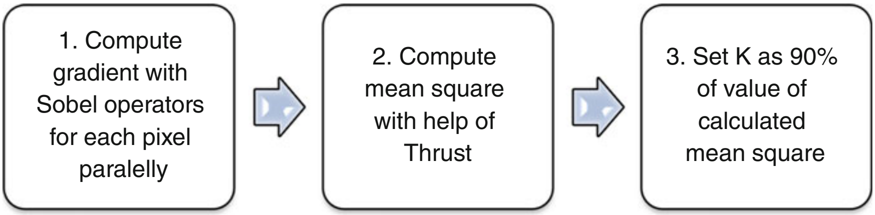

1. Compute gradient with Sobel operators for each pixel paralelly
2. Compute mean square with help of Thrust
3. Set K as 90% of value of calculated mean square

Fig. 2 Computing gradient threshold parameter K

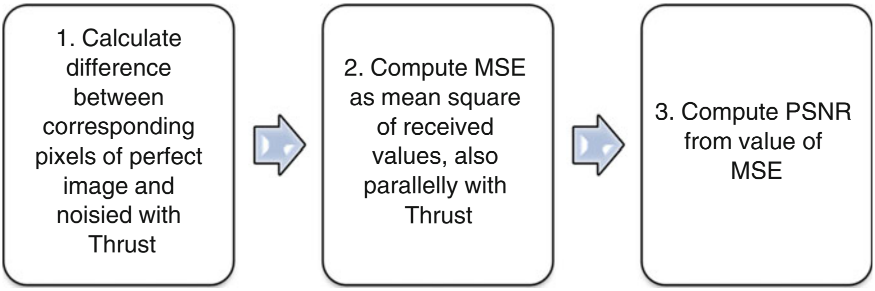

1. Calculate difference between corresponding pixels of perfect image and noisied with Thrust
2. Compute MSE as mean square of received values, also parallelly with Thrust
3. Compute PSNR from value of MSE

Fig. 3 Computing PSNR

3 Implementation

Implementation is based on nVidia developed technology, CUDA [13]. Most of calculations were done by GPU, mainly by using Thrust library. For calculation of gradient and anisotropic diffusion separate kernels were created. Each thread (*instance of kernel*), processes only one pixel and, because of considerable number of processors, computations are performed significantly faster than using traditional CPU (Figs. 2 and 3).

4 Tests

For each image the number of iterations in the first phase was set to 500.

Because of high dependence on image of the third phase of algorithm, only time of processing the first part had been taken into account.

The calculations were performed on a personal computer with an AMD Phenom II X4 3.2 GHz, 8 GB RAM, graphics card NVIDIA GeForce GTX 480 and 64-bit operating system Windows 7 Professional.

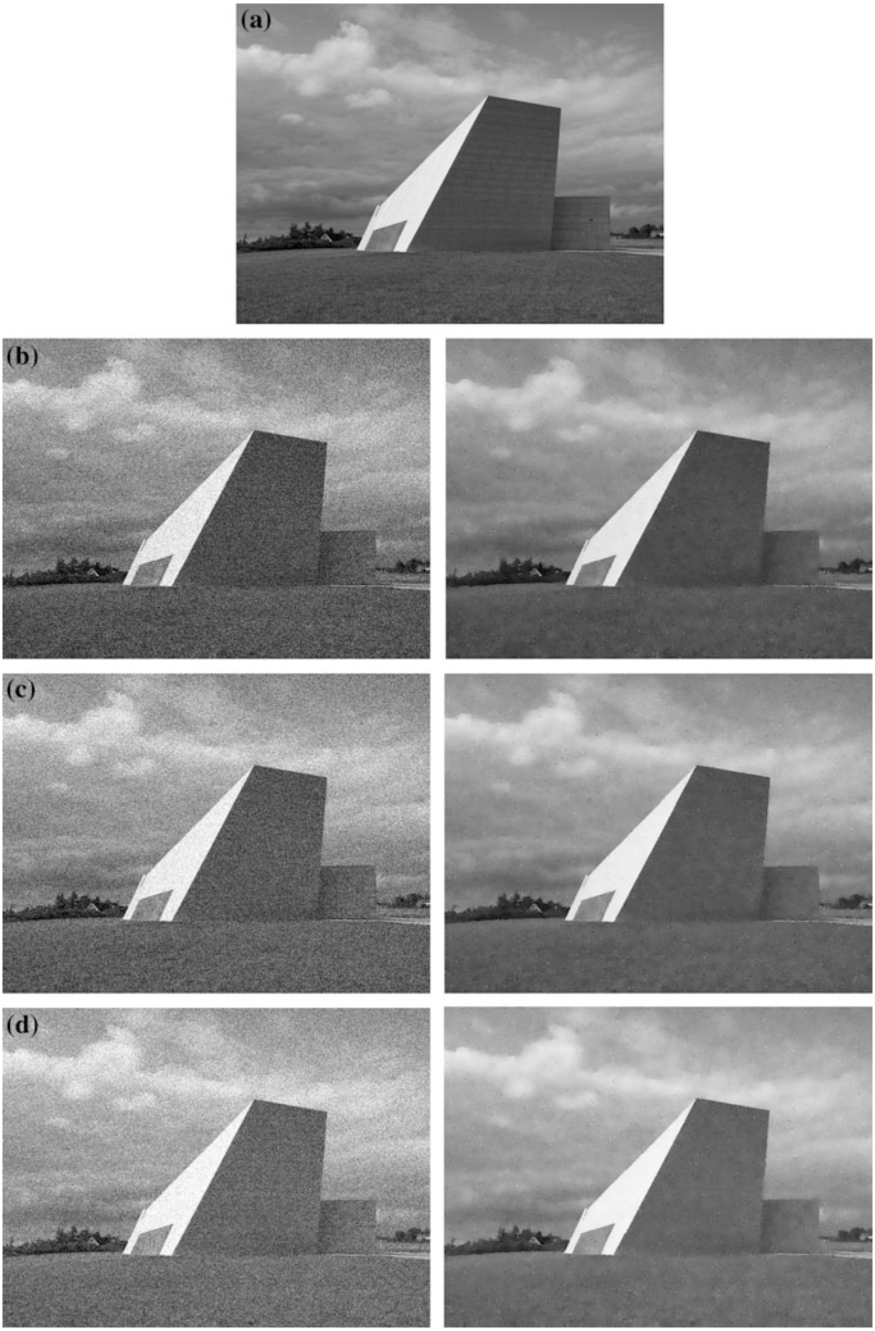


Fig. 4 **a** Original image, images below on the *left* are noised and on the *right* are results. **b** $\sigma^2 = 0.01$, **c** $\sigma^2 = 0.05$, **d** $\sigma^2 = 0.1$, resolution 640 × 480, the highest PSNR on was obtained in **b** 25th, **c** 24th and **d** 25th iteration

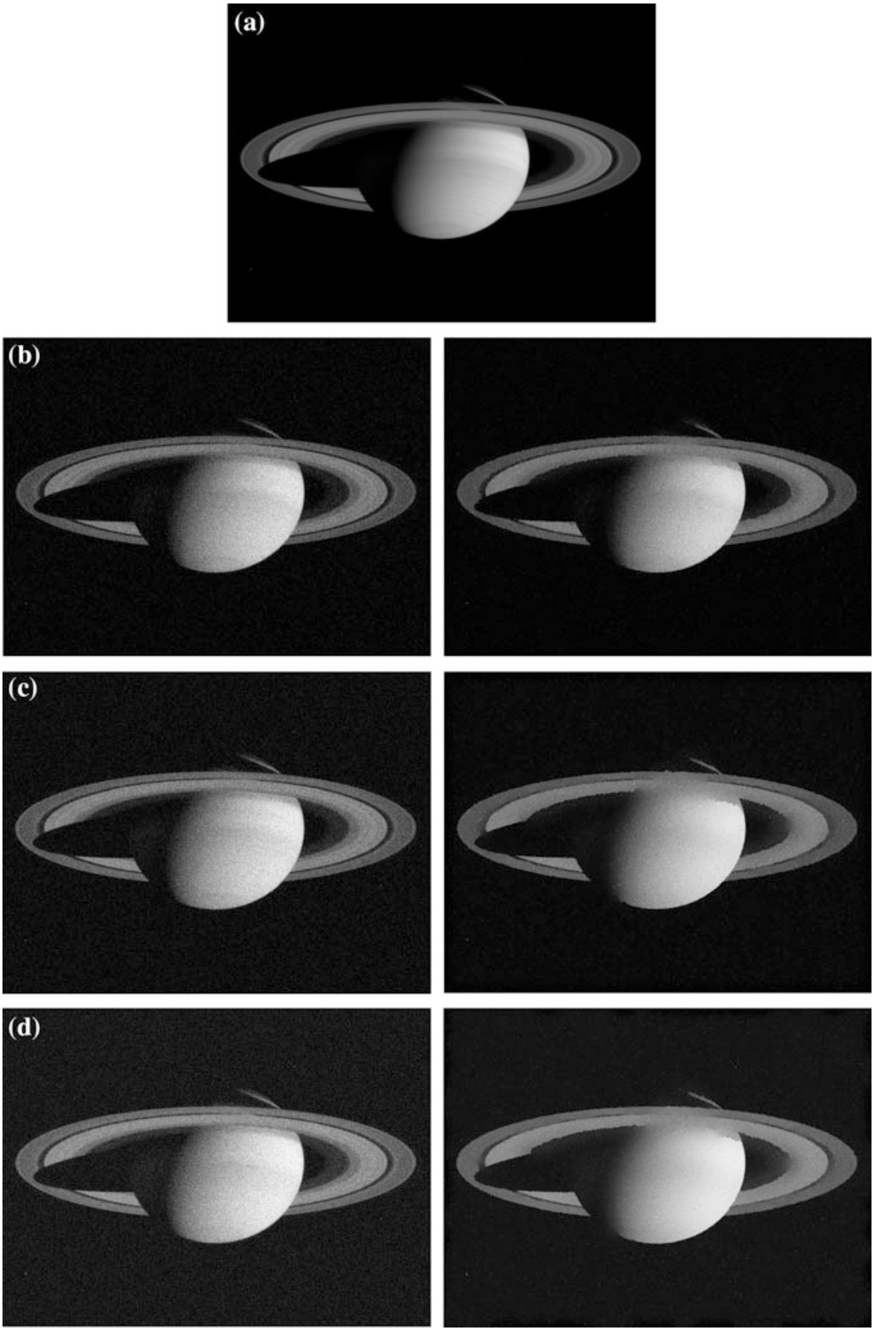


Fig. 5 Similarly as in Fig. 4, the same variance of noise, resolution 1024 × 768, the highest PSNR on **b** 180th iteration and **c** and **d** 500th

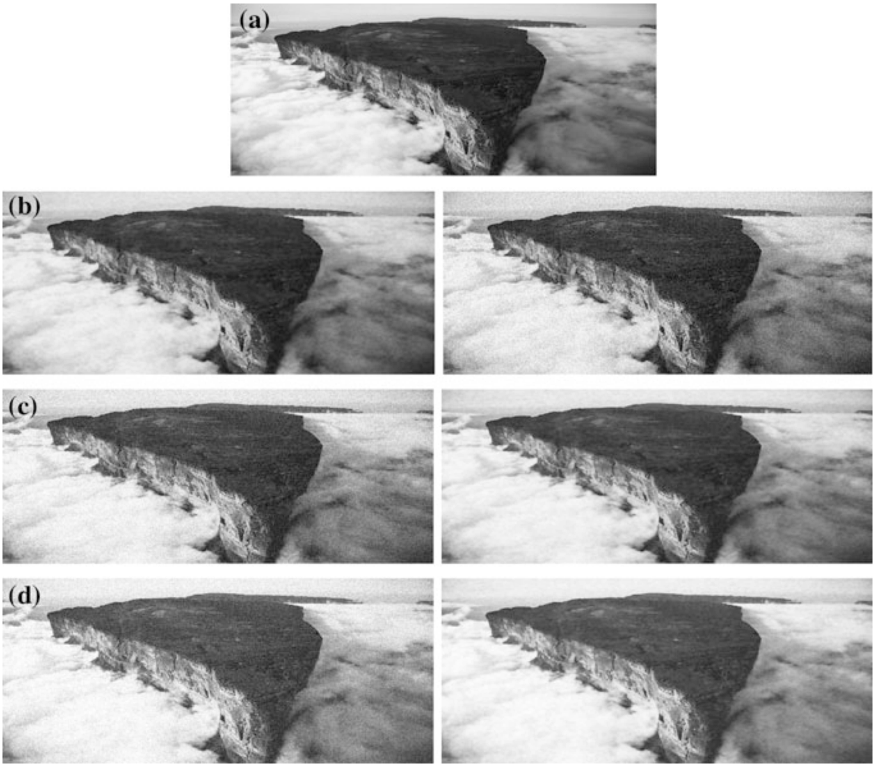


Fig. 6 Similarly as in Figs. 4 and 6 (noise variance too), resolution 963 × 640, the highest PSNR on 8th iteration in each case

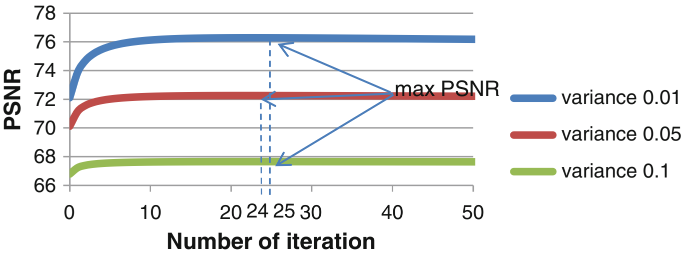


Fig. 7 PSNR of image from Fig. 4 in each iteration

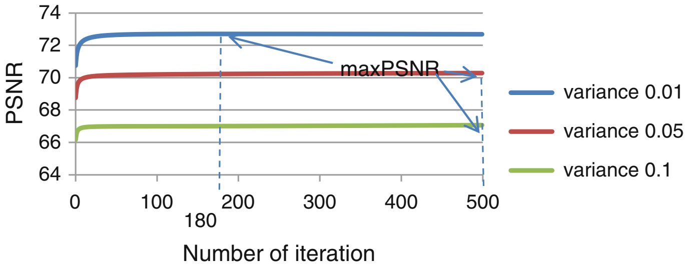


Fig. 8 PSNR of image from Fig. 5 in each iteration

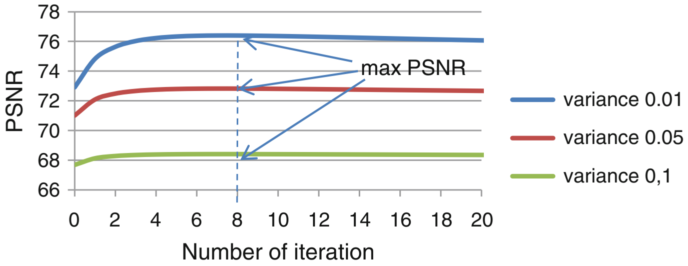


Fig. 9 PSNR of image from Fig. 6 in each iteration

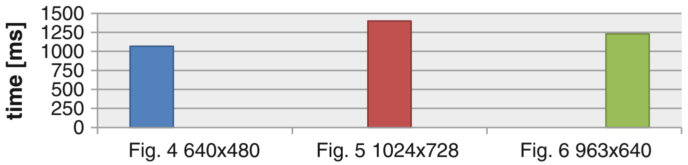


Fig. 10 Average processing time

5 Summary

The outcome of testing process was observation of one flaw of result images. An exiguous amount of noise was left every time, but it can be almost completely removed with utilization of median filter [14]. Taking into account set number of iterations, the time of processing was relatively short, because most of images had not required so much processing (Figs. 4 and 8). Presented GPU-based results suggest that it is possible to apply the presented method in real-life applications e.g. preprocessing for visual-based control of UAVs [15–18] or tracking pedestrians [19] (Figs. 7, 9, and 10).

Acknowledgment This work has been supported by Applied Research Programme of the National Centre for Research and Development as a project ID 178438 path A—Costume for acquisition of human movement based on IMU sensors with collection, visualization and data analysis software.

References

1. Jędrasiak K., Nawrat A.: The comparison of capabilities of low light camera, thermal imaging camera and depth map camera for night time surveillance applications. In: Advanced Technologies for Intelligent Systems of National Border Security, pp. 117–128 (2013)
2. Jędrasiak, K., Daniec, K., Nawrat, A., Koteras, R.: Wykorzystanie kamer termowizyjnych w systemach dozoru wizyjnego infrastruktury krytycznej sieci dystrybucyjnych gazu, Przegląd Elektrotechniczny, vol. 88, pp. 90–97 (2012)
3. Sobel, D., Jędrasiak, K., Daniec, K., Wrona, J., Jurgaś, P., Nawrat, A.: Camera calibration for tracked vehicles augmented reality applications. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 147–162 (2014)
4. Kuś, Z., Nawrat, A.: Object tracking in a picture during rapid camera movements. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 77–91. ISBN: 978-3-319-00368-9, 2013
5. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking, computer vision and graphics. Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)
6. Sharma, S.: Parameter Estimation for System Biology Models on GPU Clusters. North Carolina State University (2013)
7. Bereska, D., Daniec, K., Jędrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 115–121. ISBN: 978-3-319-00368-9, 2013
8. Sroka, M., Ściegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzałogowego pojazdu podwodnego – układ stabilizacji i utrzymania zadanego kursu, Przegląd Elektrotechniczny, vol. 89, pp. 205–217 (2013)
9. Jaskot, K., Babiarz, A., Sroka, M., Ściegienka, P.: Prototyp bezzałogowego pojazdu podwodnego – konstrukcja mechaniczna, panel operatora, Przegląd Elektrotechniczny, vol. 89, pp. 52-67 (2013)
10. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7) (1990)
11. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**(6) 1986
12. Quan, H.-T., Mohammed, G.: Scope of validity of PSNR in image/video quality assessment. Electron. Lett. **44**(13) 2008
13. Nvidia, C.U.D.A.: Nvidia cuda c programming guide. NVIDIA Corporation 120 (2011)
14. Andrzejczak, M., Ulinowicz, M.: Filtration and integration system (FIS) for navigation data processing based on Kalman filter. In: Intelligent Systems 2014, pp. 203–2010 (2015)
15. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition, computer vision and graphics. Lecture Notes in Computer Science, vol. 7594, pp. 87–94. Springer (2013)
16. Kuś, Z., Nawrat, A.M.: The limitation for the angular velocity of the camera head during object tracking with the use of the UAV. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 127–145 (2014)

17. Kuś, Z., Nawrat, A.M.: Camera head control system with a changeable gain in a proportional regulator for object tracking. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 105–125 (2014)
18. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 157–175. ISBN: 978-3-319-00368-9, 2013
19. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion, computer vision and graphics. Lecture Notes in Computer Science, vol. 7594, pp. 423–432. Springer (2013)

An Evaluation of SETh—The Method for Long-Term Feature Tracking

Karol Jędrasiak and Aleksander Nawrat

Abstract The chapter presents a novel long-term object tracking method called SETh. It is an adaptive tracking by detection method which allows near real-time tracking within challenging sequences. The algorithm consists of three stages: detection, verification and learning. In order to measure the performance of the method a video data set consisting of more than a hundred videos was created and manually labelled by a human. Quality of the tracking by SETh was compared against five state-of-the-art methods. The presented method achieved results comparable and mostly exceeding the existing methods, which proves its capability for real life applications like e.g. vision-based control of UAVs.

Keywords Object tracking · Long-term tracking · Adaptive · Image processing

1 Introduction

The modern world is full of cameras placed in supermarkets, banks, or on the streets. Each of these cameras record movies in the form of a compressed image sequence. A huge number of recorded information makes it impossible to be verified by human. Therefore there is a need for an automatic method of analysis of the sequences. One of the main issues associated with this problem is the problem of tracking objects in image sequences. For a human it is considered a simple task, but for machine it is rather a complicated process because of the need for the extraction of the data from the image [1–3].

K. Jędrasiak (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology,
Akademicka, 16, 44-100 Gliwice, Poland
e-mail: karol@polsl.pl

A. Nawrat
e-mail: jedrasiak@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_9

Currently, algorithms for tracking objects of interest between the individual frames have reached a certain level of maturity allowing accurate tracking of the objects under the assumption that the objects does not change its shape and appearance. Such restrictions are not met in real scenarios therefore the existing algorithms for long-term tracking are disappointing. Changes in the appearance of the tracked object requires a certain way of updating detection module to the new conditions. Development of a new method for long-term objects tracking is motivated by the fact that less than one percent of the recorded surveillance video is ever watched [4, 5]. The use of automated analysis of recorded material is particularly important in the crises situations, such as terrorist attacks. For example, a review of video material of bombings in Dubai in 2010 by a human lasted for several weeks, where the automatic analysis of materials of attack in Boston lasted only three days. The article presents a novel object tracking method which is computationally straightforward and performs in near real-time. It consists of three consecutive phases: detection, verification and learning in a way inspired by semi-supervised methods. The proposed approach and the developed algorithms were verified using a comprehensive set of prepared test sequences consisting of both synthetic and real scenes [6–10].

2 Literature Review

The term “method for object tracking” is defined as any method aimed to estimate the trajectory of a moving object being tracked in a sequence of images. The task of the tracker is to assign consistent labels to the tracked object in a sequence of consecutive frames [11]. Object tracking is, however, complex, e.g. due to the following problems [12]: the loss of information caused by projection of the 3D world on a 2D one; noise in the images; complex motion of objects; loose or articulated nature of the shape of objects; partial or complete occlusion of the tracked object; complicated shape of objects; changes in scene illumination; time constraints related to the real-time processing.

Visual tracking is considered one of the fundamental problems in computer vision. It is used in e.g. vision surveillance, human-computer interactions, navigation of unmanned objects, or issues related to the expanded reality [13]. Some tracking applications assume that the tracked object is known in advance, which allows to use the knowledge during the process of designing the tracking method. However, majority of the methods allows to track any object determined during the algorithm work time [12].

Below are presented some of the object tracking methods considered as the state-of-the-art reference methods. One of the most popular algorithms for tracking of the visual features is the algorithm called the Lukas-Kanade Tracker (KLT) [14]. The algorithm allows tracking features between subsequent images of the sequence. KLT can be divided into two main phases: detection of features and tracking. Detection of characteristic points is usually implemented using the autocorrelation method, e.g. Harris corner detector. Localization of feature points is found by identifying for each of the points the translation vector that minimizes the difference between the measure computed within a rectangular window centered around analyzed in pixels.

TLD method [15] (*Tracking-Learning-Detection*) is able to unequivocal state whether the defined in the first frame of a sequence tracked object is within the camera view or not. TLD method assumes that the long-term tracking of objects should consist of three phases: tracking, learning and detection. Tracking is realized by the Median-flow-tracker [16]. The task of the detection is independent of the tracking. NCC was used for the purpose. The detector can commit two types of errors: false positive and false negative. The task of the learning element is the observation of the tracker and detector and estimation on the basis error of detection and generation of new training samples in order to reduce the impact of the identified errors in the future.

FRAGtrack algorithm [17] assumes that the tracked object is represented by multiple image fragments. Each fragment vote regarding the probable position and the scale level of the tracked object by comparing the histogram of their area to the histogram of the tracked object from the first frame. The approach based on voting allows to track during partial occlusion or changes in pose of the tracked object. The authors emphasize that the proposed method is characterized by the constant computational complexity regardless of the size of the object being tracked.

VTD tracking method [12] according to the authors allows to track objects at the same time changing the appearance and character of the movement. The solution assumes the division of tracking tasks in two stages: defining the model of observation and tracking its movement. Sparse PCA is calculated on a set of basic patterns of motion and appearance features. Tracking is also composed of a number of tracking compound elements where each of them realize tracking of different type of object changes. Results returned by tracking elements are further combined into one by usage of IMCMC (*Interactive Markov Chain Monte Carlo*).

The authors in [18] note that the tracking methods based on detection are largely based on a classifier, which task is to distinguish an object from its background. Even small errors in tracking element can cause the erroneous determination of training samples of the classifier and in the result cause a drift of the solution. The authors present the solution where they use the method called MIL Track (*Multiple Instance Learning Tracking*) instead of the typical supervised learning.

3 SETh

Among many groups of different methods of tracking one of the most convenient for the user with simultaneously some of the best tracking results is the group of tracking by detection [19, 20]. Typically, the object of interest is visible in the frame for considerable amount of time. However, there is a high probability that in a non-zero time the object is outside the view of the camera. It is assumed that in the first frame of a sequence a rectangular area of interest for tracked object is selected and the aim of the tracking algorithm is to detect the object of interest in successive frames of a sequence or to specify that the object is not visible in the image. Stream processing is done frame by frame, and the process time can be infinitely long. Thus defined tracking is known as long-term tracking [15]. Long-term tracking is difficult due to, e.g. problem of determining whether an object is within the field of view of the camera. This problem belongs to the complex ones, as the tracked object at that time could change the position, orientation, or appearance, therefore its appearance known from the first frame of the sequence may become obsolete [21]. As another important problem we can identify resistance of the tracking algorithm to changes in a camera position, lighting conditions, partial and total occlusion and moving background and finally, reducing the time of processing. The long-term tracking is widely considered as a combination of two phases: tracking and detection [15].

The proposed algorithm is derived from the family of methods of tracking by detection, generalized by updating the model of both tracked object and its closest surroundings. The developed algorithm for long-term tracking, SETh [22], is based on many years of experience and researches of the author [23–25].

The algorithm is initialized, and then executed sequentially in three successive phases: detection, verification, and learning. SETh algorithm is used to determine the position of the object being tracked or unequivocal statement that the object is not visible in the image. The algorithm allows tracking in a manner inspired by the semi-supervised methods, i.e., the operator in the first frame to track of the sequence indicates the area of interest containing the object to be tracked. In subsequent frames of the sequence the task of the algorithm is to track the position of the object without any additional information (Fig. 1).

During the detection step the goal is to detect features in the image, and then assign them to the appropriate labels: object, background or indeterminate features. The result of this step are sets: $\mathfrak{T}$—features of the object, $\mathfrak{B}$—features of the background and $\mathfrak{Z}$ unrecognized features so far.

An important element of the presented algorithm is a method for detection of features. The ideal detector is defined as possibly computationally simple method for finding the areas of the image possible to detect reproducibly regardless of the change in the point of view of the camera and at the same time resistant to all possible types of transformation. Currently, the closest to the prescribed requirements and with shortest computation time is BRISK detector [26]. Therefore, it was decided to use it as an element of the proposed long-term tracking algorithm.

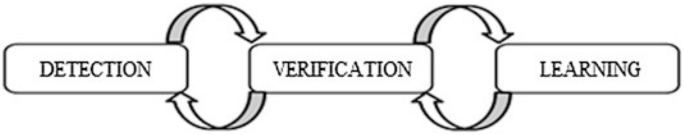


Fig. 1 Algorithm overview schema

BRISK is insensitive to scaling and rotation due to the addition of local maxima search step not only in image space, but also in the scale space.

Detected features have to be described by a descriptor that allows them to be uniquely compared. Description of the detected features should provide plenty uniqueness of the description, be computed efficiently and allows to timely and accurately compare the descriptor with a large set of data. All these advantages are met by FREAK descriptor described in [27]. FREAK was created based on the inspiration of information processes occurring in the human retina. FREAK descriptor is an efficient way to describe the feature by a cascade of binary string numbers calculated on the basis of differences in brightness in the area similar to the human retina sampling area. The sampling pattern which is used is circular and the points closer to the center have a higher density distribution. The density of occurrence decreases exponentially with the distance from the center of the feature which is described. Binary string of FREAK descriptor F (1) is a one-bit sequence coding differences in Gaussian function (DoG):

$$F = \sum_{0 \leq a \leq N} 2^a T(P_a), \tag{1}$$

$$T(P_a) = \begin{cases} 1, & \left(I\left(P_a^{r_1}\right) - I\left(P_a^{r_2}\right)\right) > 0, \\ 0, & \mathit{otherwise}, \end{cases} \tag{2}$$

where P_a is a pair of receptive fields, and N is the desired length of the descriptor, $I\left(P_a^{r_1}\right)$ is a smooth function of a Gaussian brightness value of the first pair of reception field P_a. Combinations of several tens of pairs of fields result in the thousands of possible pairs of which 512 are selected by the decorrelation.

According to the scheme of the algorithm detected and described features have to be assigned to $\mathfrak{T}$, $\mathfrak{B}$ and $\mathfrak{Z}$ sets on the basis of comparisons with the features of the object from the previous frame of the tracked object $\mathfrak{T}'$ and the background $\mathfrak{B}'$.

The main objective of the verification phase is the selection of the correct position of the object, from the proposed by the stage of detection, and to determine the certainty level $m_{\mathfrak{T}}$. Simplified schema of the verification phase is shown in Fig. 2.

Features belonging to a group with confidence level above the thresholds γ_1 and γ_2 are passed on to the stage of learning which update the model for binary classification of features between the object and the background. If there were not enough features detected during the detection phase an alternative calculations are made in order to face the problem.

The detected features are labeled as object contained in the set $\mathfrak{T}$ may indicate multiple localizations of the object being tracked. The observed scene may contain more than one object identical to the tracked. In addition, the detected characteristic points in the face of noise can be detected incorrectly. For this reason, the detection and matching of characteristic points is insufficient to determine the correct position of the object.

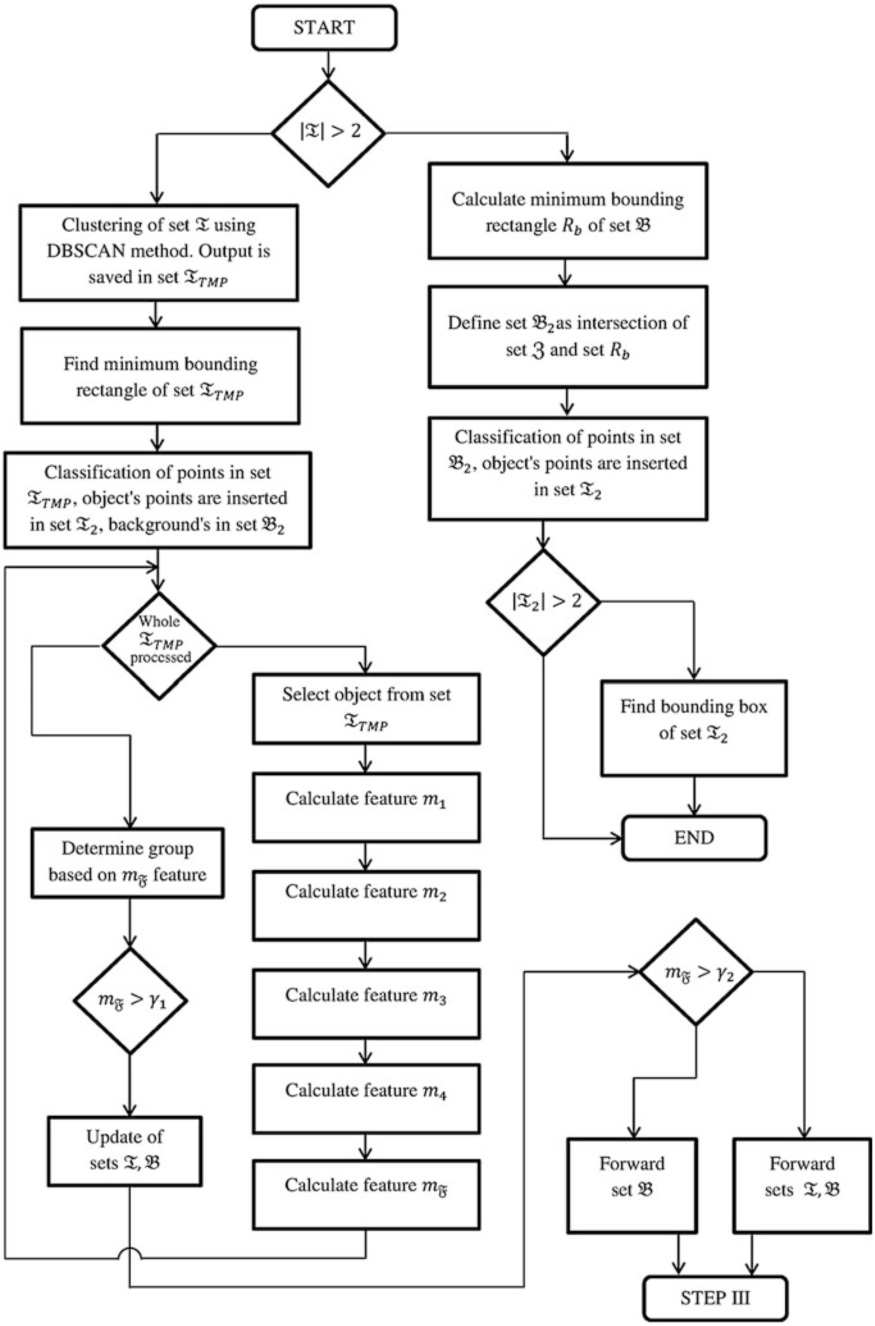

START
$|\mathfrak{T}| > 2$
Clustering of set $\mathfrak{T}$ using DBSCAN method. Output is saved in set $\mathfrak{T}_{TMP}$
Find minimum bounding rectangle of set $\mathfrak{T}_{TMP}$
Classification of points in set $\mathfrak{T}_{TMP}$, object's points are inserted in set $\mathfrak{T}_2$, background's in set $\mathfrak{B}_2$
Whole $\mathfrak{T}_{TMP}$ processed
Select object from set $\mathfrak{T}_{TMP}$
Calculate feature m_1
Calculate feature m_2
Calculate feature m_3
Calculate feature m_4
Calculate feature $m_{\mathfrak{F}}$
Determine group based on $m_{\mathfrak{F}}$ feature
$m_{\mathfrak{F}} > \gamma_1$
Update of sets $\mathfrak{T}, \mathfrak{B}$
Calculate minimum bounding rectangle R_b of set $\mathfrak{B}$
Define set $\mathfrak{B}_2$as intersection of set $\mathfrak{Z}$ and set R_b
Classification of points in set $\mathfrak{B}_2$, object's points are inserted in set $\mathfrak{T}_2$
$|\mathfrak{T}_2| > 2$
Find bounding box of set $\mathfrak{T}_2$
END
$m_{\mathfrak{F}} > \gamma_2$
Forward set $\mathfrak{B}$
Forward sets $\mathfrak{T}, \mathfrak{B}$
STEP III

Fig. 2 Simplified schema of the verification stage of SETh long-term tracking method

It is therefore required to specify unknown number of areas in a way resistance to noise and outliers. It is assumed that incorrect matches are distant from each other in Euclidean distance sense. For the purpose an unsupervised learning method DBSCAN was used [28]. The following parameter values were chosen: Min_{pts} = 3, because a group of three points can be described by the minimal area rectangle; E_{ps} = 61, as suggested in a research paper of FLIR company [29], where measured the minimal number of pixels sufficient for recognition of a person from the distance of 45 m. Discovered N groups g in a set $\mathfrak{T}$ were labeled $\mathfrak{T}_{tym}$. The tracked object is labeled using Feret box with a center in a point $c(x, y)$ in a first frame of sequence, so in each successive frame the detected area is also bounded by a rectangle. For each features in each group g in the set $\mathfrak{T}$ it is required to describe the minimal area rectangle r_i with a center in point $c_i(x, y)$. The problem of finding the minimal area rectangle bound given set of points has been solved using the SGPRC [30] method.

Analyzed during the verification phase of the algorithm areas potentially containing the tracked object P_{o_i} are defined on the basis of predefined groups g_i, describing them rectangles r_i, r_{ib}, detected and labeled background $P_{o_i}^{\mathfrak{B}}$ and object features $P_{o_i}^{\mathfrak{T}}$, as data structures consisting of the following elements: $P_{o_i} = \{r_i, r_{ib}, P_{o_i}^{\mathfrak{T}}, P_{o_i}^{\mathfrak{B}}, P_{o_i}^{\mathfrak{T}_2}, P_{o_i}^{\mathfrak{B}_2}\}$, where $P_{o_i}^{\mathfrak{T}_2}$ and $P_{o_i}^{\mathfrak{B}_2}$ are initially blank sets in which the classified features from the $\mathfrak{Z}$ set would be placed.

In order to attach the labels to the features from the $\mathfrak{Z}$ set a non-linear SVM classifier with RBF kernel function and $\gamma = 3$ is used. The classifier is trained in a supervised manner during the initialization phase. Features from the $\mathfrak{Z}$ are classified and distributed between features set $P_{o_i}^{\mathfrak{T}_2}$ and background set $P_{o_i}^{\mathfrak{B}_2}$. For each potential position of the tracked object a measure of area significance $m_{\mathfrak{F}}$ is computed. The measure is a weighted average of four partial measures introduced in order to face the typical long-term tracking. Measure m_1 allows to determine the ratio of background matching for the evaluated P_{o_i}. The measure is calculated according to the formula (3). Measure m_2 determines the degree of the evaluated P_{o_i} similarity to the current model of the object being tracked (4). Measure m_3 allows the correct detection of the object being tracked in case of detection of multiple identical objects. It is based on the assumption of a certain continuity of motion and is expressed as the Euclidean distance (5). Measure m_4 determines the ratio of object features matching for the evaluated P_{o_i} (6).

$$m_1 = \frac{\left|P_{o_i}^{\mathfrak{B}}\right| + \left|P_{o_i}^{\mathfrak{B}_2}\right|}{|\mathfrak{B}| + T_{B2}} \tag{3}$$

$$m_2 = \frac{\left|P_{o_i}^{\mathfrak{T}_2}\right|}{T_{T2}} \tag{4}$$

$$m_3 = 1 - \frac{\sqrt{(c_{ix} - c_x)^2 + (c_{iy} - c_y)^2}}{L} \tag{5}$$

$$m_4 = \frac{\left|P_{o_i}^{\mathfrak{T}}\right|}{|\mathfrak{T}|} \tag{6}$$

$$T_{B2} = \sum_{i=1}^{N} \left|P_{o_i}^{\mathfrak{B}_2}\right|, \tag{7}$$

$$T_{T2} = \sum_{i=1}^{N} \left|P_{o_i}^{\mathfrak{T}_2}\right|, \tag{8}$$

$$L = \sum_{i=1}^{N} \sqrt{\left(c_{ix} - c_x\right)^2 + \left(c_{iy} - c_y\right)^2}. \tag{9}$$

The values of all measurements were normalized to a closed interval from zero to one. For this purpose the following auxiliary variables (7)–(9) were introduced. The final measure $m_{\mathfrak{F}}$ is calculated in accordance to the following formula: $m_{\mathfrak{F}} = \alpha m_1 + \beta m_2 + \gamma m_3 + \delta m_4$, where α, β, γ, δ are the weight coefficients of each partial measure. In the study the following coefficient values were used: $\alpha = 0.3$, $\beta = 0.1$, $\gamma = 0.1$, $\delta = 0.5$. The values α and δ were as greater in order to emphasize the significance of the feature descriptors matching with respect to additional heuristics.

The tracked object is considered to be the potential object P_{o_i} characterized by the highest value of the final measure $m_{\mathfrak{F}}$. If the value of the final measurement is above the cut-off γ_1 we assume that one can update the sets of object $\mathfrak{T}'$ and background $\mathfrak{B}'$ features. Experimentally determined value $\gamma_1 = 0.3$. In addition, it is recognized that above this cut-off background features are further passed to the learning phase of the algorithm. The significance level $m_{\mathfrak{F}}$ above the cut-off γ_2 is considered reliably and above it all the object and background features are further passed to the learning phase of the algorithm. Experimentally determined value of γ_2 is 0.7.

Alternatively, if there are no more than two features in the set $\mathfrak{T}$ we suggest to conduct countermeasure based on the background features matching. The procedure is based on the observation that a moving object on the stage is not fully independent. Its presence affects the other elements of the scene by changing the reflection of light, generating shadows, shielding some area from rain, etc. Classically this phenomenon is seen as negative, hindering the process of object tracking. Here however, we assume that there is the tracked object visible in the image, however due to dynamic change of appearance the matching process didn't succeed but there is still a possibility that the object is within the best matched background area of interest. In order to point out the localization we classify the features within the matched area and treat the result as a temporary good hint.

The long-term tracking requires updating a representation of the object being tracked in order to ensure its quality in the face of changes occurring in appearance of the object. Goal of the learning phase is to iteratively build a model representation of the tracked object based on consecutive frames of the video stream. Features within $\mathfrak{T}'$, $\mathfrak{B}'$ sets selected during verification phase are used to teach nonlinear kernel SVM classifier with RBF kernel with γ selected as three.

4 Tests

An important problem from the point of view of the effectiveness of tracking algorithms is the method for evaluating the acquired results. There are applications of both the qualitative and quantitative analysis. The following measures were inspired by the known from the literature measures [15, 31, 32]. Used measures are based on the sequence length L, determined by the human expert reference ground truth labeled R_G, area of interest computed by the tracking method labeled R_W, the number of correct indications τ_P, such frames as the value $\epsilon_{OR} > 50\,\%$, the total number of indications labeled τ. There is used a measure of compliance called overlap ratio defined as the weighted ratio between the R_W and R_G areas called ε_{OR} (10), measure of the ratio of the number of correct indications to the length of the sequence is denoted by ε_{SR} (11), measure of the location error calculated as the Euclidean distance between the center of the R_W and R_G denoted by ε_{LO} (12), average error value computed as an arithmetic mean of localization error, labeled as ε_{ALO} (13).

$$\varepsilon_{OR} = 2\frac{|R_G \cap R_W|}{|R_G| + |R_W|}, \tag{10}$$

$$\varepsilon_{SR} = \frac{\tau_P}{L}, \tag{11}$$

$$\varepsilon_{LO} = \sqrt{(R_{Gx} - R_{Wx})^2 + \left(R_{Gy} - R_{Wy}\right)^2}, \tag{12}$$

$$\varepsilon_{ALO} = \frac{1}{L}\sum_{i=1}^{L} \varepsilon_{LO_i} \tag{13}$$

Acquired results by the proposed long-term tracking method SETh were compared against reference methods. For comparisons the publicly available for research purposes implementations of the selected methods were used. There were five state-of-the-art reference methods selected for comparison: Lukas Kanade Tracker, labeled in results as KLT; Tracking-Learning-Detection, labeled in results as TLD; FRAGTrack, labeled in results as FRAG; Visual Tracking Decomposition, labeled in results as VTD; Multiple Instance Learning Tracking, labeled in results as MIL Track.

Due to the limitations of length of the article only selected representative test results are presented from a set of total 102 test sequences. There have been hand annotated over 25500 frames. All sequences are documented and the data set has been made available to the public [33].

The calculations were performed on a personal computer with an Intel Core i5 2.4 GHz, 8 GB RAM, graphics card NVIDIA GeForce GT 525 M and 64-bit operating system Windows 7 Professional.

The acquired results of comparison of tracking quality of SETh and reference state-of-the-art method are presented in Fig. 3. It can be observed that presented SETh method was able to track the object during the whole sequence. It is worth to mention that other methods lost their target and had problems with tracking reinitialization. Computed measures (Fig. 4) present that both the highest localization precision and the lowest average error were scored by presented SETh method. Second result is taken by VTD and surprisingly the last place was reserved for TLD method.

In Fig. 5 visual results of object tracking during car driving are presented. The camera was attached to the head of the driver (*glasses*). It can be observed that all tracking methods except the proposed SETh and TLD methods failed. It is worth to notice that the presented SETh method is invariant to rotation and scale. However it is vulnerable to low contrast and visible in the frame 169. It can be observed that regardless the short error in tracking the proposed method was able to correctly reinitialize tracking.

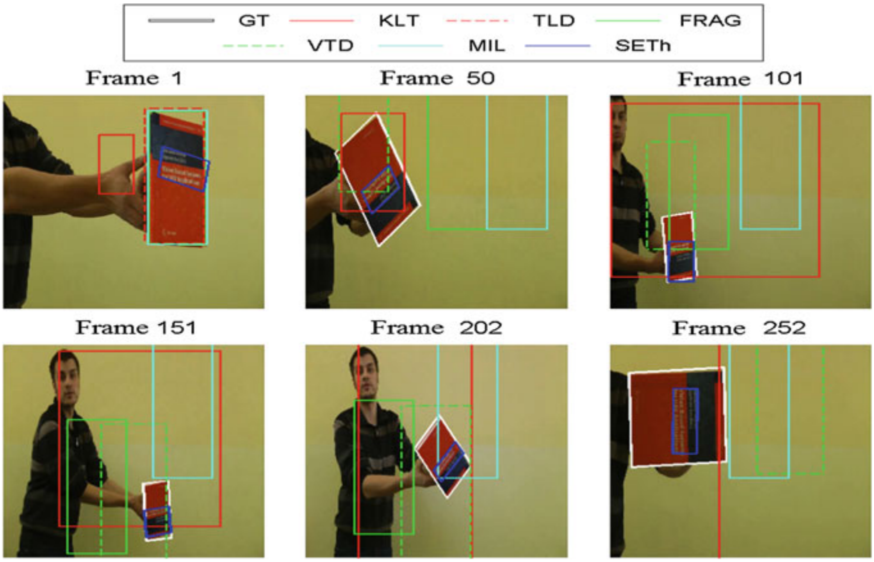


Fig. 3 Visual comparison of tracking results for rts sequence

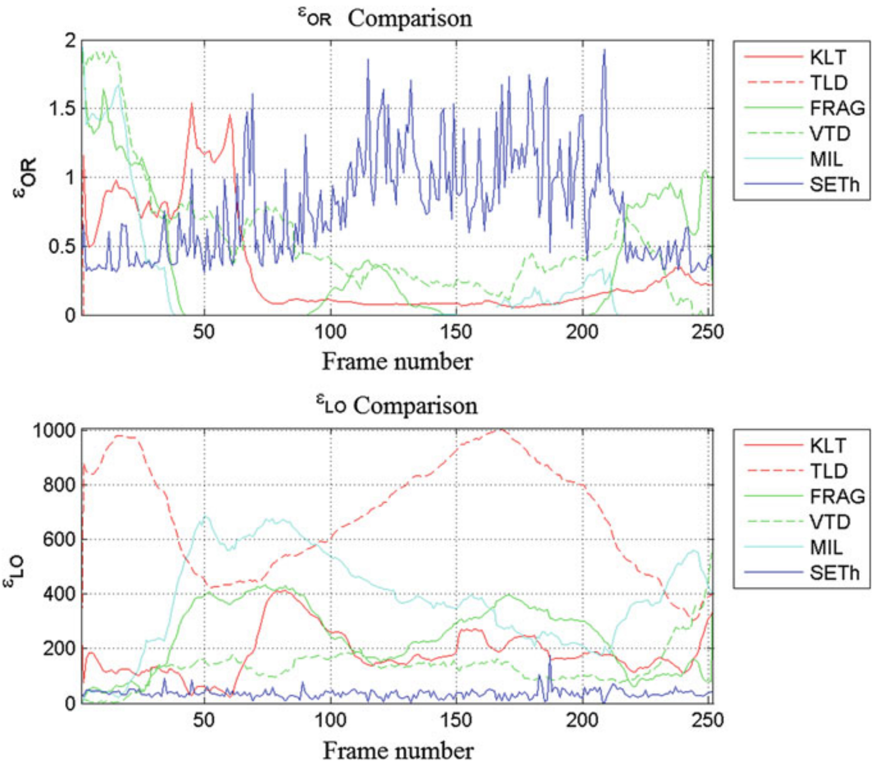


Fig. 4 Comparison of measures test sequences for rts sequence

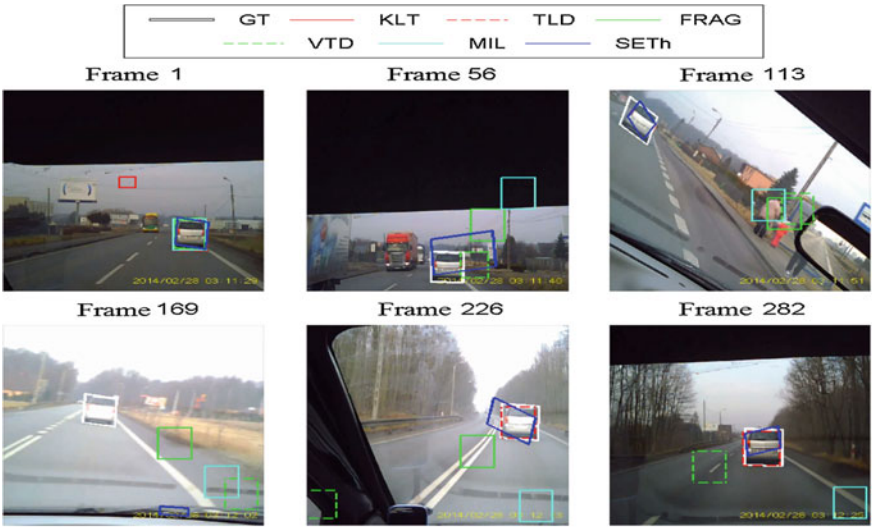


Fig. 5 Visual comparison of tracking results for car sequence

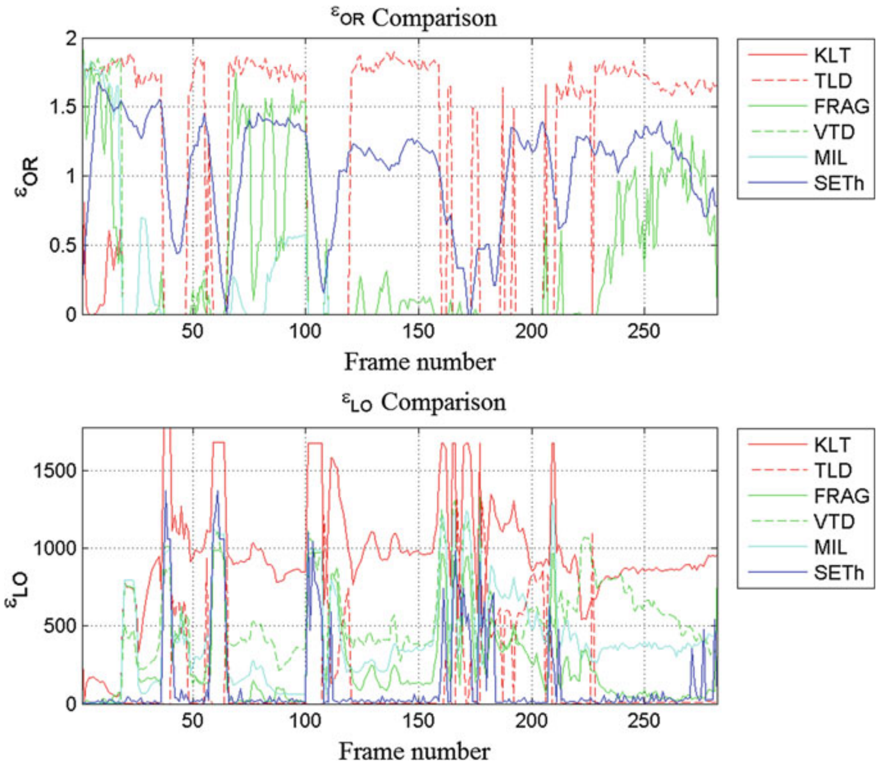


Fig. 6 Comparison of measures test sequences for car sequence

Computed measures for the car sequence are presented in the Fig. 6. It can be noticed that the best results as supposed were acquired by SETh and TLD methods, which are able to reinitialize tracking after failure (Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77 and 78; Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 and 36).

The proposed SETh method was compared in terms of computation time against the five reference state-of-the-art methods. The computed average time of processing is presented in the Fig. 79. KLT method was able to compute its output in real-time without any further optimizations. The rest of the methods achieved similar results in the range starting from 100 to 1000 ms. Indisputably the worst time of computation, which was over 6000 ms was scored by VTD method. It is worth to mention that video sequences used for measurements were 1280 × 720 pixels.

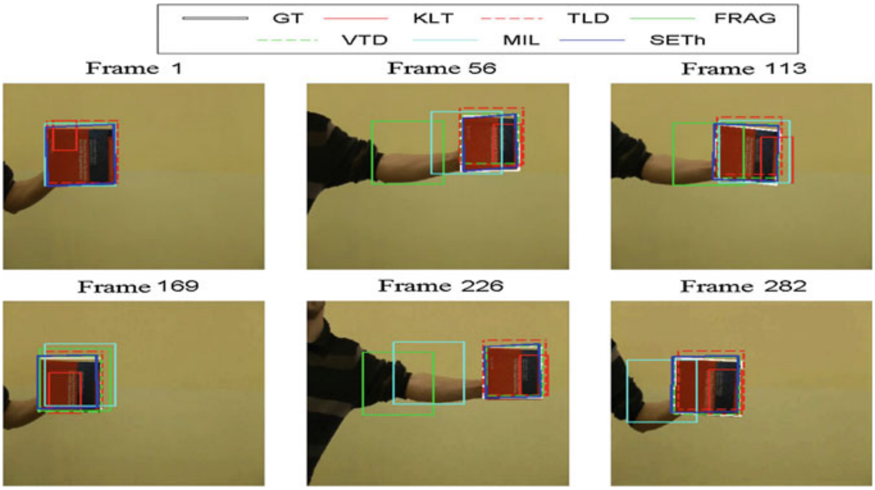


Fig. 7 Visual comparison of tracking results for test sequence

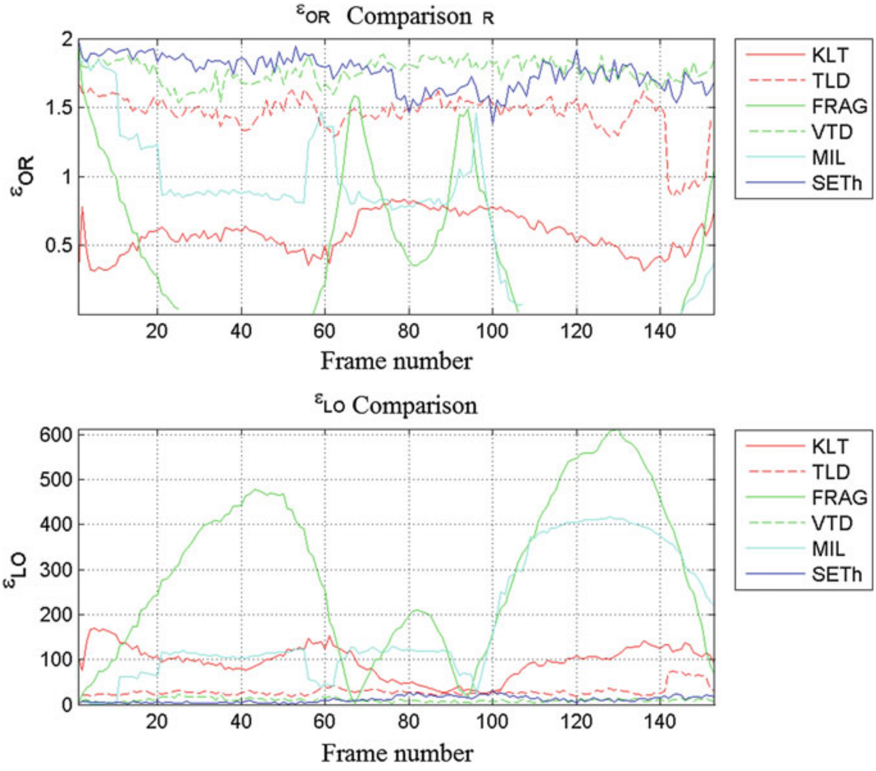


Fig. 8 Comparison of measures test sequences for test sequence

Table 1 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.73203	95.502	0.73203	0.73203
TLD	1	29.123	1	1
FRAG	0.3268	295.55	0.3268	0.3268
VTD	1	9.7787	1	1
MIL	0.65359	62.4247	0.65359	0.65359
SETh	1	10.726	1	1

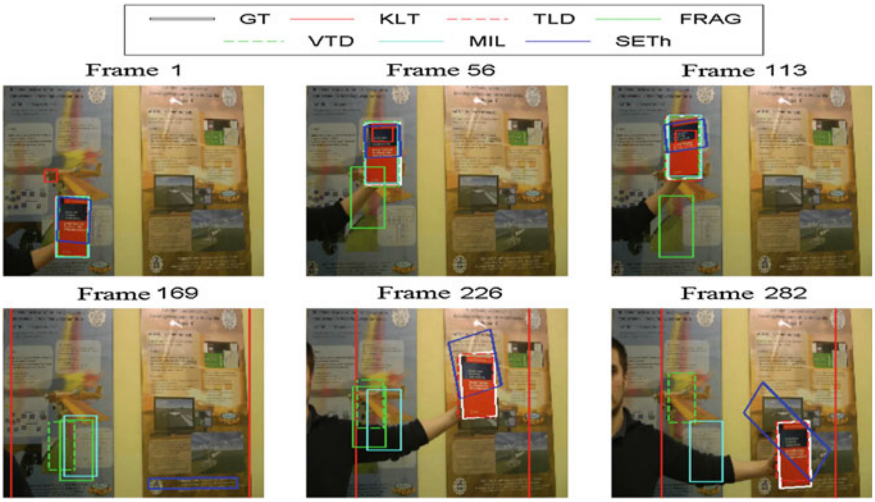


Fig. 9 Visual comparison of tracking results for test sequence

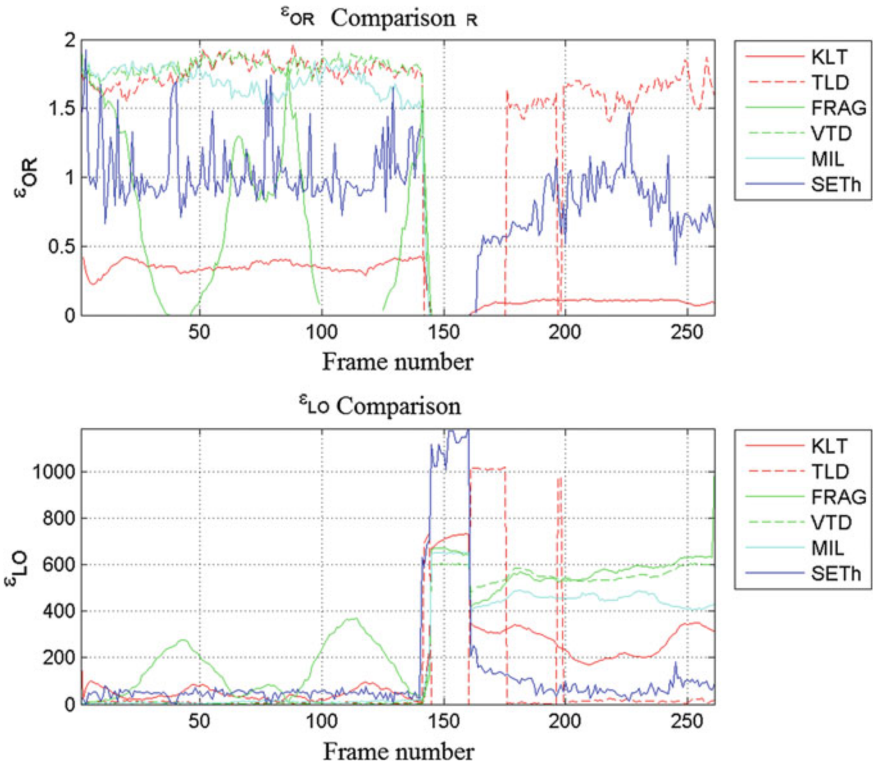


Fig. 10 Comparison of measures test sequences for test sequence

Table 2 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0	173.66	0	0
TLD	0.86207	81.185	1	0.92593
FRAG	0.27969	337.31	0.28077	0.28023
VTD	0.54406	254.35	0.54406	0.54406
MIL	0.54406	284.23	0.54406	0.54406
SETh	0.90038	135.6	0.90038	0.90038

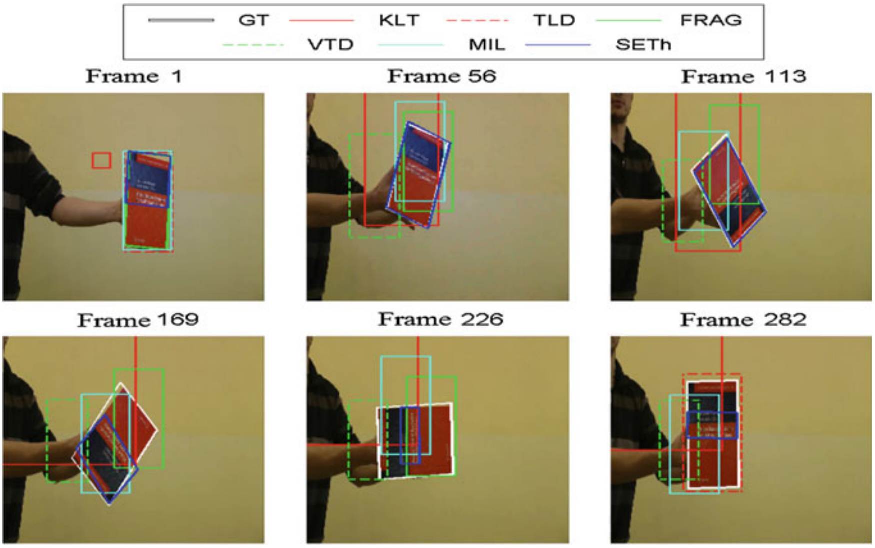


Fig. 11 Visual comparison of tracking results for test sequence

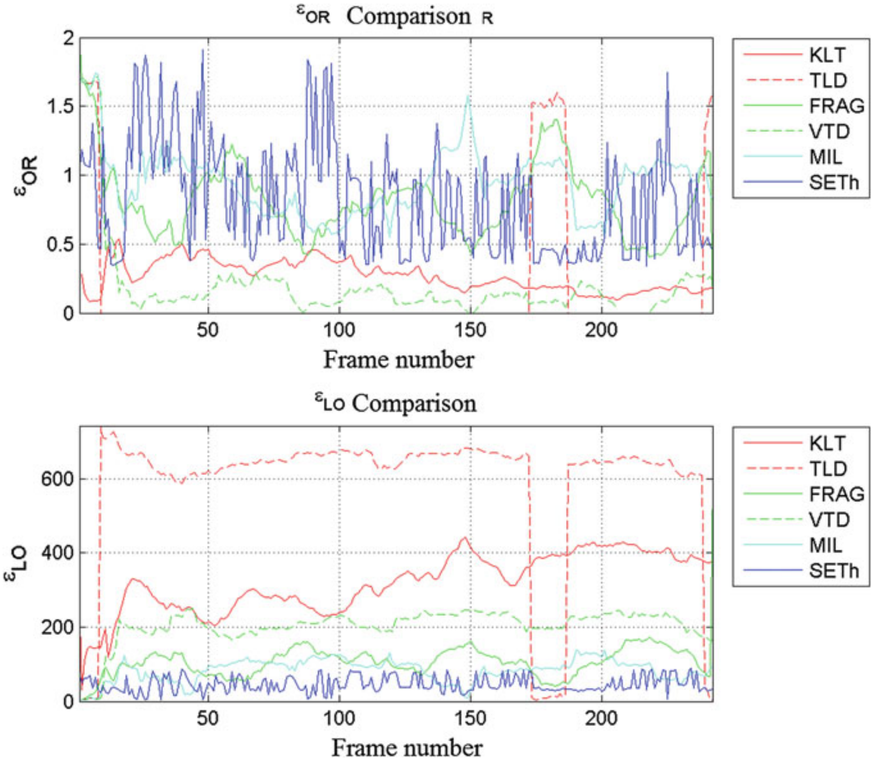


Fig. 12 Comparison of measures test sequences for test sequence

Table 3 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.012397	319.47	0.012397	0.012397
TLD	0.10744	581.54	1	0.19403
FRAG	0.88017	107.18	0.88382	0.88199
VTD	0.049587	202.65	0.049587	0.049587
MIL	1	89.59	1	1
SETh	0.67769	45.658	0.67769	0.67769

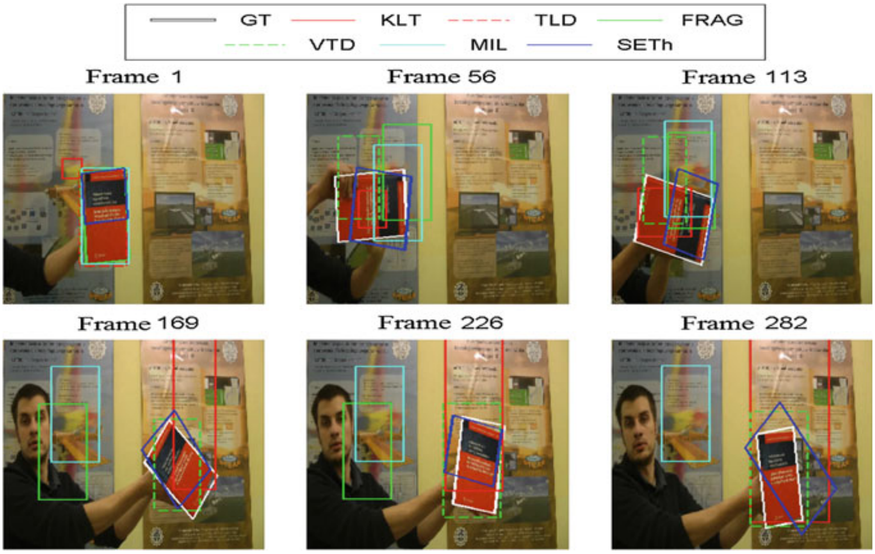


Fig. 13 Visual comparison of tracking results for test sequence

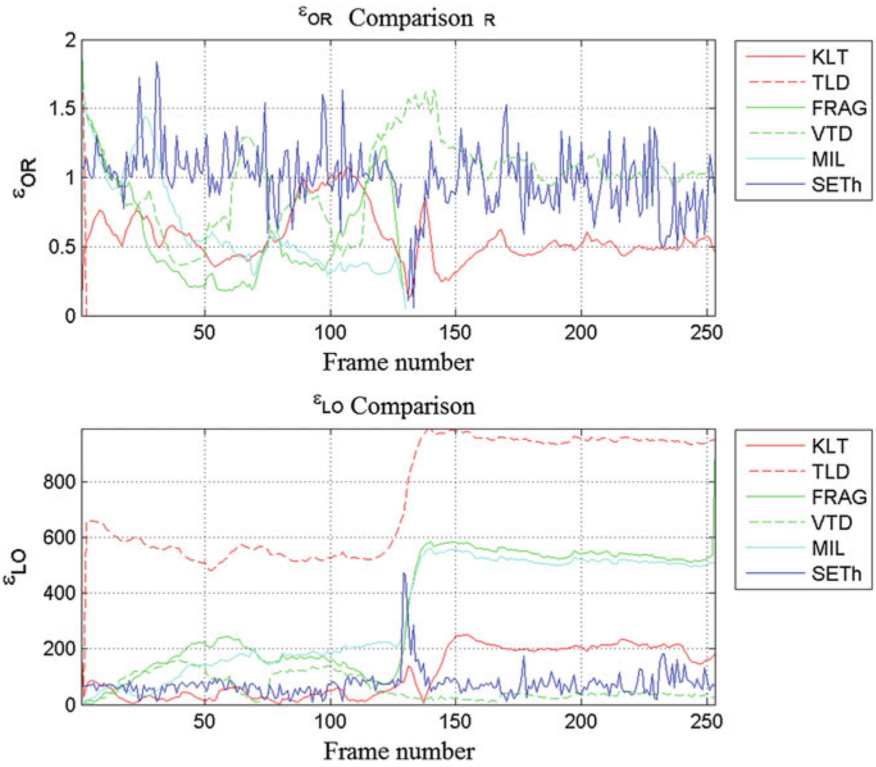


Fig. 14 Comparison of measures test sequences for test sequence

Table 4 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.58498	112.43	0.58498	0.58498
TLD	0.0079051	744.78	1	0.015686
FRAG	0.2332	338.17	0.23413	0.23366
VTD	0.917	61.202	0.917	0.917
MIL	0.27668	329.39	0.27668	0.27668
SETh	0.97628	74.423	0.97628	0.97628

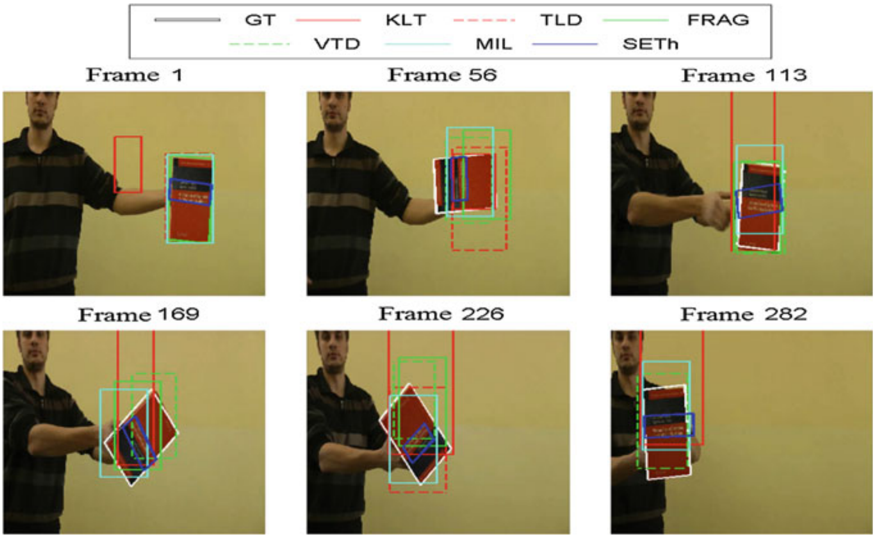


Fig. 15 Visual comparison of tracking results for test sequence

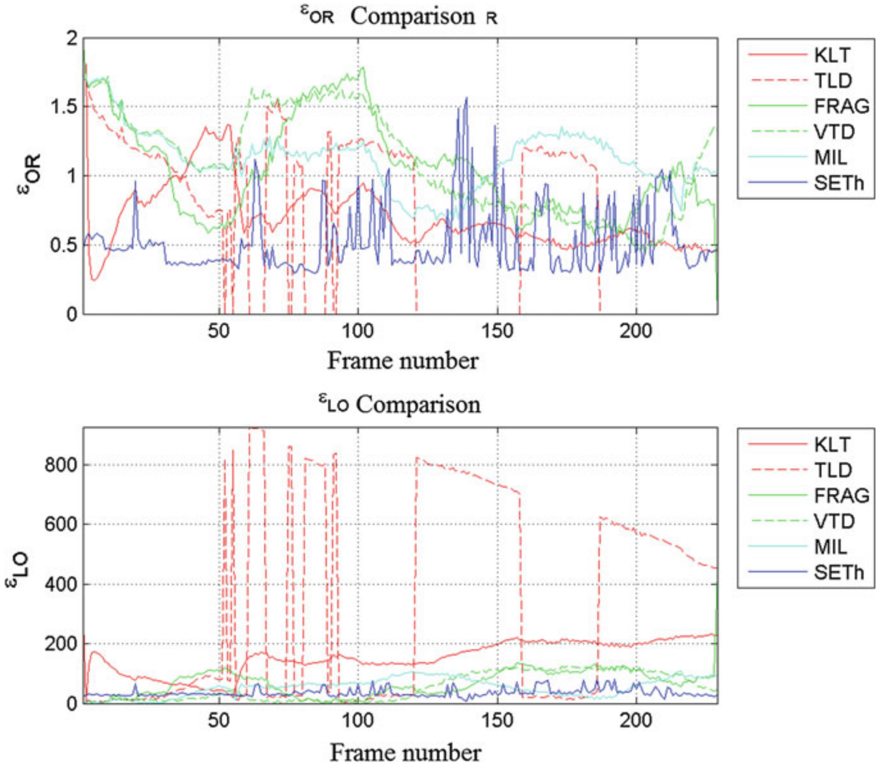


Fig. 16 Comparison of measures test sequences for test sequence

Table 5 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.84279	154.14	0.84279	0.84279
TLD	0.55895	318.76	1	0.71709
FRAG	0.99563	67.474	1	0.99781
VTD	0.9738	53.679	0.9738	0.9738
MIL	1	65.723	1	1
SETh	0.29258	35.294	0.29258	0.29258

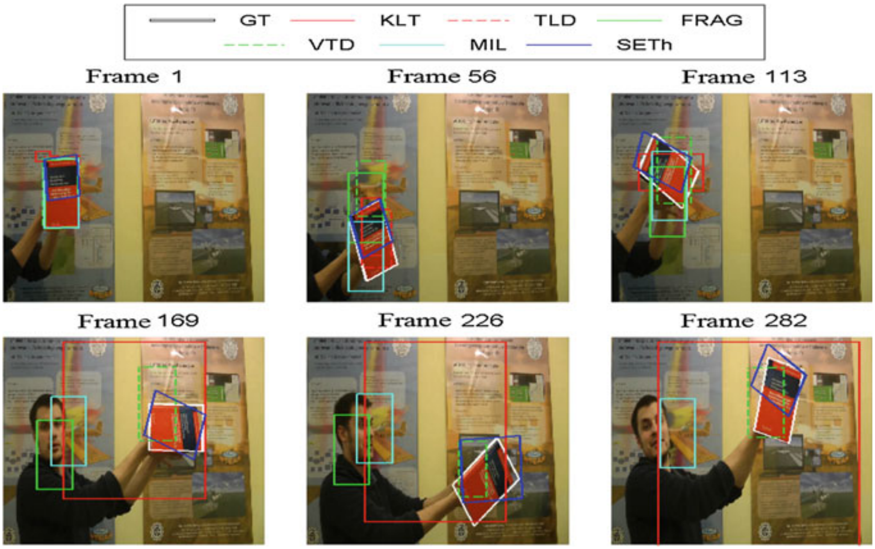


Fig. 17 Visual comparison of tracking results for test sequence

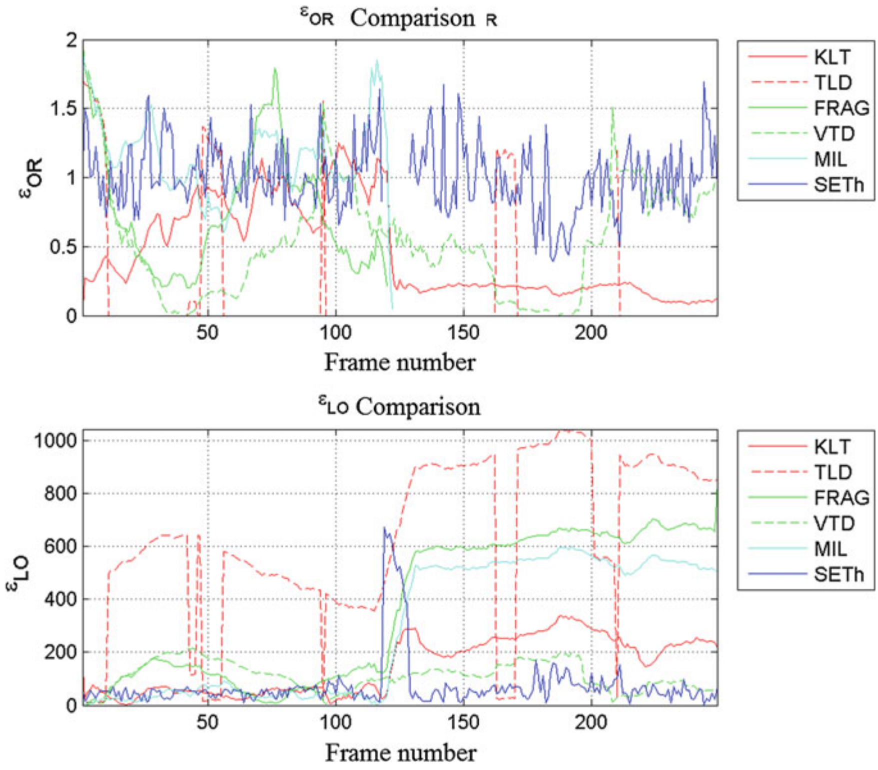


Fig. 18 Comparison of measures test sequences for test sequence

Table 6 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.38956	148.19	0.38956	0.38956
TLD	0.11245	628.59	0.7	0.19377
FRAG	0.30522	366.08	0.30645	0.30584
VTD	0.53414	111.01	0.53414	0.53414
MIL	0.48193	355.94	0.48193	0.48193
SETh	0.93574	73.023	0.93574	0.93574

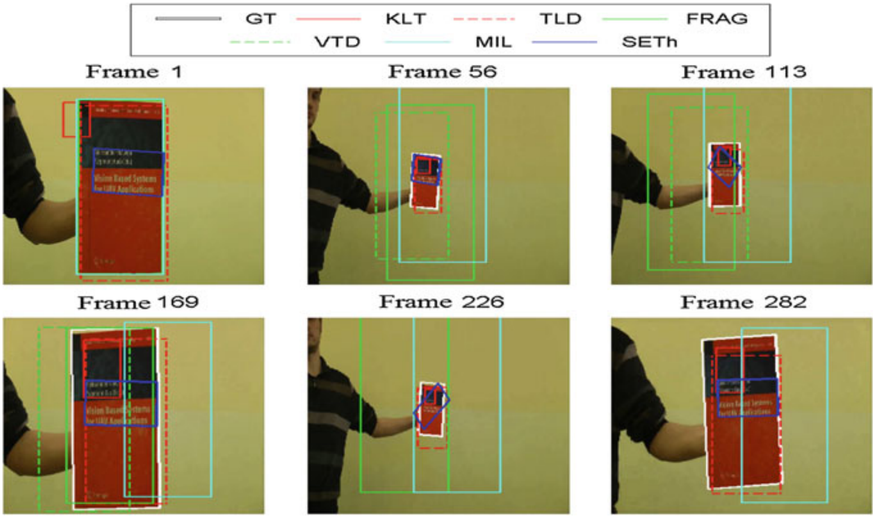


Fig. 19 Visual comparison of tracking results for scale test sequence

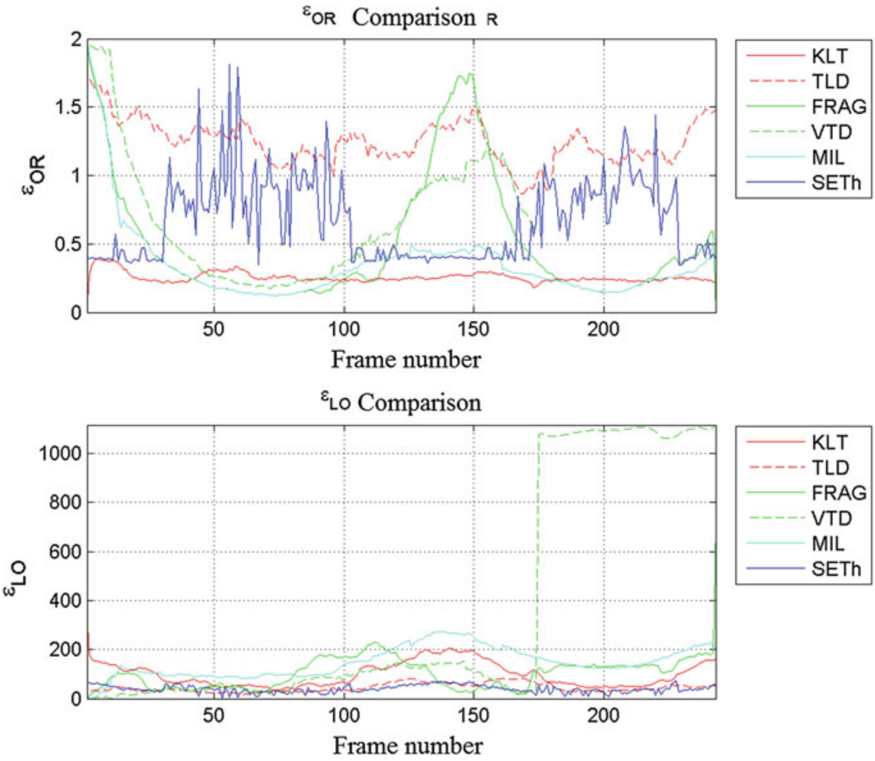


Fig. 20 Comparison of measures test sequences for scale test sequence

Table 7 Comparison of measures test sequences for scale test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0	99.073	0	0
TLD	1	44.453	1	1
FRAG	0.30453	102.96	0.30579	0.30515
VTD	0.41975	357.43	0.41975	0.41975
MIL	0.11111	133.68	0.11111	0.11111
SETh	0.53498	39.535	0.53498	0.53498

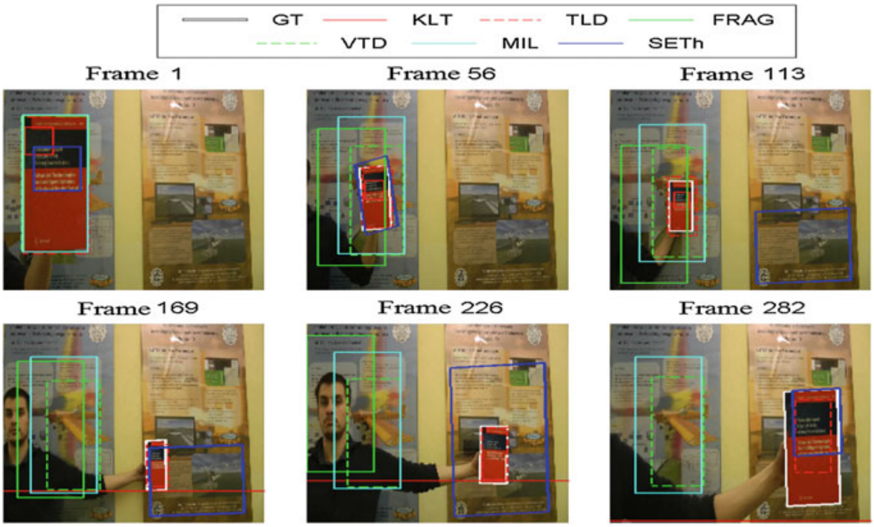


Fig. 21 Visual comparison of tracking results for scale test with textured background sequence

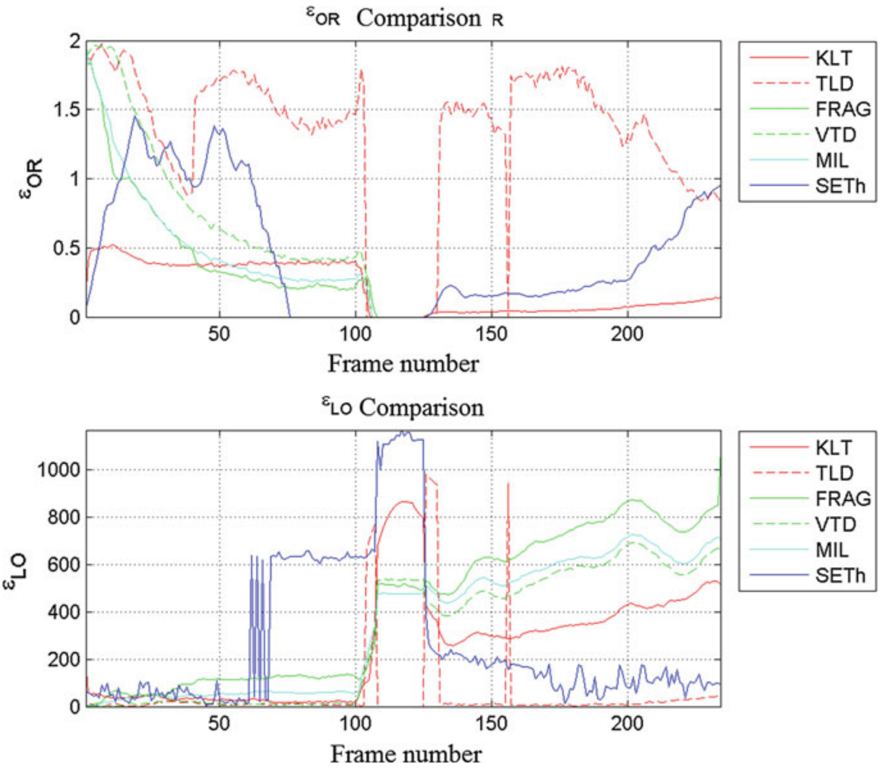


Fig. 22 Comparison of measures test sequences for scale test with textured background sequence

Table 8 Comparison of measures test sequences for scale test with textured background sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.021368	252.75	0.021368	0.021368
TLD	0.88034	47.127	1	0.93636
FRAG	0.15812	415.47	0.2	0.17661
VTD	0.26923	307.47	0.26923	0.26923
MIL	0.16667	392.63	0.16667	0.16667
SETh	0.38889	276.83	0.38889	0.38889

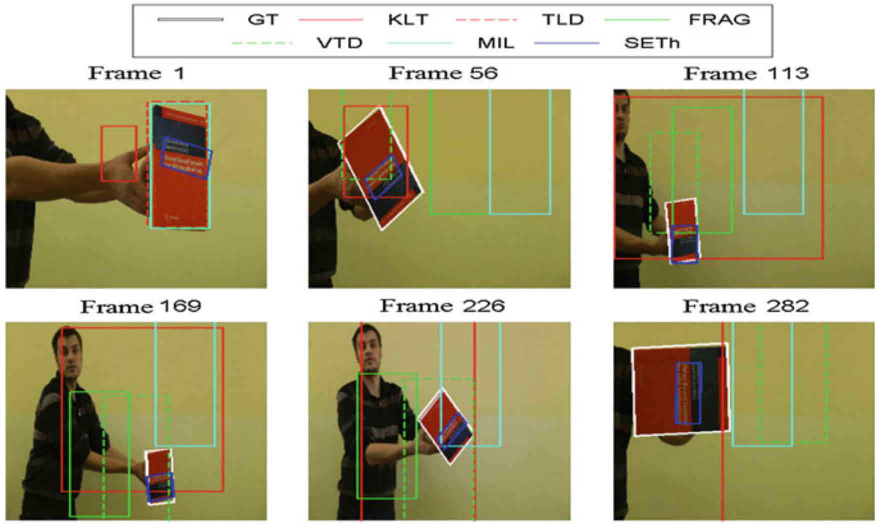


Fig. 23 Visual comparison of tracking results for rotation test sequence

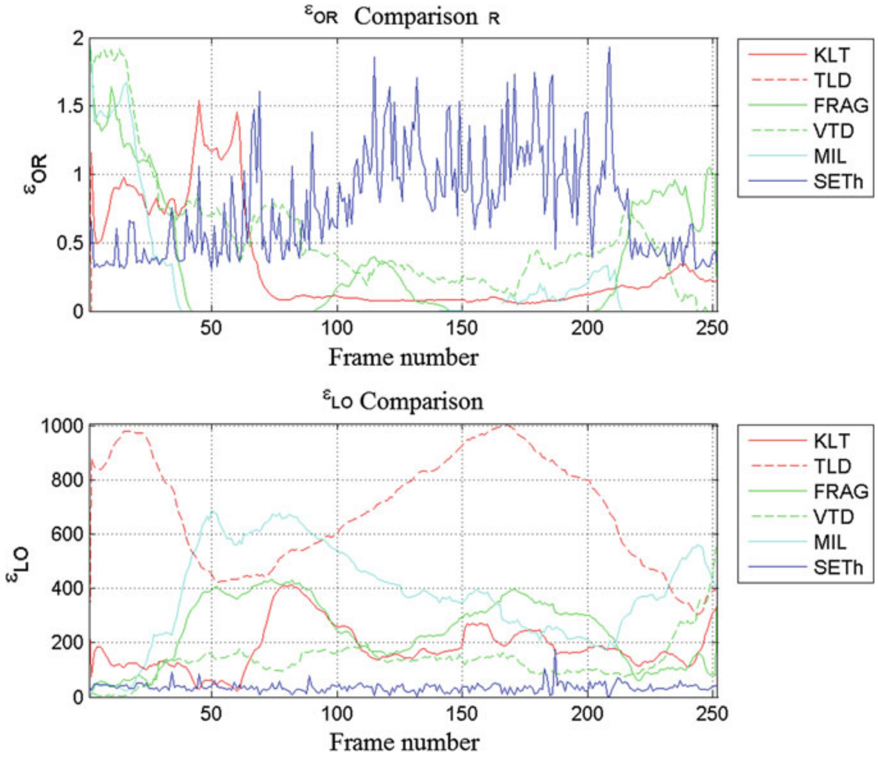


Fig. 24 Comparison of measures test sequences for rotation test sequence

Table 9 Comparison of measures test sequences for rotation test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.25	183.71	0.25	0.25
TLD	0.0039683	689.29	1	0.0079051
FRAG	0.28175	243.13	0.28287	0.28231
VTD	0.38095	134.31	0.38095	0.38095
MIL	0.10317	381.85	0.10317	0.10317
SETh	0.65476	35.687	0.65476	0.65476

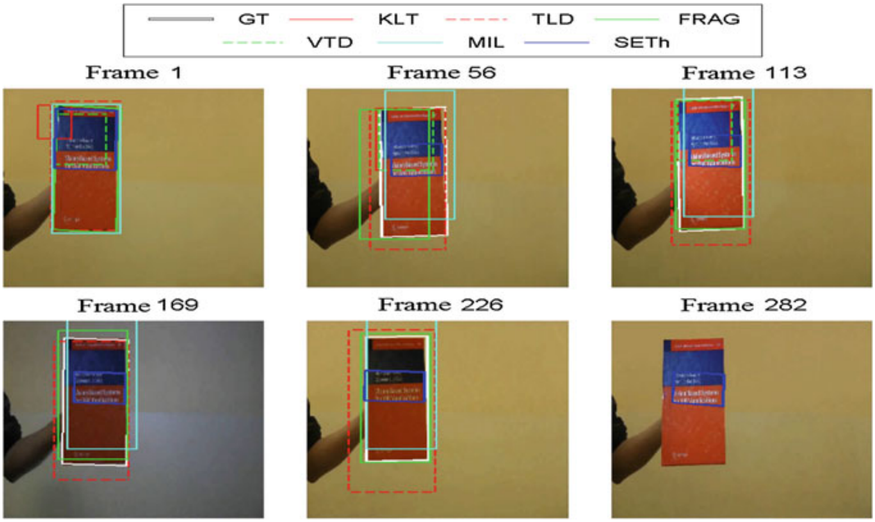


Fig. 25 Visual comparison of tracking results for lighting conditions test sequence

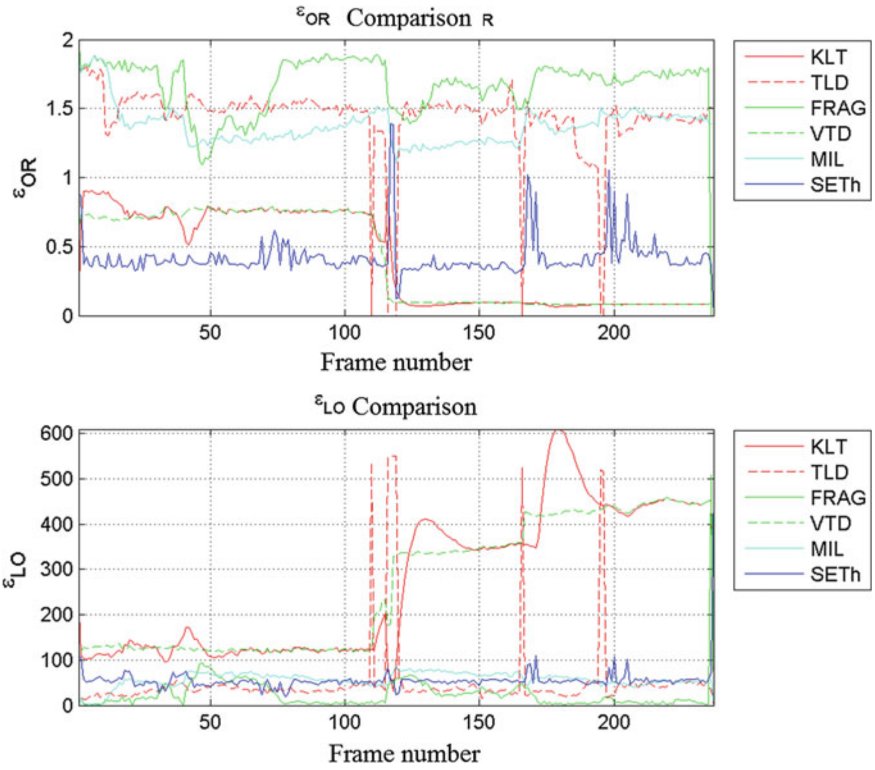


Fig. 26 Comparison of measures test sequences for lighting conditions test sequence

Table 10 Comparison of measures test sequences for lighting conditions test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.48945	269.05	0.49153	0.49049
TLD	0.96203	51.844	1	0.98065
FRAG	0.99156	24.363	1	0.99576
VTD	0.47679	263.38	0.47881	0.4778
MIL	0.99578	63.632	1	0.99789
SETh	0.092827	55.855	0.092827	0.092827

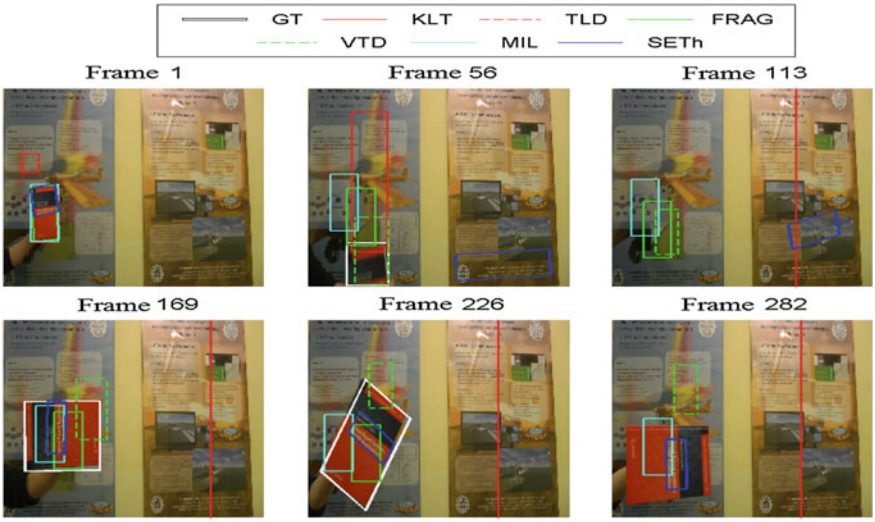


Fig. 27 Visual comparison of tracking results for rotation test sequence with textured background

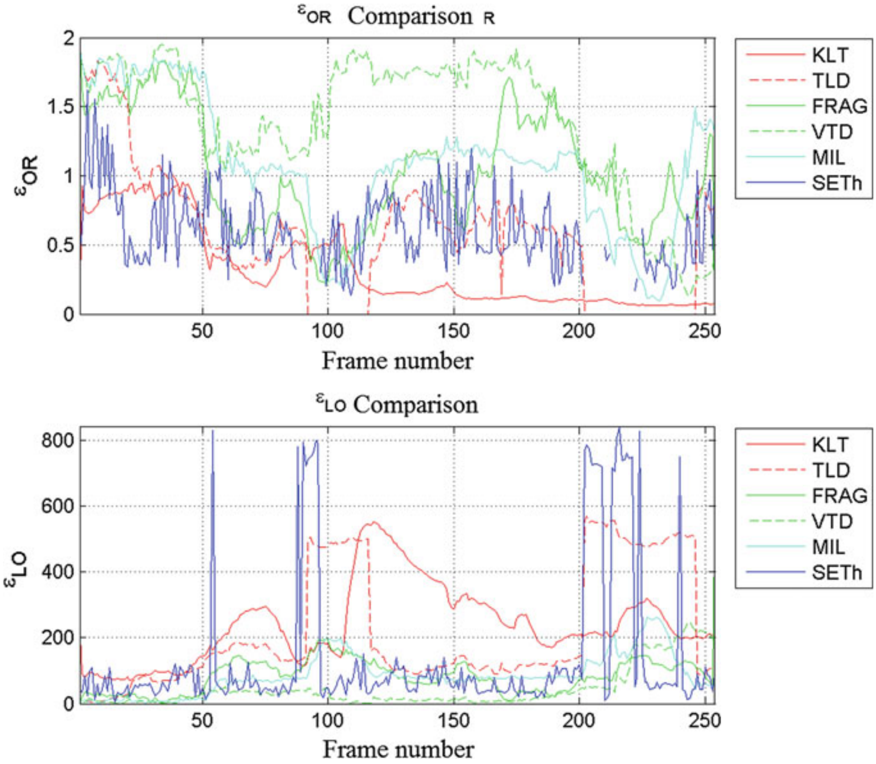


Fig. 28 Comparison of measures test sequences for rotation test sequence with textured background

Table 11 Comparison of measures test sequences for rotation test sequence with textured background

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.17004	314.2	0.17004	0.17004
TLD	0.18623	246.52	1	0.31399
FRAG	0.40486	251.37	0.4065	0.40568
VTD	0.20648	289.03	0.20648	0.20648
MIL	0.24696	197.42	0.24696	0.24696
SETh	0.12955	563.52	0.12955	0.12955

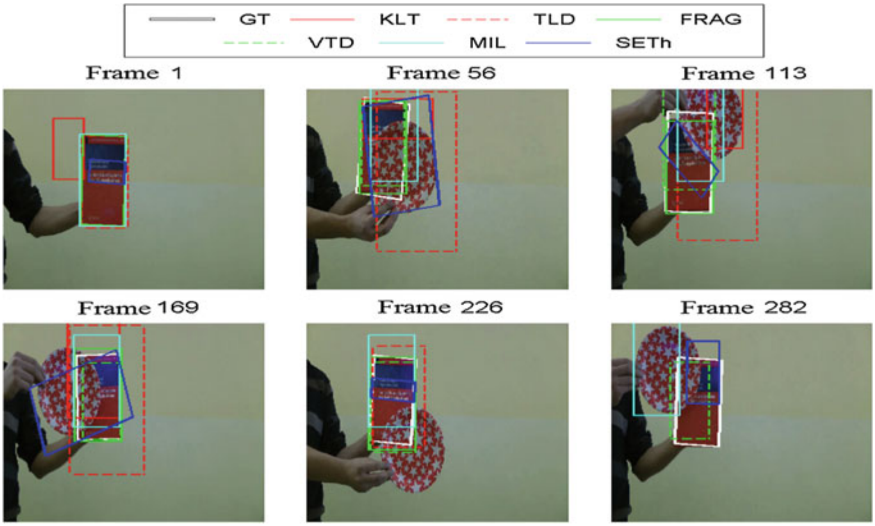


Fig. 29 Visual comparison of tracking results for test sequence occlusion

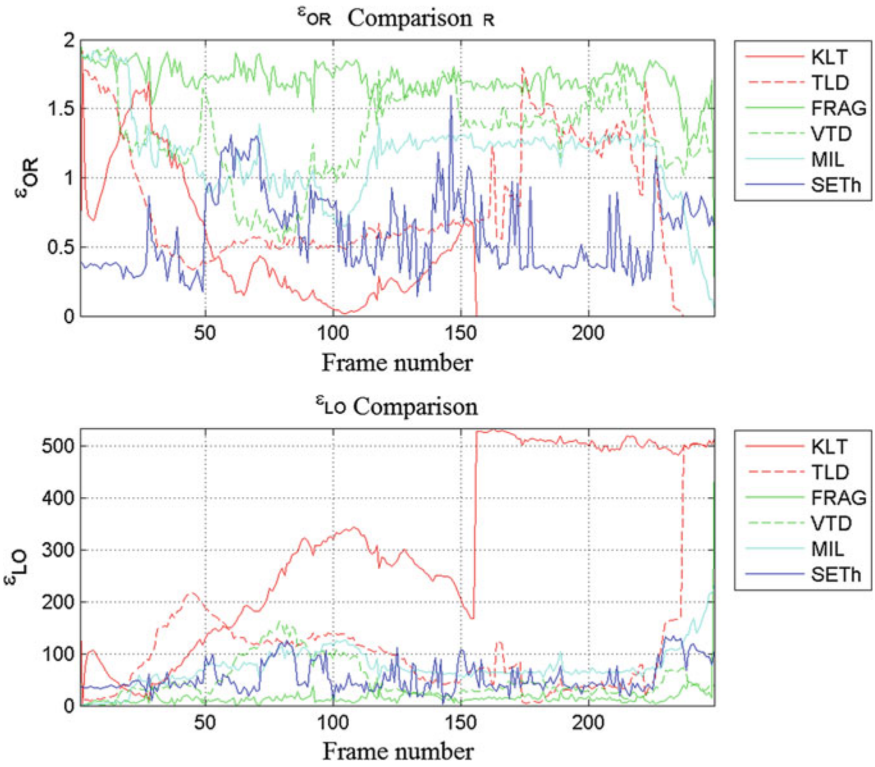


Fig. 30 Comparison of measures test sequences for test sequence occlusion

Table 12 Comparison of measures test sequences for test sequence occlusion

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.24096	312.04	0.3871	0.29703
TLD	0.77912	109.8	0.82203	0.8
FRAG	0.99598	16.056	1	0.99799
VTD	1	47.733	1	1
MIL	0.95984	63.693	0.95984	0.95984
SETh	0.51004	54.066	0.51004	0.51004

