Automation of vision inspection in urban areas

Ph. D. dissertation

Agata Chmielewska

Supervisor: prof. dr hab. eng. Adam Dąbrowski

Auxiliary supervisor: dr eng. Tomasz Marciniak

Poznan University of Technology
Faculty of Computing
Chair of Control and System Engineering
Division of Signal Processing and Electronic Systems

Poznan, 2017

Contents

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Dr Hab. Eng. Adam Dąbrowski and my auxiliary supervisor Dr Eng. Tomasz Marciniak for the guidance that they have given me over the past few years.

I would also like to express my gratitude to the co-authors of some of my publications, including Paweł Pawłowski, Radosław Weychan, Marianna Parzych, Adam Konieczka, and Agnieszka Stankiewicz.

Finally, I would like to thank my family and friends for their support, inspiring conversations and for being actors in my video sequences, which I used for tests.

The research presented in this dissertation was supported by the "Scholarschip for PhD students specializing in majors strategic development in Wielkopolska, Poland", submeasure 8.2.2 "Human Capital Operational Program", co-financed by the European Union under the European Social Fund.

The research was also supported by the INDECT (Intelligent information system supporting observation, searching and detection for security of citizens in urban environment) project of the European Union realized in the 7th framework.

Abstract

City monitoring with CCTV (closed circuit television) technology constitutes an important part of the so-called smart city solutions and is constantly being developed. Despite a progressive advance in technology, most of video monitoring is still in analog standard and video frame resolutions are relatively small. Additionally, these systems require manual observation of screens by monitoring operators. Automation of video monitoring systems is, therefore, necessary. There are many algorithms, which support video sequence processing and event detection. These solutions support implementation of the so-called "intelligent monitoring", but they are still not adapted to specific solutions and thus, often ineffective. In this work, the automatic algorithms used in urban areas for various event detection are selected and supplemented.

This PhD thesis concerns the issue regarding the automation of vision inspection in urban areas from micro to macro biometrics. The modular system for intelligent analysis of video sequence and its high functionality will allow to adjust algorithms to the needs of security providers, for example, and allow for the automatic detection of specific (often dangerous) situations in the video monitoring sequence.

The work is divided into six main parts:

- introduction – automatic, hierarchical video monitoring system description
- description of standard methods of intelligent video analysis
- video acquisition, detection and classification of moving objects
- intelligent macro biometrics – moving objects density maps generation, people counting, dangerous situation detection
- micro biometrics – face and iris detection and recognition
- implementation of the selected algorithms using a digital signal processor.

This work includes a series of innovative propositions, which help to detect various events in urban areas. The methods are designed to assist monitoring operators in real time, draw their attention to important events and improve their work efficiency. Examples might include the following:

- detection of moving objects in the outdoor scene and classification of objects into types – thanks to this feature, cars and pedestrians can be automatically distinguished and classified;
- generation of density maps of moving people in areas with a high population density contribute to optimal and safe space management, for example, at a crowded bus stop or in narrow passages;
- accurate people counting, taking into account the direction of movement. In people counting studies, the effective method for estimating the number of people detected in a single BLOB (binary large object) was developed and tested. Moreover, this method has been improved using Bayes classifier.

The thesis also concerns a study of the precise recognition of persons based on face and iris. Standard people-authentication systems assume only frontal face images in constant full light in the recognition process. That is why it was decided to depart from that assumption. It was examined and concluded that face detection and recognition is possible even when the face is not frontally directed to the camera, when it is poorly or non-uniformly lighted or when the face area in the image has a small number of pixels (i.e. low-resolution).

The implementation of the selected algorithms on digital signal processor allows the operation of the system in a stand-alone mode without a PC unit. The author of the work draws attention to the reduction of time-consuming calculations without losing the necessary information. This allows for the operation of the developed models in real time.

Streszczenie

Monitoring miejski wraz ze stosowaną w nim technologią CCTV (ang. closed circuit television) są w ciągłym rozwoju. Jednak pomimo tego, większość systemów monitoringu nadal działa w standardzie analogowym i posiada małe rozdzielczości obrazów. Dodatkowo, systemy te wymagają manualnej obserwacji ekranów przez operatorów monitoringu. W związku z tym, niezbędna jest zatem automatyzacja tych systemów. Istnieje szereg algorytmów wspierających przetwarzanie sekwencji wideo i detekcji zdarzeń. Algorytmy te ułatwiają i wspierają implementację tzw. „inteligentnego monitoringu", lecz są często nieefektywne. Dlatego, w tej pracy doktorskiej, rozwijane są systemy monitoringu używane w przestrzeni miejskiej.

Badania zawarte w pracy doktorskiej dotyczą automatyzacji inspekcji wizyjnej w monitoringu obszarów zurbanizowanych z uwzględnieniem zagadnień od makro do mikro biometrii.

Modularny system do inteligentnej analizy sekwencji wizyjnych i jego wysoka funkcjonalność pozwoli na dopasowanie algorytmów dla potrzeb instytucji, na przykład instytucji zapewniających bezpieczeństwo, i wpłynie pozytywnie na automatyczne wykrycie zdarzeń, często niebezpiecznych.

Praca została podzielona na 6 głównych części:

- wstęp - opis automatycznego, hierarchicznego systemu monitoringu wideo
- opis standardowych metod przetwarzania sekwencji wideo
- akwizycja wideo, detekcja i klasyfikacja obiektów ruchomych w scenie
- inteligentna makro biometria – generacja map gęstości poruszania się osób, zliczanie osób, wykrywanie sytuacji niebezpiecznych
- mikro biometria – detekcja i rozpoznawanie twarzy i tęczówki oka
- implementacja wybranych algorytmów z użyciem procesora sygnałowego.

Praca zawiera szereg innowacyjnych podejść do rozpoznawania sceny m.in. do detekcji sytuacji zagrożeń bezpieczeństwa w obszarach miejskich. Opracowane metody mają na celu wspomaganie operatorów monitoringu w czasie rzeczywistym – mają zwrócić ich uwagę oraz poprawić wydajność pracy. Jako przykładowe modele można przytoczyć:

- wykrycie obiektów ruchomych w scenie i ich klasyfikacja na typy – dzięki czemu istnieje możliwość automatycznego rozróżnienia rodzajów obiektów ruchomych, np. odróżnienia samochodu od osoby
- generacja map gęstości występowania i poruszania się osób w przestrzeniach miejskich; wygenerowane mapy przyczynią się do bardziej efektywnego i bezpiecznego zarządzania przestrzenią; taka analiza zwiększy ochronę osób np. na zatłoczonym przystanku komunikacji miejskiej
- precyzyjne zliczanie osób z uwzględnieniem kierunku ich ruchu; w badaniach nad zliczeniem osób opracowano i przetestowano efektywną metodę szacowania liczby osób wykrywanych w pojedynczym obiekcie typu BLOB (ang. binary large object) – wspomniana metoda została ulepszona o zastosowanie klasyfikatora Bayes'a.

Praca doktorska dotyczy również badań nad precyzyjnym rozpoznawaniem osób na podstawie twarzy oraz tęczówki oka. Standardowe systemy weryfikacji osoby na podstawie twarzy i jej elementów wymagają stosowania wyłącznie frontalnych zdjęć twarzy w pełnym, stałym oświetleniu. Autorka pracy zdecydowała się odejść od tego założenia. Zbadała i stwierdziła, że wykrycie i rozpoznawanie twarzy jest możliwe nawet w przypadku, gdy twarz jest zwrócona pod kątem w stosunku do kamery, jest słabo lub nierównomiernie oświetlona lub obszar twarzy w badanej ramce wideo zajmuje małą liczbę pikseli (tzw. *low-resolution*).

Implementacja wybranych algorytmów na procesorze sygnałowym (DSP – ang. *digital signal processor*), co również przeanalizowano w pracy, pozwoli na działanie systemu w trybie autonomicznym oraz przyczyni się do zwiększenia jego funkcjonalności przy braku konieczności wykorzystania jednostek typu PC. Autorka pracy zwraca uwagę na redukcję czasochłonności obliczeń bez utraty niezbędnych informacji i działanie modułów w czasie rzeczywistym.

Abbreviations and notations

2D two-dimensional image
3D three-dimensional image
ACC accuracy
ALU arithmetic logic unit
AVI audio video interleave
$\boldsymbol{a}$ the vector of input attributes; $\boldsymbol{a} = [a_1, a_2, \dots, a_N]^{\mathrm{T}}$ in Bayes classifier
a input attribute (Bayes classifier)
α the angle of camera location
$B(x, y)$ value of the specified pixel in the image - blue component of the RGB color space
$B_y(x, y)$ value of the specified pixel in the image - B_y component of the IRgBy color space
BLOB binary large object
β angular span of the iris
C number of distributions which belong to the background (in GMM)
C_b color component of the YC_bC_r color space
C_r color component of the YC_bC_r color space
CCD charge coupled device
CCS Code Composer Studio
CCTV closed-circuit television
CMOS complementary metal-oxide semiconductor
CP counting of people
D assignment to the class (decision making attribute in Bayes classifier)
DET detection error trade-off
DMP density maps of people
DSD dangerous situation detection
DSP digital signal processor
DVR digital video recorder
d_{f} detection of face
E set of points representing the input (tested) image (face detection with the use of geometric models)
EER equal error rate
EVM evaluation module
F set of points representing the object - model of a face (face detection with the use of geometric models)
FAR false acceptance rate
FDR face detection and recognition
FIR finite impulse response
FN false negative value
FNR false negative rate
FP false positive value
FPR false positive rate
fps frame per seconds
FRR false rejection rate
η function of Gaussian distribution (in GMM)

$G(x,y)$	value of the specified pixel in the image - green component of the RGB color space
GMM	Gausian mixture model
$\mathbf{H}$	homography (projective transformation)
HMI	human-machine interface
HSI	hue saturation intensity (color palete)
h_{mod}	modified Hausdorff distance
ICA	independent component analysis
IDR	iris detection and recognition
IQR	interquartile range
IR	infrared
IRgBy	log-opponent color scale
IVA	intelligent video analysis
K	number of Gaussian distributions (in GMM)
L_{h}	diameter of the binary object in the horizontal direction
L_{v}	diameter of the binary object in the vertical direction
LBP	local binary pattern
LCD	liquid crystal display
LDA	linear discriminant analysis
$\boldsymbol{m}_i$	average value from $\boldsymbol{a}$ vector which belongs to the i-th class (in Bayes classifier)
M	point in face image
M_x	point designating horizontal line in the face image
M_y	point designating vertical line in the face image
MAC	multiply-accumulate
MOC	moving objects classification
MOD	moving object detection
MSE	mean squared error between two curves
$\mu_i(x,y,t)$	mean of the pixels for particular Gaussian distribution (in GMM)
N	number of independent input attributes $a_1, a_2, \ldots, a_N$ in the Bayes classifier
NMF	nonnegative matrix factorization
NTSC	national television system committee
n	number of persons in the single BLOB
OCR	optical character recognition
OCT	optical coherence tomography
OF	optical flow
OIS	optical image stabilizer
OSD	on screen display
P	probability
PAL	phase alternating line
PCA	principal component analysis
PETS	performance evaluation of tracking and surveillance
PPV	positive predictive value
p	transformation parameter (face detection method with the use of geometric models)
QCIF	quarter common intermediate format
R	radius of the iris
$R(x,y)$	value of the specified pixel in the image - red component of the RGB color space
$R_g(x,y)$	value of the specified pixel in the image – the R_g component of the IRgBy color space
RAM	random access memory

RAS	Sun Image raster bitmap (format)
RGB	red green blue (color space)
ROI	region of interest
RTC	real-time control
$\boldsymbol{S}_i$	covariance matrix
SAD	sum of absolute differences
STD	standard deviation
SVDMG	side view density map generation
SVM	support vector machine
SVTD	side view threat detection
T	threshold between background and foreground (GMM)
$T_p(F)$	transformed model of face (face detection method with the use of geometric models)
TN	true negative value
TP	true positive value
TPR	true positive rate
TVL	television lines
TVODCC	top view object detection, classification and counting
$W_i(x,y,t)$	weight of the pixels for particular Gaussian distribution (GMM)
VA	video acquisition
VGA	video graphics array
VLR	very low resolution
VR	visualization of results
YUV	color space, Y-brightness of the color (luminance), U and V determine the color (chroma)
$\mathbf{Y}$	image (luminance; intensity)
$Y(x,y,t)$	value of a pixel of the video sequence frame
$Y_{\mathrm{a}}(x,y,t)$	value of a pixel of the actual video sequence frame
$Y_{\mathrm{b}}(x,y)$	value of a pixel of the background (reference) frame
$\mathbf{Y}_{\mathrm{a}}(x,y,t)$	values of pixels in the particular block of the actual video sequence frame (in OF)
$\mathbf{Y}_{\mathrm{p}}(x,y,t)$	values of pixels in the shifted block of the previous video sequence frame (in OF)
$Y_{\mathrm{o}}(x,y,t)$	value of a pixel of the output (processed) video sequence frame
Z	specified number of classes $D_1, D_2, \dots, D_Z$ in the Bayes classifier

Chapter 1

Introduction

1.1. Intelligent monitoring of urban areas - problems

Nowadays, monitoring is widely used in urban areas. The term *monitoring* means regular qualitative and quantitative measurements or observations of phenomena carried out over a certain time. The monitoring is an observation of some activities or objects, usually the observation of people in order to protect them against threats [1, 2, 3].

Most people live in urban areas. Life in big cities is very convenient, but their inhabitants have many problems, and therefore cities increase their expenditures on security every year. One significant example is to ensure safety by the use of surveillance [4]. Monitoring enables the observation of:

- populations with high density, where there is concentration of a huge number of people in a relatively small area
- communication and high traffic load resulting from an increase in the number of cars within cities; this situation causes traffic jams and increases the risk of road accidents
- crime, pathology, and dangerous situations.

Monitoring may include:

- capturing information by the use of:
  – network monitoring, which monitors a network for slow or failing components and notifies the network administrator
  – website monitoring, which tests and verifies interaction between users and websites or web application
- observation at a distance with the use of
  – electronic devices, for example the CCTV (closed circuit television) cameras.

It is estimated that more than 100 million CCTV cameras are in use in the world today [5], therefore, video monitoring systems are essential for organizations, like the Police or Fire brigade, to ensure safety [6].

The first practical expansion of commercial surveillance systems in the world was in 1968, in New York City, for the purpose of fighting crime in the main business street [7]. During the 1980s there was a rapid increase in the number of CCTV systems, especially, in order to combat theft.

There are two types of observations of monitored areas: manual observation and automatic observation. In the first kind of monitoring, the screens are viewed by a person – the monitoring observer. In the second type of monitoring, the CCTV system automatically performs an analysis of the acquired video sequence [8, 9, 10]. The modern monitoring systems provide automatic event detection and analyze data from video sequences [11]. Applications of the IVA (intelligent video analysis) used in monitoring systems are, for example, moving object detection tracking and classification, density map generation, people counting, detection of cars driving in the wrong direction or even fire recognition. All these applications have led to the development of intelligent systems. Examples of IVA algorithms, which may support monitoring systems, are shown in Fig. 1.1.

The main problem of monitoring systems installed in urban areas is low efficiency of automatic video sequence analysis, especially, with false alarms during the detection of interesting events in the video sequence [3]. The other problems are related to bad classification of moving objects (for example into types: person, vehicle) or unacceptable results in bad weather conditions [12].

An example of the problem with intelligent video analysis is as follows: in 2011, the program, which automatically monitors areas of Mexico and United States of America had to be turned off despite a huge investment ($ 1 bilion) because it did not effectively classified moving objects like people or vehicles [12]. In such cases, the algorithm caused a large number of mistakes and unjustified intervention of border guards.

Fig. 1.1. Examples of the IVA algorithms [13]

1.2. Automatic, hierarchical video monitoring

The main purpose of this dissertation is observation of people in specific situations and areas. Thus, the research is divided into the following parts: preprocessing, micro- and macro-biometrics modules and implementation of selected algorithms on the embedded system. General scheme of extracted scientific problems of automated video analysis is shown in Fig. 1.2.

Preprocessing includes the detection and classification of moving objects from the video sequence.

The next part of the thesis deals with biometrics modules. Paper [14] proposes an extension of the concept by introducing two new terms "micro-biometrics" and "macro-biometrics". The "micro-biometrics" concept includes precise techniques of the analysis of human body structures, like the face, hand, fingerprints, iris or medical techniques like OCT (optical coherent tomography). These features are used for people detection and recognition tasks. The "macro-biometrics" concept, on the other hand, refers to the technique of using CCTV to monitor urban areas, with high, temporary or constant, population density. The areas with passing people are observed, such as parking lots, public buildings, schools, railway stations, offices, commercial centers, etc. It is important to analyze abnormal phenomena causing a potential security risk. Non-standard behavior of the crowd at sport and cultural events should be detected. In this case, it is important to do both: detecting the threats and selecting appropriate responses to them.

Scheme shown in Fig. 1.2 will often be repeated in this dissertation. The adequate block discussed in the individual chapter will be expanded with additional information. The blocks in "Biometrics system modules" in Fig. 1.2 are parallel. They can be performed independently of each other in different configurations. The hardware implementation and rapid prototyping are presented at the end of this thesis. In the subsequent drawings of this series, only an algorithmic part, without implementation, will be presented.

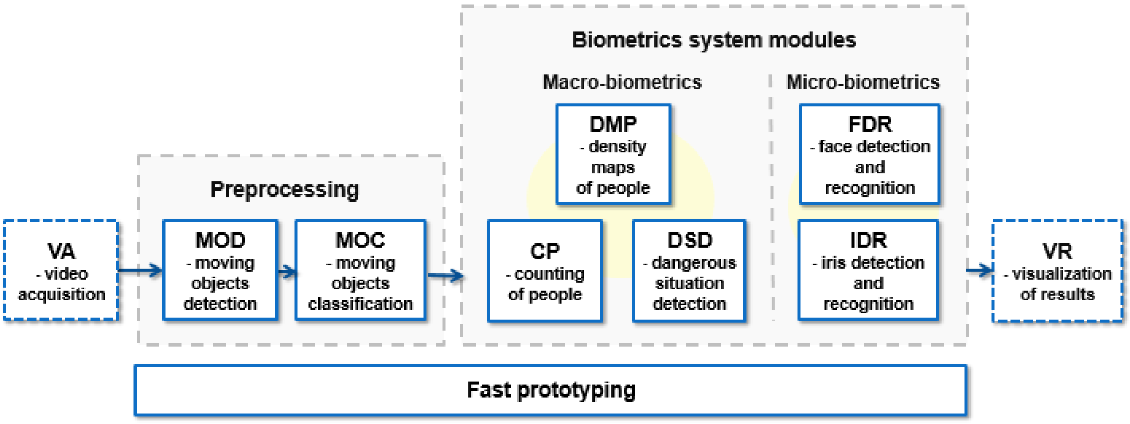


Fig. 1.2. Extracted scientific problems of automated video analysis

The first stage in the video sequence processing is observation and detection of moving objects in the video sequence. Sometimes, during the observation of the scene, detection alone is sufficient, for example, in the algorithms that are implemented to detect trespass on a forbidden zone.

The next stage is automatic scene observation, in which the moving objects may be subjected to the operation of classification. Moving objects can be classified into, for example, "person" and "vehicle" types. This classification can be used during the detection of dangerous situations, where it is important to know what kind of object (person or vehicle) is in the scene observed by the cameras in urban areas.

Then, after moving objects classification, people coordinates/position in the video frame can be processed and the density map of the peoples' movement can be generated. Since most of urban areas are equipped with video surveillance, it is natural to make use of CCTV for the generation of density maps instead of physical traffic counting. Moreover, density maps of people may be used for observing people in narrow and dangerous passages.

Apart from the estimation of the density maps of moving people, individual person can be also count. The accuracy of people counting is extremely important, especially when two or more persons are moving close to each other. The support of a people-counting system is required. Here, the additional classification of moving object (two or more persons) is needed - more precisely - classification in order to predict the number of persons in a single BLOB (binary large object) in the case, when people walk close to each other and create a single object in the binary image.

Every year, in urban areas, many road accidents occur and a lot of people lose their lives or are injured. The problem is a warning against the threats caused by situations like people running across the street or crossing the street on red light. The improved institutional efficiency supports, not only the operation of the Police, but also private individuals who want to protect themselves and their company. Algorithms for the detection of fire, vandalization of infrastructure or intrusion into prohibited areas should be used in CCTV systems in urban areas.

Algorithms dedicated to various situations and camera locations can be widely used to ensure safety. There are no universal algorithms able to, automatically, detect dangerous situations in every place and at every camera location. It is important to note that the selection of algorithms should strictly depend on camera location and the type of the scene observed.