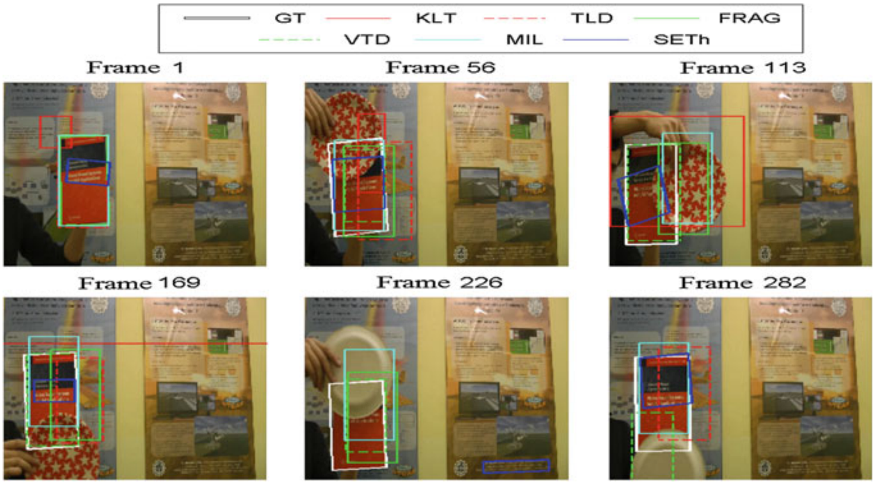


Fig. 31 Visual comparison of tracking results for test sequence occlusion with textured background

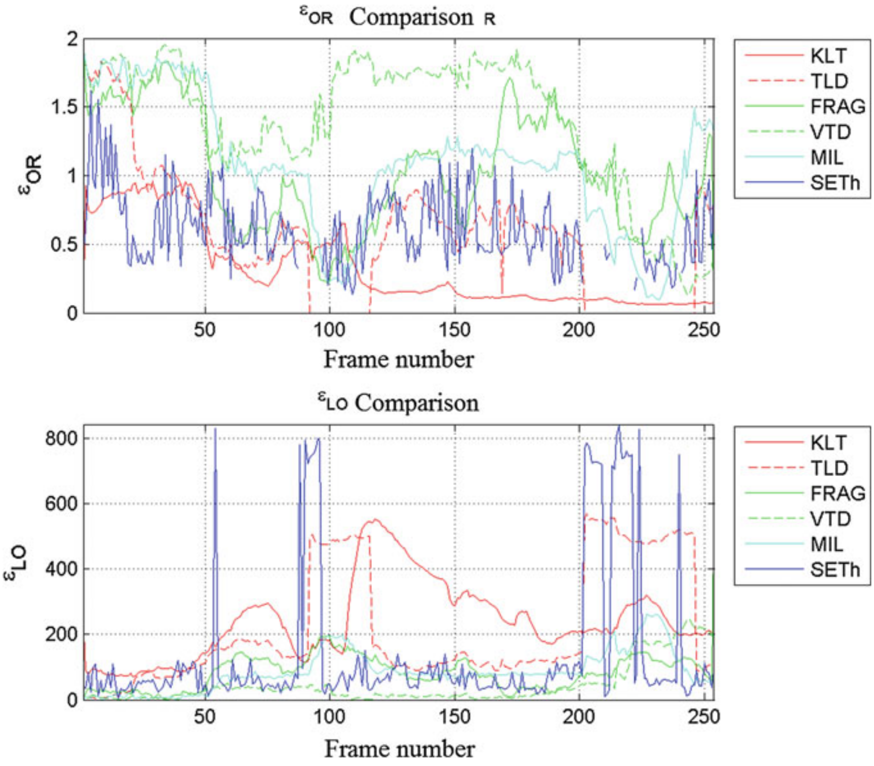


Fig. 32 Comparison of measures test sequences for test sequence occlusion with textured background

Table 13 Comparison of measures test sequences for test sequence occlusion with textured background

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.24803	238.73	0.25	0.24901
TLD	0.5748	217.4	0.79348	0.66667
FRAG	0.91339	83.357	0.917	0.91519
VTD	0.90157	44.599	0.90157	0.90157
MIL	0.85827	78.84	0.85827	0.85827
SETh	0.50394	137.22	0.50394	0.50394

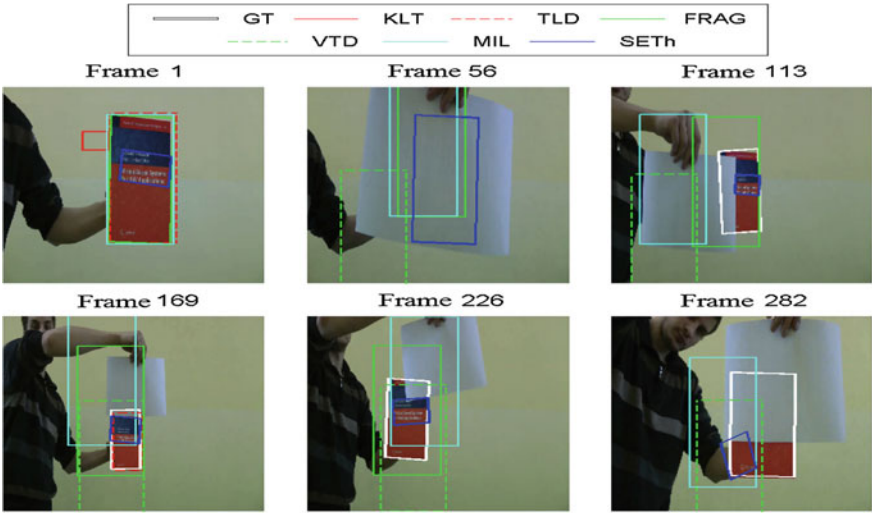


Fig. 33 Visual comparison of tracking results for test sequence full occlusion

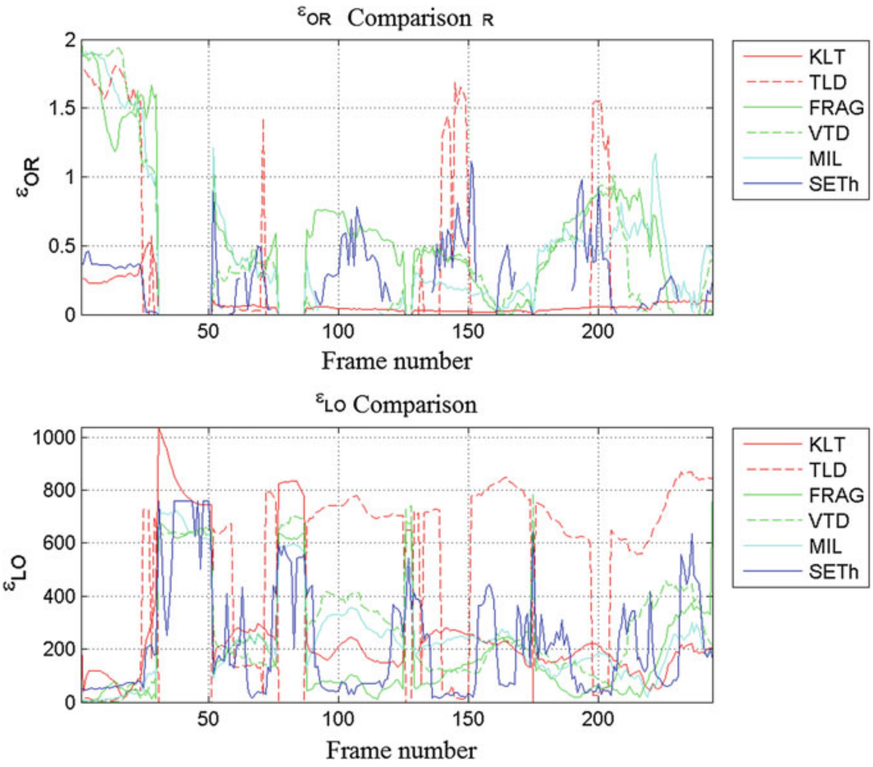


Fig. 34 Comparison of measures test sequences for test sequence full occlusion

Table 14 Comparison of measures test sequences for test sequence full occlusion

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.0081967	276.87	0.0081967	0.0081967
TLD	0.17623	463.49	0.76786	0.28667
FRAG	0.45492	199.69	0.45679	0.45585
VTD	0.2418	274.23	0.2418	0.2418
MIL	0.30738	264.18	0.30738	0.30738
SETh	0.11885	223.14	0.11885	0.11885

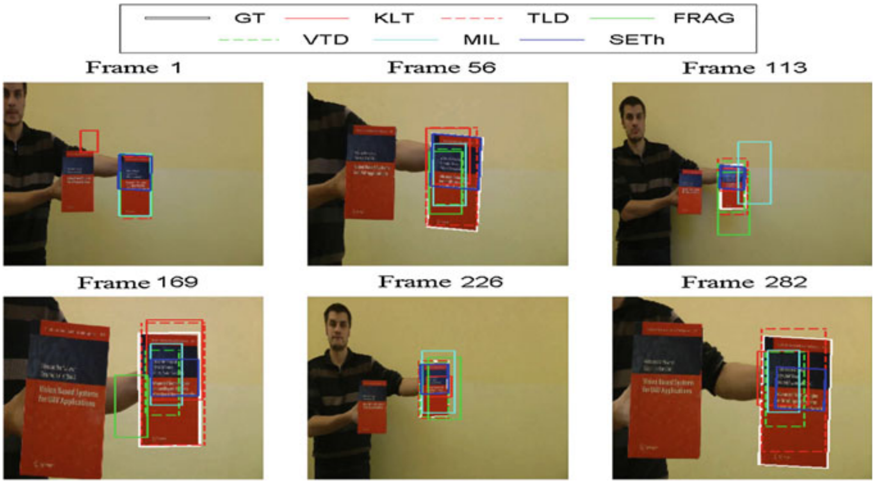


Fig. 35 Visual comparison of tracking results for test sequence

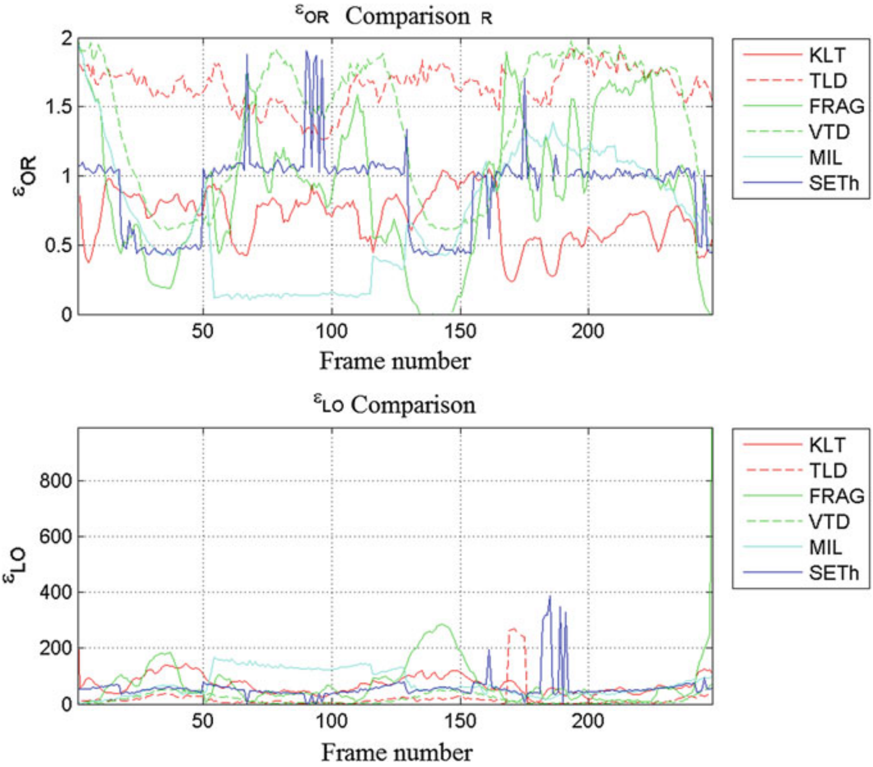


Fig. 36 Comparison of measures test sequences for test sequence

Table 15 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.83871	69.571	0.83871	0.83871
TLD	0.97177	19.783	0.97177	0.97177
FRAG	0.77823	75.371	0.78138	0.7798
VTD	1	17.212	1	1
MIL	0.58871	0.28053	0.58871	0.58871
SETh	0.75806	58.055	0.75806	0.75806

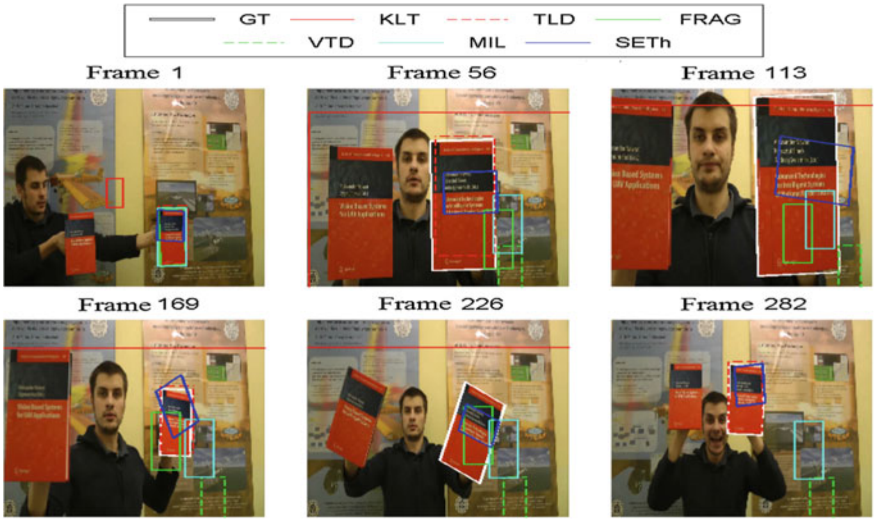


Fig. 37 Visual comparison of tracking results for test sequence

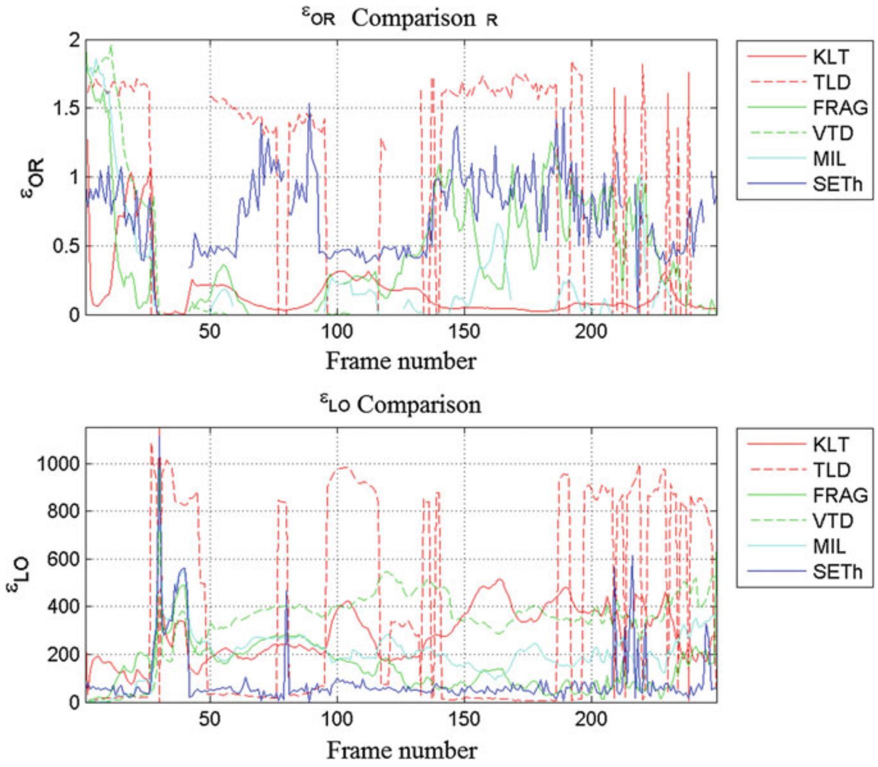


Fig. 38 Comparison of measures test sequences for test sequence

Table 16 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.064257	287.23	0.064257	0.064257
TLD	0.54217	383.72	0.89404	0.675
FRAG	0.34137	148.75	0.34274	0.34205
VTD	0.11245	340.69	0.11245	0.11245
MIL	0.11245	162.95	0.11245	0.11245
SETh	0.61044	86.789	0.61044	0.61044

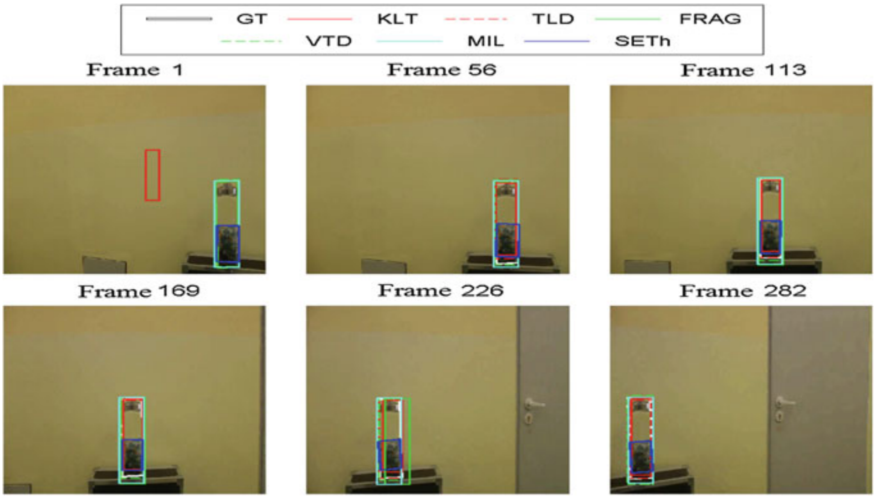


Fig. 39 Visual comparison of tracking results for test sequence

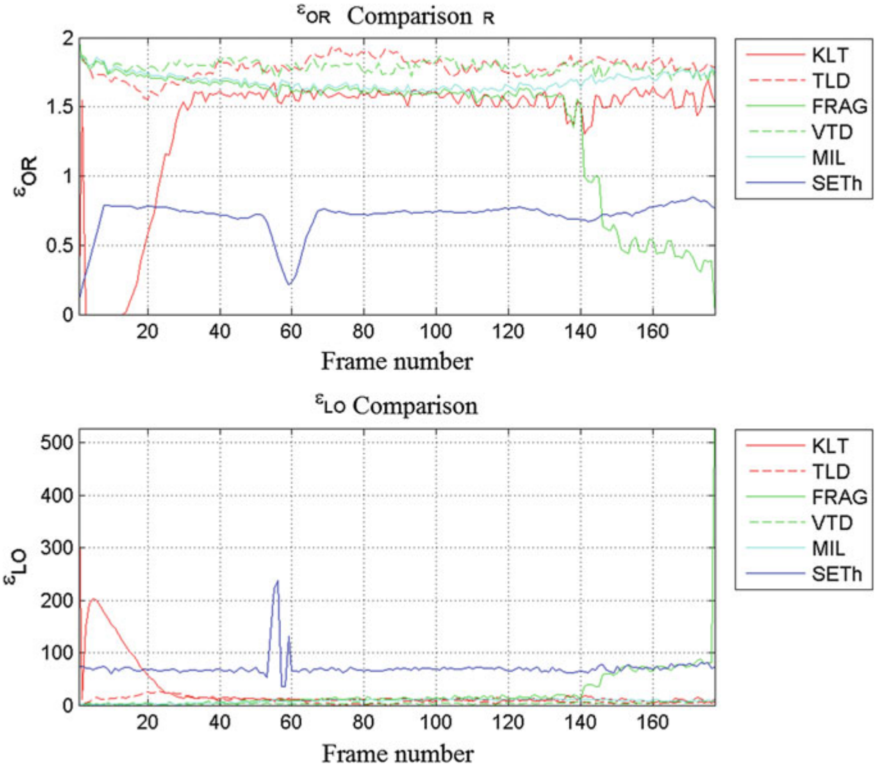


Fig. 40 Comparison of measures test sequences for test sequence

Table 17 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.89831	26.863	0.89831	0.89831
TLD	1	8.4713	1	1
FRAG	0.88136	24.255	0.88636	0.88385
VTD	1	3.3256	1	1
MIL	1	27.53	1	1
SETh	0.96045	70.834	0.96045	0.96045

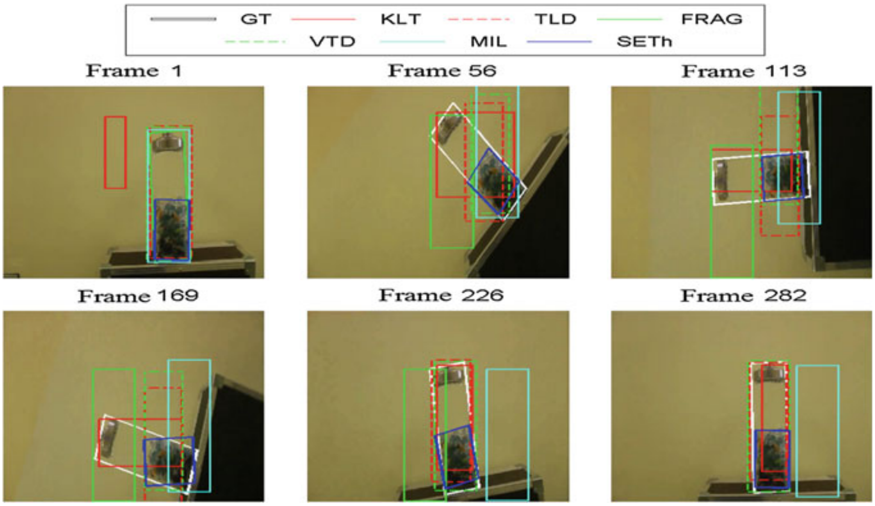


Fig. 41 Visual comparison of tracking results for test sequence

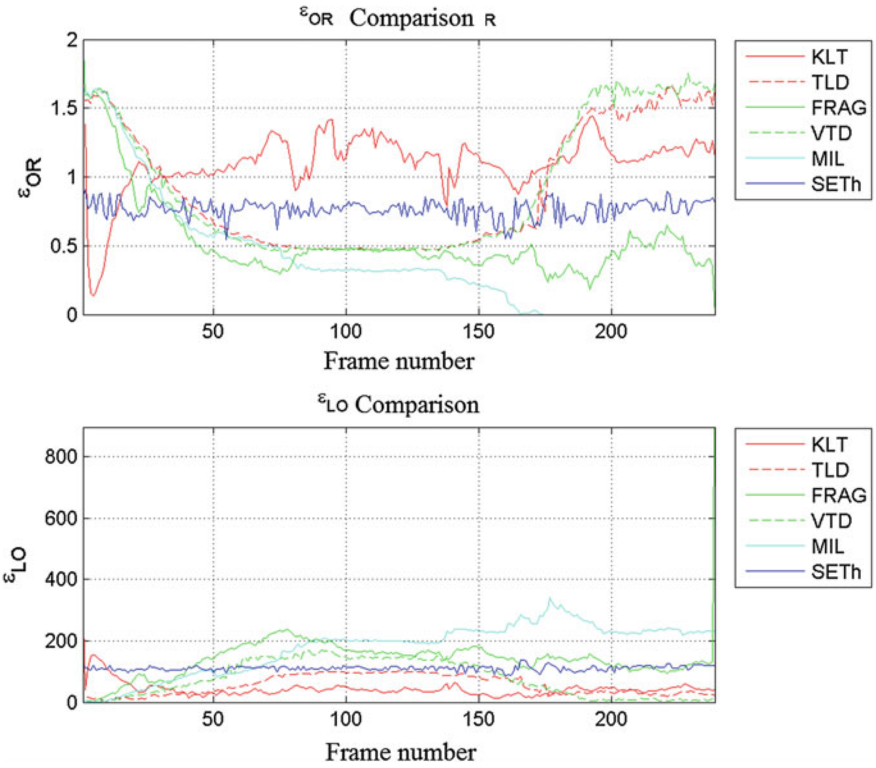


Fig. 42 Comparison of measures test sequences for test sequence

Table 18 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.96653	43.376	0.96653	0.96653
TLD	0.7364	54.515	0.7364	0.7364
FRAG	0.2636	140.45	0.26471	0.26415
VTD	0.68619	76.356	0.68619	0.68619
MIL	0.29707	194.71	0.29707	0.29707
SETh	1	110.93	1	1

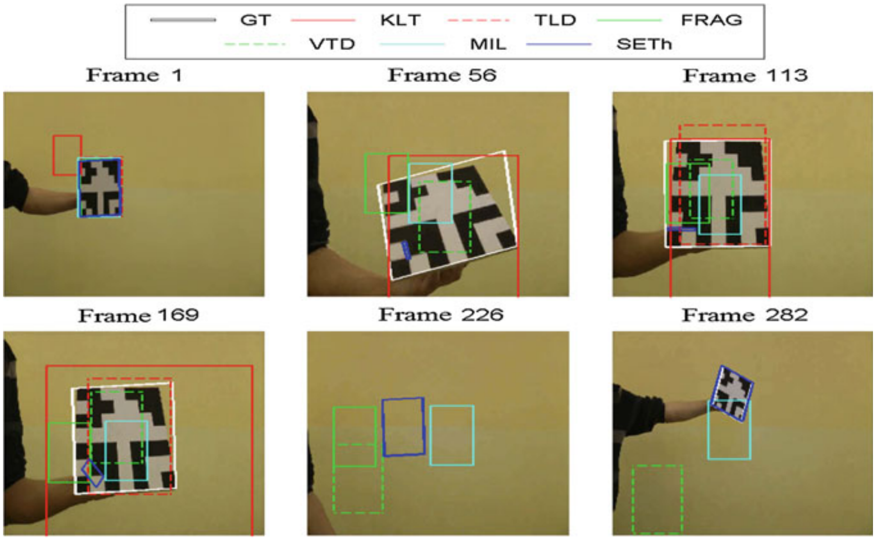


Fig. 43 Visual comparison of tracking results for test sequence

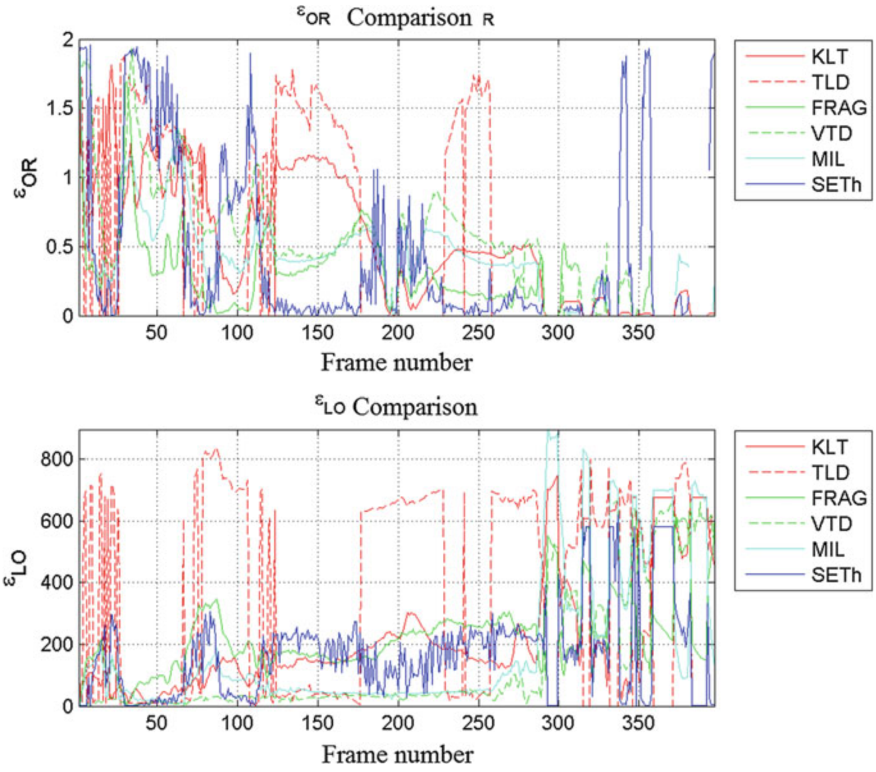


Fig. 44 Comparison of measures test sequences for test sequence

Table 19 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.43073	231.41	0.43073	0.43073
TLD	0.38539	333.08	1	0.55636
FRAG	0.15869	223.8	0.17166	0.16492
VTD	0.5063	143.38	0.51015	0.50822
MIL	0.30479	294.58	0.30479	0.30479
SETh	0.25945	167.26	0.27321	0.26615

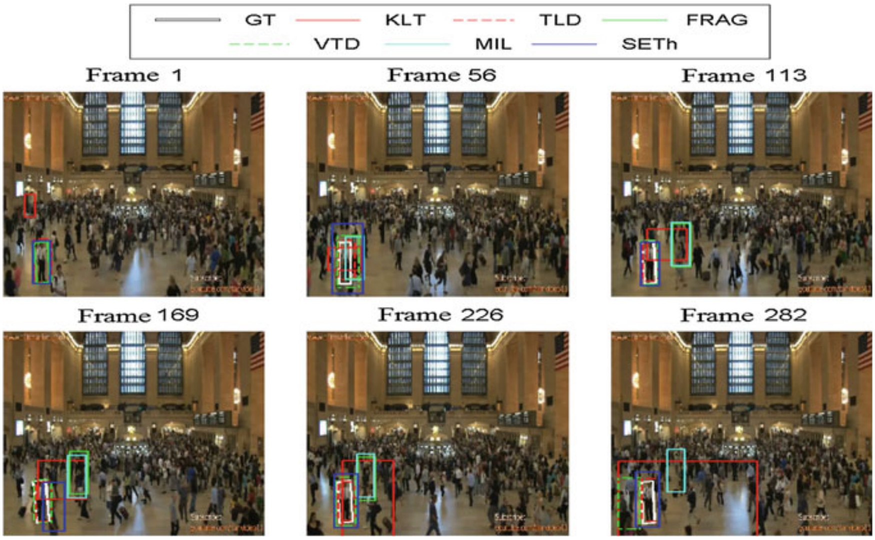


Fig. 45 Visual comparison of tracking results for test sequence

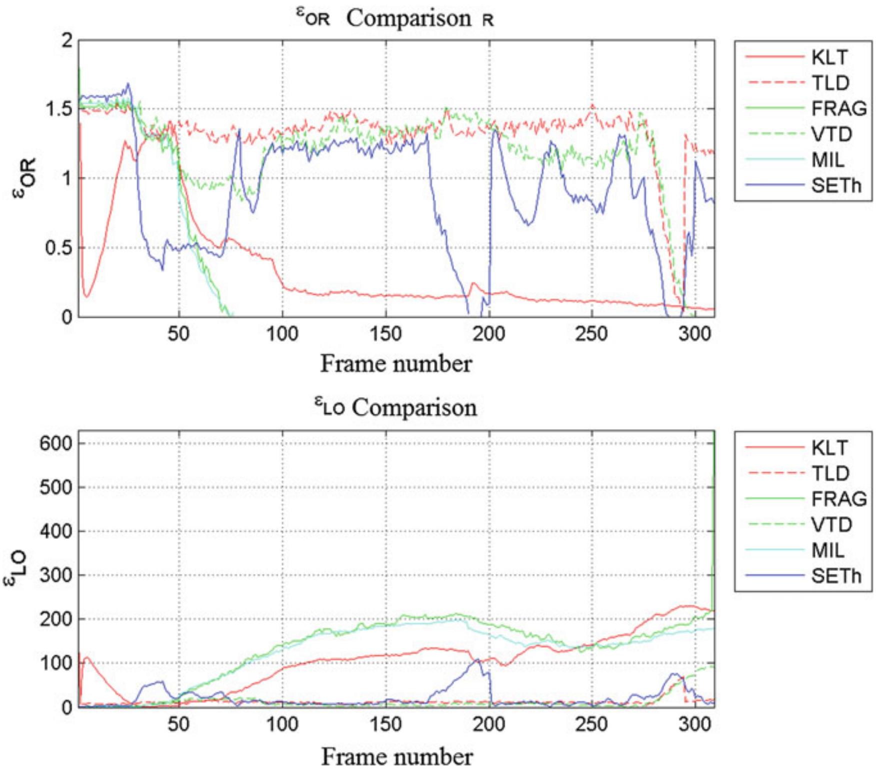


Fig. 46 Comparison of measures test sequences for test sequence

Table 20 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.21359	106.35	0.21359	0.21359
TLD	0.97735	12.041	0.97735	0.97735
FRAG	0.1877	134.07	0.18831	0.18801
VTD	0.9288	12.215	0.9288	0.9288
MIL	0.17799	129.67	0.17799	0.17799
SETh	0.79612	21.23	0.79612	0.79612

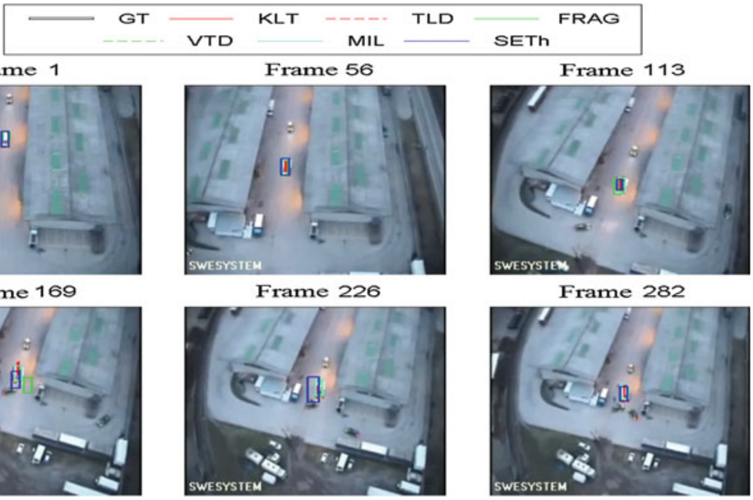


Fig. 47 Visual comparison of tracking results for test sequence

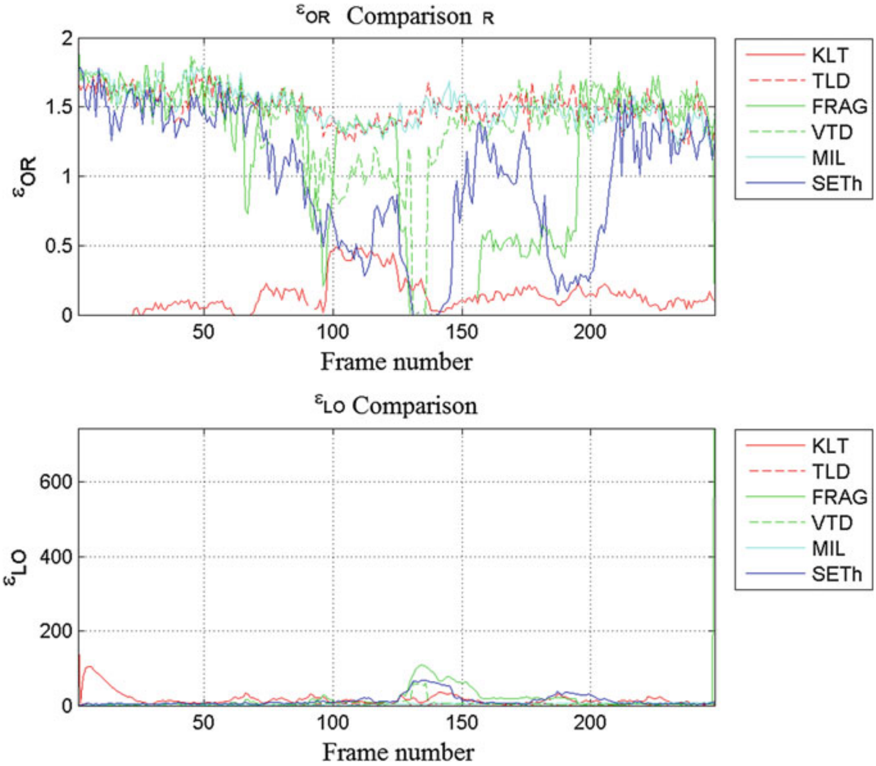


Fig. 48 Comparison of measures test sequences for test sequence

Table 21 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0	17.56	0	0
TLD	1	3.139	1	1
FRAG	0.81048	17.527	0.81377	0.81212
VTD	0.97581	5.3883	0.97581	0.97581
MIL	1	15.84	1	1
SETh	0.79032	13.125	0.79032	0.79032

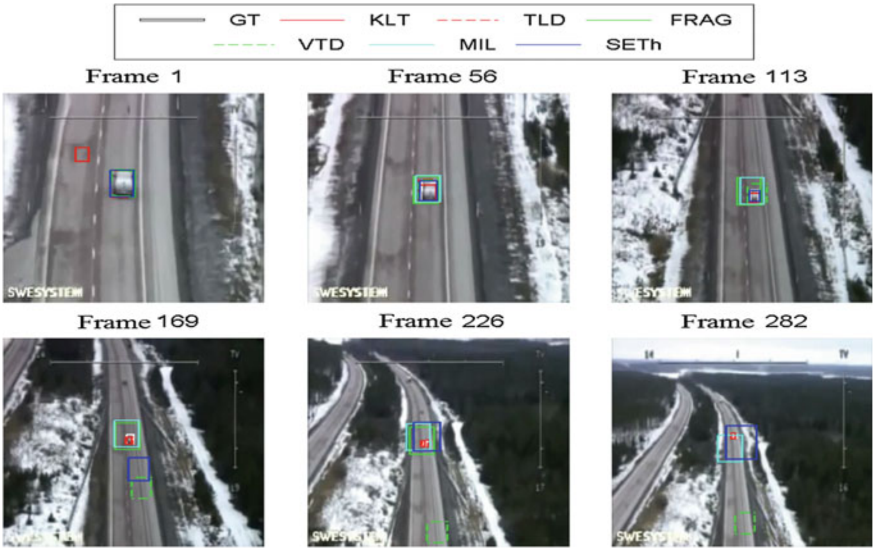


Fig. 49 Visual comparison of tracking results for test sequence

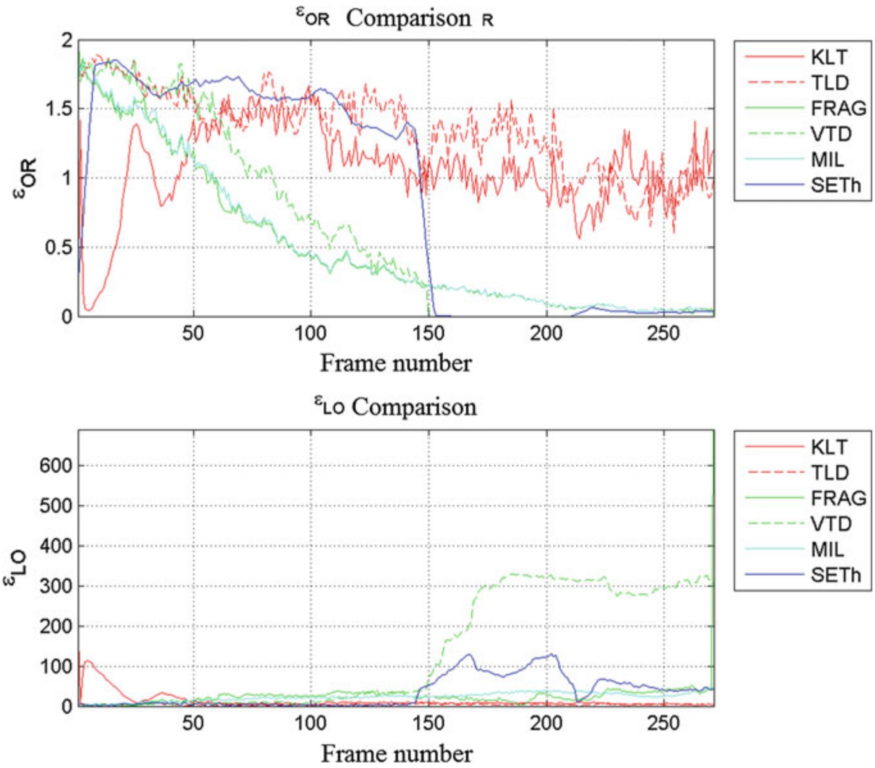


Fig. 50 Comparison of measures test sequences for test sequence

Table 22 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.94465	13.52	0.94465	0.94465
TLD	1	3.8964	1	1
FRAG	0.33948	25.567	0.34074	0.34011
VTD	0.43542	134.43	0.43542	0.43542
MIL	0.33948	29.453	0.33948	0.33948
SETh	0.53506	34.221	0.53506	0.53506

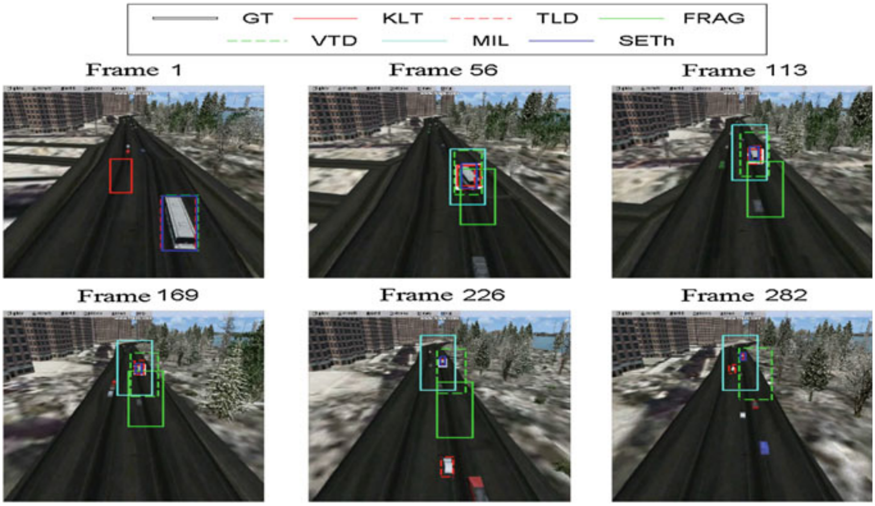


Fig. 51 Visual comparison of tracking results for test sequence

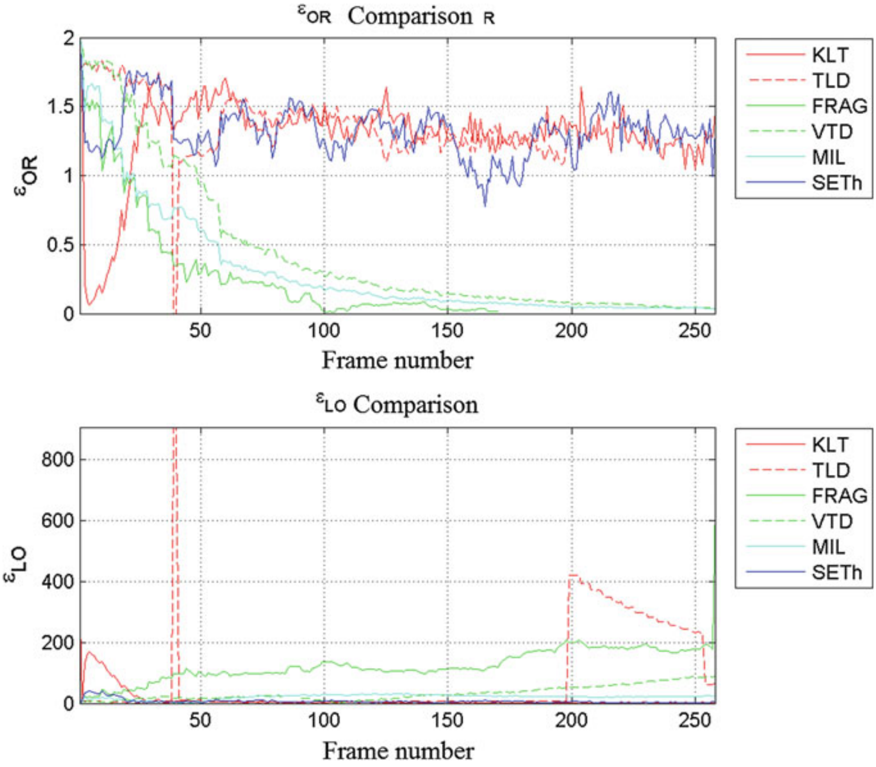


Fig. 52 Comparison of measures test sequences for test sequence

Table 23 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.94186	12.568	0.94186	0.94186
TLD	0.75969	78.659	0.76563	0.76265
FRAG	0.12791	124.43	0.1284	0.12816
VTD	0.26744	32.436	0.26744	0.26744
MIL	0.22093	84.986	0.22093	0.22093
SETh	1	6.3975	1	1

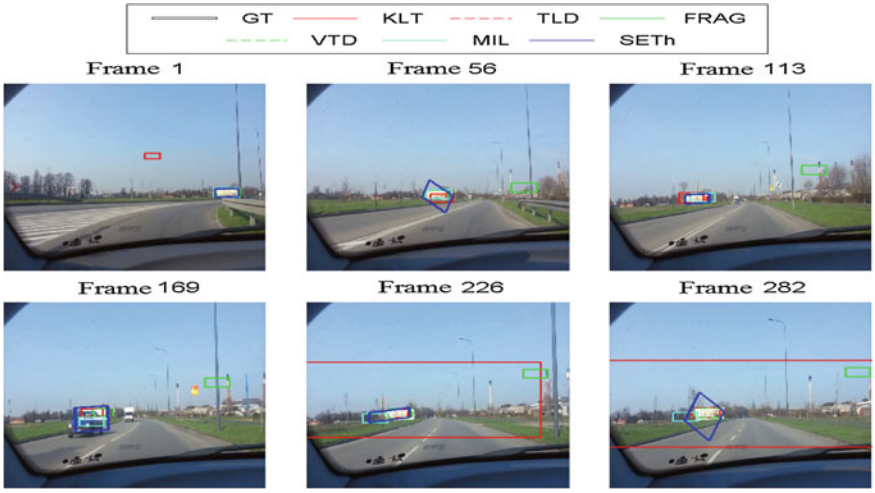


Fig. 53 Visual comparison of tracking results for test sequence

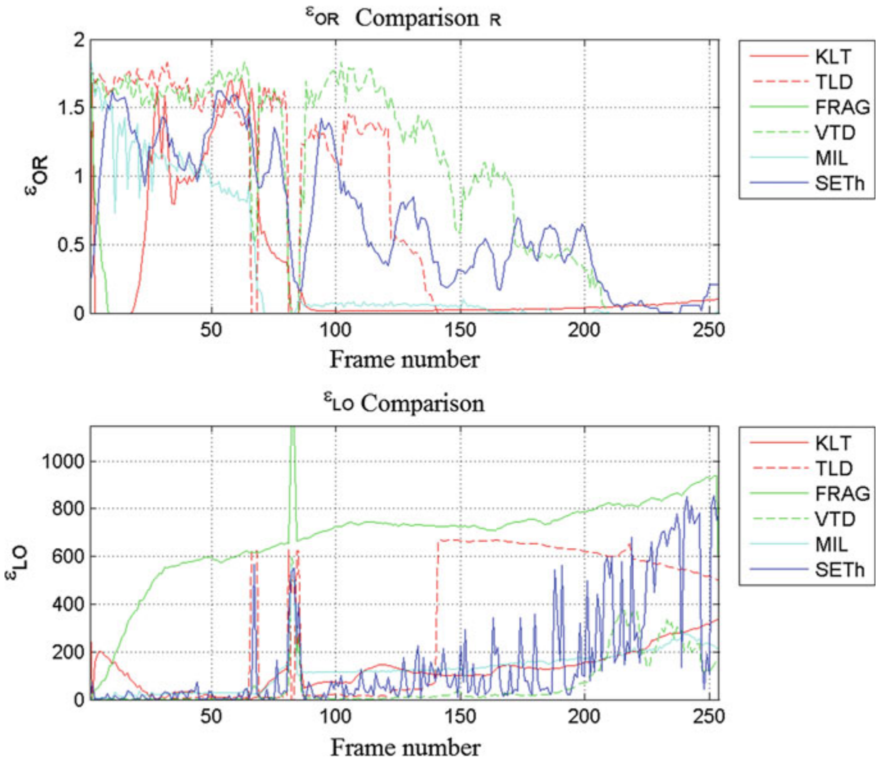


Fig. 54 Comparison of measures test sequences for test sequence

Table 24 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.20079	127	0.20079	0.20079
TLD	0.47244	298.46	0.87591	0.61381
FRAG	0.019685	679.14	0.019763	0.019724
VTD	0.65748	64.148	0.6627	0.66008
MIL	0.26378	149.34	0.27459	0.26908
SETh	0.52362	160.05	0.52362	0.52362

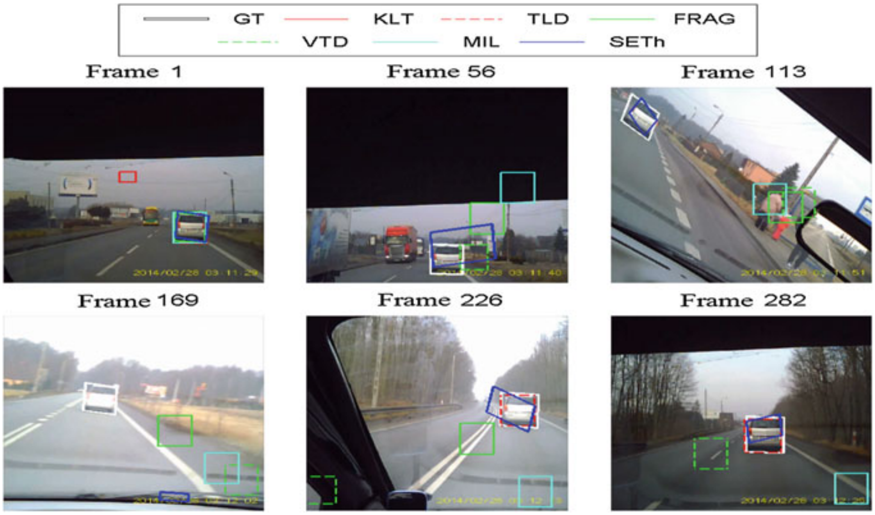


Fig. 55 Visual comparison of tracking results for test sequence

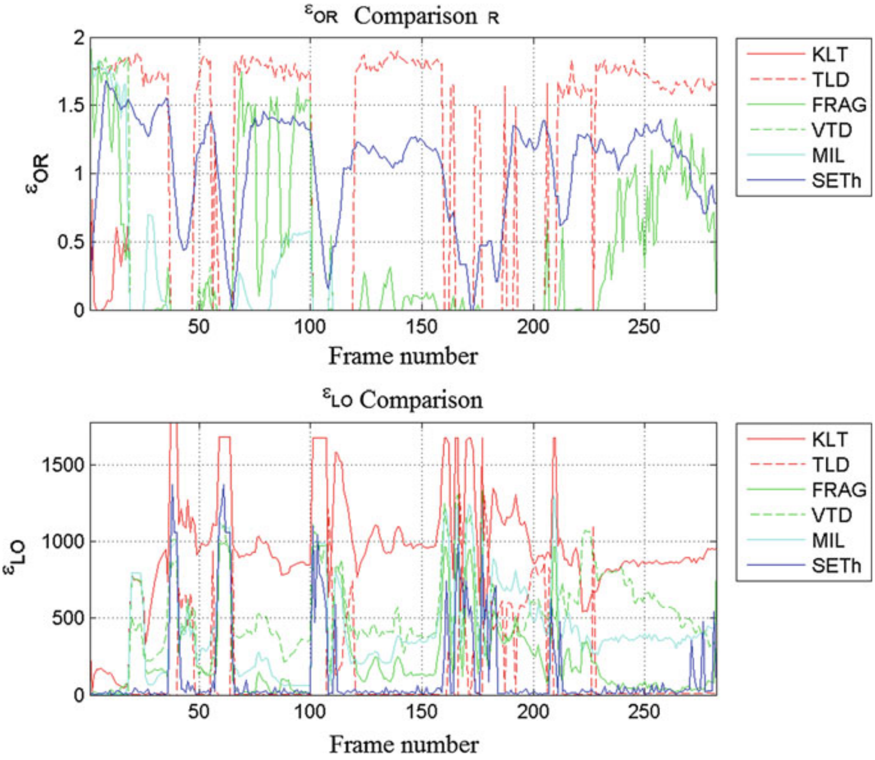


Fig. 56 Comparison of measures test sequences for test sequence

Table 25 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.010638	956.3	0.010638	0.010638
TLD	0.70922	134.03	1	0.82988
FRAG	0.3156	265.39	0.31673	0.31616
VTD	0.06383	507.94	0.064748	0.064286
MIL	0.12057	347.34	0.12057	0.12057
SETh	0.82979	133.13	0.82979	0.82979

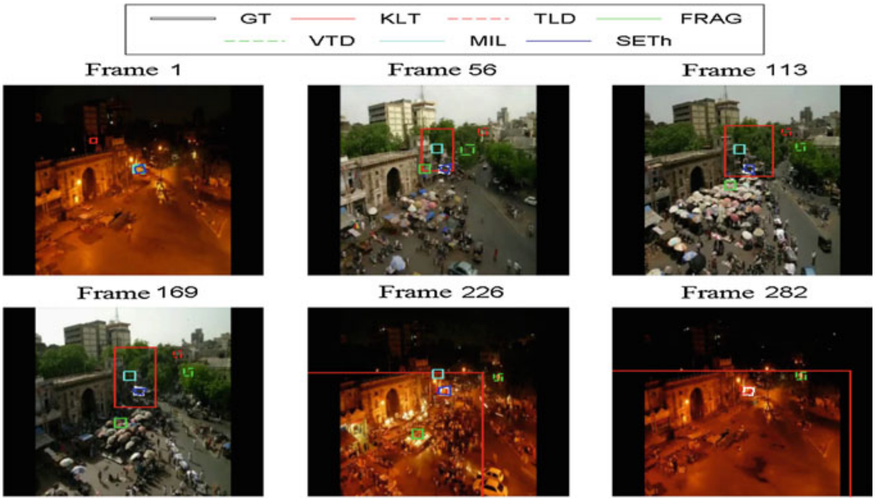


Fig. 57 Visual comparison of tracking results for test sequence

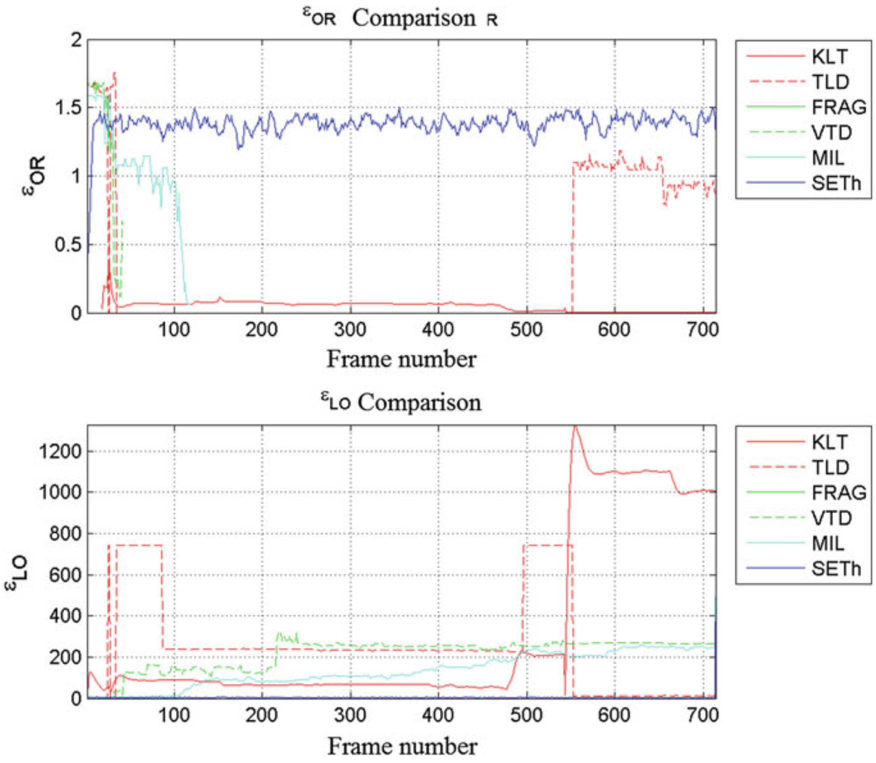


Fig. 58 Comparison of measures test sequences for test sequence

Table 26 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.0014006	322.71	0.0014006	0.0014006
TLD	0.27171	253.05	0.32172	0.29461
FRAG	0.14986	136.54	0.15007	0.14996
VTD	0.046218	216.76	0.046218	0.046218
MIL	0.14986	169.63	0.15007	0.14996
SETh	0.9986	4.4195	1	0.9993

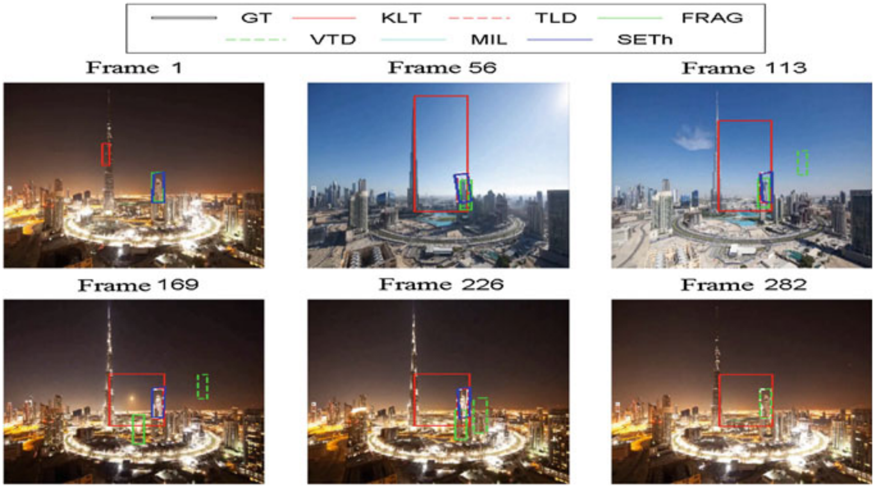


Fig. 59 Visual comparison of tracking results for test sequence

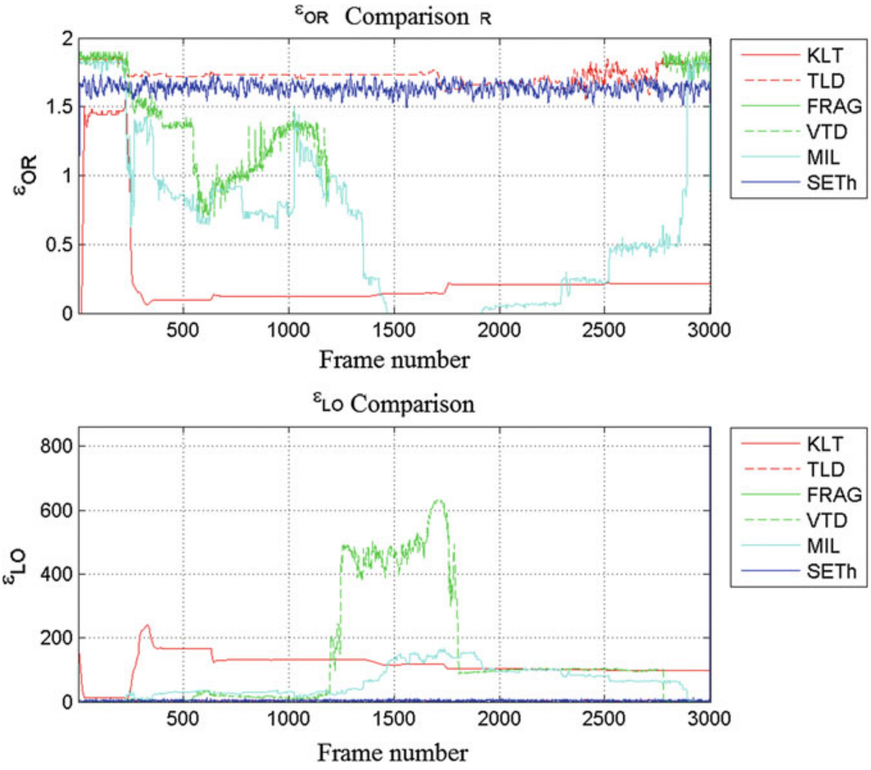


Fig. 60 Comparison of measures test sequences for test sequence

Table 27 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.076641	112.97	0.076641	0.076641
TLD	1	3.6137	1	1
FRAG	0.51616	61.245	0.51633	0.51625
VTD	0.47284	128.55	0.47284	0.47284
MIL	0.51616	95.42	0.51633	0.51625
SETh	0.99967	3.742	1	0.99983

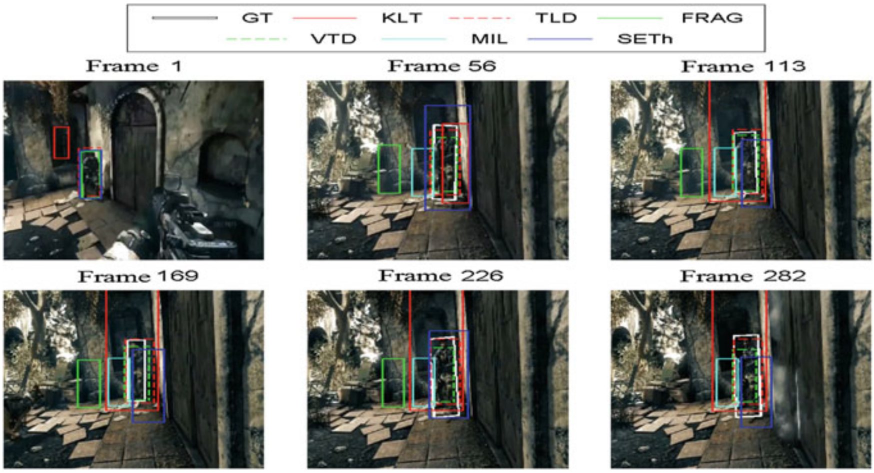


Fig. 61 Visual comparison of tracking results for test sequence

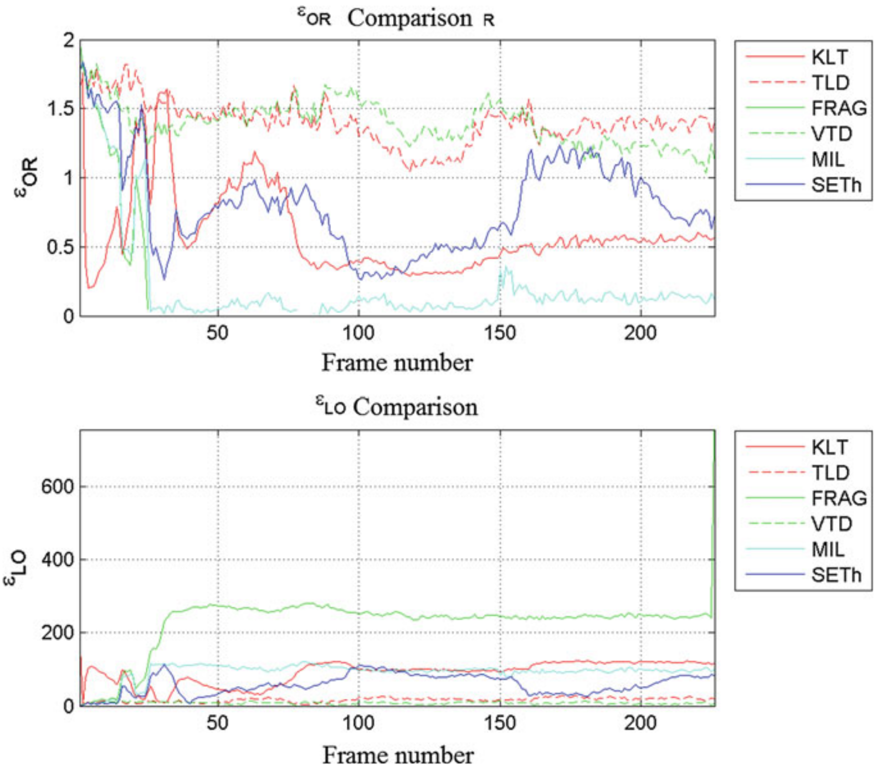


Fig. 62 Comparison of measures test sequences for test sequence

Table 28 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.5708	91.318	0.5708	0.5708
TLD	1	16.411	1	1
FRAG	0.088496	229.39	0.088889	0.088692
VTD	1	7.9347	1	1
MIL	0.088496	104.34	0.088496	0.088496
SETh	0.78761	58.28	0.78761	0.78761

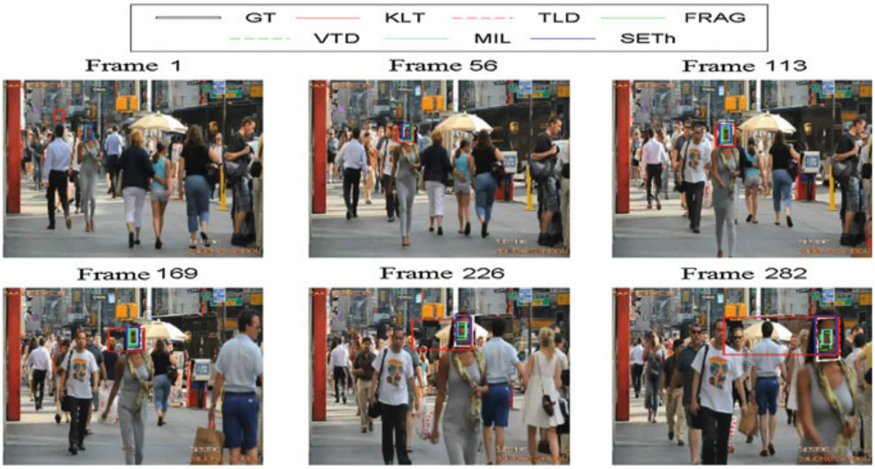


Fig. 63 Visual comparison of tracking results for test sequence

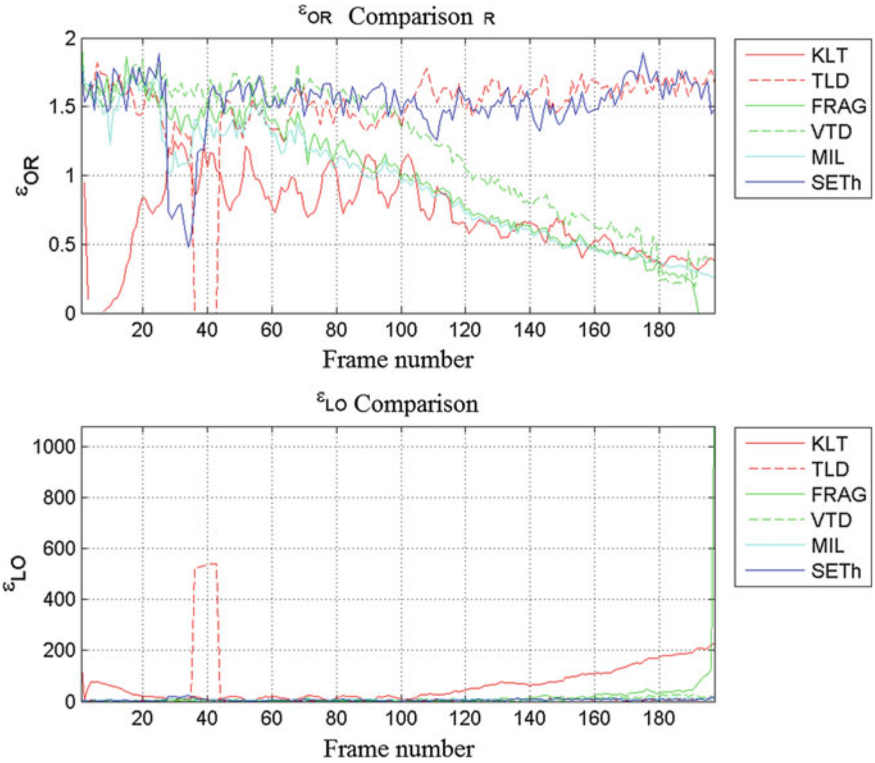


Fig. 64 Comparison of measures test sequences for test sequence

Table 29 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.73604	60.5	0.73604	0.73604
TLD	0.95939	26.372	1	0.97927
FRAG	0.78173	18.855	0.78571	0.78372
VTD	0.88832	6.3125	0.88832	0.88832
MIL	0.77157	15.87	0.77157	0.77157
SETh	0.99492	5.696	0.99492	0.99492

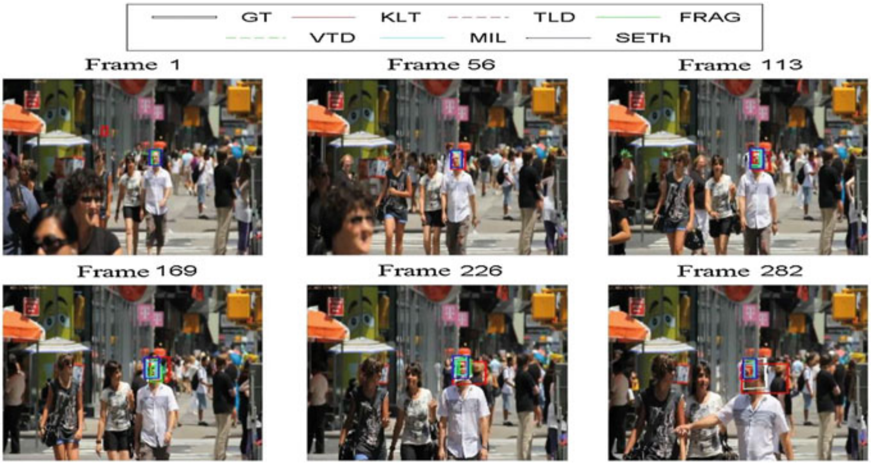


Fig. 65 Visual comparison of tracking results for test sequence

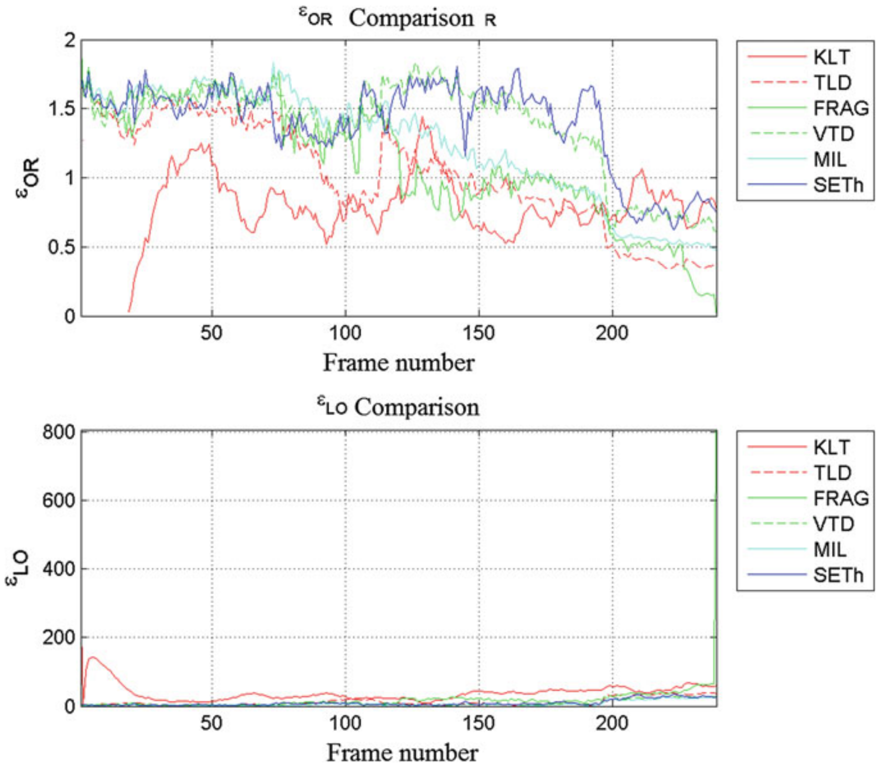


Fig. 66 Comparison of measures test sequences for test sequence

Table 30 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.90377	39.331	0.90377	0.90377
TLD	0.82845	11.928	0.82845	0.82845
FRAG	0.91213	19.075	0.91597	0.91405
VTD	1	8.3407	1	1
MIL	0.98745	16.737	0.98745	0.98745
SETh	1	8.8774	1	1

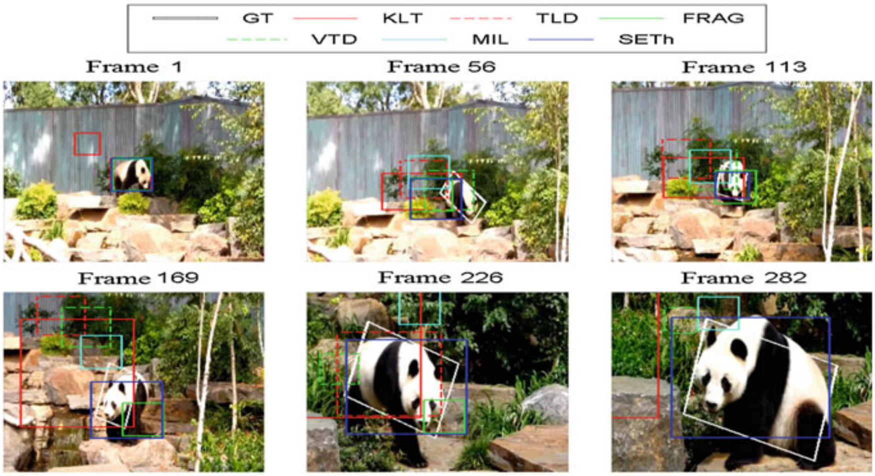


Fig. 67 Visual comparison of tracking results for test sequence

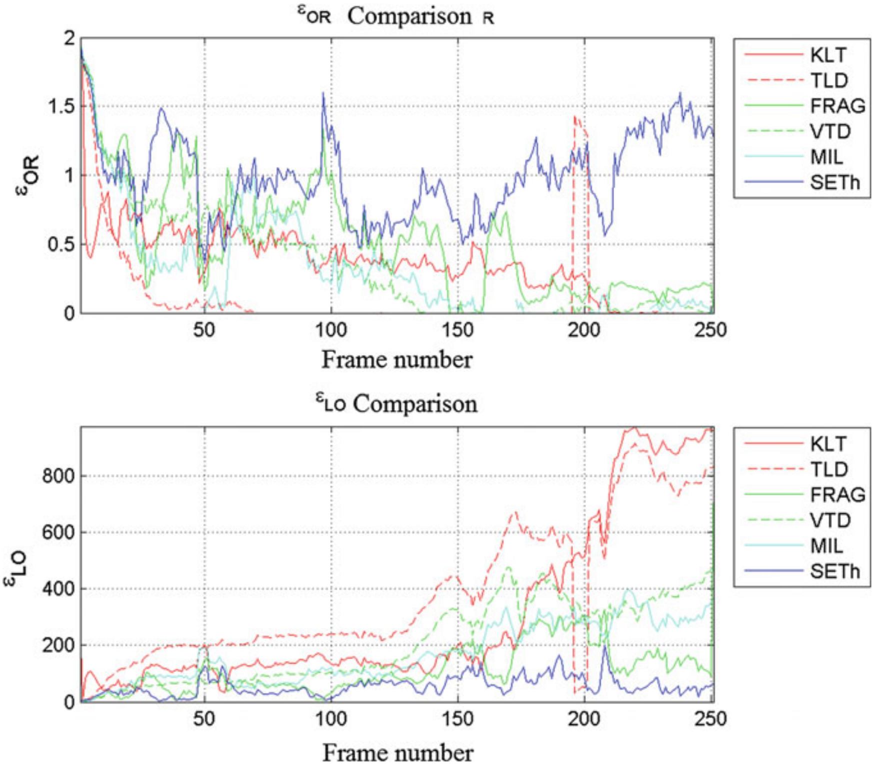


Fig. 68 Comparison of measures test sequences for test sequence

Table 31 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.28287	312.57	0.28287	0.28287
TLD	0.087649	380.98	0.1236	0.10256
FRAG	0.49004	113.96	0.492	0.49102
VTD	0.32271	207.56	0.33061	0.32661
MIL	0.22311	104.72	0.22311	0.22311
SETh	0.97211	54.937	0.97211	0.97211

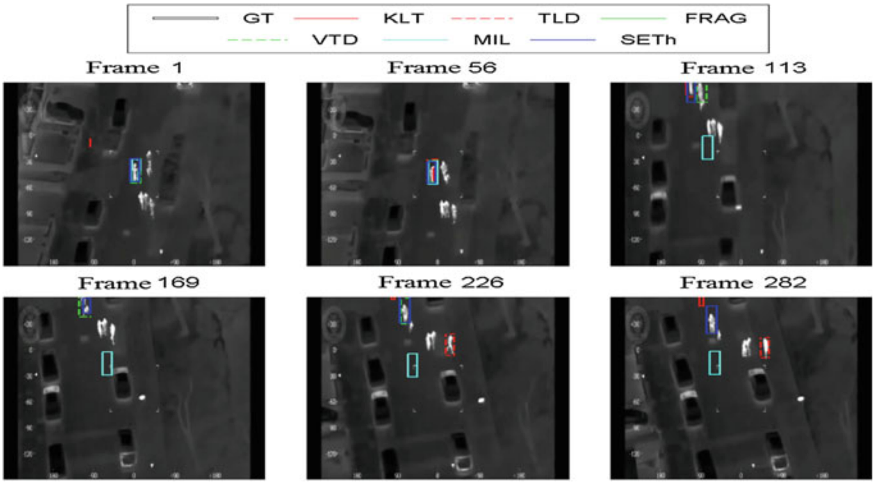


Fig. 69 Visual comparison of tracking results for test sequence

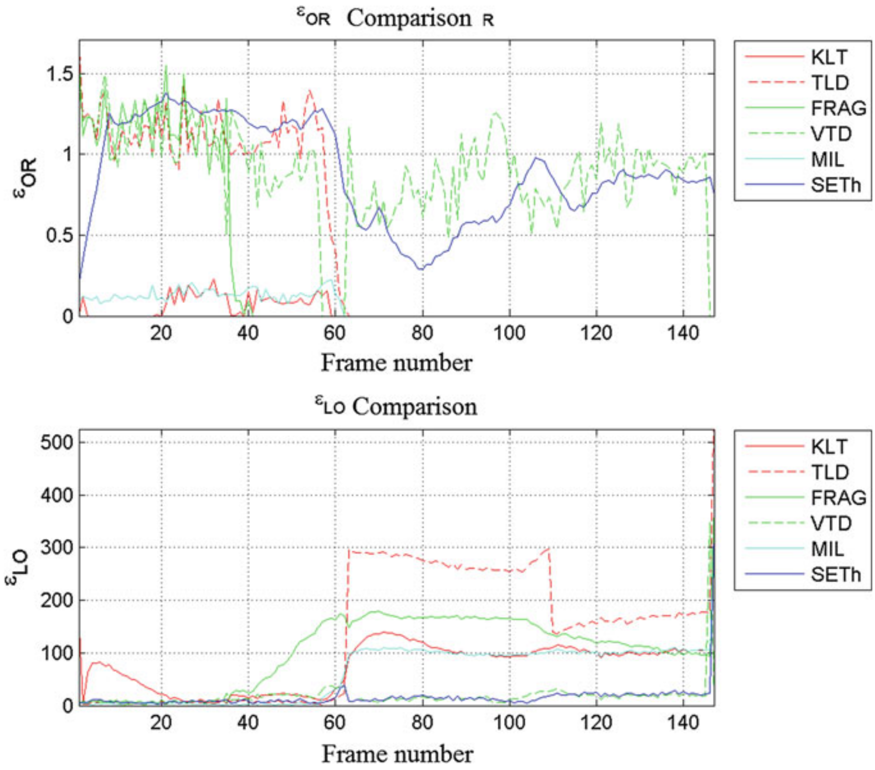


Fig. 70 Comparison of measures test sequences for test sequence

Table 32 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0	75.576	0	0
TLD	0.39456	135.19	0.58	0.46964
FRAG	0.2381	102.65	0.2381	0.2381
VTD	0.93878	17.195	0.95172	0.94521
MIL	0	68.248	0	0
SETh	0.87075	15.262	0.87075	0.87075

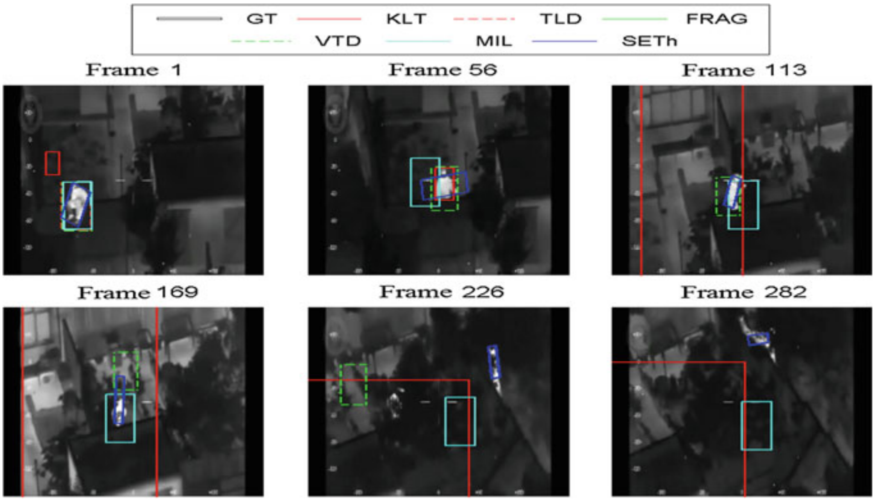


Fig. 71 Visual comparison of tracking results for test sequence

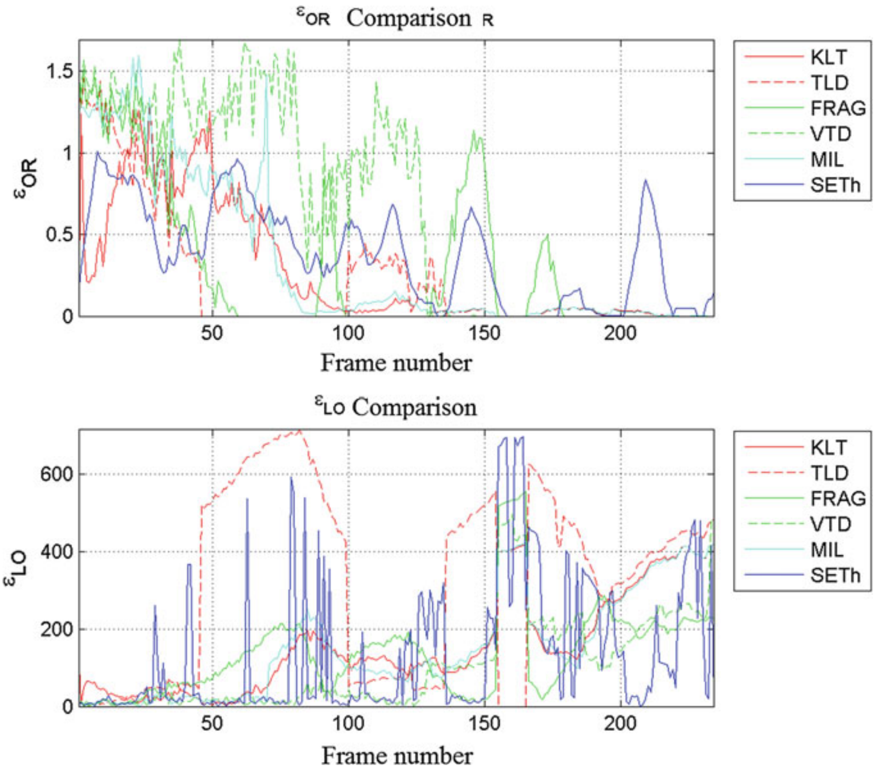


Fig. 72 Comparison of measures test sequences for test sequence

Table 33 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.24786	155.16	0.24786	0.24786
TLD	0.15385	320.35	0.43373	0.22713
FRAG	0.25641	137.27	0.25641	0.25641
VTD	0.52137	104.69	0.52586	0.52361
MIL	0.30342	131.95	0.30342	0.30342
SETh	0.39316	131.86	0.39316	0.39316

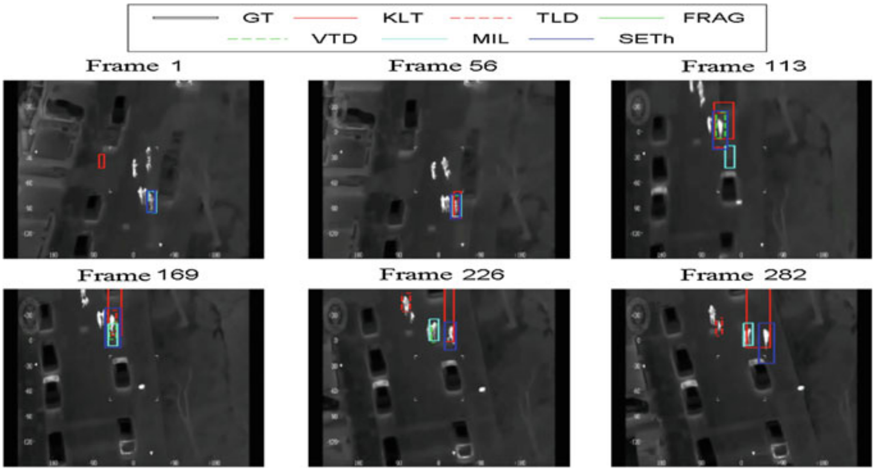


Fig. 73 Visual comparison of tracking results for test sequence

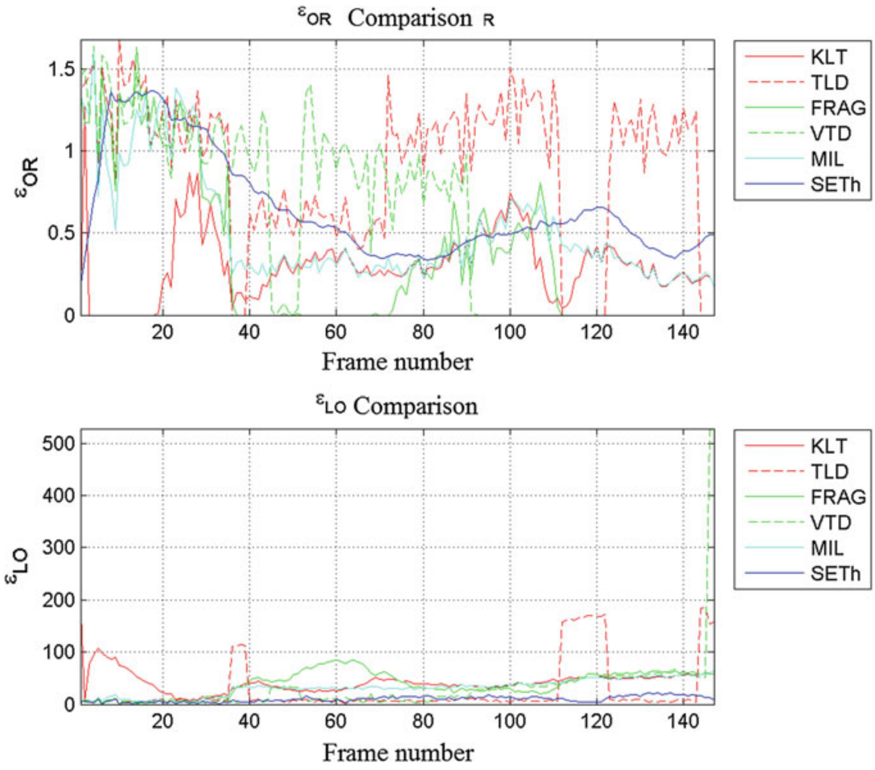


Fig. 74 Comparison of measures test sequences for test sequence

Table 34 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.14286	42.588	0.14286	0.14286
TLD	0.82313	26.12	0.82313	0.82313
FRAG	0.31293	38.71	0.31293	0.31293
VTD	0.55782	29.971	0.56552	0.56164
MIL	0.35374	33.510	0.35374	0.35374
SETh	0.61224	9.8243	0.61224	0.61224

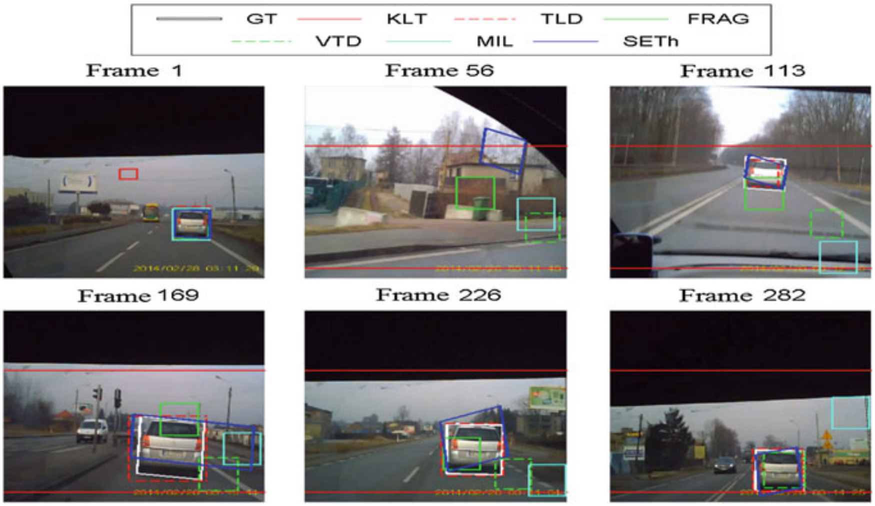


Fig. 75 Visual comparison of tracking results for test sequence

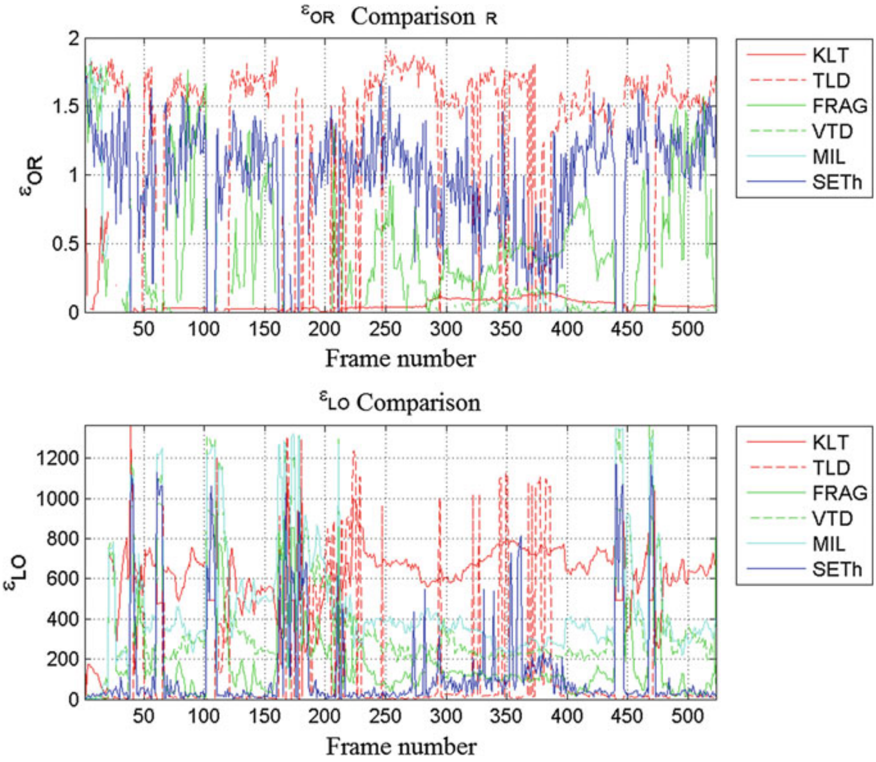


Fig. 76 Comparison of measures test sequences for test sequence

Table 35 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.013384	603.85	0.013384	0.013384
TLD	0.76099	146.21	0.99749	0.86334
FRAG	0.30402	203.57	0.3046	0.30431
VTD	0.043977	374.42	0.043977	0.043977
MIL	0.032505	270.47	0.032505	0.032505
SETh	0.79159	133.09	0.79159	0.79159

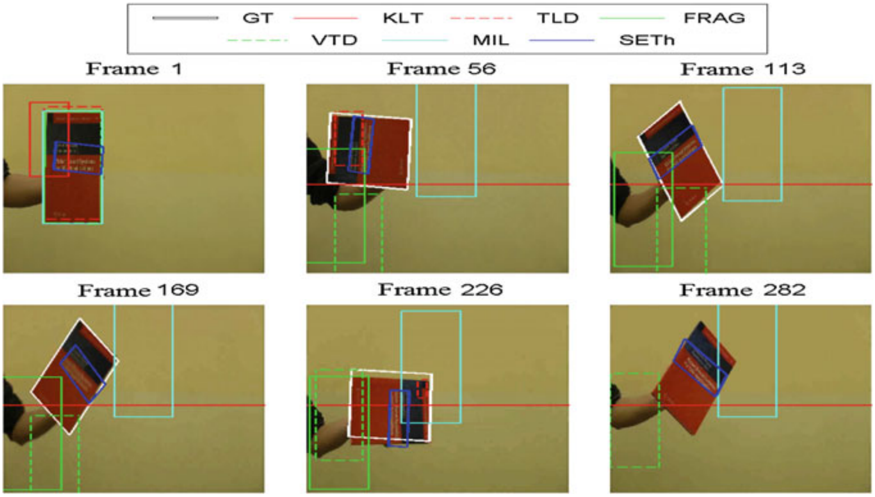


Fig. 77 Visual comparison of tracking results for test sequence

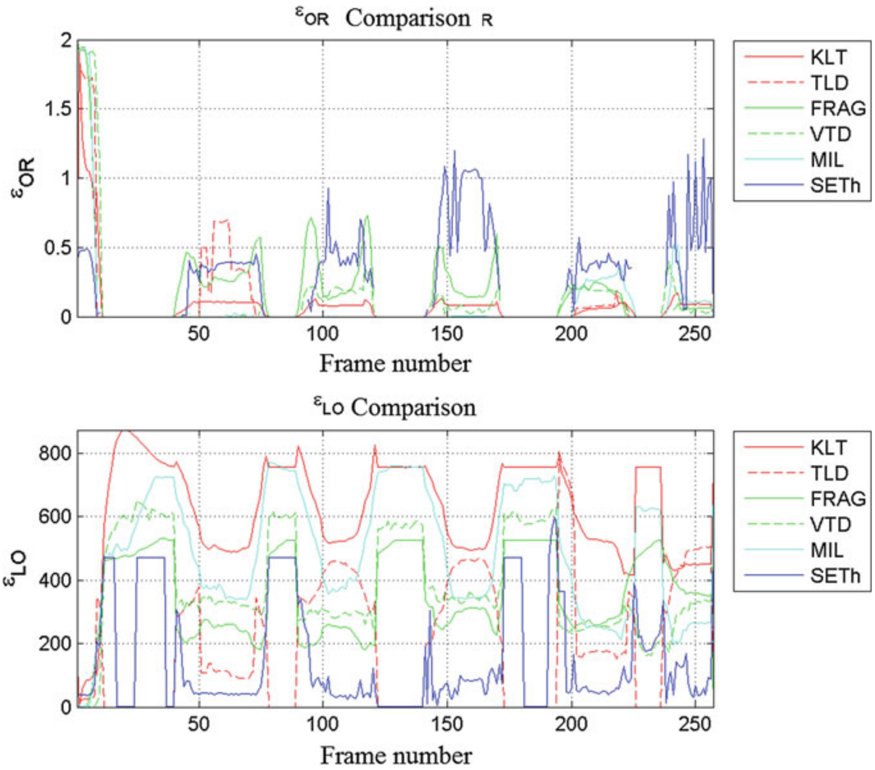


Fig. 78 Comparison of measures test sequences for test sequence

Table 36 Comparison of measures test sequences for test sequence

	ε_{SR}	ε_{ALO}	ε_{PR}	ε_{F}
KLT	0.031128	620.14	0.031128	0.031128
TLD	0.066148	198.95	0.33333	0.11039
FRAG	0.085603	351.06	0.15493	0.11028
VTD	0.038911	380.03	0.041152	0.04
MIL	0.038911	567.03	0.038911	0.038911
SETh	0.1323	143.17	0.15962	0.14468

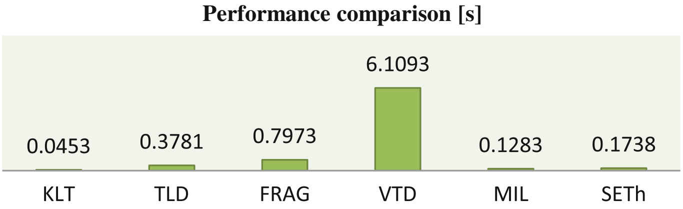


Fig. 79 Column performance comparison of the selected state-of-the-art tracking methods

5 Conclusions

The chapter presents an excessive evaluation of a novel method for long-term feature tracking named SETh [34]. The solution performs in a near real-time consecutive phases: detection, verification and learning. The proposed tracking method begins with the detection of visual features using BRISK detector and describing them using FREAK descriptor. The described features are compared with a set of known features of the object of interest and then clustered by an unsupervised method in order to determine the amount and areas potentially containing the object. Groups of features labeled as object are used to determine the region of interest—the potential area of object which surroundings are compared against the features labeled as background. Utilization during tracking not only object's features but background features as well, allows to increase the quality of tracking of objects characterized by changeable appearance. The object or background labels are attached to unspecified labels within the potential object's area using binary nonlinear SVM. For each potential object final measure consisting of four partial measures is computed. Final measurement value is double thresholded. The tracked object is assumed to be a potential object characterized by a highest value of final measure. The feature points within selected tracked object's area and neighborhood is used to update the SVM classifier.

The proposed tracking method SETh was verified on a prepared comprehensive set consisting of both synthetic and real sequences [35, 36]. Reference value for tracking quality evaluation was annotated manually by human. Manually annotated were more than 25'500 frames. Prepared data set has been made available to the public. SETh tracking method was compared with five state-of-the-art methods and achieved comparable or superior results, suggesting that it is possible to apply it in real-life applications e.g. visual-based control of UAVs [37–40] or tracking pedestrians [41–43].

Acknowledgments This work has been supported by Applied Research Programme of the National Centre for Research and Development as a project ID 178438 path A—Costume for acquisition of human movement based on IMU sensors with collection, visualization and data analysis software.

References

1. Jędrasiak, K., Daniec, K., Nawrat, A., Koteras, R.: Wykorzystanie kamer termowizyjnych w systemach dozoru wizyjnego infrastruktury krytycznej sieci dystrybucyjnych gazu. Przegląd Elektrotechniczny **88**, 90–97 (2012)
2. Ryt, A., Sobel, D., Kwiatkowski, J., Domzal, M., Jedrasiak, K., Nawrat, A.: Real-time Laser Point Tracking, Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 8671. pp. 542–551 (2014)
3. Kuś, Z., Nawrat, A.: Object tracking in a picture during rapid camera movements. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 77–91, ISBN 978-3-319-00368-9 (2013)
4. Pritch, A.Y., Ratovitch, S., Hendel, A., Peleg, S.: Clustered synopsis of surveillance video. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS'09, IEEE (2009)
5. Pritch, Y., Rav-Acha, A., Peleg, S.: Video Synopsis. The Hebrew University of Jerusalem (2014)
6. Bibik, P., Zasuwa, M., Zugaj, M.: Research and training simulator of unmanned quadrotor. In: 18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 403–407 (2013)
7. Babiarz, A., Jaskot, K., Koralewicz, P.: The control system for autonomous mobile platform. In: Advanced Technologies for Intelligent Systems of National Border Security. Studies in Computational Intelligence, vol. 440, pp. 15–28 (2013)
8. Bereska, D., Balcewicz, R., Garczyński, M.: Implementacja magistrali CAN i protokołu CANopen w robocie edukacyjnym. Szybkobieżne Pojazdy Gąsienicowe **1**, 157–162 (2008)
9. Gałuszka, A., Skrzypczyk, K., Bereska, D., Pacholczyk, M.: Re-handling operations in small container terminal operated by reach stackers. World Acad. Sci. Eng. Technol. **70**, 674–677 (2010)
10. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 157–175. ISBN 978-3-319-00368-9 (2013)
11. Yilmaz, A., Li, X., Shah, M.: Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Patt. Analy. Mach. Intell. **26**(11), 1531–1536 (2004)
12. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Computer Vision and Pattern Recognition (CVPR) (2010)
13. Wu, A.: Towards linear-time incremental structure from motion. In: IEEE International Conference on 3D Vision, pp. 127–134 (2013)
14. Tomasi, C., Kanade, T.: Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132 (1991)
15. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 1409–1422 (2012)
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: automatic detection of tracking failures. In: International Conference on Pattern Recognition, pp. 23–26 (2010)
17. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2006)
18. Babenko, Boris, Yang, Ming-Hsuan, Belongie, Serge: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
19. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: Proceedings of IEEE 12th International Conference Computer Vision (2009)

20. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR (2006)
21. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Proceedings of 10th European Conference on Computer Vision (2008)
22. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)
23. Jedrasiak, K., Nawrat, A.: Fast color recognition algorithm for robotics. Probl. Eksploatacji. Maintenance Problems (Quarterly), pp. 69–76 (2008)
24. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion feature. Advanced robotics, control and advanced manufacturing systems. In: Proceedings of the 9th WSEAS International Conference on Robotics, Control and Manufacturing Technology (ROCOM'09), pp. 67–72 (2009)
25. Jędrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 5337, pp. 391–399 (2009)
26. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: binary robust invariant scalable keypoints. In: ICCV'11, pp. 2548–2555 (2010)
27. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: fast retina keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
28. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), pp. 226–231 (1996)
29. FLIR, dostępne na: http://www.flir.com/PL (Data pobrania zasobu: 05-03-2014)
30. Toussaint, G.T.: Solving geometric problems with the rotating calipers. Proc. IEEE Melecon **83**, A10 (1983)
31. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
32. Morzy, T.: Eksploracja Danych – Metody i Algorytmy. Wydawnictwo Naukowe PWN, Warszawa (2013)
33. https://www.dropbox.com/sh/v705ac32e8006z7/s1ezUqplcE
34. Demski, P., Grzejszczak, T., Jędrasiak, K., Mikulski, M.: Automatic targeting sentry turret for distributed systems. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 47–55, ISBN 978-3-319-00368-9 (2013)
35. Sroka, M., Ściegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzałogowego pojazdu podwodnego – układ stabilizacji i utrzymania zadanego kursu. Przegląd Elektrotechniczny **89**, 205–217 (2013)
36. Jaskot, K., Babiarz, A., Sroka, M., Ściegienka, P.: Prototyp bezzałogowego pojazdu podwodnego – konstrukcja mechaniczna, panel operatora. Przegląd Elektrotechniczny **89**, 52–67 (2013)
37. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 87–94. Springer, Berlin (2013)
38. Czapla, T., Wrona, J.: Technology development of military applications of unmanned ground vehicles. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 293–309. ISBN 978-3-319-00368-9 (2013)
39. Niezgoda, T., Panowicz, R., Sybilski, K., Barnat, W.: Numerical analysis of missile impact being shot by rocket propelled grenades with rod armour. WIT Trans. Model. Simul. **51**, 625–633 (2011)
40. Kuś, Z., Nawrat, A.M.: The limitation for the angular velocity of the camera head during object tracking with the use of the UAV. In: Innovative Control Systems for Tracked Vehicle Platforms. Studies in Systems, Decisions and Control, vol. 2, pp. 127–145 (2014)

41. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 423–432, Springer, Berlin (2013)
42. Bereska, D., Daniec, K., Fraś, S., Jędrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 117–189. ISBN 978-3-319-00368-9 (2013)
43. Bereska, D., Daniec, K., Jędrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 115–121. ISBN 978-3-319-00368-9 (2013)

The System for Augmented Reality Motion Measurements Visualization

Dawid Sobel, Jan Kwiatkowski, Artur Ryt, Mariusz Domżał, Karol Jędrasiak, Łukasz Janik and Aleksander Nawrat

Abstract The system which allows for visualisation of range of motion examination has been developed. The visualisation is clear and in simple manner presents measured angles in the joints. Measurement is based on data from motion capture system about location of the markers on the patient. Markers are place at key locations in this way that connection of next three form an angle which vertex is situated in the joint. Image from the camera directed at the patient is extended by adding lines, angles and the numerical values of the current value of the angle of flexion of the limb directly on the limbs of the patient. The accuracy of the measurement depends mainly on the accuracy of the motion capture system and is of order of tens of micrometres. The precision of the measurement and repeatability is a advantage of the system because the classical methods based on the use of goniometer are dependent on the individual interpretation of the physiotherapist. In addition the test may be carried out in movement such as walking and for more limbs than one at a time. During of the examination patient is not limited in movement by the measuring device.

Keywords Range of motion · Camera calibration · Augmented reality · Motion capture

1 Introduction

Modern medicine is able to deal with many diseases once thought to be incurable. The development of medical knowledge, familiarity with the human body as well as new technologies, enables implementation of complicated operations, while the medical equipment is becoming more and more precise and reliable. Specialised

D. Sobel (✉) · J. Kwiatkowski · A. Ryt · M. Domżał · K. Jędrasiak · Ł. Janik · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland
e-mail: dawid.sobel@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_10

tools support the work of a doctor [1], sometimes even replacing him during operations in which the precision unattainable by a human is required.

One of the basic tests in physiotherapy is examination of the range of motion of a person, who as a result of an accident or due to some other cause cannot be physically fit [2]. The range of motion is measured with aid of a tool called goniometer, which is placed at a patient's joint, and then with its two movable arms, the angle formed by the flexion or extension of the limb is determined. Unfortunately, the measurements carried out with this method are sensitive to the human factor, i.e. they depend on how the tool is placed at the body and with what accuracy the value of the angle is read. The measurements done by different people can also differ and are dependent on individual interpretation. Furthermore, the examination itself requires restriction of the patient's movements and, at the same time, the therapist uses both hands to conduct the measurement. In conjunction with the above, there is a need for such a method of measurement, which is insensitive to the reading error and to the interpretation of the measurements results. The new method should also simplify the measurement as well as give the patient greater freedom of movement. It would also be great if such measurements could be taken in motion, in normal conditions during walking, running, jumping, or other exercises [3–9]. It would be even better if there was a possibility of measuring a few or all the angles simultaneously.

2 Review of Medical Solutions

The main tool used for the measurements of the range of motion is a goniometer mentioned in the introduction. It is a tool consisting of two arms connected together. The arms are placed alongside parts of body between which the angle is to be measured and then, the angle at which they are positioned is read (Fig. 1a). An advantage of such solution is the simplicity of the measurement, while a disadvantage is that the measurement largely depends on the validity of the reading from the scale and exactness of placing of the goniometer at the body. Theoretically, Gajdosik and Bohannon [2] show that by obeying strictly specified procedures a repeatable and reliable measurement can be made but it is still dependent on individual interpretation.

Another solution is an electronic goniometer allowing a measurement of the range of motion with the use of two sensors connected with a spring (Fig. 1b). The device enables a measurement in one or two planes, it is comfortable, especially on uneven surfaces of body and does not require constant holding by the doctor as it is attached directly to the patient. The result of the measurement is precise and can be simply read in a digital form [12, 13]. Manufacturer of an exemplary device, *Biometrics LTD UK* gives the accuracy of the measurement to be ±2° in range ±90° [10].

A laser goniometer HALO (Fig. 1c) has also been introduced in the market. It stands out due to the ease and speed of the measurement, because it only needs to be placed over the limb whose range is to be measured, then a button is pressed to

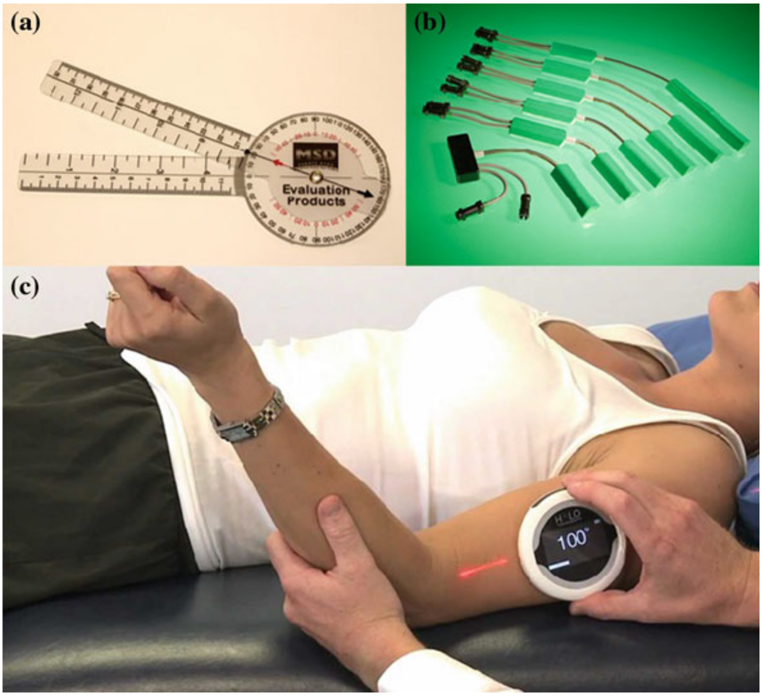


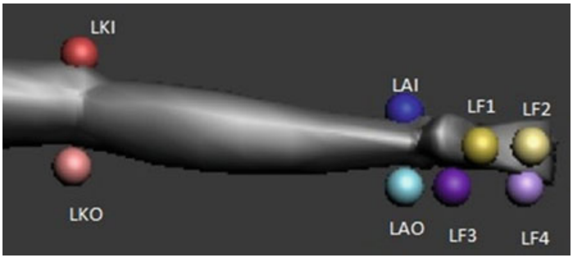
Fig. 1 Classic goniometer (**a**). Electronic goniometer [10] (**b**). Laser goniometer [11] (**c**)

determine the point of reference. The device needs to be held in the same position during the movement. At the end of the movement, the result can be read from a digital display and its accuracy is ±1° [11]. The manufacturer states that such a solution makes the obtained results independent of the individual measuring technique for each therapist, because independently of a person, every measurement is done identically, unlike in a classic goniometer. An additional advantage is recording of the results in the memory of the device. The only disadvantage that can be observed is that the result is dependent on how the Halo device is held during the measurement above or directly on the limb. Because it can never be held over it in exactly the same position, the obtained results can vary insignificantly.

3 Proposed Solution

The new method of measurement of the range of movement and also processing of the results is based on the use of the data reading from Motion Capture. Motion Capture is a system of few or over a dozen of cameras calibrated to one system of coordinates and directed onto a common space. If in this space an object designated

Fig. 2 Markers on the left leg



with aid of markers is found, the system is able to determine the position of every visible marker in a 3D space. The principle of operation is based on the application of diodes emitting infrared light, which reflects from the reflective markers and is later registered by the cameras. Afterwards, in the process of triangulation based on 2D images from the cameras, the position of the markers in space is determined. These markers are located on the tested limbs and joints on the patient's body (Fig. 2). Connecting these points into straight lines, selected angles in the patient's body are computed.

In order to create a clear visualisation of the test results of the range of motion, elements of augmented reality were used. Selected points connected with lines and angles between them were imposed onto the image obtained from the camera registering the test. In this way, after the test, a therapist can monitor the angles changing during the examination. In order to make the obtained image overlap with the lines drawn, it is necessary to calibrate the camera [14–17].

Calibration of the camera is to find intrinsic and extrinsic camera parameters. Intrinsic parameters are related to the optical and electronic properties of a camera, i.e.: focal length, radial and tangential distortion coefficients, and real coordinates of centre of the camera image. Extrinsic parameters describe transformation, rotation and translation of the system related to the observed 3D scene to the camera system. Based on this transformation, we are able to find for the points positioned in space, corresponding points on the image registered by the camera (Fig. 3). Calibration is usually done by directing the camera onto an object of a known shape and pattern or an object of known 3D coordinates its characteristic points. In case of the presented solution, the object used for the calibration was data for a few basic exercises, from

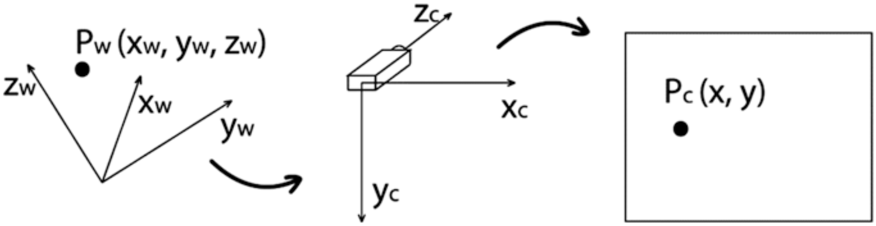


Fig. 3 Transformation of 3D coordinates into 2D image coordinates

Motion Capture system. It is worth noting, that if the position of the camera registering the image does not change, it is enough to run the calibration procedure once.

Calibration method applied was developed by Tsai [18]. Special calibration techniques using a noncoplanar set of point, which was used was described in detail in another publication [19]. After splitting the calibration procedure into two subproblems, we improve the speed of the calculation as well as the precision. The first step of the calibration algorithm is the implementation of linear methods to find approximated intrinsic and extrinsic parameter values of the camera followed by their use in the second step as the initial values in the nonlinear optimisation process of the equations below (Eqs. 1 and 2). Thus, in the second step, only two, or even one iteration is enough for finding a solution with a good precision.

$$s_x^{-1} d'_x X + s_x^{-1} d'_x X k_1 r^2 = f \frac{r_1 x_w + r_2 y_w + r_3 z_w + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \tag{1}$$

$$d'_y Y + d'_y Y k_1 r^2 = f \frac{r_4 x_w + r_5 y_w + r_6 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \tag{2}$$

where parameter s_x, d_x, d_y are CCD matrix parameter. Points X and Y are the coordinates of an image pixel horizontally and vertically respectively, and x_w, y_w, z_w are the corresponding coordinates in the 3D scene. Radial distortion coefficient k_1 and focal length f, represent the intrinsic parameters of the camera, while the rotation matrix elements $r_1 \ldots r_9$ and translation vector elements T_x, T_y, T_z represent the extrinsic parameters.

Nonlinear optimisation was carried out using a ready-made implementation of the Levenberg-Marquardt method from the Levenberg-Marquardt.NET library [20], which is a combination of the gradient descent method and the Gauss-Newton method [21].

After the calibration, we obtain a full camera model with intrinsic and extrinsic parameters. Knowing the camera model, we can utilise it for drawing a virtual object, i.e. in this case lines and the angles between them in the space. The problem of merging the real image with the virtual one is a matter of the augmented reality mentioned before. In order to add a virtual object to the image, it is necessary to create its 3D model. To add straight line on the image why it was necessary to find in the 3D space the equation of a straight line based on the points through which it was supposed to go. The next step was discretisation of the line into a series of points and transformation of each point into the image coordinates. For this purpose we use equation (Eq. 3) based on the camera parameters determined previously.

$$\begin{bmatrix} x_{camera} \\ y_{camera} \\ z_{camera} \end{bmatrix} = R \begin{bmatrix} x_{word} \\ y_{word} \\ z_{word} \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{3}$$

The obtained coordinates are the coordinated in the camera system. This is why we need to recalculate them into the image coordinates in two steps. We use equations (Eq. 4) to transfer from the camera system to the image system with the origin of the system in its centre.

$$x = f\left(\frac{x_{camera}}{y_{camera}}\right), \quad y = f\left(\frac{x_{camera}}{y_{camera}}\right) \tag{4}$$

Next, in the second step we "distort" the position (x, y) of our point in order to adjust it to the naturally distorted image (Eq. 5). The equations presented below were suggested by Silesian University of Technology researchers [22] and their application produces visually appealing results,

$$\begin{aligned} x_d &= x(1-\gamma) + C_x \\ y_d &= y(1-\gamma) + C_y \end{aligned} \tag{5}$$

where γ is described by the below equations (Eqs. 6 and 7).

$$\gamma = \lambda\frac{x^2 + y^2}{x_c y_c} \tag{6}$$

$$x = x_r - x_c, \quad y = y_r - y_c \tag{7}$$

Hence, knowing the position of the points from the 3D space on the image we can perfectly draw the selected lines overlapping with the patient's real movements in the subsequent frames (Figs. 4 and 5).

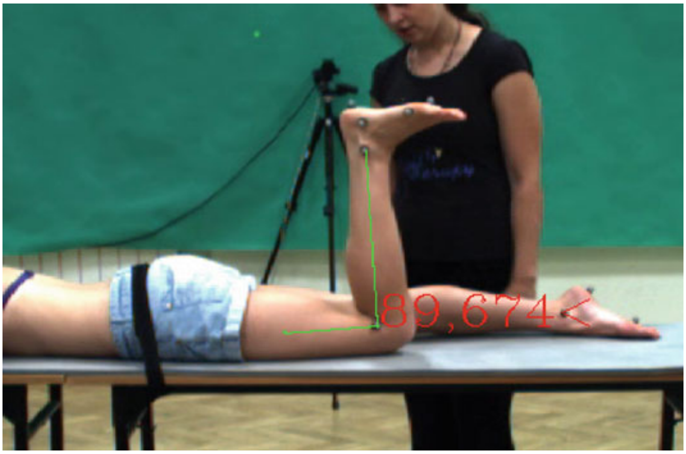


Fig. 4 Visualization of the angle in each frame with added reference points

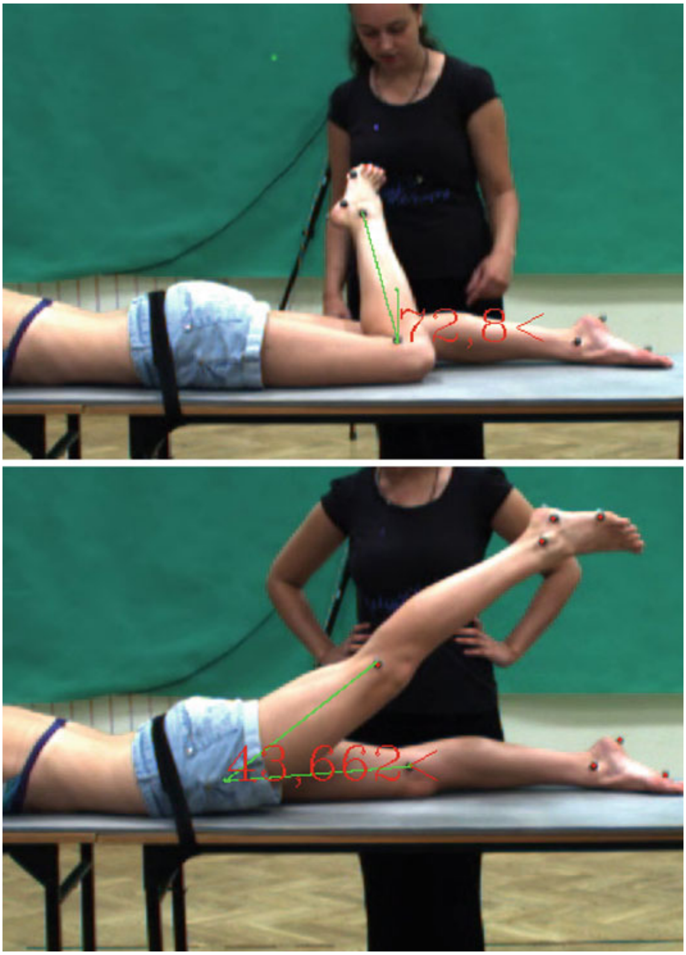


Fig. 5 Visualization of the angle in each frame

The angle presented in the form of digits and the lines are located on the image using ready-made functions of OpenCV library [23]. To calculate the equation of the line, the set of equations (Eq. 8) was used. The subsequent points of the line arise from the multiplication of the line directional coefficients with the increasing coefficients [22]. The equations of the line and the angles between them are calculated and drawn for every subsequent frame of the film.

$$\begin{cases} x = x_0 + ms \\ y = y_0 + ns \\ z = z_0 + ls \end{cases} \tag{8}$$

where:

m, n, l	are the directional coefficients of the straight line
s	is a number different from zero
(x_0, y_0, z_0)	starting point of the line
(x, y, z)	currently calculated point.

The application software that supports calibration and drawing angles has also a function of adding reference static points and lines, which allows measuring the flexion or extension with respect to, e.g. a vertical line, or another line, placed arbitrarily in space (Fig. 5).

4 User Interface

The application allows both the camera calibrations and drawing lines and angles between selected markers and adding virtual reference points by placing them in a calibrated 3D space. In the lower left corner of the user interface is placed scheme of arrangement of markers on the patient's leg. During the calibration process next points are highlighted in red on scheme and then the user selects the corresponding markers on the image placed on the patient's body (Fig. 6). In the case when the point on the image is not visible because it is covered or is on the other side of the limb point is skipped.

On the left side of the user interface is a list of frames and points assigned to them. Movie frames used in the calibration are selected by the user and represent

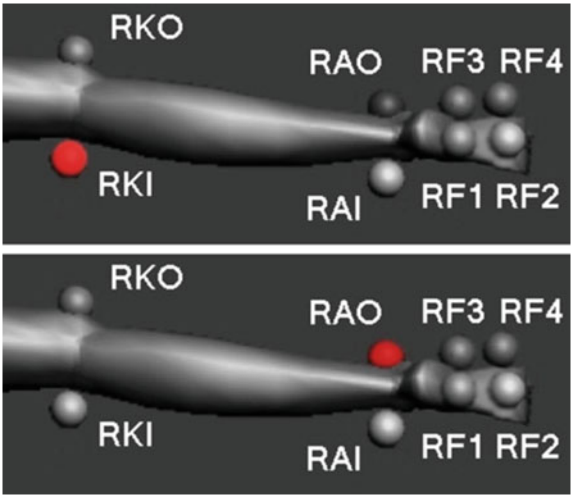
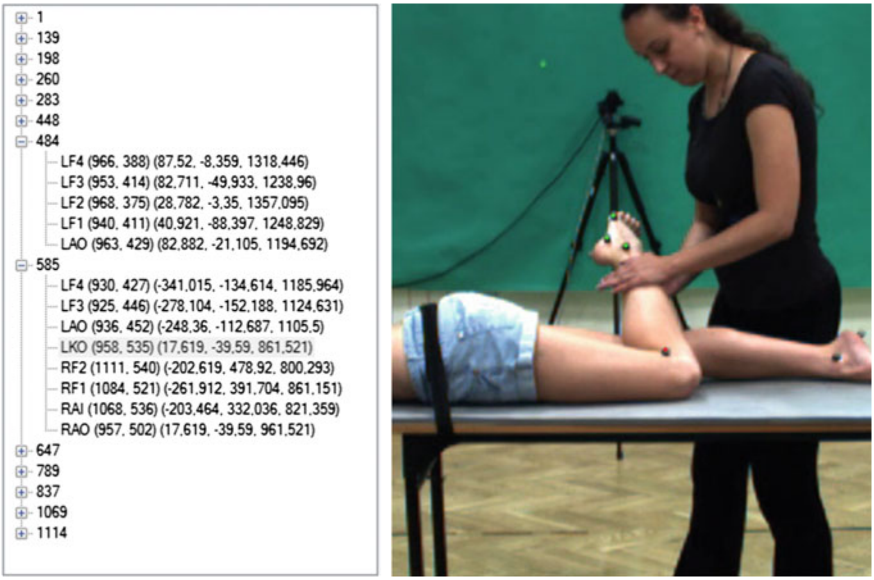


Fig. 6 Lighted markers on the leg schema



```
⊞ 1
⊞ 139
⊞ 198
⊞ 260
⊞ 283
⊞ 448
⊟ 484
    LF4 (966, 388) (87.52, -8.359, 1318.446)
    LF3 (953, 414) (82.711, -49.933, 1238.96)
    LF2 (968, 375) (28.782, -3.35, 1357.095)
    LF1 (940, 411) (40.921, -88.397, 1248.829)
    LAO (963, 429) (82.882, -21.105, 1194.692)
⊟ 585
    LF4 (930, 427) (-341.015, -134.614, 1185.964)
    LF3 (925, 446) (-278.104, -152.188, 1124.631)
    LAO (936, 452) (-248.36, -112.687, 1105.5)
    LKO (958, 535) (17.619, -39.59, 861.521)
    RF2 (1111, 540) (-202.619, 478.92, 800.293)
    RF1 (1084, 521) (-261.912, 391.704, 861.151)
    RAI (1068, 536) (-203.464, 332.036, 821.359)
    RAO (957, 502) (17.619, -39.59, 961.521)
⊞ 647
⊞ 789
⊞ 837
⊞ 1069
⊞ 1114
```

Fig. 7 An example part of the GUI of the application

key moments examination of the patient and the points assigned to each frame are those selected on the image in the calibration process (Fig. 7). Points on the list can be added and removed when the point is entered incorrectly. Selecting a point on the list causes that it is highlighted in the frame. For each point are assigned its coordinates in 2D imaging system and the coordinates of a point in 3D space.

The application also lets you to load data from a file that allows the execution of the calibration process only once. Visualization of range of motion in the joints is done by drawing lines and angles between selected points. Drawing angles is accomplished by a selection of 3 consecutive points located in 3D space where the center point is the vertex of the angle. For this purpose the user clicks in place on the image where the point should be located and the application searches its database of known points in order to find the nearest known point in 3D space.

Furthermore, it is possible to add a virtual 3D point and position it in a calibrated space. This point can be later used to draw lines and angles. Example of the use can be placement of a virtual point above the point on the knee, what enables measurement of the range of motion of the limb (Fig. 4). When the point is adding, it is necessary to specify the coordinates of the position and the application based on the calibration parameters determines its position on the image and draws it.

The application has all the elements needed to effectively measure and visualize the angles of the joints.

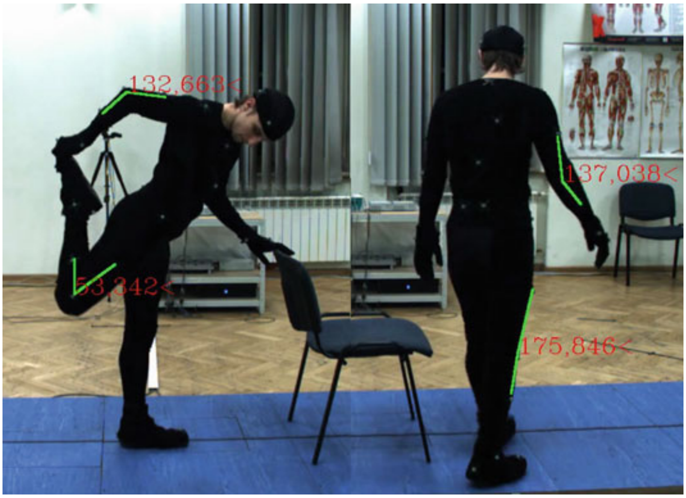


Fig. 8 Augmented reality visualization of measurements

5 Summary

As a final result, the measurement of the range of motion with accuracy of order of tens of milimeters was obtained, which is mainly due to the precision of the Motion Capture system. If the position of camera does not change, it is enough to run a single calibration. The obtained measurement is independent on the individual measurement technique of every therapist and his subjective interpretation of the results, as it was the case for a classic goniometer.

The measurement is convenient because after placing markers on the patient, he has practically total freedom of movement and it is not necessary to hold the device or a measuring tool by a doctor, who can at the same time pay more attention to the patient. Such a solution allows simultaneous measurement of lager number of joints than just one and because the neighbouring joints are dependent on each other, it influences the way in which the body moves. Additionally, the patient can easily do exercises or actions such as normal walking through a room, a squat or even a jump. A clear visualisation on a computer screen, possible to be created also in real time, enables a detailed analysis of motion and angles in the joints that change during exercising (Fig. 8).

Another quality of the solution is automatic registration of the measurements in a data base in the form of 3D coordinates of the markers position as well as films registered by the cameras [24–28]. In this way, every patient has his own catalogue with medical history visualised in the form of films with exercises, which allows a quick judgement of the rehabilitation progress.

References

1. Wochlik, I., Bułka, J., Folwarczny, Ł., Daniec, K., Jędrasiak, K., Koteras, R., Nawrat, A.: Application of telemedical technologies in remote evaluation of soldiers' vital signs during training and in combat conditions. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 189–202 (2014)
2. Gajdosik, R.L., Bohannon, R.W.: Clinical measurment of range of motion: review of goniometry emphasizing reliability and validity, Phys. Ther. **67** (1987)
3. Bibik, P., Zasuwa, M., Zugaj, M.: Research and training symulator of unmanned quadrotor. In: 18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 403–407 (2013)
4. Babiarz, A., Jaskot, K., Koralewicz, P.: The control system for autonomous mobile platform. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 15–28 (2013)
5. Bereska, D., Balcewicz, R., Garczyński, M.: Implementacja magistrali CAN i protokołu CANopen w robocie edukacyjnym, Szybkobieżne Pojazdy Gąsienicowe, vol. 1, pp. 157–162 (2008)
6. Gałuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych, Przegląd Elektrotechniczny, vol. 86, pp. 278–283 (2010)
7. Gałuszka, A., Skrzypczyk, K., Bereska, D., Pacholczyk, M.: Re-handling operations in small container terminal operated by reach stackers. World Acad. Sci. Eng. Technol. **70**, 674–677 (2010)
8. Czapla, T., Wrona, J.: Technology development of military applications of unmanned ground vehicles. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 293–309 (2013). (ISBN: 978-3-319-00368-9)
9. Niezgoda, T., Panowicz, R., Sybilski, K., Barnat, W.: Numerical analysis of missile impact being shot by rocket propelled grenades with rod armour. WIT Trans. Model. Simul. **51**, 625–633 (2011)
10. Biometrics Ltd.: Goniometr and torsiometer operating manual, acces via Internet: http://www.biometricsltd.com. Access: Apr 2014
11. HALO Proffesional Digital Goniometers, acces via Internet: http://www.halo-goniometer.com. Acces: Apr 2014
12. Rothstein, J.M., Miller, P.J., Rottger, R.F.: Goniometry reliability in a clinical setting: elbow and knee measurments. Phys. Ther. **63** (1983)
13. Świtoński, A., Josiński, H., Jędrasiak, K., Polański, A., Wojciechowski, K.: Classification of poses and movement phases. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 6374, pp. 193–200 (2010)
14. Sobel, D., Jędrasiak, K., Daniec, K., Wrona, J., Jurgaś, P., Nawrat, A.: Camera calibration for tracked vehicles augmented reality applications. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 147–162 (2014)
15. Sobel, D., Kwiatkowski, J., Ryt, A., Domżał, M., Jędrasiak, K., Janik, Ł., Nawrat, A.: Range of motion measurements using motion capture data and augmented reality visualization. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 594–601 (2014)
16. Kuś, Z., Nawrat, A.M.: The limitation for the angular velocity of the camera head during object tracking with the use of the UAV. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 127–145 (2014)
17. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)

18. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. IEEE J. Robot. Autom. **RA-3**(4) (Aug 1987)
19. Sobel, D., Jedrasiak, K., Daniec, K., Wrona, J., Nawrat, A.: Camera calibration for tracked vehicles augmented reality applications. In: Innovative Control Systems for Tracked Vehicle Platforms. Springer, Berlin (2014)
20. Kniaz, K.: LMA, acces via Internet: http://kniaz.net/software/LMA.aspx. Acces: Feb 2014
21. Gavin, H.P.: The Levenberg-Marquardt method for nonlinear east squares curie-fitting problems, Department of Civil and Environmental Engineering, Duke University, 9 Oct 2013
22. Babiarz, A., Bieda, R., Jaskot, K.: Vision system for group of mobile robots. In: Nawrat, A. et al. (eds.) Advances Technologies for Intelligent Systems, SCI 440, pp. 137–154. Springer, Berlin (2013)
23. Official OpenCv website, acces via Internet: http://opencv.org/. Acces: Mar 2014
24. Iwaneczko, P., Jędrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 87–94, (2012)
25. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 157–175 (2013) (ISBN: 978-3-319-00368-9)
26. Bereska, D., Daniec, K., Fraś, S., Jędrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 117–189 (2013) (ISBN: 978-3-319-00368-9)
27. Jaskot, K., Babiarz, A., Sroka, M., Ściegienka, P.: Prototyp bezzałogowego pojazdu podwodnego – konstrukcja mechaniczna, panel operatora, Przegląd Elektrotechniczny, vol. 89, pp. 52–67 (2013)
28. Sroka, M., Ściegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzałogowego pojazdu podwodnego – układ stabilizacji i utrzymania zadanego kursu. Przegląd Elektrotechniczny **89**, 205–217 (2013)

Real-Time Multiple Laser Points Tracking

Artur Ryt, Dawid Sobel, Jan Kwiatkowski, Mariusz Domżał, Karol Jędrasiak and Aleksander Nawrat

Abstract Laser Point Tracking (LPT) is algorithm dedicated to rapidly moving laser-generated blob. By usage of few features: brightness, area and velocity, it is able track blurred and dark trail of laser. Further research were made to overcome problems, which occur while tracking two lasers simultaneously. The acquired results are promising. The presented algorithm for tracking multiple laser points might be applied to various multimedia presentation systems e.g. gimbal camera guidance during lectures or laser shooting ranges. The presented algorithm was compared against three state-of-the art object trackers using real-life test sequences. Only LPT tracker succeeded in the task. State-of-the art general purpose trackers were able to track the target for less than 10 % of the sequence. Due to the significant improvement of tracking quality it was possible to apply the algorithm and create the professional laser shooting range characterized by outstanding quality.

Keywords Real-time tracking · Blob tracking · Shooting range

1 Introduction

Precise knowledge of a plant orientation in space is necessary information for correct implementation of various engineering applications. As examples one can mention control of plants that rotates, rating of human limbs movement, or visualization of position of human body parts in virtual reality [1–6]. As an orientation we will understand Euler angles: R-roll, P-pitch and Y-yaw, around axis related to plant.

To estimate Euler angles the idea of sensor fusion is used, which is based on aggregation of measurement data coming from different sensors. The angles can be determined by gyroscope data by matrix integration of angular velocity [7, 8] or by accelerometers and magnetometers data, using algebraic dependencies [9–11].

A. Ryt (✉) · D. Sobel · J. Kwiatkowski · M. Domżał · K. Jędrasiak · A. Nawrat
Institute of Automatic Control, Silesian University of Technology, Akademicka 16,
44-100 Gliwice, Poland
e-mail: artur.ryt@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_11

Modern tracking algorithms are focused on complicated objects, which can be described with complex methods [8]. Object tracking survey [12] lists 4 common visual features: color, edges, optical flow and texture. These features are used to build models, create templates and probability densities functions in order to find representation of object. Novel trackers can define and learn about the object not only by initialization but online as well [13]. However, tracking simple laser blob, which is seemingly immutable in real-time even in environment close to laboratory's one becomes difficult task.

The basic reason of problems is fast laser movement. Because of its small dimensions it can move by path of few diameters between two frames. What is more, rapid moves make blob blurred and darker. It means none of features listed above is preserved. Size of blob can get multiplied or divided by few times even every frame. Once area of the blob is too big information about relevant position of the laser is lost and cannot be calculated from just single frame. Our suggestion is to estimate new position using velocity vector.

Last, but not least problem is real–time processing in high resolution. This is important task, because it can increase quality of the tracking. Ability to process frames with high frequency allows usage of *faster* camera. Faster camera can have shorter exposure time, what leads less blurred blobs [9].

Based on recent tracking review [14] we have chosen few trackers that can be able to handle problems connected with laser tracking. List of them, FragTrack [15] algorithm divides tracked object to set of fragmented images, where each fragment is represented and identified by histogram. During processing every image.

Second algorithm called VTD [16] bases on 2 steps—first is defining object's observation and movement model. Usage of this data can make tracker robust to simultaneous changes of movement and shape. In second step tracking is divided to group of elements, where each one traces single, different type of object change. In order to combine results of elements together it uses Interactive Markov Chain Monte Carlo (IMCMC) method.

Locally Orderless Tracking (LOT) [7] method is based on the combination of the image space and object appearance which allows tracking of targets that are deformed. Method adapts its operation to form of the object that is being tracked. If the object is a solid body, then the value of the arrangement coefficient is close to zero and can be used in methods based on spatial alignment such as pattern matching. Otherwise, when the shape of the object is changing, LOT method cannot take advantage of spatial fit and works by matching the histogram.

2 Laser Tracking

Algorithm describes feature based tracker. In order to determine whenever found blob is laser's point or noise we use two features: color and area of the blob. Both of them must match predefined, parameterized thresholds.

Tracking processing for every frame works in two different modes: regional or global search of blob [17–19].

In regional mode search is performed only in boundaries around previous laser's position. In case of fast blob's movement it starts to estimate position of the laser using the blob's velocity and contour.

If regional search fails or position of any laser in previous frame is unknown the global search is made. It scans whole frame for blob, which can be possibly the laser's one. Algorithms returns new position of the laser as centroid of the blob or flag that the laser does not exist if none blob has been found.

Schema 1 represents flowchart of single frame processing. It contains 17 blocks, which are separated in two possible modes.

Block 1: Initialization—memory allocation, loading of configuration parameters and previous laser's position.

Decision block 2: One of two modes is selected. It goes for regional scanning if last position of all lasers is known (block 3) or for global search of blob otherwise (block 11).

Mode 1—region search:

Blocks 3: Perform regional scan for first unused pixel in region which contains in defined color interval. In order to achieve faster performance the loop step is parameterized.

Block 4: Create blob by using flood fill algorithm with same loop step. Mark all pixels in the blob as used.

Decision block 5: Test does the blob's size meets the minimal requirements. If it's passed continue processing in block 7. Otherwise return to the search for blob by going to block 6.

Decision block 6: If whole region has been scanned the global search is required, so algorithm switches to second mode, which starts on block 11. Otherwise it continues region scan loop on block 3.

Decision block 7: Estimation of laser point is not always triggered, because it's not always required and sometimes can decrease the quality of tracking [20]. We perform it only when the blob's area is bigger than defined threshold. If that's true algorithm continues to block 8, otherwise it goes to block 10.

Block 8: Calculate velocity vector as subtraction of two positions: centroid of the blob and the laser's previous position.

Block 9: Estimate and set new laser position as cross point of blob's edge and calculated velocity vector. Search for this frame is finished, continue to end block 17.

Block 10: Set new laser position as centroid of the blob. Search for this frame is finished, continue to end block 17.

Mode 2—global search:

Block 11: Perform loop on whole frame with parameterized step and push every pixel with proper shade of grey on list. Continue to block 12.

Block 12: Select median pixel in the list. If frame does not contain too much of noise pixels and laser's blob exist on image it should select one of the blob's pixels.

Block 13: Filter noise data by removal from the list every pixel which is too far from median pixel. Radius of filtering is parameterized.

Decision block 14: If median pixel is one of laser blob's pixels the filtering in block 13 will return just blob with few noise pixels. Otherwise list will be filtered around random noise pixel, so it will get almost empty, containing only a little number of noise pixels. In the first case, when size of the list is bigger than given threshold we can continue processing of the blob in block 17. Otherwise laser was not found and algorithm continues to block 15.

Block 15: Laser was not found. Search for this frame is finished, continue to end block 17.

Block 16: Set new laser position as centroid of the blob. Search for this frame is finished, continue to end block 17.

End block 17: Return new position of the laser or set flag of its absence (Fig. 1).

3 Test-Based Parameters Selection

There are several parameters used to control performance speed and quality of tracking [21–23]. All of them should be selected empirically because it can vary depending on configuration of used system e.g. camera frame rate and resolution, laser's power, background noise intensity.

Our test data includes three recordings, which we named as *Easy, Medium* and *Hard.* All of them contains two moving lasers' blobs, weak one and strong one. In these tests we will focus on weak laser, as its case is harder to process. In *Easy* test lasers' blobs move slowly and never blur. In *Medium* test lasers move with average speed and blur occasionally. In *Hard* test they move dynamically, they are blurred through majority of record and finally they disappear for few frames. Database of reference positions and areas for each frame has been made manually, so it can be used for analysis of algorithm (Plots 1 and 2).

Basic parameter, which defines quality of tracking, is threshold shade of gray. Approach we propose bases on histogram of pixel brightness on sample tiles.

Histograms were generated for 400 × 400 tiles, because this is size of our scan region. In order to calculate relevant threshold between laser and background we have used method of entropic segmentation described in [24]. It finds *topt* which is threshold that maximize the sum of Tsallis entropies for object and background, which is considered optimal value (Plot 3).

For static blob calculated *topt* is equal to 21 (Fig. 2 and Plot 4).

For blurred blob calculated *topt* is equal to 12 (Fig. 3).

We have decided to use bigger threshold *topt* = 21, because it's more safe solution, more proof to environmental background changes [25].

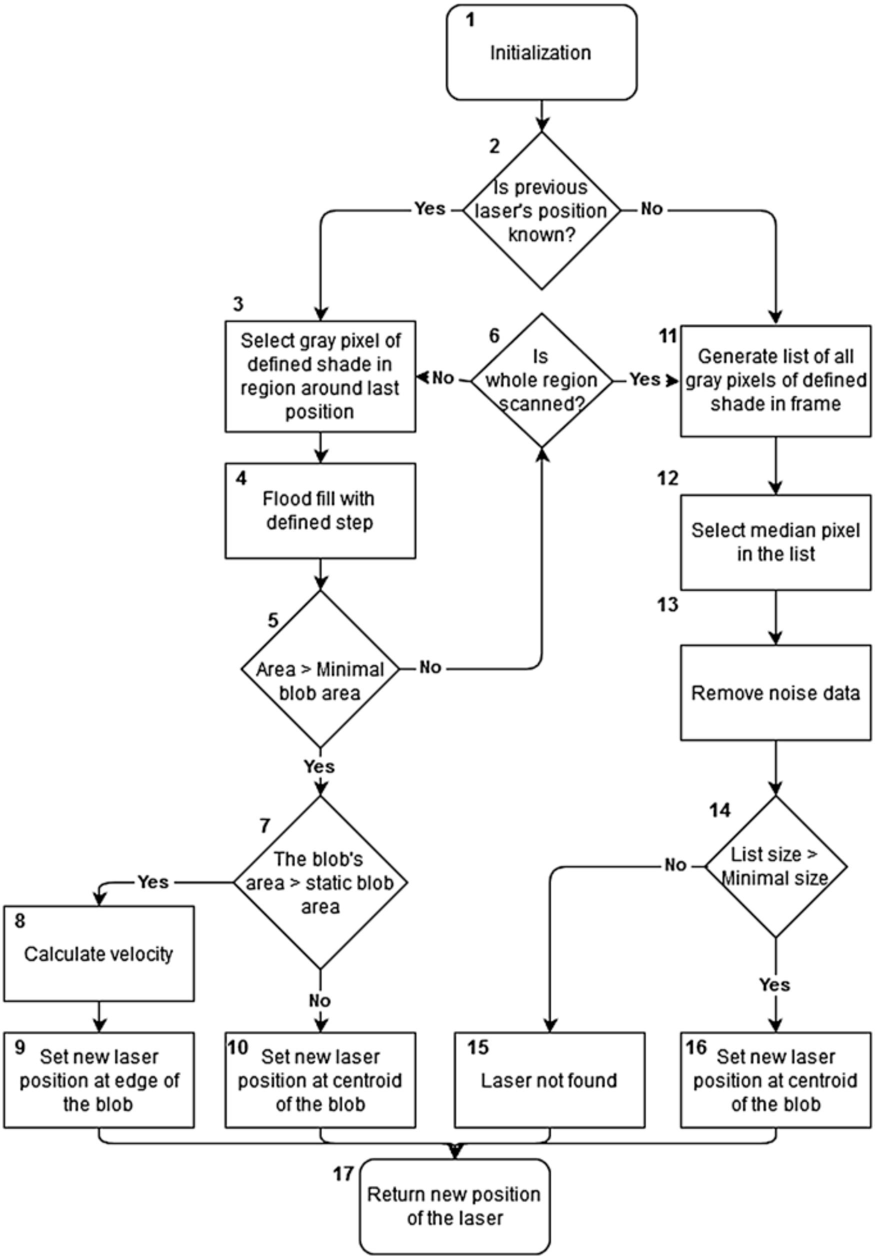
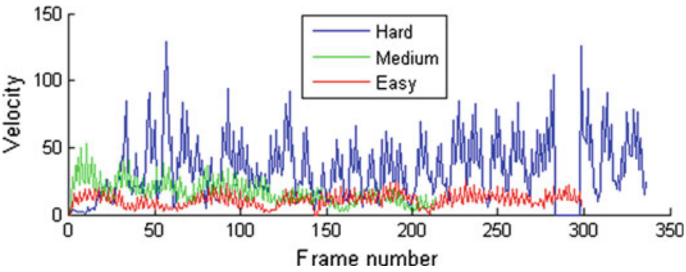
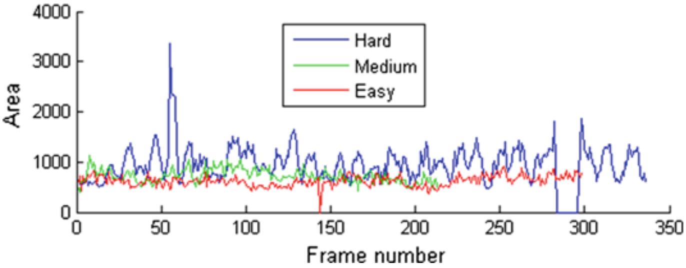

1
Initialization
2
Is previous laser's position known?
Yes
No
3
Select gray pixel of defined shade in region around last position
6
Is whole region scanned?
No
Yes
11
Generate list of all gray pixels of defined shade in frame
4
Flood fill with defined step
12
Select median pixel in the list
5
Area > Minimal blob area
No
13
Remove noise data
Yes
7
The blob's area > static blob area
Yes
No
14
List size > Minimal size
No
Yes
8
Calculate velocity
9
Set new laser position at edge of the blob
10
Set new laser position at centroid of the blob
15
Laser not found
16
Set new laser position at centroid of the blob
17
Return new position of the laser

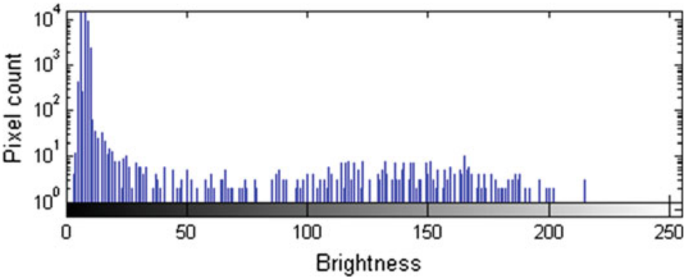
Fig. 1 Single frame processing flowchart



Plot 1 Comparison of velocity of the blob in particular tests



Plot 2 Comparison of area of the blob in particular tests



Plot 3 Logarithmic histogram for static blob

Three other parameters can be selected by analysis of blob's velocity and area in listed tests. These are: dimension of scan region, minimal area of blob, trigger area for position estimation.

In order to track laser efficiently it cannot leave scan region even during fast movement. The biggest recorded velocity in this test set is 129. It means region should be square with half of side length equal to 129. We have added safety reserve in case of faster moves and set this variable 400.

Minimal area of the blob can be calculated as minimal non-zero area of Easy test, which contains only static blobs. In our test it was equal 357. For triggering position estimation we have used double of minimal area of the blob (Plots 5, 6 and 7).

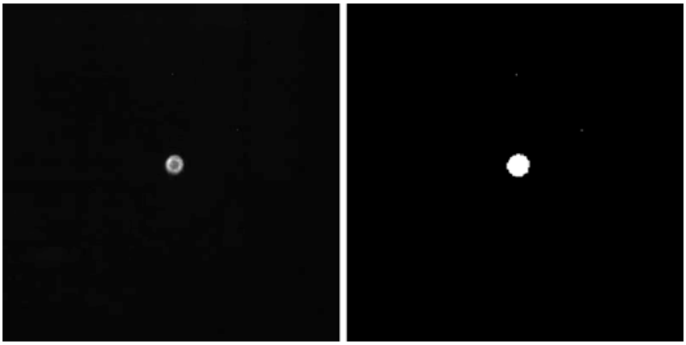
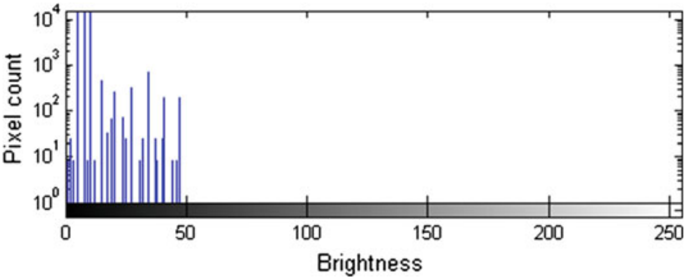


Fig. 2 Comparison of static blob (*left* image) and its threshold bitmask (*right* image)



Plot 4 Logarithmic histogram for blurred blob

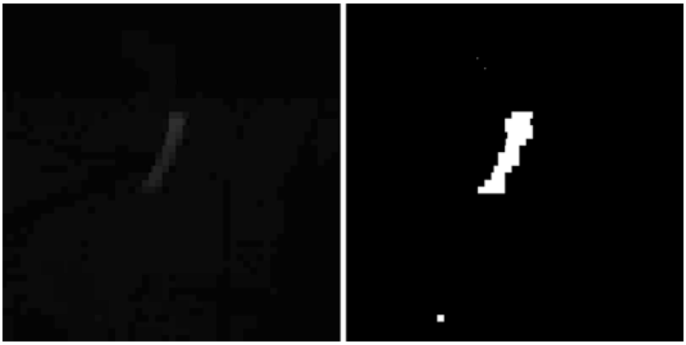
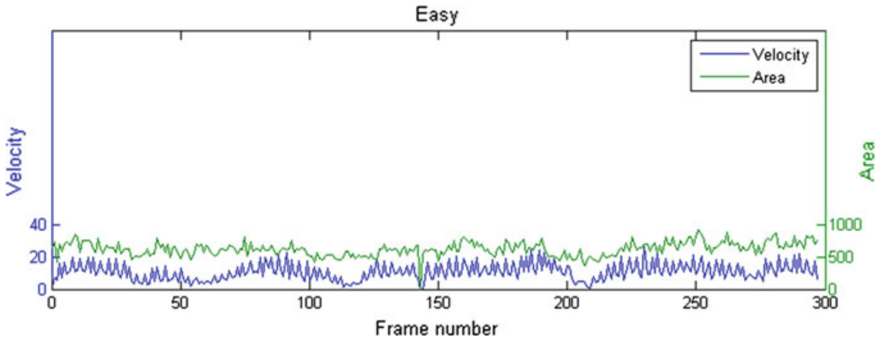
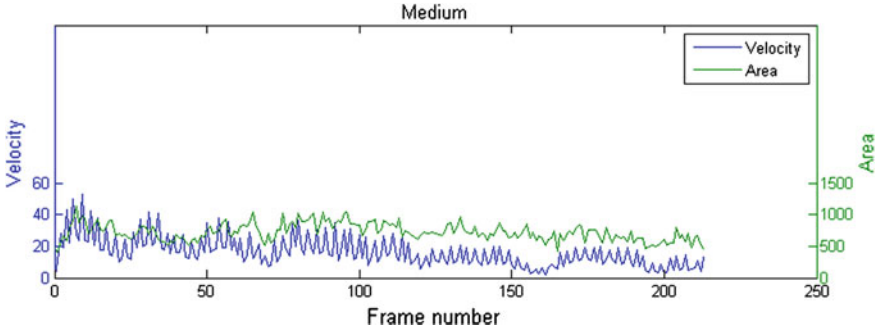


Fig. 3 Comparison of blurred blob (*left* image) and its threshold bitmask (*right* image)

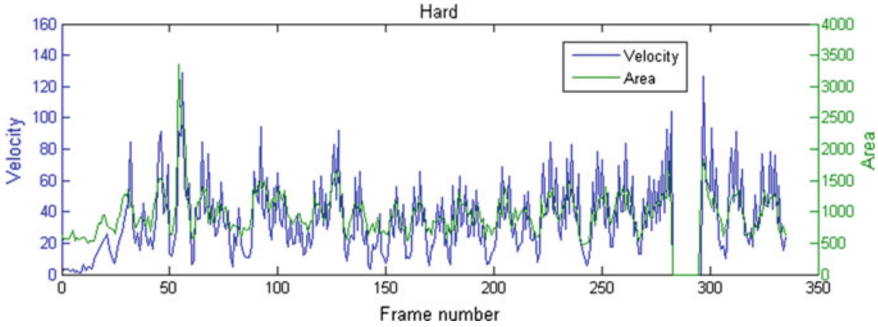
In order to achieve speed boost we can increase step variables of search loops and flood fill. It not only decreases number of iterations required, but also decreases noise influence. It happens because there is chance of stepping over noise blobs, which dimensions are smaller than step size. These parameters should be calibrated online to get desired frames per second processing ratio.



Plot 5 Relation between area and velocity in test easy



Plot 6 Relation between area and velocity in test medium



Plot 7 Relation between area and velocity in test hard

Further tests were made for simultaneous tracking of two laser points with same dimension, but different intensity. Second laser was much stronger than the one analyzed above (Fig. 4).

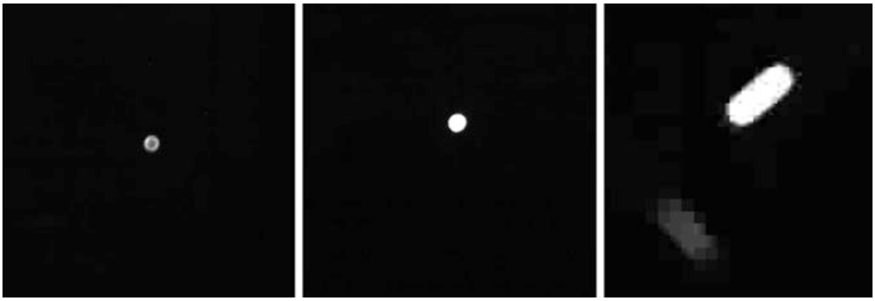
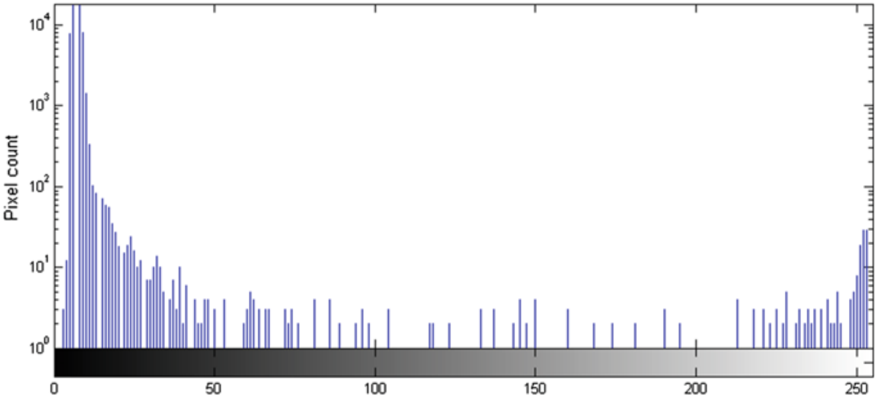


Fig. 4 Comparison of static weak laser blob (*left* image), static strong laser blob (*middle* image) and blurred blobs



Plot 8 Logarithmic histogram for strong laser's static blob

In order to distinguish blobs we need to add another threshold. Weak laser's brightness must be limited not only by minimal value, but also by maximal. According to histogram for strong laser blob (Plot 6) it can be set to 241, what's the value of minimal significant, present brightness. Same value can be used as minimal value for strong laser (Plot 8).

4 Empirical Results

In this paper, using the idea of construction of the so-called naive classifier orientation estimation algorithm in the 3D space, using three independent Kalman filters is proposed. It was assumed that the angular velocity ω^G determines a vector a consisting of three independent events. This incorrect assumption was made intentionally and it made possible to build independent Kalman filter for each axis

RPY. It enables simplifications in implementation on the target device comparing to classical.

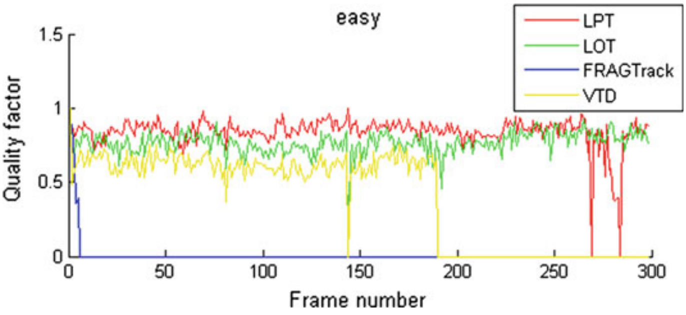
We have tested proposed Laser Point Tracker (LPT) on three test records described earlier. We have compared results with three other trackers: LOT, FragTrack and VTD. Tests were run on default parameters on implementations published by their authors.

Quality factor q was calculated as follows:

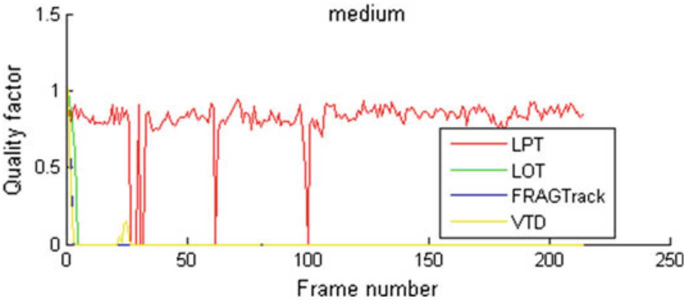
$$q = \begin{cases} 2\frac{(A \cap B)}{(A+B)}; & A > 0, B > 0 \\ 0; & A = 0, B > 0 \\ 1; & A = 0, B = 0 \end{cases} \tag{1}$$

where A is blob area in reference data, B is blob area returned by tracker. It represents how identical are these areas. Values vary from 0 to 1. Area value is equal to 0, when no blob was found on frame (Plots 9, 10 and 11).

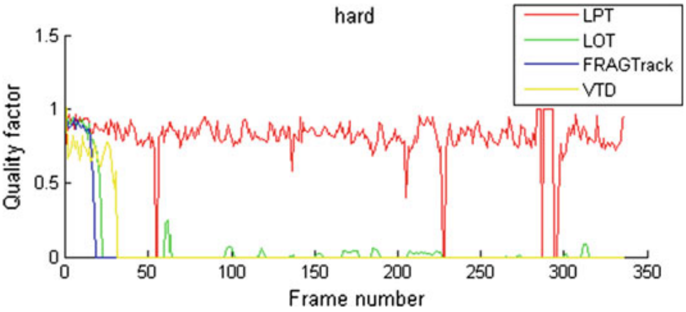
Proposed algorithm LPT is able to track laser's point almost constantly. Even if it loses laser for few frames it is able to find it again. This is the biggest advantage that it has over other trackers. High speed and strong blur in Hard test were



Plot 9 Quality factor for different tracker over time for easy test



Plot 10 Quality factor for different tracker over time for medium test

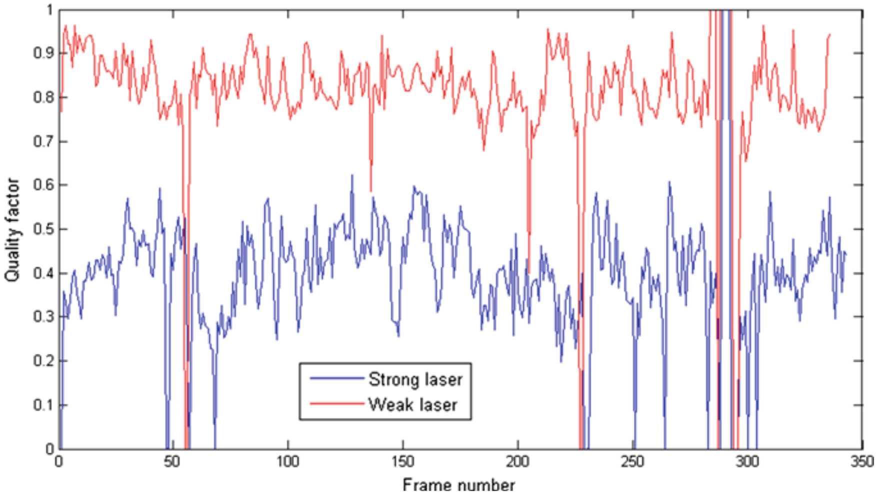


Plot 11 Quality factor for different tracker over time for hard test

successfully LOT and VTD were able to follow laser in Easy test. However, once the blob stated to blur in Medium test both of them have got stuck. Unfortunately static laser's blob wasn't complicated enough to get properly recognized by FragTrack (Plot 12).

What is more, LPT is able to track two lasers simultaneously without losing quality of tracking. Weak laser's quality factor is same as for single tracking. Strong one's has worse quality factor value over time, but is as robust as for weak one (Fig. 5).

LPT algorithm has been used as part of the real virtual range system. This system includes gun-shaped, wireless marker with IR laser attached, projector, high resolution, fast camera (1600 × 1200, 60FPS) with IR filter and central computer unit. Usage of IR wavelength allows filtering out majority of background noise. Game scene is shown on screen by projector, player can point at desired target with



Plot 12 Quality factor for two different lasers over time for hard test

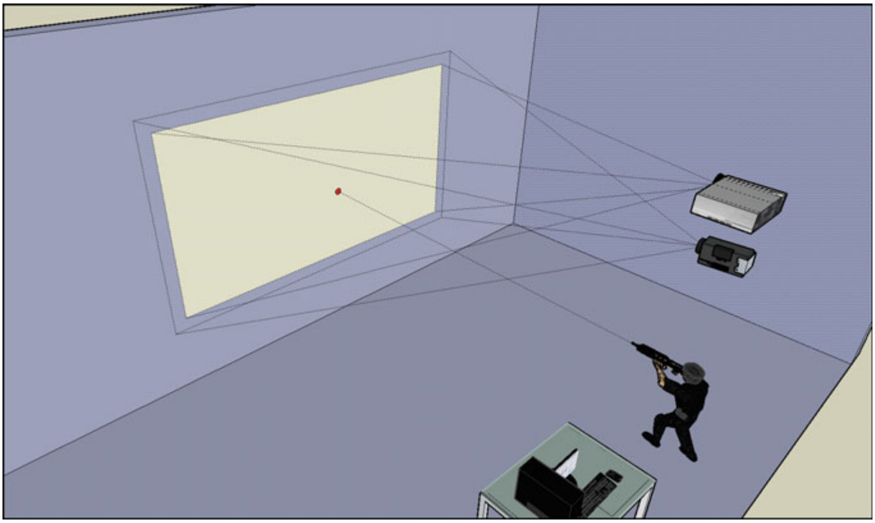


Fig. 5 Real virtual range system schema

marker. Camera’s stream is constantly processed, so position of the laser is known. Player can interact with game by built-in marker’s buttons.

Acknowledgement This work has been supported by Applied Research Programme of the National Centre for Research and Development as a project ID 178438 path A—Costume for acquisition of human movement based on IMU sensors with collection, visualization and data analysis software.

References

1. Bibik, P., Zasuwa, M., Zugaj, M.: Research and training simulator of unmanned quad rotor. In: 18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 403–407 (2013)
2. Babiarz, A., Jaskot, K., Koralewicz, P.: The control system for autonomous mobile platform. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 15–28 (2013)
3. Bereska, D., Balcewicz, R., Garczyński, M.: Implementacja magistrali CAN i protokołu CANopen w robocie edukacyjnym. Szybkobieżne Pojazdy Gąsienicowe **1**, 157–162 (2008)
4. Niezgoda, T., Panowicz, R., Sybilski, K., Barnat, W.: Numerical analysis of missile impact being shot by rocket propelled grenades with rod armour. WIT Trans. Model. Simul. **51**, 625–633 (2011)
5. Kuś, Z., Nawrat, A.M.: The limitation for the angular velocity of the camera head during object tracking with the use of the UAV. In: Innovative Control Systems for Tracked Vehicle Platforms, Studies in Systems, Decisions and Control, vol. 2, pp. 127–145 (2014)
6. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 157–175 (2013). ISBN: 978-3-319-00368-9

7. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
8. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: 9th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China (2009)
9. Jedrasiak, K., Nawrat, A., Daniec, K., Koteras, R., Mikulski, M., Grzejszczak, T.: A prototype device for concealed weapon detection using IR and CMOS cameras fast image fusion. In: Computer Vision and Graphics, vol. 7594, pp. 423–432. Lecture Notes in Computer Science, Springer, Berlin (2013)
10. Gałuszka, A., Skrzypczyk, K., Bereska, D., Pacholczyk, M.: Re-handling operations in small container terminal operated by reach stackers. World Acad. Sci. Eng. Technol. **70**, 674–677 (2010)
11. Czapla, T., Wrona, J.: Technology development of military applications of unmanned ground vehicles. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 293–309 (2013). ISBN: 978-3-319-00368-9
12. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surveys (CSUR) **38**(4), 13 (2006)
13. Jedrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. In: Computer Vision and Graphics, vol. 5337, pp. 391–399. Lecture Notes in Computer Science, Springer, Berlin (2009)
14. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking. Rev. Neurocomput. **74**(18), 3823–3831 (2011)
15. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 798–805, 17–22 June 2006
16. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Computer Vision and Pattern Recognition (CVPR) (2010)
17. Kuś, Z., Nawrat, A.: Object tracking for rapid camera movements in 3D space. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 5–76 (2013). ISBN: 978-3-319-00368-9
18. Kuś, Z., Nawrat, A.: Object tracking in a picture during rapid camera movements. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 77–91 (2013). ISBN: 978-3-319-00368-9
19. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, vol. 8671, pp. 302–315. Lecture Notes in Computer Science (2014)
20. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics, vol. 7594, pp. 87–94. Lecture Notes in Computer Science, Springer, Berlin (2013)
21. Demski, P., Grzejszczak, T., Jędrasiak, K., Mikulski, M.: Automatic targeting sentry turret for distributed systems. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 47–55 (2013). ISBN: 978-3-319-00368-9
22. Jędrasiak, K., Bereska, D., Nawrat, A.: The prototype of gyro-stabilized UAV gimbal for day-night surveillance. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 107–115 (2013)
23. Bereska, D., Daniec, K., Fraś, S., Jędrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 117–189 (2013). ISBN: 978-3-319-00368-9
24. Portes de Albuquerquea, M., Esquef, I.A., Gesualdi Mello, A.R., Portes de Albuquer-que, M.: Image thresholding using Tsallis entropy. Pattern Recogn. Lett. **25**(9), 1059–1065 (2004)
25. Switonski, A., Josinski, H., Jedrasiak, K., Polanski, A., Wojciechowski, K.: Classification of poses and movement phases. In: Computer Vision and Graphics, vol. 6374, pp. 193–200. Lecture Notes in Computer Science, Springer, Berlin (2010)

Part II
Design, Construction and Analysis for Purpose of Innovative Systems

Performance and capabilities of simulation systems largely depend on the used components. The part presents best practices for design and construction of various types of objects, both real and virtual.

It is presented how to design an electronic system for a mobile robot. Its electronic circuit was divided into functional blocks and each of them is described separately. For each, the corresponding part of schematic was presented and explained. The most important decisions on selected parts or solutions are discussed and justified.

The part presents a new solution of a semi-active suspension system. It is based on a sky-hook strategy model. This solution in 2S1 tracked platform is applied to improve driving comfort as very important factor for the vehicle crew efficiency. The solution is applied in two versions of the 2S1 vehicle suspension model. First one is a basic model. This suspension is based on existing construction of the 2S1 platform suspension. It is based on torsion bars. Second one is a modified model, based on spiral torsion springs. In this model a new solution of idler mechanism is applied. It provides constant tension of the tracks. Semi-active suspensions simulation results are compared with results of models with passive versions of the suspension to highlight the improvement level. Results of all model simulations are compared and analyzed to improve comfort and stability level in conditions of the modern battlefield.

Alternatively the part also presents the key elements of the construction of a four-wheeled armored personnel carrier models for simulation in the VBS2 environment. These principles are not only limited to this environment and are also used in other simulation engines, as well as in development environments and other manufacturers. The paper describes the components of the vehicle model as well as how it was configured and exemplary realizations of these elements.

The plan of the experiment, the pre-processing, and analysis of data are crucial for the success of the research. As part of the chapter, an exemplary research requiring careful preparation is presented, which is often impossible to replicate.

This part includes a number of important challenges in the fields mentioned above. At the same time valuable suggestions and conclusions from authors are presented and discussed in detail.

Numerical and Experimental Analysis of a Truck Frame

Eugeniusz Rusiński, Artur Iluk and Mariusz Stańco

Abstract Calculating the stress level of the load-carrying structure of a vehicle's frame based on data obtained from literature can be unreliable. It is crucial to know the operating characteristics of a given vehicle. In the presented example, it was necessary to understand the phenomena that occur during operation of a vehicle, which could not travel a selected test distance without sustaining damage to the load-carrying structure. Attempts to improve the structure based solely on the analysis of damage did not bring satisfactory results. Only after performing numerical simulation of vehicle operation and an in-depth analysis of the problem was it possible to modify the frame so that it would pass the roadworthiness tests and qualify for registration.

Keywords Numerical analysis · Experimental analysis · Truck frame

1 Introduction

Frames of trucks and other vehicles are the most important load-carrying structures, which house other systems and components, i.e. the engine, powertrain with drive axle, front axle with suspension, cabin, body, fuel tank, spare wheel and other systems. In order to properly construct a geometric model of such a frame, which is dimensioned according to criteria for tensile strength and fatigue strength, numerical methods must be applied, i.e. the finite element method [1, 2]. Since the introduction of this method into engineering, it has been possible to design structures with optimal mass and strength. The development of IT contributed to the popularization of this method. To apply FEM in practice, the designers need to have a complete understanding of machine construction, material strength and of the

E. Rusiński (✉) · A. Iluk · M. Stańco
Institute of Machines Design and Operation, Wrocław University of Technology,
Lukasiewicza 7/9, 50-371 Wrocław, Poland
e-mail: eugeniusz.rusinski@pwr.edu.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_12

numerical method itself [3, 4], which requires a detailed analysis and interpretation of the obtained results [5]. Whether the obtained results are correct is determined by properly defined loads based on the actual operating conditions of a given machine. What follows is an example of the use of the finite element method in addressing the problem of short life span of trucks [6]. The calculations are based on a vehicle, whose frame was damaged during a test drive. The article presents a procedure which begins with the identification of the stress level in the frame and ends with tests that verify the obtained results of the numerical calculations.

2 Characteristics of the Frame

During truck operation and field tests, the frame of the vehicle developed fatigue cracks. The frame is composed of two longitudinal members, which are interconnected by cross members attached by means of gusset plates. The longitudinal members are the main load-carrying members of the truck frame. Due to the requirement of high torsional flexibility they are manufactured in compactors or presses as extrusions of thin-walled open profiles [7]. They are entirely made of uniform strips of formed sheets, usually U-shaped. This shape guarantees high torsional flexibility and high resistance to transverse loads produced by the transported cargo and the force from the wheels. An example of a truck frame structure is shown in Fig. 1.

The longitudinal member, which is the main load-carrying component of the frame, is usually made of a U-shaped bar running along the vehicle. It reinforces the load-carrying structure and is used to attach the components of the undercarriage and to support the body. The bending angle of U-shaped beams of longitudinal

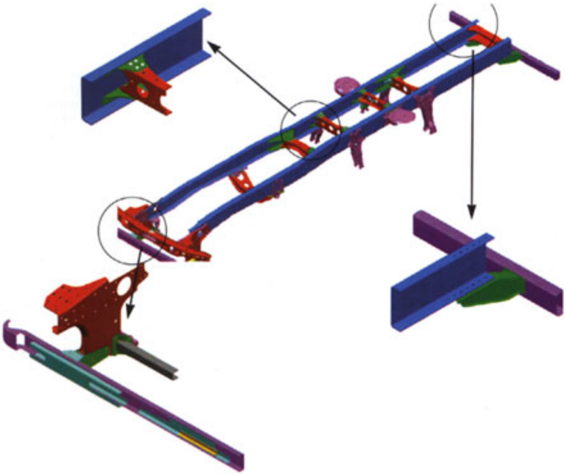


Fig. 1 Structure of a truck frame

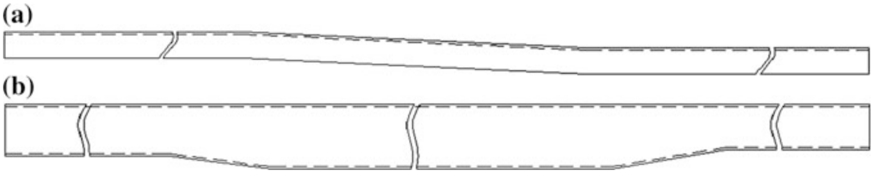


Fig. 2 Longitudinal member of a truck frame: **a** top view, **b** side view

members varies between 3° and 5°. There are also beams with a variable U-shaped section, as shown in Fig. 2.

The cross member is made of open or closed profiles and its main function is to maintain a constant distance between the longitudinal members, transmit lateral forces and torsion of the frame. The following types of cross beams can be distinguished:

- front and rear cross beams, aka bumpers,
- cross beams to support the engine,
- cross beams for transmitting loads from suspension system, etc.

Examples of cross beam shapes used in truck frames are depicted in Fig. 3.

Cross beams are usually connected to the longitudinal members by means of gusset plates, which are welded, riveted or attached by reamer bolts. Holes in longitudinal members are drilled only in the web of the longitudinal member of the frame, as shown in Fig. 4. This guarantees high fatigue strength and does not introduce imperfections into the longitudinal member in locations of maximum stress. An example of an optimum connection between a circular-section cross beam and a longitudinal member by means of the HUCK riveting system is illustrated in Fig. 5.

The analyzed vehicle is designed to transport 15-foot containers with their center of gravity placed above the axis. The mass of a full container was 8700 kg. The load-carrying structure of the vehicle is composed of two frames: the main frame and the auxiliary frame [8]. The main frame is based on longitudinal members with a U-shaped section, which are reinforced by inserts with an identical U-shaped section. The longitudinal members are interconnected by seven cross beams of varying sections. The auxiliary frame is also composed of two U-shaped longitudinal members, but their moment of inertia is lower. These longitudinal members are connected by two cross beams and two container-supporting additional beams. The load-carrying structure is shown in Fig. 6. In order to pass the roadworthiness test, a vehicle should travel at least 3000 km in off-road conditions, as it is designed to transport containers on unpaved roads with varied topography. If upon completing this distance the vehicle has not sustained any damage to the load-carrying structure of the frame and other components, it can be registered.

The aim of the study was to identify the causes of cracks that had formed during the test drive in the frame near the front container-supporting beam, as well as the causes of cracks in rivets that hold the joints connecting the main frame with the

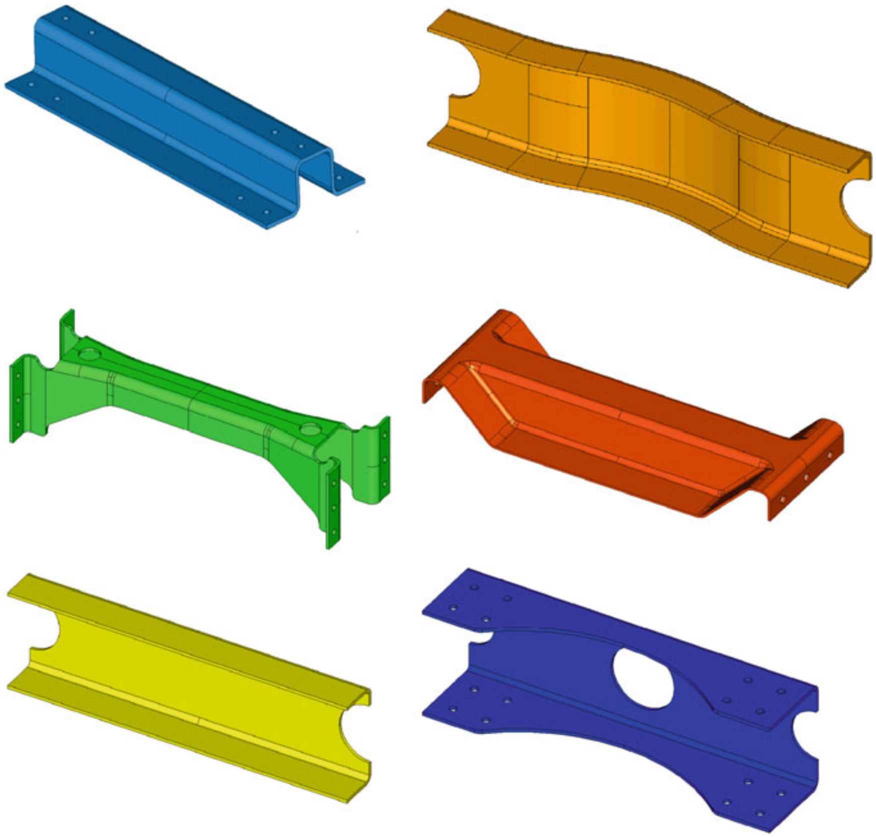


Fig. 3 Examples of different types of cross beams that connect longitudinal members in trucks

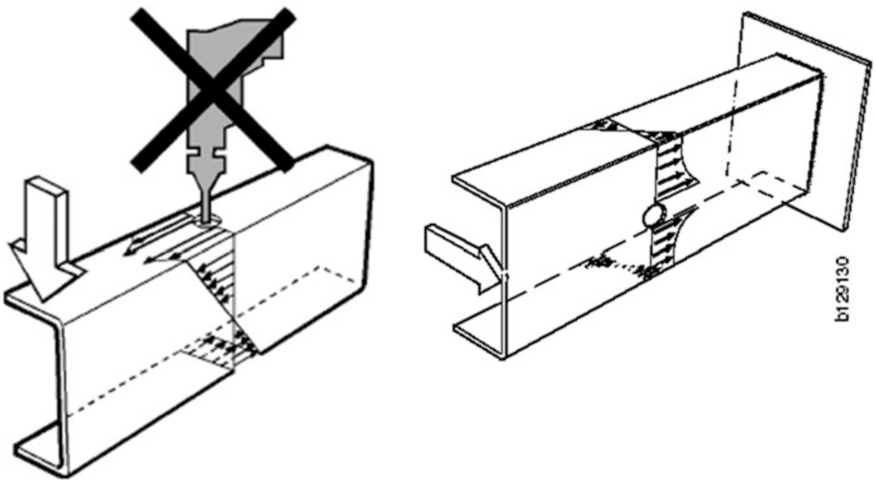


Fig. 4 Proposed locations for holes, which take into account the maximum stress

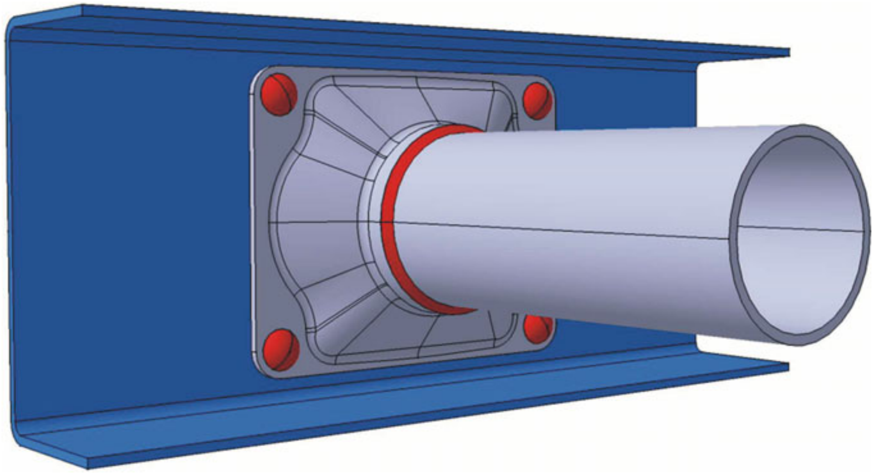


Fig. 5 Connection between a pipe cross beam and a longitudinal member using HUCK bolts

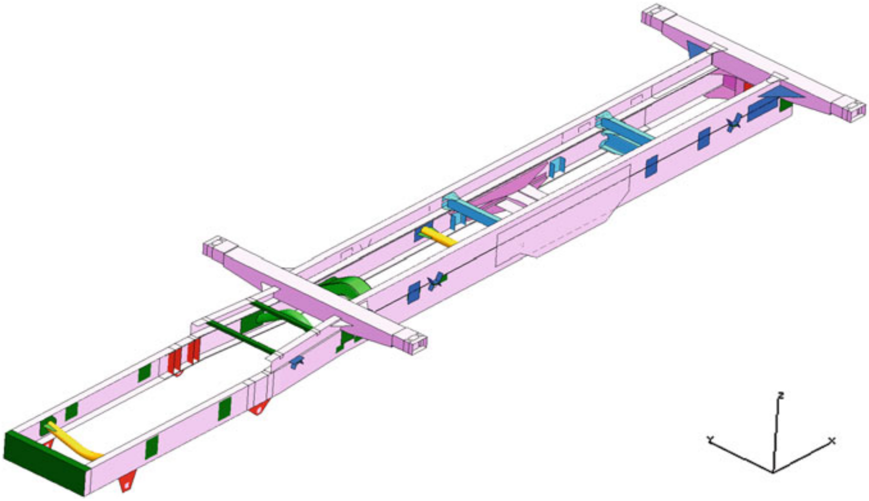


Fig. 6 Geometric model of the frame

auxiliary frame. The secondary goal of the study was to define critical locations where, after longer operation, fatigue cracks could form in welds or connecting elements.

To perform a strength analysis of the frame, a numerical calculation model was created. Due to the fact that the whole structure of the frame and individual elements is a thin-walled structure, the analysis used shell elements with six degrees of freedom in each node. In the case of auxiliary elements which model the

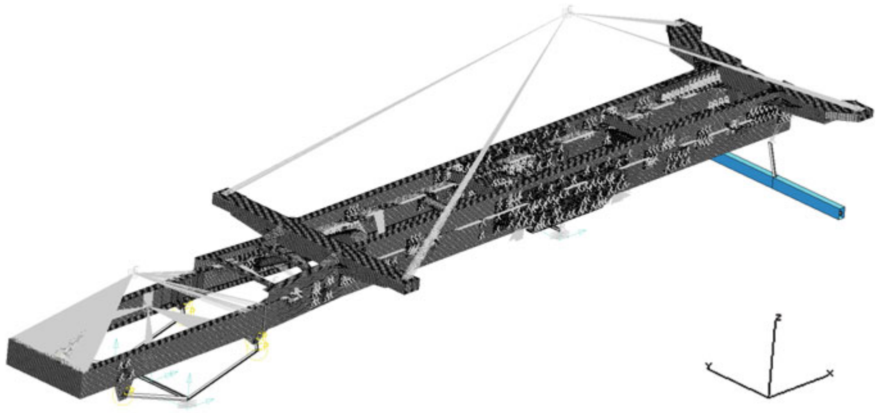


Fig. 7 Numerical model of the frame

suspension, as well as all the different types of connections and inertial loads, one-dimensional elements and solid elements were used. The numerical model was created by generating a mesh of finite elements from the previously created geometric model. The discreet model is shown in Fig. 7.

3 Definition of Loads

The vehicle's load-carrying structure is mainly subject to external loads, which determine its strength and life span. Constant external loads generated by unladen kerb mass, mass of components, as well as the mass of the transported cargo, have little effect on the fatigue strength of the frame. The life span of a frame in a higher mobility vehicle depends mainly on random tractive forces, which significantly affect the fatigue strength of the frame. Such forces occur when the vehicle travels over irregular surfaces, which is typical of paved and unpaved roads. When performing structural calculations, the automotive designer cannot establish all loads that act or will act on the load-carrying structure of the vehicle. Therefore, calculations also take into consideration the loads which occur during operation, i.e. the following:

- symmetrical vertical loads,
- asymmetrical vertical loads,
- longitudinal loads,
- lateral loads.

In the case of higher mobility vehicles, practice shows that if the structure is capable of transmitting sporadically occurring maximum loads without sustaining damage, then such a structure also has sufficient fatigue strength.

The problem of calculating loads is thus limited to finding the dynamic coefficients of static forces. The calculations of these coefficients were based on data found in existing literature and manufacturer's specifications. These coefficients were also verified by tests on the analyzed vehicle during operation.

4 Analysis of Results

The finite element method was applied in calculating strength of the load-carrying frame and the auxiliary frame. The stress values resulting from the numerical analysis were used to define zones in the frame with insufficient strength. Examples of contours of Huber-Mises equivalent stress are shown in Figs. 8 and 9. The critical zones where the frame could be damaged, defined using the above method, were consistent with the results of the prior roadworthiness tests.

The obtained results served as the basis for structural changes, which were introduced in the frame in order to decrease the stress level in zones with highest loads, and to prevent damage to the frame during operation. Because of the fact that the structure cracked during operation, additional reinforcements were introduced to the front container-supporting beam (Fig. 10). The high level of tensile forces in elements connecting the main frame with the auxiliary frame imply that very strong shearing forces occur in these locations. This could be the cause of damage to rivets, which hold the element that connects both frames.

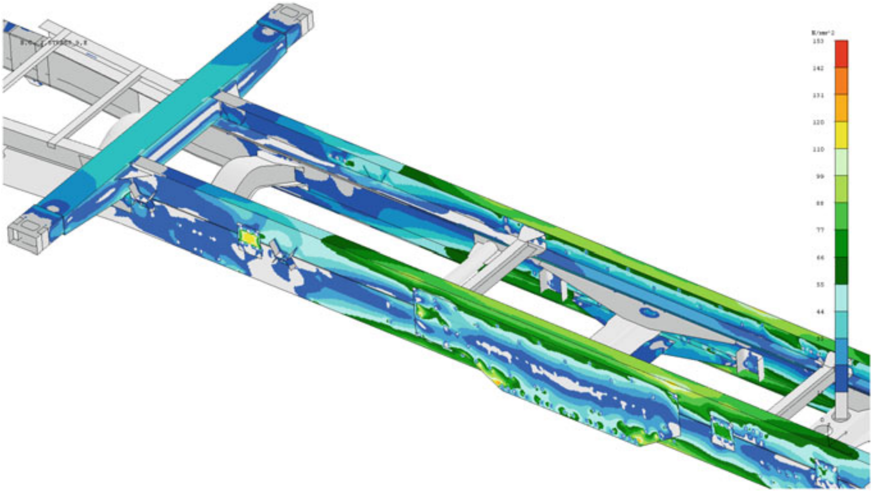


Fig. 8 Contours of Huber-Mises equivalent stress—vertical loads

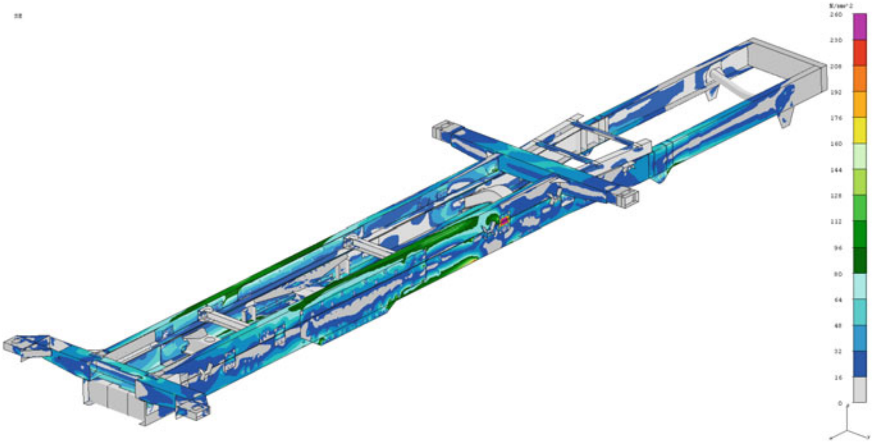


Fig. 9 Contours of Huber-Mises equivalent stress—frame torsion

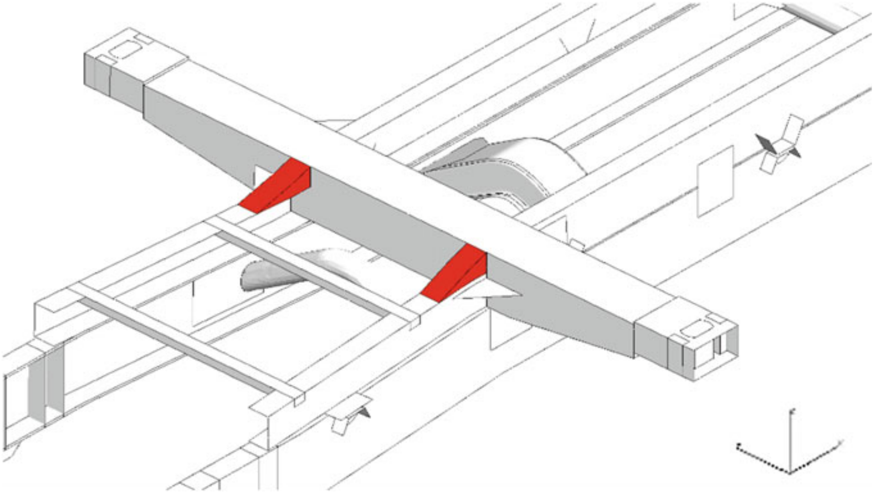


Fig. 10 Reinforcement of front cross beam

5 Modification Suggestions

A container located in the cargo area of a vehicle limits the torsional flexibility of the frame along the entire length of the container. Loading caused by twisting of the frame must be transmitted from the main frame to the auxiliary frame via binding joints. Due to the application of rigid joints, the connecting rivets were sheared.

The method of connecting both frames needed to be modified so as to eliminate the shearing of rivets. The finite element method was used to quickly check

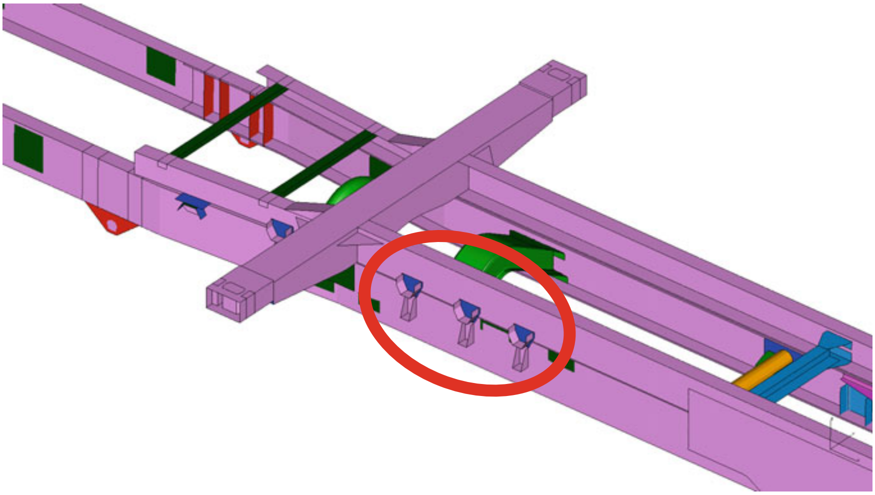


Fig. 11 The implemented flexible joint

different types of connections in order to find one that would transmit the loads from the container through the auxiliary frame to the main frame and back. It was proposed to substitute the rigid joint with a flexible joint (Fig. 11) in the form of long bolts with conical washers, so that during loading and unloading of the joint, there was constant pressure between the two frames. This solution eliminated the loosening of the joint.

Strength calculations for the load-carrying frame were performed again after the structural changes had been introduced. It was observed that the level of stress acting on the modified frame decreased and was lower than the assumed acceptable stress values. The modifications required that a new auxiliary frame be constructed, but the original frame was not suitable for operation due to damages.

6 Experimental Tests

In the next stage the stress level was calculated for the modified load-carrying structure of the vehicle. The frame was tested in individual stages of construction, and finally during operation. Since the auxiliary frame was entirely welded, the weld joints were inspected before placing the frame on the vehicle. The objective of the inspection was to eliminate welding faults and detect potential welding inconsistencies, and, via the magnetic and powder method, to detect potential discontinuities in surface and sub-surface materials in the forms of cracks (Fig. 12) [9]. It was observed that all the weld joints in the frame were done properly.

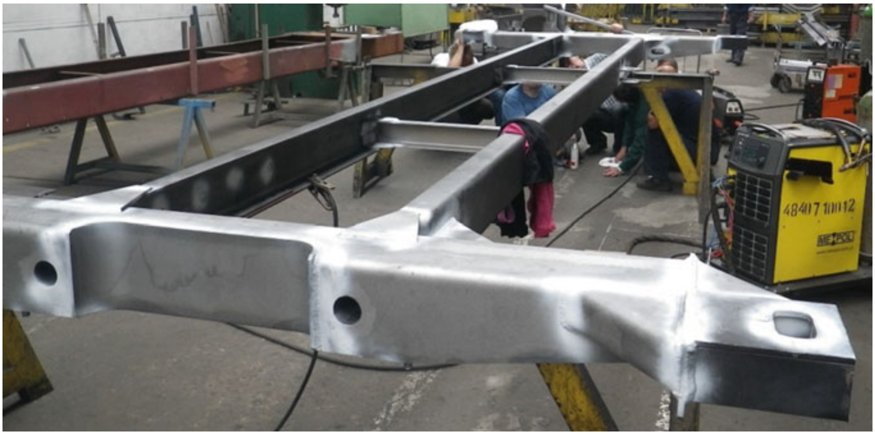


Fig. 12 Container-supporting frame—inspection of weld joints

In the last stage, experimental tests were performed on the vehicle. Experimental tests were carried out in two phases:

- Measurement of frame torsion.
- Measurements of vehicle operating under full load.

Frame torsion of the vehicle was measured in two configurations. First only the load-carrying frame was twisted in order to establish its maximum torsion angle and next the auxiliary frame was attached and the frame was twisted again. This served as the basis for calculating the torsional flexibility of the whole frame. The maximum torsional angle of the main frame of the vehicle was about 20.5° whereas after attaching the auxiliary frame, this value dropped to approximately 13°, and thus the auxiliary frame decreased the torsional flexibility of the vehicle by approximately 37 %. The tests measured the stress level in the frame using strain gauges attached in selected measurement points, which were chosen based on numerical simulations performed earlier. Three measurement points were selected on the main frame of the vehicle and another four measurement points were selected on the auxiliary frame (Fig. 13).

Figure 14 illustrates selected stress changes in the main frame without the auxiliary frame and with the front wheel raised to the height of 560 mm. The main frame was twisted by an angle of 17.9°, and the torsion angle of the front axle was 12.6°. The main frame itself is characterized by high torsional flexibility. With the auxiliary frame mounted on it, its flexibility is substantially decreased, which in turn decreases the torsion angle of the frame. With the front axle twisted by an angle of 12.6°, the total torsion angle of the load-carrying frame decreased by 46 % to the value of 9.8°. Figure 15 illustrates the stress change in selected measurement points during frame torsion obtained by lifting one of the wheels in the front axle.