The automatic recognition of dangerous situations [15, 16, 17] is also necessary in order to enhance the operators' concentration during video monitoring. These automatic systems for threat detection improve the concentration of the operators, who after 20 minutes of monotonous gaze at the monitor screens become tired and imperceptive. Intelligent video analysis, therefore, increases the operators' performance, through focusing their attention.

If the system identifies a threat, it should then also be able to recognize the person who caused it. This can be done using the face detection and recognition model. The faces of hooligans can be recognized even when the face is partially hidden with the use of a scarf or hat.

In the daily use of surveillance systems in urban areas, the face recognition model can be combined with dangerous event detection. For example, the detection of fire or vandalization of urban infrastructure can be performed automatically and adequate frames of video sequences are registered. From these frames, face can be detected and recognized.

In some cases, people can be recognized using the iris image from video surveillance. The resolution of such iris images is declared in the standards, which are described in Appendix A.

The last issue in intelligent surveillance is the possibility of algorithm evaluation in real-time in the embedded system. On the left hand side of Fig. 1.3, the video sequence is acquired from the camera . In the next step of the system, an intelligent video sequence analysis with the use of, for instance, DSP (digital signal processor), is done. Examples of algorithms, that can be implemented on the DSP in the embedded system are shown in Fig. 1.1. The output of the system will be the same as the video sequence with additional metadata. This metadata will focus the attention of the monitoring operator through the use of graphical information, for example, moving objects are surrounded with the red rectangular boxes (as in Fig. 1.3).

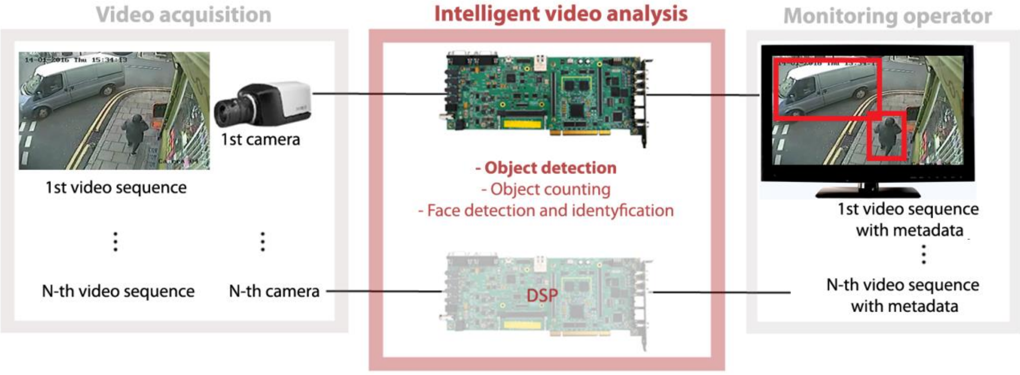


Fig.1.3. Schema of the intelligent video analysis system

The CCTV system's capabilities are targeted, mainly, toward remote monitoring of large and overcrowded areas such as airports, railway stations, industrial plants or other public areas. For the classification of events and people's behaviors in the recorded video sequences, the video systems should be assessed with the following issues [2]:

- facilities for acquisition of reliable and representative data without errors of relevant object detection and tracking,
- adaptation of algorithms in terms of diversity and complexity of exceptions for correct people detection,
- equipment with adequate models of situations without the need for large number of patterns.

1.3. Aim, scope, and scientific thesis

Intelligent video analysis is a challenging scientific problem with many promising applications. An image and video processing is a growing field of science and algorithms are intensively developed.

A motivation of this work is to create real-time models with moving object detection, moving object classification, people counting, face recognition, etc.

The author, during the work on the issues that are described in this dissertation participated in the 7th framework project of the European Union, i.e. INDECT - Intelligent information system supporting observation, searching and detection for security of citizens in urban environment (WP1: Intelligent Monitoring and Automatic Detection of Threats, WP7: Biometrics and Intelligent Methods for Extraction and Supplying Security Information) [18]. The author was also a co-author of a series of public reports related to the issue presented in this dissertation [19, 20, 21, 22, 23].

The role of the author of this dissertation in the INDECT project was quite significant. The author was developing, among others, system for automated people counting based on data from a CCTV system [24], detection of a threat at a bus stop, detection of a car driving in the wrong direction on a one-way street, detection of fire, detection and recognition of face and iris.

The research aim of the dissertation was to develop new as well as to improve existing methods of image processing and analysis of video sequences to detect specific events in urban areas.

Thus the aim of this dissertation is:

"*development of effective methods for automated video monitoring to analyze moving objects (people and vehicles) in urban areas*".

The goal of the work is also to elaborate the technical elements of the system for its rapid prototyping.

Based on the results obtained in this research, the following thesis can be formulated:

"developed and experimentally tested methods of intelligent video analysis improve effectiveness of monitoring in urban areas".

1.4. Organization of thesis

The content of the dissertation is as follows.

In Chapter 1, introduction to the subject of automatic analysis of video sequence obtained from CCTV systems is presented. The scientific problems, the aim, and the thesis are described.

Chapter 2 presents fundamental knowledge in the field described in the dissertation, divided into: preprocessing, micro- and macro-biometrics groups. Selected methods of IVA algorithms are shown, for example, object detection and classification, people counting, threat detection or people recognition.

Chapter 3 presents the video sequences acquisition, which are used for the tests, moving objects detection and classification. This Chapter presents important features of the prepared models and an appropriate choice of the control parameters, which are necessary for the programs to operate properly in outdoor scenes. Besides object detection process, two types of classification were shown. The first type, classified the detected moving objects into "person" and "vehicle" types. The second, classified moving objects in order to predict the number of persons in a single binary object, in situation where the persons are moving close to each other.

Chapter 4 shows the proposed algorithm for macro-biometric system modules, like density map generation of moving people, bi-directional people counting and detection of dangerous situations in urban areas with the use of IVA modules.

Chapter 5 describes people detection and recognition methods (based on face and iris image). Detection and recognition stages were tested under low lightning conditions, when the head is not directed towards the camera or when the head area in the video frame is small. Algorithm parameters were also chosen.

Chapter 6 presents algorithms, which are implemented on the embedded platforms. The DSP from Texas Instruments and smart camera from National Instruments were tested. Application of fast prototyping approach is presented.

Chapter 7 concludes the thesis. The resolved issues are summarized. These issues contribute to providing a comprehensive monitoring system.

At the end of the thesis, the Appendixes are presented. In the Appendixes, for example, norms and standard according to efficiently people detection and recognition in the video stream are also described.

The author of this dissertation, together with the cooperating persons, has repeatedly published scientific papers in journals. These papers concern the subject of the dissertation and is listed in Table 1.1. This published works may complement the content presented in this dissertation.

Table 1.1. Scientific papers published in the journals in the subject of the dissertation

Density maps generation	
[25]	A. Chmielewska, M. Parzych, T. Marciniak, A. Dabrowski, "New approach to traffic density estimation based on indoor and outdoor scenes from CCTV," **Foundations of Computing and Decision Sciences**, Vol. 40 No.2, pp. 119-132, 2015
People counting	
[26]	T. Marciniak, A. Chmielewska, A. Dabrowski, et. al., "People counting vision system based on ARM procesor programmed using Simulink environment," **Electronics – constructions, technologies, applications**, No. 6/2014, pp. 55-59, 2014
[27]	A. Chmielewska, T. Marciniak, A. Dabrowski, "Improved video-based people counting algorithm using BLOB classification," **Bulletin of the Polish Academy of Sciences** (to be published)
[28]	A. Chmielewska, T. Marciniak, A. Dabrowski, "Application of the naive Bayes classifier for bidirectional people counting," **International Journal of Applied Mathematics and Computer Sciences** (to be published)
[24]	A. Chmielewska, A. Dąbrowski, T. Marciniak, et. al., "The automatic counting of objects in the urban area (System automatycznego liczenia obiektów w ruchu miejskim)," **The theory and application of computer science**, Teoria i zastosowanie informatyki, Vol. 10, No. 2, pp. 83-93, 2012
Dangerous situation detection	
[15]	A. Chmielewska, P. Pawłowski, A. Dąbrowski, "Application of video sequence processing methods to support CCTV (Zastosowanie metod przetwarzania sekwencji wideo do wspomagania monitoringu miejskiego)," **The theory and application of computer science** (Teoria i zastosowanie informatyki), Vol. 9, No. 3, pp. 65-83, 2011
Face detection and recognition	
[29]	T. Marciniak, A. Chmielewska, R. Weychan, et. al., "Influence of low resolution of images on reliability of face detection and recognition," **Multimedia Tools and Applications**, Vol. 74, No. 12, pp. 4329-4349, 2013
[30]	T. Marciniak, A. Dąbrowski, A. Chmielewska, et. al., "Face Recognition from Low Resolution Images," **Communications in Computer and Information Science**, No. 287, pp. 220-229, 2012
Iris detection and recognition	
[31]	T. Marciniak, A. Dabrowski, A. Chmielewska, and A. Krzykowska, "Selection of parameters in iris recognition system," **Multimedia Tools and Applications**, Vol. 68, pp.193-208, 2014
[32]	T. Marciniak, P. Pawłowski, A. Dąbrowski, et. al., "Dobór elementów sprzętowo-programowalnych w systemie akwizycji obrazu tęczówki do celów identyfikacji osób (The selection of hardware and software elements of iris image acquisition system for people identification)," **Electrical Review**, No. 11b, pp. 18-22, 2012
[33]	T. Marciniak, A. Dabrowski, A. Chmielewska, et. al., "Analysis of Particular Iris Recognition Stages," **Communications in Computer and Information Science**, Vol. 149, pp. 198-206, 2011
Rapid prototyping	
[34]	A. Chmielewska, R. Weychan, T. Marciniak, et. al., "Fast Prototyping for Video Monitoring Systems with the Use of DSP Module," **International Journal of Electronics and Telecommunications**, Vol. 59, No. 4, pp. 375-381, 2013
[35]	A. Chmielewska, A. Dabrowski, A. Namerla, et. al., "Comparison of NI LabVIEW and NI Vision Builder AI environments in fast prototyping of video processing algorithms for CCTV using smart camera," **Electronics – constructions, technologies, applications**, No. 5, pp. 72–76, 2011

1.5. Algorithm effectiveness assessment

This Section presents the development tools used in the dissertation and the methods by which the algorithms effectiveness were evaluated.

Development tools used in the dissertation are:

- Matlab/Simulink 2013 [36] with Video and Image Processing library [37]
- Visual Studio 2015 with OpenCV library [38]
- Code Composer Studio [39] with C6EzFlo tools [40] for rapid prototyping
- National Instruments LabView [41]
- National Instruments Vision Builder Automatic Inspection [42].

The data distribution and the algorithm effectiveness can be presented with the use of algorithm efficiency statistics, for example with the use of true positive rate and visualized with the use of statistical plot or detection error tradeoff plot. These tools are described below.

Distributions of data were presented with the use of statistical plots. These statistical plots were created in the Matlab environment with the use of discrete data samples and boxplot function [43]. Figure 1.11 shows the interpretation of this plot. A typical box consists of the following parts. The "central box", which represents the central 50% of the data. The lower and upper boundary of this box adequately represent 1st (also called lower) and 3rd (also called upper) quartile. The central red line indicates the median (2nd quartile) of the data and cuts the data set into two halves. The ends of the whiskers represent the lowest datum still within 1.5 IQR (interquartile range) of the lower quartile and the highest datum still within 1.5 IQR of the upper quartile. The rest of the data not included between the whiskers are plotted as outliers.

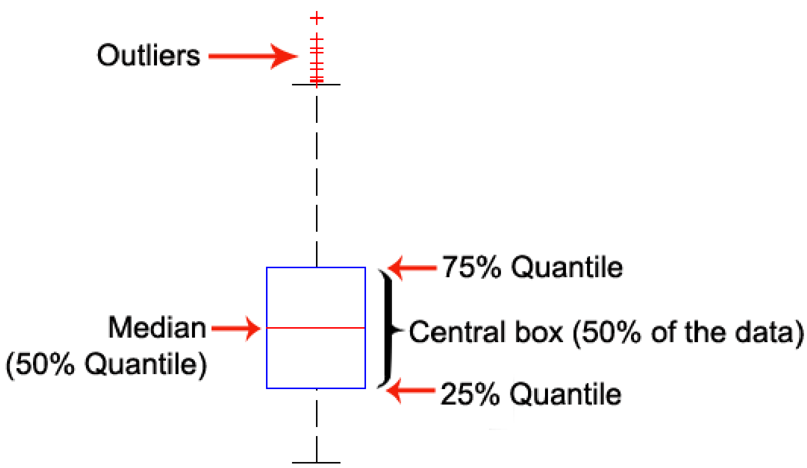


Fig. 1.11. Explanation of statistical plot markings [28]

The algorithm efficiency statistics can be calculated with the use of [44, 45]:

- TP – true positive – number of correctly identified examples from the specified class (equivalent with hit)
- FP – false positive – number of incorrectly identified examples from the specified class (equivalent with false alarm)

- TN – true negative – number of correctly rejected examples from the specified class (equivalent with correct rejection)
- FN – false negative – number of incorrectly rejected examples from the specified class (equivalent with miss).

During the experiments, the following parameters are calculated and analyzed: true positive rate or sensitivity (TPR , defined with equation 1.1), false positive rate (FPR, equation 1.2) positive predictive value or precision (PPV, given by equation 1.3), false negative rate (FNR, equation 1.4), and accuracy (ACC, equation 1.5).

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1.1}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{1.2}$$

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1.3}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} \tag{1.4}$$

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1.5}$$

In this dissertation, the EER (equal error rate) was also used. It is threshold value in which the miss probability and false alarm probability errors are equal. The EER can be calculated with the use of DET (detection error tradeoff) plots, which example is shown in Fig. 1.12.

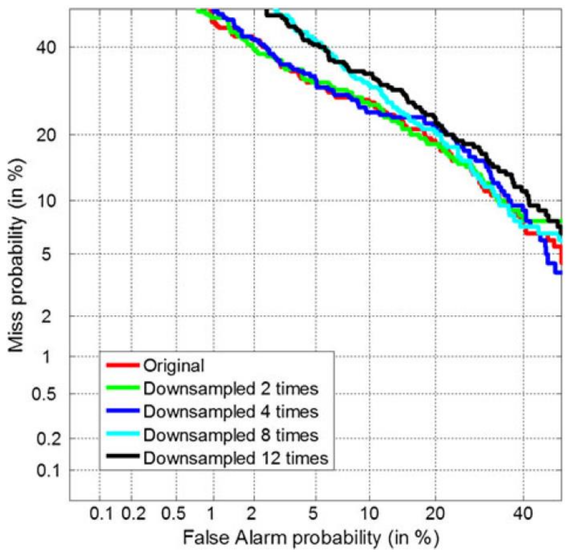


Fig. 1.12. Example of the DET plot of original signal and downsampled *n*-times [29]

Chapter 2

Standard methods of intelligent video analysis

Nowadays, an image and video sequence processing is an established field of science and algorithms are well developed. In the following Section, some algorithms used in the CCTV system has been described. The selected algorithms are, among others, an object detection and classification, a people counting, a density map generation or a people recognition.

2.1. Preprocessing of video sequences

2.1.1. Moving object detection techniques

Detecting moving objects, or motion detection, has very important significance in video processing. Motion detection is the initial stage of the moving object analysis in a video sequence.

Automatic detection of moving objects is the separation of objects occurring in the first plan of the scene from the background. There are different methods of moving objects detection [46]. In Figure 2.3 some methods are shown.

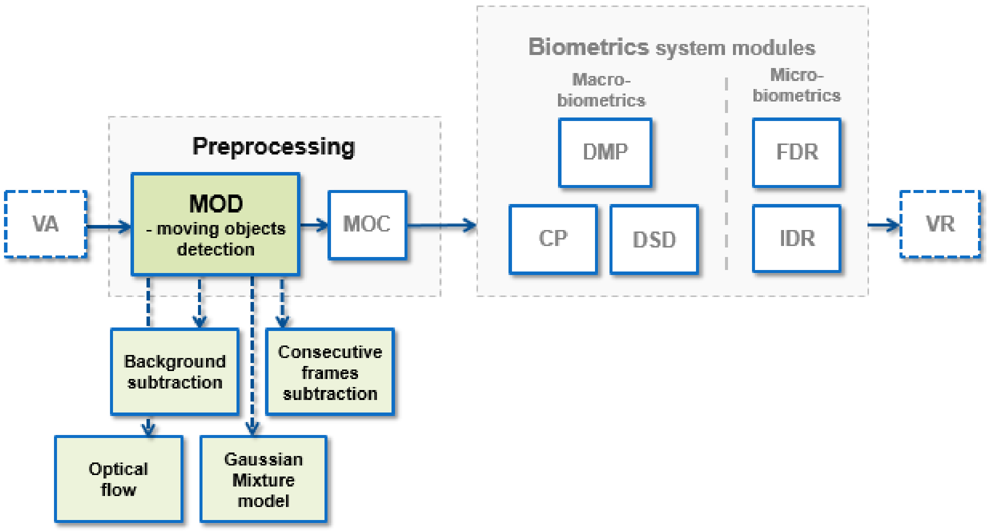


Fig. 2.1. Methods of moving objects detection

Background subtraction method

One of the methods for background modeling is background subtraction. This simple method is based on subtracting the current movie frame from the reference frame (the scene without moving objects) [47, 48]:

$$Y_{\mathrm{o}}(x, y, t) = Y_{\mathrm{a}}(x, y, t) - Y_{\mathrm{b}}(x, y), \tag{2.1}$$

where:

- $Y_{\mathrm{o}}(x, y, t)$ – value of the pixel of the output (processed) video sequence frame at (x, y) coordinates at t time,
- $Y_{\mathrm{a}}(x, y, t)$ – value of the pixel of the actual video sequence frame at (x, y) coordinates at t time,
- $Y_{\mathrm{b}}(x, y)$ - value of the pixel of the background (reference) frame at (x, y) coordinates.

Consecutive frames subtraction

One of the method of consecutive frames subtraction is the SAD (sum of absolute differences) [49]. The SAD value is calculated using the absolute value of the difference between the current and the previous image [50]. The acquired sequence is divided into a few areas (number of areas is selected depending on the image size), while the algorithm itself is used separately for each of them. The detected motion in the appropriate area is highlighted.

Optical Flow method

The detection using the OF (optical flow) algorithm is the next method that allows to detect objects in video sequences. The OF is a differential method. The OF is a motion detection method based on the extraction of block of pixels movement on the image through a comparison of the adequate blocks of consecutive frames [51] of the video sequence:

$$\mathbf{Y}_{\mathrm{a}}(x, y, t) = \mathbf{Y}_{\mathrm{p}}(x + \Delta x, y + \Delta y, t - 1). \tag{2.2}$$

where:

- $\mathbf{Y}_{\mathrm{a}}(x, y, t)$ – values of the pixels in a particular block of the actual video sequence frame at (x, y) coordinates at t time,
- $\mathbf{Y}_{\mathrm{p}}(x + \Delta x, y + \Delta y, t - 1)$ – values of the pixels in the shifted block of the previous video sequence frame at $(x + \Delta x, y + \Delta y)$ coordinates at $t - 1$ time.

Then, the correlation between them is found and a vector table called Optical Flow Field is created. These vectors define the shift of image regions, caused by the relative motion of objects and camera. The model constructs binary images based on motion vectors using the threshold. The binary image is processed by the morphological closing operation and BLOB (binary large object) analysis (used for connecting the corresponding pixels with adequate moving objects). Areas with a detected movement are marked with rectangles.

Background subtraction with the use of GMM

The GMM (Gaussian mixture model) describes the probability P of the pixel value $Y(x, y)$ observed at t time with the use of mixtures of Gaussian distribution [52, 53]:

$$P\big(Y(x, y, t)\big) = \sum_{i=1}^{K} W_i(x, y, t) \cdot \eta\big(\mu_i(x, y, t), \mathbf{S}_i(x, y, t)\big), \tag{2.3}$$

where:
- $P\big(Y(x,y,t)\big)$ – probability of pixels value at (x,y) coordinates at t time,
- K – number of Gaussian distributions,
- $W_i(x,y,t)$ – weight of the pixels at (x,y) coordinates at t time for particular Gaussian distribution,
- η – function of Gaussian distribution,
- $\mu_i(x,y,t)$ – mean of the pixels at (x,y) coordinates at t time for particular Gaussian distribution,
- $\mathbf{S}_i(x,y,t)$ – covariance matrix of the pixels at (x,y) coordinates at t time for particular Gaussian distribution.

In order to detect the moving objects, a discrimination of foreground and background distributions is required. For this purpose, Gaussian distributions are ordered and the first C distributions fulfilling following inequality are considered as a background:

$$C = \arg\min_{c}\left\{\sum_{i=1}^{c} W_i(x,y,t) > T\right\} \tag{2.4}$$

where,
- C – number of distributions which belong to the background,
- T – threshold between background and foreground.