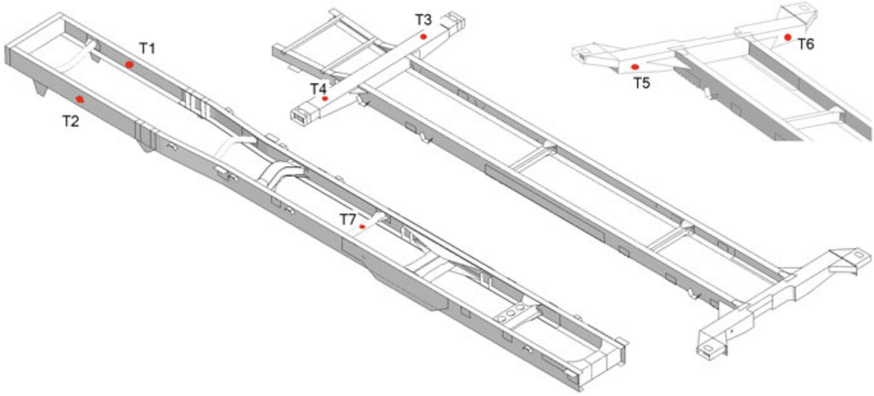


Fig. 13 Measurement points on the vehicle frame

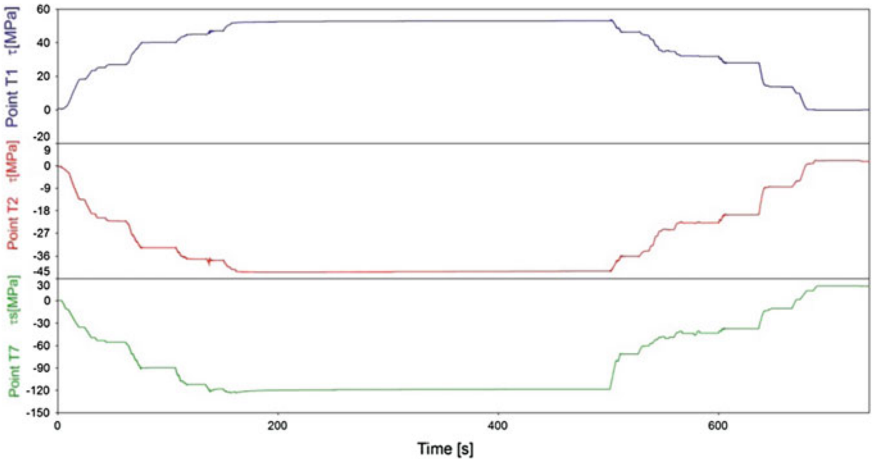


Fig. 14 Stress curves during torsion of main frame of the vehicle

In the final stage, measurements were taken during vehicle operation in realistic conditions. These measurements were taken in order to check the stress level in the frame during vehicle operation and to estimate the correctness of the assumed dynamic overload coefficients used in strength calculations. To this end, apart from strain gauges mounted on the vehicle frame, piezoresistive sensors were also installed, which measured the accelerations acting on the vehicle during drive. There was a total of 11 additional sensors installed in the vehicle, of which one measured the acceleration in the longitudinal direction (X axis), three in the lateral direction (Y axis) and seven in the vertical direction (Z axis). The location of

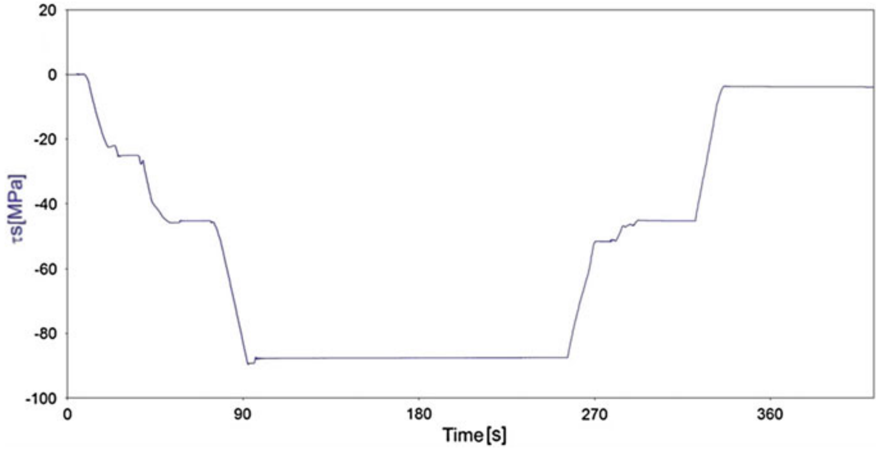


Fig. 15 Stress curves during torsion of main vehicle frame with auxiliary frame

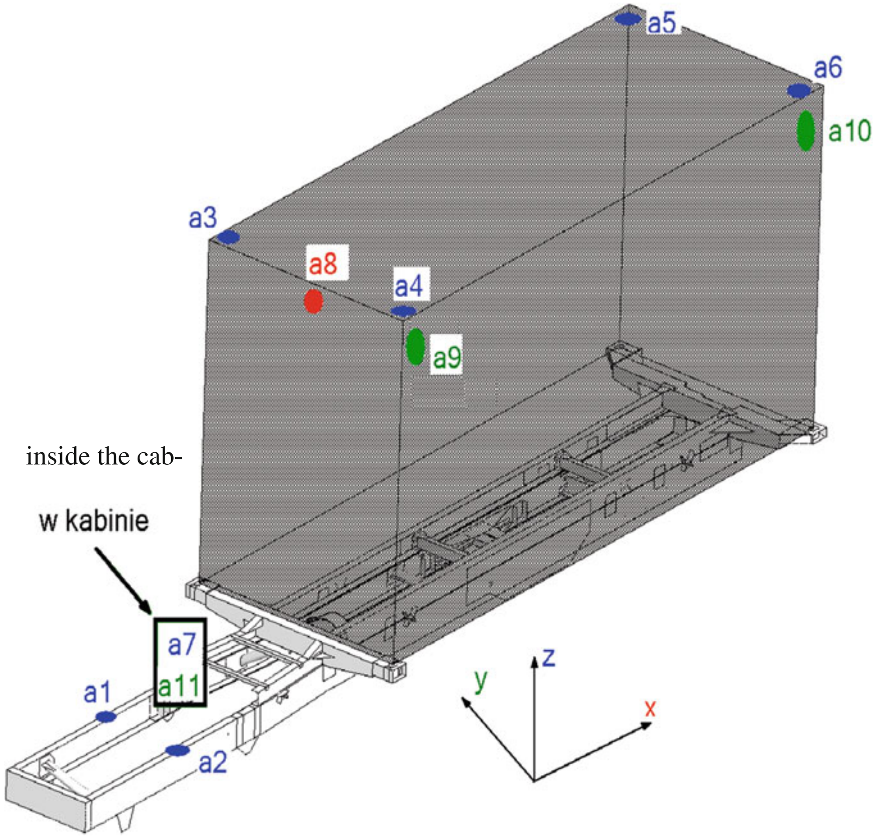


Fig. 16 Points where acceleration sensors were located

sensors is illustrated in Fig. 16. Additionally, a GPS receiver was installed in the vehicle, which allowed for the recording of the vehicle speed.

Figure 17 shows examples of accelerations acting on the container during vehicle operation.

The accelerations acting on the vehicle travelling on unpaved terrain with the container were slightly higher than those assumed in the calculations. This discrepancy confirmed the necessity to perform experimental tests, which allow for the verification of initial assumptions.

The required values of dynamic overload coefficients are different depending on the type of vehicle. This results from the vehicle's intended use, the stiffness of its load-carrying structure, characteristics of its suspension, and even the skill of the driver. A list of results obtained during vehicle operation is presented in Table 1. The speed of the vehicle was low as the terrain in which it was operating did not allow higher speeds. The average speed was approximately 16 km/h and the vehicle travelled at speeds ranging from 10 to 28 km/h.

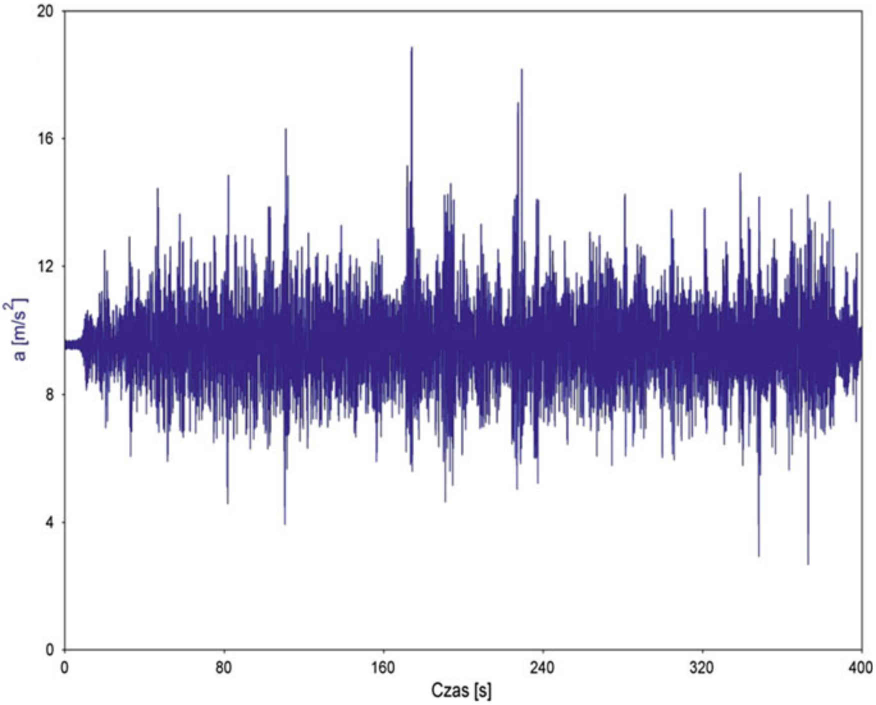


Fig. 17 Examples of accelerations acting on the container during vehicle operation

Table 1 Comparison of accelerations obtained during vehicle operation

No	a1 (m/s^2)	a2 (m/s^2)	a3 (m/s^2)	a4 (m/s^2)	a5 (m/s^2)	a6 (m/s^2)	a7 (m/s^2)	a8 (m/s^2)	a9 (m/s^2)	a10 (m/s^2)
3	5.99	6.34	5.2	5.46	7.73	8.49	–	3.97	–	–
4	9.56	8.37	8.74	9.17	10.67	10.57	3.11	5.34	4.78	4.38
5	12.99	10.48	7.16	11.94	7.96	8.78	3.06	5.05	3.51	4.89
6	9.31	8.86	5.92	6.08	10.92	10.16	3.1	5.71	3.63	3.92
7	11.03	10.22	7.63	8.12	11.73	8.55	3.73	5.79	4.91	4.59
8	11.12	9.28	8.09	6.3	9.15	9.36	2.97	4.87	3.32	3.52
9	5.99	6.76	5.04	5.37	9.69	9.5	3.49	4.86	4.86	4.6
10	7.83	7.21	5.7	5.99	7.42	7.79	2.03	3.9	3.22	3.29
11	12.35	8.07	7.27	5.77	13.29	12.29	4.06	5.83	4.88	4.77
12	6.88	6.8	5.26	5.33	7.13	7.27	2.35	4.12	3.09	3.98
13	8.55	8.35	6.16	5.51	9.44	11.2	4.25	5.48	4.48	5.25
14	16.86	12.31	10.4	8.8	10.34	10.02	2.5	5.15	3.63	4.6
15	17.92	13.4	11.37	11.08	11.58	11.51	3.47	5.52	3.94	4.92
Max	17.92	13.40	11.37	11.94	13.29	12.29	4.25	5.83	4.91	5.25

7 Conclusions

Calculating the stress level of the load-carrying structure of a vehicle's frame based on data obtained from literature can be unreliable. It is crucial to know the operating characteristics of a given vehicle. In the presented example, it was necessary to understand the phenomena that occur during operation of a vehicle, which could not travel a selected test distance without sustaining damage to the load-carrying structure. Attempts to improve the structure based solely on the analysis of damage did not bring satisfactory results. Only after performing numerical simulation of vehicle operation and an in-depth analysis of the problem was it possible to modify the frame so that it would pass the roadworthiness tests and qualify for registration.

The conducted torsion tests and tests during vehicle operation under full load confirmed that the changes introduced into the load-carrying structure yielded the expected results. These tests also allowed for the verification of the assumed dynamic overload coefficients related to vehicle mass.

References

1. Rusiński, E.: Finite Element Method. System COSMOS/M (in Polish). WKŁ, Warsaw (1994)
2. Zienkiewicz, O.C., Taylor R.L.: The finite element method. In: Solid Mechanics, vol 12, 5th edn. Butterworth-Heinemann (2000)
3. Kopczyński, A., Rusiński, E.: Bezpieczeństwo bierne, pochłanianie energii przez profile cienkościenne. Oficyna Wydawnicza PWr, Wroclaw (2010)

4. Smolnicki, T., Rusiński, E.: Superelement-based modeling of load distribution in large size slewing bearings. J. Mech. Des. **129**, 459–463 (2007)
5. Kocańda, S., Szala, J.: Podstawy obliczeń zmęczeniowych. PWN, Wyd (1998)
6. Ferenc, K., Ferenc, J.: Konstrukcje spawane. Warszawa, Projektowanie połączeń (in Polish), WNT (2000)
7. Rusiński, E., Czmochowski, J., Smolnicki, T.: Advanced finite element method for load-carrying structures of machines (in Polish). Oficyna Wydawnicza PWr, Wroclaw (2000)
8. Rusiński, E.: Design principles for supporting structures of self-propelled vehicles (in Polish). Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2002)
9. Prochowski, L.: Pojazdy samochodowe, Mechanika ruchu, podręcznik akademicki. WKiŁ, Warszawa (2008)

Designing and Implementing Elements of a Vehicle Model in the VBS2 Virtual Simulation Environment

Roman Wantoch-Rekowski and Konrad Szumiec

Abstract The paper presents the key elements of the construction of vehicle models for simulation in the VBS2 environment. These principles are not only limited to this environment and are also used in other simulation engines, as well as in development environments and other manufacturers. The paper describes the components of the vehicle model as well as how it was configured and exemplary realizations of these elements. As an example, a 4-wheeled armoured personnel carrier equipped with a turret and a mounted automatic gun, a two man crew and a troop compartment for four soldiers.

Keywords Virtual simulation · VBS · Vehicle model

1 Introduction

The construction of simulation models for the purpose of simulation environments is a complex process and must include definitions of elements related to different aspects. The presented model of the vehicle has all necessary elements for its full simulation in the VBS2 environment [1, 2]. In the following part of the paper the following elements of the vehicle simulation model are presented (Fig. 1):

- moving through the virtual environment [3],
- collisions with other objects,
- shooting [4],
- transporting people,
- the driver's, shooter's and crew member's perspective.

R. Wantoch-Rekowski (✉)
Faculty of Cybernetics, Military University of Technology, Warsaw, Poland
e-mail: rekowski@wat.edu.pl

K. Szumiec
fragOut Studio, Lodz, Poland
e-mail: k.szumiec@fragoutstudio.com

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_13

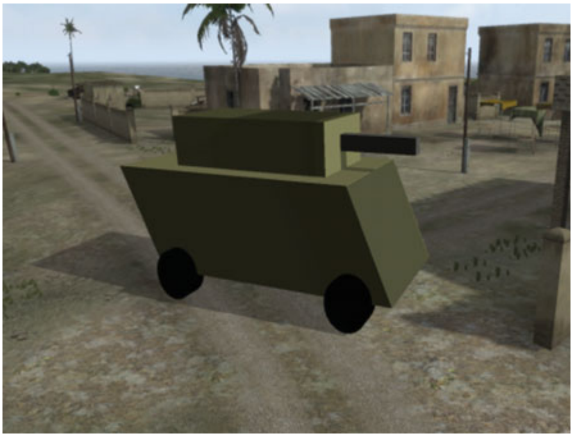


Fig. 1 Exemplary vehicle model

All elements of the model as well as additional files were created using tools supplied by the manufacturer of VBS2 in the form of the VBS2 Development Suite.

An example has been prepared on the assumption that the entire project will be placed in a directory on a virtual hard disk P:\AMV\MyVehicle, which is created normally when you install the VBS2 environment. In the project directory, subdirectories date and source should be created. Textures in .tga and .paa formats, .rvmat materials and scripts should be placed in the subdirectory date. More subfolders can also be created to group more data. The source subdirectory should contain source files, such as documentations, models and textures in formats other than those used by the VBS2 environment. The source subdirectory contains aiding materials that will not be a part of the target project (not binarized).

2 Creating the Basic Shape of the Vehicle

The vehicle model is created in P3D format using the program Oxygen (the result is saved in a file named myvehicle.p3d). In further parts the elements will be presented from the left view window (Fig. 2).

To create a vehicle model (p3d), the following steps should be carried out:

1. **Creating the vehicle body**. In the `Create` menu select the `Box` option and enter parameters as shown in Fig. 3. In the figure one unit corresponds to 1 m.

   Using the MOVE tool the created object should be moved 1.5 units upward (see Fig. 4), (Fig. 5).
2. **Creating wheels**. Wheels are created in the form of cylinders. In the `Create` menu the `Cylinder` function should be selected and parameters should be set as shown in Fig. 6.

   Analogously, just like the basic body, an operation of moving the cylinder, using the MOVE option, should be conducted. The cylinder (wheels) should be

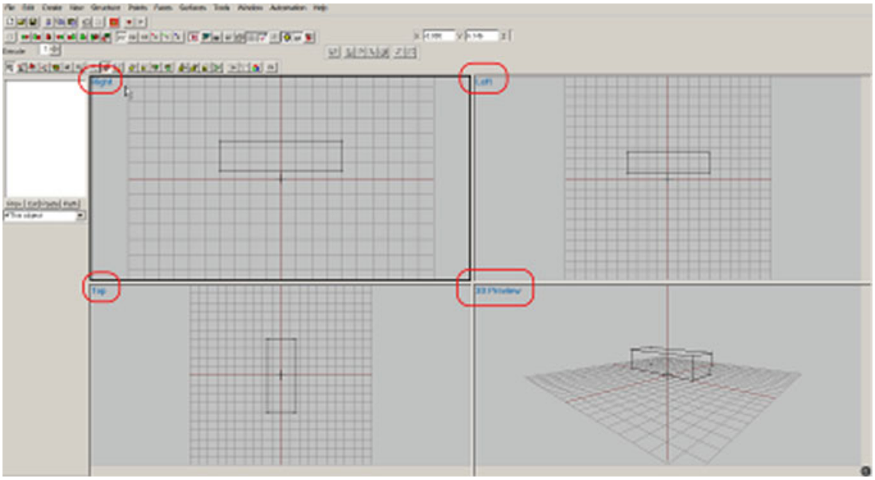
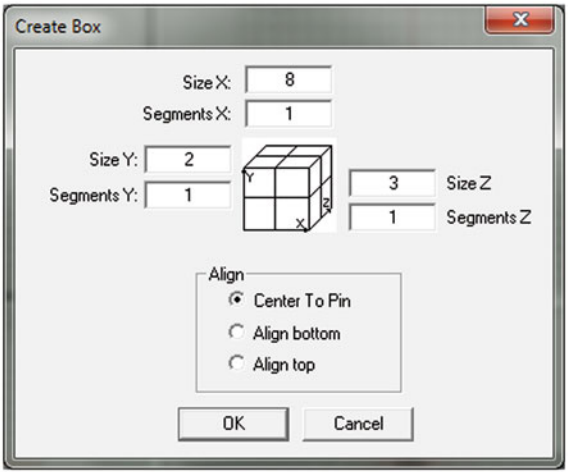


Fig. 2 Basic view of the program oxygen (*views right*, *left*, *top* and 3D preview)

Fig. 3 Inserting a BOX-type object

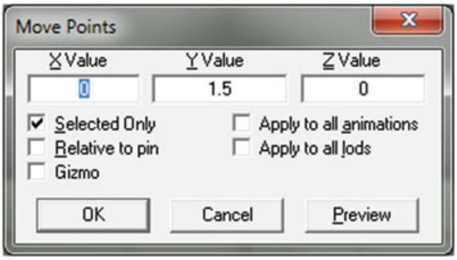


Create Box

Size X: 8
Segments X: 1
Size Y: 2
Segments Y: 1
Size Z: 3
Segments Z: 1

Align
● Center To Pin
○ Align bottom
○ Align top

Fig. 4 Moving the BOX-type object



Move Points

X Value	Y Value	Z Value
0	1.5	0

☑ Selected Only
☐ Relative to pin
☐ Gizmo
☐ Apply to all animations
☐ Apply to all lods

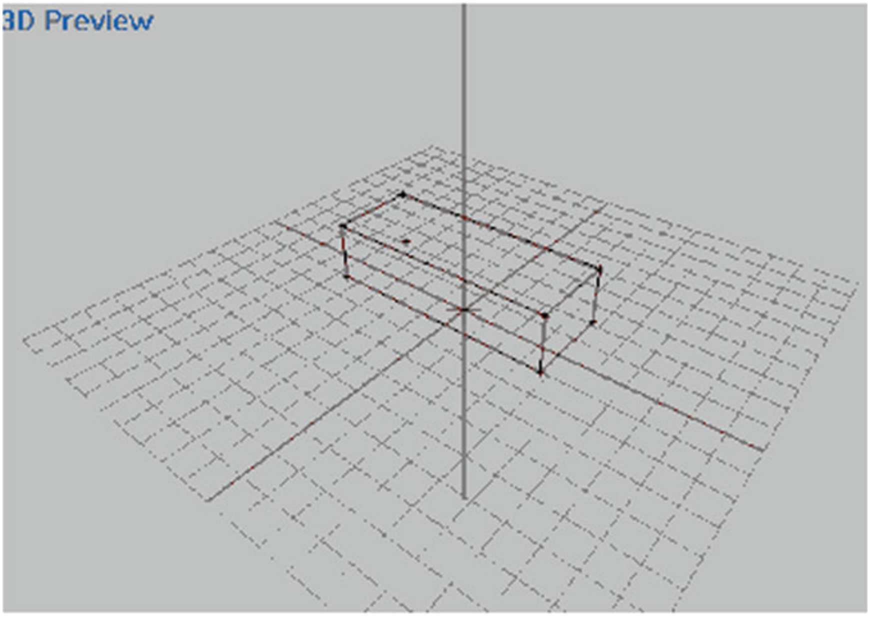
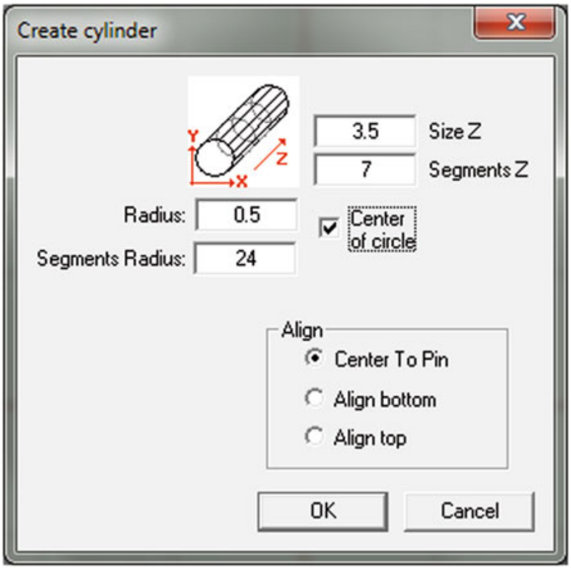


Fig. 5 3D preview



3.5 Size Z

7 Segments Z

Radius: 0.5

Segments Radius: 24

☑ Center of circle

Align

● Center To Pin

○ Align bottom

○ Align top

Fig. 6 View of the parameters of the cylinder (wheels)

moved upward (Y axis) by 0.5 units (0.5 m). Moving the cylinder enables positioning the designed wheels, so that it touches the plane as well as the "suspension" of the body.

At this stage, the inserted cylinder (see Fig. 7) actually comprises seven parts (single wheels), of which only the external will be the design of the actual wheels. Therefore, unnecessary items should also be removed. To do this, the Top view should be switched to, and then unnecessary points selected (Fig. 8). The indicated points should be deleted with the `Delete` key. The appearance of the wheels after deleting the unnecessary elements from the cylinder is presented in Fig. 9.

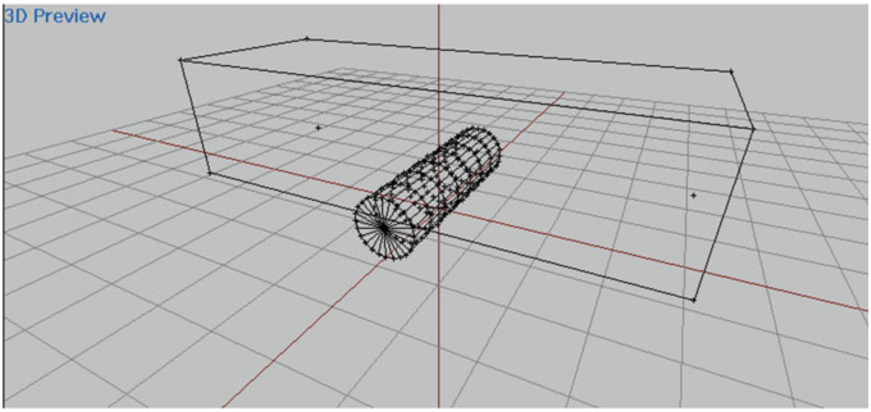


Fig. 7 View of the cylinder lifted up to the designed body

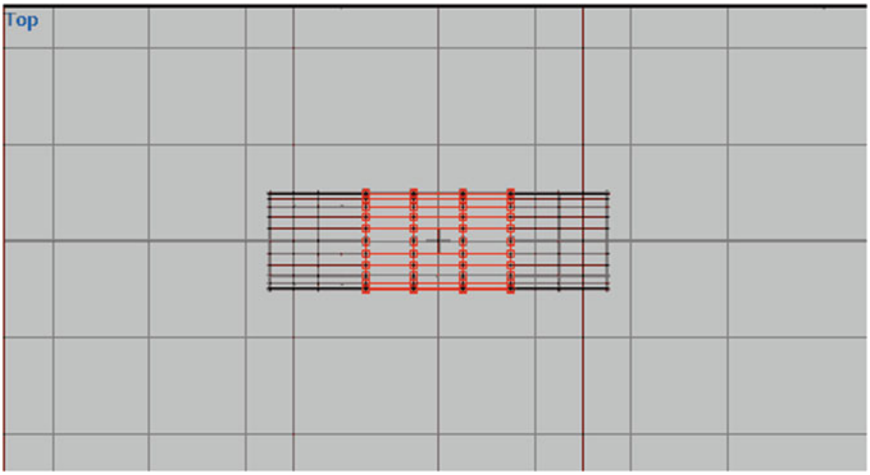


Fig. 8 View of the cylinder with *marked points* for deletion

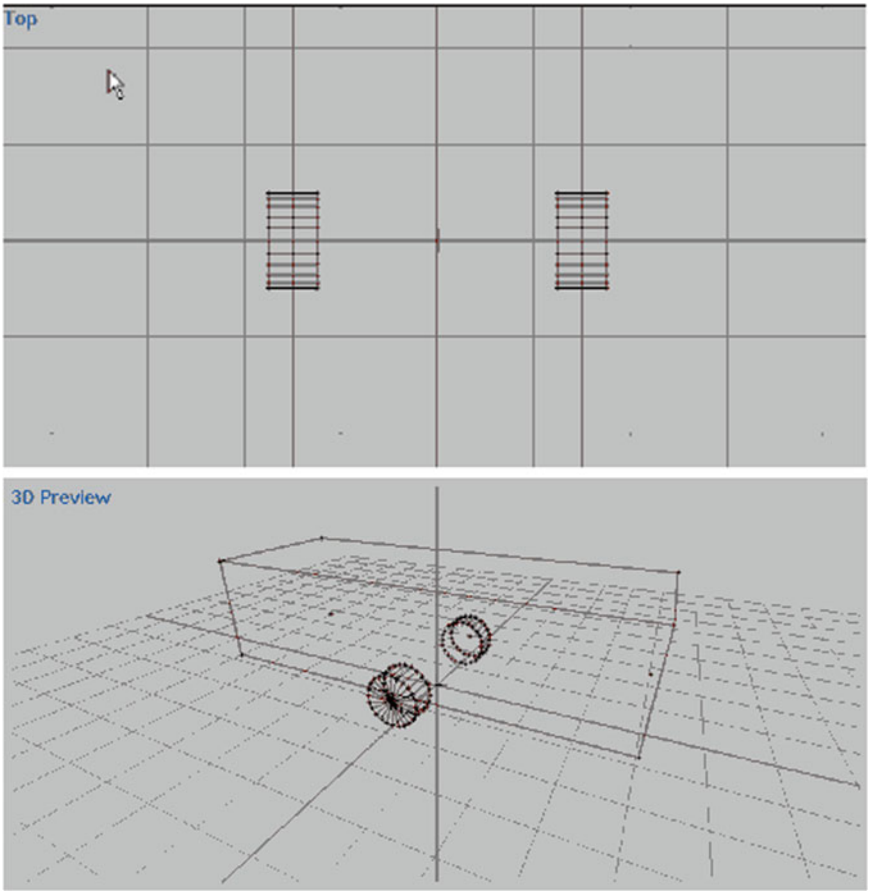
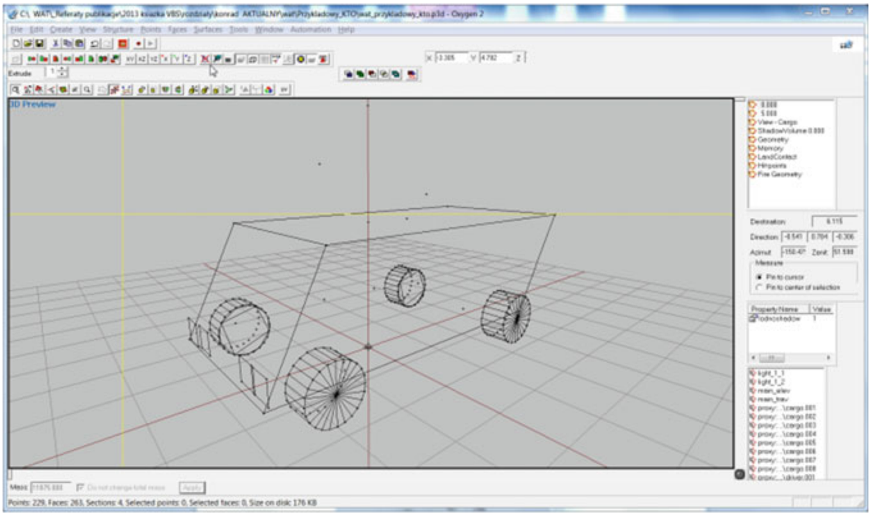


Fig. 9 View of first pair of wheels (*top* and 3D view)

Because the elements on the inner side were deleted it caused that the surfaces remained opened. This can be resolved by closing them using an option of adding a surface. Both elements of the wheels should be selected and then use the menu option `Structure/Topology/Close`.

To create the remaining wheels, the created pair needs to be copied and moved in the Z plane using the `MOVE points` option (Fig. 10).

3. **Creating the turret**. The next step will be to create a turret and gun. Both the turret and gun were created with the BOX object and moved to their proper locations (Fig. 11).
4. **Creating lights**. The last element of the created p3d vehicle model is the lights. Thanks to the adding lights a glow effect of turned on headlights will be achieved. The lights are creating in the Front view using the function `Create/Plane`. The created elements should be placed in an appropriate location using the `MOVE` function (Fig. 12).


C:\ WAT_Referaty publikacje\2013 ksiazka VBS\rozdzialy\konrad AKTUALNY\wat\Przykladowy_KTO\wat_przykladowy_kto.p3d - Oxygen 2

Points: 229, Faces: 263, Sections: 4, Selected points: 0, Selected faces: 0, Size on disk: 176 KB

Fig. 10 View of the body with the created wheels

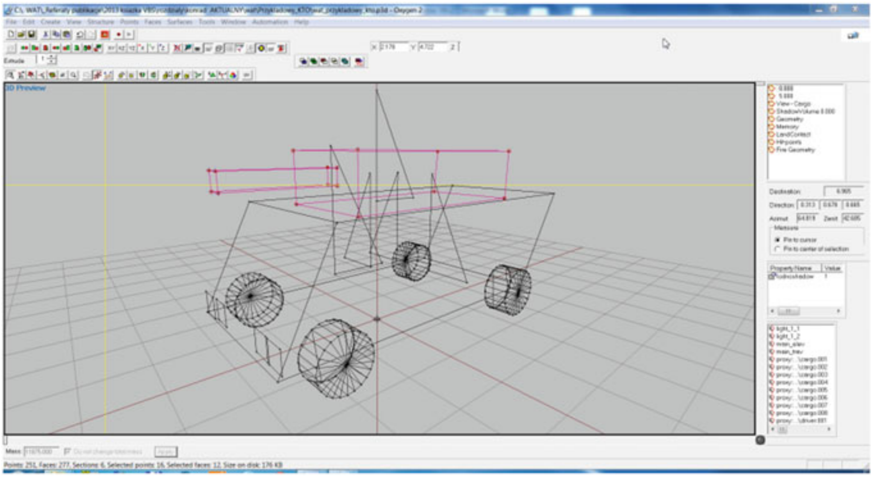
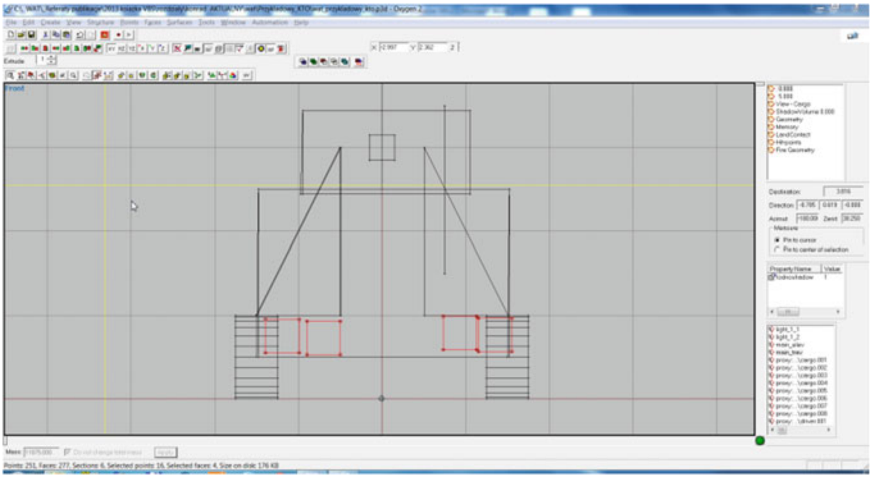

Points: 251, Faces: 277, Sections: 6, Selected points: 16, Selected faces: 12, Size on disk: 176 KB

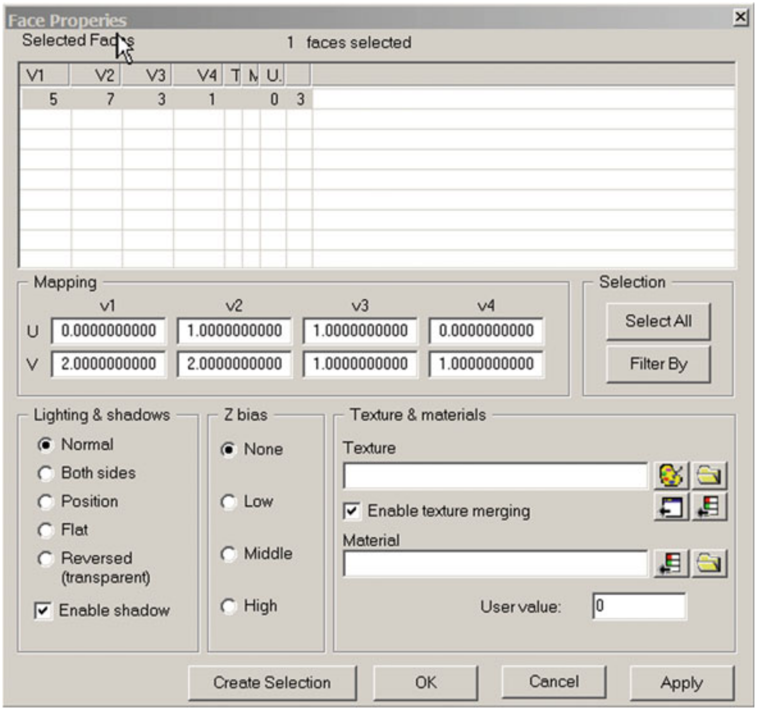
Fig. 11 View of the body with created wheels

Finally, different colours and textures can be given to surfaces of individual components. To do this, select the surfaces for which you want to set the color or texture. Then go to option `Faces/Face properties (E)`. The properties window appears, as shown in Fig. 13.



Points: 251, Faces: 277, Sections: 6, Selected points: 16, Selected faces: 4, Size on disk: 176 KB

Fig. 12 View of the body with the created headlights



Face Properies

Selected Faces 1 faces selected

V1	V2	V3	V4	T	N	U.	
5	7	3	1			0	3

Mapping

	v1	v2	v3	v4
U	0.0000000000	1.0000000000	1.0000000000	0.0000000000
V	2.0000000000	2.0000000000	1.0000000000	1.0000000000

Lighting & shadows
- ● Normal
- ○ Both sides
- ○ Position
- ○ Flat
- ○ Reversed (transparent)
- ☑ Enable shadow

Z bias
- ● None
- ○ Low
- ○ Middle
- ○ High

Texture & materials

Texture:

☑ Enable texture merging

Material:

User value: 0

Fig. 13 Surface properties window

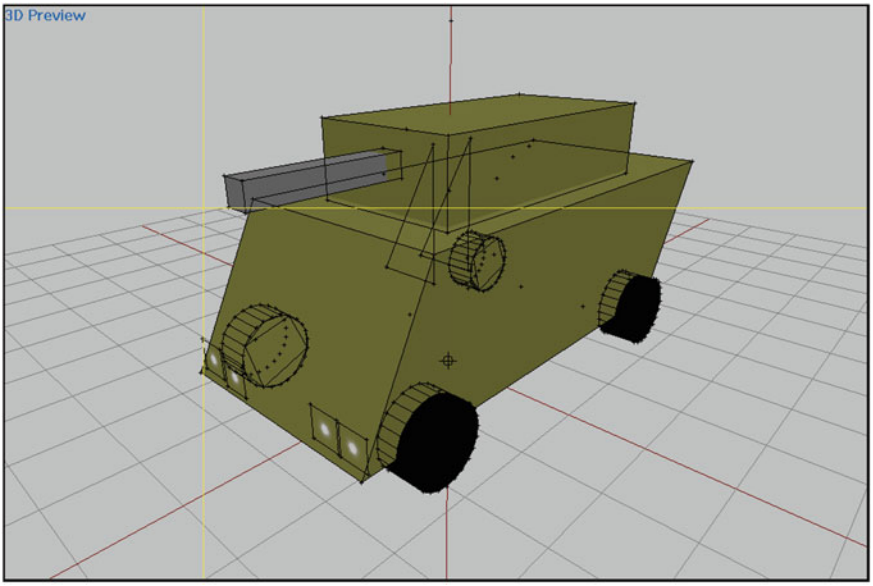


Fig. 14 Model with colours selected for the body, turret and gun

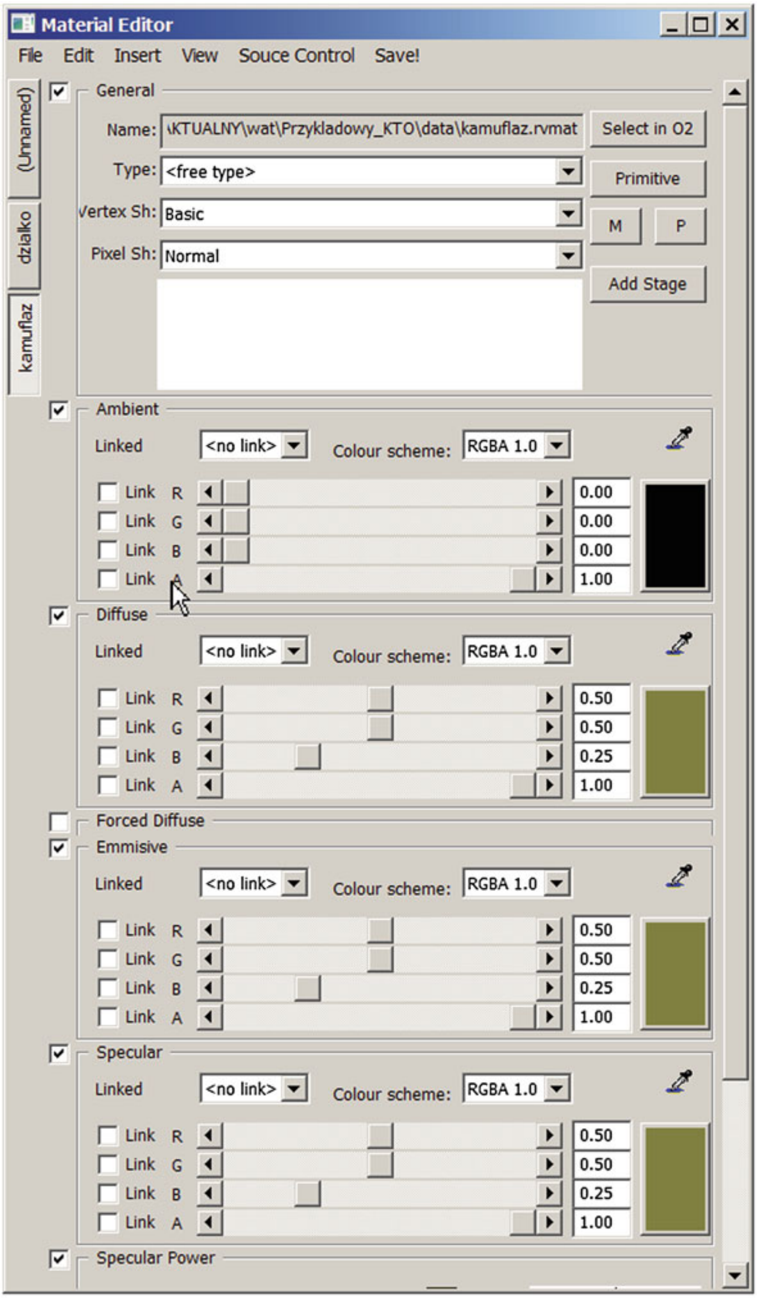
The texture file name in .tga or .paa format should be inserted in the Texture field or select a surface colour (see Fig. 14).

When using the program P:\tools\MatEditor.exe, RVMAT (Real Virtuality Material) files should be created in the date subdirectory. They contain saved configuration files with definitions of light behavior on surfaces: reflection, glow, used shading algorithms, etc. The path to the created RVMAT file is defined in the text box below the definition of the source of surface colour (see Fig. 15).

Presented below is the content of created files with RVMAT definitions for the body and turret as well as the gun and wheels (tyres).

Camouflage.rvmat file contents:

```
ambient[] = {0,0,0,1};
diffuse[] = {0.5,0.5,0.25,1};
forcedDiffuse[] = {0,0,0,0};
emmisive[] = {0.5,0.5,0.25,1};
specular[] = {0.5,0.5,0.25,1};
specularPower = 60;
PixelShaderID = “Normal”;
VertexShaderID = “Basic”;
```



☑ General

Name: \KTUALNY\wat\Przykladowy_KTO\data\kamuflaz.rvmat

Type: <free type>

Vertex Sh: Basic

Pixel Sh: Normal

☑ Ambient

Linked <no link> Colour scheme: RGBA 1.0

□ Link R 0.00
□ Link G 0.00
□ Link B 0.00
□ Link A 1.00

☑ Diffuse

Linked <no link> Colour scheme: RGBA 1.0

□ Link R 0.50
□ Link G 0.50
□ Link B 0.25
□ Link A 1.00

□ Forced Diffuse

☑ Emmisive

Linked <no link> Colour scheme: RGBA 1.0

□ Link R 0.50
□ Link G 0.50
□ Link B 0.25
□ Link A 1.00

☑ Specular

Linked <no link> Colour scheme: RGBA 1.0

□ Link R 0.50
□ Link G 0.50
□ Link B 0.25
□ Link A 1.00

☑ Specular Power

Fig. 15 Main window of the MatEditor program

Gun.rvmat file contents:

```
ambient[] = {0,0,0,1};
diffuse[] = {0.5,0.5,0.5,1};
forcedDiffuse[] = {0,0,0,0};
emmisive[] = {0.5,0.5,0.5,1};
specular[] = {0.5,0.5,0.5,1};
specularPower = 45;
PixelShaderID = “Normal”;
VertexShaderID = “Basic”;
```

Tyres.rvmat file contents:

```
ambient[] = {0,0,0,1};
diffuse[] = {0,0,0,1};
forcedDiffuse[] = {0,0,0,0};
emmisive[] = {0,0,0,1};
specular[] = {0.5,0.5,0.5,1};
specularPower = 90;
PixelShaderID = “Normal”;
VertexShaderID = “Basic”;
```

For the “light”-type surface image files are provided by the manufacturer vbs2 \data\light_front_ca.tga as well as the rvmat file vbs2\data\light_front.rvmat.

Light_front.rvmat file contents:

```
ambient[] = {1.00,1.00,1.00,1.00};
diffuse[] = {0.50,0.50,0.50,0.00};
forcedDiffuse[] = {0.50,0.50,0.50,1.00};
emmisive[] = {0.50,0.50,0.50,0.10};
specular[] = {0.00,0.00,0.00,0.00};
specularPower = 0.00;
renderFlags[] =
{
  “NoZWrite”,
  “AddBlend”
};
PixelShaderID = “Normal”;
VertexShaderID = “Basic”;
class StageTI
{
  texture = “#(argb,8,8,3)color(0,0,0,0,co)”;
};
```

3 Creating Active Selections

One of the basic properties of the vehicle models (and other objects as well) in VBS2 is the ability to define the animation [5]. In the case of a vehicle designed using defined animations of rotating wheels, turret and gun. It is divided model elements such as wheels, turret and gun that are called selections.

To create a selection, the selection window must be enabled in the program Oxygen from the Window/Named selections menu by right-clicking, in which, the following selections must be created named:

- `wheel_1_1`, `wheel_4_1` for wheels on the left side, from front to back left;
- `wheel_1_2`, `wheel_4_2` for wheels on the right side, from front to back;
- `main_trav` for the turret;
- `main_elev` for the gun;
- `light_1_1` for the left headlight;
- `light_1_2` for the right headlight.

Connections between separated elements of the model and created names of selections should be created afterwards. To do this, select all relevant points of the given selection, then right-click on the appropriate selection on the selection list, and choose the Redefine option (see Fig. 16).

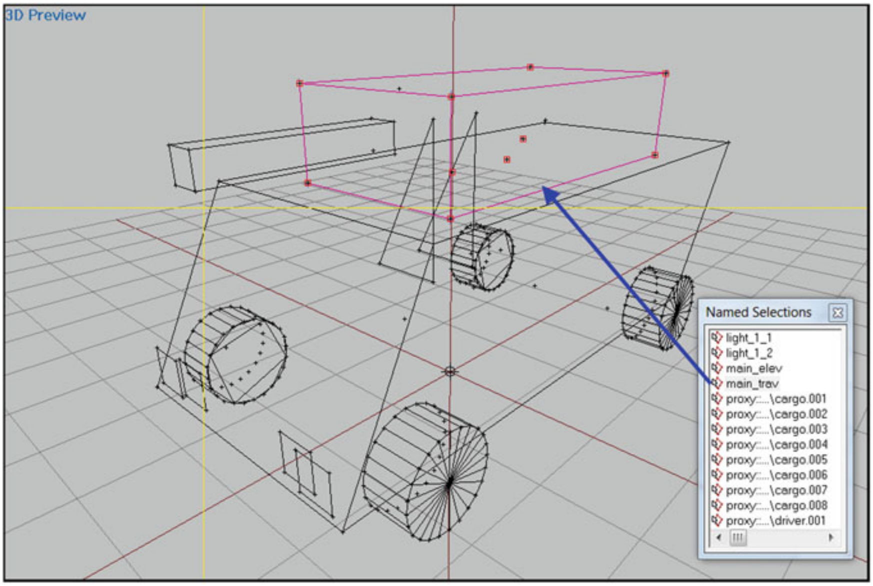


Fig. 16 View of the selection window

4 Creating Technical Layers

Object models made for the needs of the simulation in VBS2 should have the highlighted layers performing respective roles defined. The most important are:

- object visualization [6],
- counting contact and collisions with the terrain, other objects as well as bullets,
- reference points for animation,
- sensitive firing points,
- lighting up the shadow.

Creating a new layer takes place in the Lods (Level of details) window, which can be opened selecting the Window/Lods menu, then right-clicking on its free space and selecting the New option. By right-clicking on the created layer (probably 1000), select Properties. In the appearing list all possible types of layers are available (see Fig. 17).

The automatically created default layer is a layer called 0.000. The layer contains the object view with the highest accuracy used to depict the object in the simulation, as it is very close to the point from which the observer is standing.

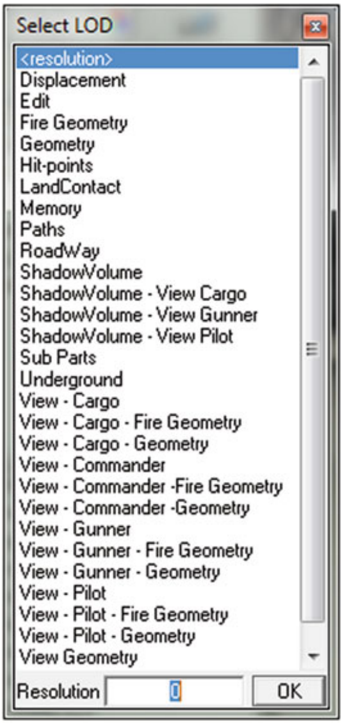


Fig. 17 P3d object layer types

Further in the chapter the method of creating the following layers will be presented:

- `5.0` (simplified image layer, displayed when the vehicle is far from the current view);
- `Shadow Volume`, `Resolution 0` (layer defining elements of the object generating shadow);
- `Geometry` (layer defining elements of the vehicle colliding with other objects);
- `View—Cargo` (layer displayed when selecting internal view of the vehicle);
- `Fire Geometry` (layer defining elements sensitive to fired bullets);
- `Land Contact` (layer defining points that contact the ground);
- `Hit-points` (layer containing sensitive firing points);
- `Memory` (layer containing reference points to other elements).

4.1 Layer 5.000

Click the right mouse button on layer `0.000` and by using the `Duplicate` option copy its contents. This will create a layer with the name of the `1.000`. Then, change the `resolution` level to `5` in the Properties window. Since the created layer must be a simplified model, it needs to be made simpler. Due to the fact that the elements containing the most points and planes are wheels, these elements will

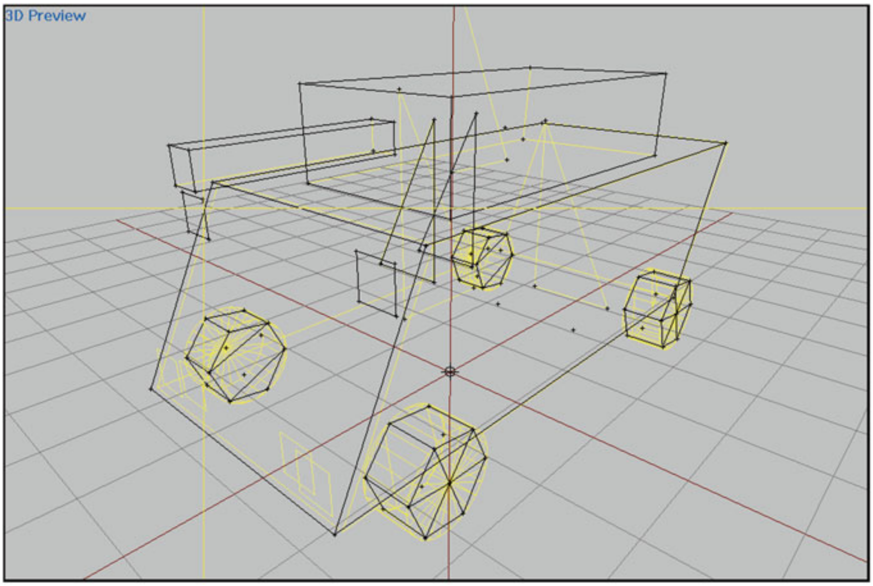


Fig. 18 LOD 5.000 layer

be made simple. For each wheel (after its selection) the `PointsMerge near` function with parameter `0.3` should be called to reduce detail. Figure 18 presents a view of a layer `5.000` with a simplified model of wheels. In the view of the layer, layer `0.000` is always shown in yellow as a reference point.

The vehicle model should include a few image layers to optimize the performance of generating images of objects (e.g. `1.000`, `2.000`, `5.000`). The vehicle model should include a few image layers to optimize the performance of generating the object images. The detail of each consecutive layer should be significantly smaller than the previous, e.g. if layer `1.000` has 12,000 points, then layer `2.000` should have no more than 6000 points. The final layer of graphical basic version of the vehicle should have no more than 500 points. The layer number is not important; the graphics engine only takes the order and degree of layer detail into account.

4.2 Shadow Volume Layer

To create a layer `Shadow Value 0.000` used to generate shadows by the vehicle, the `5.000` layer should be copied and change its type to `Shadow Volume` in the `Properties` window. Afterwards the entire contents of the layer should be selected and the options Structure/Triangulate concave selected in the menu and then change the edges to sharp using `Surface/Sharp edge`.

4.3 Geometry Layer

To create the `Geometry` layer, layer `5.000` should be copied and its type changed in the `Properties` window to `Geometry`. Afterwards, the entire contents of the layer should selected and the `Structure/Triangulate convex` option selected in the menu or `Structure/Topology/Find components` to generate points, which will be able to define the mass. Subsequently, a total mass should be assigned to each of the points. In the `Window/Mass` window, for example, `20000` should be inserted (Fig. 19).

4.4 View—Cargo Layer

In the `View Cargo` layer even very complex vehicle interiors can be modeled. In the presented example this layer will be prepared on the basis of layer `0.000` with deleted wheels. Due to the change in viewpoint of this layer, an operation of reversing, the so-called normal surfaces, from which images are generated, should be done. The `Faces/Reverce` option should be selected from the menu. In more

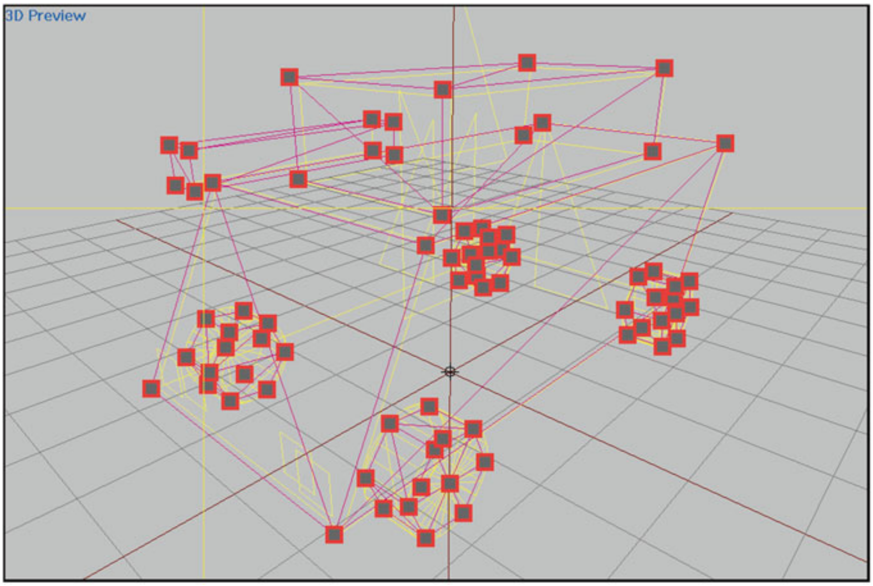


Fig. 19 Geometry layer

complex examples a detailed internal construction of stations with equipment and devices can be created.

4.5 Fire Geometry Layer

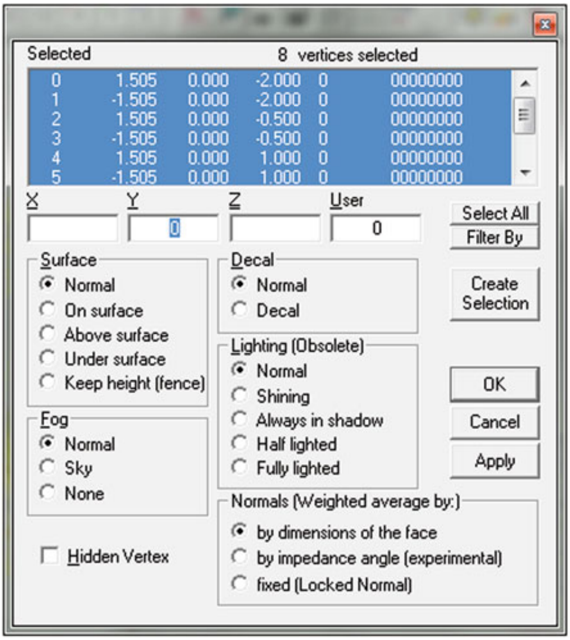
This layer should be created from a copy of the `Geometry` layer.

4.6 Land Contact Layer

A new `Land Contact` layer should be created, and then copy all wheels from layer `0.000`. Afterwards, for each wheel an operation of connecting all points should be done using the `Points/Merge` function. Each wheel will be depicted as a single point. Subsequently, for each point using the `Points/Properties` menu, the Y coordinate should be set to `0` (Fig. 20).

At the end of creating the layer the name of the selection should be changed by adding "_damper" at the end of every selection name (Fig. 21).

An additional animation configuration of contact points should be conducted in the `model.cfg` file, so that crossing bridges and overcoming other structures would be possible.



Selected 8 vertices selected

0	1.505	0.000	-2.000	0	00000000
1	-1.505	0.000	-2.000	0	00000000
2	1.505	0.000	-0.500	0	00000000
3	-1.505	0.000	-0.500	0	00000000
4	1.505	0.000	1.000	0	00000000
5	-1.505	0.000	1.000	0	00000000

X:
Y: 0
Z:
User: 0

Select All
Filter By
Create Selection

Surface
● Normal
○ On surface
○ Above surface
○ Under surface
○ Keep height (fence)

Fog
● Normal
○ Sky
○ None

□ Hidden Vertex

Decal
● Normal
○ Decal

Lighting (Obsolete)
● Normal
○ Shining
○ Always in shadow
○ Half lighted
○ Fully lighted

Normals (Weighted average by:)
● by dimensions of the face
○ by impedance angle (experimental)
○ fixed (Locked Normal)

OK
Cancel
Apply

Fig. 20 Properties of points

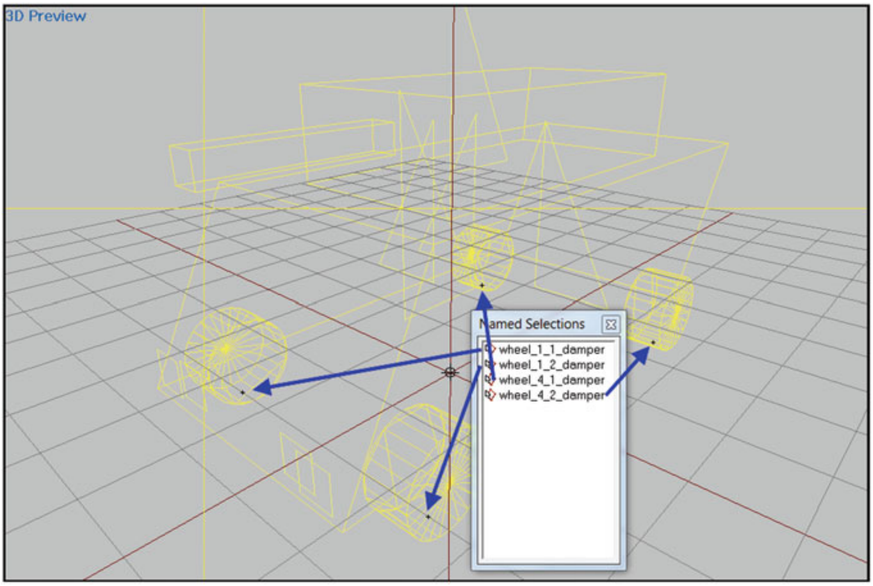


Fig. 21 Land contact layer

4.7 Hit-Points Layer

In the Hit-points layer points should be marked in areas sensitive to firing. The following sensitive points to firing were defined (see Fig. 22):

- engine (engine),
- wheels (wheel_x_y),
- fuel tank (fueltank),
- turret (main_trav),
- headlights (light_x_y).

4.8 Memory Layer

The Memory layer contains the definition of different types of points, which are in close relation with the operation of the object in the simulation. First of all, two points symbolizing the ends of the axes for each of the animated elements are added in the Memory layer. Wheels (rotation), turret (rotation) and gun (raising and dropping—see Fig. 23) are animated in the presented example. Points from layer 0.000 can be pasted and merged, so that the axes will be in their target positions. Afterwards, the created points (a pair of points creating an axis) should be added to the selection with axis at the end of the name (e.g. main_trav_axis).

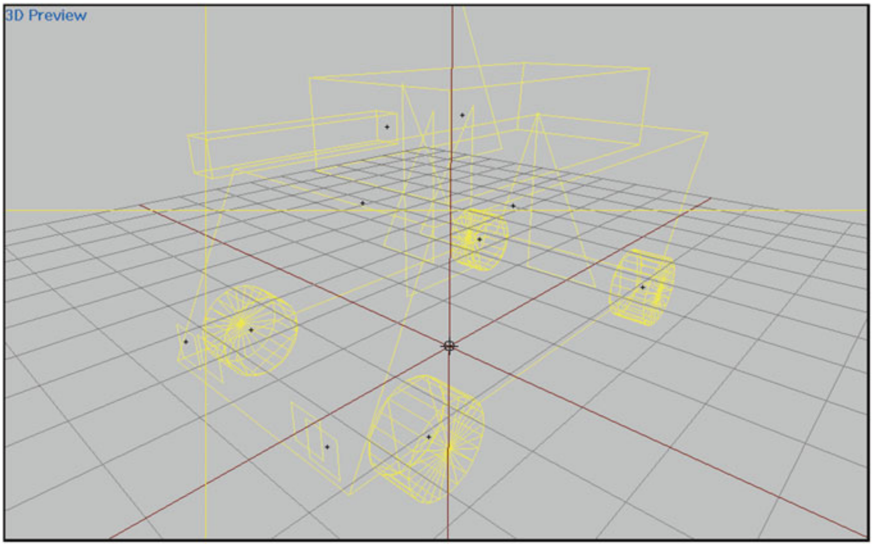


Fig. 22 Hit-points layer

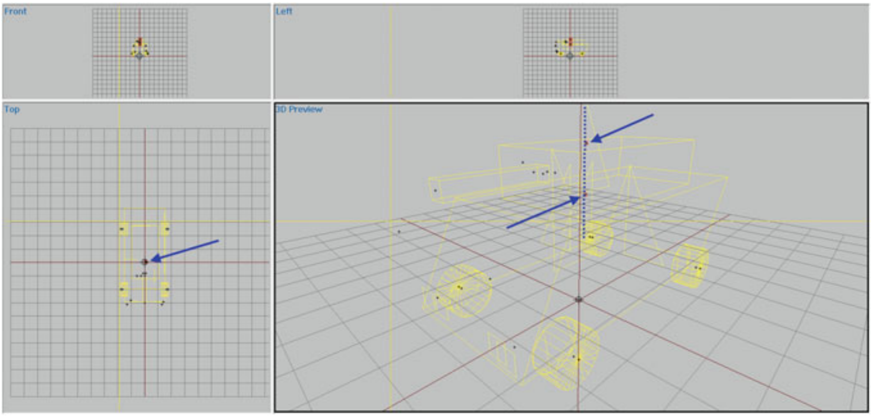
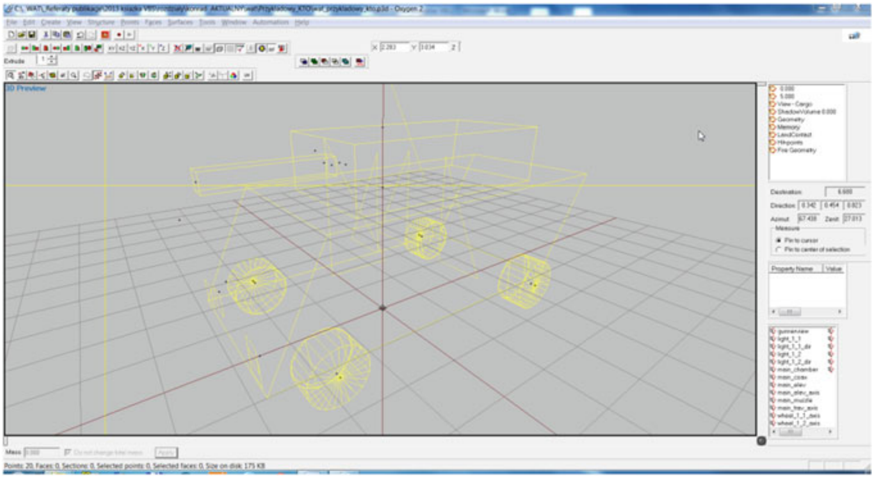


Fig. 23 Memory layer—definition of the rotational axis



Destination 6.600
Direction 0.342 0.454 0.823
Azimut 67.438 Zenit 27.013
Measure
● Pin to cursor
○ Pin to center of selection

Fig. 24 Memory layer

Additionally, the following points should be defined (see Fig. 24):

- view position from the driver's perspective (`driverview`);
- view position from the gunner's perspective (`gunnerview` in `main_elev`);
- initial and end position of the gun muzzle (`main_chamber` and `main_muzzle`, both in `main_elev`);
- coupled machine-gun position (`main_coax` in `main_elev`);
- headlight position (`light_X_Y`).

5 References to the Crew and Landing Sites

One of the elements of the construction of the vehicle model is determining the capability of placing people inside the vehicle. To do this, one should define special elements (references), so-called proxy. References are symbolized by the plain comprising three points (Fig. 25).

To insert the driver's proxy, the function `Create → Proxy` should be used and an exemplary `.p3d` model should be sought for in the `P:\vbs2\temp`

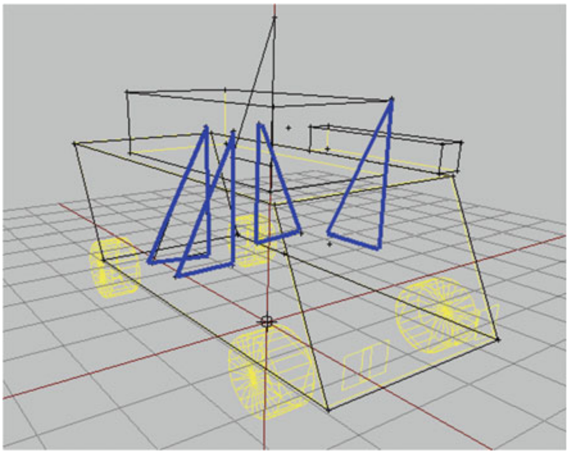


Fig. 25 Defining proxy positions

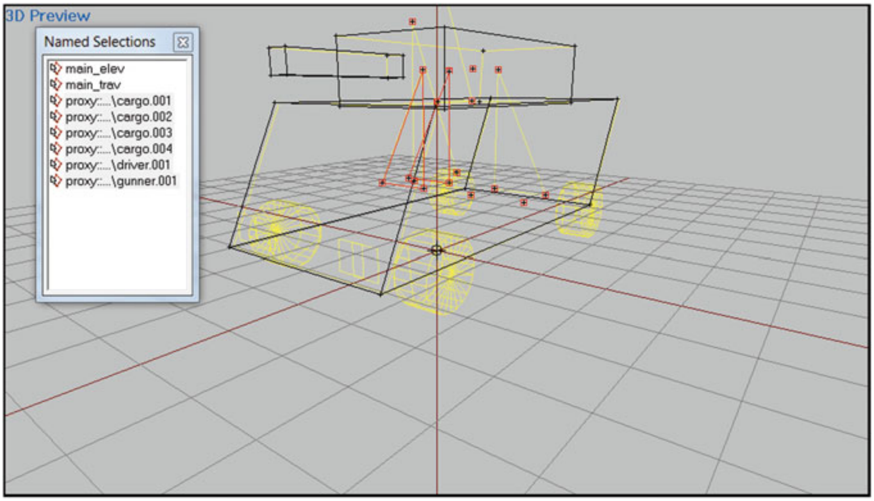


Fig. 26 View of all defined and required proxy positions

`\crew_previews\` directory presenting the crew member. Diver postures are in the file named `driver.p3d`, the gunner posture—`gunner.p3d`, and the proper postures for the landing troops—`cargo.p3d`.

The proxy of the crew members and landing troops should be copied and pasted into:

- all image layers (e.g. `1.000`, `5.000` and `View cargo`);
- all shadow layers (`Shadow Volume`);
- bullet collision layer (`Fire Geometry`) (Fig. 26).

Make sure to define the appropriate class animation in file `config.cpp`.

6 Animations

An important stage of constructing a vehicle model is defining the animation. The file model.cfg should be created in the project directory and an adequate configuration should be introduced in accordance with principles of creating configuration files.

Contents of the `model.cfg` file:

```
#include “CfgSkeletons.hpp”
#include “CfgModels.hpp”
```

During the process of binarization of the project the `model.cfg` files are loaded in order from each of the directories in the project hierarchy. In this case the files will be `P:\model.cfg`, `P:\AMV\model.cfg` and `P:\AMV\MyVehicle\model.cfg`. It is important that the framework of the classes and model be visible during binarization of the model, otherwise the animations will not work.

Classes embedded in `CfgSkeletons` contain definitions of bone animation and their hierarchy. The definition of bones is composed of the selection name and its parent, e.g. the parent of the gun is the turret bone—“`main_elev`”`,`“`main_trav`”, and they can be found in the `skeletonBones[]` table with an even number of elements. If the new bone does not have a parent then an empty series of characters should be inserted in its place—“`main_trav`”`,`“”.

With the help of the `skeletonInherit` attribute inheriting elements of the bone table can be set from another skeleton class. In the discussed example, there is a class that defines the frame of the basic version without the turret and gun.

Contents of the `CfgSkeletons.hpp` file:

```
class CfgSkeletons {
  class Default {
    skeletonInherit = "";
    skeletonBones[] = {};
    isDiscrete = 1;
  };
  class Myvehicle_Unarmed: Default {
    skeletonBones[] = {
      "wheel_1_1_damper","",
      "wheel_4_1_damper","",
      "wheel_1_2_damper","",
      "wheel_4_2_damper","",
      "wheel_1_1","wheel_1_1_damper",
      "wheel_4_1","wheel_4_1_damper",
      "wheel_1_2","wheel_1_2_damper",
      "wheel_4_2","wheel_4_2_damper"
    };
  };
Class Myvehicle_Armed: Myvehicle_Unarmed {
skeletonInherit = "Myvehicle_Unarmed";
// inhereting bones
  skeletonBones[] = {
    "main_trav","", // turret
    "main_elev","main_trav" // gun
  };
};
};
```

Classes embedded in CfgModels contain definitions of the selected skeleton class, active texturing sections and Animations class containing animation classes for bones defined in the skeleton. The name should correspond to the.p3d model file name. It is the Myvehicle class in the discussed project.

The sections[] table contains selection names, which should be able to be retexturized or hidden. They are, for example, the coaxial gun muzzle flame, headlights, damaged sections and changes in camouflage.

Classes embedded in the Animations class contain the following definitions:

- type—animation type: rotation, displacement, concealment,
- source—data source: driving wheel, wheels, turret rotation and many others, including own definitions,
- minValue and maxValue—input range: minimum and maximum source value,
- sourceAddress—behaviour out of range: stopping, mirroring, reversing,
- minPhase, maxPhase, angle0, angle1, hideValue—animation output range: moving along the axis, rotation, hide value.

For model elements that are to be defined by a special animation, which is not defined in the VBS2 graphics engine simulator, the vehicle must have its own classes embedded in a special class AnimationSources.

7 Config.cpp File

The last stage of building the vehicle model is creating a project configuration file config.cpp in the directory. Elements in the config.cpp file enable turning on a defined object in the VBS2 simulation environment. The file contains the following information (definitions):

- required additions ensuring module loading in the correct order,
- physical vehicle properties,
- weapon and cartridge clips in the turret,
- possibility of transporting marines,
- driver periscope and gunner optics perspective.

Firstly, the CfgPatches class should be created. It is an identifier of the newly created addition and information for the simulation engine, which modules should be loaded first. In this case the vbs2_vehicles_Land_Wheeled module is needed, which contains the definition used in the base class of the project. In addition, newly created classes, embedded in CfgWeapons and CfgVehicles as well as the required environment version (it can be 0), should be added in the respective tables.

Contents of the config.cpp file:

```
class CfgPatches {
  class myvehicle {
    weapons[] = {};
    units[] = {"Myvehicle"};
    requiredAddons[] =
{"vbs2_vehicles_Land_Wheeled"};
    requiredVersion = 0;
  };
};

#include "CfgVehicleClasses.hpp"
#include "CfgVehicles.hpp"
```

To create a new vehicle category, a class should be created embedded in the CfgVehicleClasses class. Its only attribute will be its name—displayName.

Contents of the cfgvehicleclasses.hpp file:

```
class CfgVehicleClasses {

   class AMV_Category {

      displayName = “AMV Category”;

   };

};
```

The following step will be to create the CfgVehicles class. The classes embedded in it, which can be placed in the simulator, or class templates of such objects. Object class templates differ from object classes due to the scope attribute—in the template it is set to 0, in the hidden class for the editor, but inserted by using a script `scope = 1`, and in the scenarios editor in the visible class `scope = 2`.

Basic vehicle class attributes:

- `displayName`: object name visible in the editor,
- `vehicleClass`: object class in the editor,
- `model`: the path to the model on disk P:,
- `hasGunner`, `hasCommander`, `hasDriver`: `Player as…` option availability in the editor,
- `crew`: vehicle crew default class,
- `armor`, `maxSpeed`, `canFloat`, `turnCoef`, `terrainCoef`: physical vehicle parameters,
- `memoryPoint…`, `selection…`: reference points, active selections,
- `driverAction`, `cargoAction[]`,…`Action`: animations for individual roles and actions [7].

The vehicle class contains the `Turrets` class with classes of turrets, e.g. `MainTurret`. Every turret can posses it own `Turrets` class, which enables adding additional stations, such as the commander behind the anti-aircraft.

Basic attributes of the vehicle turret class:

- `body`, `gun`, `animationSource…`: names of classes and sources of animation in the `model.cfg` file,
- `gunBeg`, `gunEnd`, `memoryPointGun`: reference points of the armament,
- `selectionFireAnim`: coaxial gun muzzle flame,
- `memoryPointGunnerOptics`: reference points of the optics position,
- `gunnerAction`, `gunnerInAction`: station crew animation,
- `maxHorizontalRotSpeed`, `maxVerticalRotSpeed`: rotational speed (x°/45 s),
- `weapons[]`, `magazines[]`: tables of weapons and ammunition classes.

8 Conclusion

Building models of vehicles for a virtual simulation environment requires the consideration of many aspects that enable its proper functioning during simulation. The object graphical form is only one of many aspects to be defined in the model. It is important to determine the method of action and the behaviour of the fundamental principles of physics in interactions with other simulation objects [8].

References

1. Wantoch-Rekowski, R. (scientific editor): The programmable VBS2 virtual simulation environment (2013). ISBN:978-83-01-17323-4. (PWN)
2. Website:http://www.bisimulations.com
3. Koszela, J., Drozdowski, T., Wantoch-Rekowski, R.: Advanced methods of preparing field data for the VBS2 simulator tactical level. Institute of Computer Science, Faculty of Cybernetics, Military University of Technology, Warsaw **11**, 1508–4183 (2013)
4. Koźlak, M., Kurzeja, A., Nawrat, A.: Virtual reality technology for military and industry training programs. In: Vision Based Systems for UAV Applications. http://link.springer.com/chapter/10.1007/978-3-319-00369-6_21
5. Koszela, J., Wantoch-Rekowski, R.: The concept of a simulator for training drivers of PSP combat vehicles in terms of tasks carried out in the framework of the national rescue and fire-fighting system. Saf. Fire Tech. 71–81 (2012). ISSN:1895-8443
6. Koszela, J., Drozdowski, T., Wantoch-Rekowski, R.: Preparing field data for multi resolution simulation. High Speed Track Veh. **31**(3), 109–118 (2012). ISSN:0860-8369
7. Koszela, J., Wróblewski, P., Szymańska, A., Wantoch-Rekowski, R.: Design and implementation of mechanisms of artificial intelligence in the VBS2 simulation environment. SPG **31**(3), 119–132 (2012). ISSN:0860-8369
8. Koszela, J., Wantoch-Rekowski, R.: The use of simulators for training in crisis situations. Saf. Fire Tech. 113–120 (2013). ISSN:1895-8443

Semi-active Suspension System for 2S1 Tracked Platform in Drive Comfort Improvement Application

Tomasz Nabagło, Andrzej Jurkiewicz and Janusz Kowal

Abstract In the article, a new solution of a semi-active suspension system is presented. It is based on a sky-hook strategy model. This solution in 2S1 tracked platform is applied to improve driving comfort as very important factor for the vehicle crew efficiency. The solution is applied in two versions of the 2S1 vehicle suspension model. First one is a basic model. This suspension is based on existing construction of the 2S1 platform suspension. It is based on torsion bars. Second one is a modified model, based on spiral torsion springs. In this model a new solution of idler mechanism is applied. It provides constant tension of the tracks. Semi-active suspensions simulations results are compared with results of models with passive versions of the suspension to highlight the improvement level. Results of all models simulations are compared and analyzed to improve comfort and stability level in conditions of the modern battlefield.

Keywords Tracked vehicle · Semi-active suspension · Co-simulation · ADAMS

1 Introduction

Driving comfort is very important for the tracked vehicle crew effectiveness, which may affect on the aiming accuracy or positive results of a mission. The vertical acceleration of the vehicle body or absorbed power level may be assumed as an

T. Nabagło (✉)
Faculty of Mechanical Engineering, Cracow University of Technology,
al. Jana Pawła II 37, 31-864 Krakow, Poland
e-mail: pmnabagl@cyf-kr.edu.pl

A. Jurkiewicz · J. Kowal
Faculty of Mechanical Engineering and Robotics, AGH University of Science
and Technology, Al. a. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: jurkand@agh.edu.pl

J. Kowal
e-mail: jkowal@agh.edu.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_14

indicator of this factor. In the literature especially one strategy of ride comfort improvement is popular, but problem of compromise between comfort and safety was described by Sibielak et al. [1]. The strategy of ride comfort improvement is tightly connected with Sky-Hook damping strategy, which may be applied in a Semi-Active (SA) Suspension. This strategy was described by Ahmadian and Pare [2]. It was very widely described in literature but usually for passenger cars. This strategy is very seldom taken under consideration in application to the tracked vehicles. Next question is connected with practical realization of SA suspension system. The answer is put forward in Wray's et al. article [3]. They describe SA suspension system based on Magneto-Rheological (MR) damper. In this article, it is applied in multi-axled wheeled vehicle. In comparison to passive suspension solution, a suspension with MR dampers decreases vertical acceleration of the vehicle sprung mass. Continuing of this way of thinking, application of MR dampers should be also good solution for multi-axled tracked vehicle. Very detailed model of MR damper dynamics was put forward by Alexandridis and Goldasz [4]. It should be used in continuation of the simulation researches but the authors have conducted their researches on the simple linear model, taking into account the key MR damper limitations. As the extension of this way of thinking, the work of Rączka et al. [5] may be proposed, in which the solution of control of a suspension stiffness, in the real time, is described. In the next section of our article, the authors have described two versions of tracked vehicle model suspension. The first one is basic model, based on torsion bars system. The construction process of this model as also the simulation of the model was described by Nabagło et al. [6]. The second one is modernized model. In this model, all of torsion bars are replaced with logarithmic spiral springs. As it was said, during researches, suspension system of 2S1 tracked vehicle was modernized by usage of spiral springs. But not only by this, also others features were changed in these two above modernized models. The second one element was an idler mechanism. This mechanism was replaced with a constant force actuator, which acts on the idler wheel. This solution lets to hold a constant value of track tension. In the basic model, the idler mechanism was blocked in fixed position. Track links are connected together by so called rubber-bushings, where stiffness and damping parameters of a single bushing is adequate to real parameters of track links connection. Ruy et al. [7] describe more detailed rules and parameters of this connection in case of two links of a track. From mathematical point of view, the same solution was used in two mentioned suspension models. Also Wong [8], in his work, considers a track model. However, he proposes a simplified model of the track as an elastic belt. In more detail, he take under consideration the problem of the type of soil on which the vehicle is moving. This problem can be reduced to the model represented by the contact forces between the elements of the wheel, track and ground. Contact forces act between a single track link and a single wheel as also between a single track link and the ground. A contact forces, which are applied between wheel and ground, were described by Lee [9]. Specific parameters of these connections as also parameters of forces are described in ADAMS program documentation [10]. In simulation process with contact forces usage, geometries fidelity of parts being in contact is very important. In continuation of simulation topic, Kubela et al. [11] describes multi-body mechanism simulation in ADAMS computer program. They shows co-simulation environment with usage of Matlab-Simulink program. Such solution could be applied for calculation of more complicated control tasks or nonlinear elements reactions in the Simulink program. Huh et al. [12], in his work, took under consideration a specific idler mechanism for track tension control. Simulations as also measurements of parameters of a real tracked vehicle was performed in conditions when vehicle turns on a flat road. In contrast to Huh's article, in above article, in case of the modernized tacked vehicle, the authors also describe track tension mechanism, but it is modeled with usage of single force acting in longitudinal axle of the vehicle. Also driving conditions are different, because the authors assume the straight line driving on a road profile. The road profile shape has been measured on the Yuma Proving Ground (YPG) [13]. For estimation all above mentioned vehicle solutions, the authors used an indicator, which shows absorbed power by human body. This indicator bases on human body power absorption in vertical direction, which was described by Lee and Lins [14]. This work is associated with model, developed by U.S. Army Tank-Automotive Research and Development Center (TARDEC). This model treat a human body as a filter of vibrations power, which is absorbed by it. Because the human body reacts with different sensitivity on a different vibration frequencies. According to this work, to hold a comfort feeling, the power absorption of the human body must not exceeds 6 W, what was described by Donahue and Hedrick [15]. The power transmitted in the suspension system could be also recovered and accumulated, what was described by Kowal et al. [16]. During model improvement process, the authors familiarized with other tracked vehicle model solutions, which was described by Mężyk et al. [17] as also by Assanis et al. [18].

2 Passive and Semi-active Suspension in the Tracked Vehicle Model

For a wider view of SA suspension work effectiveness, the authors used two suspension solutions. Results of its simulation will be compared in the next section of the article. Initially passive suspension model was build. In the next step, it has been expanded to the SA suspension, what will be described in this section.

2.1 Passive Suspension Models—Basic Model

The suspension model of 2S1 vehicle in original version is equipped with torsion bars in axles of all road wheels. Original hull mass in the model is assumed as

6.7E+003 kg. In the model, torsion stiffness of torsion bars was assumed as linear, with coefficient 1.11E+002 Nm/deg. All wheels in the models are numbered, beginning from front to back, what is presented on the basic model in Fig. 1. This is the basic model of 2S1 vehicle in its initial equilibrium state, from which the dynamics simulation is started. In all passive models, dampers are mounted at the first and last road wheel axles. Damping characteristics, which was assumed for the dampers, were more precisely described in the article [6] and shown in Fig. 2a. This damper characteristic was built with data collected on the real passive damper mounted in the 2S1 vehicle. For comparison, in second part of Fig. 2b, the linear 3-dimensional characteristics of a semi-active damper is shown. This characteristics will be used in the next section of the article. The last one characteristic element, the idler wheel, has been blocked in the constant position.

Another important element of the tracked vehicle is environment in which the vehicle is moving. The soil is tightly connected with this environment. This soil is an element which contacts with the tracks and which has huge influence on the vehicle dynamics [8]. In the basic model as well as in the modified model, the authors assumed, that the contact forces have linear characteristics. The contact forces appear between wheels and track links as also track links and a ground. The contact are characterized by two main parameters, namely stiffness and damping. The stiffness parameter of contacts between wheels and tracks equals 200 N/mm, the damping equals 2 Ns/mm. The stiffness parameter of contacts between tracks and ground equals 1000 N/mm, the damping parameter equals also 2 Ns/mm.

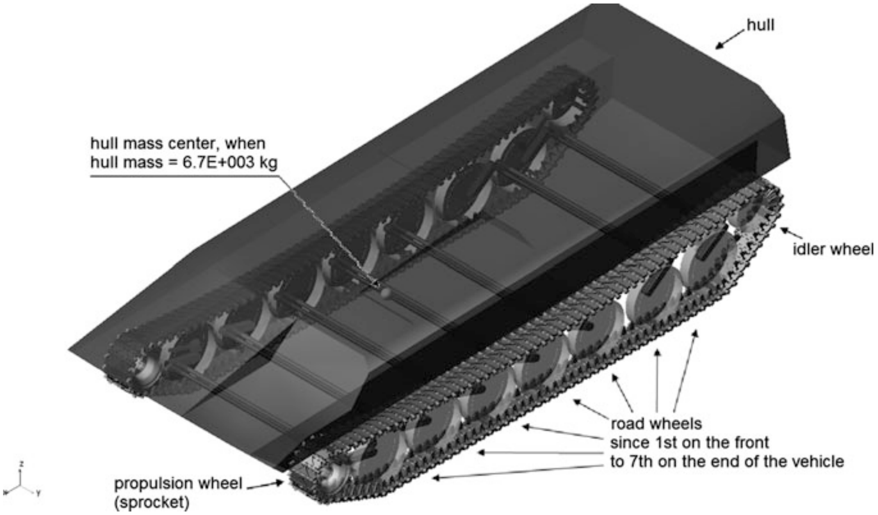


Fig. 1 Basic model of the full tracked vehicle

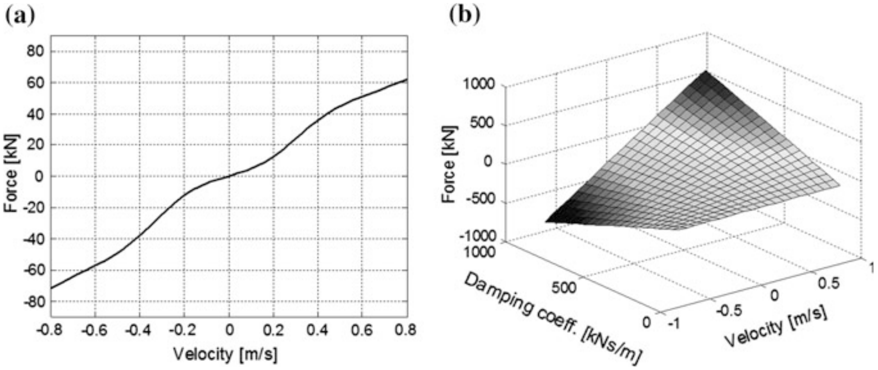


Fig. 2 Passive (**a**) and semi-active (**b**) damper characteristics

2.2 Passive Suspension Models—Modified Model

The modified passive suspension model is based on the logarithmic spiral springs (see Fig. 3b) mounted in all axles, as opposed to torsion bars (see Fig. 3a), traditionally mounted in basic version of the vehicle.

Shape of these springs is based on the shape of a logarithmic spiral (see Fig. 4a). They are mounted in packages enclosing 15 parallel connected springs, in place of the removed torsion bars. The springs package are mounted outside the hull, close to cooperating suspension arm. Thus additional free space is provided. This space could be used for installing better floor-armor or other equipment. This solution improves also safety, because all reactive elements of the suspension system are removed from floor plate and they are placed in both sides of the hull, where they may be better covered. Stiffness characteristics of the spiral springs package is presented in Fig. 4b. Because, this characteristics encloses only collapse direction part, in the final spiral spring model, the characteristics was mirrored to the rebound direction. It was done in respect of assumptions, that in close to zero area, the

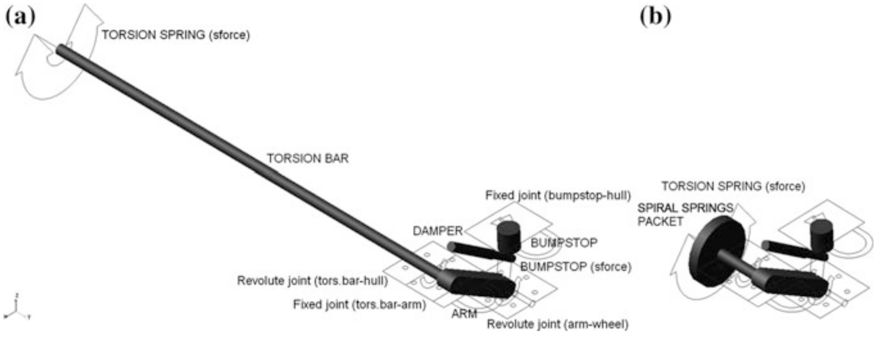


Fig. 3 Units of suspension: **a** with torsion bar, **b** with spiral springs package

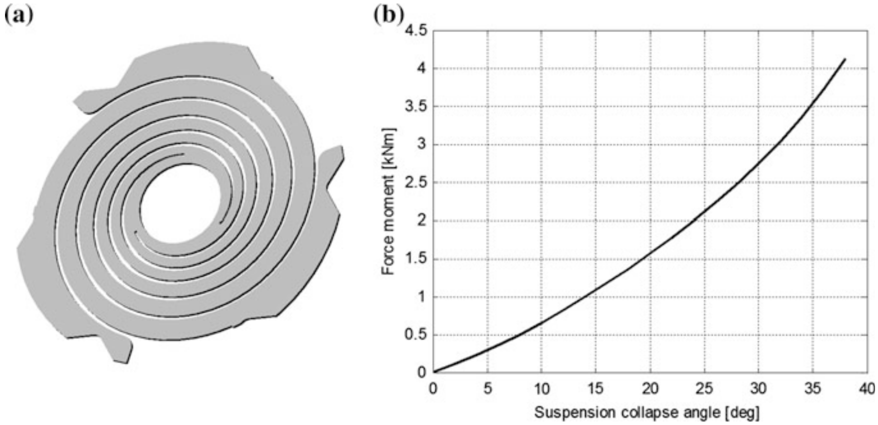


Fig. 4 Single spiral spring (**a**) and stiffness characteristics of 15 springs package (**b**)

stiffness characteristics should be symmetrical. This characteristics is result of measurement series on the real spiral springs package in laboratory conditions. In the real vehicle these springs packages works in various driving conditions. For this reason the authors had to take simplification assumptions, such like constant temperature and linear characteristics in whole range of bump rebound movement.

In the vehicle with the modified suspension system, the hull mass was increased to 11.57E+003 kg. Also the hull mass center was moved 807 mm back to the geometrical center of the hull (see Fig. 5). The last one modification is associated with idler wheel mechanism. Initially fixed to the hull, the idler mechanism was

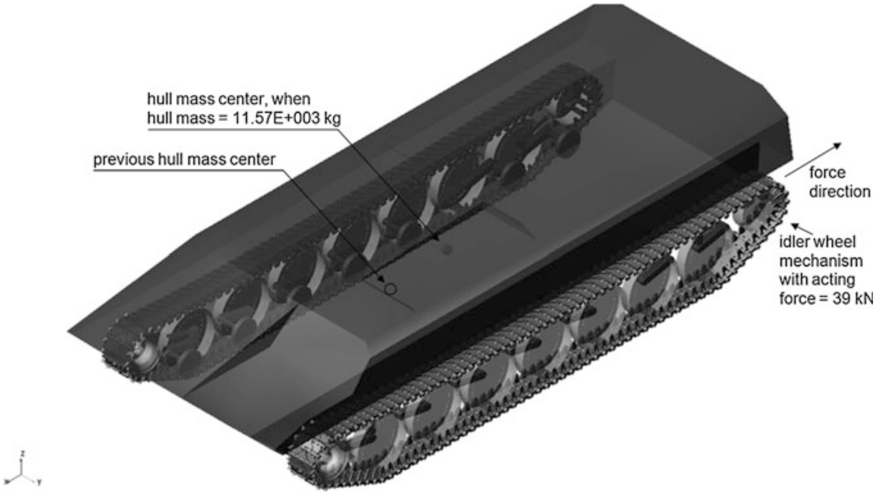


Fig. 5 Modernized model of the full tracked vehicle

rebuilt to act on the track with constant force, which equals 39 kN. This force acts in longitudinal axis of the vehicle, as it is shown in Fig. 5. This solution is used for maintaining a constant tension of the tracks. It is simplified version of real idler wheel mechanism which is based on eccentric mechanism. However, this simplification realizes the main aim of the constant track tension. Forces, acting on the idler wheel from the track side during acceleration or deceleration, may be higher than the idler force value. In this situation these forces cannot be balanced with 39 kN. In result of it, the idler wheel may be moved to the front of the vehicle. For this reason, a bump-stop is used to stop movement of the idler wheel. It is stopped after displacement 100 mm from the construction position.

These all modifications were enclosed in the modernized model of 2S1 vehicle shown in Fig. 5. In this figure the vehicle is shown in its initial equilibrium state, from which the dynamics simulation is started.

2.3 Sky-Hook Strategy Adopted to the Tracked Vehicle Suspension

The authors decided to use Sky-hook strategy in the full tracked vehicle model because of very significant comfort improvement effect for so called quarter-car model [2]. The Sky-hook system concept is illustrated in Fig. 6a. It assumes existence of a frame, which is fixed to the vertical coordinate of the absolute coordinate system. This frame is understood as "Sky". In Fig. 6b, a functional model is presented, where c_{sky} should equal $F_{sky}/(\dot{z}_1 - \dot{z}_1)$. In the figure, m_1 symbolizes ¼ part of mass of the 4-wheeled vehicle body. Mass m_2 is the unstrung mass of the suspension. Other symbols are described as follows: k_1—suspension stiffness, c_1—variable damping, k_2—wheel stiffness, c_2—wheel damping, ξ—excitation coming from the road surface.

To adopt this system to the tracked vehicle model, the authors should take some assumptions. As it is known there is not possible to isolate a single suspension unit

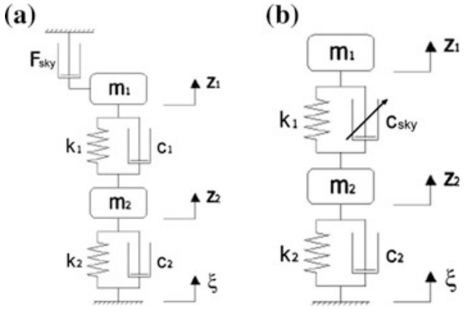


Fig. 6 Simple sky-hook suspension model: conceptual (**a**), functional (**b**)

containing wheel and appropriate part of the sprung mass, from the full tracked vehicle. Nevertheless, relation between vertical velocities of the sprung mass and the unsprung mass, in the Sky-hook strategy, is well known. For the quarter-suspension model this relation is described with Eq. (1).

$$c_{sky} = \frac{2\zeta_{sky}\dot{z}_1\sqrt{m_1k_1}}{(\dot{z}_1 - \dot{z}_2)} \tag{1}$$

where: ζ_{sky}—relation of sky-hook damping to the critical damping; $\dot{z}_1$—vertical velocity of the sprung mass; $\dot{z}_2$—vertical velocity of the unsprung mass; m_1—sprung mass; k_1—suspension stiffness.

This formula is base for a sky-hook controller, which is applied in the model. In regards to fact, that in the vehicle model, a suspension based on a torsion spring or torsion bars is applied, their stiffness values cannot be introduced to the controller directly. To introduce this stiffness values as parameter to the controller, the stiffness should be reduced to the wheel axle point, what it will be described in the next subsection.

2.4 Semi-active Suspension Models

To build the SA suspension model of the 2S1 vehicle suspension, few changes have to be introduced to the previous constructed model of the passive suspension. First one was associated with the SA control system with the Sky-hook controller. As input parameter for the Sky-hook controller, a single stiffness coefficient of the suspension unit was used. This unit was associated with a single road wheel. The above mentioned coefficient was obtained through the reduction suspension stiffness to the wheel axle point (see Fig. 7a). According to Fig. 7a, it is stiffness coefficient of the theoretical spring. This coefficient was calculated from its theoretical characteristics and the coefficient is about 65 N/mm for torsion bar and 55 N/mm for torsion springs. For the reason, that the vehicle has 14 wheels, as an input parameter, 1/14 part of the vehicle sprung mass is taken. The 1/14 of the vehicle body equals 480 kg for model with torsion bars and 826.4 kg for model with torsion springs. Second change was associated with a passive suspension system construction modification. This modification was conducted by removing of passive dampers and replacing them with SA dampers. To improve effectiveness of their work, some construction parameters was changed. To adopt the sky-hook strategy to the tracked vehicle suspension controller, also position of the damper should be modified to the position shown in Fig. 7b.

This solution is applied to ensure, that damping coefficient of the SA suspension is exactly the same as damping coefficient calculated by the Sky-hook controller.

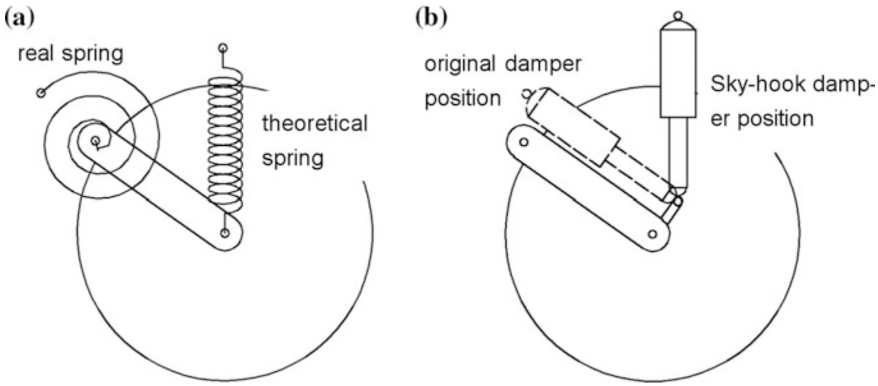


Fig. 7 Illustration of the suspension stiffness reduction (**a**), illustration of the SA and the passive dampers positions (**b**)

The Sky-hook controller has 8 inputs and 4 outputs. As the outputs signals are four damping coefficients for four semi-active dampers. Because full sky-hook controller is built with four sub-controllers, each sub-controller supports one SA damper. For this reason each sub-controller needs two input signals of vertical velocity. First one signed with v_1 is associated with sprung mass of suspension model and it is measured directly on the upper end point of the damper. This point is fixed to the vehicle body. Second input is signed with v_2 and its signal is measured on the lower end point of the damper. This point is fixed to the suspension arm and wheel axle connection. During simulation experiments the authors made corrections to the control formula. This corrections was associated with ζ_{sky} value. Generally it was associated with constant value $2_{sky}\sqrt{m_1 k_1}$. Formula $2\sqrt{m_1 k_1}$ equals so called critical or aperiodic damping coefficient and it is characteristic constant for suspension system unit. Therefore, in the natural way, only coefficient ζ_{sky} let modify this value. $\dot{z}_1$ and $\dot{z}_2$ variables are representative for Sky-hook strategy [2], but constant part should be proportional to m_1 and k_1. According this way of thinking, correct estimation of ζ_{sky} coefficient should let improve effects of cooperation tracked vehicle suspension with the Sky-hook controller. Through experiment, the authors estimated $\zeta_{sky} = 0.07$, as the best value for the SA model with torsion springs. In the simulation model, the Sky-hook controller was realized in the Simulink model (see Fig. 8). In this model, block named Sky-hook algorithm contains Eq. (1).

Because in Eq. (1) is risk of dividing by zero, the model was enriched with additional elements, which gives default damping coefficient value (c_{ini}), when this coefficient is not specified. The default coefficient equals 88 Ns/mm. In respects to the fact that the semi-active damper has limited range of the damping, in the controller, the upper limit of the damping coefficient was assumed on 880 Ns/mm, and the lower on 8.8 Ns/mm.

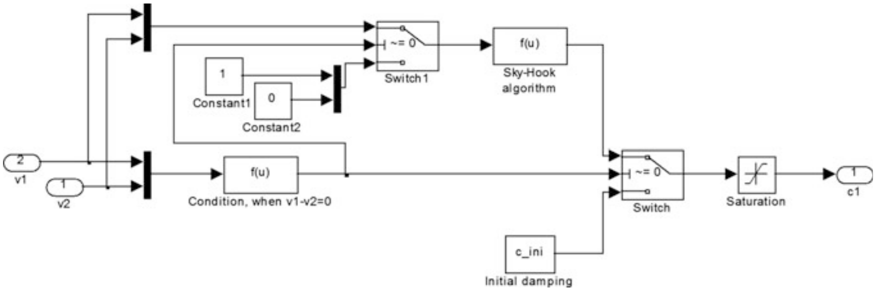


Fig. 8 Sky-hook controller unit realized in Simulink computer program

2.5 ADAMS-Matlab Co-simulation Conditions

Whole SA suspension model is simulated in co-simulation process between MSC. ADAMS and Matlab computer programs. For comparison of the control system effectiveness, simulations of the passive and SA suspension was conducted. In case of the SA suspension, the simulations were conducted in ADAMS and Matlab co-simulation conditions. For the co-simulation, a characteristic feature is so called communication interval. It equals 0.001 s. This interval should be close to communication interval in the real SA suspension system with MR dampers [4]. The SA suspension model contains two main elements. First one is a model of full tracked vehicle suspension, with changeable damping coefficients of the dampers. This part was built in MSC.ADAMS program. The second element is Sky-hook controller for four independent units associated with SA dampers. This controller was built in the Simulink program, which is part of the Matlab. One of four above mentioned units is shown in Fig. 8.

3 Simulation in the Yuma Proving Ground Conditions

To make results of the simulations verifiable, the authors used well known road profile, which was collected on the Yuma Proving Ground, and it is known as RMS3 [13]. During simulation, all mentioned models are simulated in conditions of the straight line driving with variable velocity. The time trace of this velocity is presented in Fig. 9a.

In a first phase of the movement, the vehicle accelerates. It takes 5 s. Next it decelerates. It takes 3 s, but after deceleration the vehicle starts to move with constant velocity. In Fig. 9b, a road surface shapes are presented. These are shapes for the left and the right track, and they was collected with specialized measuring device on the Yuma Proving Ground.

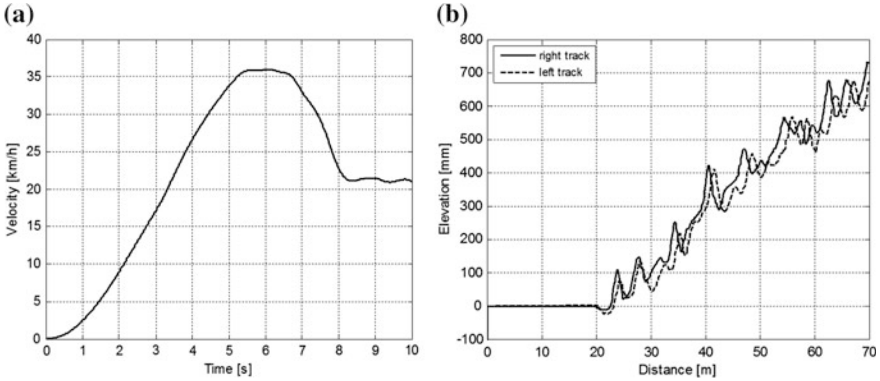


Fig. 9 Longitudinal velocity (**a**), right and left track road shape (**b**)

4 Simulation Results Analysis

In this section, two types of results are presented. First one is associated with time domain analysis, second one with frequency domain analysis. Only analysis in these both domain can gives full view on the model. It shows not only its behavior in time, but also information about frequencies might be transmitted from road wheels to crew bodies. In this section, the authors compare effectiveness of passive and Semi-Active suspension solutions.

4.1 Time Domain Analysis—Modified Model

Results of above described models are compared during their time analysis. Time analysis was conducted since 5th to 10th second of the simulation. In this subsection, the authors present time traces of vertical acceleration of the vehicle hull mass center for passive and SA suspension of the modified vehicle model (Fig. 10a). In the case of modified model, where all axles are equipped with spiral springs, amplitudes of acceleration, in passive suspension model case, are higher than in SA suspension case. It is not rule in 100 % of amplitude values, because in time traces of acceleration amplitudes, one opposite situation appears. It appears in 7th second of the simulation. Nevertheless, similar situations appear such seldom, that this control system could be acceptable for the 2S1 vehicle. In case of the hull mass vertical acceleration, the difference between passive and SA suspension model is not so big. In this situation another indicator appears to be more helpful. As it is known, for efficiency of vehicle crew, a comfort is very important. When crew is exposed to activity of too high vibration energy, their efficiency is getting lower and their reaction time increases radically. To estimate influence energy for human body, the authors used Human Response Filter (HRF) developed by U.S. Army TACOM [14].

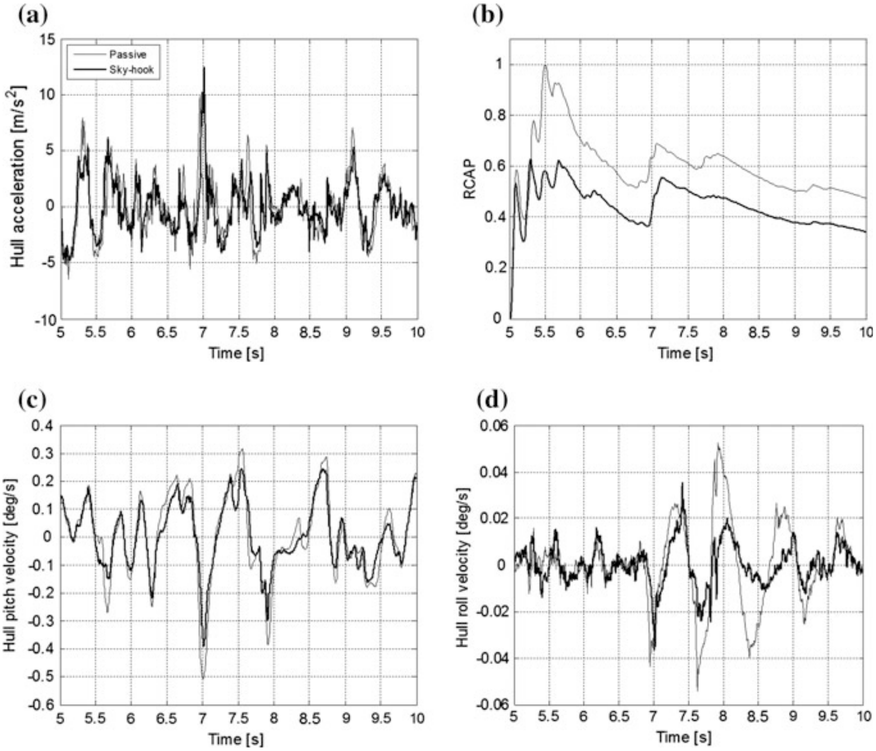


Fig. 10 Time traces of vertical acceleration of the vehicle hull (**a**), human body cumulative absorbed power (**b**), hull pitch angular velocity (**c**), hull roll angular velocity (**d**)

Evaluation of level of absorbed power by the human body is possible with usage of this filter. Based on it, the authors calculated time traces of Relative Cumulative Absorbed Power (RCAP), which are shown in Fig. 10b. All values of these time traces are relative to power absorbed by human body placed in the center of the hull mass in basic model as well as the modified model.

$$CAP = \frac{1}{T}\int_{0}^{T} P \, \mathrm{d}t \tag{2}$$

$$RCAP = CAP/CAP_{\max} \tag{3}$$

where T is time of power absorption, P is absorbed power in vertical direction, calculated with usage of vehicle hull vertical acceleration filtered by HRF. RCAP for above three models are shown in Fig. 10b.

These time traces clearly show that absorbed power in case of the SA suspension model is much lower than in the passive model case. Therefore in this situation, solution of the SA suspension model gives noticeably better results than passive suspension model solution. In Fig. 10c, pitch velocity is presented. This velocity time traces are significantly better in the SA case than in the passive case, what shows that SA suspension stabilizes vehicle along its longitudinal axis. The last part of Fig. 10d contains a comparison of roll velocity of the vehicle hull. In this case, situation is similar, but not such significant like in the pitch velocity case. SA suspension stabilizes the hull also in transverse axis of the vehicle.

4.2 Time Domain Analysis—Basic Model

In the basic model case, a comfort improvement is not such significant like in the modified model case. During observation of differences between hull acceleration amplitudes for passive and SA suspension are not clear (Fig. 11a). Also in this case RCAP indicator appears to be more useful. This indicator shows that also in the basic model case, SA suspension gives better results because of lower energy transmission in the suspension system (Fig. 11b).

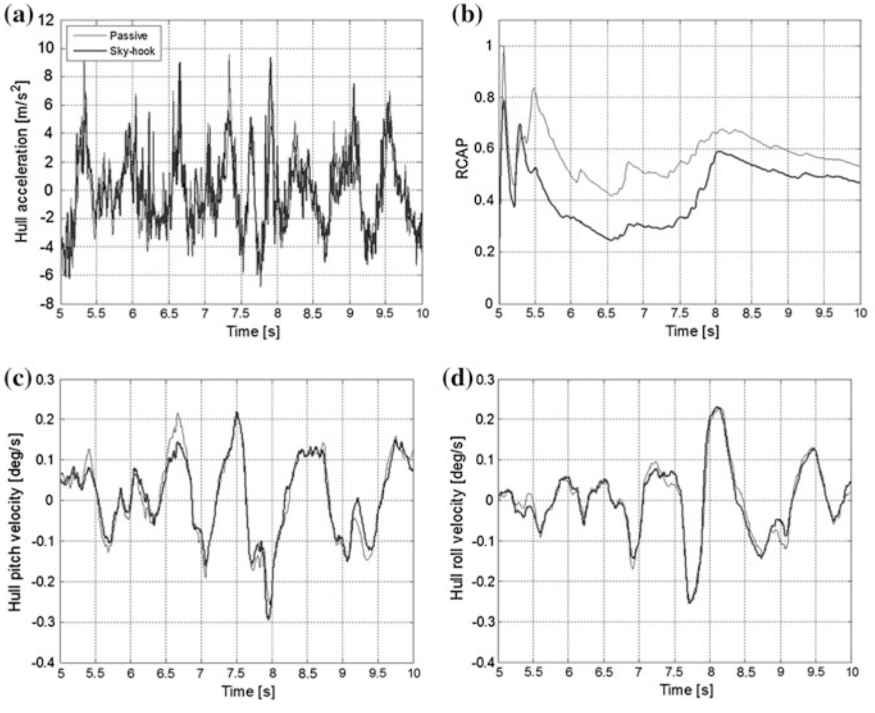


Fig. 11 Time traces of vertical acceleration of the vehicle hull (**a**), human body cumulative absorbed power (**b**), hull pitch angular velocity (**c**), hull roll angular velocity (**d**)

During analysis of the hull pitch angular velocity and the hull roll angular velocity, any significant differences between passive and SA model are not noticeable. Than in this case, stability improvement level is minimal (see Fig. 11c, d). Such weak effect of SA suspension application may be caused by fact, that in the basic model, the mass center is not in the center of the hull geometry. From this reason surrogate masses for the controllers of front and rear SA dampers are radically different. With this fact is connected another one, namely, that the critical damping for the front SA dampers are different than for the rear SA dampers. Nevertheless, the authors assumed simplification, that critical damping for all SA dampers are the same. This simplification let use the same sub-controllers parameters for all SA dampers.

4.3 Frequency Domain Analysis—Modified Model

For complete picture of the simulations, results of frequency domain analysis are shown. In Fig. 12a, one can see, that in the SA model case, low frequencies

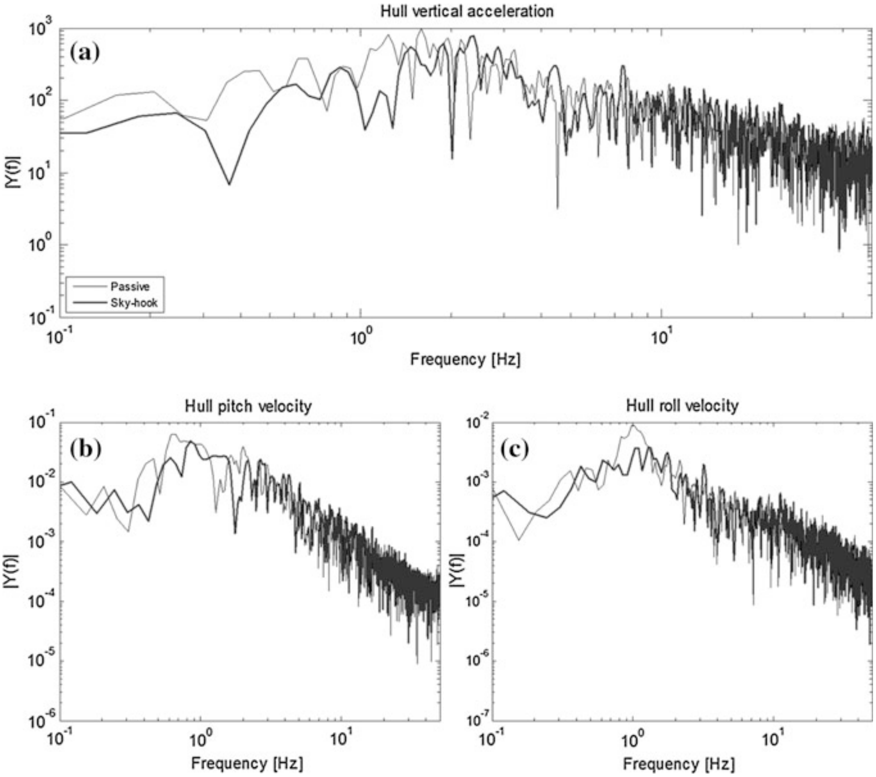


Fig. 12 Frequency characteristics of vertical acceleration of the vehicle hull (**a**), hull pitch angular velocity (**b**), hull roll angular velocity (**c**)

(0.1–10 Hz) of the hull vertical acceleration are stronger damped in comparison to the passive suspension case. Higher frequencies amplitudes are approximately on the same level for both suspension solutions. Similar situation is in case of hull pitch angular velocities and hull roll angular velocities.

Than frequency domain analysis confirms conclusions from time domain analysis, that SA suspension stabilizes rotational movements of the vehicle hull. It shows also, that this stabilization effect is associated mainly with low frequencies of these movements.

4.4 Frequency Domain Analysis—Basic Model

Similarly like in Sect. 4.2, one can observe an insignificant improvement of driving comfort through reduction of the hull vertical acceleration amplitude. It concerns mainly the low frequencies. The higher frequencies amplitudes are almost the same

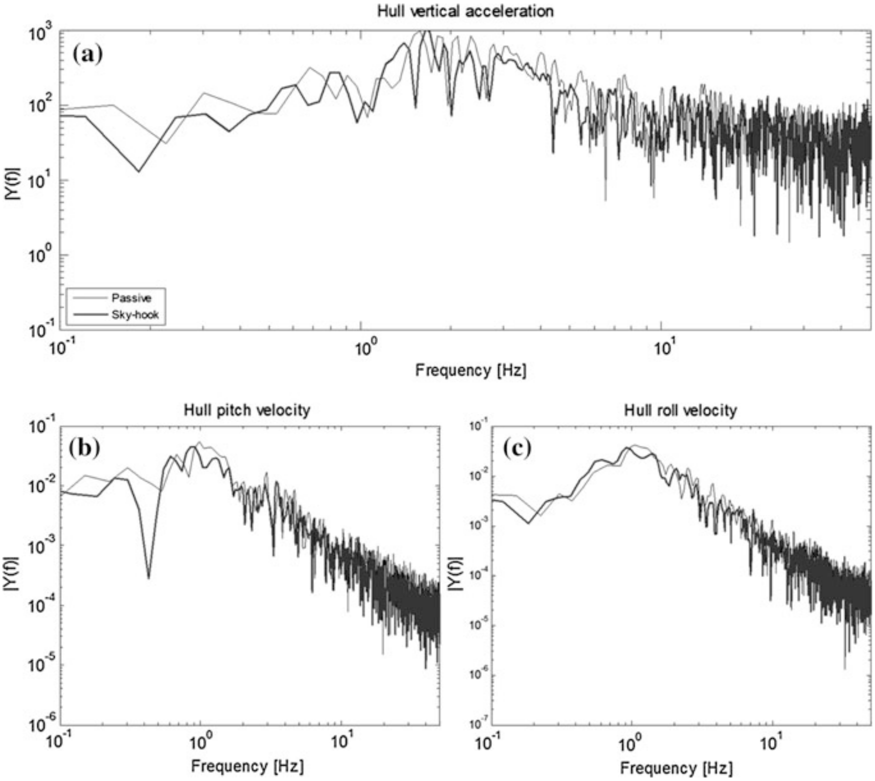


Fig. 13 Frequency characteristics of vertical acceleration of the vehicle hull (**a**), hull pitch angular velocity (**b**), hull roll angular velocity (**c**)

for the passive and SA suspension models. Also frequency characteristics for pitch angular velocity and hull roll angular velocity confirms conclusions from Sect. 4.2, that level of rotational movements stabilization is almost the same for the passive and SA suspension model. It applies to the entire frequency spectrum (Fig. 13).

5 Conclusions

In conclusion of results analysis, one can notice advantages of usage of the Semi-Active suspension system. This system, in the case of the modified suspension system, reduces vertical acceleration amplitudes. When this acceleration acts on a human body, this body absorbs power, then through reduction of the vertical acceleration, the absorbed power is also reduced. This is confirmed by the RCAP indicator. This indicator shows, that driving comfort, when SA suspension is applied, radically increases. In the simulation conditions, by the comfort improvement, a reduced level of energy absorption by the human body is understood. This factor has big influence on the vehicle crew efficiency. But not only this parameter is improved. SA suspension stabilizes also rotational movements of the vehicle body, in longitudinal as also in transverse axis of the vehicle. Then usage of this kind of controlled suspension system improves also stability of the vehicle, and may also improve its maneuverability. In the case of the basic model, the improvement level is not such significant. It could be coursed by construction parameters of this model. The modified model has two important features. First one is bigger mass of the hull. Second one, and maybe more important, is symmetrically placed mass center. This mass center is very close to geometrical center of the hull. From this reason, parameters of sky-hook sub-controllers, which control the front and the rear SA dampers may be the same. In contrast, the basic model has asymmetrically placed mass center of the hull. It is significantly moved to the front of the hull. From this reason, a bigger mass is on the front part of the suspension than on the rear part. In this situation, a critical damping on the front and the rear SA dampers should be different. In connection to this fact, also parameters controllers, which control the SA dampers, should be different. Nevertheless, in the basic suspension case, the SA suspension solution holds the performance on the same level as the passive suspension solution. SA suspension applied in the models is based on the telescopic damper with a piston. Because in the modified model, the torsion spiral springs are applied, the control system may be applied also to the rotational dampers. This damper could be autonomic unit, in parallel connected with the spiral springs, or it could be integrated with the spiral springs. This kind of damper might be realized with usage of magneto-rheological fluid as a damping medium. It appears to be the best solution, because MR dampers have low energy requirement. In summary, the usage of above-described solution of the semi-active suspension system, especially for the modernized model of the 2S1 vehicle, through increasing the stability and the comfort level may significantly increase the advantage in conditions of the modern battlefield.

References

1. Sibielak, M., Konieczny, J., Kowal, J., Rączka, W., Marszalik, D.: Optimal control of slow-active vehicle suspension—results of experimental data. J. Low Freq. Noise Vibr. Active Control **32**(1–2), 99–116 (2013). ISSN 1461-3484
2. Ahmadian, M., Pare, C.A.: A quarter-car experimental analysis of alternative semiactive control methods. J. Intell. Mat. Syst. Struct. **11**, 604 (2000)
3. Wray, A., Jimenez, A., Anderfaas, E., Hopkins, B., LeNoach, P.: Magneto-Rheological Fluid Semiactive Suspension System Performance Testing on a Stryker Vehicle. SAE Technical Paper 2006-01-1379 (2006)
4. Alexandridis, A.A., Goldasz, J.P.: Simplified model of the dynamics of magneto-rheological dampers. Mechanics **24**(2), 47–53 (2005)
5. Rączka, W., Sibielak, M., Kowal, J., Konieczny, J.: (2013) Application of an SMA spring for vibration screen control. J. Low Freq. Noise Vibr. Active Control **32**(1–2), 117–132 (2013). ISSN 1461-3484
6. Nabagło, T., Jurkiewicz, A., Apostoł, M., Micek, P.: Construction and simulation of a 2S1 tracked vehicle model and its verification using vertical forces on the road wheels while overcoming a single obstacle. Solid State Phenom. **177**, 168–176 (2011)
7. Ryu, H.S., Bae, D.S., Choi, J.H., Shabana, A.A.: A compliant track link model for high-speed, high-mobility tracked vehicles. Int. J. Numer. Meth. Eng. **48**, 1481–1502 (2000)
8. Wong, J.Y.: Dynamics of tracked vehicles. Vehicle Syst. Dyn. **28**, 197–219 (1997)
9. Lee, K.: A numerical method for dynamic analysis of tracked vehicles of high mobility. KSME Int. J. **14**(10), 1028–1040 (2000)
10. ADAMS ver. 2010, Help Documentation (ADAMS/View, Solver). MSC. Software (2010)
11. Kubela, T., Pochyly, A., Singule, V.: Advanced tools for multi-body simulation and design of control structures applied in robotic system development. Solid State Phenom. **164**, 387–391 (2010)
12. Huh, K., Cho, B.H., Choi, J.H.: Development of a track tension monitoring system in tracked vehicles on flat ground. Proc. Inst. Mech. Eng. Part D: J. Automobile Eng. **215**, 567–578 (2011)
13. Final Report for U.S. Army Tank-Automotive Command, Terrain severity data generation at Yuma Proving Ground. TACOM Report No. 13491, Contract No. DAAE07-89-C-R106, NATC Project No. 20-17-399, Oct 1989–Nov 1990
14. Lee, R.A., Lins, W.F.: Human Vibration Measuring Instrument. Technical Report No. AD785648, U.S. Army Tank-Automotive Command, Warren, Michigan (1973)
15. Donahue, M.D., Hedrick, J.K.: Implementation of an active suspension, preview controller for improved ride comfort. In: Nonlinear and Hybrid Systems in Automotive Control, pp. 1–21. Springer, London (2003). ISBN 978-1-85233-652-3
16. Kowal, J., Pluta, J., Konieczny, J., Kot, A.: Energy recovering in active vibration isolation system—results of experimental research. J. Vibr. Control **14**(7), 1075–1088 (2008). ISSN 1077-5463
17. Mężyk, A., Świtoński, E., Kciuk, S., Klein, W.: Modelling and investigation of dynamic parameters of tracked vehicles. Mech. Mech. Eng. **15**(4), 115–130 (2011)
18. Assanis, D.N., Bryzik, W., Castanier, M.P., Darnell, I.M., Filipi, Z.S., Hulbert, G.M., Jung, D., Ma, Z.D., Perkins, N.C., Pierre, C., Scholar, C.M., Wang, Y., Zhang, G.: Modeling and simulation of an M1 abrams tank with advanced track dynamics and integrated virtual diesel engine. Mech. Struct. Mach. **27**(4), 453–505 (1999)

Experimental Mobile Robot—Hardware

Krzysztof Jaskot and Tomasz Łakota

Abstract The chapter presents the hardware part of the mobile robot. Whole electronics circuit was divided into functional blocks and each of them is described separately. For each, the corresponding part of schematic was presented and explained. The most important decisions on selected parts or solutions are discussed and justified.

Keywords Mobile robots · Electronics · Microcontroller · Wireless communication

1 Introduction

Mobile mini-robots constitute an important equipment for research of control algorithms. During last decade, many constructions for this purpose were developed. One of examples is the series of Kephera robots, which are widely used at many universities in robotics courses. It is commercial product, with emphasis put on high modularity. Very different approach is presented in the work [1, 2], where the cheap, compact mobile robot is described. There are also many constructions intended to be used in robot soccer competitions [3]. One of such models is 188 Soccer Robot produced by Microrobot. In this work, a new controller for this model is presented.

The Experimental Robot is equipped with two-wheeled differential drive operated by two DC motors (see Fig. 1). On shaft of each motor there is a two channel incremental magnetic encoder, which gives 512 pulses per rotate. The goal is to make a controller which allow to realize effectively complex control algorithms planned by external, master control system. Data should be exchanged with the master system using wireless communication module. There should be a possibility to use the robot in the multi-agent system [4], so it has to be taken into account, that

K. Jaskot (✉) · T. Łakota
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: krzysztof.jaskot@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_15

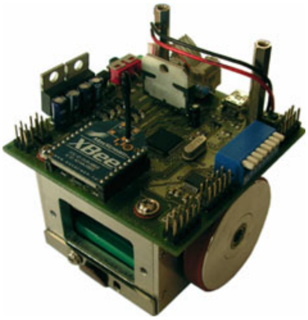


Fig. 1 A experimental mobile robot with controller

several identical robots can be present in the range. A 6-cell battery (which gives 6–7.8 V) is the power supply both for motors and for electronic circuits. The robot itself has no other sensors but the encoders, however in whole system there is a stationary camera mounted above the field where the robot is located. Together with proper software it build a vision system that provides the pose (position and orientation) of the robot in the global coordinate system [5–7]. This data can be sent to the robot using the wireless link.

In [8] previous approach to the same problem was presented. The work was based on the board equipped in the Microchip PIC18F452 microcontroller and Radiometrix BiM-418-F radio module and described Brain On-Board System [9]. The PID controller was implemented for the speed of wheels as well as for robot's orientation. Moreover, the potential field algorithm was used to find the path to destination point. It was assumed that the only obstacles on the field are other robots, which positions are received from the vision system [10–12]. Also going the shortest way to destination point was realized by the on-board system.

One of the problems is poor throughput of radio modules used it is 40 kbits at most. Moreover, the transmission is very sensitive to noises from the motors brushes. Slow and unreliable transmission is a serious drawback of the controller, since there is much data to be send (including poses of all robots in the scene).

2 Electronics

The circuitry designed for the controller is composed of the central unit (described in the next section) and several functional blocks connected to it. The block diagram of whole circuit is presented in Fig. 2.

Central processing unit is the most important part of the controller. It serves many functions in the robot. First, it coordinates work of all other subsystems. Second, it is the place, where the control algorithms are implemented. Next, data from the wireless device is received and interpreted here. Apart from these tasks, there are also other ones, which are performed by this part, so it had to be selected carefully, taking into account many criteria.

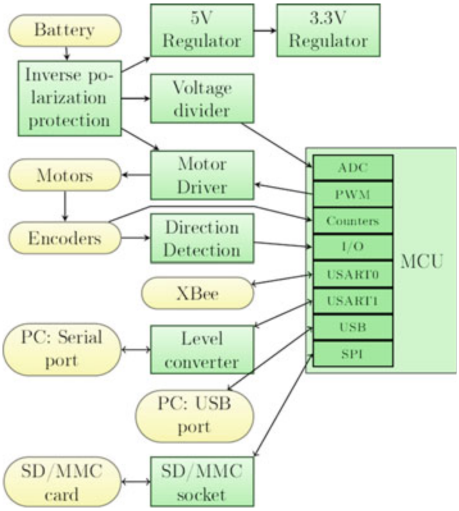


Fig. 2 The block diagram of the controller circuit. External devices are denoted by *yellow boxes*, and parts of circuits by the *green* ones (Color figure online)

2.1 MCU

Nowadays, there is a wide variety of highly integrated chips under a common name ‘microcontroller’ or shortly ‘MCU’ (MicroController Unit). They are small, cheap, and need almost no external devices to work. Moreover, many useful peripheral devices, like timers/counters, Analog to Digital converters, Uart (Universal Asynchronous Receiver/Transmitter) and USB drivers, are integrated into them. After some market research the model from AT91SAM7S series produced by ATMEL company was selected. The basic requirements and other criteria which was taken into account are discussed below.

As it was mentioned earlier, the microcontroller will perform many tasks simultaneously. Some of them are time critical, and some need relatively big computational power (as for microcontrollers). It cannot be stated precisely how speed of processing is needed, however it can be estimated that it should be at least 10 millions of simple arithmetical operations per second (including multiplication, division not necessarily). Greater speed is advantage. Selected model can operate with 55 MHz clock [13]. Number of clocks per instruction varies, however it is usually a small number (1–4) [14].

2.2 Sufficiently Large Random Access Memory

Some algorithms may need to store large amount of past data for future use (e.g. distance traveled by each wheel at many past points of time, in order to use it in position estimation, when the frame from vision positioning system is received).

It is also possible that some sophisticated tracking algorithms would need much memory for temporary arrays.

Moreover, it would be useful to be able to remember a history of robot's position, speed, control variables, some internal variables of controllers and so on. We can imagine a situation in which we run the robot with newly-programmed algorithm and observe it's behavior for several seconds. Then, we connect a robot to PC and read data registered during this 'trial-ride' in order to analyze it and fix or tune the algorithm.

Since wireless transmission is used for other purposes, it will be more convenient to store this data temporarily in the memory, rather to transmit it in the real time. For this reason, relatively big amount of memory is needed. Selected model is produced in several versions with varying RAM and FLASH size. Finally the one with 16 KB of RAM and 64 KB of FLASH is used.

2.3 PWM

The easiest way of controlling brushed DC motor from digital circuit is Pulse Width Modulation technique. Most of the modern microcontrollers have hardware support for PWM. Such a feature makes possible to focus on other tasks in software rather than manual switching logical level on some pin. Because there are two motors and each of them should be controlled independently, the chosen device is expected to have at least 2 separate channels. Selected microcontroller posses four PWM channels, so it is definitely enough for my purposes.

It would be convenient if output from encoders was sampled by the hardware and edges were counted automatically by proper peripheral device in the microcontroller. Built-in counter with external trigger should perform this effectively. Since there are two wheels, two counters are needed, each with separate trigger input. According to datasheet [13] there is one three-channel 16-bit timer/counter and three external inputs to them (which can be configured freely). Therefore, I can configure them in such a way, that each of two counters will count pulses from the encoder connected with one wheel.

2.4 Communication Channels

One UART channel is needed to communicate with wireless transmission module. Moreover, it would be helpful to have some channel to communicate with PC for debug and other development purposes, and UART interface would be appropriate for this purpose too, so the second channel is necessary. Additional SPI (Serial Peripheral Interface) would be advantage, since there is a possibility to connect SD

(Secure Digital)/MMC (Multi Media Card) flash card to this interface (as external storage for logs). Selected chip has absolutely enough communication interfaces. It has two USART controllers (which can work as UART), SPI, USB 2.0, and TWI (Two Wire Interface).

2.5 ADC Channel

It is not critical, however very useful feature. It gives us a possibility to monitor battery voltage and detect the moment, when this level is too low that can cause unstable work of electronics. This will also prevent damage of battery. Selected model has one eight-channel 10-bit Analog-to-Digital Converter.

The great advantage of selected model is the fact, than commonly known opensource compilation tools called GCC (Gnu Compiler Collection) is available for this platform.

2.5.1 Schematic

In Fig. 3 there is shown a part of schematic containing the AT91SAM7S microcontroller together with it's connections and some elements. Detailed description of individual components is given below. We can see that two distinct voltages is needed for the MCU: 3.3 V (3.0–3.6 V is acceptable [13]) and 1.8 V (1.65–1.95 V is acceptable [13]). The second one can be taken from the internal voltage regulator (VDDOUT pin). In order to ensure stable work of whole circuit, noises on the power lines have to be eliminated. For this reason decoupling capacitors are used. According to the application note for AT91SAM7S series [15], decoupling capacitors are needed for the the following pins: VDDFLASH, VDDIO, VDDIN, VDDOUT, VDDCORE, VDDPLL.

2.5.2 External Oscillator

A 'SAM-BA' (SAM Boot Assistant) bootloader, which allows to upload a program quickly using USB port, needs some particular frequency of the oscillator. This is not a limitation of computing speed, however, since any frequency of the main processor's clock, that would be finally necessary, could be obtained using divider and multiplier configured from the software level. Finally the 18.432 M HC49 crystal oscillator with frequency tolerance 30 ppm was used (Fig. 4).

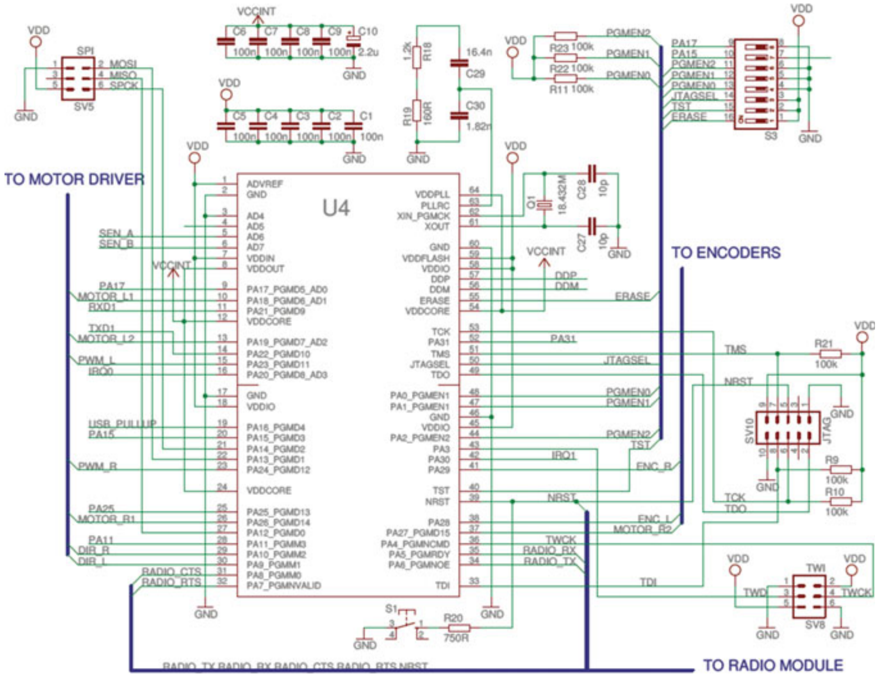
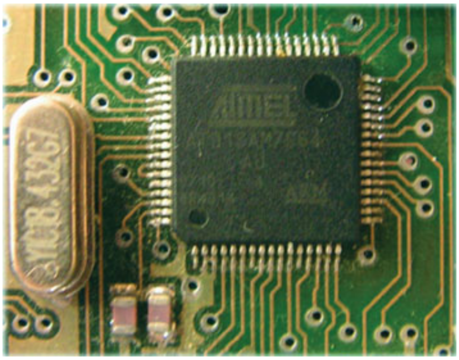


Fig. 3 The microcontroller connection

Fig. 4 The microcontroller on a board



2.6 Wireless Communication

The next very important part of the system is the wireless communication module. It has to be a reasonable compromise between price, complexity, convenience in use and transmission quality (rate, duplex, delays, noise immunity, range). Several solutions were taken into account. A short research on each of them was made.

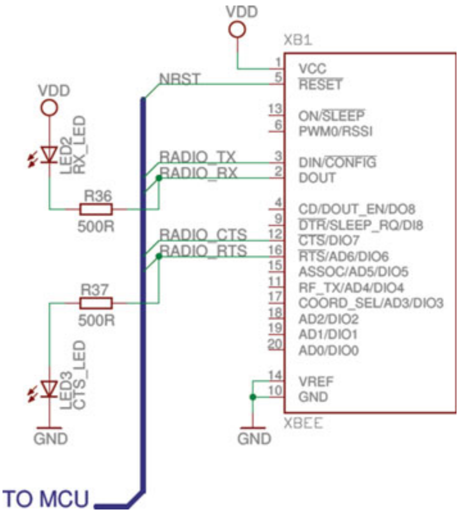
- **CC1000** This is a simple and cheap low-power wireless module. It works on frequency range 300–1000 MHz and allows to transmit or receive up to 76.8 kbps. Data direction is switched manually. Moreover some external components are required (e.g. antenna). That make this module not very convenient in usage.
- **Bluetooth devices** Bluetooth protocol has the advantage, that there are cheap USB dongles available for PC, so there would be no need of creating device for PC-side. Also transmission speed is a relatively high (few Mbps depending on version). However the Bluetooth protocol is rather complicated, which is serious disadvantage of this solution—it would cause greater CPU usage and latencies. Price of modules for use with MCUs is greater than in case of previously described devices.
- **XBee** XBee from Digi is a wireless module intended to be used in ZigBee networks. It is, however, very flexible device. In the simplest case no configuration is needed, and it works in so called 'transparent mode', which means that all (raw) data is broadcasted and received by all modules in the range. When more control needed—powerful API mode available. Implementation of protocol stack is not required as in Bluetooth. It is more sophisticated than CC1000, but easier to use. Absolutely no external devices are needed to work. Data can be transmitted in one direction a time, but it is managed automatically. The range of basic XBee modules is 30 m in urban area. The real data rate is 115 kbps, which is more than sufficient. Moreover it provides high noise immunity, which is critical because of disturbances produced by the brushes of motors. Another important feature is low latency—it is not documented, however technical support claimed that it is below 10 ms for 32 bytes at 115 kbps. This module was finally selected.

Selected XBee modules has also some features, that are not crucial for the assumed goals, but may be useful in further development. There is possibility of sending packets selectively, only to selected receiver (instead of all). The RSSI (Received Signal Strength Indication) is available, so can be read and used. There are also some sleep modes, that can reduce power consumption if managed well. Analog-to-digital converters and digital I/O pins could be used, if there were too few of them in the MCU. Figure 5 shows a part of schematic with the XBee module.

2.7 Motor Driver

The basic functionality of the controller, which is the subject of my work, is controlling the motion of the mobile robot. For this purpose, there are two brushed DC motors on the robot board. They need to be controlled by the MCU, but power supply should be taken directly from battery. For this reason, the dual full-bridge driver (H-bridge) L298 was used to supply them [16]. The L298 is an integrated

Fig. 5 The radio module connection



circuit which allow to supply two motors with voltage up to 46 V and current 2 A (per motor), providing ‘enable’ input for each channel and polarization selection. The logic supply voltage should be in range 4.5–7 V, however the voltage above 2.3 V on the inputs is guaranteed to be interpreted as ‘high’ level. We need therefore separate power supply voltage in the circuit (5 V), but the microcontroller’s outputs can be connected to the control inputs directly. The part of circuit responsible for supplying the motors is presented in Fig. 6.

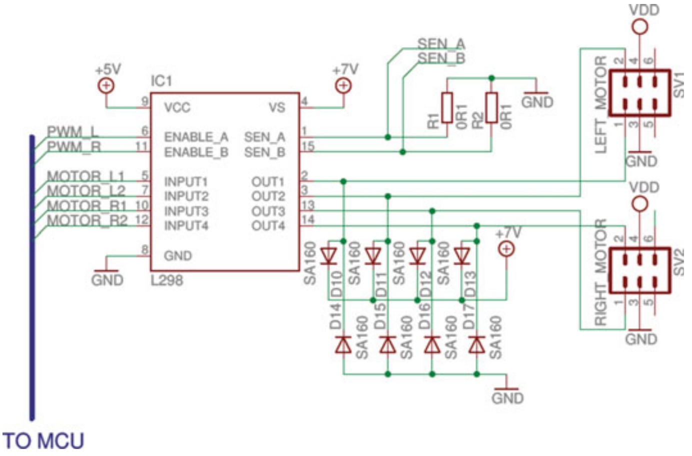


Fig. 6 The motors driver

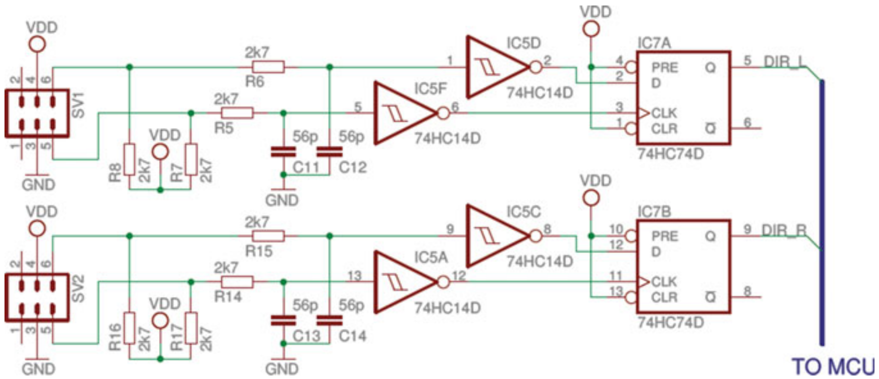


Fig. 7 The direction detection circuit

2.7.1 Encoders

On the shaft of each motor there is two-channel encoder mounted. When the wheel rotates (and the motor shaft too) we obtain a square wave on each channel, one shifted by 90° with respect to another. We can, therefore, recognize the direction of rotation by checking which one is shifted forward. The circuit which detects the direction of rotation of wheels is presented in Fig. 7.

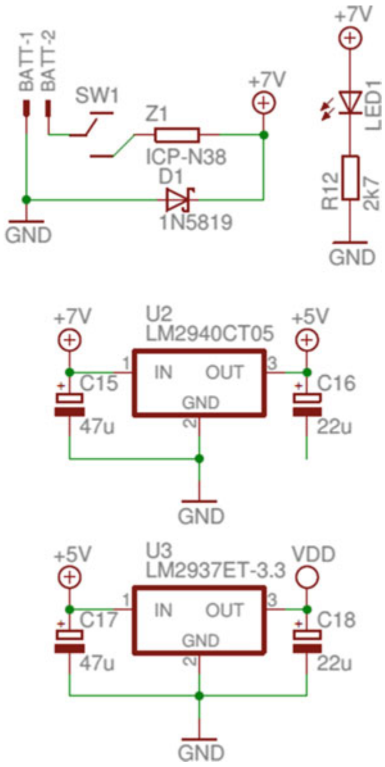
2.8 Power Supply

As it was mentioned before, two regulated power supply voltages are needed in the circuit. The first one 3.3 V will supply nearly all digital circuits excluding the logic of L298 bridge. This one will be supplied from 5 V voltage source. Moreover, the supply for power output stages of L298 should pass over as many elements as possible in order to avoid energy losses [17–19].

The robot will be supplied from 6-cell NiMH battery, so it's nominal voltage is 6 × 1.2 V = 7.2 V. The real operating voltage varies, however, depending on battery charge state, load and temperature. When the battery is fully charged, it can reach almost 9 V under a light load. When the battery is nearly fully discharged and the load is larger (e.g. 2 A), it's voltage will be approximately 6 V. Because of such spread, voltage regulators must be used.

During further development and usage it may happen, that power supply (battery) will be connected to the board by mistake with reverse polarization. I assume it will be rare case, however the circuit must be protected from such an accident. The design of power supplying part of the circuit is shown in Fig. 8.

Fig. 8 The power supply



2.9 RS232 and USB Communication

RS232 is the simplest way of communication between the MCU and PC. AT91SAM7S integrate a dedicated peripheral for this purpose (UART), therefore no external driver is needed. However, because voltage levels in PC serial port are other than in MCU, the voltage level converter must be used. There are many of integrated circuits specialized for this task available on the market. I selected the MAX3221CAE which need only connection of four small capacitors [20] (100 nF—it is important here, because they are available for surface mounting unlike some greater ones). Proper schematic part is presented in Fig. 9. The USB controller is embedded into the MCU, but some additional elements needs to be connected. Whole external circuit is described in [13]. Figure 10 shows the corresponding part of schematic.

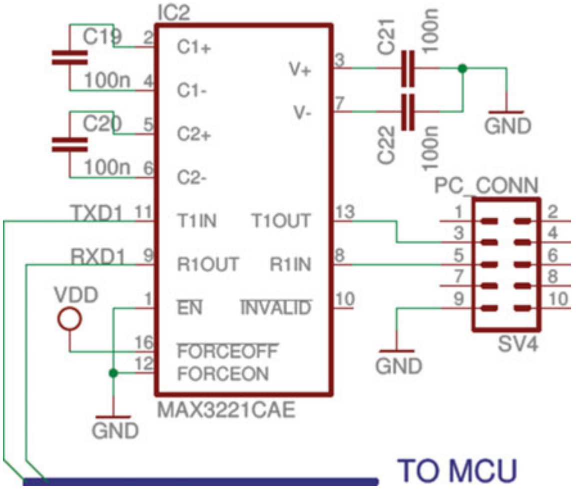


Fig. 9 The RS232 interface—voltage level converter

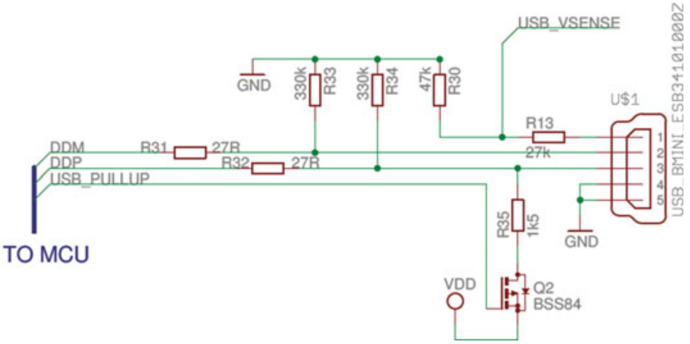


Fig. 10 The USB port connection

3 Conclusions

The final version of the controller mets all requirements that were stated. It provides a reliable platform for development of distributed control system. A common interface for all devices that allow to transmit/store data makes further development easier. The MCU that was applied provides enough computational power for implementation of nontrivial algorithms with high sampling rates. There is a possibility of reprogramming several controllers simultaneously with no need of connecting them to PC. Although the controller is by now ready to use, several things can be improved. Also some additional functionality can be added in the future [21–25]. Generally, everything on the board works properly, however the following things could be improved. When the robot is connected to USB, the logical part of the board could be supplied from it. It would be useful during development parts of controller not related to robot motion. Unused A/D converters could be available on some connectors in order to make possible adding some sensors with voltage or current output. The socket with SD/MMC card has to be attached now as an external module. Since micro-SD cards are relatively cheap currently, it would be more convenient, if there was a micro-SD socket integrated into main board. There are no general purpose LEDs that can be controlled by the MCU. It would be very useful if there were added some of them e.g. for signaling low level of battery voltage of to inform that robot is in programming mode. Three to five LEDs of different colors would be enough.

References

1. Dobroczyński, D., Dutkiewicz, P., Jedwabny, T., Kozłowski, K.,Majchrzak, J., Niwczyk, G.: Mobile robot soccer. In: Proceedings of the 6th International Symposium on Methods and Models in Automation and Robotics (MMAR), pp. 599–604, Miedzyzdroje (2000)
2. Płociński, M. Pazderski, D. Majchrzak, J.: Control system of a miniature mobile robot for multi-agent purposes, Prace Naukowe Politechniki Warszawskiej. Elektronika. z. 166, t. 1, pp. 233–242, Warszawa (2008)
3. Federation of international robot-soccer association. http://www.fira.net/
4. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley, New York (2009)
5. Ulinowicz, M., Narkiewicz, J.: Modeling and identification of actuator for flap deflection. J. Autom. Mobile Robot. Intell. Syst. **5**, 35–40 (2011)
6. Bibik, P., Gradolewski, S., Zawislak, W., Zbudniewek, J., Darakchiev, R., Krczel, J., Michalski, M., Strzelczyk, K.: Problems of detecting unauthorized satellite transmissions from the VSAT terminals. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–4, 8–9 Oct 2012. ISBN:978-1-4673-1422-0
7. Jedrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 403–408, 19–21 June 2013. ISBN:978-1-4673-6320-4
8. Kwasecki, M.: Implementation of the brain on-board system on group of mobile robots. Master's thesis, Silesian University of Technology, Gliwice, Poland (2007)
9. Jaskot, K., Babiarz, A.: New design of multiagent system. In: 10th International Carpathian Control Conference. ICCC' 2009, pp. 23–26, Zakopane, Poland (2009)
10. Babiarz, A., Bieda, R., Jaskot, K.: Vision system for group of mobile robots. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 139–156. Springer, Berlin (2013)
11. Kwiatkowski, J., Sobel, D., Jedrasiak, K.: FPGA based omnidirectional video acquisition device (OVAD). Recent Adv. Electr. Eng. Comp. Sci. 58–61 (2014). ISBN:978-1-61804-249-1
12. Kus, Z., Nawrat, A.M.: Camera head control system with a changeable gain in a proportional regulator for object tracking. In: Innovative Control Systems for Tracked Vehicle Platforms. Studies in Systems, Decisions and Control, vol. 2, pp. 105–125 (2014)
13. Atmel Inc.: AT91SAM7S series preliminary. http://www.atmel.com
14. ARM Inc.: ARM7TDMI technical reference manual. http://www.atmel.com
15. Atmel Inc.: AT91SAM7S microcontroller series schematic check list. http://www.atmel.com
16. STMicroelectronics.: L298. Full-bridge driver (product datasheet). http://www.st.com

17. Barnat, W., Dziewulski, P., Niezgoda, T., Panowicz, R.: Application of composites to impact energy absorption. Comput. Mater. Sci. **50**, 1233–1237 (2011)
18. Niezgoda, T., Barnat, W.: Analysis of protective structures made of various composite materials subjected to impact. Mater. Sci. Eng. A **483**, 705–707 (2008)
19. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical analysis of IED detonation effect on steel plate. Acta Mechanica et Automatica **6**, 10–12 (2012)
20. Maxim.: 1uA Supply-Current, True +3 V to +5.5 V RS-232 Transceivers with AutoShutdown. http://www.maximintegrated.com
21. Jedrasiak, K., Nawrat, A., Wydmanska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security. Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)
22. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Chen, S. (ed.) Proceedings of WSEAS International Conference on Mathematics and Computers in Science and Engineering, no. 9 World Scientific and Engineering Academy and Society (2009)
23. Jedrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 5337, pp. 391–399 (2009)
24. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 407–413 (2012)
25. Daniec, K., Iwaneczko, P., Jedrasiak, K., Nawrat, A.: Vision based systems for UAV applications. In: Studies in Computational Intelligence, vol. 481, pp. 219–232 (2013). ISBN:978-3-319-00368-9

Conception of a Diagnostic System for Evaluating a Technique Correctness and Effectiveness of Running

Krzysztof Skrzypczyk

Abstract The paper addresses the problem of a system design for evaluating the quality and effectiveness of human running. In this work the general conception of diagnostic system which can capture and measure some specific and key parameters of the running stride and posture is discussed. Instead of using the high resolution motion capture systems an application of relatively simple technological means are proposed. Running biomechanics improvement require changing some basic phases of motion. Extracting key elements of the overall and quite complex process of running, and further working under their improvement can result in successful transformation on running form. Moreover reducing the number of motion parameter observed allows applying relatively simple diagnostic system. In this paper an application of 2D vision system and inertial measurement units is discussed.

Keywords Running biomechanics · Inertial measurement units · Motion capture · Vision system

1 Introduction

Running has become the most popular sport activity [1]. Thousands of people all over the world are attending the street races, more and more people are run longer and longer distances during their training and preparations for the competitions. There is a common opinion which unfortunately is also promoted by media as well as by most of the running couches, that running is a natural and individual feature of each athlete. The consequence of that is the belief that running is not a sport discipline which is not "technical", what leads to marginalization of this part of the

K. Skrzypczyk (✉)
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: krzysztof.skrzypczyk@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_16

training process. The lack of the awareness of the correct running technique is the main reason of weak results even hard training amateur-runners. Moreover incorrect running form leads straight-forward to serious injuries [2]. Contemporary running science shows that running is a very sophisticated form of locomotion [2, 3]. The truth is that there exists correct running pattern and majority of amateurs as well as many of professional runners can and should undergo their running form refinement or transformation. Although the branch of running science that deals with kinetics and biomechanics of running is still in its infancy there are a lot of researches that reveal correct, evolutionary programmed form of running [4, 5]. The present day technology comes with the help to researches. Advanced stationary motion capture vision systems like Vicon [6, 7], BTS as well as portable inertial bases motion measurement devices (XSense suite) allow recording all the parameters of moving human body. On the other hand, this kind of measurement, and motion capture systems provide far too much information than is needed to evaluate the biomechanical quality of movement and identify the main imperfections that influence the running form. Moreover focusing on limited but crucial elements of the running body kinetics allows both couches and the runner itself better monitoring running form refinements. And the last issue that makes the commercially available motion capture systems does not fit well into the application discusses is their price. These systems are very expensive and hence they cannot be commonly used in small running teems and the more by private amateur users. In this paper the system concept intended for evaluating and monitoring the biomechanical features of running is discussed. In the approach proposed, simple 2D vision system is considered to be sufficient source of information about the running parameters. Aside from the system part that can be used for stationary, laboratory experiments there is the second part with the use of which it is possible to collect information while running in terrain. The discussed solution can provide information about kinematics of the moving body, the timing parameters, as well as forces that the running body interacts with the ground. Thanks to the limited information provided by this system it is possible more effectively analyze influence of refinements applied to the running technique. The system the concept of which is presented in the paper can be very useful, and widely available tool both for running technique couches and amateur runners working under refinement their running form.

2 Correct Running Form

Human evolutionary is provided with two forms of locomotion: walking and running [5]. Walking is very effective mean of our body transportation over long distances with slow speed. On the other hand nature equipped human with running which allows him moving with much higher speed and also over long distances. Human is the only representative of mammals that can run over so long distances like marathon distance for instance. Nowadays people prefers sedentary lifestyle, wear shoes from its infancy what is the reason the primary pattern of running is lost over the years. Following the contemporary running boom statistical amateur tries to adapt the walking gait to the running, just by simple replacing the double support phase of walking cycle with the flying phase. Without correct body posture this leads to applying so called rear-foot-strike pattern which in turn results in many irregularities and dangerous effects of this running pattern. This is the main reason of running ineffectively and what is more dangerous is the reason of majority of injuries. This is due to the fact that biomechanical mechanism of walking are not adapted to withstand overloads that appear during incorrect running [8]. Of course detailed analysis of biomechanics of running is beyond the scope of this study. Running is a very complex form of motion that brings together a number of body functions. Detailed analysis and discussion on running technique and the methods of its improvement can be found in [3–5]. Although running is a very sophisticated motion kind, there can be distinguished a few key issues that dominate over others. Improving them results in better use of the locomotion mechanisms, increasing the effectiveness of running what in consequence leads to increasing the running speed. For the further discussion that will appear in this paper let us point out this issues.

- The first one and the most important is a hip position during the run. The hip should be located on the line that connect the ear and the ankle in the stance phase of running.
- The second one is a slight inclination of the body during the whole run. However these two features are connected together. Positioning the hip in the manner described above is possible only together with satisfying the second issue. Such pose gives a runner gravity acceleration improves much the running effectiveness.
- Another motion feature that influence the running form is an amplitude of vertical oscillations of the body while running, which should be minimized.

The aim of the system discussed is detecting, measuring and monitoring the aforementioned key issues of the running aspects.

3 System Conception

A diagram presenting the idea of the diagnostic system discussed in the paper is shown in Fig. 1. The subject of an evaluation is a runner (an athlete or amateur) running in its individual manner. So the individual moving pattern is captured by the system and undergo further analysis. The first part of the system discussed is a measurement system that provides data about the runner motion parameters. These data are compared to some knowledge about correct values of the parameters monitored. The knowledge could has a form of expert knowledge of a couch as well as can be synthesized into the form of performance and quality indices. The result of comparing the individual running form to some correct pattern gives clues and recommendations to the refinements related to the training program.

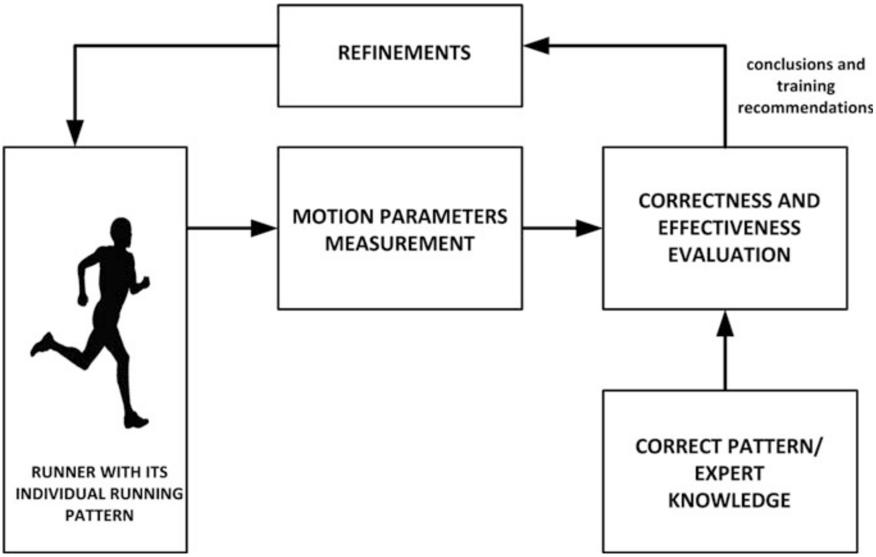

REFINEMENTS
conclusions and training recommendations
RUNNER WITH ITS INDIVIDUAL RUNNING PATTERN
MOTION PARAMETERS MEASUREMENT
CORRECTNESS AND EFFECTIVENESS EVALUATION
CORRECT PATTERN/ EXPERT KNOWLEDGE

Fig. 1 Diagram that illustrates an idea of the running diagnosis system

There are three kinds of measurements that are collected by the system for evaluating the effectiveness of running (Fig. 2). The first group of data collected is related to geometry of the running body. The effectiveness of running depends much on some key geometrical body posture elements. The most important features this part of the system is intended to track are: hip location, inclination of the body and the angle of support phase, which is defined by ground-line and ankle-knee line. Tracking the hip location it os also possible to determine vertical oscillations

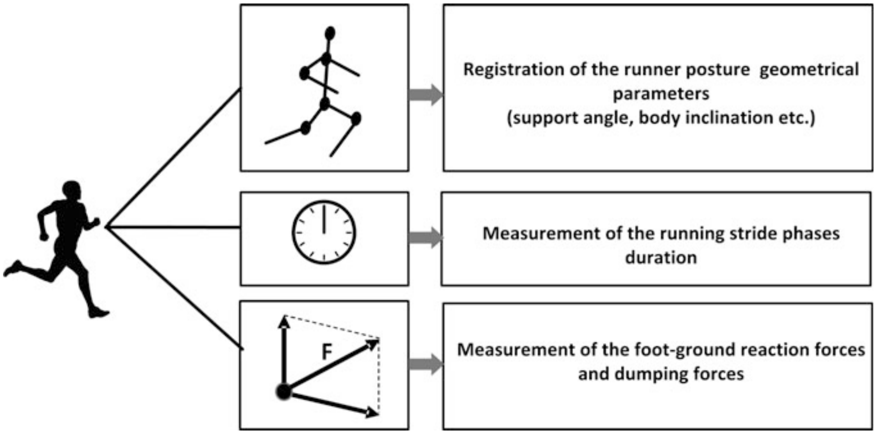

Registration of the runner posture geometrical parameters (support angle, body inclination etc.)
Measurement of the running stride phases duration
F
Measurement of the foot-ground reaction forces and dumping forces

Fig. 2 Three groups of motion parameters analyzed by the system discussed

of the body. The second group consists of timing parameters, that evidence for the quality of the motion examined. Among a number of timing parameters it can be distinguished the key ones like: cadence of the running stride and duration of the support phase. Third set of parameters is related to the forces of reaction the running body interacts with the ground. Using contemporary technologies like miniature accelerometers it is possible to estimate the forces of reaction. Using these date we can infer a number of useful issues for the running refinement process. It is important to emphasize that for the purpose of both training and refinement process is not essential to measure aforementioned motion parameters with a high accuracy. The analysis is focused on observing the crucial elements of a body motion in terms of finding some trends. It is enough to state that some value which undergo the optimization process is increasing or decreasing. This assumption allows further simplification of the system.

3.1 Stationary Part

The overall system can be considered two fold. The first part is assumed to be as a tool for the couch for detailed analysis of the running form and the refinement process progression in the laboratory. The stationary part can be divided into:

- A vision system for monitoring a runner posture [9]
- Accelerometers mounted intentionally by the couch in some key points of the runner body

The function of this part of the system is collecting all the necessary information about the running form of the athlete and further, using it for analysis. In the next part of this section we discuss the application of different technologies.

3.1.1 Vision System

As was mentioned before nowadays vision systems are the main tool for precise, high speed motion capturing in 3D space. Commercially available systems like Vicon, BTS are stationary, stereoscopic, very accurate and robust systems that meet requirements even very demanding client. From the perspective of the application discussed these systems seem too be highly redundant. Moreover the cost of this system is also very high. These issues can be a reason for attempting to simplify the vision system. The system proposed provides all the information which is necessary for evaluating the running form and providing the feedback for the technique refinement. Although the run is a quite complex form of locomotion and its effectiveness is influenced by many factors, there are few crucial ones. The most important and dominant one is the hip location and the inclination of body during the stance phase which in combination with a few others provides sufficient data for the technique refinement. In Fig. 3 a conception of the stationary part of the system

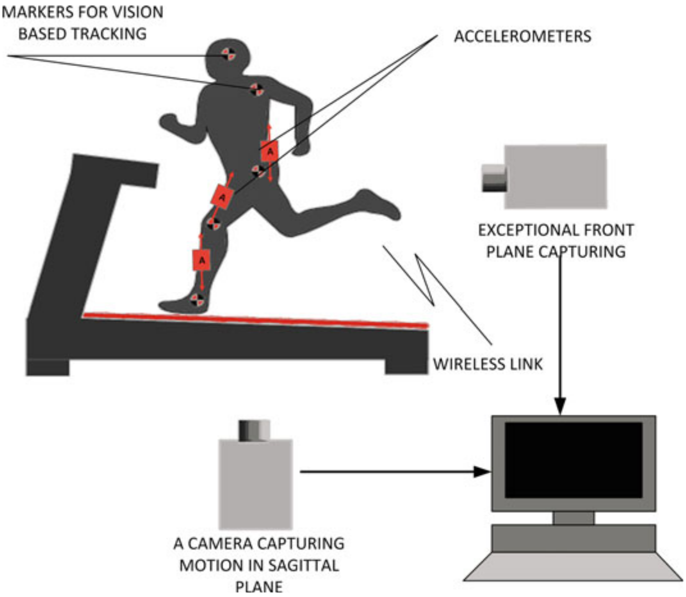


Fig. 3 Combination of simple 2D vision based motion capture system with inertial measurements

is presented. The system consists of the vision based part for tracking 5 markers located in the key points of the body: head, shoulder, hip, knee and ankle. Tracking mutual relations between points determined by these markers is enough to draw conclusions about the essential posture features. Such task is rather simple problem for image processing and can be done using simple camera. Demands for the marker tracking speed are not also high in the case of this system. The cadence of an elite runner is about 180 strides per second what gives the value about 90 strides for the single leg. If we assume that it is possible process the image with the speed 25 frames/s we obtain the tracking resolution on the level of 16 frames per single stride. It is absolutely sufficient for extracting the features necessary to analyze the motion quality.

3.1.2 Inertial System

Thanks to the great technological development in electronics (especially the M.E. M.S. technology) it comes possible to manufacture miniature high accuracy inertial sensors like accelerometers and gyroscopes. These devices can measure dynamic motion parameters like linear acceleration and rotary speed. Combining them with sophisticated filtering algorithms [] reliable and accurate measurements can be obtained. Applying these sensors to running dynamics analysis opens quite new possibilities. In Fig. 4 two proposed applications of inertial sensors are shown. Human locomotion apparatus is designed well for shock absorbing. During the

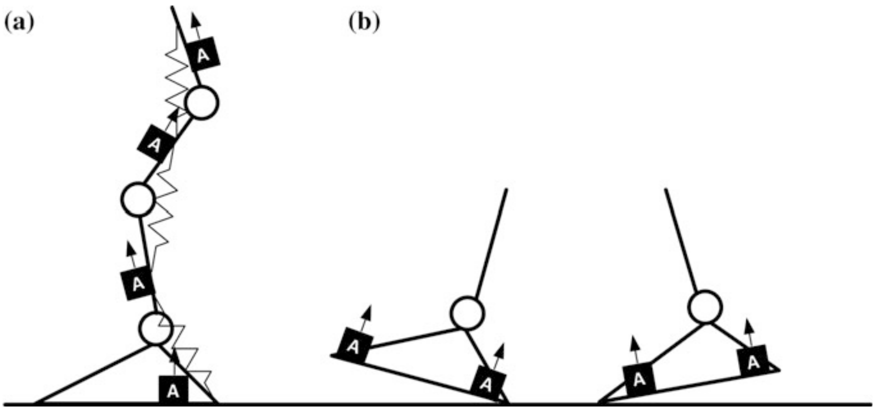


Fig. 4 Two exemplary applications of accelerometers. Shock absorption evaluation (**a**) and foot-ground interaction forces (**b**)

landing and further the stance phase muscles and tendons fixed to the leg segments works like high quality suspension system. If running technique is correct this mechanism works properly dumping the ground reaction forces applied to the skeleton. Otherwise, while running in biomechanically incorrect form, the forces acting on our body are very large what usually leads to injuries of joints, tendons and even bones. In Fig. 4a the human body suspension system is schematically presented. With the spring symbol the dominant shock absorption mechanisms are marked. Using accelerometers fixed to subsequent segments of our body it is possible estimating the forces applied. It is not important to know the exact value of the given force. It is enough to observe if these forces are increasing or decreasing while applying some modifications and refinements. Moreover mounting 4 accelerometers in the way presented in Fig. 4a the shock absorption propagation can be observed. Refinements in running technique should result in observing in negative propagation of forces acting on subsequent joints.

The next application of inertial sensors to running technique evaluation is related to measurement the foot interaction with the ground. Mounting accelerometers on different foot areas it is possible to find out the stride character. In general three patterns can be distinguished. Fore foot pattern (FFP), middle foot pattern (MFP) and rear foot pattern (RFP). Refining a running stride, first of all, should be focused on eliminating the RFP if such stride is used. Figure 4b shows an application of two accelerometers fixed to the front and rear area of the foot. Comparing foot-ground reaction forces related to the front and rear area we can assess the nature of foot ground interaction. Moreover in this configuration it is possible to find timing characteristics like duration of the stance phase.

3.2 Mobile Part

Other issue of the conception discussed is related to the mobility [10]. While the stationary part of the system is intended to be a diagnostic tool for couch working under a runner technique refinement, the mobile part is dedicated rather for personal use. Of course it also can be used in the overall coaching process like a source of additional information. The functionality of the mobile part of the system is very limited comparing to the stationary one and provides only the basic information of the running quality that can be understood and analyzed by the user without biomechanics background. The parameter that aggregates all the factors influencing the running effectiveness is vertical body oscillation. It easy to be focused on minimizing this oscillations as a result of running style modifications. Reducing this value is an evidence that the athlete is on the right way to improving his technique. Another important issue is a cadence of run which for best performance should be kept in the range of 180–190 strides/min. Moreover it is possible to monitor the body rotations and inclination in various planes for better optimizing the motion. The functions pointed out can be accomplished by a single inertial measurement unit (IMU). By analyzing the acceleration time plot, measured in different motion planes, we can extract information about cadence and a duration of the support phase. Double integrating the acceleration in combination with filtering procedures it is possible to find the body displacement in a considered direction hence the amplitude of vertical oscillations can be found. This consideration has very general form but it is only to highlight potential possibilities and applications of nowadays commonly used sensors.

4 Conclusion

In this paper the problem of the system design for evaluating the quality and effectiveness of human running was discussed. A general conception of diagnostic system based on the vision motion capture system and inertial measurements was presented. The key issue which is the advantage of the system proposed id its simplicity. This is with accordance with the assumption that the system should be widely accessible which is contrary to the high precision commercially available motion capture and analysis systems. An application of relatively simple means of technology for tracking and monitoring the basic body motion parameters was discussed. The main idea of this system is focusing on limited but the important motion parameters influencing the running technique. This is the reason that the most important part of the system (except of creating the measurement set up) is the analysis module. This module should support the couch giving him some aggregated data and performances indices which help him to monitor the progression in the running refinement process. Although only the concept of the system was discussed this work shows possible application field of simple, widely available technological means. These technologies in combination with profound knowledge of running biomechanics can give significant effects in running form refinement process.

References

1. Fixx, J.: The Complete Book of Running. Random House, New York (1977)
2. Anderson, O.: Running Science. Human Kinektics, Champaign (2013)
3. Martin, B.: Running Technique, Brian Martin, 2011 (e-book)
4. Abshire, D., Metzler, B.: Natural Running: The Simple Path to Stronger, Healthier Running. Velo Press, Boulder (2009). ISBN 978-1934030653
5. Romanov, N.: Dr. Nicholas Romanov's Pose Method of Running. Pose Tech Press, Coral Gables (2004). ISBN 0972553762
6. Switonski, A., Josinski, H., Jedrasiak, K., Polanski, A., Wojciechowski, K.: Classification of Poses and Movement Phases. Comput. Vis. Graph. **6374**, 193–200 (2010). (Lecture Notes in Computer Science)
7. Sobel, D., Kwiatkowski, J., Ryt, A., Domzal, M., Jedrasiak, K., Janik, l, Nawrat, A.: Range of motion measurements using motion capture data and augmented reality visualization. Comput. Vis. Graph. **8671**, 594–601 (2014)
8. J. Puleo, P. Milroy: Running Anatomy, Human Kinetics, 2009 ISBN: 978-0736082303
9. Kus, Z., Nawrat, A.M.: Camera head control system with a changeable gain in a proportional regulator for object tracking. Innovative Control Syst. Tracked Veh. Platforms **2**, 105–125 (2014). (Studies in Systems, Decisions and Control)
10. Wochlik, I., Bulka, J., Folwarczny, Ł., Daniec, K., Jedrasiak, K., Koteras R., Nawrat A.: Application of telemedical Technologies in Remote Evaluation of Soldiers' Vital Signs During Training and in Combat Conditions. Innovative Control Systems for Tracked Vehicle Platforms, pp. 189–202 (2014)

Part III
Design and Evaluation of Control Algorithms

Human perception is limited to tracking in full concentration only seven elements simultaneously. Control of unmanned vehicle and analysis of data recorded by the payload of unmanned vehicles is a significant excess of information for human and significantly limits its analytical capabilities. It is therefore necessary to develop new control algorithms enabling the partial or total replacement of certain human operations in order to enable a man to focus on the analysis of the obtained data from the payload. Many research groups around the world work on design and evaluation of control systems for unmanned vehicles.

The part presents the use of the modified BLT method for the synthesis of control system of the UAV. A small helicopter was an example of the UAV. This method was used for controlling linear acceleration in the vertical axis and angular velocities around the UAV's symmetry axes. The linearized multivariable model was used as model of the UAV. The presented examples illustrate correct operation of the modified BLT method in the lowest level of the UAV's control system.

Due to the popularity and usability of the BLT method for tuning multivariable control systems, we proposed a certain modification. The BLT method uses the Ziegler-Nichols (ZN) method for tuning PI regulators for the main channels. The limitations caused by the ZN method for the plant's transfer function form resulted in limiting the use of the BLT method. Therefore this paper put forwards the modification based on using the tuning methods different from ZN. This paper shows the examples of using different tuning methods than ZN which were adjusted to the form of the plant's transfer function. Such an approach allowed to extend the range of the acceptable kinds of plant's transfer functions in the main channels during synthesis of the multivariable control system. It allowed to use the knowledge of single-variable control systems for tuning multivariable control systems.

The part presents research devoted to modeling and control system design for a quadrotor UAV. The quadrotor lift, propulsion, and control loads are produced by four identical rotors, performance of which strongly depends on flight conditions. The quadrotor angular rates result in non identical flow through each of the rotors due to their non coaxial configuration which may cause disturbances of a flight path. The development of a complex dynamic model and a control law enhancement method for a quadrotor is presented in this paper. The novel approach to control system development led to increase in quadrotor handling precision and compensates disturbances of the flight path, which make a quadrotor more capable to operate in a confined space.

An important determinant of the quality control is the quality of the input data to the system. It is therefore necessary to develop sophisticated filtering algorithms which are characterized by the highest precision possible. The part presents different ways of processing data acquired for the purpose of determining the orientation of an object in space and navigation of unmanned vehicles in confined spaces.

Naive Kalman filter is introduced in the part and presented for estimating orientation in 3D space. Using the assumption of Bayesian classification systems, the angular velocity vector is treated as three separate events. Therefore, three independent Kalman filters are used to estimate Euler angles for each RPY coordinate system. Data fusion is presented for real IMU sensor which integrated data from triaxial gyroscope, accelerometer and magnetometer.

The part described visual odometry algorithm for monocular camera. It is developed for use on a mobile robot involved in criminal investigation conducted by Polish police forces. FAST keypoints with ORB descriptors are used as input to the motion estimation. Concept of pose graph is implemented in order to improve accuracy of the results obtained. The algorithm was evaluated and results are presented together with the guidance for further development of the method.

Experimental Mobile Robot—Software

Krzysztof Jaskot and Tomasz Łakota

Abstract The paper presents the software part of the experimental mobile robot. Whole software for mobile robot was divided into functional blocks and each of them is described separately. For each, the corresponding part of schematic was presented and explained. The most important decisions on selected methods are discussed and justified.

Keywords Mobile robots · Tasks · Scheduler · Command processor

1 Introduction

Mobile mini-robots constitute an important equipment for research of control algorithms. During last decade, many constructions for this purpose were developed. One of examples is the series of Kephera robots, which are widely used at many universities in robotics courses [1]. It is commercial product, with emphasis put on high modularity. Very different approach is presented in the work [2, 3], where the cheap, compact mobile robot is described. There are also many constructions intended to be used in robot soccer competitions [4]. One of such models is 188 Soccer Robot produced by Microrobot. In this work, a new software for controller for this robot is presented.

In work [3] describes a cheap controller, utilizing more modern parts. The Philips LPC2103 based on ARM7TDMI-S core was used as the main processor, and CC1000 chip was used as a radio module. The role of the processor is limited however to realize a PID controller for wheels' speed, to estimate current position using odometry and to exchange the data with base station. In such approach nearly all control tasks (except setting wheels' speed to desired value) are done by the master system and only desired speed of wheels are sent to the robot [5–7]. The

K. Jaskot (✉) · T. Łakota
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: krzysztof.jaskot@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_17

implementation of reactive algorithms, basing on data from odometry would be difficult, since transmitting the data from robot to master system and then, the command in reverse direction introduce a significant delay. Moreover, the throughput is limited similarly as in previous case.

The software for the controller is written in such a way that it can be extended and modified relatively easily. In order to make it's further development effective, several communication channels are available for use in a simple manner by the developer, so additional data can be exchanged with the controller independently of wireless link. There is also a possibility of connecting the socket with SD/MMC flash card and to use it in a software without great effort. The powerful microcontroller is able to realize some control algorithms locally on the robot, so the master system can focus on more challenging tasks [8, 9]. The reactive algorithms based on odometry is realized by the on-board system, while the planning part and the algorithms taking input data from vision system are assumed to be implemented in the master system [10–12]. The controller should also make effort to avoid sliding, so when desired speed changes stepwisely, it changes the real speed smoothly.

2 Software

In the controller, there is much of work to be done by the MCU. Starting from low level devices' drivers (battery monitoring, encoders reading, motor driving), I/O devices (RS232, USB communication, SD/MMC card interfacing) going through low level controllers (motor speed PID controller), various utilities (internal robot state logging, firmware upgrade), finishing on high level controllers. It is obvious, that the program which realizes all these tasks simultaneously will not be very simple. A great effort was made in order to produce a high quality code, which besides of it's complexity is stable, easy to extend or modify and clear for the reader. It is critical, since the aim was to make a platform for further development and being a subject of many modifications and experiments. The effect of work on software for the controller is presented in this work.

2.1 MCU Tasks

In order to design the software properly we need to define a set of separate tasks which need to be done. The primary tasks are as follows:

- Collect data from encoders.
- Control a speed of each motor.
- Estimate position and direction of the robot using odometry.
- Combine a data from odometry with frames containing data from positioning system based on camera.

- Control acceleration not limiting it's too much, but avoiding skidding and overacceleration.
- Run control algorithms (turn by angle, move by distance).
- Receive commands and positioning data using wireless link.
- Interpret, queue, and execute received commands.
- Monitor battery voltage level and turn system into suspended mode if voltage is too low.

There are also some other tasks that are not key features of the controller, but very useful during development.

- Collect and/or send some internal values which are helpful for development.
- Communicate with PC over serial port (RS232) and USB.
- Interface with connected SD/MMC card (can be used as a storage for collected data).
- Implement some user interface for debugging purposes.
- Upgrade software using wireless connection.

2.2 Design Overview

One of the features of high quality software is a modular architecture. In such an approach, the code is divided into many parts (modules) aiming at minimize dependencies between them. Each module should be responsible for a small, precisely defined concept.

In the previous section many task were listed. Most of them are spread in time and need to be executed in parallel with other. For this reason, some kind of scheduler is needed.

Another important aspect is interface for Input/Output devices, i.e. devices which allows to send or receive some data. Generally two kinds of data will be transmitted:

1. Binary packets like commands or pages of flash image (program) while software update. Such data should be protected against possible errors.
2. Stream of text used in interaction with developer. The error checking is not necessary here.

We have several devices, which could be used for these purposes:

- serial connection with PC (RS232),
- USB interface,
- wireless link [13],
- SD/MMC card.

Generally it is stated, that finally the wireless will be used for the first type of data. It would be nice feature, tough, to be able to select devices used for different purposes freely. Such approach allows to communicate through wireless link in debug mode during development or send commands via serial link during testing. It makes the software more flexible and eliminates some limitations. In order to achieve this, the common I/O interface was designed. Then, drivers implementing this interface were written for each available device.

Modules and dependencies between them are presented in Figs. 1, 2 and 3.

2.3 Scheduler

When many tasks need to be realized by one CPU, the need of scheduling arises. According to [14–16], ‘Scheduling refers to the way processes are assigned to run on the available CPU, since there are typically many more processes running than

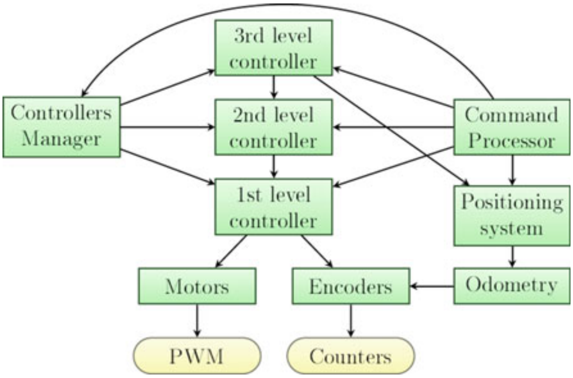


Fig. 1 Modules of the software—the main part

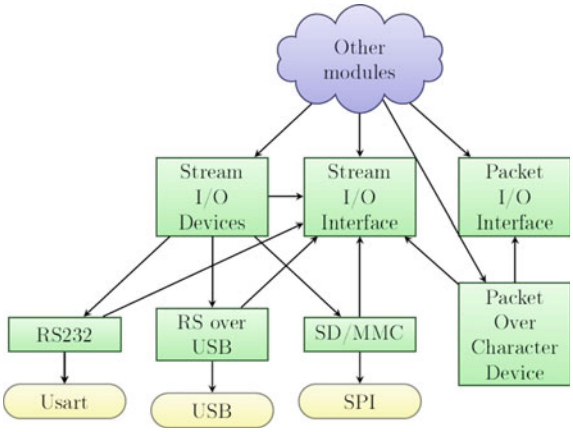


Fig. 2 Modules of the software—the I/O subsystem

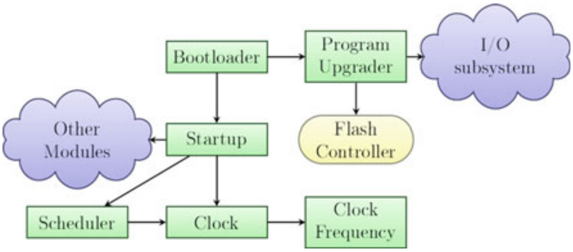


Fig. 3 Modules of the software—system utilities

there are available CPUs. This assignment is carried out by softwares known as a scheduler and dispatcher'. Scheduling is common problem in information processing [17, 18]. Many algorithms that allow to manage CPU time effectively according to selected criteria were developed [19].

Schedulers can be divided into two general categories [19]: preemptive and nonpreemptive. The preemptive schedulers are allowed to interrupt process in the middle of execution, and allow another one to use the CPU for some time. In such a way, all processes are interleaving for small time quantum giving impression of concurrency. When non-preemptive scheduling is used, there is no possibility to interrupt process until it return control to the system. Both of them has advantages and drawbacks. In the preemptive case it is always needed to switch context (save registers for one process, restore for the next) which introduces some overheat that can be avoided in the non-preemptive scheduler. Moreover, a separate, fixed-size stack is needed for each process, which is serious drawback in case of MCU (relatively small amount of RAM, no virtual memory).

2.3.1 The Solution

In presented work an intermediate solution is used. Each process is divided into series of short jobs, which are the subject of scheduling, and a priority level from range 0–7 is assigned to each of them. The scheduler is not preemptive in the meaning described before, but the job with priority greater than priority of the current one can interrupt it for time needed to complete it's work and then the interrupted one is continued. Such approach gives us several benefits:

- low response time for critical tasks (with high priority),
- small overheat—no context switching,
- no need of separate stack for each process,
- interprocess communication is simplified when the same priority level is assigned to all of them,
- simplicity of implementation

In practice, each process is represented by the function called periodically in defined time steps, and with given priority. After the job is finished, the function

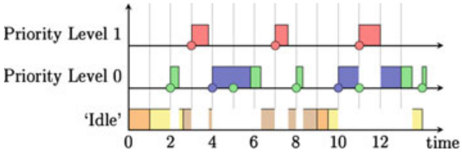


Fig. 4 A Gantt chart for exemplary tasks

returns and scheduler continues it's work. Within the same level, the FCFS (First-Come-First-Served [19]) algorithm is used. Additionally, there is a virtual priority level called 'idle'. Jobs with this level are called in the main loop of the program (so not in interrupt handlers) according to Round-Robin policy when no other jobs are running. It is considered as the lowest level so the least time critical processing can be done here. In Fig. 4 there is presented an example for set of tasks listed in Table 1. Small circles denotes the arrival time of jobs (which is determined by the period). It can be observed, that job from the lower level can be interrupted by the higher level, but when the job from the same level as the currently executed arrives, it has to wait.

There are no perfect solutions, however. One of drawbacks of this approach is necessity of division of task into parts (referred as 'jobs'). Moreover, the afford must be paid to make the jobs as short as possible in order to achieve good general responsiveness of the system. In some cases, when critical task has a lot of work, it can be split into two periodic functions with different priority levels, and the time consuming computations may be performed in the one with lower priority. In addition, when the same part of memory is used by tasks on different priority levels, some problem arises. Usually in such cases, data is locked (e.g. using semaphores) in order to prevent it's modification while it is processed. However when there is some shared data which is locked by current job, and this job is interrupted by another (higher priority) one that need access to the same data, nothing can be done

Table 1 Exemplary set of tasks

Task	Period	Level
1	3	0
2	6	0
3	4	1
4	-	Idle
5	-	Idle
6	-	Idle

by the high priority job, since data would not be unlocked until it return the control. Returning the control when job is not finished (hoping it will be done next time) is usually not good, especially in high-priority tasks. Therefore, instead of only locking the data, the process should prevent being interrupted by any other during operations of shared data. However the time when job cannot be interrupted should be as short as possible in order to achieve good responsiveness.

It should be mentioned, that such solution, which execute periodically short jobs is consistent with nature of most tasks considered in this work, since they are usually not one big task which has to be done from beginning to the end, but it's execution should be rather distributed in time. For example pulses from encoders are counted automatically (by the hardware) and the position should be corrected periodically. Also controllers works with some sampling frequency and have nothing to do in the time between successive samples.

2.3.2 Implementation

Presented mechanism is closely related to the architecture of MCU, which allows to configure interrupts with 7 different priority levels. When the interrupt handler is executed it can be interrupted only by interrupt with greater priority. During initialization of this module, the callback is installed in the system clock module. The callback is executed in the highest priority clock interrupt handler with frequency 8 kHz, so the resolution of period for scheduled jobs is 0.125 ms. The algorithm realized in the callback is presented in Fig. 5. In practice this algorithm interrupt current program (unless an interrupt with highest priority is served) and executes jobs in order from the highest priority to the one greater than the priority level of interrupted job. Then, it is continued and after finish, the rest of jobs down to the lowest level are executed. When no other jobs are waiting, the ones with 'idle' priority are being executed in the loop. Note that even if some jobs last longer than the period of scheduler (0.125 ms) it will work properly assuring proper order of jobs execution.

2.4 Device Drivers

Several modules were written in order to isolate the code which interacts with hardware. These modules are described shortly in this section.

- **Motors**—provides functions that sets given filling on PWM channels connected to motors and makes them rotating in selected directions. It allow also to turn of the motors (idle gear) in order to minimize current drawn from battery.
- **Encoders**—reads value of counters connected to encoders and current direction of wheels. Then, total distance traveled by each wheel is computed, and current

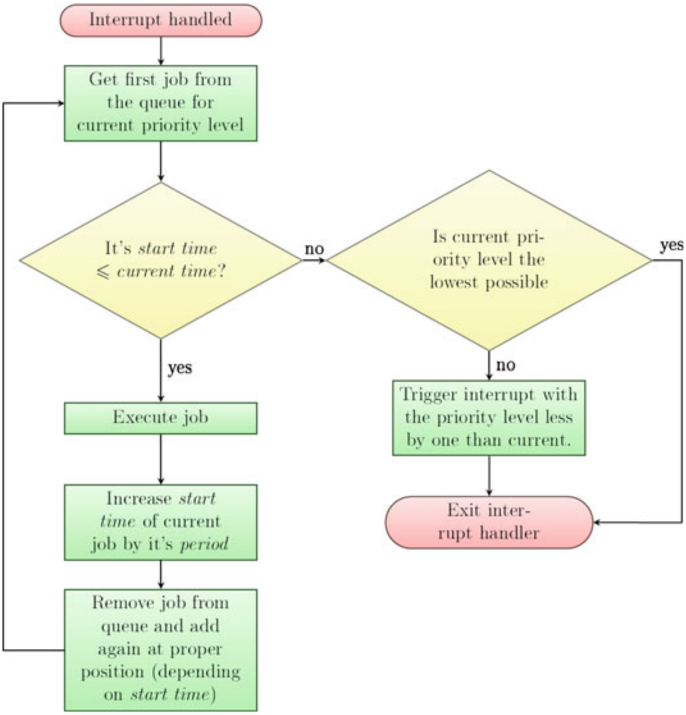

Interrupt handled
Get first job from the queue for current priority level
It's *start time* ⩽ *current time*?
no
Is current priority level the lowest possible
yes
yes
Execute job
no
Trigger interrupt with the priority level less by one than current.
Increase *start time* of current job by it's *period*
Exit interrupt handler
Remove job from queue and add again at proper position (depending on *start time*)

Fig. 5 A block diagram of the scheduler interrupt handler. Note, that if interrupt was triggered, the handler will be called again after exit

speed is estimated. Computed values can be got using proper functions from this module.

- **Battery**—perform voltage measurements using ADC from channel connected to battery monitoring circuit and computes current battery voltage level. If it is critically low, the motors are turned of, and the main processor is halted. The last measured voltage can be read at any moment of time.
- **Hardware**—some other routines such as: processor restart, getting the robot unique id [20–22].

2.5 Command Processor

The command processor module is responsible for receiving packets, parsing them, enqueueing commands and their further execution. This module utilizes the POCD packet device on a wireless transmission channel, so packets are checked against transmission errors, but it is not guaranteed that all packets will be received. Since currently only simple broadcasting mode of radio transmission is used, each robot receives also packets addressed to other ones. The ID of the target robot is contained in the packet, and these addressed to the others are discarded by the software. Also a sequential number is sent with the commands making possible to detect whether some of them was lost. Most of commands takes some time to execute (e.g. turning by given angle), so it may be useful to queue several commands on a robot making them to execute sequentially. There are three modes of queuing:

1. **No queuing**—each newly arrived command breaks current execution.
2. **Basic queuing**—each command is added to queue and executed when it's turn comes. If command is accidentally lost during transmission, it is simply ignored.
3. **Strict queuing**—like basic queuing, but if some command is missing no further one will be executed (until the queue is cleared).

The mode of queuing can be changed with each command using special field of the packet. Each mode has some area of application. If decision about robot action, in the master system, changes constantly in time, we have no reason to queue some actions for future, and the first mode is the best choice. When we want the robot to follow a path given by densely located points, it is not critical when some of them is missing, so the second mode will be sufficient. Eventually, if we send to the robot a sequence of commands like: go 1 m forward, turn 90° left, go 0.5 m forward and so on, it will make no sense to continue execution if one of commands is lost. Some group of packets does not request any robot motion and are executed immediately without queuing. Examples of such commands are status request, reset or vision positioning system data (it is not strictly a command, but is treated in the same way).

2.6 Controllers

The part of code, which realizes control algorithms is composed of several modules. The following list contain only short comment on the role of each module:

- **1st level controllers**—implements a PID regulator which controls the speed of each wheel.
- **2nd level controllers**—limits acceleration preventing sliding of the robot. Allows to set desired speed of each wheel separately as well as linear and angular speeds or linear speed and turning radius of the robot.
- **3rd level controllers**—contain implementation of 'move by distance', 'turn by angle', 'look at', and 'go to point' algorithms.
- **Odometry**—estimates robot's position basing on increments of distance traveled by each wheel (used Encoders module).

- **World**—provides information about robot's position basing both on odometry and data from vision positioning system.
- **Controllers manager**—enables or disables activity of different level controllers, synchronizes periodic functions of other modules.

Most of modules mentioned above contain a function which process some data and updates the state or outputs (e.g. read current measured speed of wheel and update filling of PWM controlling the motor). Such a function is periodically called, but it is not registered directly as a job in the scheduler. Since data produced by some modules is used then by the other (e.g. wheel speed controller needs a current speed which is obtained from other module), the order of calling them is important. For this reason, there is a common job, which calls periodic functions of particular modules.

3 Developer Console

For testing and development purposes there is implemented a simple engine that allow to interact with developer using communication channels (RS232, USB, radio module). The console is organized in a menu system, that is accessible immediately after connecting to robot using serial console software (e.g. 'Hyper Terminal' for Windows or 'gtkterm' for Linux system). Currently the following menus are available:

- Main menu (mainmenu) that provides access to other sub-menus. Also some basic information about the system is available here (scheduler and I/O devices statistics, current robot ID).
- PID menu (menu_pid) that allow testing PID regulator for wheels speed controller, changing parameters and so on.
- Speed control menu (menu_control) related to acceleration-limited speed control.
- 'Turn by' and 'move by' menu (menu_rotate_move) that allow to test these algorithms.
- 'Go to point' menu (menu_xy) that allow to test this algorithm.
- I/O devices testing menu (menu_io) used during I/O subsystem tests.
- SD card menu (menu_io).
- ADC menu (menu_adc) which main function is checking current input (battery) voltage.

If we want to use the console through USB channel under Linux system, the following command can be run:

```
gtkterm -p/dev/ttyUSB0
```

Where /dev/ttyUSB0 is a serial device under which the robot appear. If RS232 was used, additionally baudrate should be set to ‘115200 bps’, parity to ‘none’ and stop bits to ‘2’. After pressing enter the following text should appear in the window:

```
Main Menu
p) motors′ PID menu (1., level)
c) speed/turning, acceleration control (2. level)
g) turn_to_angle, go_distance (3. level)
y) goto_xy (3. level)
\) tasks′ statistics
|) buffered io statistics
i) io tests
s) SD card menu
d) ADC menu
0) stop both motors
f) flush all file buffers (sync)
o) PIO registers
n) interrupts info
l/L) turn on/off logging
X) shut down the system
I) robot ID
q) enable radioproxy
r) print radioproxy stats
```

This is a list of keys that have some function assigned. For example pressing *d* will result in opening a following menu:

```
ADC Menu
a) start conversion (motors)
b) read battery voltage
m) left motor current
```

Then, after pressing *b*, current battery level will be reported:

```
battery voltage: 7357 mV
```

The engine is designed in such a way, that adding new menus is relatively simple. The first thing that should be done is adding a function that will handle action on this menu. This should be a function that takes two arguments. The first one is pointer to *MFile* structure, corresponding to the I/O device that the user is currently using (in our example—RS over USB). When the menu is opened, or some key is pressed, the function is called with proper value of the second parameter. In the first case it is equal to -1. In the second—it is a code of the pressed key. For example the code for menu_adc is as follows:

```
void menu_adc(const MFile *f, int ch)
{
     switch(ch)
     {
     case ’a’:
       (...)
       break;
     case ’b’:
       (...)
       break;
     case ’m’:
       (...)
       break;
     default:
       mputs(f,
              "ADC Menu\n"
              "a) start conversion (motors)\n"
              "b) read battery voltage\n"
              "m) left motor current\n"
            );
     }
}
```

The second thing is adding a point where the menu can be opened. It is done using menu_push function, for example:

```
void main_menu(const MFile *f, int ch)
{
        switch(ch)
        {
            (...)
            case ’d’:
            menu_push(f, menu_adc);
            break;
            (...)
        }
}
```

The menu is ready now to be used. Of course ‘menu function’ can call any other function, use global variables, and so on.

Although presented menu system would not be very convenient in case of final user interface, it is very good solution for developer usage, since in this case flexibility and convenience of adding and modifying menus is crucial.

4 Conclusions

Most commonly used actions are already implemented, and proved to work properly, fully taking the advantage of the robot's capabilities [23–26]. Moreover, there is a reserve of computational power that allows implementing some other, more specialized algorithms on board. The software is modular, so individual parts can be modified or replaced easily, as well as new ones that extend functionality can be added. The MCU that was applied provides enough computational power for implementation of nontrivial algorithms with high sampling rates. Generally, everything on the board works properly, however the following things could be improved. Both MCU and radio module support some sleep modes. They are not used currently, but if they were, the power consumption may be reduced. The XBee module is used in a transparent mode now, what means that it simply broadcast every byte to all other modules in the range. There is available, however, a powerful 'API mode' which support packet transmission, retransmissions, addressing or even more complex features like mesh. Some research may be done in this field, and in effect transmission can be improved. The watchdog in MCU is disabled by now. It is not critical, but it may be considered whether to use it.

References

1. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley, New York (2009)
2. Kowalczyk, W., Kozłowski, K.: Co-ordinated motion control for formation of mobile robots. In: Proceedings of the 9th International Conference on Walking Robots, pp. 672–877, Brussels (2006)
3. Płocin'ski, M., Pazderski, D., Majchrzak, J.: Control system of a miniature mobile robot for multi-agent purposes. In: Prace Naukowe Politechniki Warszawskiej. Elektronika. z. 166, t. 1, pp. 233–242, Warszawa (2008)
4. Federation of International Robot-Soccer Association, http://www.fira.net/
5. Nawrat, A., Jedrasiak, K., Daniec, K., Koteras, R.: Inertial Navigation Systems and Its Practical Applications (2012)
6. Kus, Z., Fras, S.: Helicopter Control Algorithms from the set orientation to the set geographical location. In: Advanced Technologies for Intelligent Systems of National Border Security, pp. 3–13 (2013)
7. Jedrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 403–408, ISBN: 978-1-4673-6320-4, 19–21 June 2013
8. Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S.: Principles of Robot Motion, Theory, Algorithms and Implementation, A Bradford Book. The MIT Press, Cambridge, Massachusetts (2005)
9. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press, Cambridge, Massachusetts (2005)
10. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 157–175. Springer, Berlin (2013)

11. Kwiatkowski, J., Sobel, D., Jedrasiak, K.: FPGA based omnidirectional video acquisition device (OVAD). In: Recent Advances in Electrical Engineering and Computer Science, pp. 58–61. ISBN: 978-1-61804-249-1 (2014)
12. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: Design and implementation of mobile ground base station for UGV. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 57–71 (2014)
13. Bibik, P., Gradolewski, S., Zawislak, W., Zbudniewek, J., Darakchiev, R., Krczel, J., Michalski, M., Strzelczyk, K.: Problems of detecting unauthorized satellite transmissions from the VSAT terminals. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–4, ISBN 978-1-4673-1422-0, 8–9 Oct 2012
14. Durelli, G., Cresci, F., Sciuto, D., Porrmann, M., Santambrogio, M.D.: Mini-robot's performance optimization via online reconfiguration and HW/SW task scheduling. In: 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp. 437–442 (2012)
15. Ostafew, Ch. J., Schoellig, A.P., Barfoot, T.D., Collier, J.: Speed Daemon: experience-based mobile robot speed scheduling. In: Canadian Conference on Computer and Robot Vision, pp. 56–62 (2014)
16. Scheduling (computing), http://en.wikipedia.org/wiki/Scheduling_(computing)
17. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical analysis of IED detonation effect on steel plate. Acta Mechanica et Automatica **6**, 10–12 (2012)
18. Jedrasiak, K., Nawrat, A., Wydmanska, K.: SETh-link the distributed management system for unmanned mobile vehicles. Adv. Technol. Intell. Syst. Nat. Border Secur. **440**, 247–256 (2013). (Studies in Computational Intelligence)
19. Stallings, W.: Operating Systems: Internals and Design Principles. Pearson Education, New York (2009)
20. Kus, Z., Nawrat, A.: Object tracking in a picture during rapid camera movements. Vis. Based Syst. UAV Appl. **481**, 77–91 (2013). ISBN 978-3-319-00368-9 (Studies in Computational Intelligence)
21. Nawrat, A., Jedrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Proceedings of WSEAS International Conference on Mathematics and Computers in Science and Engineering, Ed. Shenyong Chen, no. 9, World Scientific and Engineering Academy and Society (2009)
22. Kus, Z., Nawrat, A.M.: Camera head control system with a changeable gain in a proportional regulator for object tracking. Innovative Control Syst. Tracked Veh. Platforms **2**, 105–125 (2014). (Studies in Systems, Decisions and Control)
23. Jedrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. Comput. Vis. Graph. **5337**, 391–399 (2009). (Lecture Notes in Computer Science)
24. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control. Comput. Vis. Graph. **7594**, 407–413 (2012). (Lecture Notes in Computer Science)
25. Daniec, K., Iwaneczko, P., Jedrasiak, K., Nawrat, A.: Vision based systems for UAV applications. Studies in Computational Intelligence, vol. 481, pp. 219–232. ISBN: 978-3-319-00368-9 (2013)
26. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. Comput. Vis. Graph. **8671**, 302–315 (2014). (Lecture Notes in Computer Science)

Application of Genetic Algorithms for Identification of Simulated Systems

Martyna Ulinowicz and Janusz Narkiewicz

Abstract A new system identification method (TSGA) based on Genetic Algorithms for system simulation is presented. The method is performed in two steps (S1 and S2). In each step of the method, a new approach to the population definition and identification process are applied for identification of parameters of system model. The simulated model with identified parameters is used for simulations of the system behavior. The method is suitable to linear as well as nonlinear models.

Keywords System identification · Genetic algorithms · Identification for simulation

1 Introduction

Simulations are widely used in manufacturing [1, 2], design and control of dynamic systems (airplanes [3–5], rotorcraft [6], ships and submarines [7], aeronautical and aerospace systems [8–10], mobile robots [11–14] etc.), signal processing [8, 15, 16], econometrics and finances, biology, biomechanics [17], biomimetics and medicine [18].

Simulations are aimed to analyze systems performance for better understanding of processes, design of control systems, predicting influence of external factors on investigated systems. Results of simulations, based on the reliable models, can significantly reduce costs of prototyping and tests in laboratory and in the field. Very often tests are limited by safety regulations. Simulations based on identified

M. Ulinowicz (✉) · J. Narkiewicz
Department of Automation and Aeronautical Systems, Warsaw University of Technology,
00-665 Warsaw, Poland
e-mail: martyna.ulinowicz@meil.pw.edu.pl

J. Narkiewicz
e-mail: jnark@meil.pw.edu.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_18

models give an opportunity to observe and analyze the system in an emergency states or conditions, also beyond assumed design constrains.