2.1.2. Moving object classification

Automatic classification of moving objects in the video is an integral part of intelligent CCTV systems. Classification supplies a bridge between the low level feature extraction and the high level video interpretation.

There is a strong need to classify objects and analyze their activities correctly. Automatic decision making system, for example the classification of moving objects into types, helps the monitoring operator to program the system with specific events of interest, such as, raising alarm, when a vehicle goes in the wrong direction on a one-way street or when a people goes in the red light on a pedestrian crossing [15].

There are different types of classifications:
- classification in order to objects type recognition,
- number of persons in the crowd detection,
- crowd behavior classification,
- classification based on motion, where algorithm analyze temporally moving objects features, for example, movement trajectories (i.e., paths through space and time).

In this dissertation two types of classification are described: classification into types and estimation the number of object in the BLOB.

Methods of classification into types

Two methods for classification of objects into types were presented – classification with the use of the shape coefficient and with the use of the support vector machine (Fig. 2.2).

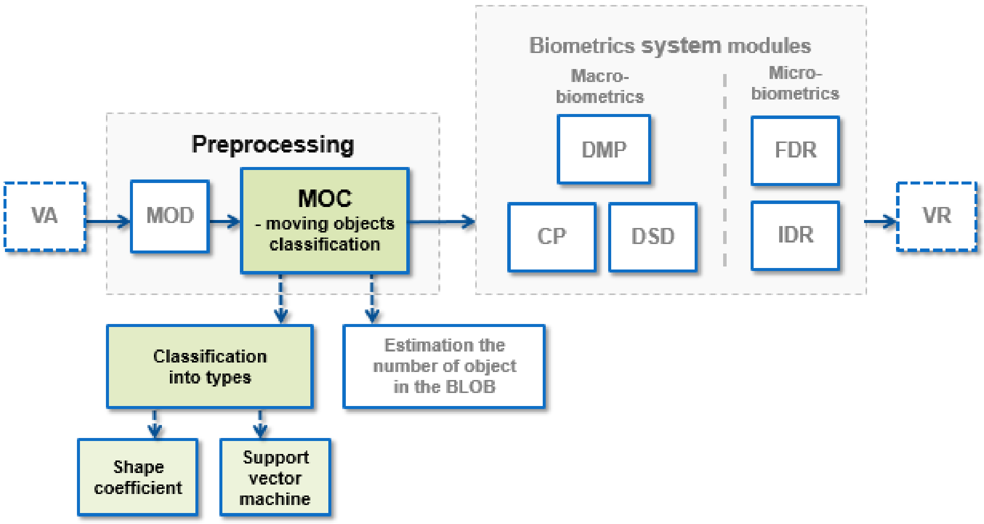
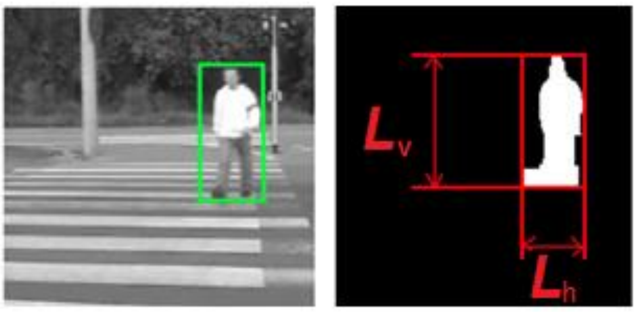


Fig. 2.2. Moving objects classification methods – classification into types

To perform a check of the object shape and orientation, the Feret's diameters [54] are most often used. The diameters describe the dimensions of binary objects (BLOBs) as the ratio of horizontal and vertical diameters - L_{h}, L_{v}, respectively; Fig. 2.3.

In the case where the detected moving object is in the vertical orientation, it is considered as a person. Otherwise, if the moving object is in the horizontal orientation it is considered as a vehicle.

The threshold of the Feret's coefficient should be selected taking into account the location and angle of view of the camera.



(a) (b)

Fig. 2.3. Visualization of Feret's diameters used in the classification of objects into types; (a) input image, (b) detected moving object in the binary image

The second method for classifying moving objects into types is SVM (support vector machine). This is the non-linear classifier, which creates supervised learning models for data analysis. For example, it allows to separate images belonging to two different classes with the maximum margin, thanks to the designation of the hyperplane [55]. A good separation of input data is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class.

The SVM classifies moving objects into types and creates a model that assigns new input attributes (images of objects) to the two categories, for example, "person" and "other". The example images of pedestrians from Inria Person database [56] used in object classification into types are shown in Fig. 2.4. The important point here is the assumption that the classifier training images must be standardized in size.

Fig. 2.4. Example of images from Inria Person database [56]

Methods of classification for estimating the number of people in a single BLOB

The main problem during moving objects detection was an inaccurate separation of objects which are close to each other. Thus, the methods for classification of objects, in order to predict the number of person in a single BLOB are described (Fig. 2.5). Two methods are presented: classification with the use of naïve Bayes classifier and with the use of histogram analysis in the specified ROI (region of interest).

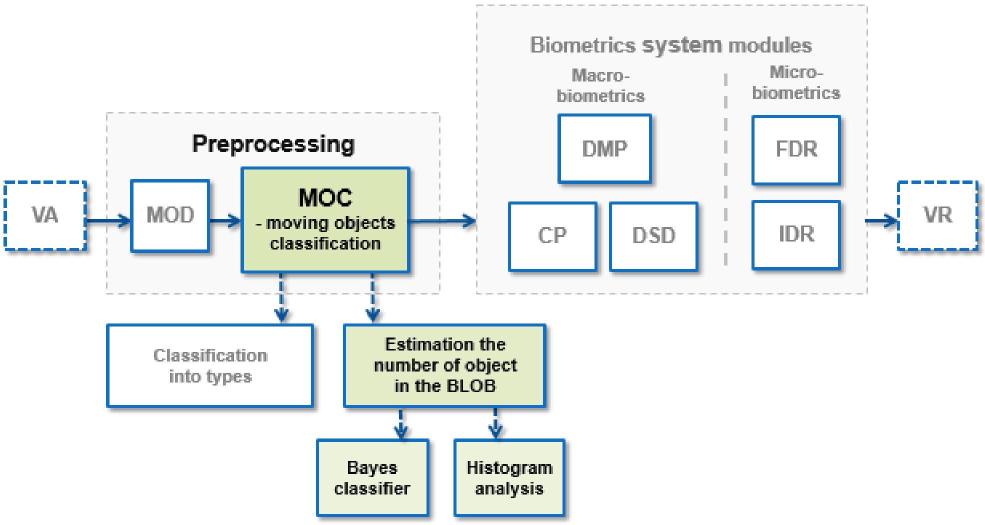


Fig. 2.5. Moving objects classification methods – estimation the number of object in the BLOB

The first method, i.e. classification with the use of Bayes classifier, is applied in order to perform a correct classification of detected moving objects (two or more persons which are close to each other) recognized as a single BLOB.

The Bayes classifier, based on the Bayes theorem, belongs to the probabilistic classifiers and is one of the machine-learning methods of solving the problem of classification [57]. The Bayes rule determines the assignment of specific input attributes to the appropriate class and is written in the form:

$$P(D/a) = \frac{P(a/D)P(D)}{P(a)} \tag{2.5}$$

where:

- D – specified class,
- a – input attribute of classifier,
- $P(D/a)$ – conditional probability - it is the probability of elements belonging to the class D when it has an input attribute a,

- $P(a/D)$ – conditional probability - it is the probability of occurrence of the input attribute a when the element belongs to D class,
- $P(a)$ – probability of occurrence of input attribute a,
- $P(D)$ – probability of that element belongs to the class D.

In this example, during the classification of the objects number in the single BLOB, the D variable means a class which assigns the number of people in the single BLOB. From the experimental studies carried out under real conditions, and described in Section 3.3.2, it indicates that the number of persons in the single BLOB is from 0 up to 3 person [58]. The a_i variable is a physical property of the binary object, for example horizontal and vertical dimensions of BLOB or BLOB area (number of white pixels of the BLOB) [58].

With the assumption of N independent input attributes $a_1, a_2, \dots, a_N$ and a specified number of classes $D_1, D_2, \dots, D_Z$, the Bayes rule (the probability of occurrence of the i-th class D_i) can be written as:

$$P(D_i/a_1, a_2, \dots, a_N) = \frac{P(a_1/D_i)P(a_2/D_i)\dots P(a_N/D_i)P(D_i)}{\sum_{k=1}^{Z} P(a_1/D_k)P(a_2/D_k)\dots P(a_N/D_k)P(D_k)} \tag{2.6}$$

After the computation of the probability of each class, the value with the biggest probability is selected for further computation.

The full Bayes classifier is a procedure relatively difficult to calculate, so in practice, a simplified version of this classifier, called naïve Bayes classifier, is used. From mathematical point of view, the naïve Bayes classifier is derived from the equation 2.5, and is presented as follows:

$$P(D_i/\boldsymbol{a}) = \frac{P(\boldsymbol{a}/D_i)P(D_i)}{\sum_{k=1}^{Z} P(\boldsymbol{a}/D_k)P(D_k)} \tag{2.7}$$

where: $\boldsymbol{a}$ – vector of input attributes $\boldsymbol{a} = [a_1, a_2, \dots, a_N]^{\mathrm{T}}$.

The denominator in the above equation is common to each class, so the decision can be taken on the basis of the numerator only:

$$P(D_i/\boldsymbol{a}) \approx P(\boldsymbol{a}/D_i)P(D_i). \tag{2.8}$$

The naïve Bayes classifier assumes that the input variables are independent so the multidimensional conditional probability is equal to the multiplication of one-dimensional probabilities by all a_i variables:

$$P(D_i/\boldsymbol{a}) \approx P(D_i)\prod_{j=1}^{N} P\left(a_j/D_i\right) \tag{2.9}$$

In the case of real variables (a_j), instead of determining the consecutive numeric values of $P\left(a_j/D_i\right)$, the multidimensional Gaussian function can be used to estimate $P(\boldsymbol{a}/D_i)$ value:

$$P(\boldsymbol{a}/D_i) \approx \frac{1}{(2\pi)^{N/2}|\boldsymbol{S}_i|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{a}-\boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{S}_i^{-1}(\boldsymbol{a}-\boldsymbol{m}_i)\right) \tag{2.10}$$

where:

- $\boldsymbol{S}_i$ – covariance matrix of input arguments for i-th class,
- $\boldsymbol{m}_i$ – average value from $\boldsymbol{a}$ vector which belongs to the i-th class.

The naïve Bayes classifier requires the knowledge of the exact values of the input arguments and classes assigned to them.

The second method for the estimation of the number of people in a the single BLOB is classification with the use of histogram analysis. The number of people in the BLOB recognition can be done on the basis of the histogram analysis using:

- a pattern that defines the people walking close to each other [59]
- computation of the local maxima with a specified threshold. When two objects are connected within the BLOB, the algorithm finds the local maximum of the binary image histogram within the specified range [60, 61]. Thanks to this operation, objects that are very close to each other can be distinguished.

2.2. Macro-biometric methods

2.2.1. Density map of moving object generation

The main purpose of people movement analysis and density map generation is to study the psychology of customer behavior in the area for increasing sales and finding "hot zones" in the salesroom; so-called BI – business intelligence. Generation of density maps is done to supporting a manual observation of physical traffic counting with an advanced video sequence analysis system.

The movement analysis can also be useful in order to optimize traffic, without using analytical tools, such as, questionnaire surveys or physical traffic counting by interviewers.

A density map is a graphical data representation, where each value contained in the matrix is represented by specific color - 2D (two-dimensional) histogram.

The motion density map from CCTV data is a 2D histogram expressing the most important regions of the objects' movement activity. Every movement is clearly visible in the image and thanks to that, density maps of moving objects are very useful.

A density map can be represented in various ways. The four sets of color images used for generating density maps are shown in Fig. 2.6. From left: (a) incandescent iron, (b) colors from navy blue through purple, pink, yellow to white, (c) colors from cold blue to warm red and (d) colors from maroon to light green. The third color scheme is commonly used in commercial systems, but in the author's opinion, the second color scheme is the most intuitive and it illustrates results in the best way possible. As a result, the second reference image scheme (Fig. 2.6. (b)) will be used for further research.

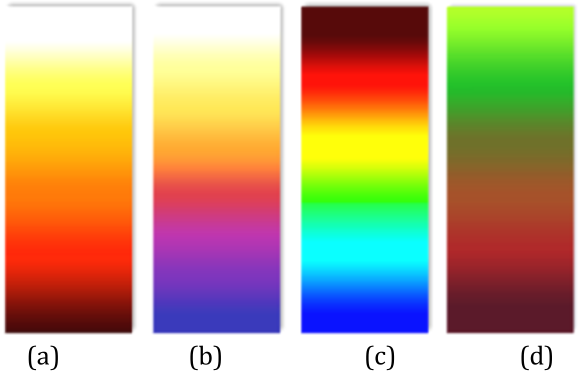


Fig. 2.6. Different colors of reference images for density map visualization [62]

A density map can be represented in a monochromatic or multicolored way.

In a monochromatic map, small values in the density map are symbolized by dark gray or black areas and large values are represented by lighter colors of gray or white. Sometimes, the path of moving objects in the scene is visualized as thin density distribution lines (one pixel wide per one video frame), defining the path along which the objects have been moving. These maps are not density maps and they are not shown with the use of 2D histogram, but are shown only in one color in the form of the paths that people follow [63].

In this dissertation only people movement are observed. During the generation of density maps [25], the observation of people in the given location is done. Only the lower part of the moving person is observed – this situation is shown in left-hand side of Fig. 2.7.

There are many of methods of density map generation. These methods can be divided into commercial and scientific. These methods are presented in Fig. 2.8.

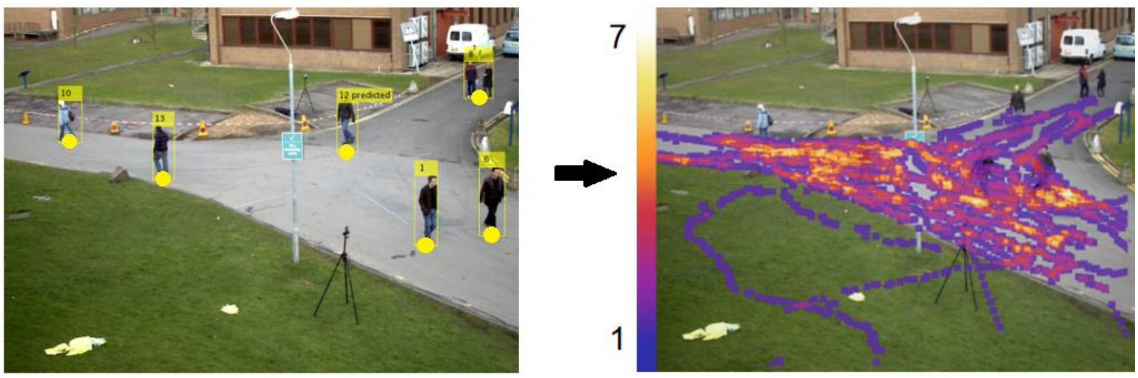


Fig. 2.7. Generated people density map based on calculation of a number of people in a specified location using PETS database

Commercial systems for density map generation

The generation of density maps, based on the data from video monitoring, is offered by several companies like Prism Skylabs [64], Mobotix [65], BOSCH [66], and Cognimatics [67].

The first software came from Prism Skylabs. The tool visualizes in-store customer behavior, then the density maps are generated and the results are also visualized using a chart with time line. This chart shows activity levels of motion in specified time periods.

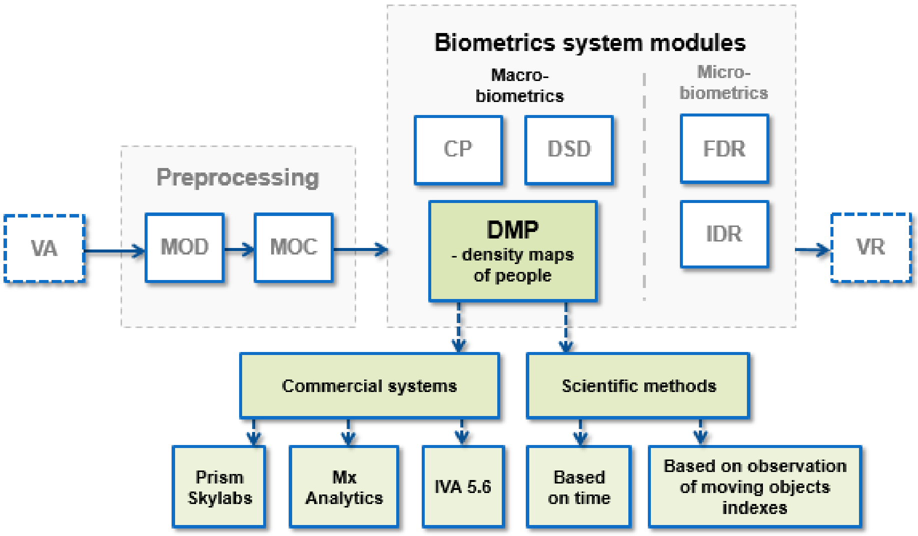


Fig. 2.8. Density map generation methods

The next tool is MxAnalytics from Mobotix company. This software collects statistical data of moving objects, for example, counts the people and generates density maps using contour lines to track moving objects. The camera records how often each contour line is passed over within a specified period. The most frequented areas are then highlighted in color on a density map.

The IVA 5.6 from Bosch does not generate density maps, but this software tracks moving objects and stores the paths of the objects' movements. The software also deals with crowd density estimation (by counting objects).

The disadvantages of commercial methods are as follows:

- the graphical results are not scaled to the highest result (for example from an hour or from a whole day)
- the software makes use of the only linear reference, so the colors on the density maps are, sometimes, wrongly mapped. In the author's opinion, the logarithmic map, instead of linear, is more useful for long recordings and with a large number of moving objects passing through the given area.

Scientific methods for density map generation

Generation of motion density maps applied to crowd monitoring on an escalator in order to detect abnormal situations has been published [68]. This approach is based on Optical Flow algorithm. Motion density maps are used for the definition of the region of interest of specific parts of the escalator. Thanks to this process, the calculation rate increases. M. Rodriquez [69] also uses information about moving-objects density to support the tracking algorithms in crowded scenes.

An alternative solution to moving objects, which are partially visible in the crowd, is a method proposed by [70]. This solution does not create paths of moving objects, but it generates density maps based on the extraction of local features, which enables the identification of objects in the scene.

The algorithm described in [71], based on 3D (three-dimensional) image from a time-of-flight camera allows for controlling the number of people and crowd density in real-time to assess safety in public spaces. The camera used in the system reduces the range of vision, but is ideal only for small spaces, tunnels or narrow passages.

There are two types of density maps – based on time and based on the observation of moving objects indexes.

In the case of density map based on time, every cell in the accumulation matrix is incremented every time (in each video frame) a moving object is detected in a given point [62, 72]. In the case of short video sequences, where the moving objects' distributions are small, this kind of density map is useful for the indirect determination of the speed of moving objects.

The second type of density map - based on the observation of moving objects' indexes - cells in the matrix are accumulated only once for each person in a given location [73]. This kind of density map shows the paths that moving objects follow – the stopping time and the velocity of moving objects do not matter.

2.2.2. Methods of standard solutions for people counting

There are different methods for people counting (Fig. 2.9). The methods used in scientific institutions and in commercial systems have been recapitulated. The methods are also divided into groups according to the position of the camera and type of data obtained from the camera (2D vs. 3D).

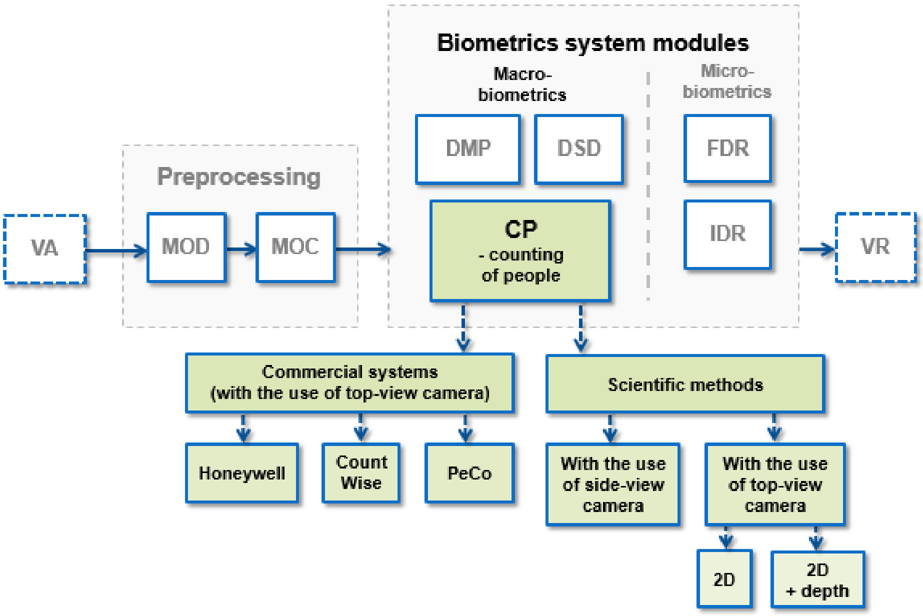


Fig. 2.9. People counting methods

Description of commercial systems for people-counting

Commercial systems for people-counting are becoming more and more popular every year. They can be, mainly, used to control the movements of people in public spaces (such as offices, buildings, shopping centers, pedestrian walkways etc.), providing a lot of useful information in real-time such as sale success, the most popular pedestrian pass (in the case of using several sensors), the rush hour or crowd density. Many companies like CountWise, Honeywell, VideoTurnstile, and VisualTools, have begun offering people counting algorithms ready to work together with CCTV systems [74].