A system model should reflect, as close as possible, the performance of a real system and should mimic all its important features. Usually several simplifying assumptions are applied during a model design, such as for instance: neglecting couplings, simplifying the influence of the environment, neglecting variation in time or external disturbances.

System identification may improve reliability of the simulation model. System identification may be defined as a process of constructing an adequate mathematical model of a system based on data gathered during an identification experiment. For practical applications combination of a dedicated model and a proper identification method generates the useful simulation model.

2 Genetic Algorithms for Identification

The identification methods may be divided into two groups: deterministic and heuristic ones. The review of the methods applied to system identification showed that deterministic methods, frequently based on Least Squares approach, are mainly used [19, 20]. A major disadvantage of deterministic methods is a tendency to converge to a local extremum of cost function instead of a global one. Deterministic methods may be sensitive to initial values of parameters, which can lead to difficulties in numerical convergence. Identification based on data with significant noises may result in a poor accuracy of identified model parameters.

Genetic Algorithms (GA) are heuristic methods having many advantages in comparison to deterministic approaches. Inspired by nature and based on principles of genetics and natural selection, Genetic Algorithms mimic processes in nature to optimize natural phenomena [21]. GA search the solution space more extensively, which results in their ability to find global extremum. Genetic Algorithms, previously created to solve problems from all kinds of research fields, are universal methods, but deterministic solutions may be more accurate. One of proposed solutions to this problem is a combination of GA and one of a deterministic method such as conjugate-gradient optimization method [22]. A more general idea is proposed in [21], where hybridizing Genetic Algorithms with more successful optimization method in a particular domain of application is a fusion that lead to more accurate solutions.

A new identification method called the Two-Step Genetic Algorithm (TSGA) is developed. In the TSGA algorithm a different approach to population and identification process is applied in the first (S1) and the second (S2) step of the method, which is a way to find an accurate solution avoiding local minima traps during the identification process. The developed method explores more widely the search space to find the solution and is resistant to convergence to local minima. The method provides simultaneous search of the cost surface based on a wide sampling. When a local optimum is achieved, the algorithm generates a new point in a search space and moves toward other regions of search space using one of the operators created for this purpose—annihilation or mutation. It is also well suited for parallel computations. The method enables to describe considered system in a form of a linear or a nonlinear model. It deals well with a large number of variables and variable coupling, which may cause problems in case of deterministic approaches. Another strong advantage of the method is that it provides a list of solutions (one not a single one) close to optimal one. Such a list can be helpful in practical applications, where many factors may cause that the optimal solution cannot be achieved. The method converges quickly and it is resistant to disturbances contained in the data used for identification.

3 Two-Step Genetic Algorithm

The method TSGA is executed in two subsequent steps named shortly S1 and S2. Each step uses a similar structure of Genetic Algorithm (Fig. 1), but in both the population is defined in different way [23].

The main weakness of Genetic Algorithms is that solution obtained may be not as accurate as when the deterministic method is used. A chance to solve this problem may be seen in hybrid methods [24], which contain a Genetic Algorithms to search the solution space extensively and to avoid stopping in local minima and a deterministic method to find a solution more accurately. In the TSGA algorithm there is a different approach to population and identification process, applied in the first and the second step of the method, which is a way to find accurate solution avoiding local minima traps during the whole identification process.

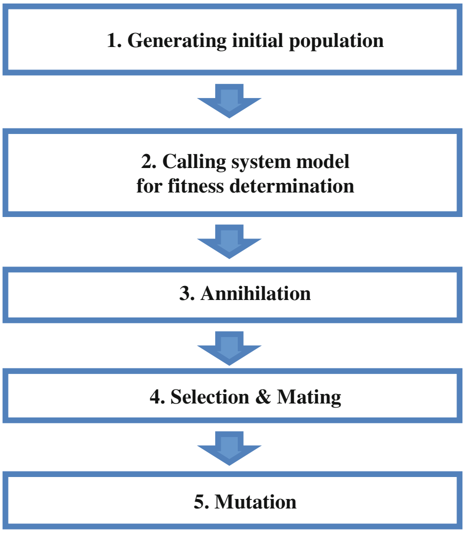

1. Generating initial population
2. Calling system model for fitness determination
3. Annihilation
4. Selection & Mating
5. Mutation

Fig. 1 Main levels (structure) of each step (S1 and S2) of two-step genetic algorithm for aeronautical applications (TSGA) algorithm

System identification may be understood as the process of constructing an adequate mathematical model of the system on the basis of input-output data gathered during identification experiment. When the mathematical model of the system is derived, the specific values of some parameters should be estimated to make the model reliable by being close to the real system. In other words a set of N_{par} parameter values should be estimated. Every of N_{par} identified parameters represent a different coefficient that may reflect some physical parameters, which are used in the mathematical model of considered system. A parameter value is a real number or bit representation.

The TSGA method description is described using Genetic Algorithms terminology but the reference to identification process and estimated set of N_{par} parameters is presented as well.

3.1 The Population

The population in GA is a limited size set of individuals. Each individual has its own data structure, named chromosome. One individual may have several chromosomes. In the first step of TSGA, an individual (parameter) has only one chromosome (parameter value). In that case, it is common practice to treat an individual as a chromosome (parameter value) and to use terms individual and chromosome interchangeable.

In S1 step of the TSGA method, for each of identified Ω_{CO_N} parameter types the own population of Ω_{MK} individuals (parameter values) is created. One generation is composed of N_{par} populations (representing parameter types) of N_{ch} individuals (parameters values) each. In S1 step of TSGA, each population represents N_{ch} individuals (parameter values) of one population (parameter type). A set of parameters, which should be estimated during identification process, is composed by taking one of N_{ch} individuals from each of N_{par} populations.

First the initial generation (for the first step of the algorithm) is created. There is N_{par} populations and each of them contains N_{ch} individuals, so the matrix $\boldsymbol{\theta}_{gz_{S1}}$ of size $N_{par} \times N_{ch}$ of uniformly distributed random numbers in a decimal form is generated (Fig. 2). These individuals represented in a decimal form, are transformed into binary strings of length L and they make $\boldsymbol{\theta}_{gb_{S1}}$ matrix (Fig. 3).

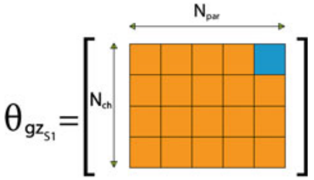


Fig. 2 Generating $N_{par} \times N_{ch}$ decimal random individuals and naming it $\boldsymbol{\theta}_{gz_{S1}}$

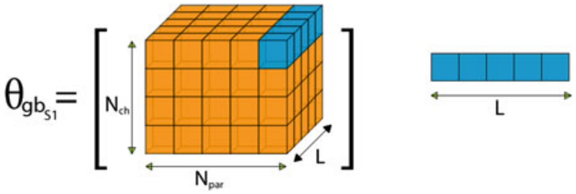


Fig. 3 Transforming individuals in decimal form into binary strings of length L and naming it $\boldsymbol{\theta}_{gb_{S1}}$

The knowledge about the identified system is exploited by determining constrains for each identified parameter. In this algorithm, constrains are formulated as two separate vectors with upper ($\mathbf{c}_{\max}$) and lower ($\mathbf{c}_{\min}$) boundary values of each individual. These constrains are taken into account for each individual during transformation from decimal into real number form, which results in creation of $\boldsymbol{\theta}_{g_{S1}}$ matrix (Fig. 4).

Next, individuals are paired randomly by taking one from each of N_{par} populations from matrix $\boldsymbol{\theta}_{g_{S1}}$ to get N_{ch} sets of N_{par} individuals (sets of parameters) (Fig. 5) and create the matrix $\boldsymbol{\theta}_{set_{S1}}$. For each set, the stages of TSGA algorithm called Model, Fitness, Annihilation, Selection & Mating and Mutation are applied.

In S1 step, TSGA algorithm successively processes populations of individuals (producing one generation) and replacing them by other populations, making the next generations in order to find solution to considered identification problem. The criterion of solution adequacy is expressed in the form of fitness function and is named the fitness criterion. To evaluate the level of individual (parameter)

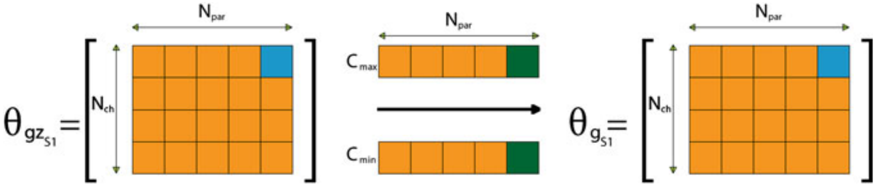


Fig. 4 Transforming each individual in decimal form into real number within constrains $\mathbf{c}_{\min}$ and $\mathbf{c}_{\max}$ and naming it $\boldsymbol{\theta}_{g_{S1}}$

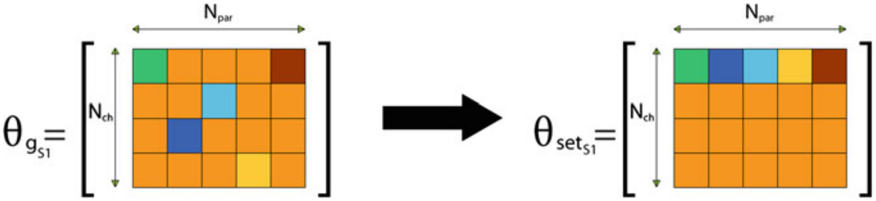


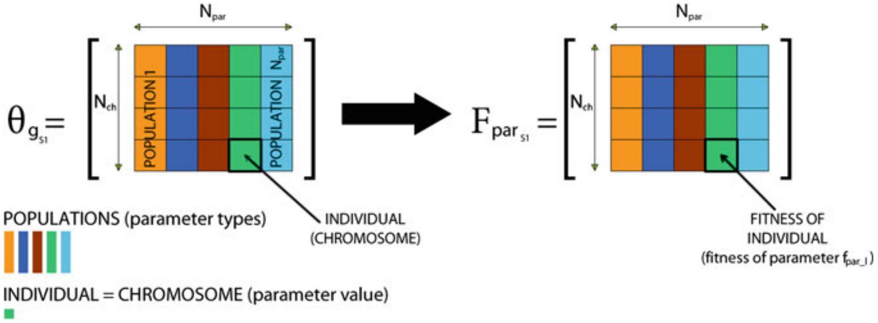
Fig. 5 Random pairing of individuals from matrix $\boldsymbol{\theta}_{g_{S1}}$ by taking one from each of N_{ch} populations to get N_{ch} sets of N_{par} parameters and create $\boldsymbol{\theta}_{set_{S1}}$ matrix

adjustment, the value of fitness of each individual is used. A fitness value is calculated using fitness function, which is an equivalent to cost function in deterministic methods. Each of N_{ch} individuals has its own fitness value (named fitness of parameter f_{par_I}). In one generation N_{ch} fitness values are calculated for each of N_{par} populations.

Beside fitness of each individual (f_{par_I}), the fitness of parameter sets (f_{par_s}) created from randomly chosen individuals of each of N_{par} populations (representing parameter types) are determined (Fig. 6).

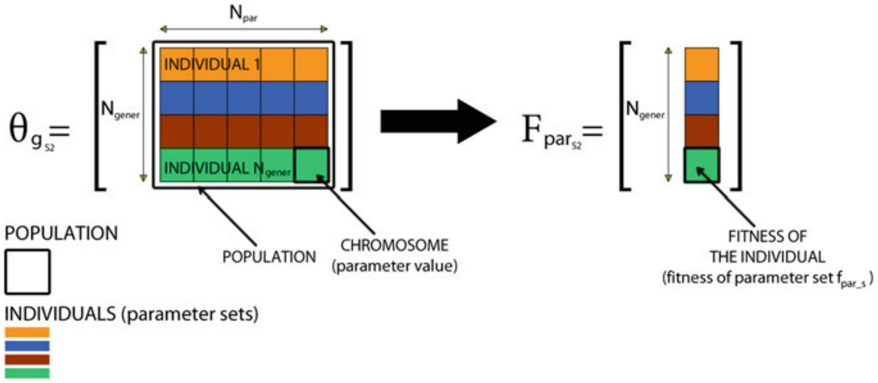
In the second step of TSGA, the approach to population and identification problem differs from the one in the previous step. The different approach to the population is a result of the following reasoning. Even if the fitness of each parameter separately is very good, these parameters gathered in a set may not lead to the best adjustment when they are used in the model of considered system.

In S2 a set of N_{par} parameters is treated as an individual described by N_{par} chromosomes (parameter types) and fitness of the whole set (f_{par_s}) is evaluated to determine the level of individual (parameter set) adjustment (Fig. 7). In each generation only one population is created from N_{gener} individuals (sets of N_{par}



N_par
θ_{g_S1} =
N_ch
POPULATION 1
POPULATION N_par
F_{par_S1} =
N_par
N_ch
POPULATIONS (parameter types)
INDIVIDUAL (CHROMOSOME)
INDIVIDUAL = CHROMOSOME (parameter value)
FITNESS OF INDIVIDUAL (fitness of parameter f_par_I)

Fig. 6 Population, individual and fitness of individual definition in S1 step of TSGA algorithm



N_par
θ_{g_S2} =
N_gener
INDIVIDUAL 1
INDIVIDUAL N_gener
F_{par_S2} =
N_gener
POPULATION
POPULATION
CHROMOSOME (parameter value)
INDIVIDUALS (parameter sets)
FITNESS OF THE INDIVIDUAL (fitness of parameter set f_par_s)

Fig. 7 Definition of population in S2 step of the TSGA algorithm

parameters each). As these N_{gener} individuals are the best ones created in the first step of the method, the selected initial population of the second part may lead to global solution, which is important advantage over deterministic methods.

3.2 System Model for Fitness Calculations

In this step of the algorithm, the knowledge of the system considered is exploited. After initial generation is postulated, the simulation model of the system is called and appropriate equations are solved. The model of the considered system should be formulated as a set of first order Ordinary Differential Equations in the state space form. The identification is performed based on the data gathered during off-line identification experiment. The results of identification experiment (measurements) are described in the form of observation matrix **Z**, which gathers observations at defined time steps. The observations are compared with corresponding values (model outputs **Y**) calculated using mathematical model of the considered system with identified parameter set. In S1 step of algorithm these sets are created by randomly selected individuals (parameter values) each of them from different population (parameter type). The real-value representation of individuals is used in the model stage of the algorithm. Based on observations **Z** and model outputs **Y**, fitness of each individual is calculated.

The fitness is determined in order to check the accuracy of values used for parameters to be estimated.

The fitness of each of Y_{all} model outputs y_m in reference to respective observation z_m is calculated as:

$$f_{y_i}\begin{cases} \left|\frac{y_i}{z_i}\cdot 100-100\right| & for\ z_i<y_i \\ \left|\frac{z_i}{y_i}\cdot 100-100\right| & for\ z_i\geq y_i \end{cases} \tag{1}$$

for $i = 1{:}N_{data}$

$$f_{y_m}=\frac{\sum_{i=1}^{N_{data}} f_{y_i}}{N_{data}} \tag{2}$$

for $m = 1{:}Y_{all}$.

Each parameter fitness f_{par_l} is obtained as an average value from the fitness of k output values y_j that depend from considered parameter:

$$f_{par\_l}=\frac{\sum_{j=1}^{k} f_{y_j}}{k} \tag{3}$$

for l = 1:N_{par},

where

l number of parameter considered,
k number of outputs y_j that depends from the parameter considered.

The values of parameter finesses (f_{par_l}) which refers to $\boldsymbol{\theta}_{set_{S1}}$ are gathered in matrix $\mathbf{F}_{par_{S1}}$ (Fig. 8).

The relations between parameters to be determined $\boldsymbol{\theta}$ and each model outputs $\mathbf{y}$ are represented by $\mathbf{P}_{out}$ matrix (Fig. 9).

The fitness of parameter set f_{par_s} is calculated as average from the fitness of $m = 1{:}Y_{all}$ outputs y_m:

$$f_{par\_s} = \frac{\sum_{m=1}^{Y_{all}} f_{y_m}}{Y_{all}}, \tag{4}$$

The values of finesses of N_{ch} parameter sets, which refers to $\boldsymbol{\theta}_{set_{S1}}$, are gathered in matrix $\mathbf{F}_{set_{S1}}$ (Fig. 10).

In S2 part of the algorithm, similar procedures in the Model and Fitness stage are performed as those from the first step (S1). The main difference is a result of different approach to the population in those steps. In S2 each individual (parameter set) is used to calculate the model outputs $\mathbf{y}_i$ in every time instant i = 1, 2, …, N_{data} (where N_{data} is number of data gathered during the measurements). Fitness of each individual (parameter set) f_{par_s} in the population is calculated using formula presented by Eq. 2. There is no need to calculate each parameter fitness f_{par_l}, so this element of procedure is omitted during execution of the algorithm for S2 step.

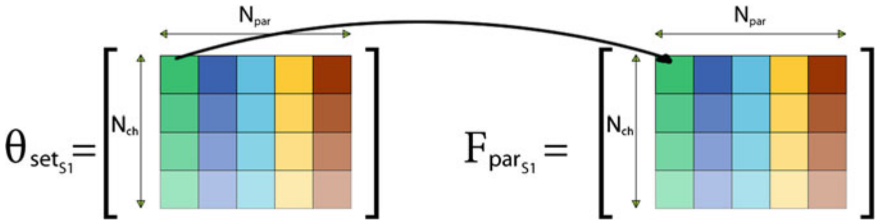


Fig. 8 Parameter fitness in S1 part of TSGA algorithm

Fig. 9 Definition of $\mathbf{P}_{out}$ matrix

$$P_{out} = \begin{array}{c|cccccc} \theta \backslash Y & Y_1 & Y_2 & Y_3 & \dots & \dots & Y_i \\ \hline \theta_1 & 1 & 1 & 1 & \dots & \dots & 0 \\ \theta_2 & 1 & 0 & 1 & \dots & \dots & 0 \\ \theta_3 & 0 & 1 & 1 & \dots & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta_j & 0 & 0 & 1 & \dots & \dots & 0 \end{array}$$

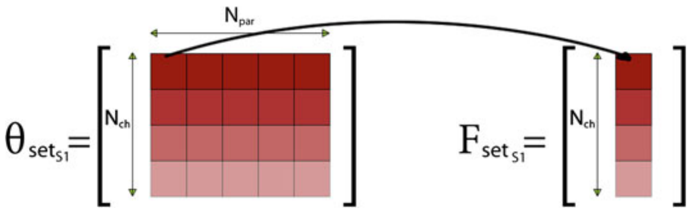


Fig. 10 Parameter set fitness in S1 step of TSGA algorithm

3.3 Introducing Genetic Operators

Selection and mating (together) and mutation are Genetic Algorithms operators, which are used to find the efficient solution. In [21] it is shown that the optimal solution is to use all these operators instead using only one of them (selection and mating or mutation). Selection and mating operators explore the solution space defined by the gene values of the parents in the current population. They are not able to introduce new values of genes and in that way to change the solution space that they explore. A mutation operator introduces new gene values continuously but in contrast to selection and mating, it may not take the larger directed steps. The efficiency of the method may be improved by applying all of these operators together. A mutation is a source of new gene values and a selection and mating explores the search space defined by the genes (also new ones) in population. In other words application of both reproductive operators ensures perturbation of existing solution by mutation and hybridization of existing solutions by recombination (selection and mating) [21]. In both stages of TSGA method selection, mating and mutation operators. In addition to that the annihilation process is performed.

In S1 (S2) the individuals which have fitness $f_{par\_l} \geq M_{f_{par\_l}}$ ($f_{par\_s} \geq M_{f_{par\_s}}$) are annihilated (removed) to force the algorithm to converge to optimal populations of individuals (Figs. 11 and 12). The $M_{f_{par\_l}}$ ($M_{f_{par\_s}}$) is the reference value that is assumed before the identification process. A new random individuals are created in

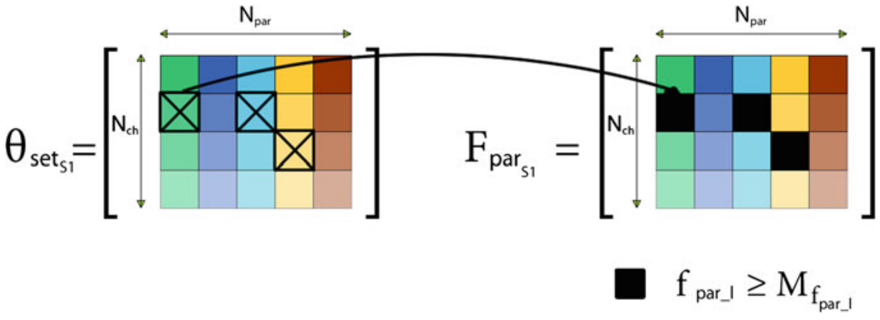


Fig. 11 Annihilation stage in S1 step of TSGA algorithm

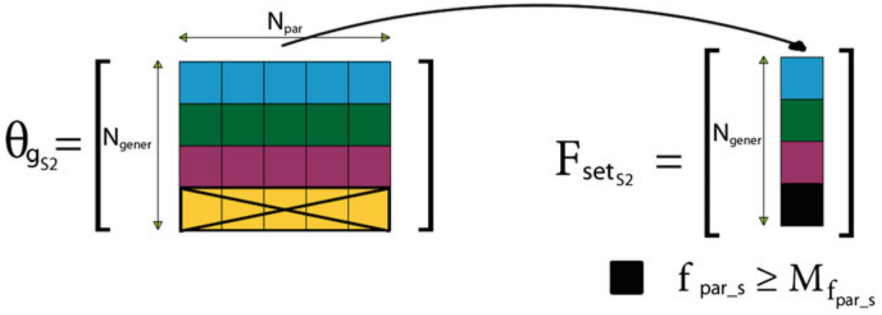


Fig. 12 Annihilation stage in S2 step of TSGA algorithm

the same way, as in the generation stage of individuals, and they replace annihilated ones taking into account parameter constrains. In the S1 also the binary form of individuals, needed in further calculations, is prepared.

On the basis of fitness values, these individuals which are the most representative are selected to become parents of future offspring. This process is known as natural selection. The pairing of individuals results in selecting parents, who produce one or more offspring (mating process).

In S1 step of the algorithm, the binary form of individuals which are left after annihilation is investigated in the Selection and Mating stage.

The individuals, which are left after annihilation, are selected with probability P_c and then randomly paired—prepared for mating (Figs. 13, 14 and 15). The mating process depends on the individual fitness value (parameter fitness value f_{par_l} in S1) compared with the reference one calculated for each parameter type:

$$P_{fr_{S1}} = f_{par\_l_{\min}} + \frac{f_{par\_l_{mean}} - f_{par\_l_{\min}}}{2}, \tag{5}$$

where

$f_{par\_l_{\min}}$ minimum individual (parameter) fitness value in the population,
$f_{par\_l_{\text{mean}}}$ mean individual (parameter) fitness value in the population.

If fitness of both individuals are lower than $P_{fr_{S1}}$, the mating is not applied because the level of adjusting of those individuals is adequate. If fitness of one

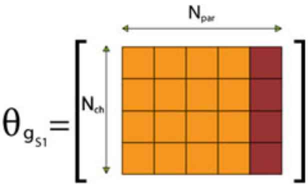


Fig. 13 One generation in decimal form in S1 step of TSGA algorithm

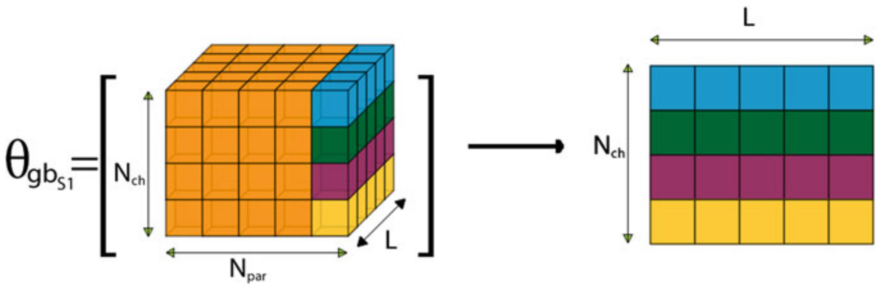


Fig. 14 One generation in binary form in S1 step of TSGA algorithm

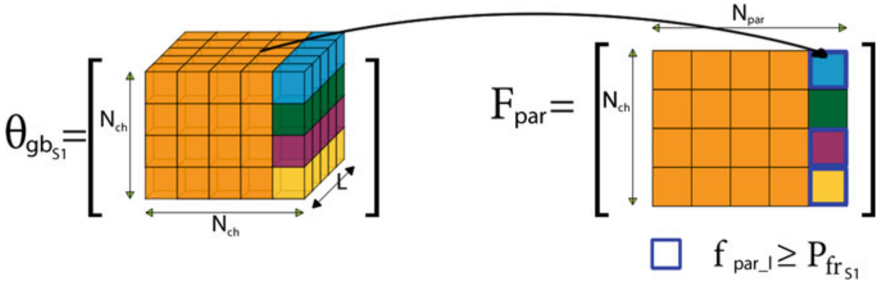


Fig. 15 Selecting parameters for crossover in S1 step of TSGA algorithm

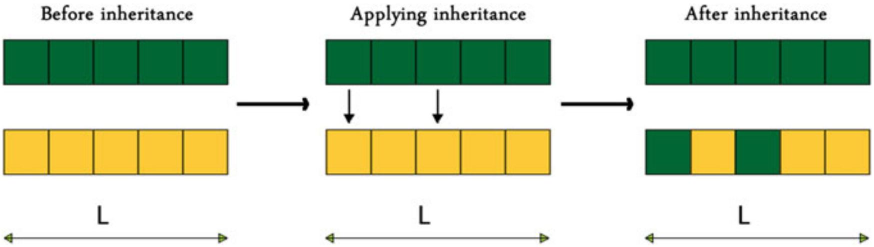


Fig. 16 Inheritance procedure in S1 and S2 stage of TSGA algorithm

individual from a pair is lower than $P_{fr_{S1}}$, the individual with the fitness lower than reference fitness value inherits from other individual from the same pair (Fig. 16).

The regular crossover, where every child has genetic material from both parents, appears when fitness of both individuals are lower than $P_{fr_{S1}}$. In result individuals (parents) are exchanging their features (Fig. 17), making two new individuals (offspring).

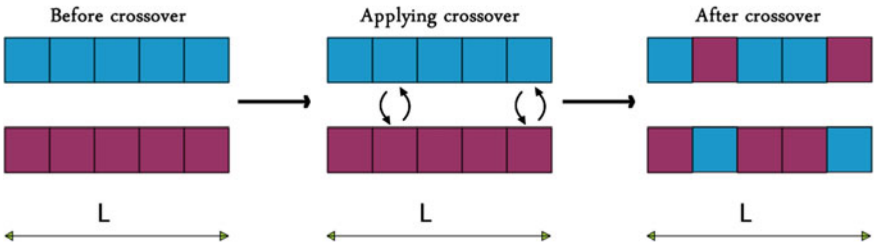


Fig. 17 Regular crossover procedure in S2 stage of TSGA algorithm

In S2 the Selection and Mating process is performed similarly to the corresponding one in the first step of the algorithm. The main difference is due to using real-number form of individuals (in S2) instead of the binary one.

The mating process depends on the fitness value of individual compared to the previously determined reference one (Fig. 18). In S2 the individual is defined as parameter set, so the fitness of parameter set f_{par_s} is used for comparison. The reference fitness level $P_{fr_{S2}}$ is calculated for each individual (parameter set) from the relation:

$$P_{fr_{S2}} = f_{par\_s_{\min}} + \frac{f_{par\_s_{mean}} - f_{par\_s_{\min}}}{2}, \tag{6}$$

where

$f_{par\_S_{\min}}$ minimum individual (parameter set) fitness value in the population,
$f_{par\_S_{\mathrm{mean}}}$ mean individual (parameter set) fitness value in the population.

Mutation is another way to explore the solution space by genetic algorithm. This operator, by introducing traits beyond the original population, prevents the algorithm from completing, before sampling the entire search surface. The individuals that left after annihilation, selection and mating processes are subjected to mutation

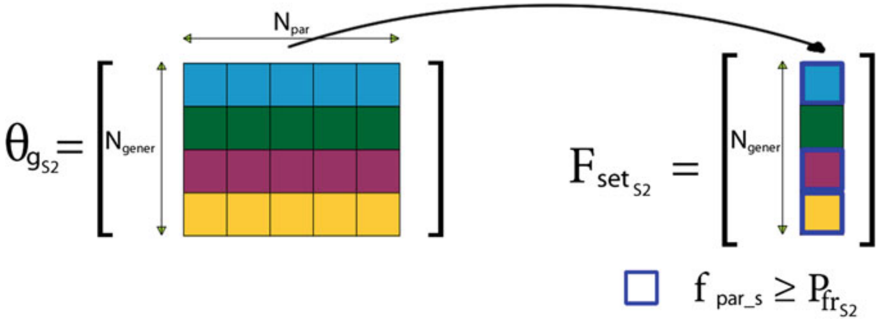


Fig. 18 Selecting parameters for crossover in S2 step of TSGA algorithm

with probability P_{mut}. As in Selection & Mating processes, the binary form of individual representation is used.

Mutation may be performed in one of four ways: replacement, end for end swap, adjacent swap, random swap, with probability R_{ch}. If a binary form is used, replacement is performed by swapping zeros to ones and vice versa in an individual selected for mutation (Fig. 19).

The mutation named 'end for end swap', requires to define randomly swap place and then changing equal number of features before and after this randomized swap place (Fig. 20).

In an adjacent swap mutation, the swap place is picked randomly. The feature of considered individual from this swap place is replaced by following feature and vice versa (Fig. 21).

In so called random swap mutation not one but two randomly chosen features from individuals are chosen and their places are exchanged (Fig. 22).

Due to specific character of the population (the individual is defined as parameter set and each parameter has different constrains), in S2 only one kind of mutation (replacement) is used.

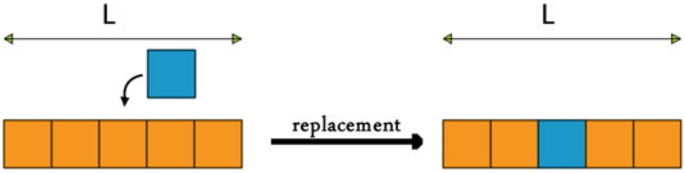


Fig. 19 Replacement mutation performed in S1 step of TSGA algorithm

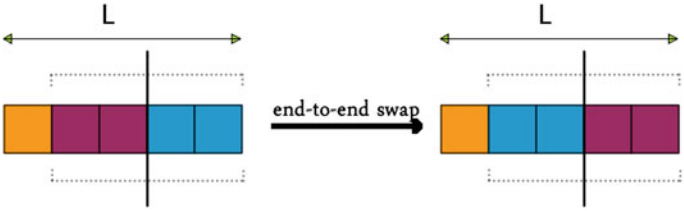


Fig. 20 End-to-end swap mutation performed in S1 step of TSGA algorithm

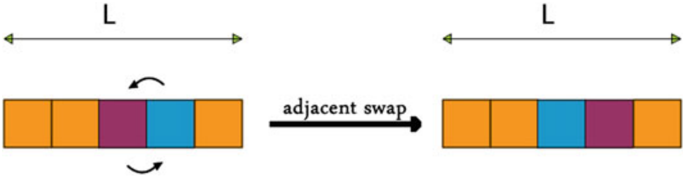


Fig. 21 Adjacent swap mutation performed in S1 step of TSGA algorithm

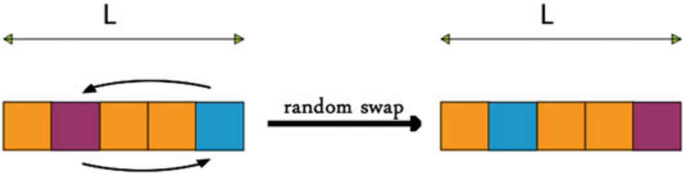


Fig. 22 Random swap mutation performed in S1 of TSGA algorithm

3.4 Initial Population for S2

At the end of the first step (S1) of the method an initial population for S2 step of the algorithm is prepared. After annihilation, selection, mating and mutation, the parameters from N_{par} populations placed in $\boldsymbol{\theta}_{g_{S1}}$ matrix are replaced by individuals transformed in all these transformation processes. In that manner, a new generation of individuals is created.

After $N_{iter\_\max S1}$ iterations N_{gener} sets of parameters are chosen (without repetitions) and placed in $\mathbf{T}_{best}$ matrix. This matrix gathers the N_{par_s} sets of parameters with the best fitness of the whole set (f_{par_s}) (Fig. 23) and another N_{par_l} sets.

The last ones are created from N_{par_l} individuals of the best fitness (f_{par_l}) from each of N_{par} population and then they are combined in N_{par_l} sets, starting from the individuals of the best parameter fitness f_{par_l} in each population (Fig. 24).

Summing up, at the end of the first step of TSGA, the N_{par_s} parameter sets with best fitness of the whole set (f_{par_s}) are selected. Another N_{par_l} sets are added,

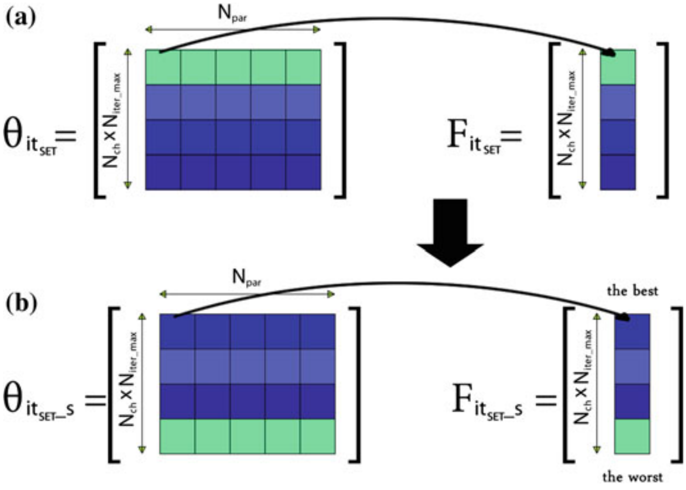


Fig. 23 **a** $N_{iter\_\max S1}$ generations of parameter sets gathered in one matrix $\boldsymbol{\theta}_{it}$ and respective parameters sets fitness placed in $\mathbf{F}_{it_{SET}}$; **b** Parameter sets from $\boldsymbol{\theta}_{it}$ matrix sorted by fitness of parameter sets ($\mathbf{F}_{it_{SET}\_s}$) and gathered in $\boldsymbol{\theta}_{it_{SET}\_s}$ matrix

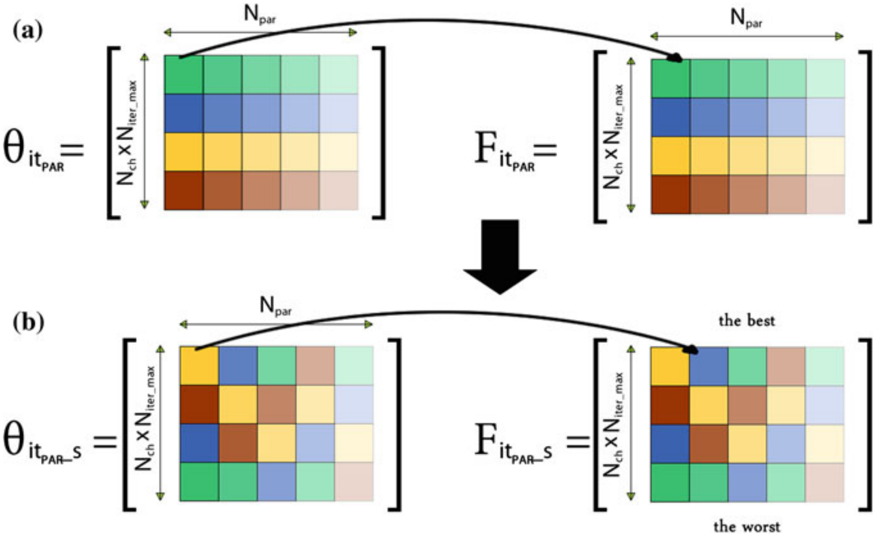


Fig. 24 **a** $N_{iter\_\max S1}$ generations of parameters gathered in one matrix $\boldsymbol{\theta}_{it}$ and fitness of respective parameters in the matrix $\mathbf{F}_{it_{PAR}}$; **b** Parameters from $\boldsymbol{\theta}_{it}$ matrix sorted by their fitness and gathered in $\boldsymbol{\theta}_{it_{PAR\_s}}$ matrix and fitness of respective parameters sorted analogically is placed in matrix $\mathbf{F}_{it_{PAR}}$

created from N_{par_l} individuals of the best fitness from each of N_{par} populations, combined in N_{par_l} sets, starting from the best ones. In that manner $N_{gener} = N_{par\_s} + N_{par\_l}$ sets of parameters are selected ($\mathbf{T}_{best}$) and are used as initial generation in the second step of the algorithm (Fig. 25).

The algorithm stops when the level of accuracy is acceptable $f_{par\_s} \leq A_l$, where A_l is a reference value specified previously or after previously specified maximum iteration number $N_{iter\_\max S2}$ is exceeded.

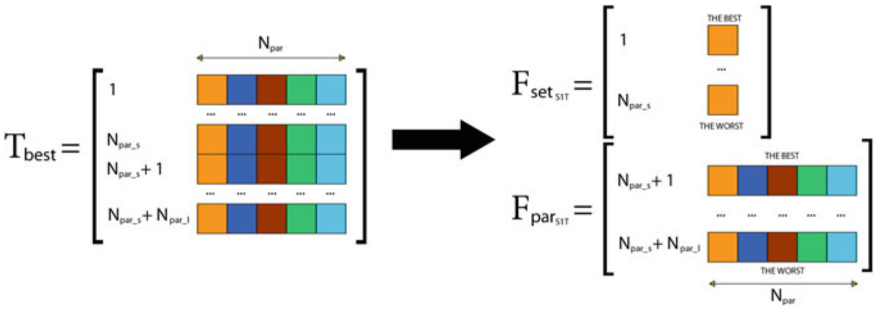


Fig. 25 Initial population for S2 step of the TSGA algorithm definition

4 Verification Method of TSGA Verification by Simulation

The Two-Step Genetic Algorithm (TSGA) is successfully verified by simulation experiment performed on an Individual Flap Deflection System described by linear model in a form of first order ordinary differential equations [23]. Also identification of the nonlinear airplane model in a first order ordinary differential equations form on the basis of the flight-test data [25] using the TSGA method are performed.

Parameters $\boldsymbol{\theta}_{ACR_i}^{id}$ of the best fitness are used to calculate output variables $\mathbf{y}_{ACR_i}$ for every time instant $i = 1, 2, \ldots, N_{data}$ (N_{data}—number of data gathered during measurements) of nonlinear airplane model are calculated and compared to observation data gathered during the flight $\mathbf{z}_{ACR_i}$ (Fig. 26). As it may be seen in the Fig. 26, all calculated outputs $\mathbf{y}_{ACR_i}$ of nonlinear airplane model are very close to the observation data gathered during the flight $\mathbf{z}_{ACR_i}$. The results indicate that TSGA identification method enables successful estimation of considered model parameters.

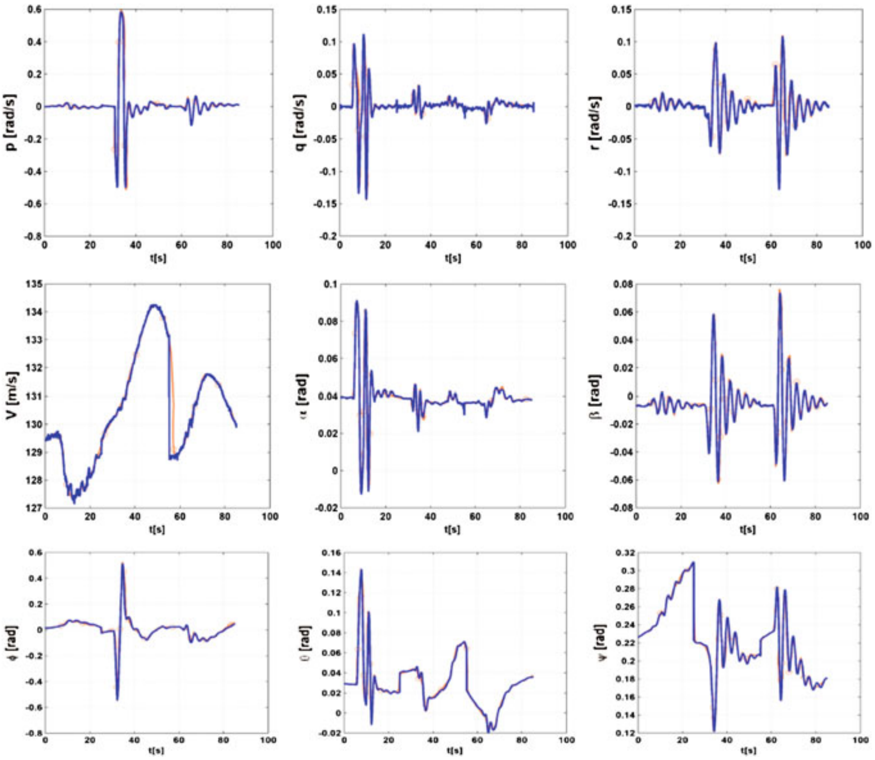


Fig. 26 Airplane outputs $\mathbf{y}_{ACR_i}$ calculated using nonlinear model and identified parameters $\boldsymbol{\theta}_{ACR_i}^{id}$ of the best fitness, compared to observations $\mathbf{z}_{ACR_i}$

5 Conclusion

A new identification method for system simulation, called the Two-Step Genetic Algorithm (TSGA), is presented in details. In the TSGA algorithm there is a different approach to population and identification process, applied in the first (S1) and the second (S2) step of the method, which is a way to find an accurate solution avoiding local minima traps during the identification process. The developed method explores more widely the search space to find the solution and is resistant to convergence to local minima. The method provides simultaneous search of the cost surface based on a wide sampling. When a local optimum is achieved, the algorithm generates a new point in a search space and moves toward other regions of search space using one of the operators created for this purpose—annihilation or mutation. It is also well suited for parallel computations.

The method enables to describe considered system in a form of a linear or a nonlinear model. It deals well with a large number of variables and variable coupling, which may cause problems in case of deterministic approaches. Another strong advantage of the method is that it provides a list of solutions close to optimal one not a single one. Such a list can be helpful in practical applications, where many factors may cause that the optimal solution cannot be achieved. The TSGA converges quickly and it is resistant to disturbances contained in the data used for identification.

As the verification of the method will be a subject of another publication, it is just mentioned that to these purpose the Individual Flap Deflection System described by linear model and an airplane described as nonlinear model are used.

References

1. Mężyk, A., Dobrzaniecki, P.: Optimization of operating parameters of the drive system on the example of a mining rail-vehicle. Ind. Transp. Mach. **2**, 46–49 (2011)
2. Nawrat, A., Skorek, J.: Inverse approach and sensitivity analysis for identification of ingot mould thermal resistance in continuous casting of metals. Int. J. Comput. Fluid Dyn. (2005). ISSN:1061 8562
3. Puttige, V.R., Anavatti, S.G.: Real-time multi-network based identification with dynamic selection implemented for a low cost UAV. In: IEEE International Conference on Systems, Man and Cybernetics, Montreal (2007)
4. Zugaj, M., Narkiewicz, J.: Autopilot for reconfigurable flight control system. In: 7th International Seminar on Recent Research and Design Progress in Aeronautical Engineering and Its Influence on Education, Tallin, Estonia (2006)
5. Ward, D.G., Barron, R.L., Carley, M.P., Curtis, T.J.: Real-time parameter identification for self-designing flight control. In: Proceedings of the IEEE 1994 National Aerospace and Electronics Conference, vol. 1, pp. 526–531 (1994)
6. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guidance Control Dyn. **35**(4), 1354–1362 (2012). doi:10.2514/1.51371

7. Andrzejczak, M., Narkiewicz, J.: An on-line system prediction of the vessel position based on Julier-Uhlmann filter. In: Proceedings of 11th Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg (2004)
8. Söderström, T., Stoica, P.: System Identification. PWN, Warszawa (1997)
9. Ulinowicz, M., Narkiewicz, J.: Identification of EMA Dynamic Model, Mechatronics Recent Technological and Scientific Advances, pp. 375–385. Springer, Berlin (2011)
10. Ulinowicz, M., Narkiewicz, J.: Modeling and identification of actuator for flap deflection. J. Autom. Mob. Robot. Intell. Syst. JAMRIS (2012) http://link.springer.com/chapter/10.1007%2F978-3-642-05241-5_10
11. Jarzębowska, E.: A velocity observer design for tracking task-based motions of unicycle type mobile robots. Commun. Nonlinear Sci. Numer. Simul. **16**, 2301–2307 (2011). Elsevier
12. Mężyk, A.: The use of optimization procedures in tuning vibration dampers. Eng. Optim. **34** (5), 503–521 (2002)
13. Priifer, M., Schmidt, C., Wahl, E.: Identification of robot dynamics with differential and integral models: a comparison. In: IEEE International Conference on Robotics and Automation, San Diego CA, vol. 1, pp. 340–345 (1994)
14. Babiarz, A., Bieda, R., Jaskot, K.: A distributed control group of mobile robots in a limited area with a vision system. In: Vision Based Systems for UAV Applications. Studies in Computational Intelligence, vol. 481, pp. 157–175 (2013). ISBN:978-3-319-00368-9
15. Czornik, A., Nawrat, A.: On controllability of linear systems with jumps in parameters. Adv. Intell. Soft Comput. **118**, 57–81 (2012)
16. Osowski, S.: Dynamic Systems and Processes Modeling and Simulation. WUT Publishing House, Warsaw (2007)
17. Burczyński, T., Kuś, W., Brodacka, A.: Multiscale modeling of osseous tissues. J. Theor. Appl. Mech. **48**(4), 855–870 (2010)
18. Kuś, W., Burczyński, T.: Bioinspired Algorithms in Multiscale Optimization, Computer methods in Mechanics, pp. 183–192. Springer, Berlin (2010)
19. Dimogianopoulos, D.G., Hios, J.D., Fassois, S.D.: Statistical fault detection and identification in aircraft systems via functionally pooled nonlinear modeling of flight data dependencies, Greece, 2006
20. Klein, V., Murphy, P.C.: Aerodynamic parameters of high performance aircraft estimated from wind tunnel and flight testing. NATO Res. and Tech. Org., March 1999 RTO-MP-11, Paper 18
21. De Jong, K.A.: Evolutionary Computation: A Unified Approach. The Massachusetts Institute of Technology Press, Cambridge (2006)
22. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer, Berlin (2008)
23. Ulinowicz, M.: Identification of aeronautical system parameters using a genetic algorithm. Ph. D. thesis, WUT (2013)
24. Zugaj, M., Narkiewicz, J.: Autopilot for Reconfigurable Flight Control System. J. Aerosp. Eng. **22**(1), 78–84 (2009)
25. Jategaonkar, Ravindra V.: Flight Vehicle System Identification, A Time Domain Methodology. American Institute of Aeronautics and Astronautics, Reston, Virginia (2006)

Quadrotor Dynamics and Control for Precise Handling

Przemysław Bibik, Janusz Narkiewicz, Maciej Zasuwa and Marcin Żugaj

Abstract The paper contains part of the results of a research done at the Warsaw University of Technology (WUT) devoted to modeling and control system design for a quadrotor UAV. The quadrotor lift, propulsion and control loads are produced by four identical rotors, performance of which strongly depends on flight conditions. The quadrotor angular rates result in not identical flow through each of the rotors due to their not coaxial configuration which may cause disturbances of a flight path. The development of a complex dynamic model and a control law enhancement method for a quadrotor is presented in this paper. The novel approach to control system development led to increase in quadrotor handling precision, due to compensates disturbances of the flight path, which makes a quadrotor more capable to operate in a confined space.

Keywords UAV · Quadrotor · Flight dynamic · Flight control system

1 Introduction

The quadrotor seems to be the most popular type of rotorcraft UAV configuration, being recently investigated by a vast amount of companies, research institutes and universities [1–3]. There are several reasons for this fact. There is a great interest in

P. Bibik (✉) · J. Narkiewicz · M. Zasuwa · M. Żugaj
The Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology, Warsaw, Poland
e-mail: pbibik@meil.pw.edu.pl

J. Narkiewicz
e-mail: jnark@meil.pw.edu.pl

M. Zasuwa
e-mail: mzasuwa@meil.pw.edu.pl

M. Żugaj
e-mail: zugaj@meil.pw.edu.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_19

small, unmanned flying vehicles for both civil and military applications, and this type of vehicle provides unique capabilities of hover, vertical and horizontal flight in all directions; its controlling is simpler comparing to other rotorcraft configurations; the cross-couplings between the various degrees of freedom are not so severe as in single rotor helicopters. It is relatively easy to design control system using simple classical control algorithms. A quadrotor may also be an efficient, versatile testbed for control and navigation algorithms and operation of various types of payload [4–6].

Many papers were focused on development and analysis of quadrotor stability and control systems. Various methods such as H infinity [7], LQR [8, 9], sliding mode [10, 11], backstepping [10, 12], reference model [13–15] or PID [2, 9] were investigated at quadrotors for their robustness [16, 17], fault tolerance [18] and affordability [7] to stabilise and control of the vehicle. The control system design and validation usually was done using simple dynamic model of quadrotor where rotor thrust and torque were modeled as linear terms expressed as function of the square of rotor angular velocity [9, 13, 14, 18, 19]. This simplification allows to validate model only for hovering conditions and controllers, which were developed based on it, provide poor performance in a real application [2, 8, 9] especially for high forward velocity and for rapid maneuvers. The nonlinear complex rotor model is needed to develop and validate the quadrotor control system, which will guarantee the good stability and controllability performance for all flight regimes and fully exploit the advantages of this rotorcraft configuration. Two quadrotor simulation models were develop: a comprehensive nonlinear and simplified linear, needed for real time operation in simulator.

This research was part of the project aimed to investigate application of a quadrotor indoor and in urban environment. Operation in such conditions is hazardous due to proximity of obstacles and turbulences produced by the quadrotor rotors, so it requires robust flight control system which will provide precise handling. The enhanced classical PID control method is presented in this paper. The control law enhancement methods applied additional couplings including the rotor model techniques, to improve the control quality for high speed and rapid maneuvers.

The quadrotor model with flight control system was tested on UAV simulator. The simulator platform is a multiuse hardware and software testbed, developed by WUT research team. The simulator has a functionality of software reconfiguration and may be connected to real hardware, which makes it very useful platform for any new algorithm and mobile platform tests. The simulator is composed of advanced hardware components: visual system, generic stand (panel) of vehicle operator, instructor stand operated by computing hardware. The simulator has no motion system for the platform, which is not necessary for simulators of unmanned vehicles.

For quadrotor implementation, some software modifications were made, so the dynamic and control models may be uploaded fast and easily. Simulator accepts now Matlab/Simulink™ models, so they can be easily debugged and upgraded outside simulator.

2 Quadrotor Simulation Model

The quadrotor simulation model was developed under two main constraints: a requirement for a real-time operation in simulator for training of rotorcraft operators and modeling of complex aerodynamic interactions of the propellers and obstacles. These requirements led to modeling of the vehicle in two steps. First one was development of comprehensive models of propellers loads which were validated using data from full scale experimental test. Then, validated propeller loads were used to generate multivariable maps of loads which were applied in the 6 DOF quadrotor dynamic model.

The main assumptions for dynamic model development were:

- quadrotor is axially symmetric,
- quadrotor is controlled in “x” configuration,
- rotation axes of rotors are parallel to the body vertical axis of symmetry,
- all parts of the quadrotor are rigid,
- mass of the vehicle is constant,
- induced velocity is modeled with the Glauert formula for forward flight,
- equations of motion describe the motion of the body center of mass.

The main coordinate systems used in the model development are shown in Fig. 1.

Using assumptions above, nonlinear equations of quadrotor motion in $0_b x_b y_b z_b$ coordinate system were formulated in a general form [20, 21]:

$$\mathbf{I_p}\dot{\mathbf{x}} = \mathbf{f}_I(x) + \mathbf{f_{GK}}(\mathbf{y}) + \mathbf{f_{AK}}(\mathbf{x},\mathbf{y}) + \mathbf{f_{AW}}(\mathbf{x},\delta_i), \tag{1}$$

$$\dot{\mathbf{y}} = \mathbf{Tx} \tag{2}$$

where $\mathbf{x} = [U \quad V \quad W \quad P \quad Q \quad R]^{\mathrm{T}}$—state vector composed of linear velocities and angular rates, $\mathbf{y} = [x_g \quad y_g \quad z_g \quad \Phi \quad \Theta \quad \Psi]^{\mathrm{T}}$—vector describing position

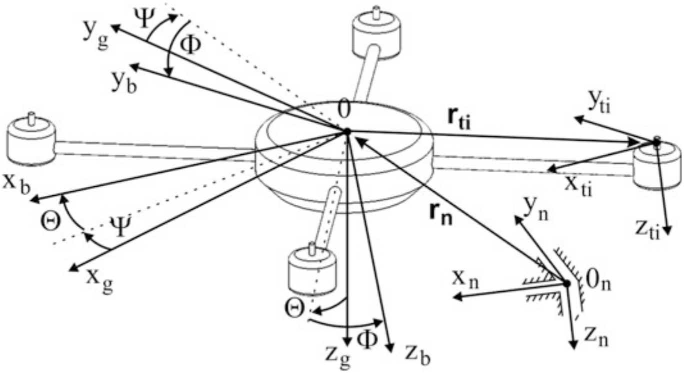


Fig. 1 The main coordinate systems used in the quadrotor dynamic model

coordinates and angles of attitude, δ_i—ith engine control signal, $\mathbf{I_p}$—inertia matrix, $\mathbf{f}_I(x)$—inertia loads, $\mathbf{f_{GK}}(\mathbf{y})$—gravity loads, $\mathbf{f_{AK}}(\mathbf{x}, \mathbf{y})$—fuselage aerodynamic loads, $\mathbf{f_{AW}}(\mathbf{x}, \boldsymbol{\delta})$—rotors aerodynamic loads, $\mathbf{T}$—transformation matrix in kinematic equations [20, 21].

The gravity loads are calculated in a standard way for the 6 DOF mobile rigid object. It was assumed that the fuselage generates only drag force and the drag force coefficients were calculated using simple geometrical model of the fuselage for given inflow velocity, angle of attack and bank angle. The main effort in quadrotor modeling was dedicated to obtaining verified loads generated by propellers. The total vector of propeller loads $\mathbf{f_{AW}}$ is a sum of loads generated by each rotor, transferred to the body coordinate system.

$$\mathbf{f_{AW}}(\mathbf{x}, \delta_i) = \sum_{i=1}^{4} \mathbf{f_{AWi}}(\mathbf{x}, \delta_i) \tag{3}$$

In a general case, vector (3) consists of three components of a force and three components of a moment. The quadrotor propellers are sources of lift force, propulsion force and control forces and moments. This is why proper modeling of these loads is crucial for the quadrotor model fidelity. Several various models of propeller loads may be found in literature. The simplest models describe only thrust and torque, more complex ones contain also drag. The models differ also due to methods of calculating forces and moments. The simplest models [9] describe rotor thrust and torque as quadratic functions of rotor angular velocity. More complex models take into account variation of inflow and induced velocities in different flow states [6]. Quite often the constant coefficients of the loads functions, independent of the flight conditions, are calculated using the static thrust and torque measurements. Such an approach leads to simple and effective models, however, these models are valid only for hover. Comparison of rotor thrust and torque calculated using such model with the results of measurements performed in WUT wind tunnel is presented in Figs. 2 and 3 for various inflow velocities.

Axial inflow positive velocities W appear for instance when a quadrotor is climbing vertically. In results presented, rotor thrust is very sensitive to variation of inflow velocity, so keeping thrust constant for all inflow velocities results in a significant overestimation, even for axial flight. It can be also noticed that the rotor torque is not so much sensitive and axial inflow velocity variations do not change it very much.

Not only quadrotor linear velocities contribute to change of inflow velocities. Propellers are placed not in the quadrotor center of gravity (Fig. 4) so also body angular rates may contribute to inflow velocities:

$$\mathbf{v_{ti}} = \mathbf{v_b} + \boldsymbol{\omega}_\mathbf{b} \times \mathbf{r_{ti}} \tag{4}$$

where $\mathbf{v_b}$, $\boldsymbol{\omega}_\mathbf{b}$—vectors of quadrotor velocities and rates in the body reference frame, $\mathbf{r_{ti}}$—position vector of a propeller in the body reference frame (Fig. 4).

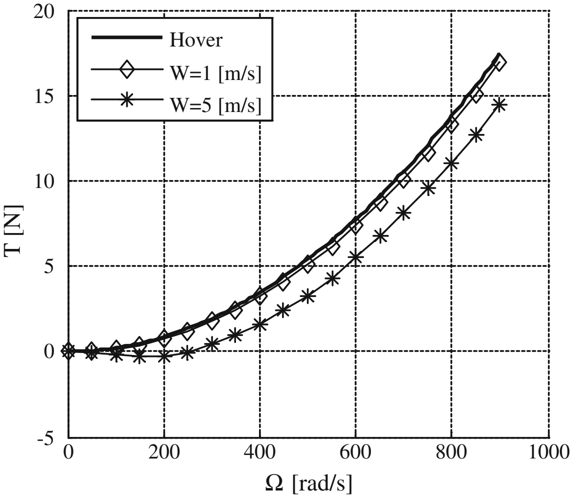


Fig. 2 Rotor thrust versus rotor angular speed for various inflow velocities

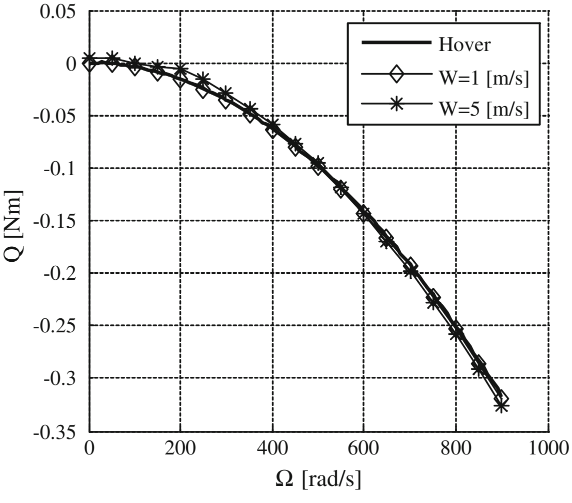


Fig. 3 Rotor torque versus rotor angular speed for various inflow velocities

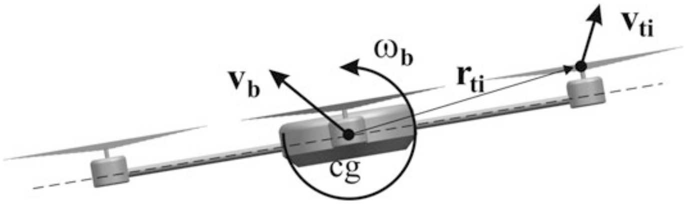


Fig. 4 Propeller velocity vector in body reference frame

Neglecting this effect for the quadrotor, which is a very agile vehicle, limits simulation model to hover and low speed flights. As another effect resulting from (4) in a maneuver flight, each rotor is subjected to different inflow and thus generates different loads.

These observations were taken into account in the WUT developed model. The quadrotor propellers loads consist of a full set of six components—three forces and three moments of forces, calculated using Blade Element Method. First, the velocities at the blade cross-section are expressed as functions of body linear velocities and angular rates, including also velocities resulting from rotors angular velocities and induced velocity. The induced velocity is modeled using Glauert formulae with corrections for forward flight [21]. Then the total flow velocities are used to calculate lift and drag forces acting in a blade cross-section, taking into account blade chord, twist, and lift and drag coefficients of the airfoil. In the next step the six components of the rotors loads are calculated by integrating forces and moments along the blade span. Sample results for 15 × 4″ propeller, for the case of inflow with velocities U = 5 m/s and W = 5 m/s are presented in Fig. 5.

The model of the propellers loads was validated against experimental data from WUT wind tunnel. In this case, the test stand allowed to measure thrust and torque only. The validation was done through correction of the aerodynamic coefficients of the blades airfoils. The selected 15 × 4″ propeller's geometry has been scanned using 3D scanning. The blade was divided into four sections along the span and in each section blade airfoils were assumed to be constant. Using published data for these airfoils, aerodynamic coefficients for these four airfoils were obtained. Then, the lift and drag coefficient were adjusted to match measured thrust and torque. In Figs. 6, 7, 8 and 9 thrust and torque characteristics are shown before and after validation process. The most important case of validation was the test point close to rotor angular velocity 600 rad/s, which was the design hover point for the quadrotor in this research.

This approach to loads correction was also checked in other inflow cases and was proven to give adequate good results.

The adjusted airfoils coefficients were used to calculate all six components of the rotor aerodynamic loads. But implementing the developed models of aerodynamic loads into Blade Element Method did not allow for a real-time computations needed in simulator. The reason was, that integration along blades and iterations of the induced velocities done separately for each rotor, took a long time.

It was decided to develop the simulation model in two steps using so called loads maps. In the first step, the comprehensive model of rotor loads was used to calculate components of aerodynamic loads for an assumed range of inflow velocities (horizontal and perpendicular to the plane of rotation) as a function of rotor angular speed. All components of aerodynamic loads were stored in 3D arrays. These arrays were used in a real-time simulation model. Actual aerodynamic loads of each rotor were interpolated for a given state (rotor angular velocity, inflow velocity, angle of attack, and bank angle). Such a solution allows real time operation of the quadrotor model.

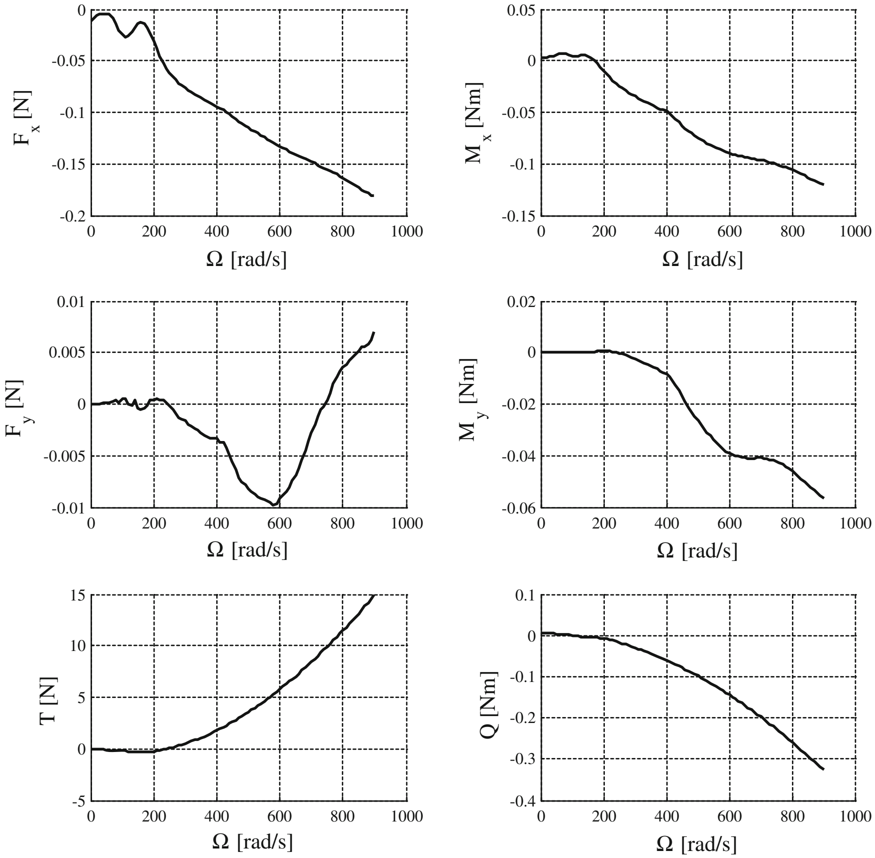


Fig. 5 Loads of a 15 × 4″ propeller, for inflow velocities U = 5 m/s, W = 5 m/s

Fig. 6 Thrust characteristic before validation

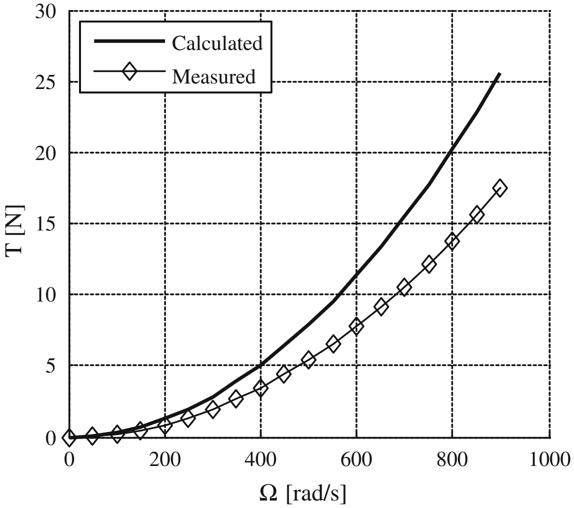


Fig. 7 Torque characteristics before validation

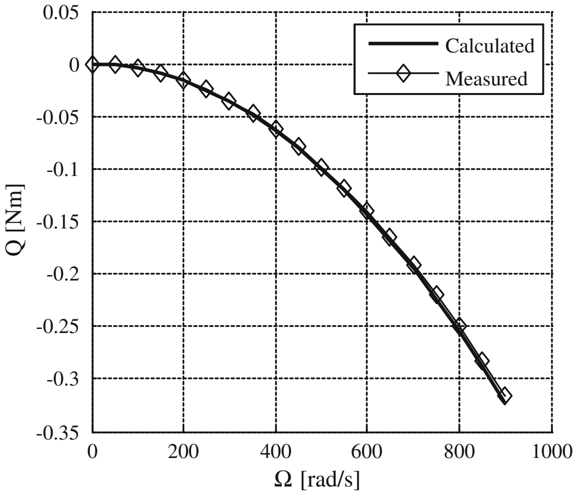
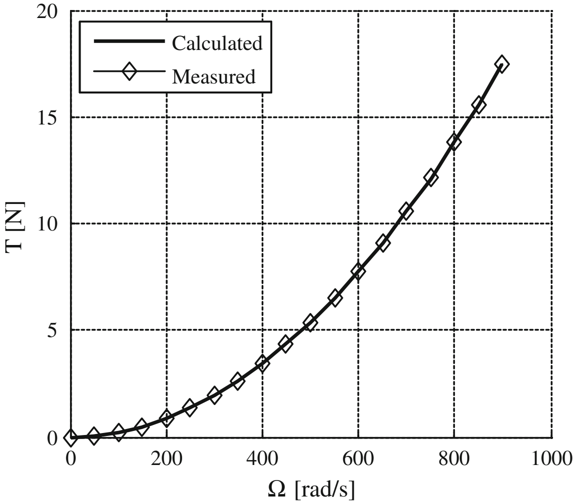
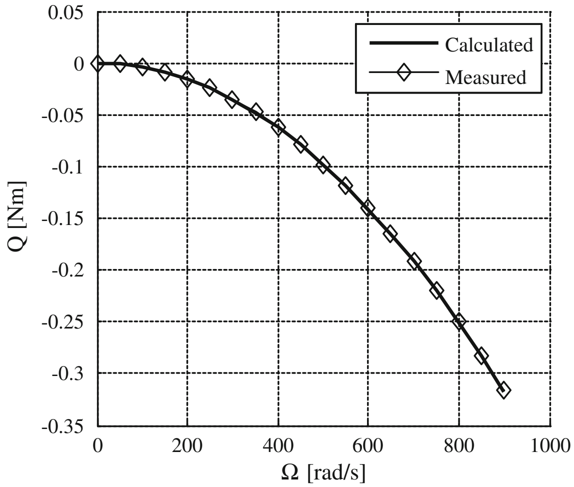


Fig. 8 Thrust characteristics after validation



However, it turned out that in some situations the interpolations also took too much time for real time calculations. To improve the efficiency of the simulation it was decided to use only four components of propellers loads: thrust, drag, and torque and rolling moment, as values of the rest two components were at least one order of magnitude lower. These simplification assured a real-time operation of a quadrotor model with flight control system at all tested flight cases.

Fig. 9 Torque characteristics after validation



3 Quadrotor Flight Control System

The flight control system for unmanned quadrotor consists of ground Control Station and onboard Autopilot, Fig. 10. The Control Station calculates and sends demanded values of controls to on-board autopilot via wireless communication. The Autopilot generates control signals to quadrotor rotors depending on the demanded and measured values of flight parameters. The control signal is organized as a vector **Ω**, which components are the required values of rotors angular velocities.

The control system operates in semiautonomous mode. An operator of a quadrotor sets demanded values of quadrotor linear velocities and azimuth via Control Station. The signals are send to the autopilot, which controls the quadrotor attitude and lift to ensure demanded flight conditions. The longitudinal and lateral velocity are controlled by the pitch and roll angles. The control of flight level is obtained by the vertical speed variation. If the demanded value of vertical speed is equal to zero the altitude controller holds actual flight level.

The overall autopilot structure is shown in Fig. 11. It consists of three main blocks: Velocity Control, Attitude Control and Control Signal Formation. The control signal is a vector of demanded rotors angular rates:

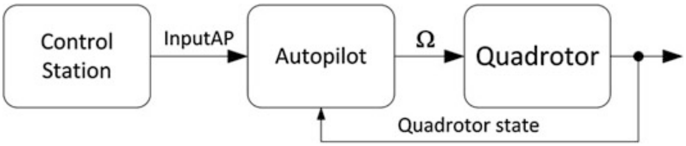


Fig. 10 Quadrotor flight control system structure

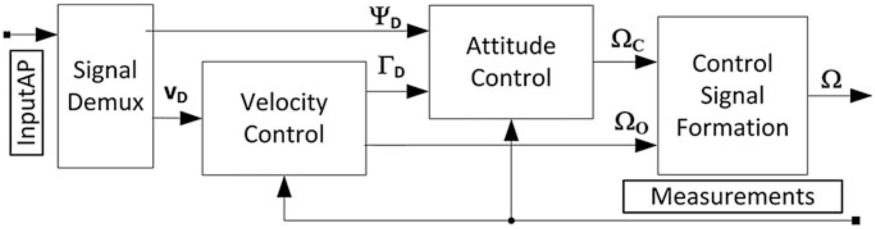


Fig. 11 Autopilot structure

$$\boldsymbol{\Omega} = \Omega_H[1 \quad 1 \quad 1 \quad 1]^T + \boldsymbol{\Omega_D}. \tag{5}$$

The control signal $\boldsymbol{\Omega}$ is calculated as a increment of rotors angular rates $\boldsymbol{\Omega_D}$ relative to hovering condition Ω_H. The inherent feature of quadrotor is that lift and control loads are produced only by changing angular velocities of rotors. The rotor thrust, expressed as function of the rotor angular rate, is divided into lift and attitude parts. The lift part provides the altitude control whereas the attitude part guarantees maneuverability.

The demanded values of rotors angular velocities are calculated in CSF block as:

$$\boldsymbol{\Omega_D} = w_\Omega \cdot \boldsymbol{\Omega_O} + (1 - w_\Omega) \cdot \boldsymbol{\Omega_C}, \tag{6}$$

where $\boldsymbol{\Omega_O}$ is a collective control signal used for altitude control, $\boldsymbol{\Omega_C}$ is a cycling control signal comes from attitude controllers, and a weighting coefficient w_Ω expresses the portion of rotor thrust that is allocated for lift control.

The Attitude Control block contain three independent controllers for pitch, roll and yaw angles. The yaw angle controller is based on PID control law. The pitch and roll controllers have identical PD control laws but the output signals are distributed to the rotors in a different way. The output signals vector of AC block, named cycling control signal, is a sum of output signals of all controllers distributed according to the selected quadrotor configuration:

$$\boldsymbol{\Omega_C} = u_\Phi \begin{bmatrix} -0.33 \\ -0.33 \\ 0.33 \\ 0.33 \end{bmatrix} + u_\Theta \begin{bmatrix} 0.33 \\ -0.33 \\ -0.33 \\ 0.33 \end{bmatrix} + u_\Psi \begin{bmatrix} -0.34 \\ 0.34 \\ -0.34 \\ 0.34 \end{bmatrix}, \tag{7}$$

where u_Φ, u_Θ, u_Ψ are output signals from roll, pitch and yaw controllers.

The Velocity Control block contains controllers for horizontal (longitudinal and lateral) velocities and flight level. The longitudinal and lateral controllers are based on PID control laws with adaptive integral coefficients, and provide demanded values for pitch and roll control signals, which are sent to the Attitude Control block as an $\mathbf{\Gamma_D}$ vector. The flight level controller contains altitude and vertical speed control laws. The classical P altitude control law produces demanded value for PID vertical speed control law. The demanded value of vertical speed can also comes from an operator, when it is not zero. The vertical speed controller provides the collective control vector $\mathbf{\Omega_O}$:

$$\mathbf{\Omega_O} = u_W[1 \quad 1 \quad 1 \quad 1]^T. \tag{8}$$

The tests of quadrotor flight control system were performed using UAV simulator. The tests were focused on capability of indoor flight especially in confined areas such as rooms, corridors and stairways. The tests results revealed significant disturbances of altitude and azimuth during horizontal acceleration and deceleration. The detailed analysis indicated that the disturbances resulted from non uniform and asymmetric variations of thrust of rotors. In the next tests the flight level and azimuth controllers were improved. An additional attitude feedback loop was applied to compensate fluctuations in altitude, which were mainly caused by the rotors inclination angle, Fig. 12. The rotors angular velocity increment $\Delta\omega$ was then calculated using actual rotor attitude gained by factor k_γ to fit with actual controller settings. The constant altitude in a forward flight is possible if a rotor thrust T_F at specific inclination angle γ (Fig. 13) fulfills the condition:

$$T_F \cos(\gamma) = T_H, \tag{9}$$

where T_H is the rotor thrust at hovering conditions. The inclination angle can be estimated as:

$$\cos\gamma = \cos\Phi \cos\Theta. \tag{10}$$

If a rotor thrust is expressed as a square of function of rotor angular rate:

$$T_i = C_T\Omega_i^2, \tag{11}$$

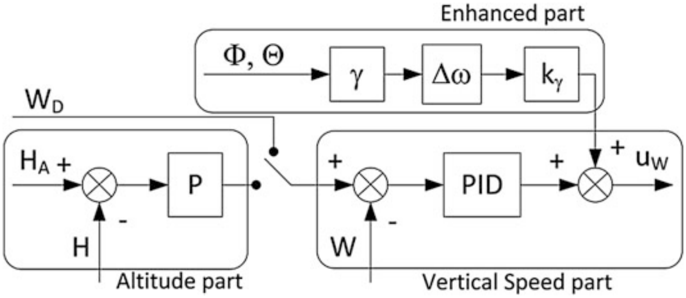


Fig. 12 Enhanced flight level controller structure

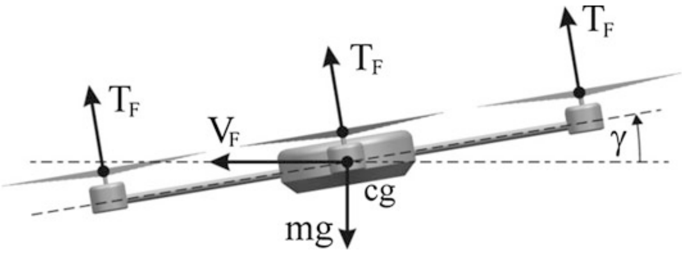


Fig. 13 Rotors thrusts in forward flight

where C_T is thrust constant, and the relationship between rotor angular rates at hover and forward flight conditions are:

$$\Omega_F = \Omega_{\mathrm{H}} + \Delta\Omega, \tag{12}$$

then the Eq. (9) can be rewritten as:

$$(\Omega_{\mathrm{H}} + \Delta\Omega)\cos(\gamma) = \Omega_{\mathrm{H}}. \tag{13}$$

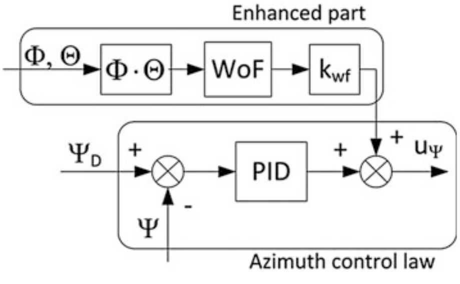
The increment of rotor angular rates ΔΩ for forward flight calculated as:

$$\Delta\Omega = \left(\frac{\sqrt{\cos\gamma}}{\cos\gamma} - 1\right)\Omega_{\mathrm{H}}. \tag{14}$$

The azimuth controller was enhanced by additional attitude feedback loop (Fig. 14). The product of pitch and roll angles with washout filter *WoF* are used to compensate an azimuth disturbances caused by attitude change. The k_{wf} coefficient is used to gain the signal of enhancement part.

The simulation tests of quadrotor flight control system were performed to investigate enhanced controller performance using the quadrotor dynamics model from Chapter "The Method of Guaranteeing the Separation Between the Recognised Object and Background". The maneuver was to accelerate and to

Fig. 14 Enhanced azimuth controller structure



decelerate quadrotor at constant altitude and azimuth. The autopilot was set to hold constant values of flight level and azimuth, while both pitch and roll angles were increased to 45°, and after 10 s decrease to 0 (Figs. 15 and 16). The quadrotor have accelerated to 19 m/s of horizontal velocity at inclination angle of 60° and then without using thrust decelerate to the 0 m/s (Figs. 17 and 18). The quadrotor flight was smoother while the enhancement controllers were applied. The variation of altitude and azimuth were significantly reduced (Figs. 19 and 20). Integral Square Errors (ISE) are much lower for enhancement controllers (Table 1). The overshoots

Fig. 15 Roll angle for basic and enhanced autopilot

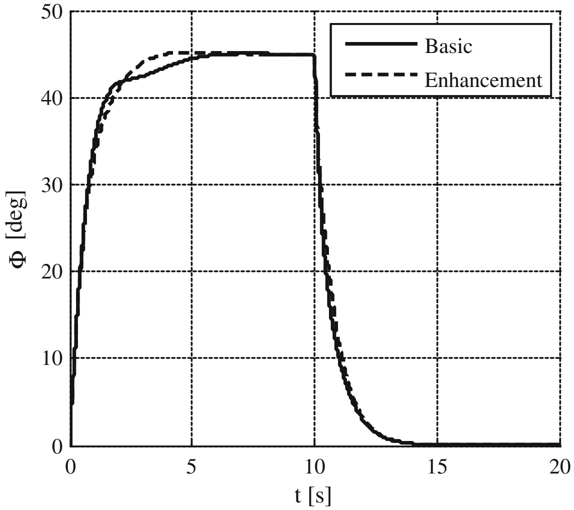


Fig. 16 Pitch angle for basic and enhanced autopilot

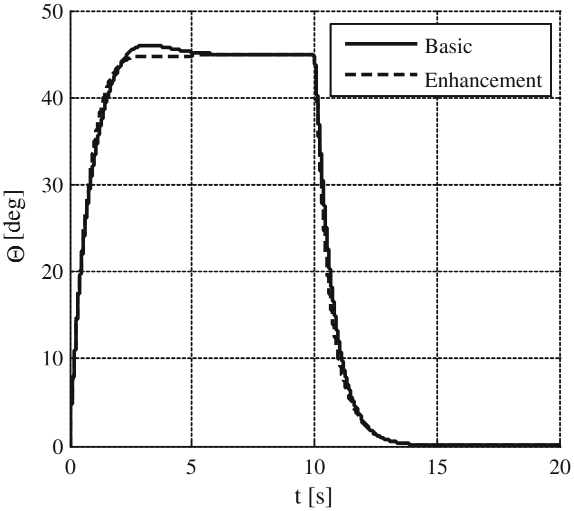


Fig. 17 Horizontal velocity for basic and enhanced autopilot

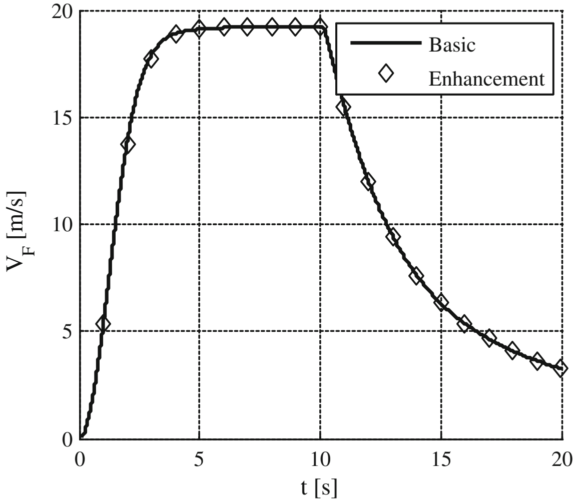
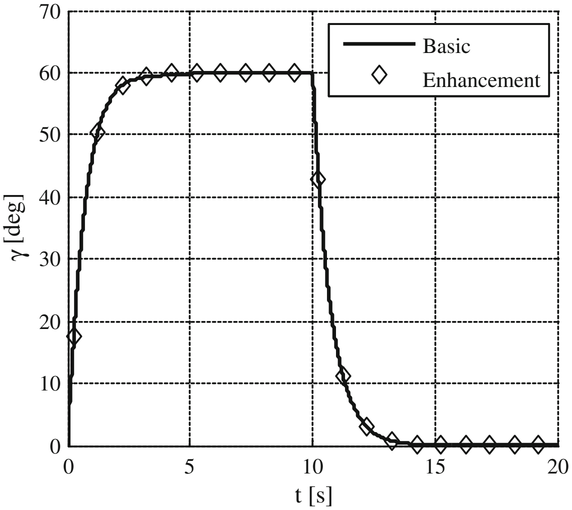


Fig. 18 Inclination angle for basic and enhanced autopilot



and disturbances in pitch and roll angles, and velocities were also compensated. The test results indicate that the proposed enhancements of quadrotor control system considerably improved its flying qualities. Altitude and azimuth disturbances were larger in deceleration than in acceleration flight phase. It resulted from different flight velocity in both cases, which significantly influenced quadrotor performances. That is why an additional flight velocity coupling should be considered to avoid such inconveniences.

Fig. 19 Altitude for basic and enhanced autopilot

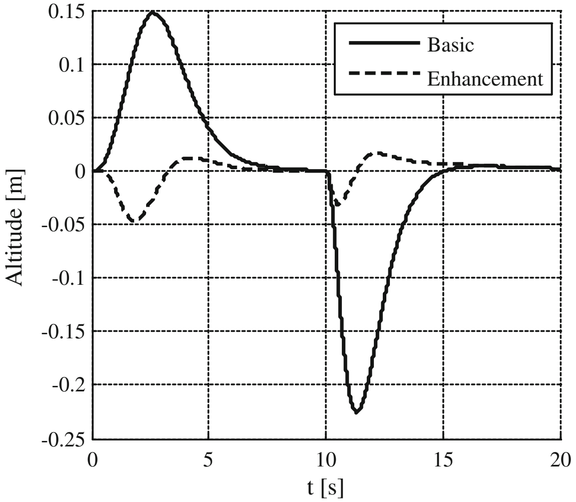


Fig. 20 Azimuth angle for basic and enhanced autopilot

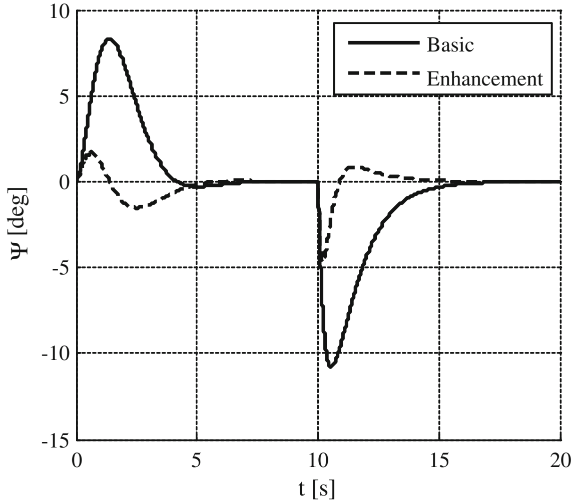


Table 1 Integral square errors

Controlled parameter	Controller type		
	Basic IB	Enhancement IE	IB/IE
Altitude	0.1231	0.0039	31.85
Azimuth	248.87	14.08	17.68

4 Conclusions

A quadrotor model with a flight control system was developed and tested in a UAV simulator. The quadrotor simulation model contains 6 DOF and a comprehensive model of rotor aerodynamic loads, based on experimental data gathered during wind tunnel tests. The deficiencies of a wind tunnel measuring equipment (only thrust and torque might be measured) were overcame by a special correction procedure. A real time simulation capability was obtained by adequate assumptions based on airloads orders of magnitude.

The quadrotor autopilot control system was developed to cooperate with a nonlinear dynamic model and nonlinear rotor aerodynamic loads. The method of PID control laws enhancement was applied based on a simple model of rotor thrust and additional state variables couplings, for compensation of variation of a quadrotor dynamics. The improved control system provides better quadrotor performance for high speed flights and rapid maneuvers.

The nonlinear quadrotor model with autopilot was implemented to the UAV simulator and allows to simulate and investigate the control system quality.

Acknowledgment The presented research was part of the project “Development of small unmanned rotorcraft”, under grant from Polish National Centre for Research and Development, No 0032/R/ID1/2011/01.

References

1. Bartoszek, J., Narkiewicz, J., Zasuwa, M.: Innovative, reconfigurable simulator of mobile robots to support anti-crisis operations. In: 16th International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, pp. 122–126. August 22–25, 2011
2. Milhim, A., Zhang, Y., Rabbath, C-A.: Quad-rotor UAV: high-fidelity modeling and nonlinear PID control. In: AIAA Modeling and Simulation Technologies Conference, Toronto, Ontario, Canada, 2–5 Aug 2010
3. Gałuszka, A., Pacholczyk, M., Bereska, D., Skrzypczyk, K.: Planning as artificial intelligence problem—short introduction and overview. In: Advanced Technologies for Intelligent Systems for National Border Security, pp. 95–103 (2013)
4. Hoffmann, G., Huang H., Waslander S., Tomlin C.: Quadrotor helicopter flight dynamics and control: theory and experiment. In: AIAA guidance, navigation and control conference and exhibit, Hilton Head, South Carolina, 20–23 Aug 2007
5. Castillo, P., Dzul, A., Lozano, R.: Real-time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Trans. Control Syst. Technol. **12**(4), 510–516 (2004)
6. Waslander, S.L., Hoffmann, G.M., Jang, J.S., Tomlin, C.J.: Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. In: Proceedings of the IEEE/RSJ international conference on intelligent robotics and systems 2005, Edmonton, Alberta, August 2005, pp. 468–473
7. Chen, M., Huzmezan, M.: A combined MBPC/2 DOF $H\infty$ controller for a quad rotor UAV. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, 11–14 Aug 2003

8. Wu, J., Peng, H., Chen, Q.: RBF-ARX model-based modeling and control of quadrotor. In: IEEE international conference on control applications part of 2010 IEEE multi-conference on systems and control, Yokohama, Japan, 8–10 Sept 2010
9. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: IEEE/RSJ international conference on intelligent robots and systems, Sendai, Japan, 28 Sept–2 Oct 2004
10. Besnard, L., Shtessel, Y., Landrum, B.: Control of a quadrotor vehicle using sliding mode disturbance observer. In: AIAA guidance, navigation and control conference and exhibit, Hilton Head, South Carolina, 20–23 Aug 2007
11. Galzi, D., Shtessel, Y.: Unmanned rotorcraft tight formation flight control using sliding mode control driven by sliding mode disturbance observers. In: AIAA guidance, navigation and control conference and exhibit, Honolulu, Hawaii, 18–21 Aug 2008
12. Pollini, L., Metrangolo, A.: Simulation and robust backstepping control of a quadrotor aircraft. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Honolulu, Hawaii, 18–21 Aug 2008
13. Whitehead, B., Bieniawski, S.: Model reference adaptive control of a quadrotor UAV. In: AIAA guidance, navigation, and control conference, Toronto, Ontario, Canada, 2–5 Aug 2010
14. Liao, F., Lum, K-Y., Wang, J.: Flight control of VTOL quadrotor aircraft via output feedback. In: AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada, 2–5 Aug 2010
15. Kuś, Z., Fraś, S.: Helicopter Control Algorithms from the set orientation to the set geographical location, Advanced Technologies for Intelligent Systems of National Border Security, pp. 3–13, 2013
16. Michini B., How J.: L1 Adaptive Control for Indoor Autonomous Vehicles: Design Process and Flight Testing. AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois, 10–13 Aug 2009
17. Kendoul, F., Lara, D., Fantoni-Coichot, I., Lozano, R.: Real-time nonlinear embedded control for an autonomous quadrotor helicopter. J. Guidance Control Dyn. **30**(4), 78–84 (2007)
18. Zhang, X., Zhang, Y.: Fault tolerant control for quad-rotor UAV by employing Lyapunov-based adaptive control approach. In: AIAA guidance, navigation, and control conference, Toronto, Ontario, Canada, 2–5 Aug 2010
19. Jedrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE conference on industrial electronics and applications (ICIEA), pp. 403–408. ISBN: 978-1-4673-6320-4, 19–21 June 2013
20. Żugaj, M., Narkiewicz, J.: Autopilot for reconfigurable flight control system. ASCE J. Aerosp. Eng. **22**(1), 78–84 (2009)
21. Bramwell, A.R.S., Done, G., Balmford, D.: Bramwell's Helicopter Dynamics, 2nd edn, pp. 77–102. Butterworth-Heinemann/Linacre House, Jordan Hill/Oxford (2001)

The Application of the Modified BLT Method for the Synthesis of UAV's Control System

Zygmunt Kuś and Aleksander Nawrat

Abstract The following paper presents of the use of the modified BLT method for the synthesis of the control system of the UAV. The small helicopter was the example of the UAV. This method was used for controlling the linear acceleration in the vertical axis and angular velocities around the UAV's symmetry axes. The linearized multivariable model was used as the model of the UAV. The presented examples illustrate the correct operation of the modified BLT method in the lowest level of the UAV's control system.

Keywords BLT method · Multivariable control systems · Helicopter model

1 Introduction

This paper is devoted to the synthesis of the multivariable control system for the controlled plant (UAV). The UAV's control system always requires several control levels. The number of the levels depends on the assumed degree of the UAV's flight autonomy. The literature provides various solutions of the abovementioned problem [1–6].

The subsequent control levels may serve various functions: on the highest control level it is e.g. non-collision trajectory or other tasks connected with a mission; on the middle control level it is guaranteeing that the UAV follows the set trajectory with the set orientation and velocity; and on the lowest control level it is setting a required angle for the rotor blades and a set angular velocity of the rotor. In the case of different solutions of the abovementioned problem, there may also appear other intermediate levels. In order to carry out simulations [7–10], we will

Z. Kuś (✉) · A. Nawrat
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16 Street, Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_20

take into account a particular kind of a UAV such as a small helicopter. It is a ‘miniature version’ of a conventional helicopter with a main rotor and tail rotor.

It is a common knowledge that during the UAV’s flight (in this case it is a model of a small or miniature helicopter) one can distinguish different flight conditions. Depending on the flight condition in a given moment (hover, forward flight and autorotation), it is essential to set different sets of set values of the signals which determine orientation and angular velocities for particular axes.

Among various methods of the synthesis of the multivariable control systems it can be distinguished for example the following methods: Ziegler-Nichols method [11], BLT method [12], Dominant Pole Placement Tuning method [13], LQ method [14], Davison method [15] or decoupling method [16].

The following study examines the control system of the lowest control level for the angular velocities around the symmetry axes and for the linear acceleration in the vertical axis. In order to synthesise the multivariable control system, we will use the modified BLT method [12].

2 The Linear Multivariable Model for the UAV

In each multivariable model the first stage of developing the model consists of determining the main and coupling channels. We will examine the model of a helicopter-type UAV. The extensive body of literature on helicopters’ behaviour provides us with a detailed knowledge about particular control channels [1–6]. Assuming that we have four input signals (the inputs of the multivariable model) and four controlled signals (the outputs of the multivariable model), we have to choose appropriate transfer functions for each control channel.

The last stage of developing the model consists of selecting the parameters’ values for the transfer functions in couple channels.

The considered UAV (helicopter) may be modelled as a system with four inputs and four outputs on the lowest control level. Figure 1 presents input signals and output signals of the helicopter model [17–19].

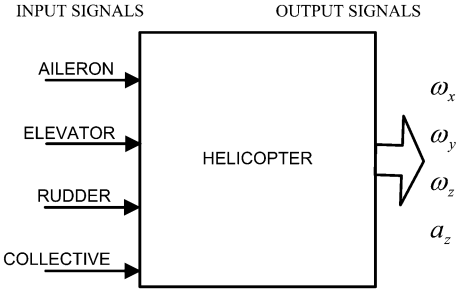


Fig. 1 Input signals and output signals of the helicopter model

We can define three coordinate axes which are connected with the helicopter: longitudinal *X*, lateral *Y* and vertical *Z*. There are defined three angular velocities around *X*, *Y*, *Z* axes—$\omega_x, \omega_y, \omega_z$.

We assume the following signal notation:

Input signals:

$$\text{AILERON} = U_{AI}, \text{ELEVATOR} = U_{EL}, \text{RUDDER} = U_{RU}, \text{COLLECTIVE} = U_{CO}.$$

Output signals:

$$\text{Angular velocities: } \omega_x, \omega_y, \omega_z,$$

Helicopter linear acceleration in vertical axis: a_z.

The helicopter is a multidimensional and non-linear plant. Therefore in the synthesis of the control system the knowledge of the main and coupling channels was used [1–6]. The synthesis of the first control level uses the identification of the control channels.

We used the identification method based on a step response [1].

The analysis of the step response and behaviour of the helicopters presented in caused that we assume the transfer functions for the specific channels presented in (1).

$$\begin{aligned} &G_{11}(s) = \frac{1}{0.05991 * s^2 + 0.1672 * s + 11.7} \quad G_{22}(s) = \frac{1 + 0.7641 * s}{0.1161 + 4.124 * s + 0.5758 * s^2 + 0.03567 * s^3} \\ &G_{33}(s) = \frac{1}{0.3427 + 0.804 * s} \qquad G_{44}(s) = \frac{0.111}{1 + 0.1 * s} \\ &G_{ii}(s) = \frac{0.1}{0.01s + 1} \text{ for all } i \neq j \quad \text{where } i, j = 1, 2, 3, 4. \end{aligned} \tag{1}$$

The scheme of the multivariable control system is presented in Fig. 2.

In Fig. 2, we assume the notation of the input/output signals for the particular control channels according to (2).

$$\begin{array}{llll} w_1 = U_{AI}; & w_2 = U_{EL}; & w_3 = U_{RU}; & w_4 = U_{CO}; \\ y_1 = \omega_x; & y_2 = \omega_y; & y_3 = \omega_z; & y_4 = a_z; \end{array} \tag{2}$$

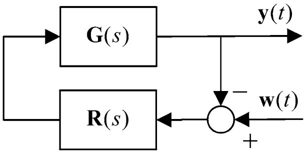


Fig. 2 Multivariable control system. ***R***(*s*)—diagonal multivariable controller, ***G***(*s*)—multivariable plant, ***w***(*t*)—set values vector, ***y***(*t*)—outputs vector

Figures 3, 4 and 5 present the illustrations of methods of tuning the PI controllers in order to obtain the previous parameters of the PI controllers. These parameters, obtained by means of different methods, are modified in the following steps of BLT method. The calculation of the coefficient F is the part of the BLT method which guarantee that properties of all control channels are taken into account during synthesis of the control system.

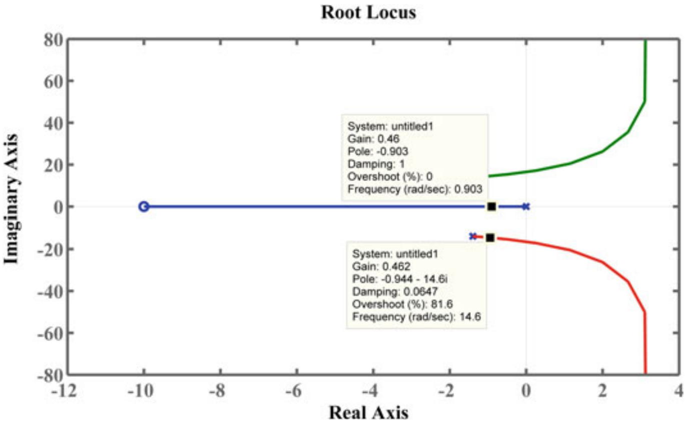


Fig. 3 Root locus for $G_{11}(s)$ with PI controller—illustration of the method guaranteeing an appropriate stability degree

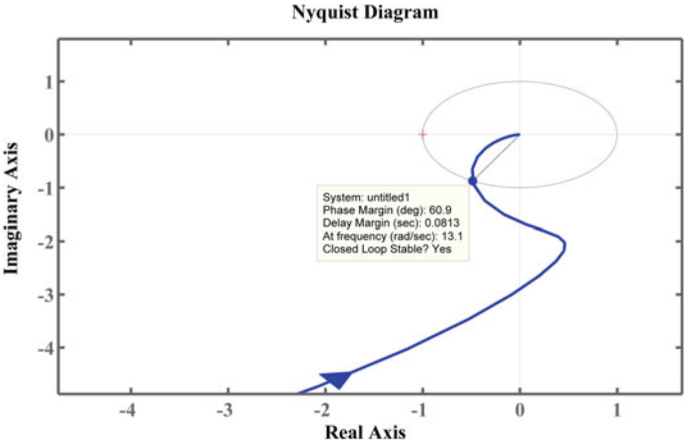


Fig. 4 Nyquist diagram for $G_{22}(s)$ with PI controller—illustration of the method guaranteeing an appropriate phase margin

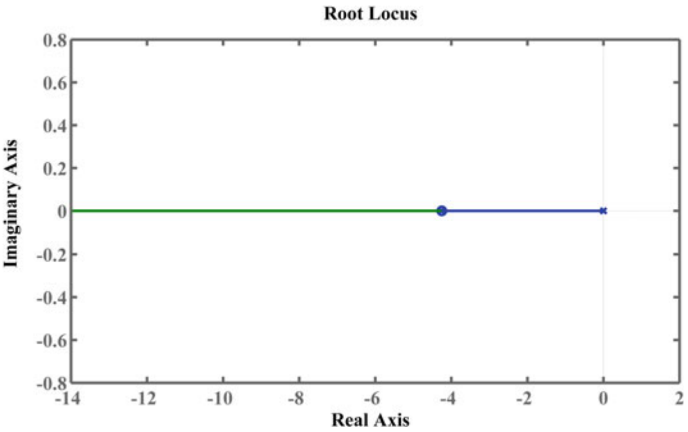


Fig. 5 Root locus for $G_{33}(s)$ with PI controller—illustration of the method of reducing the main time constant

3 An Example of the Control System Operation

In order to illustrate the proposed modification of the BLT method [12] for controlling the UAV, we can assume that the controlled plant is the lowest control level of the small helicopter described in [1]. For the model presented in Sect. 2, we conducted the synthesis of the multivariable regulator which resulted in obtaining the set of PI regulators (3).

$$\begin{array}{ll} PI_{11}(s) = \dfrac{0.2499s + 0.071}{3.52s} & PI_{22}(s) = \dfrac{5s + 0.2841}{17.6s} \\ PI_{33}(s) = \dfrac{2.353s + 0.2841}{8.282s} & PI_{44}(s) = \dfrac{s + 0.2841}{3.52s} \end{array} \tag{3}$$

Figure 6 presents the transients of the output y_1 in response to the changes of the subsequent set values w_1 to w_4.