The people-counting system from Honeywell is a part of Intelligent Video Analytics [75]. The system can operate independently or can be integrated with other technologies such as the standard CCTV system. Its output data supplies information about the number of people who came into and came out of a room. Additionally, the algorithms check the movement track of people within the camera's field of view. The Honeywell system can operate up to 40 channels while archiving data from all devices for up to 60 days. The appropriate operation of the system is dependent on video sequence resolution. It must be at least in the QCIF (quarter common intermediate format), i.e., a resolution of 176×144 pixels.

Another DVR (digital video recorder) with people-counting function is the PeCo system from Visual Tools [76]. The accuracy of counting is high, thanks to the analysis of the full image resolution, up to 25 fps (frames per second). The optimal camera placement is above the entrance to the room. The manufacturer of this system provides free software for collecting data from several PeCo recorders, as well as for posterior analysis, statistics and reports. The system compares detected objects with a reference object, and only objects that are at most 50% smaller or larger than the reference object are treated as one person. Thanks to this approach, even two people who are very close to each other will not be treated erroneously as one but as two people.

The next well-known approach for counting people and controlling their density is based on the application offered by CountWise [77]. This is a set of products designed for various specific tasks, which include: a system to precisely examine the number of people in a crowd (I-Count), an application that monitors the behavior of customers in stores and shopping centers (Z-Count), a queue management at cash registers in the stores with the use of the RTC (real-time control), i.e. Q-Count).

Another commercial system is the Video Turnstil, which is a tool for counting people [78]. This system offers the possibility of installing a counter that which displays the current number of people in the building. The counter can be connected to several CCTV cameras at all entrances to the building in order to continuously provide current data.

Description of scientific algorithms for people counting

In 2001, the authors in a paper [79] proposed a real-time algorithm (over 30 fps rate) for pedestrian tracking and counting from grayscale video sequences and images. The main processing task was the BLOB tracking analysis and the Kalman filtering – used for estimating the location of pedestrians. The system output is a set of temporal and spatial coordinates of each pedestrian. In 2007, another approach to the BLOB analysis was proposed [80]. The difference between the previous method [79] and the new approach [80] is that, the image is trained to predict the number of persons in the BLOB.

A bi-directional people counter for pedestrian-flow passing through a gate, based on area and color analysis, was proposed in 2006 [59]. Each pattern of a person can be recognized, thanks to the analysis of its HSV histogram. A result obtained with quantized histograms of intensity (or hue) is compared with the result of the preliminary counting. Thanks to the people-touching pattern analysis, the authors have solved the problem of objects that are very close to each other.

A bi-directional projection of the histogram was published in 2008 [81]. The authors used the grayscale histogram of the two-frame difference image. Thanks to this, the proposed method is effective for multiple binary object segmentation.

In 2008, the authors in an article [59] proposed another method for people counting using the flow analysis [82]. A frame is divided into several blocks, then each block is classified according to its motion vector; and if there is a similarity in movement, then the respective blocks are regarded as belonging to the same moving object.

In 2011, the authors in another paper [83] formulated and solved the problem of people tracking using a motion (or even uncalibrated) camera. A new method, namely, the tracking-by-detection model in a particle filtering framework, was used. This algorithm does not rely on the background modeling.

In 2012, A. B. Chan and N. Vasconcelos [84] worked on people counting in the inhomogeneous crowd. Low-level features were extracted from each segmented region and used for estimating the number of people in the segment. This algorithm, with the use of Bayesian regression, counts the overall number of people in each video sequence frame.

In 2013, the authors in a paper [71], calculated the density of people thanks to a top-view positioned ToF (Time-of-Flight) camera with depth information and people counting algorithms. It should be noted that ToF camera is very expensive. This solution is, thus, suitable for narrow passages.

For public events and analysis of a large number of people, the system presented in [85] was used. The Optical Flow method for moving-object detection was used in order to estimate the number of people entering a building.

Types of system for people counting due to the location of the camera

There are two types of systems for people counting depending on camera localization.

The first is when the camera is located above the scene and observes the scene vertically downward. This camera location allows for using two types of image data: a two-dimensional image data type and a three-dimensional image data type (2D with information about the depth of the scene).

When the camera is located above the scene and inside the building, the two-dimensional image data type is used [74]. In this situation, models for people counting are based on spatio-temporal coordinates of detected objects and thanks to that, the direction of the object's movement is taken into account. In such a system, the algorithm efficiency is strongly dependent on the methods used for moving objects detection (described in Section 3.2).

When the camera is located above the scene, inside the building and in narrow passages, the three-dimensional image data type is used. The operation of people counting in such system is done with the use of Time-of-Flight camera [71]. This camera is a range imagining camera system that calculates the distance from the camera to the object based on the speed of light. Most of these cameras work by modulating the outgoing beam and then, measuring the phase shift of that carrier on the receiving end. Thanks to that, even in a scene, which is strongly illuminated by natural light, the system properly detects the depth of the scene and objects in the first plan of the scene. However, the cost of this type of camera is still very high and still do not provide a high enough image-data resolution in comparison with the cameras in modern CCTV systems.

The second type of people counting system is defined by the camera pointed at the scene at an angle. This camera position is usually used in large, crowded outdoor scenes. Due to the hiding of objects one behind another, the number of objects in the scene must be estimated [84].

Generally, the above methods need to be improved upon in order to correct the efficiency of counting, especially for objects moving close to each other. This problem can be solved by determining the number of person in the single BLOB and is described in the next part of the thesis.

2.2.3. Analysis of methods for threat detection

There are no universal algorithms, which automatically detect dangerous situations in every place and at every camera location (Fig. 2.10). It is important that the selection of algorithms should strictly depend on camera location and the type of the scene to be observed.

Depending on what the camera observes, there are different events. As an example, the algorithms that can be used at the airport or railway station will be described. These include detection for weapons, abandoned luggage and fire.

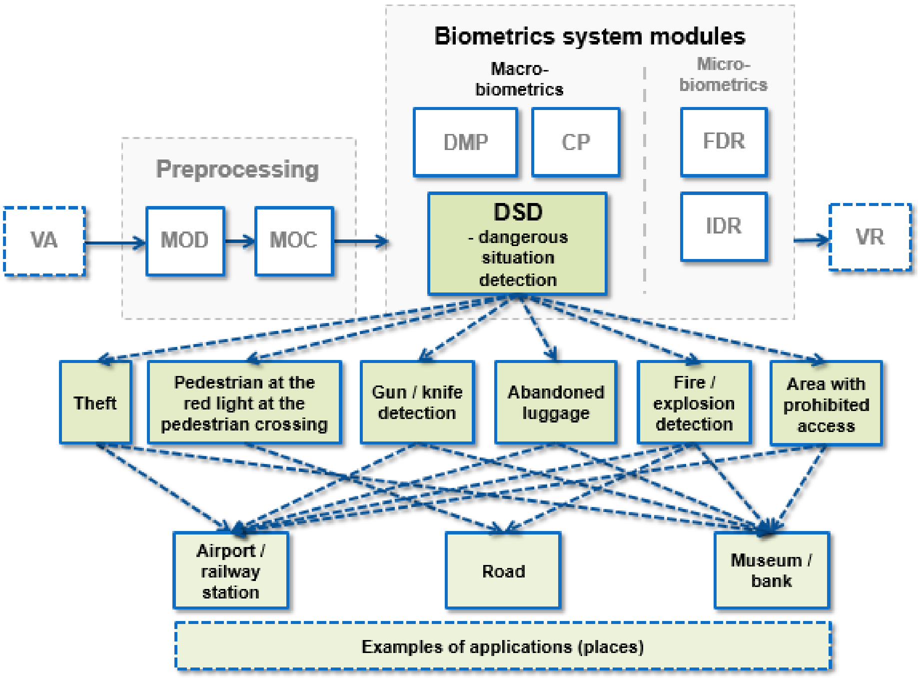


Fig. 2.10. Places where algorithms for detecting dangerous situations can be used

Automatic detection and recognition of a gun (or a knife) is based on the analysis of the results of pattern recognition [86]. In the first step, the moving objects are detected, and then the objects are classified into two types: offender and victim. Then, the algorithm detects a weapon held by an offender or a weapon, which is close to the victim's corpse. Unfortunately, the model is adapted only to operate inside a building, due to the chosen method of moving object detection. This model is based on the current frame and background subtraction, which is not resistant to the light changes.

Thanks to the observation of regions in the video frames, which are stable over time, an abandoned luggage can be detected [87]. At the same time, moving objects (owners of the luggage) located in the baggage area are also verified. The difficult with this algorithm is the selection of the time threshold after which the luggage is considered to be abandoned.

The third algorithm which can be used at airports is an algorithm for fire detection [88]. This model uses the HSV color palette in order to combine the information about the colors of fire. The model also utilizes the edge detection algorithm to detect the edges of the flames. This algorithm can be connected with the detection of a gunfire sound or bomb explosion [89]. However, there is a problem with the automatic detection of fire (based on muzzle flash) in a strongly illuminated scene.

A general issue with algorithms for dangerous-situation detection is the problem of false alarms, which reduce the effectiveness of the algorithms.

2.3. Micro-biometric methods

Micro-biometrics refers to metrics related to distinctive, measurable human characteristics, like face, iris or retina recognition, fingerprints, hand geometry, ear features, typing rhythm, signature recognition, gait and voice characteristics. Types of micro-biometric systems along with their market share by technology [90] are shown in Fig. 2.11.

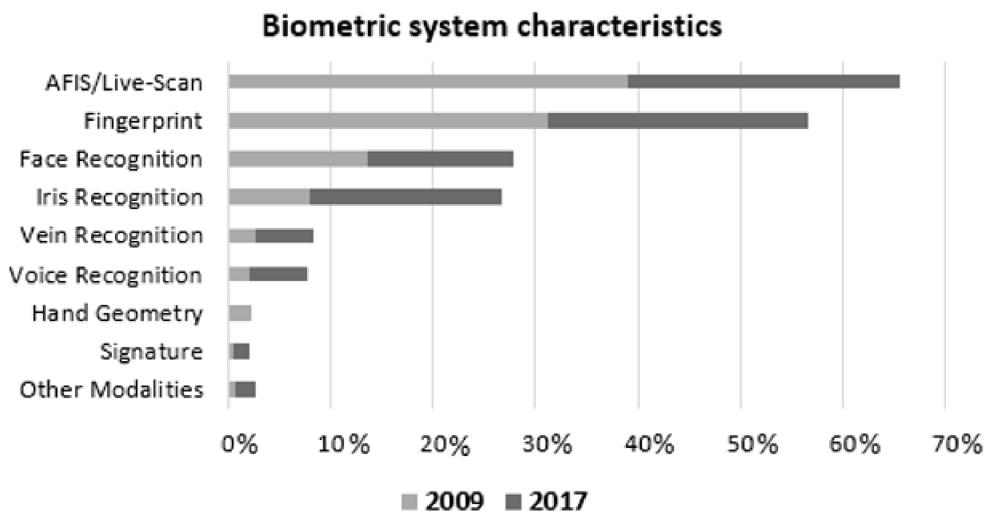


Fig. 2.11. Micro-biometric system modalities and market share by technology in 2009 and 2017

Micro-biometric in computer science is used for identification and access control. For example, biometric systems are used by airport security (with the use of iris characteristics), for checking access to buildings (using fingerprints or facial characteristics) or access to cars (using voice or fingerprint recognition). Biometric systems are also widely used in the private consumer sector, for example, in smartphones or in electronic devices like computers - instead of entering a security code, users prefer to use fingerprints for authentication.

During a fingerprint registration process, the person's unique characteristic is captured by a sensor and stored in a database. Later, when verification of the person is required, a new record is captured and compared with the database. If the person's characteristic matches that in the database record, then the person's identity is confirmed.

The accuracy of the face recognition systems is medium (due to various light conditions, the person's age and other factors), but cost less, because image device required is just a camera. The social acceptability of face recognition systems is high.

The accuracy of iris recognition systems accuracy is very high (iris features are stable over a lifetime), but the social acceptability is low, probably because the system is intrusive and some users exhibit a certain degree of discomfort with eye-based technology.

Generally, micro-biometric systems improve security and customer experience. The disadvantages of these systems is as follows: the environment and usage can affect measurements; the systems are not 100% accurate and require integration with hardware.

2.3.1. Face detection and recognition methods

Face detection

During a face detection process, the position and scale of the face (or faces) is estimated. Consequently, these values are used as input data in the identification/verification algorithm. The face detection subsystem should correctly detect objects similar to faces, like a hand (in the case of skin color algorithm) or a ball (in the case of geometric modeling) and reject them. The location of a human face in the image is one of the most important steps in the process of face recognition. This stage reduces the calculation time by reducing the analyzed area in the identification/verification stage.

Face detection is, typically, implemented in three phases. The detection means finding the location of the face in the image or the designation of its position. The first step is the reduction of distortion with the use of image processing techniques such as noise reduction and the equalization of the histogram. The next stage consists of finding areas where faces are most likely to appear. The final step verifies the previously selected areas and the face is detected and marked [91].

There are a lot of programs for face detection [92, 93, 94, 95]. An example of a web application is "The Face Annotation Interface" [96].

In this Section, three public software packages for face detection are described (Fig. 2.12). The first is based on skin detection algorithm [92]. The second makes use of geometric models [93] exploiting the Hausdorff distance [94]. The third algorithm uses Haar-Like classification [97, 95].

Face detection using skin color filter

With the use of color images, a human face can be detected using skin color filter. Skin color is the predominant factor, which is considered in the face detection process. The human skin is distinguished, rather, by its intensity than the color – thanks to which the skin color differences can be effectively removed.

In this part of the research, software for face detection, using the skin color filter, was used [92]. The input image (in RGB color palette format) is converted to the log-opponent IRgBy (YRgBy)color palette:

$$Y(x,y) = (105 \cdot \log(R(x,y)+1) + 105 \cdot \log(B(x,y)+1)105 \cdot \log(G(x,y)+1))/3 \tag{2.11}$$

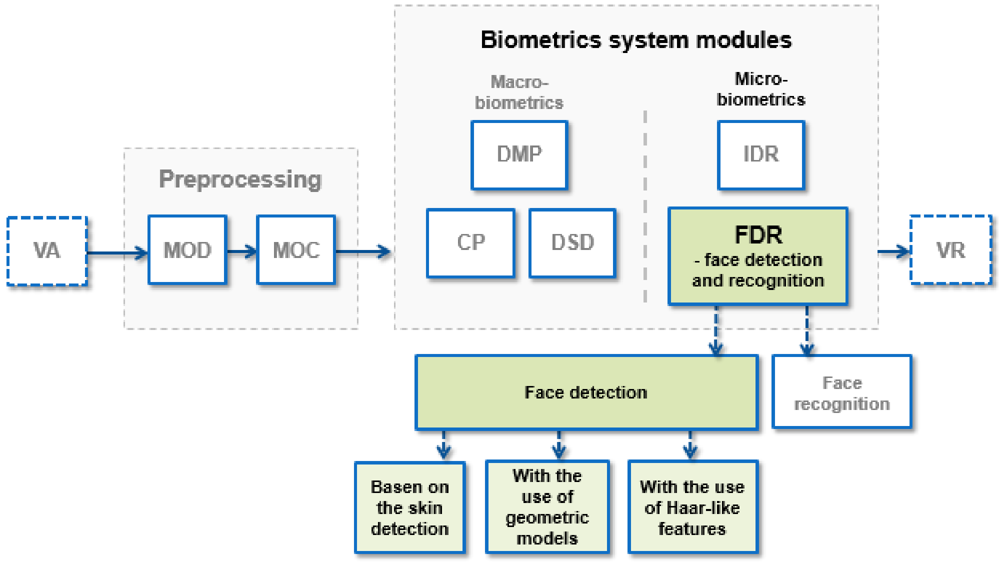


Fig. 2.12. Face detection methods

$$R_g(x,y) = 105 \cdot \log(R(x,y)+1) - 105 \cdot \log(G(x,y)+1) \tag{2.12}$$

$$B_y(x,y) = 105 \cdot \log(B(x,y)+1) - (105 \cdot \log(G(x,y)+1) + 105 \cdot \log(R(x,y)+1))/2 \tag{2.13}$$

Then, the Rg and By matrices are filtered by the median filter. The hue (2.14) and saturation (2.16) values are used to select those areas where the color matches the color of the skin – the result is a binary skin map (Fig. 2.13).

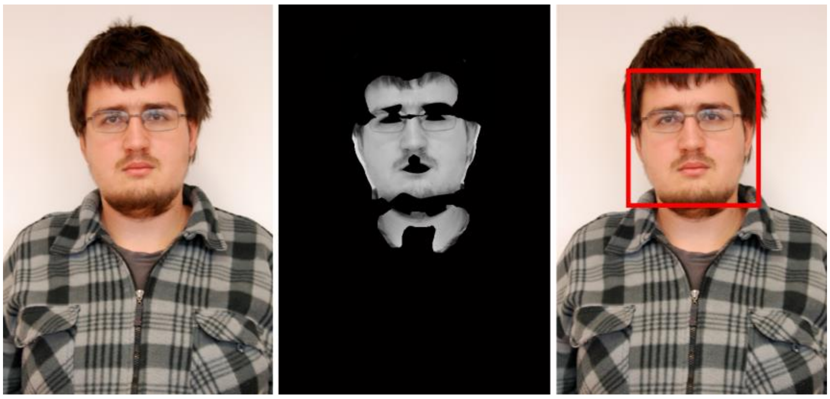
$$\text{hue}(x,y) = \text{atan2}(R_g(x,y), B_y(x,y)) \tag{2.14}$$

where:

$$\text{atan2}\left(R_g(x,y), B_y(x,y)\right) = \begin{cases} \arctan\left(R_g(x,y)/B_y(x,y)\right), \text{when } R_g(x,y) > 0, \\ \arctan\left(R_g(x,y)/B_y(x,y)\right) + \pi, \text{ when } R_g(x,y) < 0 \text{ and } B_y(x,y) \geq 0 \\ \arctan\left(R_g(x,y)/B_y(x,y)\right) - \pi, \text{ when } R_g(x,y) < 0 \text{ and } B_y(x,y) < 0 \\ \pi/2, \text{ when } R_g(x,y) = 0 \text{ and } B_y(x,y) \neq 0 \\ -\pi/2, \text{ when } R_g(x,y) = 0 \text{ and } B_y(x,y) \neq 0 \end{cases} \tag{2.15}$$

$$\text{saturation}(x,y) = \sqrt{(R_g(x,y))^2 + (B_y(x,y))^2} \tag{2.16}$$

The binary skin map (Fig. 2.13) and the original image are used to detect faces in the picture. This technique is based on the selection of designated areas that have holes such as eyebrows, eyes, mouth or nose. Theoretically, all the regions where skin is detected without holes are not faces. All these operations, such as finding holes in binary images, are performed using morphological operations.



(a) (b) (c)

Fig. 2.13. Original image (a), skin map multiplied with original image in the grayscale (b), image with detected face (c)

Face detection with the use of geometric models

During the processing of gray-scale images, skin detection algorithm cannot be used in the detection process. Therefore, the use of geometric models can provide a solution to the mentioned issue. This method is based upon the knowledge of the geometry of a typical human face and its relations, e.g., assumes its natural symmetry. The appropriate algorithm finds a rule to describe the shape, size, and other face characteristic points such as the eyes, nose, or chin. Additionally, the relations between them (positions and distances) are also important.