The subsequent Figs. 7, 8 and 9 present the transients analogous to Fig. 6 for the subsequent y_2 to y_4. Each figure presents the transient of $y(t)$ for one channel in response to the step changes of the set value signals w_1 to w_4. The signals w_1 to w_4 appear one by one with the interval that allows to reach the steady state by the signal $y(t)$.

According to Figs. 6, 7, 8 and 9 we can observe that the signal y_1 achieve its set value considerably slower than other outputs y_i. At the same time there is slight influence of signals w_2 to w_4 on signal y_1.

Figure 7 shows that the step change of signal w_2 results in overshoot in the output y_2. The other set value signals do not influence on signal y_2.

According to Fig. 8 we can observe that y_3 achieves set value in short time without any overshoots. Figure 9 shows that y_4 changes slowly but without

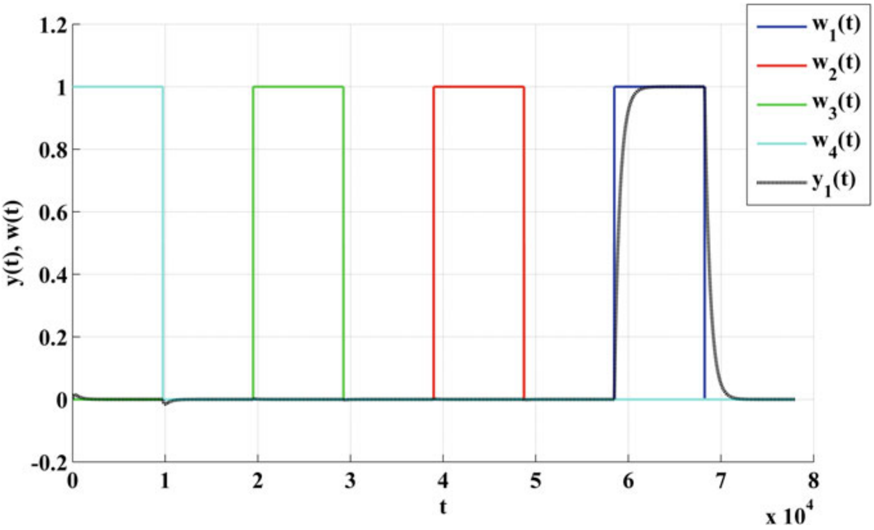


Fig. 6 The transients of the output signal y_1 at the subsequent step changes of the set values w_1 to w_4

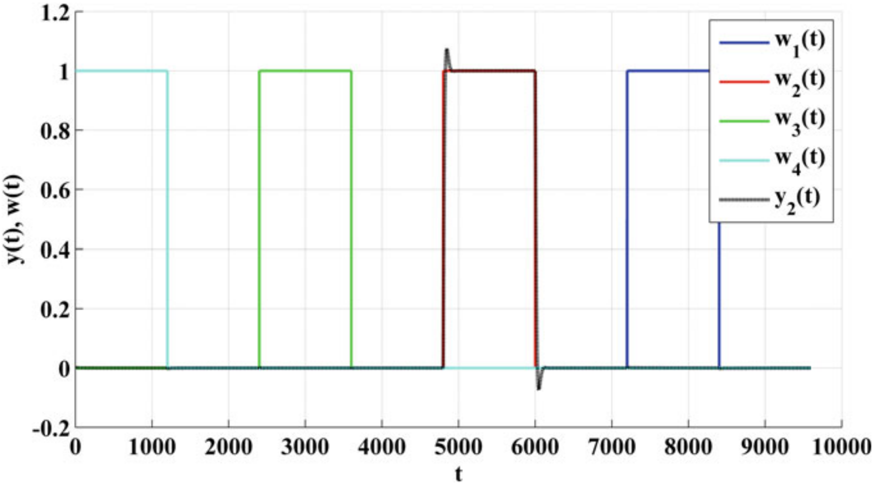


Fig. 7 The transients of the output signal y_2 at the subsequent step changes of the set values w_1 to w_4

overshoots in response to the step change of w_4. We can see there also some influence of the signal w_1 on output y_4.

The steady state error is equal zero in each channel because of the integrator in the PI controller.

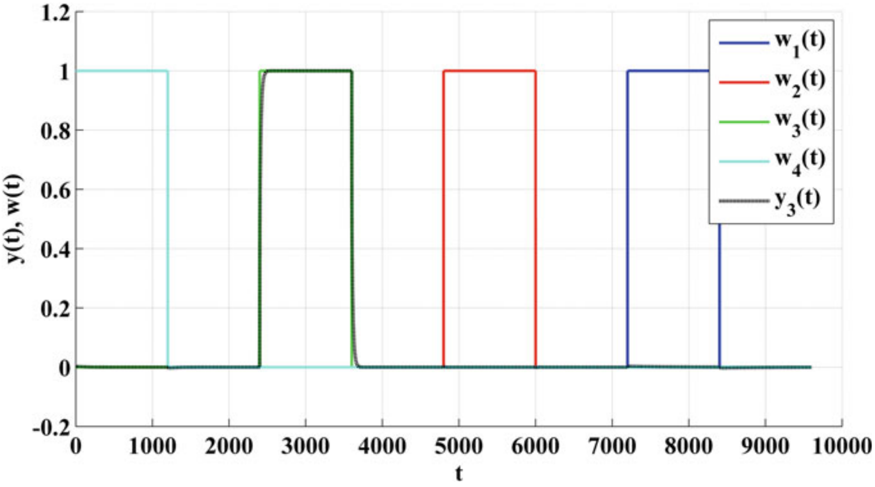


Fig. 8 The transients of the output signal y_3 at the subsequent step changes of the set values w_1 to w_4

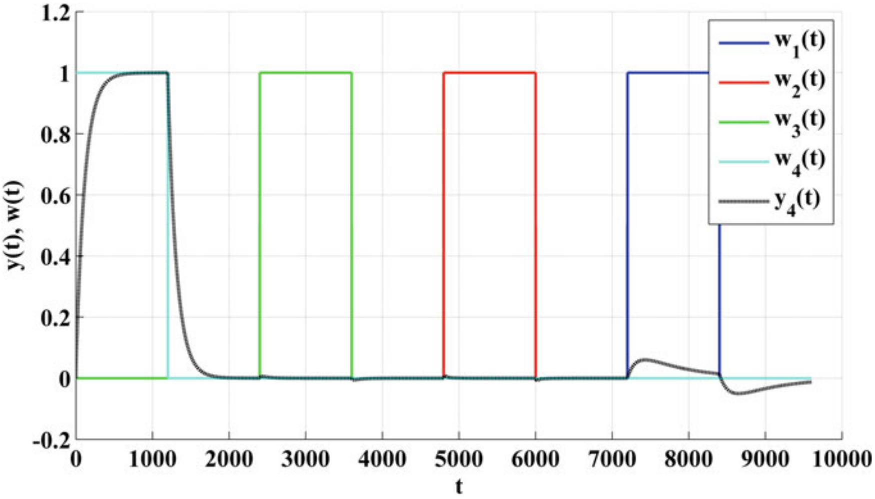


Fig. 9 The transients of the output signal y_4 at the subsequent step changes of the set values w_1 to w_4

The presented example illustrates the correct operation of the control system tuned with the use of the modified BLT method [20].

The transients y_1 to y_4 in Fig. 10 possess different dynamical properties. We can observe maximum overshoot in y_2 and smaller overshoot in y_4 whereas y_1 and y_3 have no overshoots. All outputs $y_i(t)$ achieve the set value (error equal zero) in acceptable time.

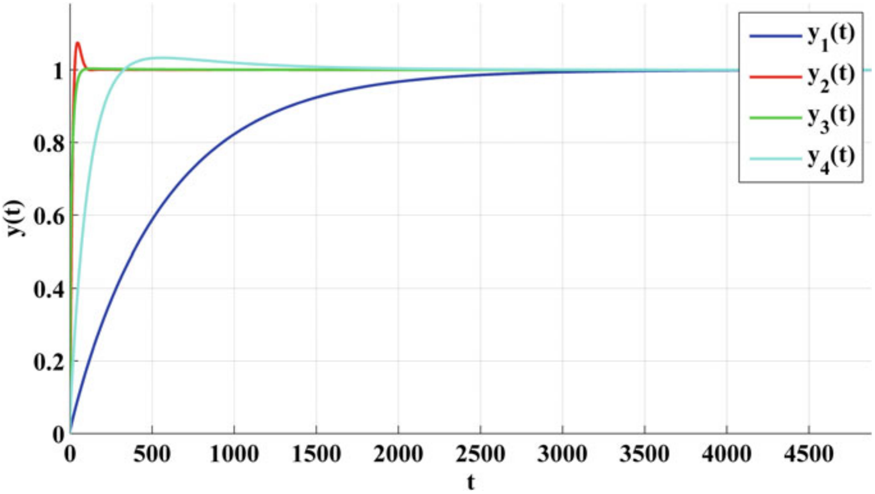


Fig. 10 The transients of output signals for all control channels (y_1 to y_4) in response to the step change of the set value $w(t)$ in each channel $(w_1(t) = w_2(t) = w_3(t) = w_4(t) = 1(t))$

To sum up, the proposed modified BLT method which was applied in the UAV's control system allows to obtain the correct behaviour of the lowest level of the control system.

4 Conclusions

This paper has focused on the problem of UAV control on the lowest control level [21–24]. This means that the task of the control system was to keep the set values of angular velocities around the UAV's symmetry axes and to keep the set value of linear acceleration in vertical axis. The paper presented the use of the modified BLT method for synthesising the control system for the UAV (a small helicopter).

The authors proposed the four-variable linearised model as a model of the UAV. Moreover, we presented the example of the control system operation for the step-changing set values. The presented example illustrates the correctness of the modified BLT method applied to the lowest control system [25] in the UAV [26].

References

1. Kuś, Z., Fraś, S. Helicopter control algorithms from the set orientation to the set geographical location. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, p. 440. Springer, Berlin (2013)
2. Nawrat, A.: Modelowanie i sterowanie bezzałogowych obiektów latających. Wydawnictwo Politechniki Śląskiej, Gliwice (2009)

3. Valavanis, K.P. (ed.): Advances in Unmanned Aerial Vehicles. Springer, Berlin (2007)
4. Castillo, P., Lozano, R., Dzul, A.E.: Modelling and Control of Mini-Flying Machines. Springer, Berlin (2005)
5. Gareth, D.: Padfield, Helicopter Flight Dynamics. Wiley, New York (1996)
6. Manerowski, J.: Identyfikacja modeli dynamiki ruchu sterowanych obiektów latajacych. WN ASKON, Warszawa (1999)
7. Sobel, D., Kwiatkowski, J., Ryt, A., Domzal, M., Jedrasiak, K., Janik, L., Nawrat, A.: Range of motion measurements using motion capture data and augmented reality visualization. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 8671. pp. 594–601 (2014)
8. Babiarz, A., Bieda, R., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 3–25 (2013). ISBN: 978-3-319-00368-9
9. Bereska D., Daniec K., Jedrasiak K., Nawrat A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 115–121 (2013). ISBN: 978-3-319-00368-9
10. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. In: Computer Vision and Graphics, vol. 8671, pp. 302–315. Lecture Notes in Computer Science (2014)
11. Wang, Q.-G., Lee, T.-H., Zhang, Y.: Multiloop version of the modified Ziegler-Nichols method for two input two output processes. Ind. Eng. Chem. Res. **37**, 4725 (1998)
12. Luyben, W.L.: Simple method for tuning SISO controllers in multivariable systems. Ind. Eng. Chem. Process Des. Dev. **25**, 654 (1986)
13. Zhang, Y., Wang, Q.-G., Astrom, K.J.: Dominant pole placement for multi-loop control systems, Automatica **38**(7), 1213 (2002)
14. Stein, G., Athans, M.: The LQG/LTR procedure for multivariable feedback control design. Autom. Control IEEE Trans. **32**, 105 (1987)
15. Davison, E.J.: The robust control of a servomechanism problem for linear time-invariant multivariable systems. Autom. Control IEEE Trans. **21**, 25 (1976)
16. Falb, P.L., Wolovich, W.: Decoupling in the design and synthesis of multivariable control systems. Autom. Control IEEE Trans. **12**, 651 (1967)
17. Bibik, P., Zasuwa, M., Zugaj, M.: Research and training simulator of unmanned quadrotor. In: 18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 403–407 (2013)
18. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guid. Control Dyn. **35**, 1354–1362 (2012)
19. Bibik, P., Narkiewicz, J.: Helicopter modeling and optimal control in autorotation. In: Annual Proceedings-American Helicopter Society, vol. 64, no. 2, pp. 986 (2008)
20. Jedrasiak, K., Nawrat, A., Wydmanska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)
21. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: Design and implementation of mobile ground base station for UGV. In: Innovative Control Systems for Tracked Vehicle Platforms, pp. 57–71 (2014)
22. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical Analysis of IED detonation effect on steel plate. Acta Mechanica et Automatica **6**, 10–12 (2012)
23. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. In: Applied Mechanics and Materials, vol. 249, pp. 1234–1246 (2013)
24. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorbtion during the progressive fracture process. WIT Trans. Model. Simul. **51**, 625–633 (2011)

25. Kwiatkowski, J., Sobel, D., Jedrasiak, K.: FPGA based omnidirectional video acquisition device (OVAD). In: Recent Advances in Electrical Engineering and Computer Science, pp. 58–61 (2014). ISBN: 978-1-61804-249-1
26. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics, vol. 7594, pp. 87–94. Lecture Notes in Computer Science (2012)
27. Fras, S., Jedrasiak, K., Kwiatkowski, J., Nawrat, A., Sobel, D.: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 123–136 (2013). ISBN: 978-3-319-00368-9

The Modified BLT Method for Multivariable Control Systems

Zygmunt Kuś and Aleksander Nawrat

Abstract The goal of the following paper was the synthesis of multivariable control systems. Due to the popularity and usability of the BLT method for tuning such systems, we proposed a certain modification. The BLT method uses the Ziegler-Nichols (ZN) method for tuning the PI regulators for the main channels. The limitations caused by the ZN method for the plant's transfer function form resulted in limiting the use of the BLT method. Therefore this paper put forwards the modification based on using the tuning methods different than ZN. This paper shows the examples of using different tuning methods than ZN which were adjusted to the form of the plant's transfer function. Such an approach allowed to extend the range of the acceptable kinds of the plant's transfer functions in the main channels during the synthesis of the multivariable control system. It allowed to use the knowledge of single-variable control systems for tuning the multivariable control systems. The example presented in this paper illustrates the usefulness of such an approach.

Keywords BLT method · Multivariable control systems

1 Introduction

The multivariable control systems appear in the case when the particular control channels cannot be treated as not influencing each other—separated control channels. In this case, there are couple channels which cause that one input signal of the controlled plant influences on more than one output signal of this plant [1–3].

Z. Kuś (✉) · A. Nawrat
Silesian University of Technology, Institute of Automatic Control,
Akademicka 16 Street, Gliwice, Poland
e-mail: zygmunt.kus@polsl.pl

A. Nawrat
e-mail: aleksander.nawrat@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_21

The body of literature provides the various methods of the synthesis of the multivariable control systems. We can distinguish for example the following methods: Ziegler-Nichols method [4], BLT method [5], Dominant Pole Placement Tuning method [6], LQ method [7], Davison method [8] or decoupling method [9].

The following paper proposes the modification of the BLT method. It is common knowledge that the stages of the BLT method may be grouped in two parts: the first one consists of tuning the PI regulators on the basis of the main channels' transfer function (it is assumed that the ZN method will be used); the second one consists of checking the condition which has to be met by the maximum value of a substitute transfer function module and it consists of tuning a certain coefficient until this condition is met.

Due to the certain limitations of the ZN method, we will propose a modification of the BLT method. This modification is based on the assumption that the parameters of the PI regulator for a given main channel are tuned with the use of any condition (method) which guarantees an appropriate control quality for this main control channel.

2 The Description and Proposed Modification of the BLT Method

The BLT method (Biggest Log—Modulus Tuning) [5] allows to substitute the problem of tuning a single-variable control system for the problem of tuning the multivariable regulators' parameters. It is assumed that the main channels are described by means of stable transfer functions.

Figure 1 presents the block scheme of the multivariable control system. A diagonal multivariable controller consists of transfer functions $K_{ii}(s)$ at the main diagonal and zeros beyond this diagonal. The examined system has the multivariable controller which size is 4 × 4.

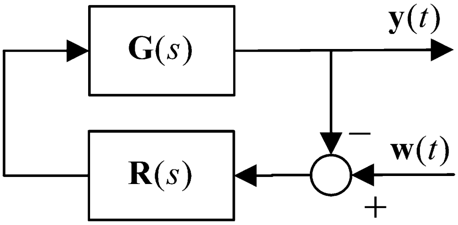


Fig. 1 Multivariable control system. ***R***(*s*)—diagonal multivariable controller, ***G***(*s*)—multivariable plant, ***w***(*t*)—set values vector, ***y***(*t*)—outputs vector

2.1 The Algorithm of the Multivariable Controller Synthesis with the Use of the BLT Method

1. We tune the PI controller parameters for each main channel (not considering the couple channels).
2. We determine the lowest value of the coefficient F for which the control system is stable for the diagonal multivariable controller PI with the parameters calculated according to (1).

$$K_i = \frac{K_{iZN}}{F}; \quad T_{ci} = T_{ciZN} \cdot F; \quad \text{for} \quad i = 1\ldots l \tag{1}$$

where:

K_i, T_{ci} the tuned parameters of the controllers
l the number of single-variable systems which constitute the multivariable plant
K_{iZN}, T_{ciZN} the parameters tuned with the use of the ZN method.

3. We calculate the parameters of the multivariable controller according to (1).
4. We determine the form of the scalar function W(s) according to (2).

$$W(s) = -1 + det(\boldsymbol{I} + \boldsymbol{G}(\mathrm{s})\boldsymbol{R}(s)) \tag{2}$$

where:

$\boldsymbol{G}(s)$ transfer functions matrix for the multivariable system
$\boldsymbol{R}(s)$ transfer functions matrix for the multivariable diagonal controller with the parameters K_i, T_{ci}.

5. We calculate log modulus function according to (3).

$$L_{cm}(j\omega) = 20\log\left|\frac{W(j\omega)}{1 + W(j\omega)}\right| \tag{3}$$

6. We check the criterion (4) of stopping the algorithm.

$$\max L_{cm}(j\omega) = 2l[dB] \tag{4}$$

If (4) is not met then we increase the value of F and return to step 5.

For the lowest value of F which allows to meet (4), we obtain the tuned multivariable controller.

The basic assumption of this modification consists in tuning the PI regulator with the use of the methods allowing for tuning these parameters for the various types of plants. The ZN method used in the BLT method has the limitations resulting from the necessity of calculating ultimate gain and period.

2.2 The Proposed Modification of the BLT Method

Therefore we propose in the first step of the BLT method (according to Sect. 2) not to determine the method of tuning PI regulator. The only requirement we impose is the necessity of tuning the PI regulator, for the main channel, which guarantees an appropriate control quality of this control channel.

The examples of such methods, depending on the plant's transfer function, may be: the method basing on the gain margin and phase margin, the method of reducing the main time constant, root locus method or the ZN method.

3 The Example of the Control System Synthesis

In order to illustrate the proposed modification of the BLT method, we can assume that the controlled plant is four-input and four-output plant. The transfer functions of main channels were chosen in such a way that it allows to illustrate the use of the various methods of tuning the PI regulator.

The first transfer function $G_{11}(s)$ is third order lag with the same time constants. In this case we will use the ZN method.

The second transfer function $G_{22}(s)$ is first order lag with the integrator. In this case we will use the method guaranteeing an appropriate phase margin.

The third transfer function $G_{33}(s)$ is third order lag with different time constants. In this case we will use the method of reducing the main time constant.

The fourth transfer function $G_{44}(s)$ is second order lag with integrator. In this case we will use the root locus method.

All transfer functions which are the components of the multivariable model of the controlled plant are presented in (5).

$$\begin{aligned} G_{11}(s) &= \frac{10}{27s^3 + 27s^2 + 9s + 1} \quad G_{22}(s) = \frac{10}{3s^2 + s} \\ G_{33}(s) &= \frac{10}{27s^3 + 31.5s^2 + 10.5s + 1} \quad G_{44}(s) = \frac{10}{4.5s^3 + 4.5s^2 + s} \\ G_{ii}(s) &= \frac{1}{0.1s + 1} \quad \text{for all } i \neq j \quad \text{where} \quad i, j = 1, 2, 3, 4. \end{aligned} \tag{5}$$

Figures 2, 3 and 4 present the illustrations of abovementioned methods of tuning the PI controllers. The parameters obtained by means of these methods are previous parameters which are modified in the following steps of BLT method.

While tuning F, we met the requirement (4) for F equal 35.2.

Finally, by means of the modified BLT method, we obtain PI controllers for the subsequent channels in the form presented in (6).

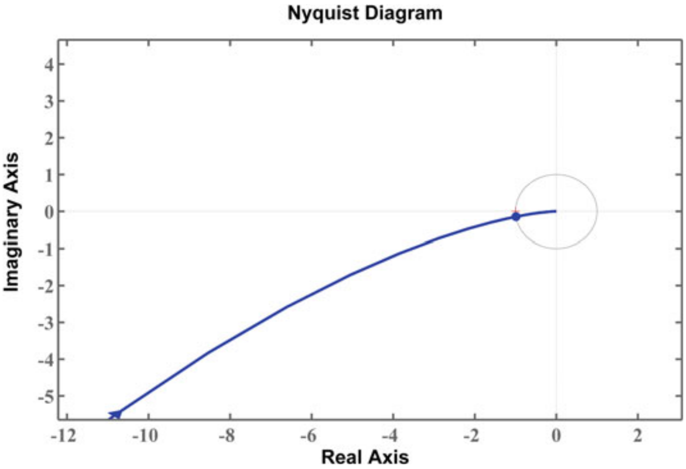


Fig. 2 Nyquist diagram for $G_{22}(s)$ with PI controller—illustration of the method guaranteeing an appropriate phase margin

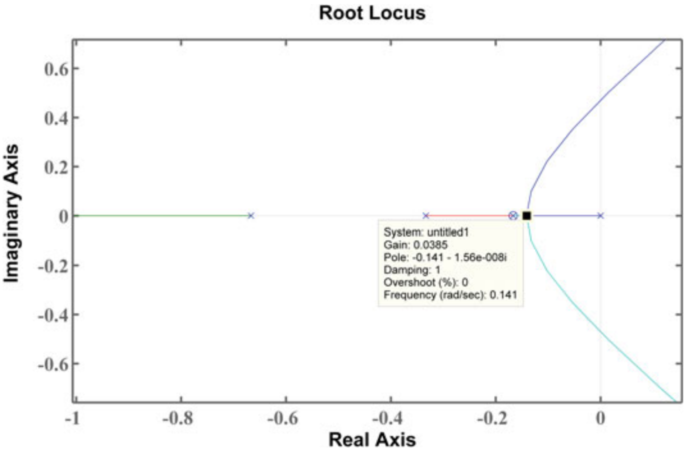


Fig. 3 Root locus for $G_{33}(s)$ with PI controller—illustration of the method of reducing the main time constant

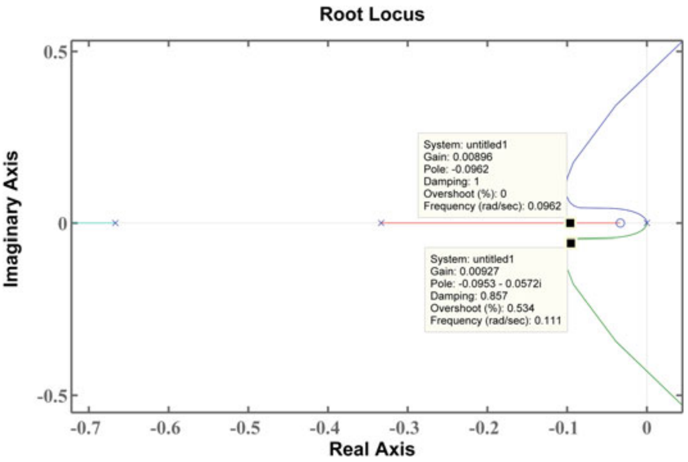


Fig. 4 Root locus for $G_{44}(s)$ with PI controller—illustration of the method guaranteeing an appropriate stability degree

$$PI_{11}(s) = \frac{0.5197s + 0.06209}{8.371s} \quad PI_{22}(s) = \frac{15s + 0.1724}{87s}$$
$$PI_{33}(s) = \frac{0.6766s + 0.08731}{7.749s} \quad PI_{44}(s) = \frac{0.07955s + 0.007759}{10.25s} \tag{6}$$

Figures 5, 6, 7 and 8 present the transients of the output signals for all control channels (y_1 to y_4) in response to the step changes of the subsequent set values w_1 to w_4. Each figure presents the transient of y(t) for one channel in response to the step changes of the set value signals w_1 to w_4. The signals w_1 to w_4 appear one by one with the interval that allows to reach the steady state by the signal $y(t)$.

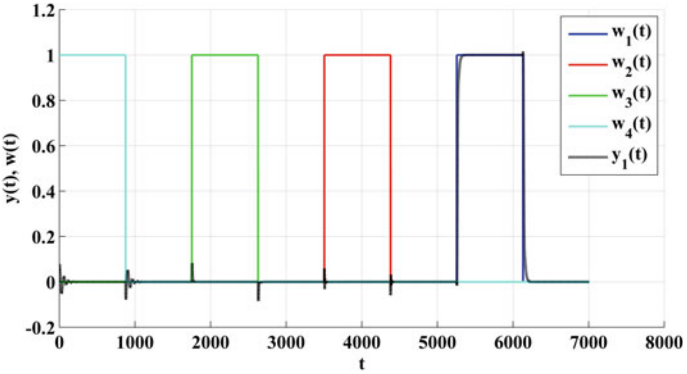


Fig. 5 The transients of the output signal y_1 at the subsequent step changes of the set values w_1 to w_4

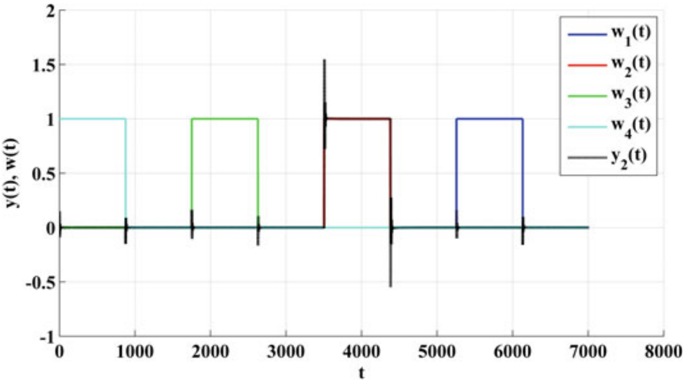


Fig. 6 The transients of the output signal y_2 at the subsequent step changes of the set values w_1 to w_4

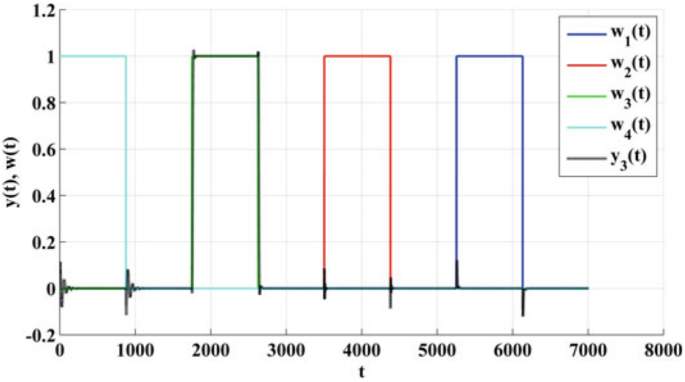


Fig. 7 The transients of the output signal y_3 at the subsequent step changes of the set values w_1 to w_4

According to Figs. 5, 6, 7 and 8, in the transients of the output signals y_2 and y_4 there are some overshoots, but not exceeding the acceptable values. At the same time, we may state that practically there are no overshoots in transients y_1 and y_3. At all output signals $y(t)$, due to using the PI regulators, there is an error signal equal zero in the steady state.

Figure 9 presents the transients of output signals for all control channels (y_1 to y_4) in response to the step change of the set value $w(t)$ in each channel $(w_1(t) = w_2(t) = w_3(t) = w_4(t) = 1(t))$.

The transients y_1 to y_4 in Fig. 9 possess different dynamical properties. We can observe maximum overshoot in y_4 and smaller overshoot in y_2 whereas y_1 and y_3 virtually have no overshoots. All outputs $y_i(t)$ achieve the set value (error equal zero) in acceptable time.

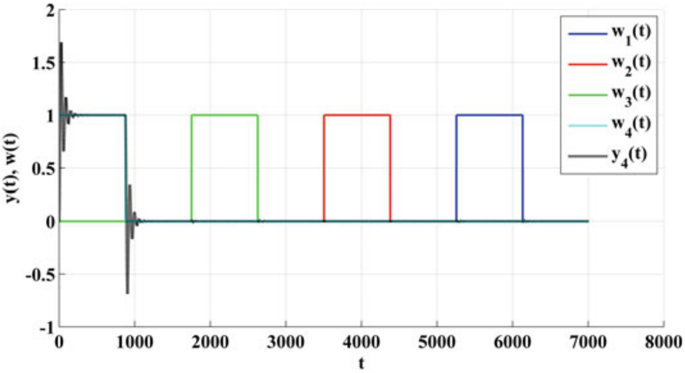


Fig. 8 The transients of the output signal y_4 at the subsequent step changes of the set values w_1 to w_4

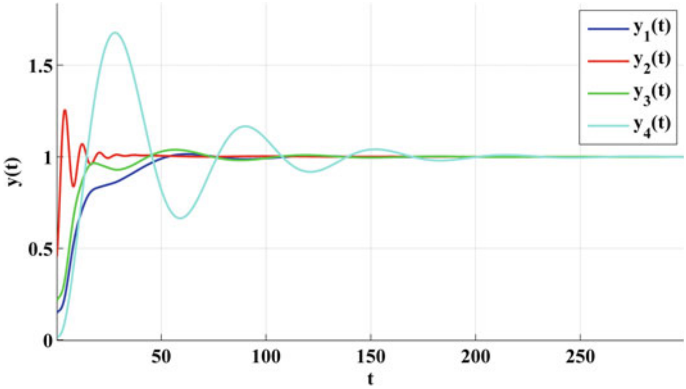


Fig. 9 The transients of output signals for all control channels (y_1 to y_4) in response to the step change of the set value $w(t)$ in each channel $(w_1(t) = w_2(t) = w_3(t) = w_4(t) = 1(t))$

4 Conclusions

This paper proposed the modification of the BLT method which allowed to extend the acceptable range of the transfer functions of the main channels in the synthesis of the multivariable control system [10–12]. It was obtained by allowing the methods different from the ZN method in order to tune the PI parameters [13–16].

Such an approach allows to use the knowledge of single-variable control systems and adjust the PI tuning method to the type of the plant's transfer function in a given control channel [17–19]. The paper presented the example which illustrated the usefulness of such an approach.

References

1. Andrzejczak, M., Ulinowicz, M.: Filtration and integration system (FIS) for navigation data processing Basedon Kalman Filter. In: Intelligent Systems? 2014, pp. 203–2010 (2015)
2. Ulinowicz, M., Narkiewicz, J.: Identification of EMA dynamic model in mechatronics. In: Mechatrocnis, Recent Technological and Scientific Advances, pp. 375–383 (2012)
3. Iwaneczko, P., Jedrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. Comput. Vision Graph. **7594**, 87–94 (2012). (Lecture Notes in Computer Science)
4. Wang, Q.-G., Lee, T.-H., Zhang, Y.: Multiloop version of the modified Ziegler Nichols method for two input two output processes. Ind. Eng. Chem. Res. **37**, 4725 (1998)
5. Luyben, W.L.: Simple method for tuning SISO controllers in multivariable systems. Ind. Eng. Chem. Process Des. Dev. **25**, 654–660 (1986)
6. Zhang, Y. Wang, Q.-G., Astrom, K.J.: Dominant pole placement for multi-loop control systems. Automatica, **38**(7), 1213–1220 (2002)
7. Stein, G., Athans, M.: The LQG/LTR procedure for multivariable feedback control design. IEEE Trans. Autom. Control **32**, 105–114 (1987)
8. Davison, E.J.: The robust control of a servomechanism problem for linear time-invariant multivariable systems. IEEE Trans. Autom. Control **21**, 25–34 (1976)
9. Falb, P.L., Wolovich, W.: Decoupling in the design and synthesis of multivariable control systems. IEEE Trans. Autom. Control **12**, 651–659 (1967)
10. Kus, Z., Fras, S.: Helicopter Control Algorithms from the set orientation to the set geographical location. In: Advanced Technologies for Intelligent Systems of National Border Security, pp. 3–13 (2013)
11. Antosz, P., Bereska, D., Gatys, K., Niedziela, T., Szota, P.: Multilacznik IMPRESJA IQ-element instalacji inteligentnego budynku wykorzystujacej magistrale CAN, Szybkobieńe Pojazdy Gasienicowe, vol. 1, pp. 99–106 (2011)
12. Galuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych. Przeglad Elektrotechniczny **86**, 278–283 (2010)
13. Jedrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. Comput. Vis. Graph. **5337**, 391–399 (2009). (Lecture Notes in Computer Science)
14. Grzejszczak, T., Mikulski, M., Szkodny, T., Jedrasiak, K.: Gesture based robot control. Comput. Vis. Graph. **7594**, 407–413 (2012). (Lecture Notes in Computer Science)
15. Daniec, K., Iwaneczko, P., Jedrasiak, K., Nawrat, A.: Vision based systems for UAV applications. Studies in Computational Intelligence, vol. 481, pp. 219–232. ISBN: 978-3-319-00368-9 (2013)
16. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking. Comput. Vis. Graph. **8671**, 302–315 (2014). (Lecture Notes in Computer Science)
17. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical analysis of IED detonation effect on steel plate. Acta Mechanica et Automatica **6**, 10–12 (2012)
18. Jedrasiak, K., Nawrat, A., Wydmańska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)
19. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorbtion during the progressive fracture process. WIT Trans. Modell. Simul. **51**, 625–633 (2011)

Analysis of Human Arm Nonlinear and Linear Mathematical Models

Robert Bieda and Krzysztof Jaskot

Abstract This paper concentrate on the kinematics and dynamics analysis of human upper arm and extends it to the human manipulator. Kinematics is the study of motion without regard to the forces that create it. The representation of the hand position and orientation through the geometries of arm (joint and link parameters) are called direct kinematics. Using forward kinematics, the mathematical model is developed to compute the position and orientation of arm based on the given human joint position. Each human joint is considered as revolute joint. Dynamics of the arm refers to the interaction between forces in the system and change of state of the system. Based on the dynamic equation of motion of the arm nonlinear and linear model of human upper arm has been defined. The presented models of arm motion has been derived using the Euler-Lagrange equation. In this paper the results of a simulation of the proposed models are presented.

Keywords Human arm · Mathematical model · Kinematics · Dynamics

1 Introduction

Kinematics of the human body is concerned with formulating and solving for the translational and rotational position, velocity, and acceleration analysis problems for each human body segment of interest, for various realworld motions. Forward kinematics calculates the pose (position and orientation) of each human body segment of interest given the joint angles [1–3]. The forward kinematics is the problem of finding an end-effector (hand) or tool pose from a set of given joint angles.

R. Bieda (✉) · K. Jaskot
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: robert.bieda@polsl.pl

K. Jaskot
e-mail: krzysztof.jaskot@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_22

As a result, statics analysis requires the positions and angles of each segment for static free-body diagrams [4–7].

Forward dynamics, a much harder mathematical problem, calculates the unknown kinematics terms given the joint torques; this requires the solution of coupled nonlinear differential equations. Dynamics requires the translational and rotational position, velocity, and acceleration variables for each human body segment, plus the center-of-mass (CoM) translational accelerations, for dynamic free body diagrams [8–10].

In this paper full analysis of the kinematics and dynamics of two mathematical models of human upper arm is presented.

2 The Synthesis of a Mathematical Model of the Human Arm

2.1 Kinematics

Kinematics (forward) is a relation of effector position and orientation relative to the base (inertial) frame depending on the position of the joints. In this case, the kinematics of the human arm is easy to position the orientation of the hand (palm) with respect to the base frame associated with shoulder, depending on the position (rotation) in the shoulder and elbow joints. In this paper model/system of *m* degrees-of-freedom (DoF) consists of *n* rigid bodies called links (rigid limb) will be considered. One of the arm link is characterized by a few values:

l_i the length of the *i*th link,
m_i mass of the *i*th link,
$l_{c,i}$ the distance of the center-of-mass (CoM) of the *i*th link measured from the joint point of reference,
I_i moment of inertia of the *i*th link.

The arm is composed of *m* rotatable pairs (revolute joints). The mobility of the arm joints is characterized by the ability to change the orientation:

φ_i rotation angle around the X axis *i*th link in the *i*th joint,
θ_i rotation angle around the Y axis *i*th link in the *i*th joint,
ψ_i rotation angle around the Z axis *i*th link in the *i*th joint

One end of the chain forming the model arm is connected to the fixed base (e.g. shoulder) while the other end (hand) remains free.

Pose

In the case of the human arm kinematics position of the *i*th link of length l_i with respect to the $(i-1)$th coordinate frame in which it is fixed, is expressed by the vector $\mathbf{l}_i = [x_i \quad y_i \quad z_i]^T$. Similarly, the center-of-mass (CoM) of the *i*th link relative to the $(i-1)$th frame is defined as vector $\mathbf{r}_{c,i} = [x_{c,i} \quad y_{c,i} \quad z_{c,i}]^T$. Accordingly, rotation of the link with respect to *i*th joint is expressed by the rotation angles vector $\boldsymbol{\alpha}_i = [\varphi_i \quad \theta_i \quad \psi_i]^T$ witch defines rotation about the X, Y and Z axis of the $(i-1)$th frame.

As a result, the position of the *k*th link of the arm model relative to the base coordinate frame is determined by relationship:

$$\mathbf{l}_k = \sum_{i=1}^{k} \mathbf{l}_i \tag{1}$$

for $k = 1, 2, \ldots, n$.

Similarly, the pose (CoM position and the orientation) of the *k*th link is defined by dependencies:

$$\mathbf{r}_{c,k} = \sum_{i=1}^{k-1} \mathbf{l}_i + \mathbf{l}_{c,k} \tag{2}$$

$$\boldsymbol{\alpha}_k = \sum_{i=1}^{k} \boldsymbol{\alpha}_i \tag{3}$$

In the case of the kinematic chain of the human arm model we consider pairs of rotating (revolute joints) only. Because of the fixed length of links sliding pairs do not occur. For the purpose of further analysis of the kinematic model variable values (rotations) in the kinematic chain will be replaced by the generalized variables:

$$\mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_m]^T \tag{4}$$

Generalized coordinates are the set of independent coordinates explicitly defining position of the system. The number of generalized coordinates is equal to the number of degrees-of-freedom (DoF) of the system. $\mathbf{q}$ is the vector of joints angles.

On human arm model generalized coordinates describing the motion can be selected by specifying the rotation of the first link with respect to the inertial frame and the other link's rotation relative to the previous link.

Velocities

In the case of analysis of direct kinematics of the arm velocity is defined depending on changes in time the position and orientation of the links. Linear velocity *k*th link of the arm relative to the base can be expressed as follows:

$$\mathbf{v}_k = \sum_{i=1}^{k} \frac{d\mathbf{l}_i}{dt} = \frac{d\mathbf{l}_k}{dt} = J_k^v(\mathbf{q})\dot{\mathbf{q}} \tag{5}$$

where:

$$J_k^v(\mathbf{q}) = \frac{\partial \mathbf{l}_k}{\partial \mathbf{q}} \tag{6}$$

is the (position-dependent) Jacobian matrix of the transformation between Cartesian and joint space. $\dot{\mathbf{q}}$ are the joints velocity vector.

Similarly, the linear velocity of the CoM and the angular velocity of *k*th link are defined by dependencies:

$$\mathbf{v}_{c,k} = \mathbf{v}_{k-1} + \frac{d\mathbf{l}_{c,k}}{dt} = \frac{d\mathbf{r}_{c,k}}{dt} = J_{c,k}^v(\mathbf{q})\dot{\mathbf{q}} \tag{7}$$

$$\boldsymbol{\omega}_k = \sum_{i=1}^{k} \frac{d\boldsymbol{\alpha}_i}{dt} = \frac{d\boldsymbol{\alpha}_k}{dt} = J_k^{\omega}(\mathbf{q})\dot{\mathbf{q}} \tag{8}$$

Velocities of *k*th link can be determined using the above definition of generalized variables (4). Jacobian matrix for the *k*th link describe change the position and orientation expressed in the system of internal coordinates model $\mathbf{q}$:

$$J_k(\mathbf{q}) = \begin{bmatrix} J_{c,k}^v(\mathbf{q}) \\ J_k^{\omega}(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}_{c,k}}{\partial \mathbf{q}} \\ \frac{\partial \boldsymbol{\alpha}_k}{\partial \mathbf{q}} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{q}} x_{c,k}^T \\ \nabla_{\mathbf{q}} y_{c,k}^T \\ \nabla_{\mathbf{q}} z_{c,k}^T \\ \nabla_{\mathbf{q}} \varphi_k^T \\ \nabla_{\mathbf{q}} \theta_k^T \\ \nabla_{\mathbf{q}} \psi_k^T \end{bmatrix} \tag{9}$$

where:

$$\nabla_{\mathbf{q}} = \begin{bmatrix} \frac{\partial}{\partial q_1} & \frac{\partial}{\partial q_2} & \cdots & \frac{\partial}{\partial q_m} \end{bmatrix}^T \tag{10}$$

is a vector operator gradient calculated on the generalized variables of the arm model.

For so defined Jacobian *k*th link velocities can be determined from the relationship:

$$\begin{bmatrix} \mathbf{v}_{c,k} \\ \boldsymbol{\omega}_k \end{bmatrix} = J_k(\mathbf{q})\dot{\mathbf{q}} \tag{11}$$

Acceleration

The last group of signals that describe the kinematics model of the arm are acceleration. Linear acceleration *k*th link of the kinematic chain is defined from the relationship:

$$\mathbf{a}_k = \frac{d\mathbf{v}_k}{dt} = \frac{d^2\mathbf{l}_k}{dt^2} = \frac{dJ_k^v(\mathbf{q})\dot{\mathbf{q}}}{dt} = \frac{dJ_k^v(\mathbf{q})}{dt}\dot{\mathbf{q}} + J_k^v(\mathbf{q})\,\ddot{\mathbf{q}} \tag{12}$$

where:

$$\frac{dJ_k^v(\mathbf{q})}{dt} = \frac{\partial J_k^v(\mathbf{q})}{\partial \mathbf{q}}\dot{\mathbf{q}} = \frac{\partial^2 \mathbf{l}_k}{\partial \mathbf{q}^2}\dot{\mathbf{q}} \tag{13}$$

Similarly, the CoM linear acceleration and angular acceleration *k*th link, are defined as:

$$\mathbf{a}_{c,k} = \frac{d\mathbf{v}_{c,k}}{dt} = \frac{d^2\mathbf{r}_{c,k}}{dt^2} = \frac{dJ_{c,k}^v(\mathbf{q})\dot{\mathbf{q}}}{dt} = \frac{dJ_{c,k}^v(\mathbf{q})}{dt}\dot{\mathbf{q}} + J_{c,k}^v(\mathbf{q})\,\ddot{\mathbf{q}} \tag{14}$$

$$\boldsymbol{\varepsilon}_k = \frac{d\boldsymbol{\omega}_k}{dt} = \frac{d^2\boldsymbol{\alpha}_k}{dt^2} == \frac{dJ_k^\omega(\mathbf{q})\dot{\mathbf{q}}}{dt} = \frac{dJ_k^\omega(\mathbf{q})}{dt}\dot{\mathbf{q}} + J_k^\omega(\mathbf{q})\,\ddot{\mathbf{q}} \tag{15}$$

Using the definition of the Jacobian (9) for *k*th link and dependence (13) accelerations can be determined using generalized variables:

$$\begin{bmatrix} \mathbf{a}_{c,k} \\ \boldsymbol{\varepsilon}_k \end{bmatrix} = \frac{dJ_k(\mathbf{q})}{d\mathbf{q}}\dot{\mathbf{q}}\dot{\mathbf{q}} + J_k(\mathbf{q})\ddot{\mathbf{q}} \tag{16}$$

2.2 *Dynamics*

In the analysis of the dynamics of the arm equations of motion that describe the motion parameters of the arm and the related forces and torques or external forces applied to the system are considered. The dynamics of the arm refers to the interaction between forces in the system and change of state of the system. A torque acting on the joints causes a change in the joint orientation and velocity.

One way to describe the dynamics of manipulators (arms) is to use the Lagrange equations. They describe the dynamic properties of the system, depending on the kinetic and potential energy expressed as a function of the internal (generalized) coordinate.

Kinetic energy

The kinetic equation is the equation for the energy of motion. The kinetic energy of the *k*th link is determined by the relation:

$$K_k = \frac{m_k v_{c,k}^2}{2} + \frac{I_k \omega_k^2}{2} \tag{17}$$

The first component of the above equation is the kinetic energy of translational motion with the velocity of the CoM, the second is the kinetic energy of rotation.

The total kinetic energy for the system can be found by summing the linear and rotational components for each rigid body. A point of interest is that the rotational kinetic energy of the next link depends on the rotational velocity of itself and the previous link. This is because of how the generalized coordinates were selected:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^{n} K_i = \sum_{i=1}^{n} \left(\frac{m_i \mathbf{v}_{c,i}^T \mathbf{v}_{c,i}}{2} + \frac{I_i \boldsymbol{\omega}_i^T \boldsymbol{\omega}_i}{2} \right) \tag{18}$$

Potential energy

The potential energy equation determines the energy of a rigid body of mass m located in a gravitational field. For the kth link potential energy is determined by the relation:

$$P_k = -m_k g r_{c,k} \tag{19}$$

where $g \approx 9.81\,\mathrm{m/s^2}$ is the gravitational acceleration of the Earth.

The potential energy for the system is quantified by the change in CoM within a gravitational field. Using the Eq. (19), the total potential energies for arm model can be expressed as sum of potential energy for each rigid body:

$$P(\mathbf{q}) = \sum_{i=1}^{n} P_i = -\sum_{i=1}^{n} m_i \mathbf{g}^T \mathbf{r}_{c,i} \tag{20}$$

Lagrangian function

The Lagrangian L of arm system is the representation of a system of motion. The Lagrangian can only be used when a system is conservative. Derived from Newton's Laws, the Lagrangian says that if you can find the kinetic equation K and the potential equation P of our complex system in terms of general coordinates and their time derivatives then you can find the equations of the motion of the system in terms of generalized coordinates using the Lagrangian.

If we define total kinetic energy K (18) and total potential energy P (20) of the system, we can introduce the concept of the Lagrange function (kinetic potential) in the form:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - P(\mathbf{q}) \tag{21}$$

Euler-Lagrange equation of motion

The Lagrangian is defined as the difference between kinetic and potential energy and is used with the Euler-Lagrange equation as follows:

$$\frac{d}{dt}\left(\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}\right)-\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \mathbf{q}}=\boldsymbol{\tau} \tag{22}$$

where $\boldsymbol{\tau}$ represent the m-dimensional vector of generalized torques produced by muscles on the m joints of a limb, and $\mathbf{q}$ is the resulting joint angle trajectories.

Torque is applied at each of the joints and as a result we obtain m equations, one for each DoF.

Generalized equation of motion

To a good approximation, human arm dynamics can be modeled as the motion of an open kinematic chain of rigid links, attached together through revolute joints, with control torques applied about each joint. Human arm motions can thus be modeled by the same equations used to model revolute robot manipulators:

$$M(\mathbf{q})\ddot{\mathbf{q}}+C(\mathbf{q},\dot{\mathbf{q}})+G(\mathbf{q})=\boldsymbol{\tau} \tag{23}$$

where $\mathbf{q}$ are the joint angle of the arm. The $M(\mathbf{q})$ is a (position-dependent) symmetric and positive definite inertia matrix. This ensures that the mass is always positive and real valued. The vector $C(\mathbf{q},\dot{\mathbf{q}})$ contains the Coriolis and centripetal torques and the vector $G(\mathbf{q})$ contains the gravitational torques. The generalized forcing input $\boldsymbol{\tau}$ represents the control torques applied at each arm joint.

2.3 Nonlinear Model of Arm

The design of the arm dynamics model is based on the system of m differential equations describing the motion of the system in the space of generalized coordinate:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)-\frac{\partial L}{\partial \mathbf{q}}-\boldsymbol{\tau}=\mathbf{0} \tag{24}$$

or in the form of generalized systems manipulating robots:

$$M(\mathbf{q})\ddot{\mathbf{q}}+C(\mathbf{q},\dot{\mathbf{q}})+G(\mathbf{q})-\boldsymbol{\tau}=\mathbf{0} \tag{25}$$

In the general case determination of changes in the value of generalized coordinates reduced to the solution of the equation:

$$F(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\boldsymbol{\tau})=\mathbf{0} \tag{26}$$

In the present case, the model of the human arm, having regard to the form of the equations of motion (25), angular acceleration of joints determines the relationship:

$$\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}) = M(\mathbf{q})^{-1}(\boldsymbol{\tau} - C(\mathbf{q}, \dot{\mathbf{q}}) - G(\mathbf{q})) \tag{27}$$

Finally, the form of nonlinear model describing the dynamics of changes in the system of generalized variables is defined as follows:

$$\begin{cases} \ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}) \\ \dot{\mathbf{q}} = \int \ddot{\mathbf{q}} dt \\ \mathbf{q} = \int \dot{\mathbf{q}} dt \end{cases} \tag{28}$$

2.4 Linear Model of the Arm

A common problem in determining the dynamics of the mathematical model is to determine the function $f(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau})$ which is the solution of the Eq. (26). An alternative approach is to define a model of the linear nature. This model provides a significant simplification of the calculation of the cost of approximation properties of the nonlinear model in a range of actions.

Due to the nature of this problem it was decided that the state-space representation was a good way to approach the problem. Thus, the proposed linear model is described by two equations: the equation of state changes and output equation:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases} \tag{29}$$

State, output and control variables

Having selected state-space formulation for representing the model, it is now important to choose the state vector that will better describe the system. For the proposed model of human arm state variables $\mathbf{x}$, signal output $\mathbf{y}$ and control $\mathbf{u}$ are defined as follows:

$$\begin{aligned} \mathbf{x} &= [\mathbf{q} \quad \dot{\mathbf{q}}]^T \\ \mathbf{y} &= \ddot{\mathbf{q}} \\ \mathbf{u} &= \boldsymbol{\tau} \end{aligned} \tag{30}$$

Linearization process takes around an equilibrium point. In the case of a model composed of n rotatable joints there are two equilibrium points: stable and unstable. In the presented system stable equilibrium point corresponds to a situation of free overhang of the human arm.

In the general case, however, we consider the operating point:

$$\boldsymbol{\rho}_0 = [\,\mathbf{q}_0 \quad \dot{\mathbf{q}}_0 \quad \ddot{\mathbf{q}}_0 \quad \boldsymbol{\tau}_0\,]^T \tag{31}$$

For such defined operating point linearization process equations of motion (26) leads to a linear relationship:

$$\left.\frac{\partial F}{\partial \mathbf{q}}\right|_{\boldsymbol{\rho}_0} \Delta\mathbf{q} + \left.\frac{\partial F}{\partial \dot{\mathbf{q}}}\right|_{\boldsymbol{\rho}_0} \Delta\dot{\mathbf{q}} + \left.\frac{\partial F}{\partial \ddot{\mathbf{q}}}\right|_{\boldsymbol{\rho}_0} \Delta\ddot{\mathbf{q}} - \mathbf{I}\Delta\boldsymbol{\tau} = \mathbf{0} \tag{32}$$

Given the initial conditions of the arm system, and bearing in mind that $\mathbf{z} = \Delta\mathbf{z} + \mathbf{z}_0$, for $\mathbf{z} \in \{\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\tau}\}$ the above linear equation of motion can be written in the general form:

$$\mathbf{A}_0\mathbf{q} + \mathbf{B}_0\dot{\mathbf{q}} + \mathbf{C}_0\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{0} \tag{33}$$

where $\mathbf{A}_0 = \left.\frac{\partial F}{\partial \mathbf{q}}\right|_{\boldsymbol{\rho}_0}$ corresponds to joint stiffness, $\mathbf{B}_0 = \left.\frac{\partial F}{\partial \dot{\mathbf{q}}}\right|_{\boldsymbol{\rho}_0}$ represents joints damping (viscosity), and $\mathbf{C}_0 = \left.\frac{\partial F}{\partial \ddot{\mathbf{q}}}\right|_{\boldsymbol{\rho}_0}$ is the inertia matrix.

Taking into account the definition (30) of the state variables, output and control signal the above equation becomes:

$$[\,\mathbf{A}_0 \quad \mathbf{B}_0\,]\mathbf{x} + \mathbf{C}_0\mathbf{y} - \mathbf{u} = \mathbf{0} \tag{34}$$

Finally, from the Eq. (34) can be derived equations of state changes and output of the linear arm model:

$$\begin{cases} \dot{\mathbf{x}} = -\mathbf{C}_0^{-1}\begin{bmatrix} \mathbf{0} & -\mathbf{C}_0 \\ \mathbf{A}_0 & \mathbf{B}_0 \end{bmatrix}\mathbf{x} - \mathbf{C}_0^{-1}\begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix}\mathbf{u} \\ \mathbf{y} = -\mathbf{C}_0^{-1}[\,\mathbf{A}_0 \quad \mathbf{B}_0\,]\mathbf{x} + \mathbf{C}_0^{-1}\mathbf{u} \end{cases} \tag{35}$$

Substituting for the variables of the model in state space signals directly related to the change in the joints of the arm angles and torques, we get a linear model (29):

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \mathbf{A}\begin{bmatrix} \mathbf{q} \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{B}\tau \\ \ddot{\mathbf{q}} = \mathbf{C} + \mathbf{D}\tau \end{cases} \tag{36}$$

where:

$$\begin{array}{ll} \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \multicolumn{2}{c}{\mathbf{C}} \end{bmatrix} & \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{D} \end{bmatrix} \\ \mathbf{C} = \begin{bmatrix} -\mathbf{C}_0^{-1}\mathbf{A}_0 & -\mathbf{C}_0^{-1}\mathbf{B}_0 \end{bmatrix} & \mathbf{D} = \mathbf{C}_0^{-1} \end{array} \tag{37}$$

3 1link-1DoF Arm Model

First proposed variant of the human arm model is very simple. It consists only of a single rigid body (upper arm and forearm plus palm) suspended in the joint with one degree of freedom. The idea of the construction of the proposed model is shown in Fig. 1.

Taking into account the definition of the generalized coordinates in the proposed model the rotation angle of the joint ψ_1 is labeled as variable q_1.

Next the equations that govern the forward kinematics and dynamics of the arm and represent Cartesian or hand state in terms of joint state are presented.

3.1 Kinematics

Pose

Using the definitions of the positions (1) and (2) and orientation (3) the transformation from joint angle to handle position (in subject Cartesian coordinates) is given by:

$$\mathbf{l}_1 = \begin{bmatrix} l_1 s_1 \\ -l_1 c_1 \\ 0 \end{bmatrix} \tag{38}$$

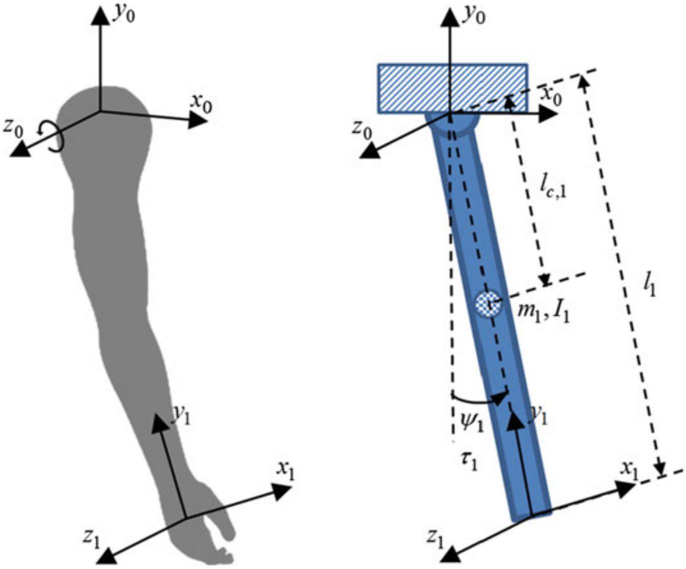


Fig. 1 Diagram of a simple model of human arm by one link and one degree-of-freedom

$$\mathbf{r}_{c,1} = \begin{bmatrix} l_{c,1}s_1 \\ -l_{c,1}c_1 \\ 0 \end{bmatrix} \tag{39}$$

$$\boldsymbol{\alpha}_1 = \begin{bmatrix} 0 \\ 0 \\ q_1 \end{bmatrix} \tag{40}$$

where $s_i = \sin(q_i); c_i = \cos(q_i)$.

Velocity

For such defined position and orientation Jacobian (9) of the arm model can be determined:

$$J_1 = \begin{bmatrix} l_{c,1}c_1 \\ l_{c,1}s_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{41}$$

Using the above Jacobian definition and relationship (11) velocities of the arm can be determined:

$$\begin{bmatrix} \mathbf{v}_{c,1} \\ \boldsymbol{\omega}_1 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 l_{c,1}c_1 \\ \dot{q}_1 l_{c,1}s_1 \\ 0 \\ 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \tag{42}$$

Acceleration

Linear and angular acceleration of the arm in the presented model we determine using the relationship (16):

$$\begin{bmatrix} \mathbf{a}_{c,1} \\ \boldsymbol{\varepsilon}_1 \end{bmatrix} = \begin{bmatrix} -l_{c,1}\left(\dot{q}_1^2 s_1 - \ddot{q}_1 c_1\right) \\ l_{c,1}\left(\dot{q}_1^2 c_1 + \ddot{q}_1 s_1\right) \\ 0 \\ 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix} \tag{43}$$

3.2 Dynamics

In order to analyze the dynamic of this model Lagrangian (21) can be defined as:

$$L(\mathbf{q},\dot{\mathbf{q}}) = 0.5\left(I_1 + m_1 l_{c,1}^2\right)\dot{q}_1^2 + m_1 g l_{c,1} c_1 \tag{44}$$

Euler-Lagrange equation of motion
For so defined Lagrangian function the equation of motion has been set according to the formula (24):

$$0 = \ddot{q}_1 I_1 + \ddot{q}_1 m_1 l_{c,1}^2 + m_1 g l_{c,1} s_1 - \tau_1 \tag{45}$$

Generalized equation of motion
We obtained the generalized equation of motion by converting of Eq. (45) into a generalized form (25) with corresponding coefficients:

$$M(\mathbf{q}) = I_1 + m_1 l_{c,1}^2 \tag{46}$$

$$C(\mathbf{q},\dot{\mathbf{q}}) = 0 \tag{47}$$

$$G(\mathbf{q}) = m_1 g l_{c,1} s_1 \tag{48}$$

3.3 Nonlinear Model of Arm

Nonlinear model presented version of the human arm model is determined by the solution of the Eq. (25) with coefficients (46), (47) and (48):

$$\ddot{q}_1 = M(q)^{-1}(\tau - C(q,\dot{q}) - G(q)) \tag{49}$$

In the form of explicit nonlinear model is defined as:

$$\ddot{q}_1 = \frac{-m_1 g l_{c,1} s_1 + \tau_1}{I_1 + m_1 l_{c,1}^2} \tag{50}$$

which, together with the definition of (28) allows the simulation of the behavior of the arm.

3.4 Linear Model of Arm

Linearized equation of motion (45) takes the form of (33), in which the coefficients are described relations:

$$\mathbf{A}_0 = m_1 g l_{c,1} c_1 \tag{51}$$

$$\mathbf{B}_0 = 0 \tag{52}$$

$$\mathbf{C}_0 = I_1 + m_1 l_{c,1}^2 \tag{53}$$

Form of the linear equation of motion (33) allows the definition of a linear model of the arm described by the relation (29) in which the matrices of state changes and output equation are defined as follows:

$$\begin{array}{ll} \mathbf{A} = \begin{bmatrix} 0 & 1 \\ \frac{-m_1 g l_{c,1} c_1}{I_1 + m_1 l_{c,1}^2} & 0 \end{bmatrix} & \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{I_1 + m_1 l_{c,1}^2} \end{bmatrix} \\ \mathbf{C} = \begin{bmatrix} \frac{-m_1 g l_{c,1} c_1}{I_1 + m_1 l_{c,1}^2} & 0 \end{bmatrix} & \mathbf{D} = \frac{1}{I_1 + m_1 l_{c,1}^2} \end{array} \tag{54}$$

4 1link-2DoF Arm Model

The second of the proposed mathematical model of the arm is a little more complicated. It takes into account the fact that the upper arm consists of one bone and is connected to the shoulder in the gleno-humeral joint (GH-joint). The GH-joint is well modeled as a ball-and-socked joint with 3 DoF. In proposed second model of human arms this GH-joint is modeled as 2 DoF joint. But still, like the first model, this model is constructed from a single rigid body. The idea of the construction of the proposed model is shown in Fig. 2.

Taking into account the definition of the generalized coordinates in the proposed model the rotation angles of the GH-joint (ψ_1, φ_1) are labeled as variables (q_1, q_2).

4.1 Kinematics

Pose

The pose (position and orientation) of the hand in the presented model is described by the following vectors:

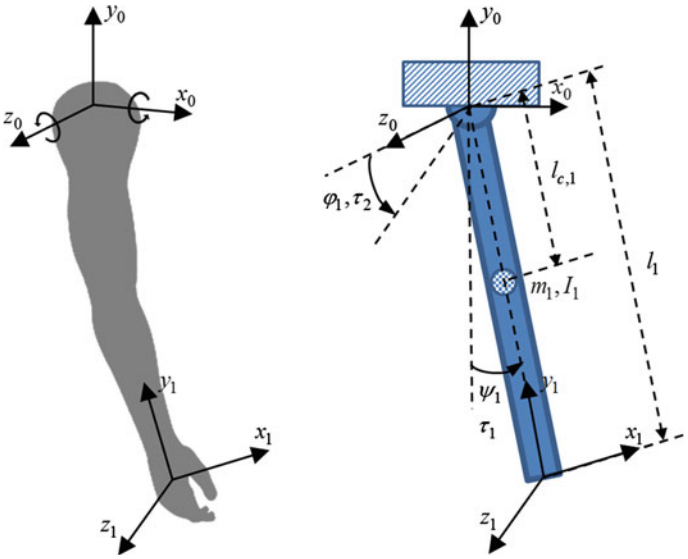


Fig. 2 Diagram of a simple model of human arm by one link and two degree-of-freedom

$$\mathbf{l}_1 = \begin{bmatrix} l_1 s_1 c_2 \\ -l_1 c_1 c_2 \\ -l_1 s_2 \end{bmatrix} \tag{55}$$

$$\mathbf{r}_{c,1} = \begin{bmatrix} l_{c1} s_1 c_2 \\ -l_{c1} c_1 c_2 \\ -l_{c1} s_2 \end{bmatrix} \tag{56}$$

$$\boldsymbol{\alpha}_1 = \begin{bmatrix} 0 \\ q_2 \\ q_1 \end{bmatrix} \tag{57}$$

Velocity

Defining the Jacobian (9) of link for the model of the arm:

$$J_1 = \begin{bmatrix} l_{c,1} c_1 c_2 & -l_{c,1} s_1 s_2 \\ l_{c,1} s_1 c_2 & l_{c,1} c_1 s_2 \\ 0 & -l_{c,1} c_2 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{58}$$

we can determined linear velocity of CoM and angular velocity by the Eq. (11):

$$\begin{bmatrix} \mathbf{v}_{c,1} \\ \boldsymbol{\omega}_1 \end{bmatrix} = \begin{bmatrix} l_{c,1}(c_1 c_2 \dot{q}_1 - s_1 s_2 \dot{q}_2) \\ l_{c,1}(s_1 c_2 \dot{q}_1 + c_1 s_2 \dot{q}_2) \\ -l_{c,1} c_2 \dot{q}_2 \\ 0 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} \tag{59}$$

Acceleration

Using the Jacobian (58) and the dependence (16) we obtain an equation for the acceleration of the arm system:

$$\begin{bmatrix} \mathbf{a}_{c,1} \\ \boldsymbol{\varepsilon}_1 \end{bmatrix} = \begin{bmatrix} -l_{c,1}\left(s_1 c_2 \dot{q}_1^2 + 2\dot{q}_1 c_1 s_2 \dot{q}_2 + s_1 c_2 \dot{q}_2^2 - c_1 c_2 \ddot{q}_1 + s_1 s_2 \ddot{q}_2\right) \\ l_{c,1}\left(c_1 c_2 \dot{q}_1^2 - 2\dot{q}_1 s_1 s_2 \dot{q}_2 + c_1 c_2 \dot{q}_2^2 + s_1 c_2 \ddot{q}_1 + c_1 s_2 \ddot{q}_2\right) \\ l_{c,1}\left(s_2 \dot{q}_2^2 - c_2 \ddot{q}_2\right) \\ 0 \\ \ddot{q}_2 \\ \ddot{q}_1 \end{bmatrix} \tag{60}$$

4.2 Dynamics

We now formulate the Lagrangian using Eqs. (18) and (20):

$$L(\mathbf{q}, \dot{\mathbf{q}}) = 0.5\left(m_1 l_{c,1}^2 c_2^2 \dot{q}_1^2 + I_1 \dot{q}_1^2 + m_1 l_{c,1}^2 \dot{q}_2^2 + I_1 \dot{q}_2^2\right) + m_1 g l_{c,1} c_1 c_2 \tag{61}$$

Euler-Lagrange equation of motion

Then using the Lagrangian formula we compute the equations of motion:

$$\begin{aligned} 0 &= -2\dot{q}_2 m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1 + \ddot{q}_1 m_1 l_{c,1}^2 c_2^2 + \ddot{q}_1 I_1 + m_1 g l_{c,1} s_1 c_2 - \tau_1 \\ 0 &= \ddot{q}_2 m_1 l_{c,1}^2 + \ddot{q}_2 I_1 + m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1^2 + m_1 g l_{c,1} c_1 s_2 - \tau_2 \end{aligned} \tag{62}$$

Generalized equation of motion

In an alternative generalized representation of the above equation assumes the form of Eq. (25), which matrices are defined as follows:

$$M(\mathbf{q}) = \begin{bmatrix} m_1 l_{c,1}^2 c_2^2 + I_1 & 0 \\ 0 & m_1 l_{c,1}^2 c_1^2 + I_1 \end{bmatrix} \tag{63}$$

$$C(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} -\dot{q}_2 m_1 l_{c,1}^2 \dot{q}_1 s_{22} \\ \frac{1}{2} m_1 l_{c,1}^2 \dot{q}_1^2 s_{22} \end{bmatrix} \tag{64}$$

$$G(\mathbf{q}) = \begin{bmatrix} m_1 g l_{c,1} s_1 c_2 \\ m_1 g l_{c,1} c_1 s_2 \end{bmatrix} \tag{65}$$

where $s_{ij} = \sin\left(q_i + q_j\right); c_{ij} = \cos\left(q_i + q_j\right)$.

4.3 Nonlinear Model of Arm

In its present version of the model of human arm solution of the system of differential Eqs. (62) is expressed by the relation:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = M(\mathbf{q})^{-1}\left(\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - C(\mathbf{q},\dot{\mathbf{q}}) - G(\mathbf{q})\right) \tag{66}$$

or in the form of explicit:

$$\begin{aligned} \ddot{q}_1 &= \frac{2\dot{q}_2 m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1 - m_1 g l_{c,1} s_1 c_2 + \tau_1}{m_1 l_{c,1}^2 c_2^2 + I_1} \\ \ddot{q}_2 &= \frac{-m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1^2 - m_1 g l_{c,1} c_1 s_2 + \tau_2}{m_1 l_{c,1}^2 + I_1} \end{aligned} \tag{67}$$

4.4 Linear Model of Arm

Linearized equation of motion of the arm model assumes the form of Eq. (33) with matrices:

$$\mathbf{A}_0 = \begin{bmatrix} m_1 g l_{c,1} c_1 c_2 & \left(2\dot{q}_2 l_{c,1}\dot{q}_1 - 4\dot{q}_2 c_2^2 l_{c,1}\dot{q}_1 - 2\ddot{q}_1 c_2 l_{c,1} s_2 - g s_1 s_2\right) l_{c,1} m_1 \\ -m_1 g l_{c,1} s_1 s_2 & l_{c,1} m_1 \left(-l_{c,1}\dot{q}_1^2 + 2c_2^2 l_{c,1}\dot{q}_1^2 + g c_1 c_2\right) \end{bmatrix} \tag{68}$$

$$\mathbf{B}_0 = \begin{bmatrix} -2\dot{q}_2 c_2 l_{c,1}^2 m_1 s_2 & -2c_2 l_{c,1}^2 m_1 s_2 \dot{q}_1 \\ 2c_2 l_{c,1}^2 m_1 s_2 \dot{q}_1 & 0 \end{bmatrix} \tag{69}$$

$$\mathbf{C}_0 = \begin{bmatrix} c_2^2 l_{c,1}^2 m_1 + I_1 & 0 \\ 0 & l_{c,1}^2 m_1 + I_1 \end{bmatrix} \tag{70}$$

Finally, a linear model of the human arm assumes the form of the system (29) with matrices of state changes and output equations:

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ & \mathbf{C} & & \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \multicolumn{2}{c}{\mathbf{D}} \end{bmatrix} \\
\mathbf{C} &= \begin{bmatrix} \dfrac{-m_1 g l_{c,1} c_1 c_2}{m_1 l_{c,1}^2 c_2^2 + I_1} & C_{12} & \dfrac{2\dot{q}_2 m_1 l_{c,1}^2 c_2 s_2}{m_1 l_{c,1}^2 c_2^2 + I_1} & \dfrac{2 m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1}{m_1 l_{c,1}^2 c_2^2 + I_1} \\ \dfrac{m_1 g l_{c,1} s_1 s_2}{m_1 l_{c,1}^2 + I_1} & C_{22} & \dfrac{-2 m_1 l_{c,1}^2 c_2 s_2 \dot{q}_1}{m_1 l_{c,1}^2 + I_1} & 0 \end{bmatrix} \\
\mathbf{D} &= \begin{bmatrix} \dfrac{1}{m_1 l_{c,1}^2 c_2^2 + I_1} & 0 \\ 0 & \dfrac{1}{m_1 l_{c,1}^2 + I_1} \end{bmatrix}
\end{aligned}
\tag{71}
$$

with:

$$
\begin{aligned}
C_{12} &= m_1 l_{c,1}\left(4\dot{q}_2 l_{c,1} c_2^2 \dot{q}_1 - 2\dot{q}_2 l_{c,1} \dot{q}_1 + 2\ddot{q}_1 l_{c,1} c_2 s_2 + g s_1 s_2\right) / \left(m_1 l_{c,1}^2 c_2^2 + I_1\right) \\
C_{22} &= m_1 l_{c,1}\left(l_{c,1} \dot{q}_1^2 - 2 l_{c,1} c_2^2 \dot{q}_1^2 - g c_1 c_2\right) / \left(m_1 l_{c,1}^2 + I_1\right)
\end{aligned}
$$

5 Results of the Simulation

For the presented models of the human arm simulation analysis of the properties of non-linear models and the corresponding linear models was carried out.

In order to examine the dynamic properties of the models information of the physical features of the individual parts of the human arm was used. According to [11]

Table 1 Anatomical physical parameters of adult human arm

	Male			Female		
	Upper arm	Forearm	Hand	Upper arm	Forearm	Hand
Length (m)	0.315	0.287	0.105	0.272	0.252	0.091
Weight (N)	28.91	16.64	5.78	12.90	6.98	2.22
Mass (kg)	2.947	1.696	0.589	1.315	0.712	0.227
CoM (m)	0.137	0.127	0.049	0.125	0.109	0.043
Moment of inertia (kg m^2)	0.0859	0.0387	0.0020	0.0309	0.0127	0.0006

Table 2 The numerical values of model arm parameters

	1L-1DoF, 1L-2DoF
l_1 (m)	0.707
$l_{c,1}$ (m)	0.313
m_1 (kg)	5.232
I_1 (kg m^2)	0.1266

representative anatomical and physical value of the arm are shown in Table 1. They contain the average values of the arm for an adult male with a height of 1.829 m and a weight of 90.7 kg, and women with a height of 1.575 m and a weight of 45.3 kg.

Based on Table 1 parameter values of individual models of human arm were defined. These parameters are used in simulation studies are presented in Table 2.

The first group of the simulations was to determine the similarity the approximate linear model to the nonlinear model dynamics of the arm. These simulations assumed the free movement of the arm (with no an external force $\boldsymbol{\tau} = \mathbf{0}$) for selected initial conditions different from a stable equilibrium point. Linear models have been defined at operating point equals stable equilibrium point $\boldsymbol{\rho}_0 = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]^T$.

The initial conditions of simulation (initial configuration of the model arm) for the model were as follows:

- model 1link-1DoF: $q_1 = 20,\ \dot{q}_1 = 0,\ \ddot{q}_1 = 0,$
- model 1link-2DoF: $[q_1 \quad q_2] = [20 \quad -10],\ \dot{\mathbf{q}} = \mathbf{0},\ \ddot{\mathbf{q}} = \mathbf{0}.$

For so defined initial conditions the variation of individual internal coordinates of model and their velocity and acceleration are shown in Figs. 3 and 4.

Using the definition of the kinematics of the arm model was possible to determine the change in position of the hand model in space associated with a shoulder. For easier interpretation of the results visualization of the trajectory of the movements of the arm for the analyzed cases is presented. Change of the coordinates position and the course of the hand trajectory are shown in Figs. 5 and 6.

The results confirm the effectiveness of the use of linear models. This approach simplifies the synthesis of model. Properties of the resulting solutions show a high similarity in the behavior of the nonlinear model. They are retained even when operating point is strongly different than point of linearization.

The second group of experiments was checking the properties of linear and nonlinear models in the event of an external force (torque). The results of simulation experiments for the case where the external torques ensured the position of the arm in the initial configuration are presented below.

The initial conditions of simulation were identical to the previous one. Accordingly, the external torques for the model were as follows:

- model 1link-1DoF: $\tau_1 = 5,$
- model 1link-2DoF: $[\tau_1 \quad \tau_2] = [5 \quad -2].$

Change of the coordinates position and the course of the hand trajectory are shown in Figs. 7 and 8.

Fig. 3 Changes in the value of the internal coordinate of model 1link-1DoF

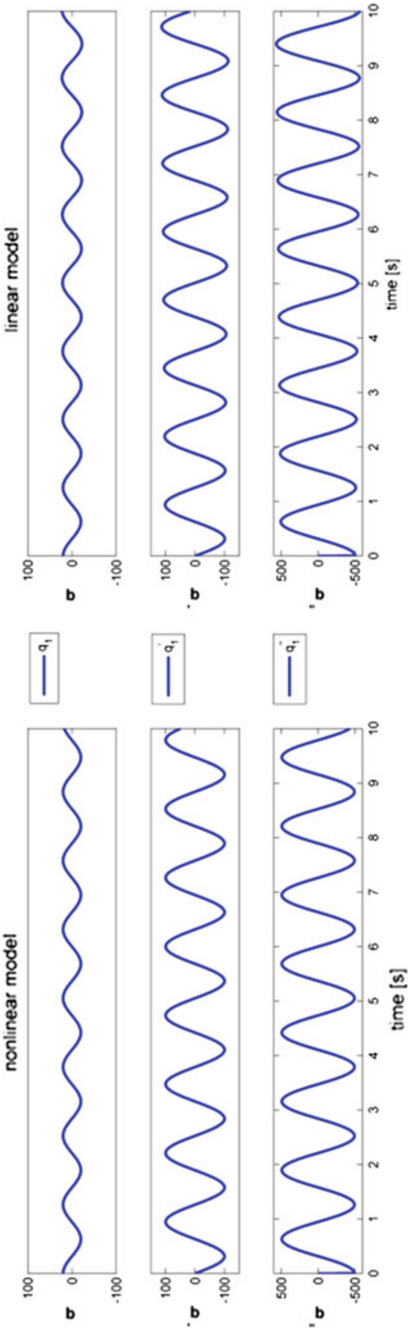
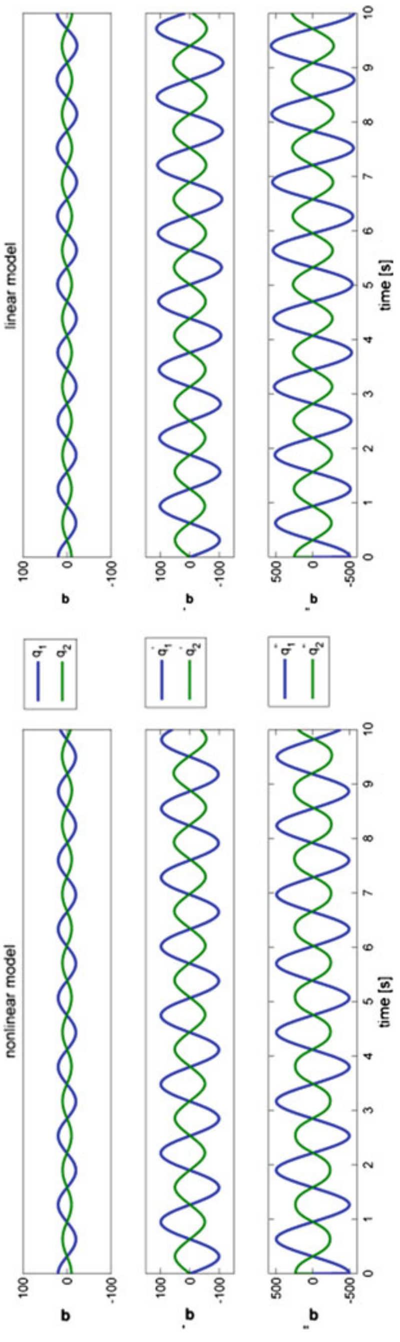


Fig. 4 Changes in the value of the internal coordinate of model 1link-2DoF



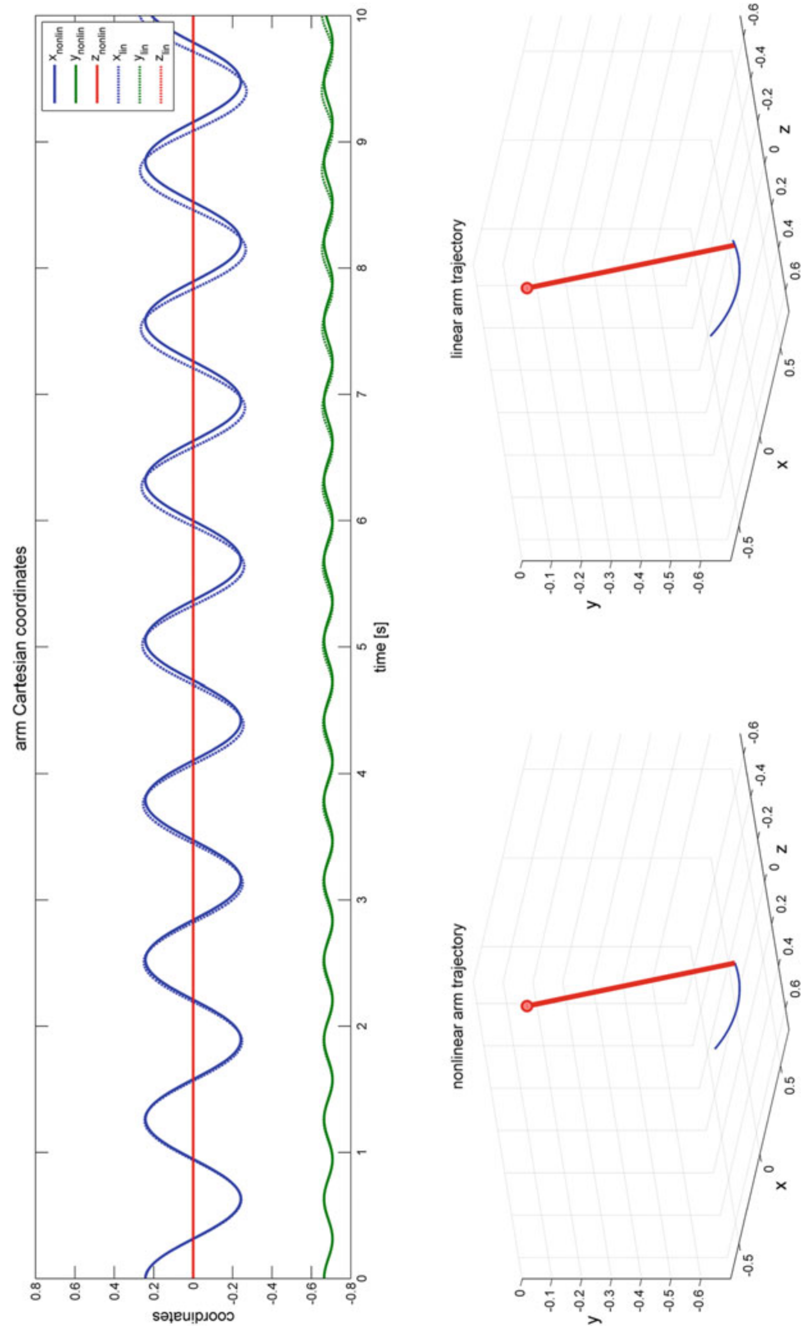


Fig. 5 The trajectory of the position of the 1link-1DoF hand model without torque

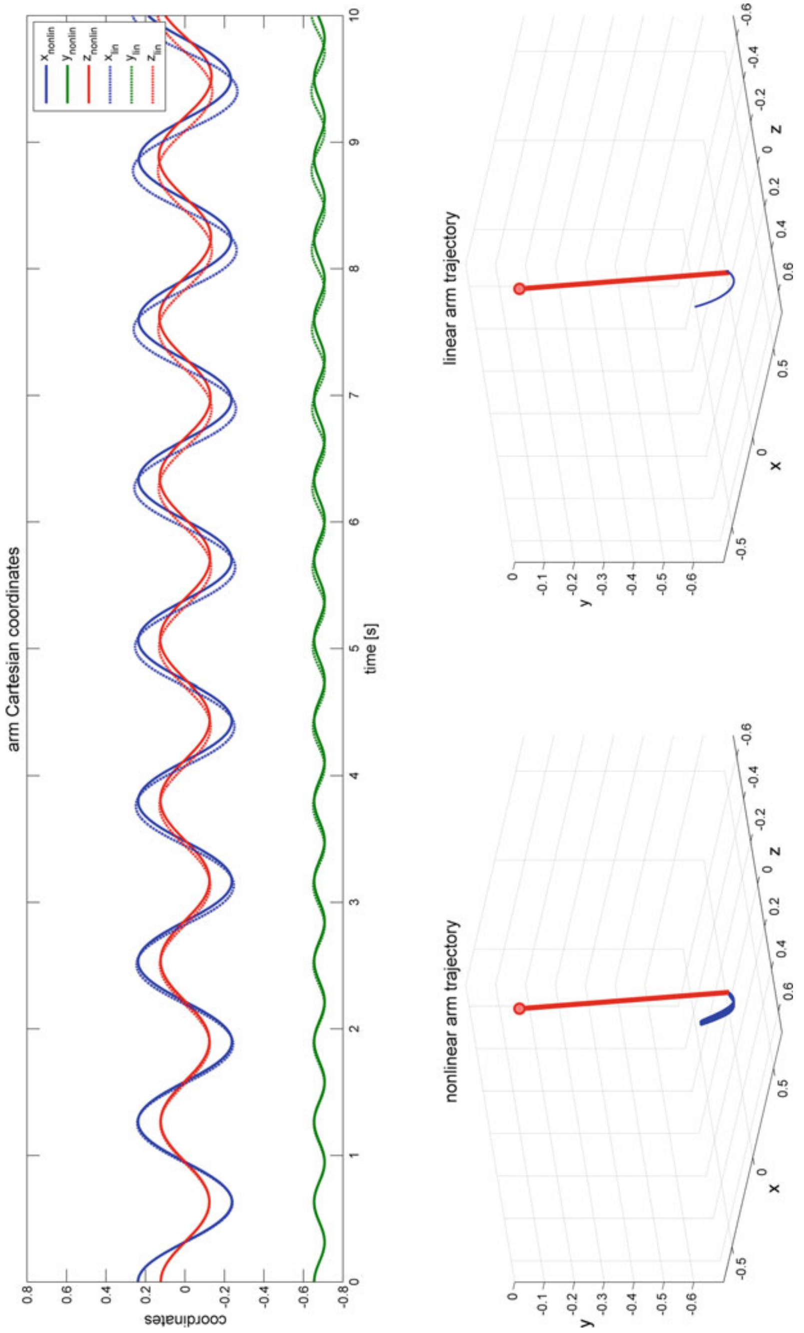


Fig. 6 The trajectory of the position of the 1link-2DoF hand model without torques

Fig. 7 The position coordinates of the 1link-1DoF model arm in the shoulder frame with the external torque

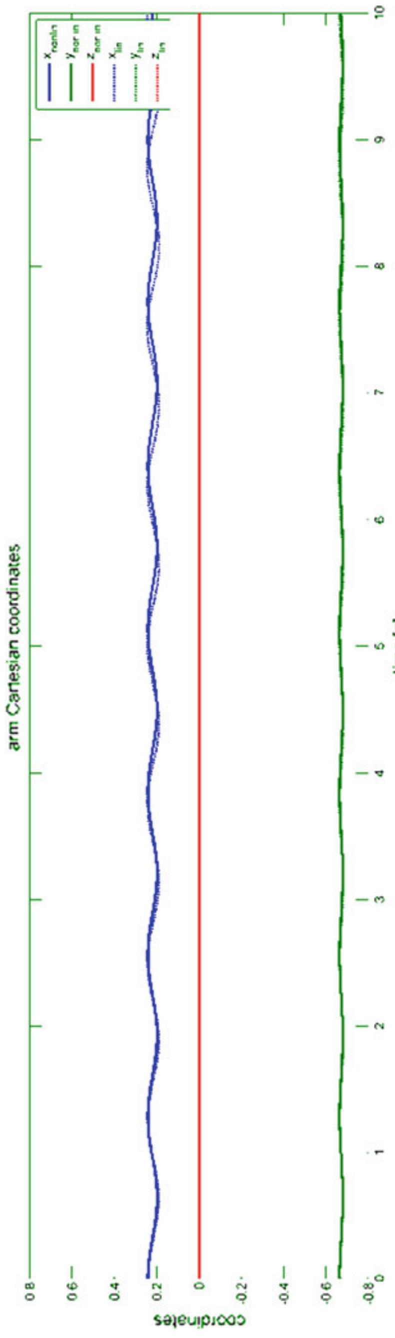
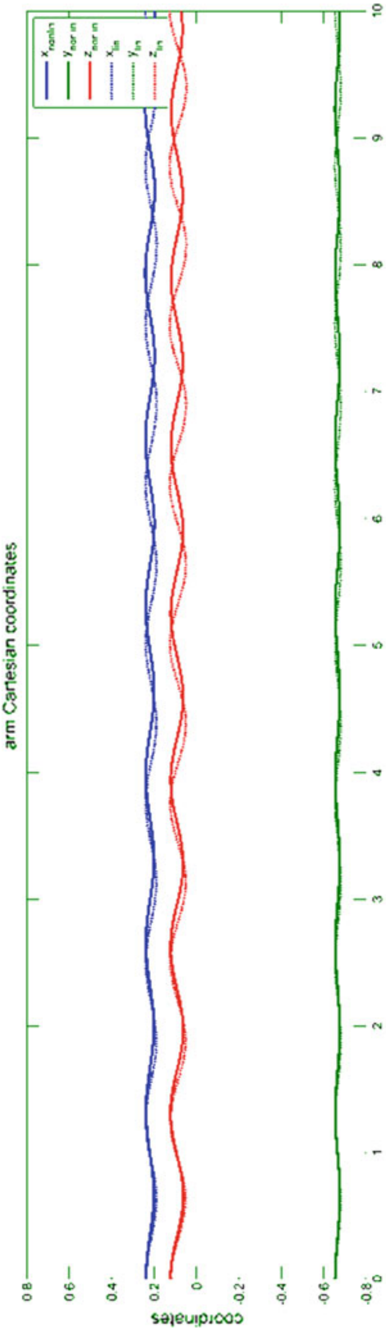


Fig. 8 The position coordinates of the 1link-2DoF model arm in the shoulder frame with the external torques



The simulations show the effectiveness of linear models of the dynamics of the human arm. Through the process of linearization synthesis of linear dynamic model is reduced significantly compare to the nonlinear models. At the same time their dynamic properties allow the analysis and synthesis of close-loop control systems [12–15].

References

1. Tee, K.P., Burdet, E., Chew, C.M., Milner, T.E.: A model of force and impedance in human arm movements, Biol. Cybern. **90**, 368–375 (2004)
2. Burdet, E., Tee, K.P., Mareels, I., Milner, T.E., Chew, C.M., Franklin, D.W., Osu, R., Kawato, M.: Stability and motor adaptation in human arm movements. Biol. Cybern. **94**, 20–32 (2006)
3. Grecu, V., Dumitru, N., Grecu, L.: Analysis of human arm joints and extension of the study to robot manipulator. In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, IMECS 2009, Hong Kong, vol. II (2009)
4. Li, W.: Optimal control for biological movement systems, PhD Thesis, University of California, San Diego (2006)
5. Driver, J., Thorpe, D.: Design, build and control of a single/double rotational inverted pendulum. The University of Adelaide, Australia (2004)
6. Wang, T.: Control force change due to adaptation of forward model in human motor control. MSc Thesis, The Johns Hopkins University Baltimore, Maryland (2000)
7. Jędrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 403–408 (19–21 June 2013) (ISBN: 978-1-4673-6320-4)
8. Omar, N.: Mathematical model of the arms during Kayaking using Gordon's method. In: International Conference Mathematical and Computational Biology, International Journal of Modern Physics: Conference Series, vol. 9, pp. 174–177 (2012)
9. Sanner, R.M., Kosha, M.: A mathematical model of the adaptive control of human arm motions. Biol. Cybern. **80**, 369–382 (1999)
10. Ulinowicz, M., Narkiewicz, J.: Identification of EMA dynamic model in mechatronics. In: Mechatrocnis, Recent Technological and Scientific Advances, pp. 375–383 (2012)
11. Williams, R.L. II.: Engineering Biomechanics of Human Motion, NotesBook Supplement for ME 4670/BME 5670 Engineering Biomechanics of Human Motion, Ohio University (2014)
12. Kuś, Z., Nawrat, A.: Object tracking for rapid camera movements in 3D space. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 57–76, (2013) (ISBN: 978-3-319-00368-9)
13. Jędrasiak, K., Nawrat, A., Wydmańska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)
14. Nawrat, A., Jędrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Shenyong Chen (ed.) Proceedings of WSEAS International Conference on Mathematics and Computers in Science and Engineering, no. 9. World Scientific and Engineering Academy and Society, Singapore (2009)
15. Grzejszczak, T., Mikulski, M., Szkodny, T., Jędrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 407–413 (2012)

Naive Kalman Filtering for 3D Object Orientation

Robert Bieda, Rafal Grygiel and Adam Galuszka

Abstract In the paper Naive Kalman filter is introduced and presented for estimating orientation in 3D space. Using the assumption of Bayesian classification systems, the angular velocity vector is treated as three separate events. Therefore, three independent Kalman filters are used to estimate Euler angles for each RPY coordinate system. Data fusion is presented for real IMU sensor which integrated data from triaxial gyroscope, accelerometer and magnetometer.

Keywords 3D orientation · IMU · Kalman filter · Innovative simulation systems

1 Introduction

Precise knowledge of a plant orientation in space is necessary information for correct implementation of various engineering applications. As examples one can mention control of plants that rotates, rating of human limbs movement, or visualization of position of human body parts in virtual reality. As an orientation we will understand Euler angles: R-roll, P-pitch and Y-yaw, around axis related to plant.

To estimate Euler angles the idea of sensor fusion is used, which is based on aggregation of measurement data coming from different sensors. The angles can be determined by gyroscope data by matrix integration of angular velocity [1, 2] or by accelerometers and magnetometers data, using algebraic dependencies [3–5].

R. Bieda (✉) · R. Grygiel · A. Galuszka
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: robert.bieda@polsl.pl

R. Grygiel
e-mail: rafal.grygiel@polsl.pl

A. Galuszka
e-mail: adam.galuszka@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_23

There are various methods and tools used to orientation estimation: quaternion calculus, complementary filters or Kalman's filter [6–11]. Kalman filtering methodology requires mathematical formulation of plant model, which in most cases is based on the nonlinear equations of rotational kinematics of rigid bodies. It follows that linearization operations are necessary. This approach is correct but in many cases leads to difficulties in computations and implementations.

In this work authors were searching for simplest algorithm for orientation estimation and it led to searching for the linear model of the problem. The main thesis of this work is that it is possible to determine orientation (Euler's angles) by independent processing of gyroscope indications in three independent axis of coordinate system connected with a plant. It leads to simplified innovative simulation model of Naïve Kalman Filter. Although the thesis seems to be wrong we present modelling methodology that confirm thesis correctness. It leads to conclusion the thesis should not be rejected.

In this work full description of sensor systems is presented. Basing on this a method of Euler's angles determination and two approaches of sensor fusion modelling are shown. Experiments were performed using data obtained from IMU sensor (Fig. 1) designed and produced by project team from Institute of Automatic Control of Silesian University of Technology [12].

2 Reference System

In problem of orientation estimation in 3D space various coordination systems are used. There are defined base and related to plant coordination systems [13–15]. The base system (reference system) is defined as a stationary system of navigation frame (called: n-frame or navigation-frame) [1, 2], e.g.: ECI, ECEF, LTP including: ENU and NED [1, 16]. Second group of systems are system related to the plant (called b-frame or body-frame) [2, 17] and system related to measurement unit placed on the plant (called s-frame or sensor-frame) [18]. In problem analyzed here the structure NWU (North-West-Up) as the reference system is proposed (Fig. 2).

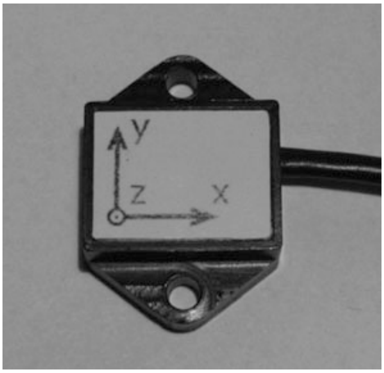


Fig. 1 IMU sensor

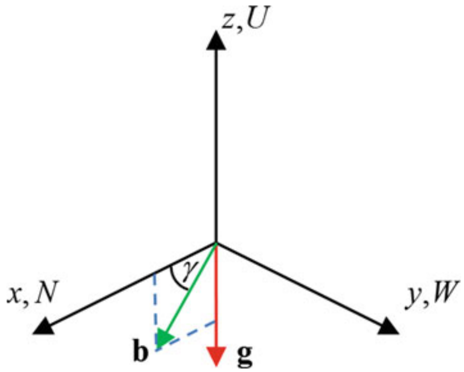


Fig. 2 Location of gravity acceleration vectors *g*, and earth's magnetic field *b* in reference system NWU

As a moving space, for which orientation with respect to the reference system is determined, form often used in navigation that allows to determine the angle of rotation around three axes called *Roll-Pitch-Yaw* was assumed. Additionally it is assumed that system associated with the object (called b-frame) is identical to the IMU sensors measuring system (called s-frame). In this study estimations of the roll angle, pitch angle and deviation angle (called yaw) to determine the orientation of the object are made. They are denoted by ϕ, θ, ψ, respectively.

3 Naive Kalman Filtering

Sensor fusion integrates sensory information into a coherent whole from various, generally physically separated and independent sources [19–22]. In case of orientation estimation the fusion concerns information from accelerometers, gyroscopes and magnetometers. In this work Kalman filtering which requires knowledge of a linear process model is used:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{u}_k + \boldsymbol{Q}\boldsymbol{w}_k \\ \boldsymbol{y}_k &= \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{r}_k \end{aligned} \tag{1}$$

It is assumed that sensor data (Fig. 1) are collected with the same period of time equal to Δt. It is therefore necessary to create a model of the process of determining the orientation in space of three independent channels. Similar models may be found in Ref. [e.g. 17], however, in these cases the description relates to rotation around one axis. Other approaches include nonlinear rotational kinematics of a rigid body that should be linearized. The measurement of the angular velocity ω_k^{G} of the gyroscope is subject to bias error and an additional stochastic disturbance modeled as a discrete white Gaussian noise with zero expected value and variance σ_ω^2:

$$\omega_k^{\mathrm{G}} = \tilde{\omega}_k + \beta_k + w_{\omega,k} \tag{2}$$

Zero error is one of the low-frequency noise and can be for-modeled in the form of implementation of the first order Markov process with random white noise with distribution $\mathrm{E}[\varepsilon_k] = 0; \mathrm{E}\left[\varepsilon_k^2\right] = \sigma_\varepsilon^2$:

$$\beta_{k+1} = \beta_k + \varepsilon_k \tag{3}$$

In order to determine the angle from the measured angular velocity, discrete integration is used according to the relation:

$$\omega(t) = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} \approx \frac{\alpha(t+\Delta t) - \alpha(t)}{\Delta t} = \frac{\Delta\alpha(t)}{\Delta t} \tag{4}$$

Then taking into account Eqs. (3) and (4) in Eq. (2) leads to the model:

$$\alpha_{k+1} = \alpha_k + \left(\omega_k^{\mathrm{G}} - \beta_k + w_{\omega,k}\right) \cdot \Delta t = \alpha_k + \omega_k^{\mathrm{G}} \cdot \Delta t - \beta_k \cdot \Delta t + \Delta t \cdot w_{\omega,k} \tag{5}$$

Aggregating description of the model (5) and Eq. (3) to the state equations (1) the following matrices and vectors are obtained:

$$\begin{gathered}\boldsymbol{F} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}; \boldsymbol{G} = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}; \boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}; \boldsymbol{Q} = \begin{bmatrix} \Delta t & 0 \\ 0 & 1 \end{bmatrix}; \\ \boldsymbol{x}_k = \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix}; u_k = \omega_k^G; y_k = \alpha_k^{\mathrm{AM}}; \boldsymbol{w}_k = \begin{bmatrix} w_{\omega,k} \\ \varepsilon_k \end{bmatrix}; r_k = r_k^{\mathrm{AM}}\end{gathered} \tag{6}$$

where r_k^{AM} is the noise of measurement $\left(\mathrm{E}\left[r_k^{\mathrm{AM}}\right] = 0; \mathrm{E}\left[\left(r_k^{\mathrm{AM}}\right)^2\right] = \sigma_\alpha^2\right)$ of angle determined by using the accelerometers and magnetometers. Model (1) defined by the elements (6) has a control input $u(t)$ as the angular velocity. This approach seems to be incorrect because the signal $u(t)$, according to Kalman filtering assumptions, should be a deterministic. However, from the standpoint of the modeling this approach is correct.