For the sake of testing, face detection using geometric models algorithm was used [93]. As previously mentioned, the algorithm is based on gray-scale images and in this software uses the modified Hausdorff distance [94, 98], which is adapted to the needs of the detection stage:

$$h_{\mathrm{mod}}(E,F) = \frac{1}{|E|}\sum_{e\in E}\min_{f\in F}\|e-f\| \tag{2.17}$$

where E and F are sets of points ($\{e_1, e_2, \dots, e_n\}$ and $\{f_1, f_2, \dots, f_n\}$) representing the input (tested) image and the object (model of face) - each point in the image is an image feature such as the edge point:

$$E = \{e_1, e_2, \dots, e_n\}, \quad F = \{f_1, f_2, \dots, f_n\} \tag{2.18}$$

The purpose is to find the transformation parameter p, where the Hausdorf distance between the transformed model $T_p(F)$ and E is minimized.

The problem of face detection can be formulated as:

$$d_{\mathrm{f}} = \min_{p\in P} h_{\mathrm{mod}}\left(E, T_p(F)\right). \tag{2.19}$$

Face detection with the use of Haar features

The third technique for face detection is the method which uses the Haar features [99] extended with edge, line and central features [100] (see Fig. 2.14). This method is based on the object detector proposed by Paul Viola and Michael J. Jones [101] and then improved by Rainer Lienhart and Jochen Maydt [95]. The programs input and output are the image and parameters, respectively, with positioning of the localized area—the face. This program uses four classifiers: the detection of the face (which is seen from the front), both eyes, as well as the left and right eye separately. Thanks to histogram equalization, this algorithm is able to deal with variable lighting and is also effective in the case of side illumination.

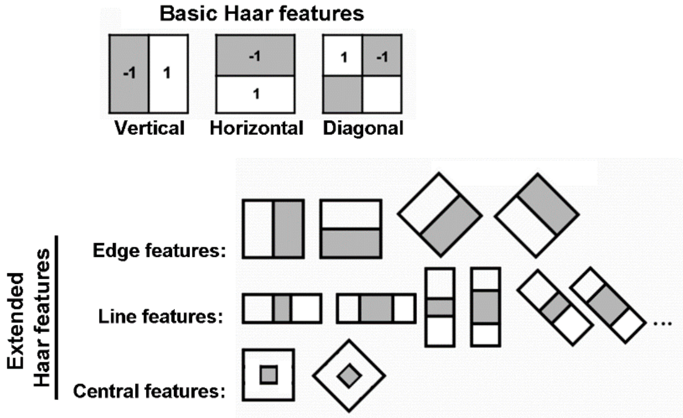


Fig. 2.14. Basic and extended Haar features

Face recognition

The methods used for the face recognition are presented. There is an abundance of methods for face recognition from an image [102] (Fig. 2.15). The identification of the face is typically solved using the following techniques: Eigenfaces based on PCA (principal component analysis [103, 104, 105]), Fisherfaces based on LDA (linear discriminant analysis [106]), ICA (independent component analysis [103]) and other approaches such as e.g., NMF (nonnegative matrix factorization [102]) or LBP (local binary pattern [107]). A comparative study of Eigenfaces vs. Fisherfaces vs. ICA faces can be found in [108].

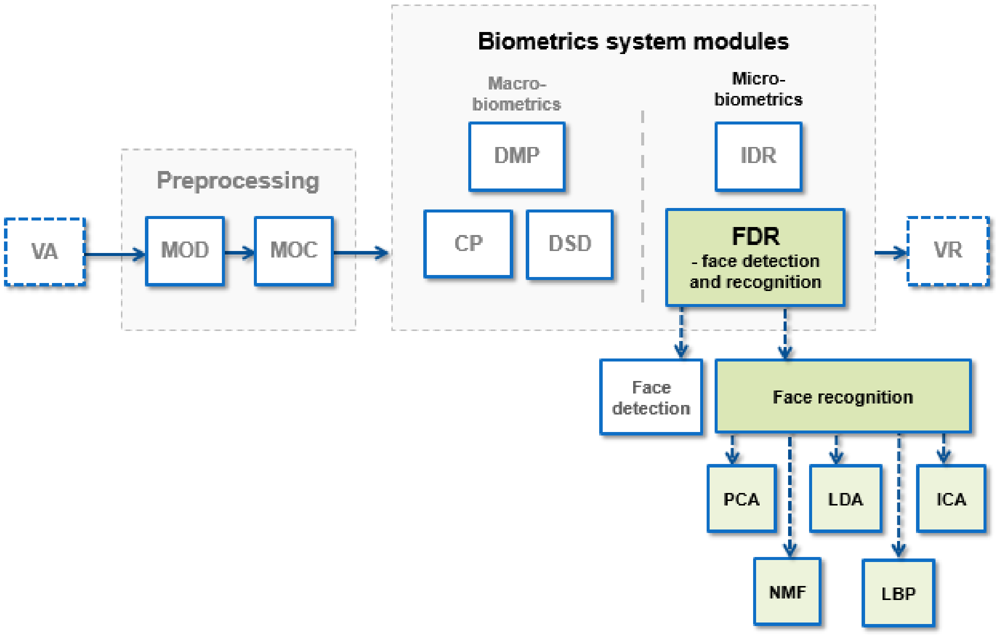


Fig. 2.15. Face recognition methods

The local binary pattern method is based on the classification of image textures. The model compares particular pixels of the image with neighboring pixels and creates images of face features. Then, a histogram is determined and based on the histogram, feature vectors are created. The feature vectors are processed by a machine learning algorithm.

The second method for face recognition is principal component analysis, which uses eigenvectors called eigenfaces (when PCA is used to analyze the face image) [109]. The method reduces the feature space by extracting and coding the information about the variance in a set of images. It uses the 2^{nd} order statistics. The face recognition is based on the distance from the nearest class, according to the numbering assigned to individual photographs at the beginning (indicating a person in the class).

The third method, independent component analysis, uses statistics of higher order. The method designates independent variables from the input images, so that important data on the faces generated by the method do not overlap - but are located in different places. Face recognition is also based on the distance from the nearest class.

The last described method is linear discriminant analysis. This method, in combination with the linear Fisher's discriminator, creates a linear combinations of face features in the image (so-called Fisher faces) and ensures class separation.

The important thing is that the face in the image during the recognition step must be in the same size and position. Therefore, the previous step described in this Section i.e. precise face detection, is highly important.

2.3.2. Iris detection and recognition

An iris is a part of the human eye, which has developed in the early phase of human life and remains unchanged for a considerable part of life [32, 31]. The construction of the iris is independent of genetic relationships of each person in the world. Even twins possess different irises.

Commercial systems

Currently, several iris acquisition devices, as well as whole iris recognition systems, can be found on the market (Fig. 2.16).

However, the technology for iris recognition is currently, not yet in widespread use, hence, the prices of these devices are high and access to purchase them is limited. As examples of the iris devices the following products can serve:

- IrisGuard IG-H100 – a manual camera for iris acquisition and verification; this camera can be held in the hand or can be mounted on a tripod or on a desk. 8 pictures of the eye can be registered in less than 3 s. The system is equipped with RS170 connector composite video – NTSC (national television system committee), and the pictures are monochrome with resolution 470 TVL (television lines)
- IrisGuard IG-AD100®multi-modal capture device, which makes it possible to combine face and iris recognition; dual iris capture takes below 4 seconds (normal conditions). A person's eye image can be acquired anywhere within a range from 21 cm to 37 cm away from the unit and be perfectly authenticated within the range [110]

- OKI IrisPass-M – this camera is designed for mounting on the wall to control access to secured areas or can be used at a border control, or to register travelers [111]; it takes a picture of the iris in a time of 1 s or less (depending on lighting conditions), identifies the iris in the time of 1 s or less (depending on the configuration of the PC as a controller and the size of iris database); false acceptance rate (FAR): 1/(1.2 M); shooting from a distance of 30–60 cm
- Panasonic BM-ET330 – designed for control of access to buildings and mounted on the wall [112]; iris recognition time is about 1 s, acquisition possible from a distance of 30–40 cm. The camera field of view: 115 degrees horizontal, 85 degrees vertical; the maximum number of irises in the database is 1000 and FAR is 1/(1.2 M).

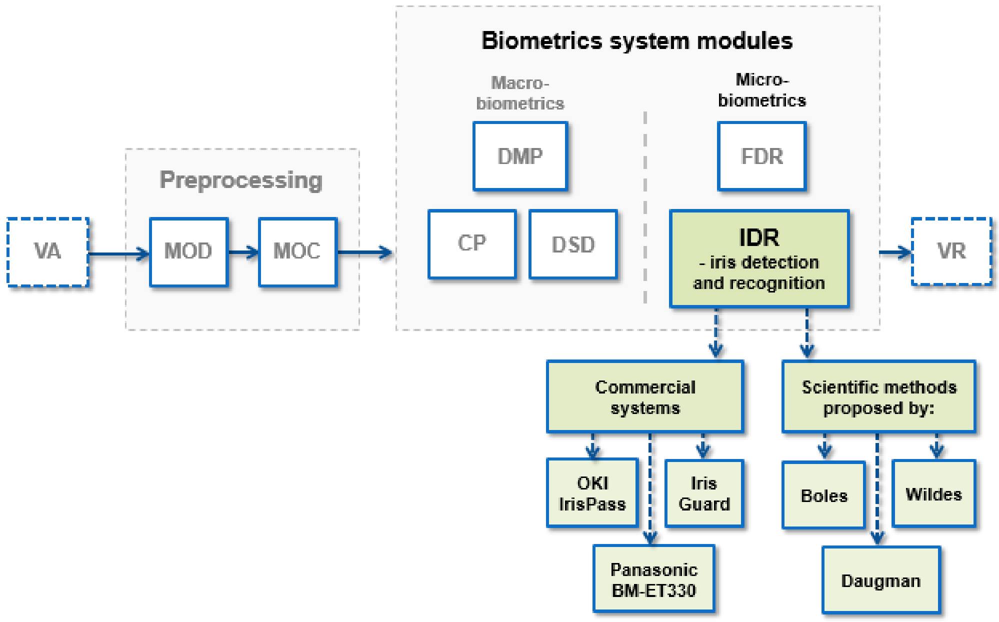


Fig. 2.16. Methods for iris recognition

Available iris acquisition devices, as well as whole iris recognitions systems, are compared in the Table 2.1.

There are systems, which enable the detection of the iris of persons who are in motion too. In 2009, the company Sarnoff [113] presented the first device in the series Iris-On-the-Move, which realizes this assumption. The IOM Passport Portal System allows the detection and identification of thirty people per minute. The system can be used, effectively, to secure objects with a large flow of people, such as, embassies, airports, or factories. The system uses a card reader, which is a preliminary step in a persons identification. The person is detected by a system of cameras, the iris is, then, detected and the iris code is determined, which is compared with the pattern. An advantage of this system is its ability to identify people who wear glasses or contact lenses. The system allows for the identification of people from a distance of three meters.

Methods for iris detection and recognition

Identification techniques based on iris analysis (Fig. 2.16) [114] has gained popularity and scientific interest since John Daugman introduced it in 1993 - the first algorithm for the identification of persons based on the iris of the eye [115].

Table 2.1. Comparison of the iris devices [31]

Iris database name	**IrisGuard IG H100**	**IrisGuard IG AD100**	**OKI IrisPass-M**	**Panasonic BM ET330**	**LG IrisAccess 4000** [116]
Description	Light manual camera for iris acquiring and verification	Multi-modal capture device, makes it possible to combine face and iris recognition	Designed for mounting on the wall to control access to secured areas	Designed for access control to buildings, mounted on the wall	Identifcation is possible by using the left, right or both eyes
Speed of acquisition	8 images registered in less than 3 seconds	Dual iris capture takes below 4 seconds (normal conditions)	Take a picture of the iris atthe time of1 s or less (depending on lighting conditions)	Iris recognition time of about 1 s	Not specified
Distance of acquisition	Distance from the candidate is from 12 to 30 cm	Person eyes can be acquired anywhere within a range from 21 to 37 cm	Shooting from a distance of 30-60 cm	Acquisition from a distance of 30-40 cm	Operating range is from 26 cm to 36 cm
FAR	0.00132 %	Not specified	0.000350 %	0.001290 %	0.0001 %

However, many other researchers have presented new solutions in this area [117, 118, 119, 120, 121, 122, 123, 124, 125].

One of the earliest systems for iris acquisition was developed using concepts proposed by Daugman [115] and Wildes [117]. Daugman's system takes an image of the iris with a diameter typically between 100 and 200 pixels, taking pictures from a distance of 15–46 cm, using a 330 mm lens. The Daugman model is based on information regarding the iris phase [126]. In this model, the representation of the iris is a binary-coded response of filter with iris texture information using two-dimensional Gabor wavelets. Classification is done based on the distance from the nearest class.

Another technique is iris recognition based on texture, which was proposed by R. P. Wildes [127]. In the case of Wildes' proposal, the iris image has a diameter of about 256 pixels and the photo is taken within a distance of 20 cm using an 80 mm lens. In Wildes' approach, the Laplacian of Gaussian filter is used for iris image processing in multiple scales. Then, the Laplace pyramid is constructed. Classification is based on the correlation coefficient.

The third method for iris recognition is Boles model [118]. The input iris images are processed to obtain one-dimensional signals (the zero crossing representation) using the dyadic wavelet transform. Here, iris classification is based on the comparison of the dimensions of the rectangular pulses which represents the zero crossings.

A common feature of the described methods for iris recognition is that the effectiveness of these methods depends on the correct segmentation of the iris in the image.

Stages of creation of iris code

Three successive phases in the process of creating the iris code can be identified [128]. They are determined respectively as image acquisition, segmentation, normalization and feature encoding shown in Fig. 2.17.

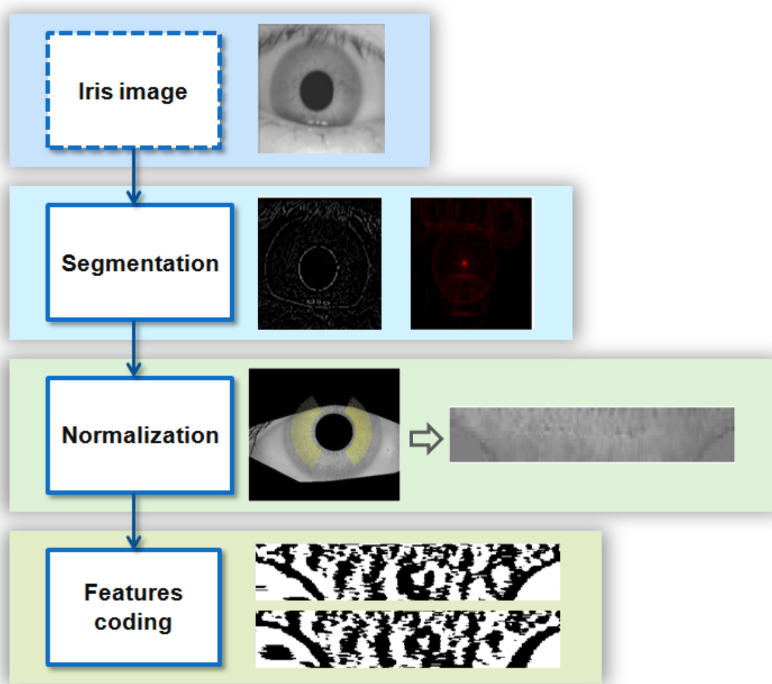


Fig. 2.17. Stages of creation of iris code

Separation of the iris from the whole eye area is realized during the segmentation phase. At this stage, it is crucial to determine the position of the upper and lower eyelids, as well as the exclusion of areas covered by the lashes. In addition, attention should be paid to the elimination of regions caused by light reflections from the cornea of the eye. The technique of iris location was proposed by the precursor in the field of iris recognition, John G. Daugman [115]. This technique uses the so-called integro-differential operator, which acts directly on the image of the iris, seeking a maximum normalized standard circle along the path - the partial derivative of blurred image relating to the increase of a circle radius. The integro-differential operator behaves like a circular edge detector in the picture, operating in a three-dimensional parameter space (x, y, r), i.e. the center coordinates and radius of the circle are identified, determining the edge of the iris. The algorithm first detects the outer edge of the iris, and then, limited to the area of the detected iris, looks to get its inside edge. Using the same operator, but by changing the contour of the arc path, the edges of the eyelids can also be identified, which may in part overlap the photographed iris.

The next step of iris recognition is iris image normalization. The main aim of the normalization step is the transformation of the localized iris to a defined format in order to allow comparisons with other iris codes. This operation requires a consideration of specific characteristics of the iris like a variable pupil opening, non-coordinated pupil and iris center points. The possibility of a circulation of the iris by tilting the head or as a result of eye movement in the orbit should be noticed.

Having successfully located the image area occupied by the iris, the normalization process has to ensure that the same areas in different iris images are represented on the same scale and same place of the created code. Only with equal representations can the comparison of two iris codes be correctly justified. Daugman suggested a standard transformation from Cartesian coordinates to the ring in this phase. This transformation eliminates the problem of the pupil's non-central position relative to the iris as well as the pupil opening variations with different lighting conditions. For further processing, points contained in the vicinity of 90 and 270 degrees (i.e. at the top and at the bottom of the iris) can be omitted. This reduces errors caused by the presence of eyelids and eyelashes in the iris area.

The last stage of feature extraction, which encodes the characteristics, aims to extract the normalized distinctive features of an individual's iris and to transform them into a binary code. In order to extract individual characteristics of normalized iris, various types of filtering can be applied. Daugman coded each point of the iris with two bits using two-dimensional Gabor filters and quadrature quantization.

2.4. Fast prototyping

In this dissertation, the main assumption is that the system has to work in an autonomous mode. There is many devices that support implementation of algorithm and fast prototyping techniques. This small device, compared to the PC unit, performs all the calculations on the input data from the CCTV camera (Fig. 2.18). The algorithm implemented on these devices allows the realization, which will eliminate the need for the transmission of the whole video signal from the camera to the PC, using the embedded system.

Fast prototyping has several advantages:

- The use of an adequate environment that enables verification of simulation in the real world.
- The graphical programming language adds clarity and enables the possibility of rapid algorithm modifications.
- Additionally, the implementation of the algorithm does not require any knowledge of the nuances (often very complex) related to the programming of digital signal processor module directly. For example, in case of Matlab / Simulink [36] environment, the Code Composer Studio [39] using C-language can be used.
- The main advantage of the intelligent monitoring system is that, it does not require expensive human resources for analysis of a real-time video sequence (in comparison with traditional CCTV systems).
- Another advantage is that, these kinds of systems are easy to deploy and generate real-time alerts (for example, alert for moving objects like person or vehicle entering a prohibited area).

Figure 2.18 shows an experimental vision system based on the Texas Instruments evaluation module. The video signal is transmitted directly from the camera to the DSP module. The processed images can be observed using a standard TV monitor.

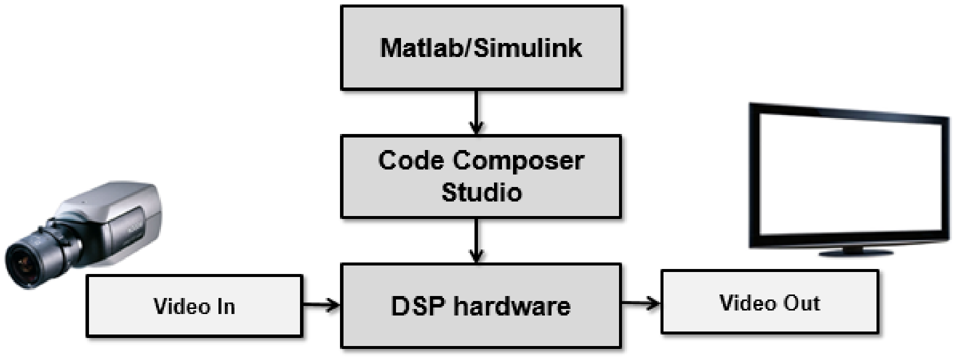


Fig. 2.18. Schema of fast prototyping with the use of Matlab/Simulink and Code Composer Studio environments and DSP hardware [34]

Evaluation modules for fast prototyping

There are many devices enabling video sequence processing and fulfilling the assumption of fast prototyping techniques. Some of these devices are presented in Fig. 2.19.

The first two devices in Fig. 2.19 are Smart Camera 1742 from National Instruments [129] and TMS320DM6437 EVM(evaluation module) from Texas Instruments [130]. The technical details and the possibility of rapid prototyping are described briefly latter. These two devices were chosen for the research because of their availability in the Division of Signal Processing and Electronic Systems at Poznan University of Technology. These devices are designed for processing small video sequence resolutions, see Section 6.2.

The third example of fast prototyping devices described in the thesis is the Raspberry Pi v.2.0 microcomputer [131]. The fast prototyping and implementation on this hardware is supported by Matlab/Simulink environment and Support Package for Raspberry Pi Hardware library. This library includes blocks enabling peripheral connections to that platform. This library provides, among others, direct access to GPIO (general-purpose input/output) video acquisition from a connected camera and audio capture.

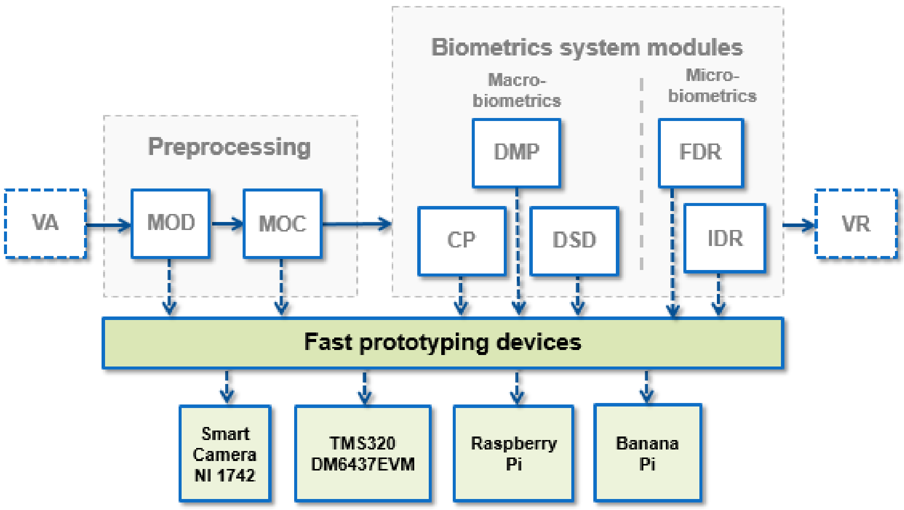


Fig. 2.19. An example of a fast prototyping device

The 2.0 version of Raspberry Pi microcomputer is based on the System-on-Chip Broadcom BCM2853, which consists of ARM1176JZF-S 700MHz processor, VideoCore IV graphics and 512 MB of RAM shared with the graphics processor. Raspberry Pi module can be supported by Linux (Raspbian distribution has been used in the research) and RISC OS. The device does not have a hard disk, hence, an SD card is used as data storage and operating system loading space. Besides the SD card slot, a platform is equipped with the following connectors:

- micro USB for power connection,
- DSI (display serial interface) port for LCD (liquid crystal display) connection,
- two USB 2.0 ports connected through a USB hub,
- LAN port 10/100 Mb/s,
- a full-size HDMI port, via which it is possible to connect the device to a monitor with Full HD video sequence resolution,
- CSI (camera serial interface) connector for Raspberry Pi Camera,
- 26 GPIO (general-purpose input output) pins, which can be controlled by software.

In conjunction with the Raspberry Pi microcomputer, a dedicated Raspberry Pi Camera can be used. The camera is equipped with a sensor of 5 megapixels resolution. The maximum resolution during recording mode is 1920×1080 pixels and the highest recording speed is 30 fps (frame per seconds).

The next example of device for fast prototyping is Banana Pi. This device is similar to the Raspberry Pi, but is more powerful. For example, in comparison to the Raspberry Pi, Banana Pi clock's speed is faster; has greater rate of data transfer, and is equipped with more RAM and SATA connector.

Description of Smart Camera NI 1742

National instruments Smart Camera 1742 [129] whose block diagram is presented in Fig. 2.20, is an embedded system for video acquisition, video processing and creation of automated inspection applications. The Smart Camera can be used in various applications [132, 133, 134, 135] e.g. quality control, object classification, static and dynamic event recognition, emergency situation recognition, OCR (optical character recognition), face recognition and many more.

The VGA (video graphics array; 640×480 pixels) CCD image sensor can acquire monochrome images of up to 60 fps. The camera is equipped with 128 MB RAM, 128 MB flash memory and is powered by a 533 MHz PowerPC processor. Thanks to these components real-time image-processing is possible without an external computing unit. This kind of solution enables the creation of a rugged, self-sufficient and independent system. Image acquisition can, therefore, be set off using an external trigger or captured in real time.

Communication is achieved with the use of two-gigabit Ethernet ports that also allow multiple users to remotely control it. Other noteworthy features include an external lightning, that is powered and controlled directly by the camera's software. The RS232 serial port, industrial protocols and quadrature encoder are also built-into the system. In some cases, a monochrome CCD sensor may become insufficient due to a lack of image-processing based on color frames. In this case, the color frame processing can only be performed in the simulation mode.

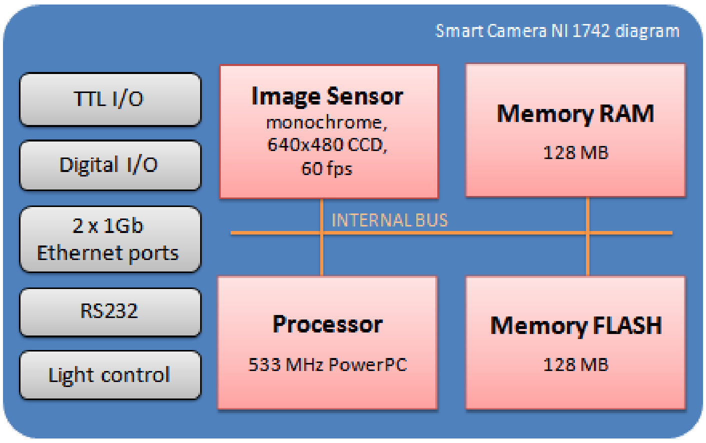


Fig. 2.20. Diagram of Smart Camera NI 1742

DSP module with TMS320DM6437 as a video processing system

TMS320DM6437 EVM is a fixed point unit dedicated to video processing [130] (Fig. 2.21). The analog video data is processed using the TVP5146M2 decoder, supports all typical video transmission standards within the NTSC and PAL formats. The Texas Instruments DM6437 processor operates up to 600MHz.

The Texas Instruments DM6437 consist of (Fig. 2.21) eight functional units (two multipliers for 32-bit results and six ALUs (arithmetic logic unit), which can produce four 16-bit MACs (multiply-accumulate) instructions per clock cycle, i.e., 2400MMACs (at 600MHz). In case of 8-bit data, it produces 4800MMACs per second. EVM is also provided with various kinds of memories: 128Mbytes of DDR2 DRAM, 16Mbytes of non-volatile flash memory, 64Mbytes NAND flash and 2Mbytes SRAM. In summary, the C6000 processor family provides high-performance computing at relatively low power (it can, therefore, function as an autonomous system).

The data transfer between the EVM module and PC can be realized via the USB connection or the Ethernet network.

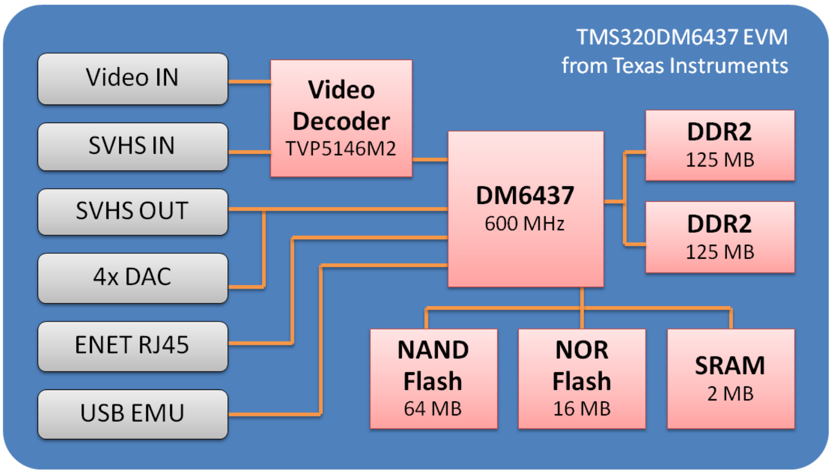


Fig. 2.21. Diagram of TMS320DM6437 EVM from Texas Instruments

Chapter 3

Preprocessing of video sequences from CCTV

In Chapter 3, the algorithms used in preprocessing the video sequences from CCTV systems has been described. The preprocessing of the video sequences consists of several stages.

The first stage is the detection of moving objects in the video sequence. Then, the moving objects are classified. In Chapter 3, besides the video acquisition and improved movement detection process, two types of moving objects classification are presented. The first one, classifies the moving objects into "person" and "vehicle" types. The second one is classification of moving object in order to predict the number of person in a single BLOB, when two or more objects are move close to each other.

3.1. Video acquisition in urban areas

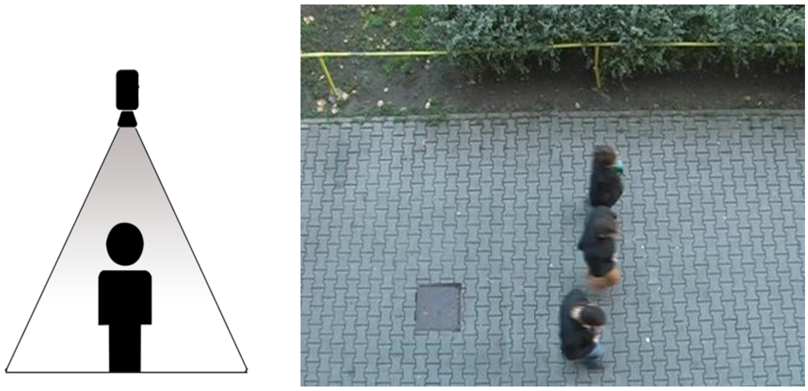
In general, the following dissertation concerns the effective analysis of moving objects in video sequences. These video sequences come from monitoring systems in urban areas. The goal is to detect various events (for example, people counting, dangerous situation detection, people recognition) in various conditions (like: lightning, weather or camera localizations).

During the experimental studies presented in this dissertation, a lot of video sequences and databases have been used. Some of them are described in Appendix A. In the task of moving objects detection, PETS database is very popular. However, this database is not diversified. The lack of various conditions like camera angles, various lighting or adverse weather conditions such as rain is noticeable.

For the above reasons, it was decided to create new databases of video sequences for the purpose of research.

For the recording of video sequences used for tests, the Panasonic NV-GS500 color camera has been used [136]. The essential features of the camera are as follows. The 3CCD (charge coupled device) 1/4.7" camera system was used for no light loss that allowed to obtain image details and rich color gradation. The OIS (optical image stabilizer) system minimizes vibrations, which can be caused by strong winds (the test recordings were performed with the use of a tripod). The Leica Dicomar lens (with 12x optical zoom and 3.3 - 39.6 [mm] focal length) was used in this video camera in order to prevent light reflections, glares, rings of light and "ghosts". The video sequences were recorded as uncompressed AVI (audio video interleave) files with 720×576 pixels with 25 fps.

The first set of recordings (named TVODCC - top view object detection, classification and counting) was made for the purpose of testing the system for moving objects detection, classification and primarily, people counting. Figure 3.1 shows the field of view of the camera in the first video sequences set. The camera was positioned vertically downward at a height of 8.30 m above the observed scene – what corresponds to the third floor of the building. The camera was set to observe people (sometimes animals and cyclists) who walk along sidewalks situated in parallel to blocks of flats. The length of the video sequences used for tests was 4 hours (the names of the video sequences are: 1 p.m., 2 p.m., 3 p.m., and 4 p.m.; Table 3.1). These sequences were recorded in normal weather conditions. In order to check the performance of the algorithm during bad weather conditions, a one-hour long video sequence showing heavy rain (the name of the video sequence is: RAIN), was recorded.



(a) (b)

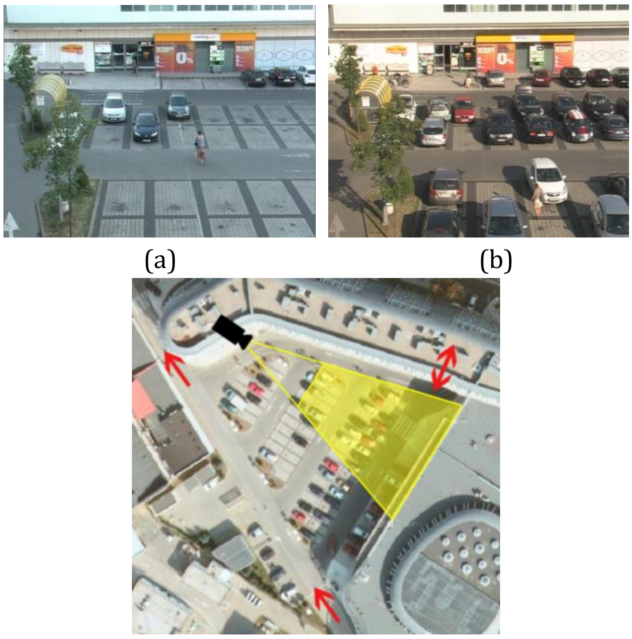
Fig. 3.1. (a) the camera location and (b) the field of view in the first set of video recordings named TVODCC

Table 3.1. Video recordings (TVODCC) used for research in moving object detection and classification and in people counting

No. of recordings	Recording name	Recording time
1.	1 p.m.	1:00 p.m. – 2:00 p.m.
2.	2 p.m.	2:00 p.m. – 3:00 p.m.
3.	3 p.m.	3:00 p.m. – 4:00 p.m.
4.	4 p.m.	4:00 p.m. – 5:00 p.m.
5.	RAIN	12:00 a.m. – 1:00 p.m.

The second set of video recordings (SVDMG – side view density map generation) was performed with the use of a camera located above a parking lot. Figure 3.2 shows illustrative images from the video sequences used during tests for moving objects detection, density estimation and top-view of the parking lot. The recordings were carried out in front of Green Point shopping center in Poznan – under real-life conditions. The camera field of view included the parking lot, the entrance to the shopping center and other retail shops. The recordings have been carried out from the third floor of a building located opposite to the shopping center – allowing to cover most of the surface of the parking lot and the most important areas (for example entrance to the shopping center) with the use of only one camera. Thanks to the video camera placed at the height of 16.3 m above the parking lot, the moving objects are entirely and clearly visible, which facilitates correct objects detection and tracking.

During the recordings, the Green Point shopping center was opened from 6:00 a.m. to 9:00 p.m. As a consequence, eight one hour long video recordings were registered, starting from 6:00 a.m., with one hour breaks between recordings (Table 3.2).



(a) (b)

(c) modified image from GoogleMaps

Fig. 3.2. (a), (b) sample video frames from the author's test database named SVDMG [25]; (c) top-view of the observed parking lot and the field of view of the camera (highlighted in yellow)

Table 3.2. Video recordings (SVDMG) times used for research in density maps of moving people generation

No. of recordings	Recording name	Recording time
1.	6 a.m.	6:00 a.m. – 7:00 a.m.
2.	8 a.m.	8:00 a.m. – 9:00 a.m.
3.	10 a.m.	10:00 a.m. – 11:00 a.m.
4.	12 a.m.	12:00 a.m. – 1:00 p.m.
5.	2 p.m.	2:00 p.m. – 3:00 p.m.
6.	4 p.m.	4:00 p.m. – 5:00 p.m.
7.	6 p.m.	6:00 p.m. – 7:00 p.m.
8.	8 p.m.	8:00 p.m. – 9:00 p.m.

The third set of video recordings (SVTD – side view, threat detection) was performed with the use of a camera located near the pedestrian crossing in order to threat detection. Figure 3.3 shows marked areas of interest (pedestrian crossing and traffic lights) in the three video sequences chosen for the tests. The first pedestrian crossing in the image, nearest to the camera, was observed in order to detect dangerous situations. The camera was set up in two places and the recordings lasted 40 minutes.

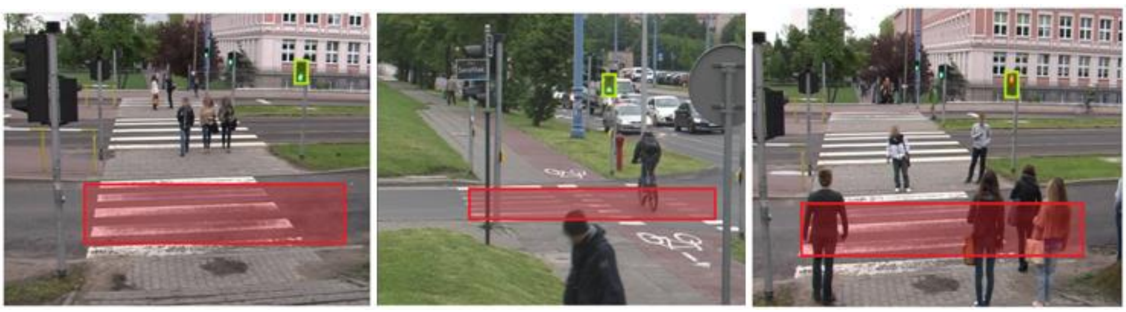


Fig. 3.3. Sample video frames from the author's test database named SVTD

3.2. Detection of moving objects in video frames

In this dissertation the first step in the preprocessing stage is moving object detection (Fig. 3.4). The moving object detection methods were described in Chapter 2. In the current Chapter methods are compared and the best method is chosen in order to the best operation in the outdoor scenes. In addition, the parameters of the moving objects detection model are selected and described in this dissertation.

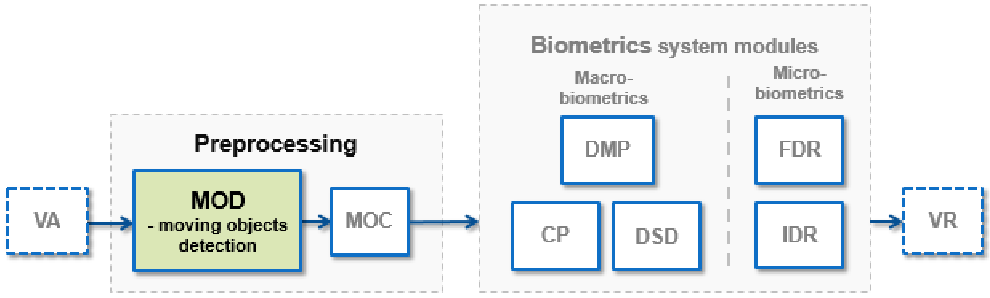


Fig. 3.4. Individual issue presented in the dissertation – moving object detection

3.2.1. Comparison of methods

To compare the moving objects detection methods, the previously described methods (Section 2.2.1) were tested with the use of PETS database (described in Appendix B).

Due to the fact, that research in this dissertation is related to outdoor scenes, the background subtraction method is not taken into account in this comparison – it can only be used indoors or in constant scenes due to a low resistance to interferences, for example: light, sliding doors, swaying trees or noise in the image (caused by low lighting conditions).

Subtraction of consecutive video frames

As the first, subtraction of consecutive video frames algorithm was tested. An example of a video frame from the PETS database processed by the SAD algorithm is presented in Fig. 3.5. In this example, objects in motion are clearly visible in particular areas of image. If an object stands still, even for more than 0.28 s, it will not be detected in the particular areas of image. This situation is visible on the right hand part of Fig. 3.5 – the object that suddenly stopped has not been detected. In some situations it is a huge drawback of this method.

The subtraction of consecutive video frames method for movement detection is fast - does not require a large amount of calculations. The SAD method does not return the exact coordinates of the moving object in the image. The method only returns the region of the image on which the motion is. This is also the downside of this method.

Optical Flow method

In the OF for motion estimation in video frames, the Horn-Schunck method [137] is performed for computing OF between actual and previous frames. Areas with a detected movement are marked with rectangles as presented in Fig. 3.6(d). Unfortunately, the method detects only boundaries (contours) of moving objects. Boundaries after morphological operations are visible on the binary image in Fig. 3.6(c). In addition, by using morphological operations with too large structuring element, the moving objects are not correctly separated from each other – Fig. 3.6(c). Through this, the Optical Flow model detects only 3 moving objects instead of 5.

An OF method can be used to track a region defined by a shape with clearly visible contours, which are well separated from the background.

Background subtraction with the use of GMM

The next compared method is the GMM with the use of "Motion-Based Multiple Object Tracking" algorithm [138]. This algorithm is designed with the use of GMM, morphological operations and the BLOBs analysis. The motion of each person is estimated and predicted by the Kalman filter [139].

Example of the algorithm [139] output (after optimization described later in Section 3.2.2, Table 3.3) executed on the PETS sequence is shown in Fig. 3.7. The GMM minimizes the impact of changes in lighting in case of outdoor scenes. Moreover, this model does not use a reference frame, making it resistant to slight loss of sharpness in the image caused by changes in weather condition like illumination or rain (this advantage of the GMM - resistance to weather changes - is shown and described in Section 3.2.2). This model, after selecting the appropriate parameters, is also resistant to slight movements caused by trees swaying in the wind.

It is important to select the GMM parameters according to the observed scene, because only then will the moving objects be properly detected.

Summarizing, GMM gives better results and is more resistant to changes in brightness than, for example, moving object detection with background subtraction technique or consecutive frame subtraction. Moreover, there is no problem with selecting the appropriate reference frame, which often changes during the day.

Therefore, GMM was selected for further study.