In order to get rid of the control input from the model determining the angle of the gyro, angular velocity has been included in the state vector. The process can now be described by the following matrices:

$$\begin{gathered}\boldsymbol{F} = \begin{bmatrix} 1 & \Delta t & -\Delta t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \boldsymbol{Q} = \begin{bmatrix} \Delta t & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \\ \boldsymbol{x}_k = \begin{bmatrix} \alpha_k \\ \omega_k \\ \beta_k \end{bmatrix}; \boldsymbol{y}_k = \begin{bmatrix} \alpha_k^{\mathrm{AM}} \\ \omega_k^{\mathrm{G}} \end{bmatrix}; \boldsymbol{w}_k = \begin{bmatrix} w_{\omega,k} \\ \varepsilon_k \end{bmatrix}; \boldsymbol{r}_k = \begin{bmatrix} r_k^{\mathrm{AM}} \\ r_k^{\mathrm{G}} \end{bmatrix}\end{gathered} \tag{7}$$

Model (1) as described by the matrices (7) does not include control signal but the output signal consists of two signals: the angle and angular velocity.

In the proposed solution model written for a single angle (axis) was applied to the other two angles. The extension of the proposed approach for all three angles of rotation was possible under the assumption of independence of the individual components of the angular velocity vector as in systems of Bayesian classification [23, 24]. In these systems so called naive classifier is introduced. It is based on the assumption that the probability of the event vector (multivariate random variable) in the analyzing process is equal to the product of the probabilities of elementary events. These consequences arises from the assumption of independence of elementary events that define a multi-dimensional random variable. So, for the analyzed case has been made the assumption that the angular velocity vector $\boldsymbol{\omega}^{\mathrm{G}}$ modeling a random event, can be analyzed as consisting of three independent events. Based on this assumption, the above formulas (6) and (7) can be extend to a three-dimensional case, thus leading to definition of the Naive Kalman filtering model.

Synthesizing a model for the orientation of the sensory fusion using Kalman filter two variants were considered. In the first approach, the model with the control input was applied ($\boldsymbol{G} \neq \boldsymbol{0}$) and it led to the following description:

$$\begin{aligned}
&\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{u}_k + \boldsymbol{Q}\boldsymbol{w}_k \\
&\begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\beta}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\Delta t \cdot \boldsymbol{I}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \end{bmatrix} + \begin{bmatrix} \Delta t \cdot \boldsymbol{I}_{3\times3} \\ \boldsymbol{0}_{3\times3} \end{bmatrix} \boldsymbol{\omega}_k^{\mathrm{G}} + \begin{bmatrix} \Delta t \cdot \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{\omega,k} \\ \boldsymbol{\varepsilon}_k \end{bmatrix} \\
&\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{r}_k \\
&\boldsymbol{\alpha}_k^{AM} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \end{bmatrix} + \boldsymbol{r}_k^{\mathrm{AM}}
\end{aligned} \tag{8}$$

where angles vectors, bias in the respective axes and angular velocity are defined as follows:

$$\begin{aligned}
\boldsymbol{\alpha}_k &= \begin{bmatrix} \phi_k & \theta_k & \psi_k \end{bmatrix}^T \\
\boldsymbol{\beta}_k &= \begin{bmatrix} \beta_{x,k} & \beta_{y,k} & \beta_{z,k} \end{bmatrix}^T \\
\boldsymbol{\omega}_k^G &= \begin{bmatrix} \omega_{x,k} & \omega_{y,k} & \omega_{z,k} \end{bmatrix}^T
\end{aligned} \tag{9}$$

Matrices $\boldsymbol{I}_{n\times n}, \boldsymbol{0}_{n\times n}$ are the identity matrix and matrix of zeros of size $n \times n$, respectively. Model outputs in a correction stage are angles determined from the accelerometers and magnetometers:

$$\boldsymbol{\alpha}_k^{\mathrm{AM}} = \begin{bmatrix} \phi_k^{\mathrm{AM}} & \theta_k^{\mathrm{AM}} & \psi_k^{\mathrm{AM}} \end{bmatrix}^T \tag{10}$$

The second variant of the model, including the angular velocity of the state vector results in the description:

$$\begin{aligned}
&\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{Q}\boldsymbol{w}_k \\
&\begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\omega}_{k+1} \\ \boldsymbol{\beta}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \Delta t \cdot \boldsymbol{I}_{3\times3} & -\Delta t \cdot \boldsymbol{I}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{\beta}_k \end{bmatrix} + \begin{bmatrix} \Delta t \cdot \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{\omega,k} \\ \boldsymbol{\varepsilon}_k \end{bmatrix} \\
&\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{r}_k \\
&\begin{bmatrix} \boldsymbol{\alpha}_k^{\mathrm{AM}} \\ \boldsymbol{\omega}_k^{\mathrm{G}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{\beta}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{r}_k^{\mathrm{AM}} \\ \boldsymbol{r}_k^{\mathrm{G}} \end{bmatrix}
\end{aligned} \tag{11}$$

In this model, the output vector extends of the measurement from the gyroscope $\boldsymbol{\omega}_k^{\mathrm{G}}$.

For the model described by the system of equation (1) Kalman filter [17] is defined by a prediction step:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}\boldsymbol{u}_k \tag{12}$$

with

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^T + \boldsymbol{Q}\boldsymbol{W}\boldsymbol{Q}^T \tag{13}$$

and filtration step:

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}\left[\boldsymbol{y}_{k+1} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k+1|k}\right] \tag{14}$$

with

$$\begin{aligned}
\boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1|k}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^T + \boldsymbol{R}\right)^{-1} \\
\boldsymbol{P}_{k+1|k+1} &= (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H})\boldsymbol{P}_{k+1|k}
\end{aligned} \tag{15}$$

with the following distributions of stochastic disturbances and measurement noise in the model:

$$\mathrm{E}\left[\hat{\boldsymbol{x}}_{0|-1}\hat{\boldsymbol{x}}_{0|-1}^T\right] = \boldsymbol{X}_0,\ \mathrm{E}\left[\boldsymbol{w}_k\boldsymbol{w}_k^T\right] = \boldsymbol{W},\ \mathrm{E}\left[\boldsymbol{r}_k\boldsymbol{r}_k^T\right] = \boldsymbol{R} \tag{16}$$

Presented description of the process model in two variants (8) and (11) can be decomposed into three independent subproblems, for each axis (angle) separately. The idea of using a Naive Kalman filter, in three independent channels is shown in Fig. 3. This decomposition simplifies the implementation of the Kalman filter on the target device [12, 25–30].

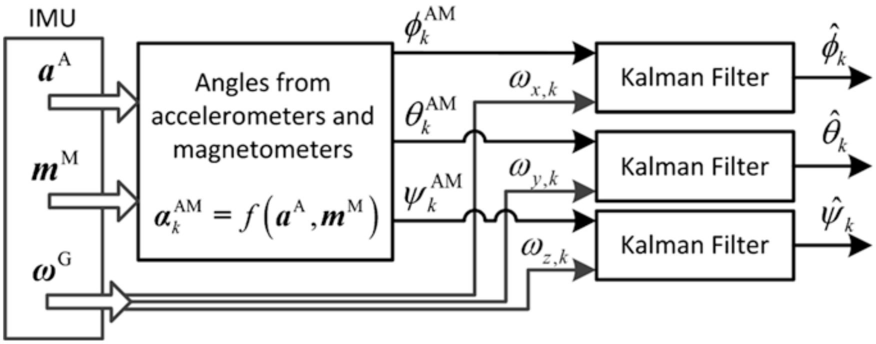


Fig. 3 The idea of Naive Kalman filter implementation

4 Exemplary Experiments

For the purpose of testing IMU sensor simulator was designed, which for defined referenced evolutions of rotation angles in time determines time responses returned by the sensor systems of linear acceleration $\left(\boldsymbol{a}_k^{\mathrm{A}}\right)$, angular velocity $\left(\boldsymbol{\omega}_k^{\mathrm{G}}\right)$ and the vector of magnetic field $\left(\boldsymbol{m}_k^{\mathrm{M}}\right)$:

$$\begin{aligned} \boldsymbol{a}_k^{\mathrm{A}} &= \tilde{\boldsymbol{a}}_k^{\mathrm{A}} - \boldsymbol{g}_k^{\mathrm{A}} + \boldsymbol{w}_{a,k}^{\mathrm{A}} \\ \boldsymbol{\omega}_k^{\mathrm{G}} &= \tilde{\boldsymbol{\omega}}_k + \boldsymbol{\beta}_k + \boldsymbol{w}_{\omega,k}^{\mathrm{G}} \\ \boldsymbol{m}_k^{\mathrm{M}} &= \tilde{\boldsymbol{b}}_k^{\mathrm{M}} + \boldsymbol{d}_k + \boldsymbol{w}_{m,k}^{\mathrm{M}} \end{aligned} \tag{17}$$

Generally, the IMU simulator performs the task described by the formula:

$$\left(\boldsymbol{a}_k^{\mathrm{A}}, \boldsymbol{\omega}_k^{\mathrm{G}}, \boldsymbol{m}_k^{\mathrm{M}}\right) = f_{\mathrm{IMU}}\left(\boldsymbol{\alpha}_k, \tilde{\boldsymbol{a}}_k, \boldsymbol{\beta}_k, \boldsymbol{d}_k, \boldsymbol{\sigma}_{\mathrm{sensor}}\right) \tag{18}$$

where:

$\boldsymbol{\alpha}_k$	referenced values of rotation angles,
$\tilde{\boldsymbol{a}}_k$	linear acceleration in navigation system,
$\boldsymbol{\sigma}_{\mathrm{sensor}} = \left\{\sigma^{\mathrm{A}}, \sigma^{\mathrm{G}}, \sigma^{\mathrm{M}}\right\}$	vector of sensor variances.

Plots were generated from the IMU sensor simulator for the three series under the assumption of no disturbances of the magnetic field $(\boldsymbol{d}_k = \boldsymbol{0})$ and linear acceleration of a plant $(\tilde{\boldsymbol{a}}_k = \boldsymbol{0})$ in the presence of gyroscopes bias:

$$\boldsymbol{\beta}_k = \begin{bmatrix} 0.01 & 0.01 & 0.01 \end{bmatrix}^T [\mathrm{rad/s}] \tag{19}$$

Figure 4 presents reference angles (“ref”) and angles determined using only accelerometers (AM) and only results of matrix integration of gyroscopes indications (G).

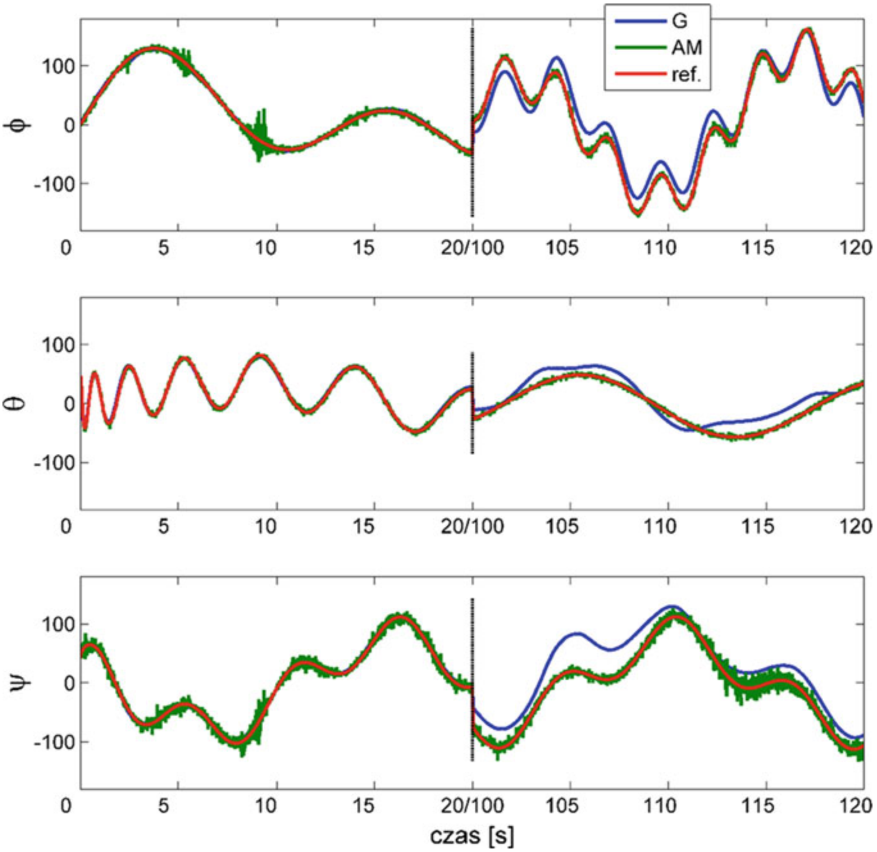


Fig. 4 Plots generated using reference angles ("ref") and angles determined from the accelerometers–magnetometers system ("AM") and gyroscopes ("G")

Figure 5 shows angle estimates using the Kalman filter model to a control input (8) for variable values of the variance σ^2 (matrix $\boldsymbol{W} = \boldsymbol{I} \cdot \sigma^2$). Figure 6 shows a comparison of the quality of operation of filtering Klaman model with control input (8) and without control input (11) against referenced angles ("ref").

Plots analysis confirm that solution using naive Kalman filter is correct and introduced alternative solutions with model (8) and (11) lead to practically identical results.

For quantitative analysis of the accuracy of estimation MAE (mean absolute error) was used:

$$\mathrm{MAE} = \frac{1}{n}\sum_{k=1}^{n} |\hat{\alpha}_k - \alpha_k| \tag{20}$$

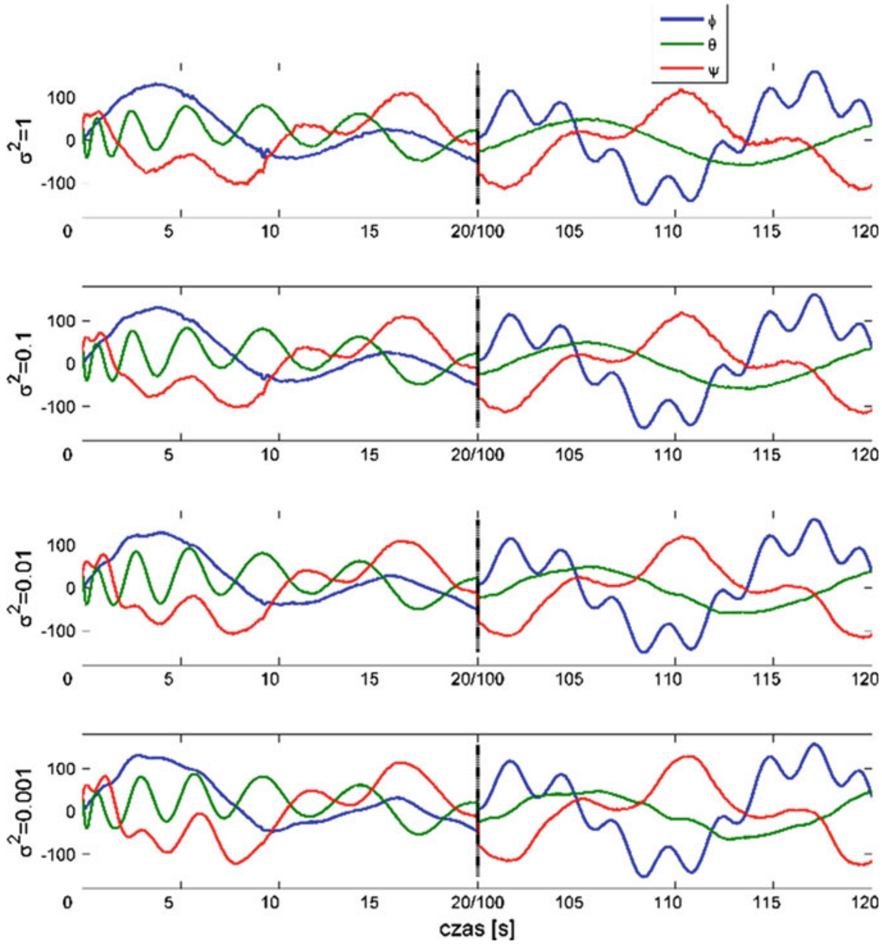


Fig. 5 Time plots for model with control input (F.K. 1) (8) for different values σ^2 = {1, 0.1, 0.01, 0.001}, where $\boldsymbol{W} = \boldsymbol{I} \cdot \sigma^2$

where

$\hat{\alpha}_k$ the estimated value of the angle,

α_k correct value of the angle.

Analyses of results presented in Table 1 show that the best results in terms of the smallest MAE rate for the entire 3D orientation was obtained for $\sigma^2 = 0.1$. Therefore, this value was chosen for the experiments presented in Fig. 6.

Table 1 The mean absolute error (MEA) in degrees for the estimation of the Euler angles using different integration methods and variable angular velocity of the variance

Method		Roll − MAE (deg)	Pitch − MAE (deg)	Yaw − MAE (deg)
AM		*31,781*	*23,067*	*57,622*
G		*128,030*	*88,333*	*181,268*
Naive KF with model in (8)	σ_{ω}^{2} = 0.001 s	4.2272	6.1709	7.2023
	σ_{ω}^{2} = 0.01 s	1.7079	2.7200	3.1978
	σ_{ω}^{2} = 0.1 s	**0.9293**	1.1316	**1.8757**
	σ_{ω}^{2} = 1 s	1.0287	**0.8560**	1.9553
Naive KF with model in (11)	σ_{ω}^{2} = 0.001 s	5.0257	5.5097	6.5706
	σ_{ω}^{2} = 0.01 s	1.7244	2.6046	3.0352
	σ_{ω}^{2} = 0.1 s	**0.9246**	1.1169	**1.8603**
	σ_{ω}^{2} = 1 s	1.0285	**0.8546**	1.9545

Bold Minimal error values

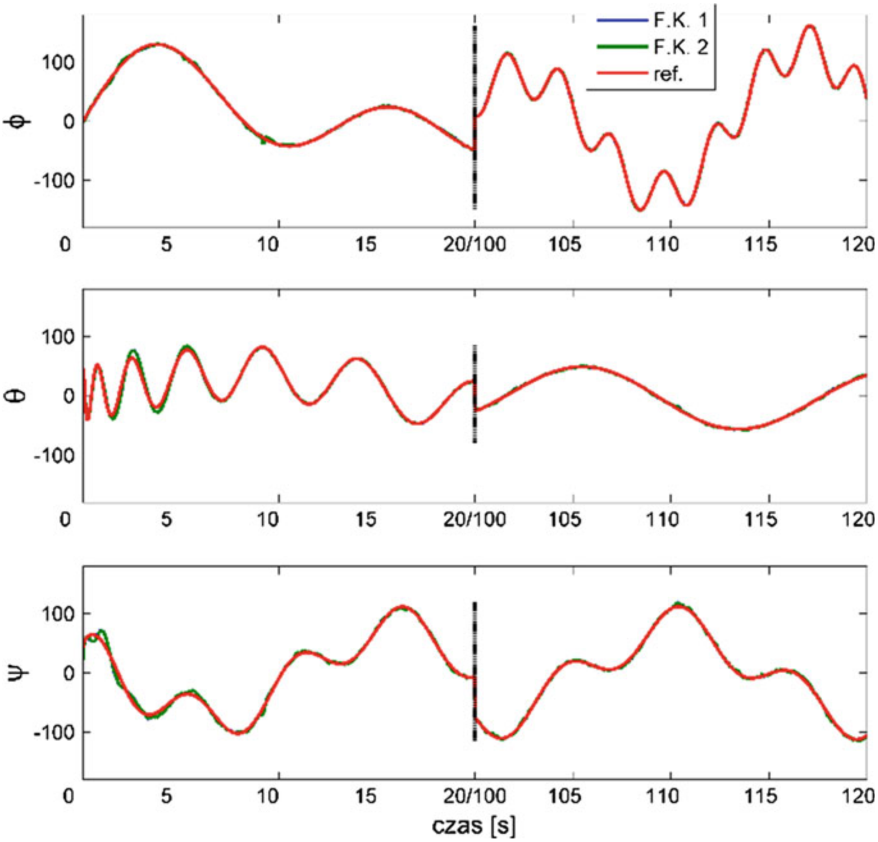


Fig. 6 Time plots for simulated data when $\sigma^2 = 0.1$ for model (8) (F.K. 1) and (11) (F.K. 2) against referenced plots (“ref”)

5 Conclusion and Future Work

In this paper, using the idea of construction of the so-called naive classifier orientation estimation algorithm in the 3D space, using three independent Kalman filters is proposed. It was assumed that the angular velocity $\boldsymbol{\omega}^{\mathrm{G}}$ determines a vector a consisting of three independent events. This incorrect assumption was made intentionally and it made possible to build independent Kalman filter for each axis RPY. It enables simplifications in implementation on the target device comparing to classical approaches. This approach we called Naive Kalman filter. Then using measurements of the gyroscope, accelerometers and magnetometers, appropriate Euler angles was determined.

The modeling process in the form of discrete integration, may be performed using a model with control input or without control input [31]. The proposed Kalman filters determines also the gyro zero errors. Appropriate parameterization of the model allows to obtain time plots of estimated angles with the desired variance, accuracy and regulation time.

Acknowledgements This work has been supported by NCBIR grant No. for the first and second author and by Institute of Automatic Control BK grant No. 214/RAu1/2013 for the third author in the year 2014.

References

1. Titterton, D.H., Weston, J.L.: Strapdown Inertial Navigation Technology, 2nd edn. The Institution of Electrical Engineers, Stevenage (2004)
2. Pusa, J.: Strapdown inertial navigation system aiding with nonholonomic constraints using indirect Kalman filtering, MSc Thesis, Tampere University of Technology (2009)
3. Pedley, M.: Tilt sensing using a three-axis accelerometer. Freescale Semicond. Appl. Note, n. AN3461 (2013)
4. Songlai, H., Wang, J.: A novel method to integrate IMU and magnetometers in attitude and heading reference systems. J. Navig. **64**, 727–738 (2011)
5. Caruso, M.J.: Applications of Magnetoresistive Sensors in Navigation Systems. Honeywell Inc, New Jersey (1998)
6. Fux, S.: Development of a planar low cost inertial measurement unit for UAVs and MAVs, MSc Thesis, Eidgenössische Technische Hochschule Zürich (2008)
7. Peter Haywood King: Low Cost Localization Solution Using a Kalman Filter for Data Fusion, Praca magisterska. Virginia Tech, Blacksburg (2008)
8. Andrzejczak, M., Ulinowicz, M.: Filtration and integration system (FIS) for navigation data processing based on Kalman Filter. In: Intelligent Systems' 2014, pp. 203–2010 (2015)
9. Ulinowicz, M., Narkiewicz, J.: Identification of EMA Dynamic Model in Mechatronics, Mechatrocnis, Recent Technological and Scientific Advances, pp. 375–383 (2012)
10. Kuś, Z., Nawrat, A.: Object tracking for rapid camera movements in 3D space. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 57–76 (2013) (ISBN: 978-3-319-00368-9)
11. Barnat, W., Panowicz, R., Niezgoda, T., Dybcio, P.: Numerical Analysis of IED detonation effect on steel plate. Acta Mech. et Autom. **6**, 10–12 (2012)

12. Jedrasiak, K., Daniec, K., Nawrat, A.: The low cost micro inertial measurement unit. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 403–408, 19–21 June 2013 (ISBN: 978-1-4673-6320-4)
13. Haid, M., Breitenbach, J.: Low cost inertial orientation tracking with Kalman filter. Appl. Math. Comput. **153**, 567–575 (2004)
14. Gucma, M., Montewka, J.: Podstawy morskiej nawigacji inercyjnej. Akademia Morska w Szczecinie, Szczecin (2006)
15. Woodman, O.J.: An introduction to inertial navigation, Technical Report 696, University of Cambridge (2007)
16. Grewal, M.S., Weill, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, and Integration. Wiley, New York (2001)
17. Medith, J.S.: Estymacja i sterowanie statystycznie optymalne w układach liniowych. WNT, Warszawa (1975)
18. Roetenberg, D.: Inertial and Magnetic Sensing of Human Motion, PhD Thesis, University of Twente (2006)
19. Kim, K., Park, C.G.: A new initial alignment algorithm for strapdown inertial navigation system using sensor output. In: Proceedings of the 17th World Congress The International Federation of Automatic Control (IFAC), pp. 13034–13039 (2008)
20. Roetenberg, D., Henk, J., Luinge, H.J., Chris, T.M., Baten, C.T.M, Veltink P.H.: Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. IEEE Trans. Neural Syst. Rehabil. Eng. **13**(3), 305–405 (2005)
21. Talat Ozyagcilar, T.: Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. Freescale Semicond. Appl. Note, n. AN4248 (2012)
22. Sadłowski, P.: Parametryzacje rotacji i algorytmy rozwiązywania równań dynamiki z rotacyjnymi stopniami swobody, Praca doktorska, *Polska Akademia Nauk* (2007)
23. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Academic Press, Elsevier (2006)
24. Koronacki, J., Ćwik, J.: Statystyczne systemy uczące się, wydanie drugie. Akademicka Oficyna Wydawnicza EXIT, Warszawa (2008)
25. Jędrasiak, K., Nawrat, A., Wydmańska, K.: SETh-link the distributed management system for unmanned mobile vehicles. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 247–256 (2013)
26. Nawrat, A., Jędrasiak, K.: SETh system spatio-temporal object tracking using combined color and motion features. In: Chen, S. (ed.) Proceedings of WSEAS International Conference on Mathematics and Computers in Science and Engineering, no. 9 World Scientific and Engineering Academy and Society (2009)
27. Iwaneczko, P., Jędrasiak, K., Daniec, K., Nawrat, A.: A prototype of unmanned aerial vehicle for image acquisition. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 87–94 (2012)
28. Jędrasiak, K., Nawrat, A.: Image recognition technique for unmanned aerial vehicles. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 5337, pp. 391–399 (2009)
29. Grzejszczak, T., Mikulski, M., Szkodny, T., Jędrasiak, K.: Gesture based robot control. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 407–413 (2012)
30. Daniec, K., Iwaneczko, P., Jędrasiak, K., Nawrat, A.: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 219–232 (2013) (ISBN: 978-3-319-00368-9)
31. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)

Non-GPS Navigation System for Criminalistic Investigation on Mobile Robot

Robert Głębocki, Paweł Kicman and Janusz Narkiewicz

Abstract In the paper we describe visual odometry algorithm for monocular camera. It is developed for use on a mobile robot involved in criminal investigation conducted by Polish police forces. FAST keypoints with ORB descriptors are used as input to the motion estimation. Concept of pose graph is implemented in order to improve accuracy of the results obtained. The algorithm was evaluated on a 400 m long outdoor dataset included in KITTI Benchmark (18.41 % position error) as well as on a short dataset collected indoors in one of the offices at our University (5.34 % position error). Detailed discussion of the results is presented together with the guidance for further development of the method.

Keywords Navigation · Visual odometry · Pose graph

1 Introduction

Nowadays the Global Positioning System (GPS) has proved a boon for robots and vehicles control and navigation system. GPS navigation is most popular one in many applications on the ground, on the sea and in the air. But the satellite-based system is not without its shortcomings. Something as simple as going indoors or entering a tunnel can render the system useless. In presented papers authors described the mobile robot system involved in criminal investigation conducted by Polish police forces. The robot should work in the places where CBRN hazards are possible and where area is too dangerous for policeman to work. The robot is

R. Głębocki (✉) · P. Kicman · J. Narkiewicz
Department of Automation and Aeronautical Systems, Warsaw University of Technology,
Warsaw, Poland
e-mail: rglebocki@meil.pw.edu.pl

P. Kicman
e-mail: pkicman@meil.pw.edu.pl

J. Narkiewicz
e-mail: jnark@meil.pw.edu.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_24

remotely controlled during the work by radio line. The problem is that this kind of police operation takes place inside buildings, very often in cellars and hard to reach places. Robot must penetrate a stairs, move doors etc. Police has a problem when robot can lose connection with operator. To find it inside building is dangerous by the reason that robot is used to penetrate CBRN hazard area. Because of that robot has to have a possibility to autonomously return without GPS. Navigation system has inertial navigation but it is insufficiently precise. This way we need to use additional non GPS navigation system. There are some approaches to determine position estimation without GPS. Typically, these systems require external references, such as preinstalled active beacons, receivers, or optical retroreflectors. Problem is that navigation system devices must be installed in the work space at precisely surveyed locations before the system can be used. This installation is in our case completely impossible. Police penetrate unknown places of crime without any possibility to prepare navigation devices. Robot must return autonomously based on equipment which we can mount on its board. Another way of implementing absolute position estimation is computer vision system. This method is developed in our experiments.

2 Visual Odometry

Video camera is a very good navigation sensor as the images contain large amounts of information about surrounding environment. Camera can be a good complement for traditional navigation systems as it provides positioning information both indoors and outdoors which is not the case for GNSS systems. Those opportunities are well recognized in the research community and therefore use of visual systems in navigation is increasing. So far there have been many successful applications of visual systems for navigation. However, there are still challenges in creating reliable and robust algorithms that will provide valid navigation information in various scenarios and in different environments. Another important factor that speaks in favor of visual systems is the cost. Price of a video camera and computer processing the images is significantly lower, compared to high quality navigation systems. Moreover, cameras are being installed nowadays on default in many cars and robots. Thanks to that fact vision based navigation can become a very widespread method of motion estimation in the near future. So it is important to develop algorithms that are safe and robust.

In case of our scenario visual navigation, where small robot is exploring unknown buildings, the visual system integrated with additional sensors (IMU and magnetic compass) can provide reliable position information. In such configuration the system is independent and does not rely on any external source of information. This is especially important as the robot is meant to operate without access to satellite navigation signals.

For estimation of motion from camera images visual odometry algorithm was selected. Next to visual-SLAM it is one of the methods that enables calculation of movement of the vehicle from video stream. It uses images captured by a camera mounted rigidly to the body of the car or robot. Based on analysis of those images the algorithm estimates movement of the object. The algorithm is based on dead reckoning principle which means that transitions between consecutive positions are convoluted to provide current position.

2.1 Related Work

The most commonly used class of visual odometry algorithms is based on features detected in images. The features are characteristic points that can be reliably recognized and matched between consecutive image frames. In ideal situation they can be identified from different observation positions, different distances and under changing light conditions. There are several algorithms used for detection of features such as SURF, SIFT, FAST, ORB and many others. For each pixel selected in the image a descriptor vector is calculated. It contains number of unique properties of that point that are used in the matching process. Pairs of points matched between two images are used to calculate transition of the camera between places where images were taken. Features can be matched directly, by comparison of their descriptor vectors, or can be tracked between images, by using for example optical flow.

Different approach to estimation of motion is represented by appearance-based methods. In these class of algorithms parts of images are compared between consecutive images. This method can give good results even when little texture in image is observed. However, it is less often used because the algorithms are more susceptible to occlusions and in general less reliable.

Large amount of algorithms in visual odometry are developed for stereo-vision cameras. One of the recent developments in the visual odometry algorithm show that the position accuracy achieved using only visual information can be as low as 1.3 % of distance travelled when stereo sequences are used [1]. With additional use of IMU and optimization of results it is possible to reduce positioning error to values below 0.5 % [2]. But different camera types have also been successfully used for estimation of motion, i.e. monocular camera [3], as well as omnidirectional system [4].

The most significant challenge in visual odometry algorithms is the growth of position error with time. The method is based on a dead-reckoning principle which means that errors from all time steps accumulate over time. One of the possible solutions to this issue is a fusion of visual information with data coming from other sensors, like for example GPS or IMU [2]. Another solution is to use additional constraints between more than just two recent frames in order to optimize and reduce the overall error. Two methods that have been demonstrated to work in literature are Windowed Bundle Adjustment (WBA or SBA from Sparse Bundle Adjustment) and Pose Graph Optimization (PGO). WBA is a modification of a bundle adjustment algorithm proposed in [2]. It adapts the traditional original bundle adjustment algorithm so that it optimizes path over several recent frames only. Thanks to that change the algorithm is capable of working in the real time.

The Pose Graph Optimization was demonstrated in [5]. In this method a graph of poses (position and orientation in a given time) is created based on visual information. Edges of the graph are transitions between the poses. The algorithm optimizes transitions between several recent nodes of the graph in order to minimize mean squared error of robot movement. This method uses only information about poses and transitions between them, while WBA estimates 3D positions of features as well as poses of a robot. Therefore WBA gives more accurate results, but requires higher computational resources compared with PGO.

For further in-depth analysis of methods and current developments in visual odometry we recommend a recent survey provided by Scaramuzza et al. in his two-part tutorial [6, 7].

2.2 Algorithm Outline

Flowchart of the algorithm that was implemented at WUT is provided in the Fig. 1. Visual odometry algorithm that uses feature detection and matching on images recorded by a monocular camera was used. At first the image is rectified (removal

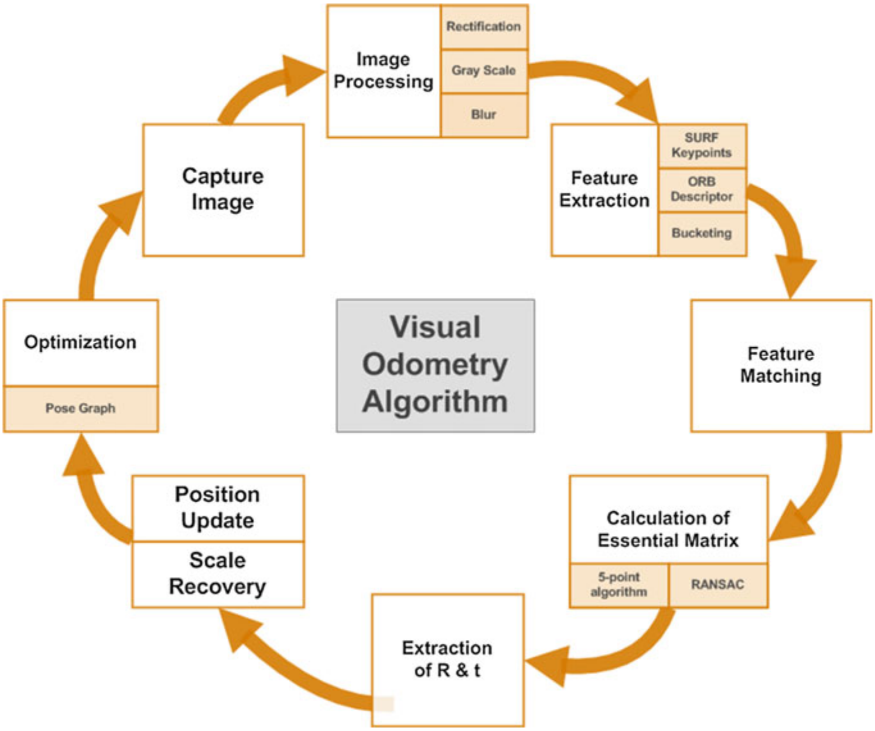


Fig. 1 Visual odometry process flow

of distortions) and preprocessed (conversion to grayscale and Gaussian blur). The motion is calculated based on the analysis of point features translations. Several different algorithms were tested and the best results were achieved with FAST keypoints algorithm [8]. Results of this experiment are not included in this paper. For each extracted point a descriptor vector containing set of characteristics values is calculated. In WUT developed algorithm the ORB feature descriptor was used [9]. Based on the descriptor vector the features are matched using a brute force matching algorithm which works acceptably fast when executed on a GPU.

The camera motion is estimated using algorithm described by Stewenius in [10]. It uses five pairs of matched points to calculate essential matrix that binds two images. To ensure the reliable calculation the RANSAC framework is used to reject any outliers (especially false matches). Therefore up to 1000 randomly selected 5-point subsets of matched points are tested and evaluated. Based on the Sampson Distance error measure the features are regarded as inliers or outliers. After the evaluation the selected subset of inliers is used to recalculate final essential matrix representing motion between the two frames. Following this step rotation matrix and translation vector are extracted from the essential matrix using algorithm given by Horn [11].

When using monocular camera the essential matrix estimated in the process does not preserve scale of the movement. Therefore it is not possible to directly calculate the real distance traveled by the robot. To surpass this issue reference to an external source for reliable estimation of the travelled distance is done. For the purpose of the tests the data comes either from IMU data or measurements performed by hand. The new pose is used to update a pose graph (more details in the next chapter).

2.3 Pose Graph

Poses calculated by the visual odometry algorithm are stored in a structure called pose graph (see In the future implementation of the Pose Graph Optimization algorithm that will use information encoded in the recent frames to minimize both position and orientation error is planned Fig. 2).

Nodes of the graph represent poses at given time step. Edges of the graph represent the transitions between the poses. They are calculated based on the features matched between those frames. It is not necessary to perform matching procedure between every image in the graph. Direct matching is performed only on consecutive frames and matches with older frames are found by backtracking of the old matches. Depth of the matching history is set by a sliding window size which in the presented sample case in. In the future implementation of the Pose Graph Optimization algorithm that will use information encoded in the recent frames to minimize both position and orientation error is planned.

Figure 2 is set to 4.

In our implementation the information contained in the pose graph is used to evaluate validity of the motion. Edges coming to the node are checked for

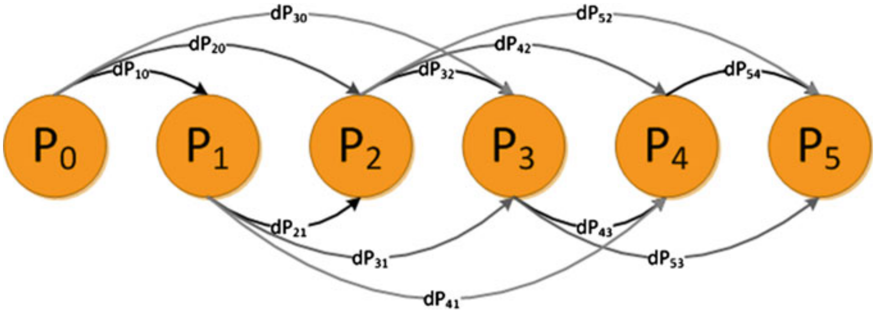


Fig. 2 Pose graph created from consecutive poses. Window size is set to 4

consistency (by voting). The inconsistent edges are rejected as outliers. This is an efficient and simple method that enables observation of the failures of the motion estimation algorithm.

In the future implementation of the Pose Graph Optimization algorithm that will use information encoded in the recent frames to minimize both position and orientation error is planned.

3 Experimental Evaluation

3.1 KITTI Benchmark

The algorithm has been evaluated at the data provided by KITTI Vision Benchmark Suite which is a joint project of Karlsruhe University of Technology and Toyota Technological Institute. The dataset is meant to challenge research community and provide a universal measure for development of various tasks in computer vision, such as visual odometry, optical flow or object recognition. The data is available online. It has been used by multiple research institutions and results of their tests can be found on a website. Therefore it is possible to compare developed algorithms with the state-of-the-art solutions.

Detailed description of the dataset is provided by authors in [12]. Data has been collected using a standard car equipped with multiple sensors. The sensors include two stereo-cameras (black-white and color), laser range scanner and accurate GPS/INS unit with RTK corrections that is providing ground-truth information. The data was collected on various types of roads in the city of Karlsruhe including rural areas and highways. Sample images from the dataset are visible in Fig. 3.

Our tests were performed offline, on previously acquired data. Performing experiment on the KITTI dataset enabled comparison of performance on well established test dataset. After calculation of the path travelled the results were compared with the available ground truth data and evaluated. Test results for the

Fig. 3 Sample images from KITTI dataset

dataset *2011_09_26_drive_0014_sync* are presented in Fig. 4. One can see that the results provided by our visual odometry algorithm are slowly diverging from the recorded ground truth. Position error for this test expressed as a function of distance travelled is given in Fig. 5. It can be seen that the final error after traveling over 400 m equals 74.06 m, that is 18.41 % of the overall distance travelled. The error starts to grow quickly after the car completes initial turn, which would indicate that errors induced by inaccurately estimated rotation are contributing significantly to the entire error. This is consistent with observations presented in [4], where authors

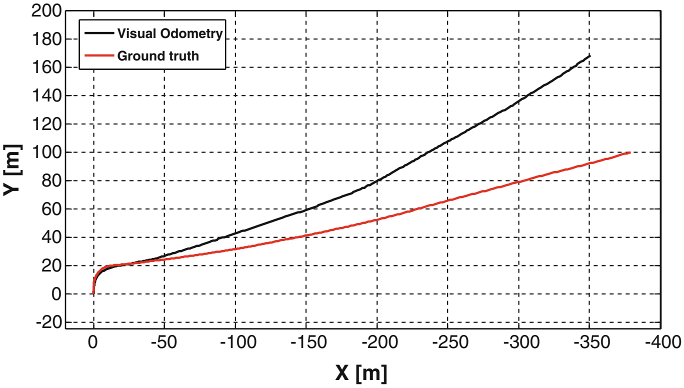


Fig. 4 Visual odometry results for one of the KITTI test sets

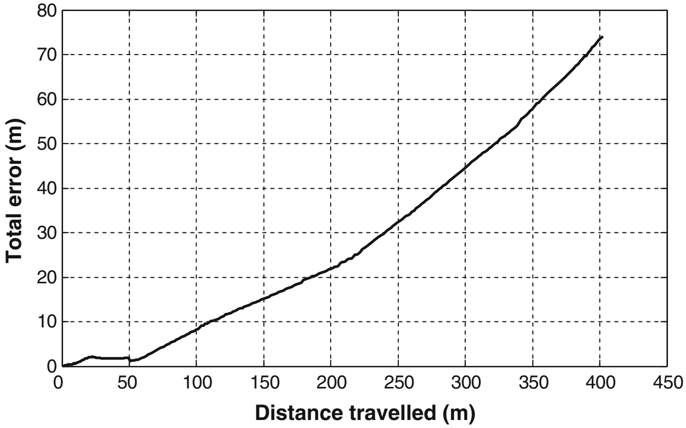


Fig. 5 Position error as a function of distance travelled for KITTI dataset

point out that accuracy of orientation estimation based on a basic visual odometry approach is low. Therefore, in our future works it is planned to explore possible modifications that could address this issue.

3.2 WUT Indoor Test

The second test was performed on a photos collected in one of the offices at Warsaw University of Technology. The camera was placed around 60 cm above the ground on a tripod. Such conditions are more similar to the operation scenario of the platform for which the algorithm is developed. Therefore, they provide more reliable prediction of the algorithm's performance. Positions at which the photos were collected were measured using measuring tape. To simplify evaluation orientation of the camera was kept constant during the entire experiment. Sample photographs that come from a dataset are shown in Fig. 6. They present typical

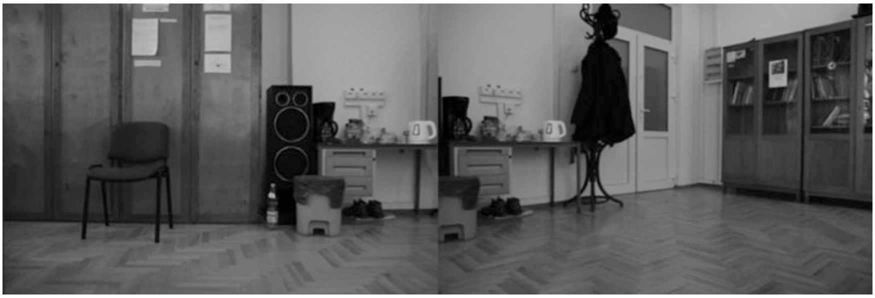


Fig. 6 Sample images taken in one of University offices

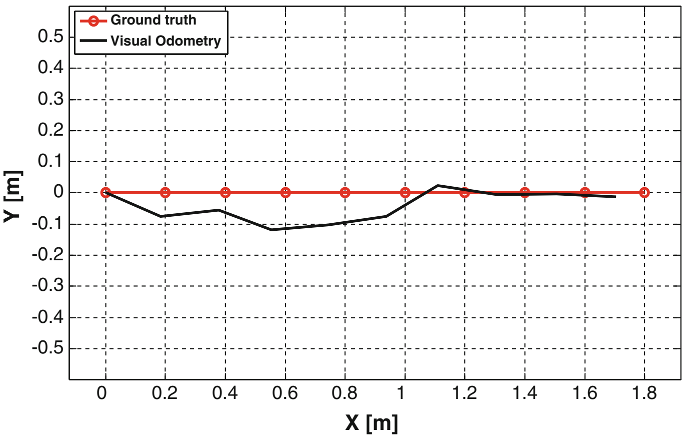


Fig. 7 Path in office test

office environment. It is noticeable that large areas of images contain flat surfaces with little or low texture. Such conditions are very challenging for reliable extraction and matching of features.

The results obtained in this test are presented in Fig. 7. The ground truth is represented by the red line marked with circles and the path generated by our algorithm is shown as the solid black line. During the test the camera travelled a distance of 1.8 m with the separation of 20 cm between each image capture. The final position error was equal to 0.096 m (9.6 cm) which is equal to 5.34 % of the overall distance travelled. The graph with position error as a function of distance travelled is given in Fig. 8.

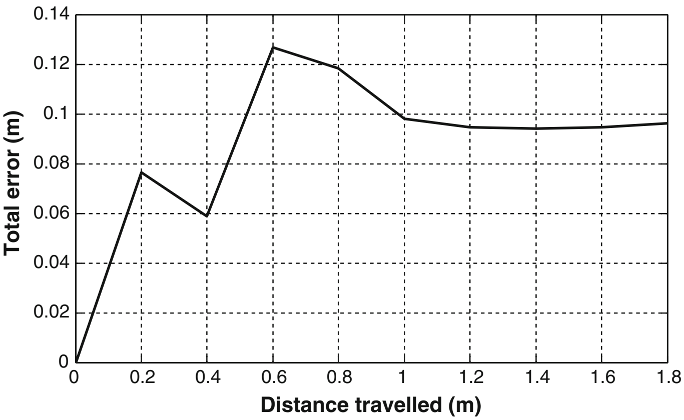


Fig. 8 Position error in office test

4 Conclusions and Future Work

In the paper a navigation method based on visual odometry was presented. The system uses monocular camera for capturing images of the environment and additional sensor (IMU or handmade measurements) that is used to provide the scale of translational motion. The method has been evaluated on indoor as well as outdoor datasets. Most of the visual odometry algorithms developed nowadays are based on stereo-vision cameras as they provide more accurate results. However, in the paper it has been shown that it is possible to navigate a ground vehicle for short distances using data from the monocular camera. For long-distance travels further improvements of the algorithm should be implemented. Especially errors arising from orientation estimation should be reduced. In the future it is planned to implement several modifications of the algorithm in order to improve its accuracy.

First improvement would be to revise orientation estimation algorithm. It might be beneficial to estimate rotation of the vehicle independently, based only on the points that are far away from the vehicle and their distance is much larger than translational motion of the vehicle. In such case the "pure rotation" motion mode can be assumed, which simplifies calculations.

Second improvement would be to develop a Kalman filter that incorporates motion model of a vehicle. Implementation of the model will reduce errors arising from observation noise. This approach requires development of a motion model as well as an error model for visual odometry. Additionally, information about expected movement of the robot will be beneficial in outlier rejection procedure. It could provide speed-up in calculation process and increase accuracy of the process.

Final improvement would be implementation of the pose graph optimization procedure. It is expected to additionally reduce drift errors arising from additive characteristic of the error in the algorithm.

In a scenario for which robot is being developed it is expected that it will have to travel autonomously no more than tens of meters. With planned modifications the developed algorithm is expected to provide sufficiently accurate position to recreate path travelled by the robot indoors and to return without supervision in case of loss of communication.

References

1. Badino, H., Yamamoto, A., Kanade, T.: Visual odometry by multi-frame feature integration. In: International Workshop on Computer Vision for Autonomous Driving @ ICCV (2013)
2. Konolige, K., Agrawal, M., Sola, J.: Large-scale visual odometry for rough terrain. In: Kaneko, M., Nakamura, Y. (eds.) Robotics Research. Springer Tracts in Advanced Robotics, vol. 66, pp. 201–212. Springer, Berlin (2011)
3. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. J. Field Robot. **23**(1), 3–20 (2006)
4. Scaramuzza, D., Siegwart, R.: Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. Robot. IEEE Trans. **24**(5), 1015–1026 (2008)

5. Lategahn, H., Geiger, A., Kitt, B., Stiller, C.: Motion-without-structure: real-time multipose optimization for accurate visual odometry. In: Intelligent Vehicles Symposium (IV), IEEE, pp. 649–654 (2012)
6. Fraundorfer, F., Scaramuzza, D.: Visual odometry: part II: matching, robustness, optimization, and applications. Robot. Autom. Mag. IEEE **19**(2), 78–90 (2012)
7. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. Robot. Autom. Mag. IEEE **18**(4), 80–92 (2011)
8. Rosten, E., Drummond, T.: Machine Learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision â€ ECCV 2006. Lecture Notes in Computer Science, vol. 3951, pp. 430–443. Springer, Berlin (2006)
9. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571 (2011)
10. Stewenius, H., Engels, C., Nister, D.: Recent developments on direct relative orientation. ISPRS J. Photogrammetry Remote Sens. **60**(4), 284–294 (2006)
11. Horn, B.K.: Recovering baseline and orientation from essential matrix. J. Opt. Soc. Am. (1990)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Rob. Res. (IJRR) (2013)

Path Planning Algorithms for Autonomous Mobile Platform

Krzysztof Jaskot and Radomir Mysliwiec

Abstract Autonomous mobile platform (AMP) is a machine that can operate in a human-made environment. In this paper was considered problem of automatic control of mobile platform using information from GPS system, electronic compass and encoder. The mobile platform is equipped in two-stroke glow engine, heavy-duty drive train and wide-track suspension and controller based on ARM7 microcontroller and using MaxStream XBee Pro 2.4 GHz radio modem communication module. The paper describes examples of algorithms that use GPS, encoder and electronic compass. Results of work on autonomous mobile platform that can operate in human environment are presented. The obtained properties of the system have been affected that it can be used for future research and autopilot design project.

Keywords Mobile robots · Navigation · Path planning

1 Introduction

The aim of the project was to create an autonomous mobile platform (AMP), which could operate in open terrain. Autonomous mobile platform is a machine that can operate in a human-made environment [1–6]. The key to autonomy is a control system built on the basis of information concerning the position and goal [7, 8]. Use GPS NMEA (National Marine Electronic Association) protocol allows you to obtain information in text form about the current location of the object. After adding information about the intermediate target points (Waypoint), we can receive information about the current direction [9]. The work was considered problem use information derived from the GPS, rotation sensor—encoder and IMU (Inertial Measurement System) [10] as a source of control signal. IMU currently provides

K. Jaskot (✉) · R. Mysliwiec
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: krzysztof.jaskot@polsl.pl

A. Nawrat and K. Jędrasiak (eds.), *Innovative Simulation Systems*,
Studies in Systems, Decision and Control 33,
DOI 10.1007/978-3-319-21118-3_25

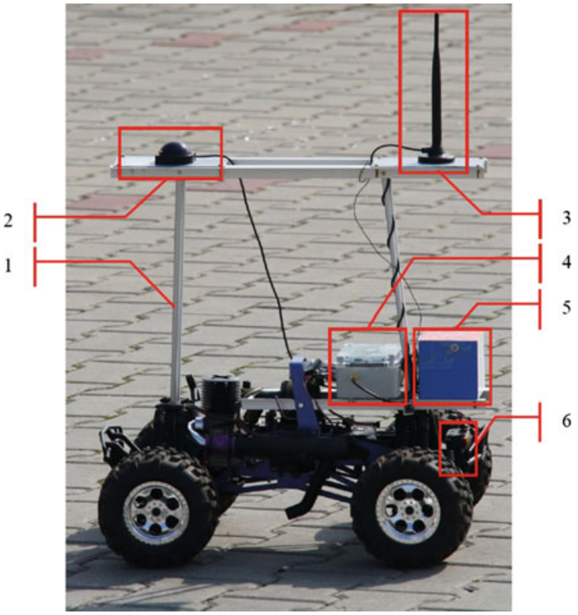


Fig. 1 The mobile platform used in the tests. *1*—aluminum frame, *2*–GPS, *3*—antenna 2.4 GHz, *4*—main controller, *5*—IMU, *6*—batteries

only information from the electronic compass. This information is needed to determine the azimuth of mobile platform. Information taken from three types of sensors (GPS, encoder, and electronic compass) is used as the basis for the implementation of navigation algorithms[11–13]. For testing we used the mobile platform built using a remote-controlled car on a scale 1/8th (length 55 cm, width 43 cm) and it was delivered by the HPI Racing. We selected terrain model with an independent suspension and four-wheel drive (4WD), because we wanted to create an autonomous platform that can operate in open terrain. The appearance of mobile platform with installed controller, GPS system, IMU and communication antenna is shown in Fig. 1. As the propulsion system used in this model, two-stroke internal combustion engine with a capacity of 3.5 cm^3 and 2HP (Horse Power). This allows the dispersal model to speed about 60 km/h. In addition to the chassis and drive train in the composition of the platform includes two servos, which are responsible for controlling the throttle/brake and course. These two servos give us the ability to control the platform traction [14]. Detailed description of the construction of the mobile platform is presented in [15].

2 Algorithms

In this section we will describe variations of steering algorithms implemented in AMP. They are divided by the complexity and sensors used [16].

2.1 GPS-Based

First successfully implemented algorithm was described in [17, 18]. It is based on GPS measurement. But as it came out, it is hard to receive enough information to control an autonomous vehicle solely from GPS. The main problem was to obtain good estimation of AMP heading in certain situations. That is why, azimuth from the magnetometer is additionally used [19]. Great advantage of using GPS is that we can get all the information we need from a single sensor! Furthermore, it gives good results without any additional mathematics involved. Even the simplest code will keep autonomous vehicle on track (in defined margins of error) for hours and hours, without need of help from any other method, as GPS error is non-cumulative —positioning is very reliable and predictable [20]. But of course, there are some drawbacks of using GPS. First of all, using it as the only source of position and speed estimation, algorithm is suffering from dramatically low refresh rate, equal to 1 Hz. Combined with latency caused by lot amount of information transferred in low baud rates (4800 bps), it is impossible to track sharp maneuvers without additional information. More importantly, GPS does not always have same, sufficient accuracy. GPS accuracy is affected by a number of factors, including satellite positions, noise in the radio signal, errors in transmitted clock, atmospheric conditions, and natural barriers to the signal [21] . Noise can create an error between 1 and 10 m and results from static or interference from something near the receiver or something on the same frequency. Objects such as mountains or buildings between the satellite and the receiver can also produce error, sometimes up to 30 m. The most accurate determination of position occurs when the satellite and receiver have a clear view between each other and no other objects interfere. In other case, aforementioned factors will affect GPS accuracy. To overcome or get around these factors, other technology, AGPS, DGPS, and GPS augmentation, has been developed to aid in determining an accurate location [22]. However, main task for this project was to find a best solution for route tracing, with use of available equipment. So, while there is no doubt that technologies such as DGPS would greatly improve GPS accuracy, we will not use it on this stage of the project yet.

2.2 Encoder-Based

This variation was made to test how AMP estimates its position without using absolute position measurement [23]. It is important to notice, that encoder on the drive shaft is a very important sensor, when taking into consideration algorithms of car localization, as it is a good example of odometry—which is already present in every car. After switching up AMP algorithm is only using GPS once, to determine the starting position and subscribe it as beginning of the coordinate system against geographical coordinates. That is only done for the test convenience, and has no influence on AMP's algorithm further estimation uncertainty. When the starting position is known, AMP is only using magnetometer from IMU and encoder on the drive shaft. The mathematics of this algorithm are very simple. It was measured, that one tick of the encoder translates into distance covered in a following way:

$$Enc_{tick} = 14.52\ \text{cm} \tag{1}$$

Output from the magnetometer is taking values from the interval [−32768, 32767]. This can be easily transferred to angle expressed in radians:

$$\Theta = \frac{azim\pi}{32768} \tag{2}$$

Using this two values (as r and Θ respectively), it is possible to present AMP movement in polar coordinate system. However as geographical position is presented in Cartesian system, we have transform it once again (Fig. 2).

$$X_{est} = \sin(\Theta)r \tag{3}$$

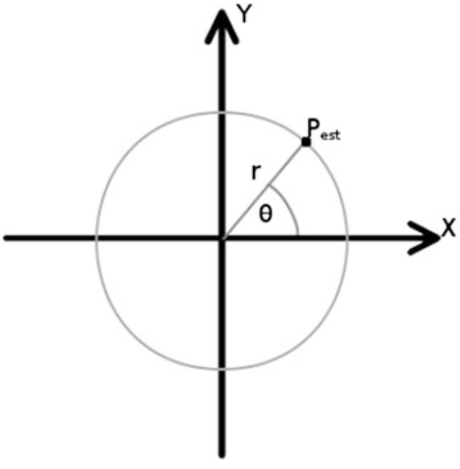
$$Y_{est} = \cos(\Theta)r \tag{4}$$

where:

r —distance traveled between ticks of the encoder,
Θ —actual azimuth

This formula is adequate representation of the movement in a straight line, although it is obvious that it cannot represent properly the movement of AMP on the turns. That is why, in a case when current azimuth measured during the tick is

Fig. 2 Representation of AMP movement in polar coordinate system



different from the previous one, we assume, that the car was driving on a uniform curve between those two points. In most cases, this will hold true, as direction servo changes its setting significantly less often than the tick of encoder occurs. Fortunately, we can use the same formula as previously, to calculate change in position during a turn, where:

l—path of AMP of known length
c—length of the chord between *P* and beginning of coordinate system (Fig. 3).

To calculate *r*, we threat the arc between *P* and beginning of coordinate system, as a base for sector of a circle, where:

l—distance made between ticks (known constant)
Θ_t—difference of azimuth between ticks

Then, from a simple formula we obtain radius R of a circle on which our arc is based:

$$R = \frac{l}{\Theta_t} \tag{5}$$

And finally, from a formula for isosceles triangle basis:

$$c = R\sqrt{2(1 - \cos(\Theta_t))} \tag{6}$$

Odometry is based on simple equations that are easily implemented, and that utilize data from inexpensive incremental wheel encoders. However, odometry is based on the assumption that wheel revolutions can be translated into linear displacement relative to the floor. This assumption is only of limited validity. Generally, errors can be divided into two categories, according to their sources:

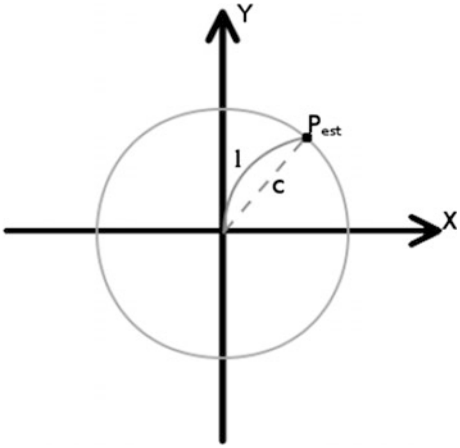


Fig. 3 Representation of position’s change during a turn

systematic errors and non-systematic errors. In order to reduce odometry inaccuracies it is very important to understand the difference between those two groups. Systematic errors are predictable and expected. They are connected to imperfect calibration of measurement devices, changes of the environment interfering with the measurement process or imperfect methods of observation. Those errors are grave because they accumulate constantly. Odometry systematic errors are for example:

- unequal wheel diameters
- difference between nominal wheel diameter and average of actual diameter
- finite encoder resolution
- finite encoder sampling rate

Non-systematic errors are difficult to predict and may appear unexpectedly. Those errors may cause large position inaccuracies. Odometry non-systematic errors are for example:

- travel over uneven floors
- travel over unexpected objects on the floor

That is why, this kind of mobile robot position measurement is rather inadequate for periods of time that exceed a few minutes. The biggest advantages of dead-reckoning based measurements are:

- good precision in estimation of small movements
- they have very high possible sampling rate
- not as vulnerable to external disturbances as IMU or GPS

While the disadvantages are:

- even slightest uncertainty can cause massive error over time
- have a lot of different sources of uncertainty

When not being equipped into many wheel encoders, it is impossible to estimate direction of movement. That is why for dead-reckoning algorithm AMP additionally uses inertial navigation. This combination is less affected by systematic errors of positioning that algorithms based solely on encoders, but as IMU's readings are also a victim of many sources of uncertainty, the cumulative error is still a case to deal with [24–26].

2.3 Combination of GPS and Odometry

The next step was to combine two previous algorithms—as one is representing absolute position estimation, and the second one uses purely relative methods. It was pointed out, that the biggest problem of dead-reckoning is cumulative error. Although, on short distances it can be by far more precise than GPS. Taking it into consideration it seemed like a good idea to leave odometry as base estimation source, while constantly correlating it by GPS position measurement. This idea can be written down in a general formula:

$$est_{final} = \frac{est_{odo} + wGPS_{tick}}{1 + w} \tag{7}$$

where: w - weight $\ll$1.

The main advantage of this algorithm, against the one using exclusively GPS, is that when GPS satellites are temporarily invisible (for example when entering a tunnel) or there is generally a problem with satellite visibility (caused by trees, dense clouds or so called “urban canyon”) AMP can still drive, maintaining good position estimation during short periods of GPS signal absence. The problem is, that if estimating position would be constantly corrected by poor actual GPS readings, this could in fact make estimation worse than the one calculated from odometry. On the other hand, with good GPS signal strength, the estimation of position can be very precise. That is why the final formula for the estimation is using weight in relation to actual GPS signal quality:

$$est_{final} = \frac{est_{odo} + w_{quality}GPS_{tick}}{1 + w_{quality}} \tag{8}$$

Whole calculation is done by this short excerpt from the code of AMP:

```
float w = DATA_RD_wgAlgo_int16; //value remotely set by the
        user
if(w<=0 || w>=1000) //if out of boundaries, set to 1
w = 1;
//multiply weight by number of GPS satellites
w * = 0.001*DATA_RD_gpsSV_int16;
estPosX  =(estPosX  +  data_read(DATA_gpsX_INT32)._int32_
          t*w)/(1 + w);
estPosY  =(estPosY  +  data_read(DATA_gpsY_INT32)._int32_
          t*w)/(1 + w);
```

This approach is probably the simplest way of joining these two different families of positioning. However even algorithm that is based on this combination should be able to compensate biggest weaknesses of two methods, which are:

- low refresh rate of GPS
- cumulative error of dead-reckoning

Additionally using encoder tick as update signal for position estimation, automatically prevents drift of position estimation, caused by GPS.

2.4 Magnetic Azimuth

During measurements, it became clear, that there is unnaturally great error source in odometry-IMU based algorithm. During test drives AMP constantly drifted in north-east or north-west direction. Drift had various intensity, but direction was always the same. Even, when test drives were consisting of doing laps on a circular-like circuit. That fact, excluded most sources of errors, because every source that could come from encoder's part, should be symmetrical in this case. The same applies to ellipsoid-shaped characteristics of magnetometer reading, or incorrect magnetic declination. To find the source of this error, a simple test was made. The car was switched to manual steering, and was driven to do 10 laps on a minimal radius (Fig. 4). As we can see on the picture, the error is very significant, in comparison to traveled distance (approximately 10 m of drift after 60 m traveled). When looking closer, one can notice, that error is mostly limited to the moment, when car is facing west. From the chart, we can conclude, that in this moment of

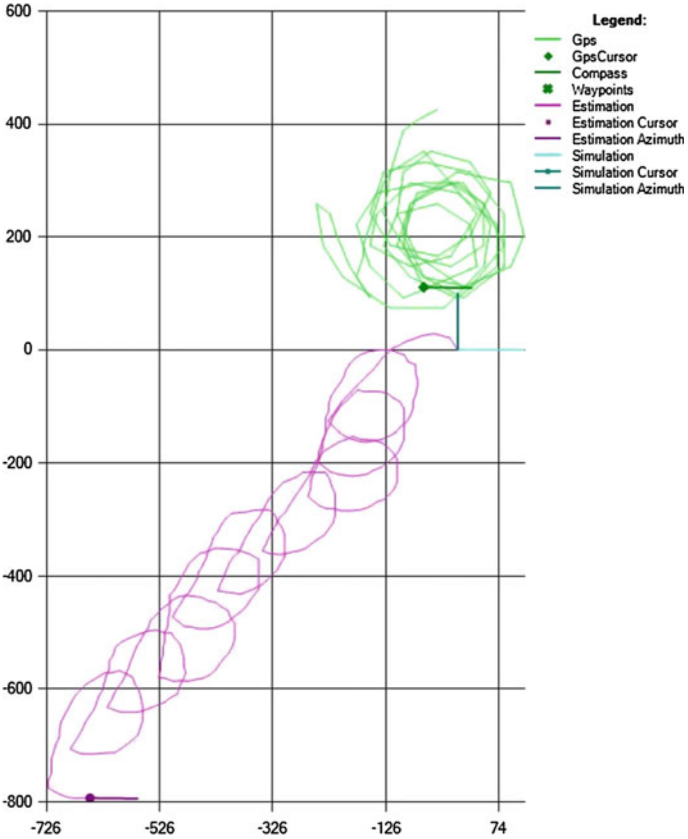


Fig. 4 Graph showing error caused by magnetometer's azimuth uncertainty

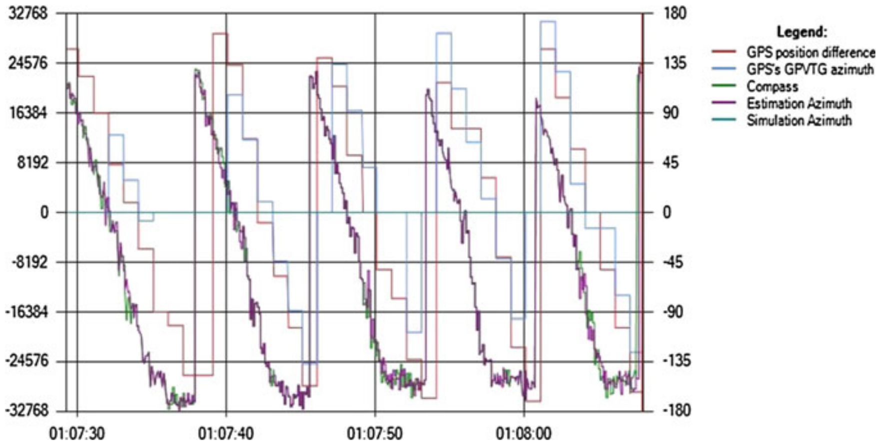


Fig. 5 Representation of magnetometer's azimuth change over time

circular motion, the reading of azimuth flatten some range of directions, into one value. More leads are shown, when looking at plot of azimuth against time (Fig. 5). Here, it is important to point out, that the car was running at constant speed almost all the time. While for the values greater than −25,000, function is rather linear, below this value, plot is non-linear and noisy. As IMU was designed to work in three dimensions, it should point magnetic azimuth not only when it is parallel to earth surface, but also in any possible arrangement. Fortunately, in contrast to aircrafts or missiles, AMP is only driving in parallel to the surface, with small variations of when being perpendicular to gravity vector. Thanks to this fact, we could assume that magnetic vector is presented by only 2 axis—neglecting Z axis. This assumption brings us to much simpler function of azimuth (9). For this purpose, raw data of magnetometer's vector values were used.

$$azimuth = arctan(\frac{m_x}{m_y}) \tag{9}$$

where:

m_x,m_y—normalized values of magnetic vector.

To implement this formula, we are using function *atan*2, which is a variation of arctangent function.

```
magX = (float)DATA_RD_magRawX_int16; //raw values of X and Y
       axis
magY = (float)DATA_RD_magRawY_int16; //of strenght of mag-
       netic field
norm = sqrt(magX*magX + magY*magY); //lenght of vector
```

```
magXn = magX/norm;
magYn = magY/norm;
heading = atan2(magYn,magXn); //azimuth
heading += 0.0738856; //local magnetic declination
if(heading>2*M_PI)
heading -= 2*M_PI;
```

Where *magX* and *magY* are raw data obtained from magnetometer.

There is no need to know the data range that magnetometer is using, as we just normalize this values to get the direction of the magnetic vector. After this, we take into consideration the actual magnetic declination on our coordinates (this is necessary to compensate the difference between magnetic north and geographical north). For our coordinates actual magnetic declination is equal to 4°14′E [27]. At the end, we have to check if this correction didn't cause heading to increase above full 360°. After applying this formula (9), once again, the exactly same test was made as in previous point (Fig. 6). Test was made with the exactly same circle radius, and with the same speed [28, 29]. The improvement is clearly visible. While uncertainty around the same place is still occurring, it is smaller and less influential. With use of this formula drift was around 5 m after 60 m of distance traveled.

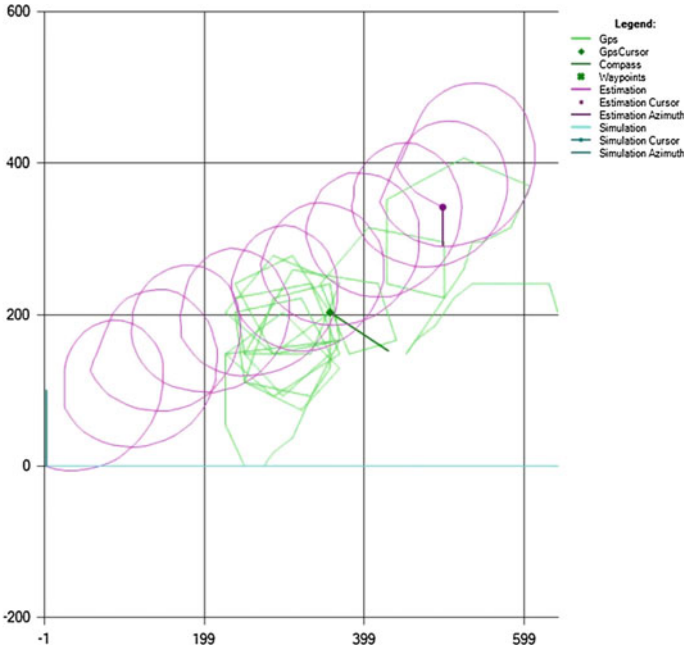


Fig. 6 Estimation of position after changing the formula for magnetic azimuth

2.5 Waypoint Finding Method

As proposed steering algorithms are not so precise yet, and there is no sensor used to implement obstacle avoidance, it was sufficient to use simple waypoint based route tracing to check its possibilities. The principal of operation is very simple. Algorithm compares our current estimated position with position of the waypoint and calculates the direction in which AMP would have to go to pass through it, then it checks AMP's current heading and steer to obtain the situation where both directions will be the same. There is defined distance in which AMP has to be to mark waypoint as passed, when distance between AMP and waypoint is estimated to be lower than this value, AMP recognize waypoint as passed and switches to the next one.

```
//if current algorithm is based on GPS
//distance needed to mark WP is greater
if((DATA_RD_nrAlgo_int16) == 0)
gpx = 2;
else
gpx = 1;
if (DistToWP < (50*gpx))
{
//next Waypoint
WP_curr++;
WP_curr %= WP_total;
DATA_WR_ctlWP_uint8(WP_curr);
wp_change = 1;
}
```

3 Tests

This section will present results of experimental measurements using all the variations of steering algorithm. It is important to notice, that all the test were made in the exactly same conditions, on the same area. With cloudless sky and no high objects around. This led to very good GPS accuracy, which is not guaranteed every time of use. Every measurement was performed once for each algorithm. So presented data is not chosen best from multiple tries.

3.1 Test no. 1—The Far Point

First experiment was designed to examine precision of measurement and repeatability. Firstly, vehicle was manually driven across certain checkpoints to the

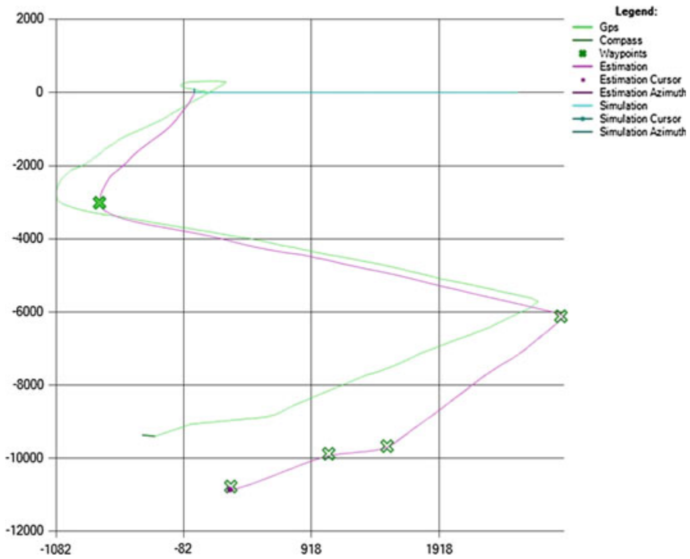


Fig. 7 Reference run as seen by odometry and GPS

ending point. In this case, AMP was not using steering algorithm to control the car, however it was constantly estimating its position in relation to the starting point. Basing on this estimation, waypoints were marked on the map in Autonom, in the places where characteristic checkpoints should occur (Fig. 7).

3.1.1 GPS-Based

First to be tested was GPS-based algorithm. The accuracy of this trial was better than expected. None of the checkpoints was passed by more than half a meter away, however as it is presented on the record (Fig. 8) it was not a clear run. Despite the fact that sometimes AMP was actually passing through the checkpoint, algorithm was not considering it as passed, and vehicle had to turn around. It can be caused by two things. First is obviously measurement uncertainty (vehicle passed the waypoint, but estimated position was different). Second theory is that waypoint is marked as passed only if estimation point is closer than 50 cm from the waypoint, and frequency of GPS signal refresh causes, that with speed around 5 km/h it can travel 1.4 m between signal ticks. So with a little of luck, algorithm can miss the moment of passing through the waypoint. That is why in next tests—only for GPS-based algorithm—distance needed to confirm the waypoint pass has been changed to 150 cm.

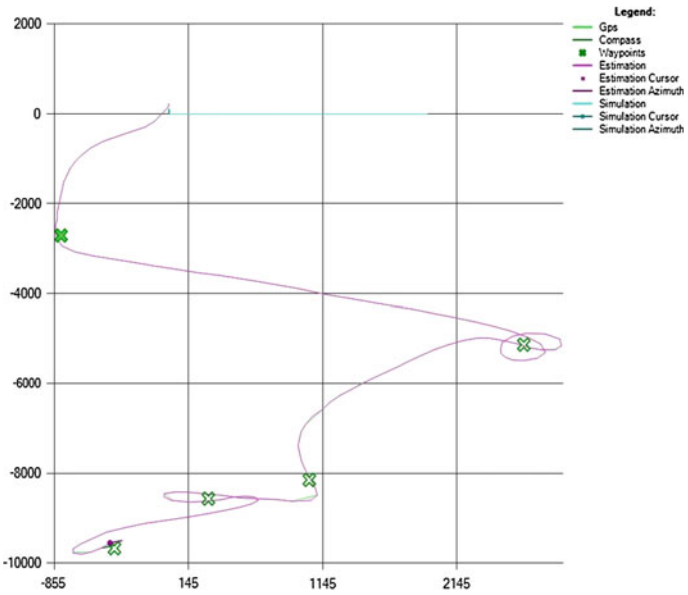


Fig. 8 GPS-based algorithm's record of the far point test

3.1.2 Encoder-Based

As we can see on the Fig. 9 cumulative error is growing to significant numbers over such a distance. It is important to spot, that while the distance measurement from the encoder is quite precise, there is visible difference in azimuth measured by magnetometer and the real heading of AMP. This uncertainty is different depending on direction, what is probably caused by mathematical model (as it was described in the previous section). Apart from that, repeatability of two runs was quite good. First two checkpoints were passed with difference smaller than 50 cm. Then AMP started to drive away from the route, but final checkpoint was still about 3 m from the original one.

3.1.3 Combined

In this run (Fig. 10), weight of the GPS reading in position estimation was set to ~2 %. Despite the fact that at the end, GPS reading is different than estimated position (which suggests, that weight was too small to correct the cumulative error caused by magnetometer uncertainty in this directions) this test was said to measure repeatability of runs. As it came out, this run was almost the same as reference run, with vehicle passing over all 5 checkpoints within width of the car.

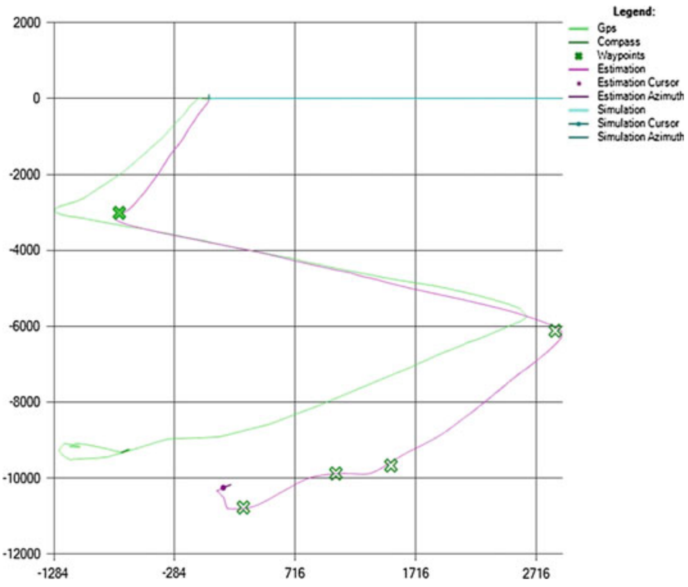


Fig. 9 Encoder-based algorithm's record of the far point test

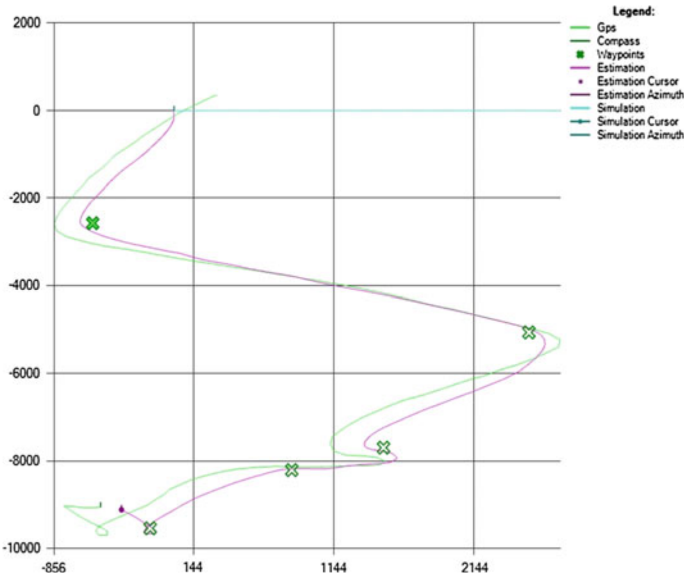


Fig. 10 Combined algorithm's record from the far point test

3.2 Test no. 2—The Circuit Test

This task was prepared to show how algorithms are handling driving along same route repeated multiple times. Records collected during this simple test, show best the main differences, advantages, and weaknesses of each approach.

3.2.1 GPS-Based

As it cannot be seen on the picture, first thing to say is that during this task AMP (when using this algorithm) never missed a waypoint by more than 1 m. First thing which is noticeable when analyzing Fig. 11, is that route made by AMP is surprisingly uneven. This could be somehow strange, as GPS receive quality was very good (9 satellites in view) and the distance between waypoints was not small enough to cause trouble with turning radius (10 m between each consecutive one). The source of this problem is also visible on the visualization. Namely, clear discretization of route estimation is caused by the biggest drawback of GPS-measurement rate. As frequency of GPS tick is 1 Hz, and NMEA 0183 protocol has bit rate equal to 4800 bps (which is causing additional latency of about 100 ms) the information that we get about our position can be as old as 1.1 s! With this refresh rate, even at small speeds vehicle can pass distance sufficient to miss the opportunity to correct its way to checkpoint. When maintaining predictable direction and speed, this can be corrected by software estimation, but when it comes to making sharp turns and handling speed changes, additional data has to be known.

3.2.2 Encoder-Based

As the axis of this circuit (Fig. 12) is placed in such a way, that AMP had to drive in directions causing the biggest errors of magnetometer, this run is close to the worst case scenario for the encoder-based algorithm. The error was so great, that vehicle had leaved the measurement area a long before the test should end.

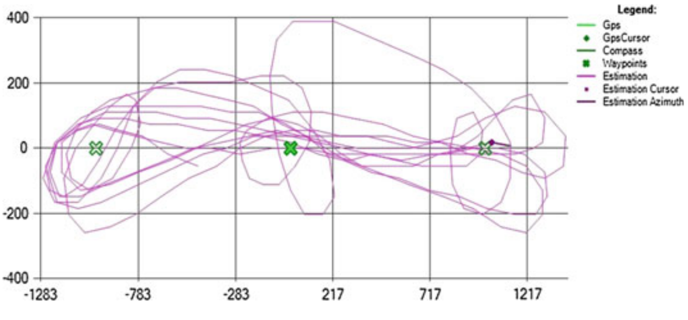


Fig. 11 GPS-based algorithm's record of the circuit test

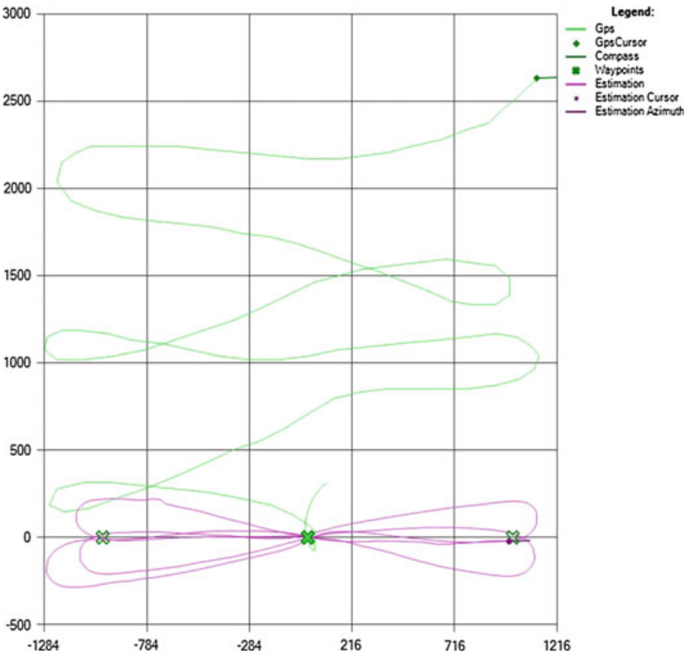


Fig. 12 Encoder-based algorithm's record of the circuit test

3.2.3 Combined

Having in mind, that magnetometer's measurement in this task was extremely erroneous, the weight of the GPS in final estimation was set to w = 4 (with 9 satellites in view this gives 3.47 % of total value). During this test, after first run the car was passing all the waypoints in distance (depending on waypoint) that ranged from 0 to 100 cm. Although, the most interesting observation is not the precision of waypoint passing, but the great repeatability of laps, plainly shown on the Fig. 13. This is even

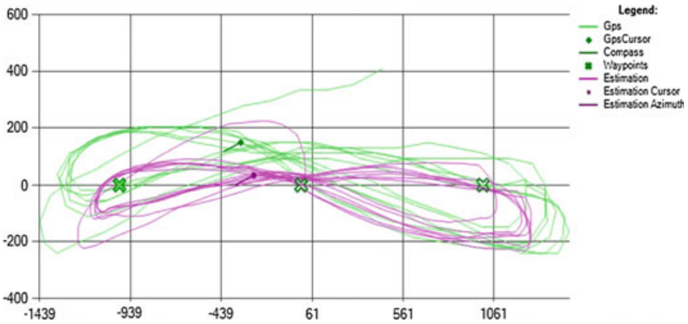


Fig. 13 Combined algorithm's record of the circuit test

more surprising, considering the performance of AMP with GPS- and encoder-based algorithms separately. Some very important conclusions can be made basing on this experiment. Firstly: it is proving that combining two different methods can efficiently overcome drawbacks of both. Secondly: it is very important for future project development, that route following of AMP can be so predictable, because it means, that with right mathematical model, errors of this method can be compensated.

3.3 Test no. 3—The Slalom Test

This test consisted of 7 tokens placed on the ground in precise reference to the geographical coordinates. Thanks to this effort, it was possible to observe precision of finding route between waypoints in a single run, from a fresh start.

3.3.1 GPS-Based

First algorithm let AMP to pass waypoints in distance not exceeding 20 cm from the back of the wheel, which is about 40 cm from the car's center. However, as it was seen in previous tests, we can see from the Fig. 14 how the car was maneuvering in this run. The cause of this behavior was already explained in points 3.1.1 and 3.2.1.

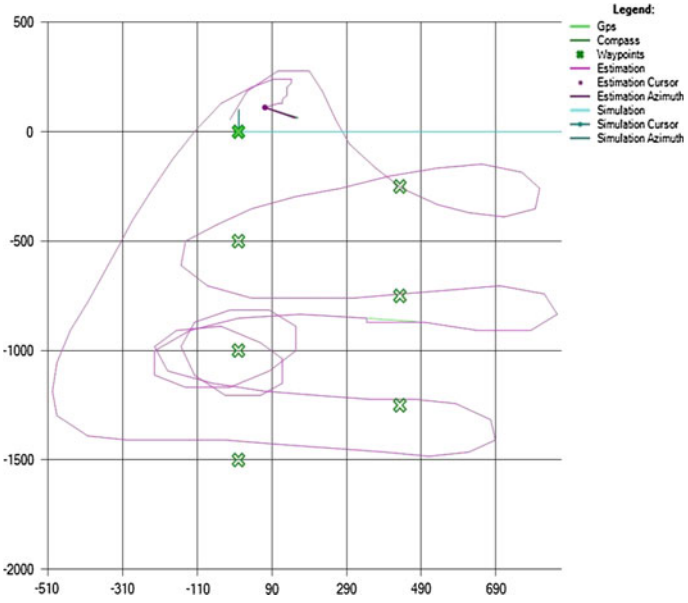


Fig. 14 GPS-based algorithm's record of the slalom test

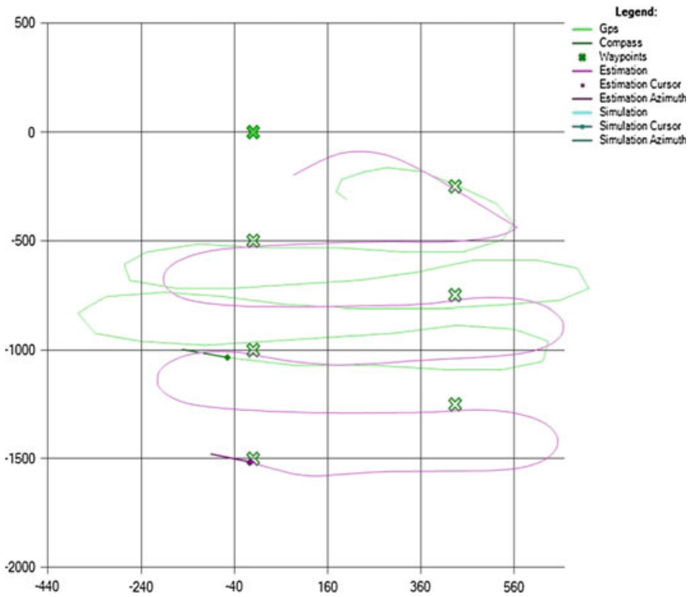


Fig. 15 Encoder-based algorithm's record of the slalom test

3.3.2 Encoder-Based

In this run, from the operator's fault, starting point was displaced 2 m further. Apart of this, with little correction, test went satisfactorily. Algorithm based on the encoder is driving a lot smoother than GPS-based, however it loses precision on every turn, which results in error growing up to 5 m at the final waypoint (Fig. 15).

3.3.3 Combined

First run of the combined algorithm resulted in passing all the point within distance of 60 cm from the back of the car's wheels, what is equal to total of ~80 cm from the car's center. As the experiment went a lot better than with other algorithms, operator let the AMP to do several more laps to present more data. Not including the beginning of the first try, concentration of vehicle's traces would let it to drive the same route on the track having width of an average sidewalk. However, this would probably require a lot of changes in steering algorithm, so that the turning radius was predictable (Fig. 16).

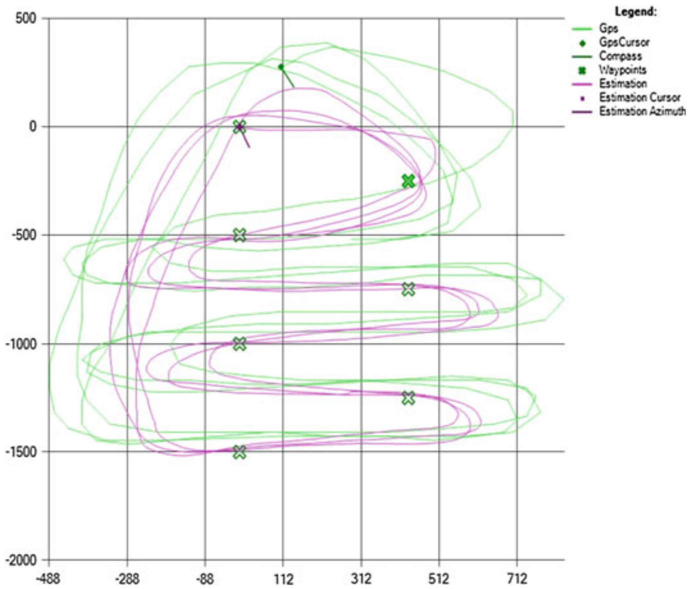


Fig. 16 Combined algorithm's record of the slalom test

3.4 Comparison of the Results

Before summing up the results, there is a quick comparison in Table 1: During the tests it was easy to observe the weaknesses of GPS- and Encoder-based position estimation methods. GPS is probably the easiest method of position estimation nowadays, and in contrast to others—it is giving immediate results, without any calculations or additional transmitters. That makes it a great sensor to begin with when it comes to position estimation. However, as it is sufficient when it comes to long and not complicated routes, it can't be the only source of position estimation in projects such as AMP. Mostly because of it's low refresh rate GPS is not good when it comes to fast decision making. Dead-reckoning algorithm is providing element of smooth route by frequent position estimation refreshes and very good precision in short periods, however it extreme conditions it can have uncertainty of about 20 % which is making it impossible to drive on longer distances. Combination of this two algorithms is not gaining precision in long time position estimation, as it is slowly averaged to the position estimated by GPS. But it don't loose it either—while gaining all the profits of dead-reckoning navigation, what makes it much better in route following (Table 1).

Table 1 Short comparison

Test	GPS based	Encoder based	Combined
Test no. 1	– max. 0.7 m difference on waypoints – best absolute position estimation – had to turn around to hit waypoint	– difference on waypoints grew up to 3 m – worst absolute position estimation –smoothest driving	– max. 0.4 m difference on waypoints – good absolute position estimation – smooth driving except 1 moment
Test no. 2	– max. 1 m from the waypoint – very uneven drive	– drived out of the test area	– max. 1 m from the waypoint – very repeatable drive – absolutely smooth route
Test no. 3	– max. 0.4 m from the waypoint – had a couple of tries to hit one of the waypoints	– distance from the waypoint grew to 5 m	– max. 0.8 m from the waypoint – after multiple tries route was the same every time

4 Conclusions

The goal of this project was to analyze the existing Autonomous Mobile Platform and to find a way to improve it's functionality concerning route planning. Author focused on using available sensors to experiment with different approaches on positioning problem, analyzing their strengths and weaknesses to find the best solution based on available equipment. After investigation on existing ways of positioning, author decided to compare two main methods: dead-reckoning and absolute measurement. For this purpose, he prepared one algorithm of each method and performed tests to show their results. Algorithm which was previously used by AMP (based on GPS) already proved that it is able to drive between waypoints with good results. However, it was only aiming to pass closer than 3.6 m from the waypoint before going to another one. If this value was smaller, vehicle had tendency to drive around waypoint in effort to pass it—if value was smaller than 1 m it was even able to "lock" itself in infinite loop, driving around waypoint with maximal turn radius. By improvement of azimuth calculation it was possible to lower this value to 1.5 m, but still this precision was not sufficient for precise route following. Many problems were caused by GPS frequency of measurement equal to 1 Hz. Dead-reckoning based on encoder ticks and magnetometer's azimuth is refreshing position estimation much faster and thanks to this—it is driving much smoother. Unfortunately, uncertainty of used sensors build up significant cummulative error over time, and make it impossible to drive longer routes basing only on this method. However, as it came out, it is sufficient to correct dead-reckoning by averaging its estimation value with the one of GPS in proportions of 100:3 to neglect the influence of cumulative error and obtain much better route following then GPS itself. In this way author received algorithm that is much more suitable for

AMP for the purpose of route following. Test have proven that, although it did not make better results by passing closer to the waypoints—it took much better route and (in contrast to solely GPS-based) was unreliable in finding its way near the waypoint.

References

1. Baker, C.R., Dolan, J.M.: Street smarts for boss–behavioral subsystem engineering for the urban challenge. In: Robotics and Automation Magazine, IEEE. Carnegie Mellon Univ., Pittsburgh (2009)
2. Behringer, R., Maurer, M.: Results on visual road recognition for road vehicle guidance. In: Proceedings of the 1996 IEEE Intelligent Vehicles Symposium, Tokyo, 1996
3. Braun, T., Schäfer, H., Berns, K.: Topological large-scale off-road navigation and exploration —RAVON at the European land robot trial 2008. In: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, Oct 2009
4. Freire, E., Bastos-Filho, T., Sarcinelli-Filho, M., Carelli, R.: A new mobile robot control approach via fusion of control signals. Syst. Man Cybern. Part B Cybern. IEEE Trans. **34**(1), 419–429 (2004)
5. Harkins, R. et al.: Design and testing of an autonomous highly mobile robot in beach environment. In: Proceedings of the World Congress on Engineering and Computer Science 2008 WCECS 2008, San Francisco, USA, 22–24 Oct 2008
6. Li, T.H.S., Chang, S.J., Tong, W.: Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. Fuzzy Syst. IEEE Trans. **12**(4), 491–501 (2004)
7. Babiarzm, A., Jaskot, K.: The concept of collision-free path planning of UAV objects. In: Advanced Technologies for Intelligent Systems of National Border Security, pp. 81–94 (2013)
8. Kus, Z., Fras, S.: Helicopter control algorithms from the set orientation to the set geographical location. In: Advanced Technologies for Intelligent Systems of National Border Security, pp. 3–13 (2013)
9. Shair, S., Chandler, J.H., Gonzalez-Villela, V.J., Parkin, R.M., Jackson, M.R.: The use of aerial images and GPS for mobile robot waypoint navigation. Mechatron. IEEE/ASME Trans. **13**(6), 692–699 (2008)
10. Soloviev, A.: Tight coupling of GPS and INS for urban navigation. Aerosp. Electron. Syst. IEEE Trans. **46**(4), 1731–1746 (2010)
11. Galuszka, A., Pacholczyk, M., Bereska, D., Skrzypczyk, K.: Planning as artificial intelligence problem—short introduction and overview. In: Advanced Technologies for Intelligent Systems for National Border Security, pp. 95–103 (2013)
12. Bibik, P., Gradolewski, S., Zawislak, W., Zbudniewek, J., Darakchiev, R., Krczel, J., Michalski, M., Strzelczyk, K.: Problems of detecting unauthorized satellite transmissions from the VSAT terminals. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–4, 8–9 Oct 2012. ISBN: 978-1-4673-1422-0
13. Bibik, P., Narkiewicz, J.: Helicopter optimal control after power failure using comprehensive dynamic model. J. Guid. Control Dyn. **35**, 1354–1362 (2012)
14. Bibik, P., Narkiewicz, J.: Helicopter modeling and optimal control in autorotation. In: Annual Proceedings—American Helicopter Society, vol. 64, no. 2, p. 986 (2008)
15. Babiarz, A., Jaskot, K., Koralewicz, P.: The control system for autonomous mobile platform. In: Advanced Technologies for Intelligent Systems of National Border Security Studies in Computational Intelligence, pp. 15–29. Springer, Heidelberg (2013)
16. Babiarz, A., Bieda, R., Jedrasiak, K., Nawrat, A.: Machine vision in autonomous systems of detection and location of objects in digital images. In: Vision Based Systems for UAV

Applications, Studies in Computational Intelligence, vol. 481, pp. 3–25 (2013). ISBN: 978-3-319-00368-9

17. Sroka, M., Sciegienka, P., Babiarz, A., Jaskot, K.: Prototyp bezzalogowego pojazdu podwodnego - uklad stabilizacji i utrzymania zadanego kursu. Przeglad Elektrotechniczny **89**, 205–217 (2013)
18. Jaskot, K., Babiarz, A., Sroka, M., Sciegienka, P.: Prototyp bezzalogowego pojazdu podwodnego - konstrukcja mechaniczna, panel operatora. Przeglad Elektrotechniczny **89**, 52–67 (2013)
19. Bieda, R., Grygiel, R.: Wyznaczanie Orientacji Obiektu w Przestrzeni z Wykorzystaniem Naiwnego Filtru Kalmana. Przeglad Elektrotechniczny **90**, 34–41 (2014)
20. Daniec, K., Jedrasiak, K., Koteras, R., Nawrat, A.: Embedded micro inertial navigation system. In: Applied Mechanics and Materials, vol. 249, pp. 1234–1246 (2013)
21. Barnat, W., Niezgota, T., Panowicz, R., Sybilski, K.: The influence of conical composite filling on energy absorption during the progressive fracture process. WIT Trans. Model. Simul. **51**, 625–633 (2011)
22. Ayman El-Fatatry: MEMS Aerospace Applications. Montreal (2002)
23. Galuszka, A., Bereska, D., Simek, K., Skrzypczyk, K., Daniec, K.: Wykorzystanie Elementów Teorii Grafów w Systemie Analiz Kryminalnych. Przeglad Elektrotechniczny **86**, 278–283 (2010)
24. Jedrasiak, K., Bereska, D., Nawrat, A.: The prototype of gyro-stabilized UAV gimbal for day-night surveillance. In: Advanced Technologies for Intelligent Systems of National Border Security, Studies in Computational Intelligence, vol. 440, pp. 107–115 (2013)
25. Bereska, D., Daniec, K., Fras, S., Jedrasiak, K., Malinowski, M., Nawrat, A.: System for multi-axial mechanical stabilization of digital camera. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 117–189 (2013). ISBN: 978-3-319-00368-9
26. Bereska, D., Daniec, K., Jedrasiak, K., Nawrat, A.: Gyro-stabilized platform for multispectral image acquisition. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 115–121 (2013). ISBN: 978-3-319-00368-9
27. Magnetic-declination webpage, http://magnetic-declination.com
28. Kus, Z., Nawrat, A.: Object tracking for rapid camera movements in 3D space. In: Vision Based Systems for UAV Applications, Studies in Computational Intelligence, vol. 481, pp. 57–76 (2013). ISBN: 978-3-319-00368-9
29. Jedrasiak, K., Andrzejczak, M., Nawrat, A.: SETh: the method for long-term object tracking, computer vision and graphics. Lecture Notes in Computer Science, vol. 8671, pp. 302–315 (2014)