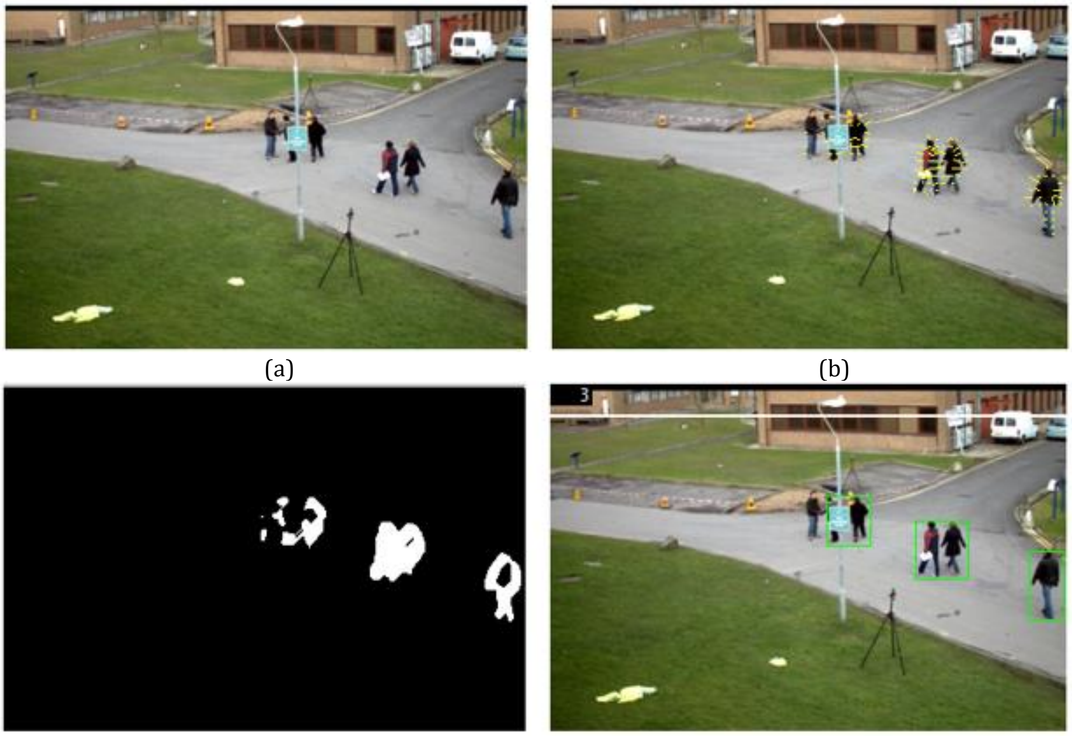
3.2.2. Selection of moving objects detection parameters

As was shown in the previous Section (3.2.1), GMM [138] is the best choice for correct moving object detection in outdoor scenes, especially in various conditions like weather or lighting.

Besides the selection of moving objects detection method, it is important to choose the model parameters according to the type of scene observed. In this research, the moving objects detection model is tested with the use of video sequence recorded on the sidewalk (named: TVODCC ; see Section 3.1).

The parameters of the moving object detection (and also tracking) were determined in order to precisely operate in outdoor scenes. Schema of the selected algorithm parameters, which was determined experimentally, are shown in Fig. 3.8.

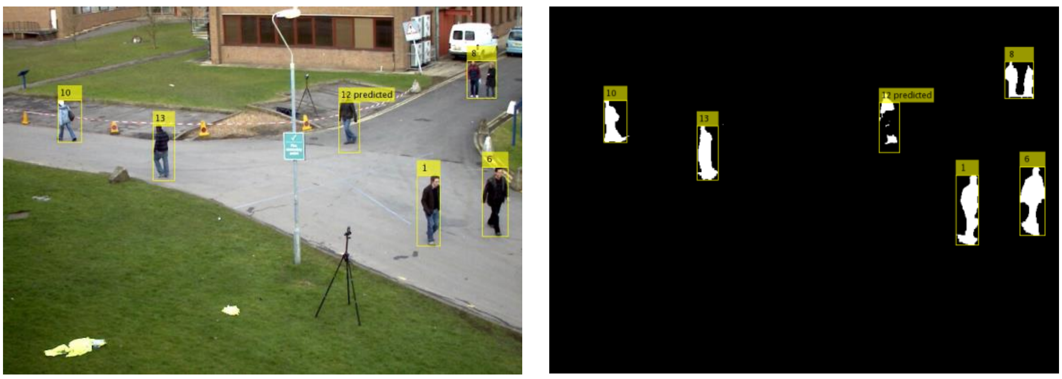
Fig. 3.5. Example of SAD method applied on the PETS database



(a) (b)

(c) (d)

Fig. 3.6. Example of the Optical Flow algorithm executed on the PETS database: (a) original image, (b) original image with marked pixels shifts, (c) binary image with detected objects in motion, (d)original image with marked objects in motion



(a) (b)

Fig. 3.7. Example of a modified GMM algorithm applied on the PETS database; (a) input image and (b) binary image with marked moving objects [73]

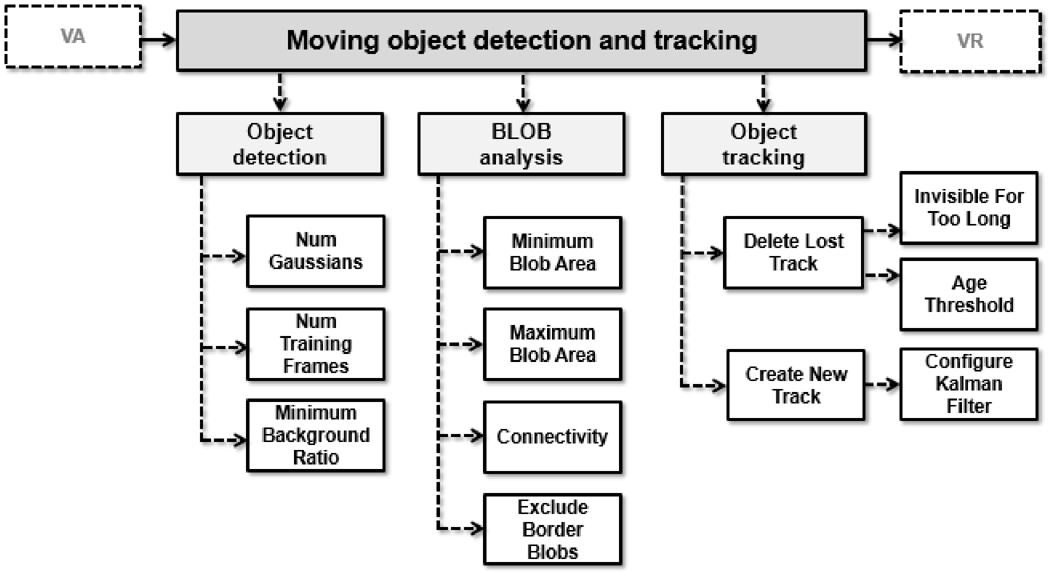


Fig. 3.8. Schema of moving objects detection (GMM), BLOB analysis and tracking and selected parameters

Object detection

The first selected parameter is the *NumTrainingFrames* (Table 3.3) specified due to the precise moving object detection. In case of larger values of this parameter (larger than 10), the shape of the BLOB (in the binary image) does not match the shape of the adequate moving object (in the RGB color space image), which is currently analyzed. In the case of unimproved *NumTrainingFrames* parameter, the shape of objects in the binary image is elongated depending on the direction of movement. This parameter is important in order to accurately determine the object shape for later operations (for example, for the classification of a number of objects in a single BLOB – see Section 3.3.2).

The most important parameter in GMM is the threshold which discriminates between background and foreground objects - the *MinimumBackgroundRatio.* The process of selecting the *MinimumBackgroundRatio* is shown in Fig. 3.9. The threshold was chosen by observing of the average area of binary representation of the moving objects. When the average area of the moving objects (in the binary image) was the same as the actual area of the moving objects (in the RGB image), the threshold was set at the value 0.87. Too large values of this parameter result in too small BLOB area (in the binary image) in relation to the actual area of the object in motion (in the RGB color space image). On the other hand, too small values of the *MinimumBackgroundRatio* parameter result in the opposite situation, i.e., the BLOB area is too large compared to the actual moving object area.

BLOB analysis

The next step of the moving object detection is the BLOB analysis. It is an analysis of groups of pixels in the binary image defined as a large binary object. This method allows for the detection of a person as a whole and not separately, for example, the hands alone. Because of this analysis, the precision of the moving object detection has increased.

Before BLOB analysis, the morphological operations are used for noise removal (morphological closing after the opening operation). Before performing the mentioned morphological operations, black holes in the objects of binary image have to be filled in.

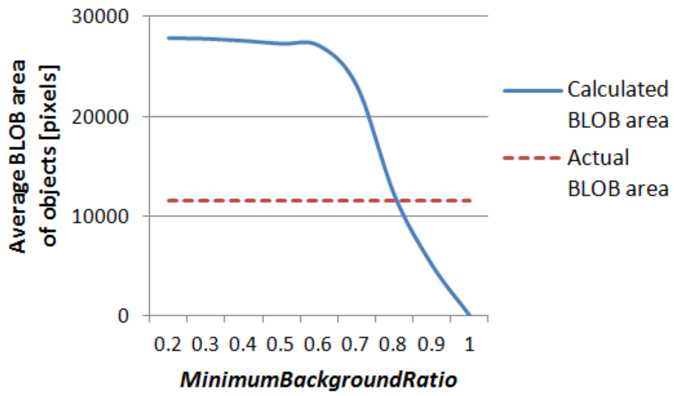


Fig. 3.9. The selection of the *MinimumBackgroundRatio* parameter in order to estimate moving objects [27]

The output of the BLOB analysis consists of three values:

- *BoundingBoxOutputPort* – stores position (X and Y coordinates) and dimensions (width and height) of the objects in the image,
- *AreaOutputPort* – stores a number of white pixels in specific BLOBs, it should be noted that this value is not obtained by multiplication of width and height values,
- *CentroidOutputPort* – stores positions (X and Y coordinates) of object centers in the image.

The *MinimumBlobArea* parameter defines the threshold under which a specified number of white pixels assigned to the moving object are not considered as BLOB object. The rejected objects do not participate in further analysis. The *MinimumBlobArea* parameter has been matched to take small objects into account like children, but do not take too small objects into account, such as: birds, dogs or the noise caused by a low light level in the scene.

The *MaximumBlobArea* parameter assures that objects larger than the prescribed threshold are not detected as moving objects. This option allows, for instance, to counting people or people density estimation only, with the exclusion of cars.

The *Connectivity* parameter specifies pixels that are connected to each other. In the algorithm, the value 8 was chosen in order to have a better connection of pixels in a single BLOB.

The *ExcludeBorderBlobs* parameter assures that the BLOBs which are connected to the image border are detected, but not labeled and used in further analysis. After optimizing the algorithm, this option of *BlobAnalysis* function is enabled – only objects, which are entirely visible by the camera are detected.

Object tracking

The next operation is object tracking. The detected moving objects were tracked with the use of a Kalman filter [139]. The tracking algorithm removes the object label in the case where the object is invisible for too many consecutive frames (more than 5 frames) and creates a new track for the presence of a new object in the observed scene after the time of 0.2 s (for 25 fps). The motion filter specified as *ConstantVelocity* applies to both X and Y directions. To achieve a better accuracy of the filter output, the *ConstantVelocity* algorithm parameter was reduced. Thanks to that, the predicted positions of the walking people are more precise.

Table 3.3. Selected parameters of Motion-Based Multiple Object Tracking algorithm from Matlab environment for bidirectional people counting; * 100000 (in the people counting), 4500 (in the density map generation system), 200000 (in threat detection system); ** this parameter is disable in the threat detection system

Algorithm module		Algorithm parameter	Default	Selected
Vision. Foreground Detector		NumGaussians	3	3
		NumTrainingFrames [frames]	40	10
		MinimumBackgroundRatio	0.7	0.87
Vision. Blob Analysis		Imopen (‘rectangle’, u) [px]	$[u, u]$ =[3, 3]	$[u, u]$ = [3, 3]
		Imclose (‘rectangle’, v) [px]	$[v, v]$ = [15, 15]	$[v, v]$ = [9, 9]
		MinimumBlobArea [px]	400	800
		MaximumBlobArea [px]	no specified	*
		Connectivity	no specified	8
		ExcludeBorderBlobs	false	true, **
Vision. Object Tracking	function: deleteLost Tracks()	InvisibleForTooLong [frames]	10	5
		AgeThreshold [frames]	8	5
	function: createNew Tracks ()	ConfigureKalman Filter: ConstantVelocity [px]	[200, 50], [100, 25], 100	[50, 25], [30, 15], 40

Algorithm accuracy

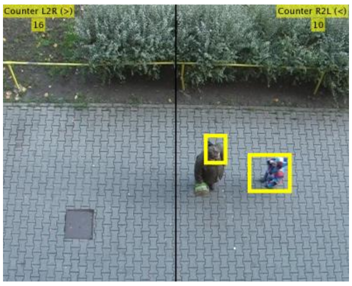
With the use of the above-described GMM parameters presented in Table 3.3, the algorithm for moving object detection was improved and tested with the use of a video sequence described in Section 3.1 (TVODCC) and visible in Fig. 3.10, 3.12 and 3.13.

During the research of the accuracy of movement detection with the use of video sequences recorded by the camera located above the sidewalk, 2582 BLOBs were examined. The assumption is that the BLOB must be visible entirely in the scene and every object is examined in the interval of 5 frames of video sequence.

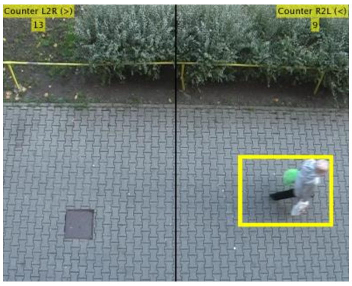
In order to determine the accuracy of the algorithm, weights for detected moving objects (BLOBs) are assigned. The weights define how well the GMM algorithm operates. A five-point scale was established (from 1 to 5) where weight no. 1 is very low and weight no. 5 is a very high quality of objects detection. Weight no. 3 means medium detection of moving objects’ quality. An example of the weights assessment of the detected objects is shown in Fig. 3.10.

The detection accuracy of moving objects was performed with the use of assigned weights. Fig. 3.11 shows the results of this analysis. As it can be seen in the histogram, the algorithm for object detection operates decently. Only 6.46 % of moving objects (of almost 2,600 objects) were assigned as low (weight no. 1) and very low (weight no. 2) detection quality. About 22.39 % of moving objects were assigned the medium (weight no. 3) detection quality. The rest of the results (71.15 %) were classified as objects with high (weight no. 4) and very high (weight no. 5) detection quality. Only one object was not detected by GMM algorithm. It was a person dressed in the same color as the background color.

The precisely selected algorithm parameters strongly influence the correct detection and tracking of objects in outdoor scenes (Fig. 3.12). Moving objects (even the objects which are close to each other) are well separated (Fig. 3.12 a, b) and small objects, like animals, are not detected - Fig. 3.12 c.



Weight for left object:
1 (very low)
Number of objects in the BLOB:1



Weight: 2(low)
Number of objects in the BLOB:1

Weight for both objects:
3 (medium)
Number of objects in the BLOBs:1

Weight:4 (high)
Number of objects in the BLOB:1

Weight:5 (very high)
Number of objects in the BLOB:2

Fig. 3.10. Examples of weights assessment to the detected moving objects; detected objects in motion are marked by yellow rectangles [27]

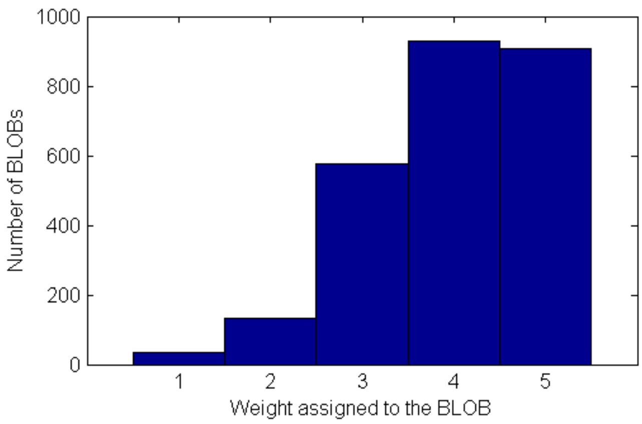


Fig. 3.11. Analysis of the moving objects detection accuracy with the use of the GMM algorithm, where weights are assigned as follows: 1 – very low, 2 – low, 3 – medium, 4 – high, 5 – very high detection accuracy [27]

(a) (b) (c)

Fig. 3.12. Examples of correct detection of moving objects with the use of the improved GMM algorithm; detected objects in motion are marked by yellow rectangles [28]

Algorithm operation during bad weather conditions

The algorithm also correctly detects moving objects in bad weather conditions, e.g. during a heavy rainfall. With the selected parameters, the proposed model does not detect small changes (like raindrops) in the analyzed video sequence. Only a large, steady stream of water (quite uncommon in video surveillance systems) can rarely be falsely detected as a moving object. Examples of these situations are shown in Fig. 3.13.

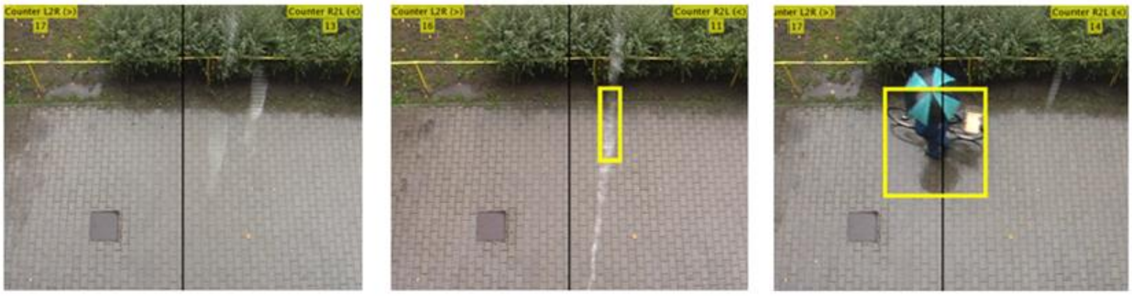


Fig. 3.13. Examples of algorithm for movement detection during strong rain; detected objects in motion are marked by yellow rectangles [27]

3.3. Classification of moving objects

The next step of preprocessing of video sequences after 'moving object detection' is 'moving objects classification'. Two methods for object classification are examined in this dissertation (Fig. 3.14):

- classification of moving object into two types, "person" and "vehicle" allows to distinguish objects between each other;
- classification according to the estimated number of people in a single BLOB helps to predict the number of persons in a single BLOB in a situation where people walk close to each other and are detected (in the previous step – moving objects detection) as a single BLOB in the binary image.

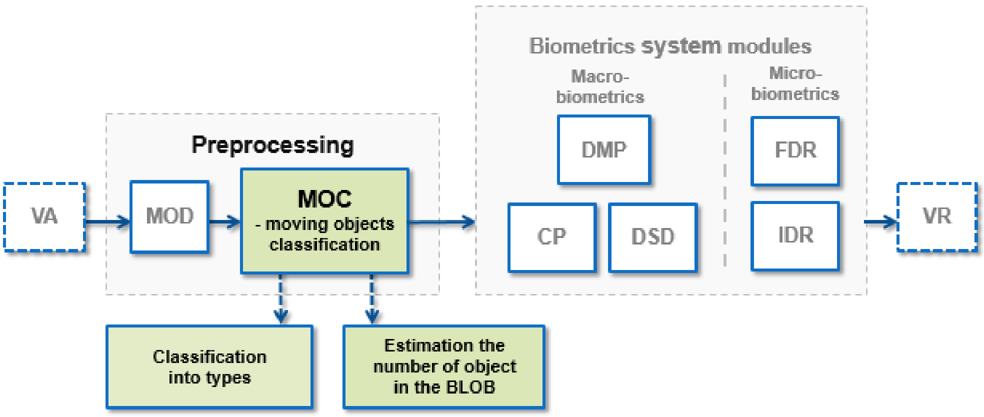


Fig. 3.14. Individual issue presented in the dissertation – moving object classification

3.3.1. Selection of classification parameters for "person" and "vehicle" types

The study of the classification into "person" and "vehicle" types using Feret's coefficient according to camera location is presented.

Feret's coefficient is a low level classifier, but it is useful for systems where every frame must be processed in real-time, for example, in embedded systems, in which the processor is clocked at a lower frequency than a standard PC processor.

Feret's coefficient threshold should be selected depending on the α angle – this is the angle of the camera location in relation to the ground. The visualization of the α angle of camera location is shown in Fig. 3.15.

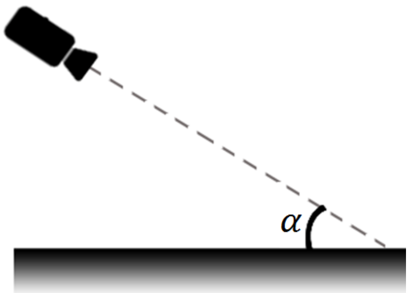
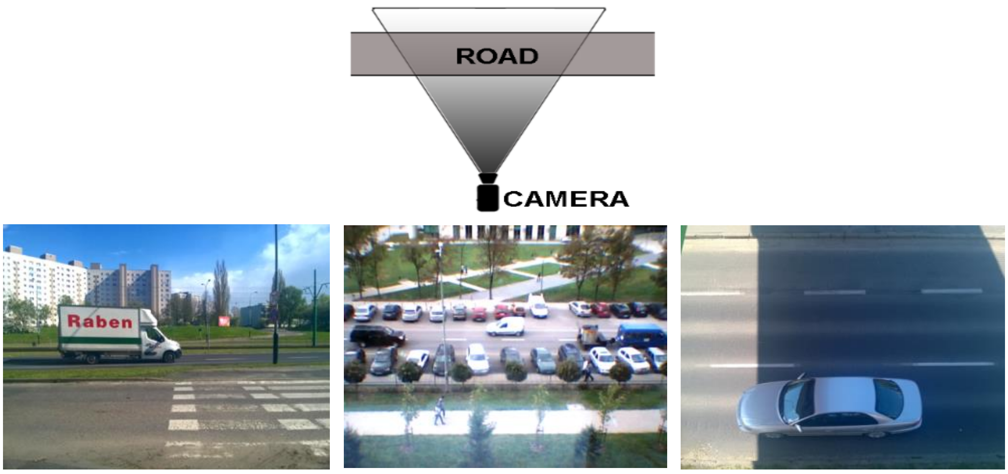


Fig. 3.15. The visualization of the angle of camera location in relation to the ground

In the case when the camera is located perpendicular to the road (Fig. 3.16) and the moving objects in the scene are visible from the side, there is no problem with classification of objects into types "person" and "vehicle" using Feret's classifier (Fig. 3.17). The moving objects can be well classified for all angles of camera location – as well as angles equal to 0°, 45° or 75°. However, there is no test data (the video sequence) with vehicles in the scene when the angle is equal to 90°.



$\alpha = 0°$ $\alpha = 45°$ $\alpha = 75°$

Fig. 3.16. The visualization of the angle of camera location in situations, when the camera is located perpendicular to the road

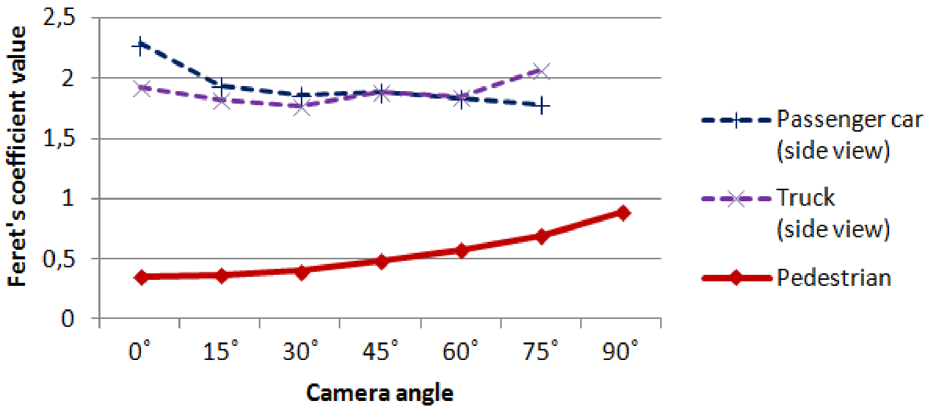
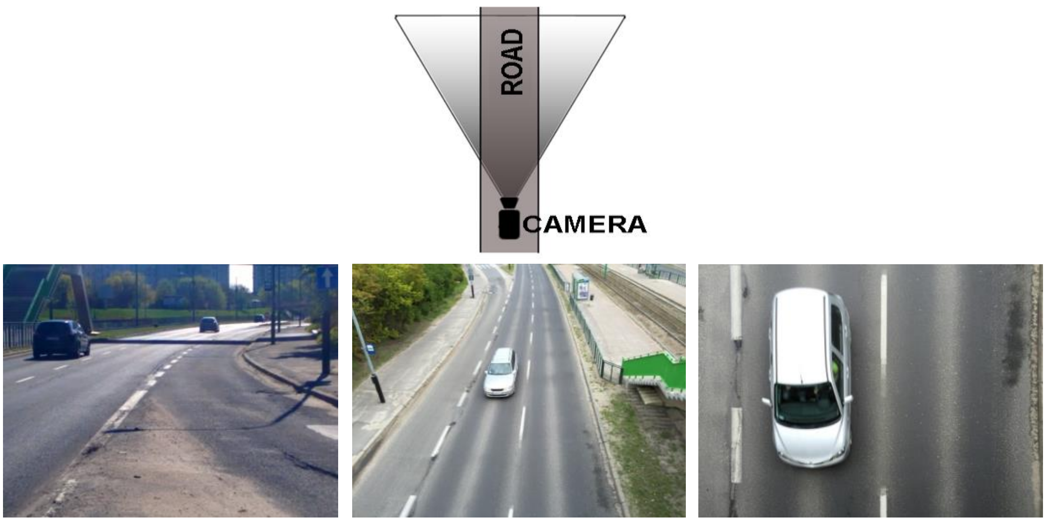


Fig. 3.17. Plots of Feret's coefficient value depending on the angle of camera location for different types of objects: "pedestrian" and "vehicle"; the camera is located perpendicular to the road

In the case when the camera is located parallel to the road (Fig. 3.18) and the moving objects in the scene are visible from the front/back there is no problem with object classification, but only for angles smaller than 30° (Fig. 3.19). In the case of larger values (larger than 30 °), a classification on the basis of Feret's coefficient is not possible and another parameters must, therefore, be used – for example, the BLOB area of moving objects (performed in the binary image).

An example of the classification of moving objects into "person" and "vehicle" types is shown in Fig. 3.20. In this example, Feret's coefficients are used for object classification into two types: "people" and "vehicle". With an adequate camera localization (perpendicular to the road) and a proper selection of the camera angle (α=15°), there is no problem with object detection and classification. In this example, the moving objects are detected with the use of optimized GMM (described in Section 3.2.2). Then, the objects are classified into types according to Feret's coefficient threshold. In this camera location and in this type of observed scene, the author set the classification threshold at a value equal to '1.0' – according to the plot shown in Fig. 3.17. When the Feret's coefficient of the moving object is smaller than the threshold, the moving object is considered to be a person ("P" notation in the Fig. 3.20) or otherwise as a vehicle ("V" notation in the Fig. 3.20).



$\alpha = 0°$ $\alpha = 45°$ $\alpha = 90°$

Fig. 3.18. The visualization of the angle of camera location in a situation, when the camera is located parallel to the road

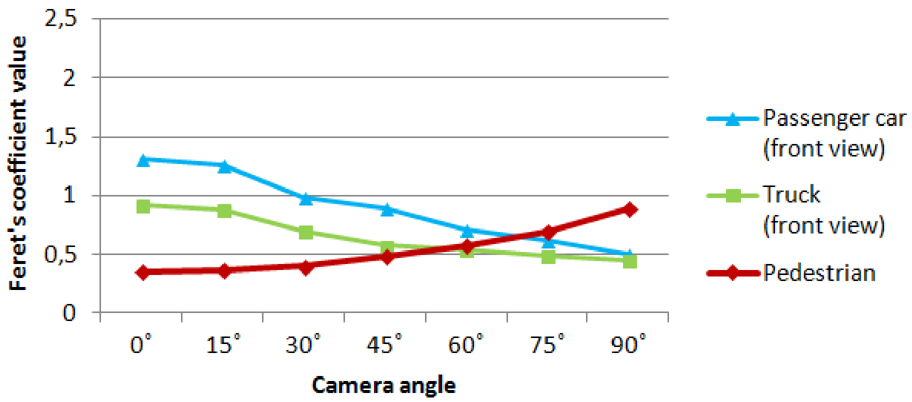


Fig. 3.19. Plots of Feret's coefficient value depending on the angle of the camera location for different types of objects: "pedestrian" and "vehicle"; the camera is located parallel to the road

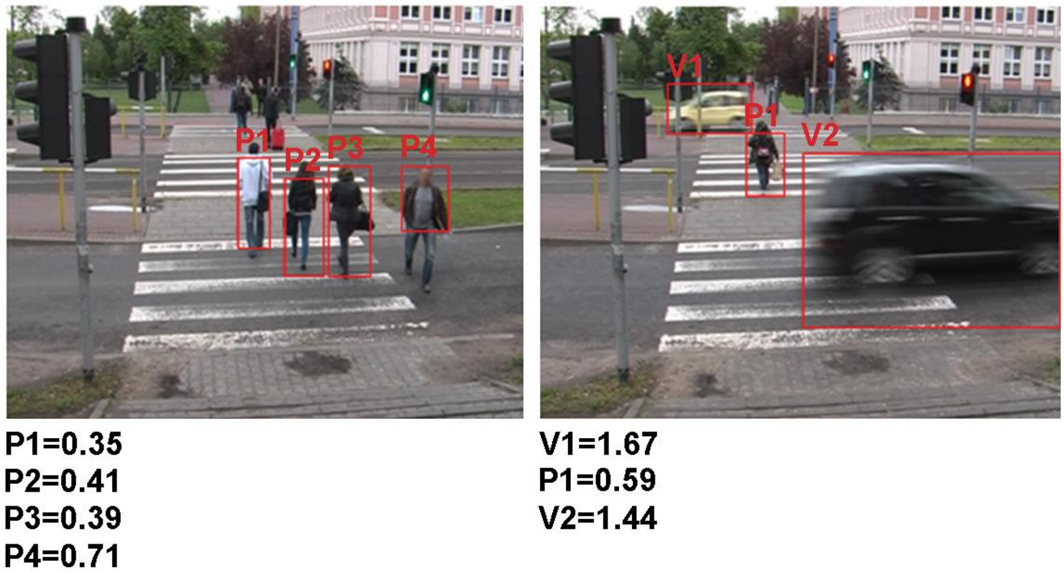


Fig. 3.20. Classification of moving object with the use of the Feret's coefficient; notations: 'P'-person, 'V'-vehicle

3.3.2. Influence of people in the BLOB prediction parameters

The second part of the classification study is the number of persons in the single BLOB prediction in the situation, when people are too close to each other or people are holding hands. In this situations, they are not properly separated by the algorithm for moving object detection.

It is important to separate the objects within a single BLOB. Figure 3.21 shows examples of such situations.

It was examined that improper separation of objects in the single BLOB happens frequently. About 13% of all examined BLOBs (more than 2.5 thousands) contain 2 or more people. This statistics is shown in Table 3.4. This table shows the number of people in the detected BLOBs with division into particular video sequences in the database (TVODCC).

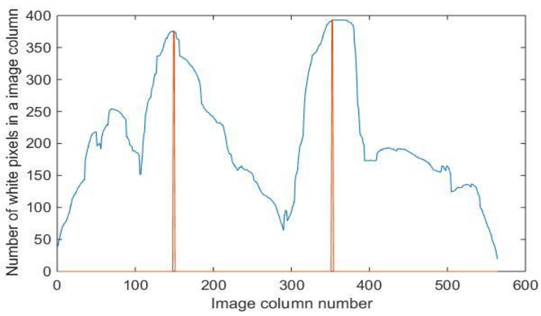
Fig. 3.21. Examples of incorrect detection of moving objects with the use of the optimized GMM algorithm; detected objects in motion are marked by yellow rectangles [28]

Table 3.4. Number of people in the detected BLOBs in the particular video sequences

	1 p.m.	2 p.m.	3 p.m.	4 p.m.	Total	Total [%]
BLOB not classified as person (animals and others)	24	31	31	9	95	3.68
1 person in the BLOB	595	494	600	459	2148	83.19
2 person in the BLOB	75	36	100	98	309	11.97
3 person in the BLOB	0	0	13	17	30	1.16
Total	694	561	744	583	2582	-

In a previous work [61], in order to avoid a detection of two objects as one, the author of this dissertation together with a team, applied an additional criterion. When two objects are connected within the BLOB, the local maximum of the binary image histogram within the specified range is searched (description in Section 2.1.2). The analysis of histogram in order to object classification is shown in Fig. 3.22. The method of histogram analysis was used in determining the number of people in the BLOB.

(a) (b) (c)



(d)

Fig. 3.22. Example of moving objects detection (background subtraction method), which are very close to each other: (a) input image, (b) background, (c) binary image after morphological operations with marked region of interest, (d) number of white pixels in particular image columns of selected BLOB area with marked maxima

In the current solution (which used object detection using GMM approach), the previous algorithm that separates close objects with the use of histogram, cannot be efficiently used. This is because the objects that are detected with the use of the GMM, in the binary image have simplified shapes and inaccurate contours compared to the actual objects, i.e. those in the RGB image. For this reason, the separation of the objects with the use of histograms is, in my opinion, not recommended when we use GMM in object detection.

Instead, to estimate the number of objects in individual BLOBs using a classifier was decided.

Before using the classifier, 2582 BLOBs were examined and rated to train the classifier. The assumptions are that the BLOB (moving objects in the binary image) must be entirely visible in the scene and every object is examined in the interval of 5 frames of video sequence. This is manually done to create a model of classifier, which will later be applied to the people counting algorithm.

The BLOBs width, height and area (the number of pixels in the BLOB) were extracted from the detected objects and the actual number of people in the BLOB were assigned to the specified object in the binary image. The results of the analysis of the BLOB's dimensions (the width and the height) and the area distributions according to the assigned BLOB class are shown in Fig. 3.23-3.25.

The BLOB dimensions: the width, the height and the area do not rise linearly with the number of objects in the BLOB (Fig. 3.23-3.25). The data between classes overlap and it is hard to predict the number of people in the BLOB in the linear way.

In order to correct classification, a probabilistic naïve Bayes classifier was used. This classifier makes use of a multidimensional Gauss function. It is also an independent model of features, which is easy to implement and, additionally, is not computationally time-consuming. The part of the research on the calculation time is presented later, in Section 4.2.3.

When creating a classifier model, different configurations of features, which can assist in the prediction of the number of people in the BLOB, were tested. These are the following pairs of features (listed together with correlations between attributes):

- width and area – correlation is equal to 0.805;
- height and area – correlation is equal to 0.840;
- width and height – correlation is equal to 0.559;
- area, width, and height.

In principle, the naïve Bayes classifier should operate at its best for the least correlated data [57]. This assumption was tested for the default parameters of the classifier. Table 3.5 shows the selection of BLOB features used for naïve Bayes classifier training. In this table, a feature with the highest efficiency for various tests was marked using a green color.

The best results of classifier effectiveness (at 66% of the tests, Table 3.5) were obtained with the use of the BLOB width and height, where the correlation between the features was 0.55. In 25 % of the tests, a pair of BLOB features which gave the highest classifier results were the height and area. In this case, the correlation between the features was 0.84.

Taking the above results into account, the author decided to use the BLOB's width and height feature to build the classifier model.

Table 3.5. Efficiency of naïve Bayes classifier for various sets of training and test video sequences and for combination of various BLOB's features [28]

		BLOB's features			
		Width and Area	Height and Area	Height and Width	Height, Width and Area
Correlation:		0.8	0.84	0.55	-
Video sequence (name)					
Training	**Test**				
1 p.m.	**2 p.m.**	85.74	92.51	92.16	91.09
2 p.m.	**3 p.m.**	77.02	82.93	84.01	83.06
3 p.m.	**4 p.m.**	69.47	71.18	71.53	66.72
4 p.m.	**1 p.m.**	72.91	88.76	76.80	77.52
1 p.m., 2 p.m.	**3 p.m., 4 p.m.**	76.11	81.31	82.06	81.69
2 p.m., 3 p.m.	**4 p.m., 1 p.m.**	78.86	79.56	82.54	78.86
3 p.m., 4 p.m.	**1 p.m., 2 p.m.**	75.54	90.44	86.06	82.79
4 p.m., 1 p.m.	**2 p.m., 3 p.m.**	83.49	88.35	89.73	86.90
1 p.m., 2 p.m., 3 p.m.	**4 p.m.**	83.45	72.21	74.61	71.87
2 p.m., 3 p.m., 4 p.m.	**1 p.m.**	81.84	88.90	89.63	86.02
3 p.m., 4 p.m., 1 p.m.	**2 p.m.**	84.14	91.62	92.16	87.89
4 p.m., 1 p.m., 2 p.m.	**3 p.m.**	80.24	85.48	87.10	84.27

Table 3.6 shows the selection of parameter of various features of *Fitcnb* function. The *Fitcnb* function from the Matlab environment creates and returns a multiclass naïve Bayes model.

The best chosen parameters (with the highest efficiency of a classifier) of the *Fitcnb* function were marked with green color in the Table 3.6.

The first adjusted parameter was the *Distribution* of data. The *Distribution* parameter can take the following options: the *Normal* (Gaussian), the *Kernel* smoothing density estimation, *Multinomial* and *Multivariate multinomial*. The default (marked by "d" in Table 3.6) distribution is Normal, however, the *Kernel* smoothing density estimate gives the highest efficiency of naïve Bayes classifier for the test data.

The *Prior* probabilities feature options are: *Empirical* (determines class probabilities from class frequencies), *Uniform* (sets the class probabilities equal) and *Vector* (specifies one scalar value for each class). The last *Vector* option of the *Prior* feature*,* gives the greatest efficiency of the naïve Bayes classifier. The vector components were created with the use of data from Table 3.4 and gives a little bit higher results than the *Empirical* parameter.

While testing the influence of one of the feature parameters have been checked, the rest of the parameters were set to the default value.

In the next step, the chosen features (marked in green color in Table 3.6) have been used simultaneously for the best effectiveness of naïve Bayes classifier. Then, the chosen features together with *Weights* for input data (described earlier in Section 3.2.2) have been verified, but with the use of weights the effectiveness of the classifier was lower, finally, the weights were not used in later analysis.

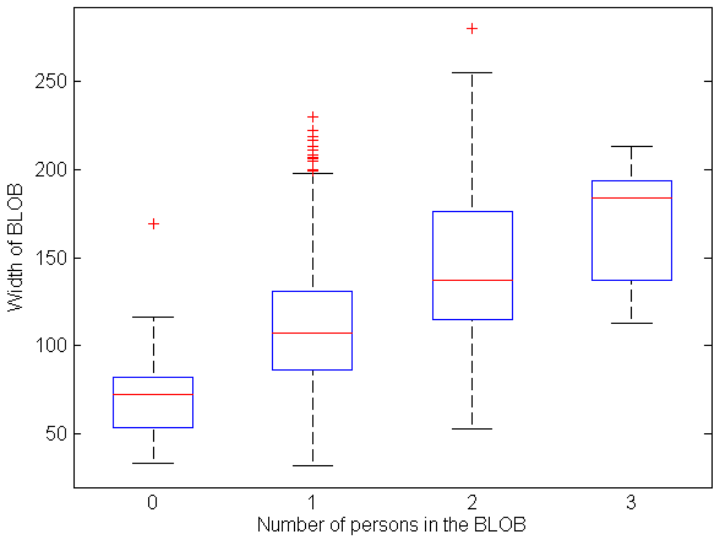


Fig. 3.23. Width of the BLOB according to the number of people in the BLOB [28]

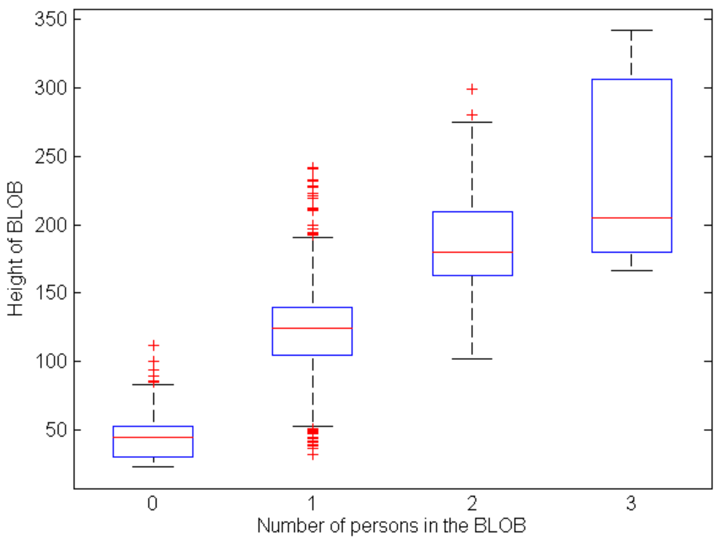


Fig. 3.24. Height of the BLOB according to the number of people in the BLOB [28]

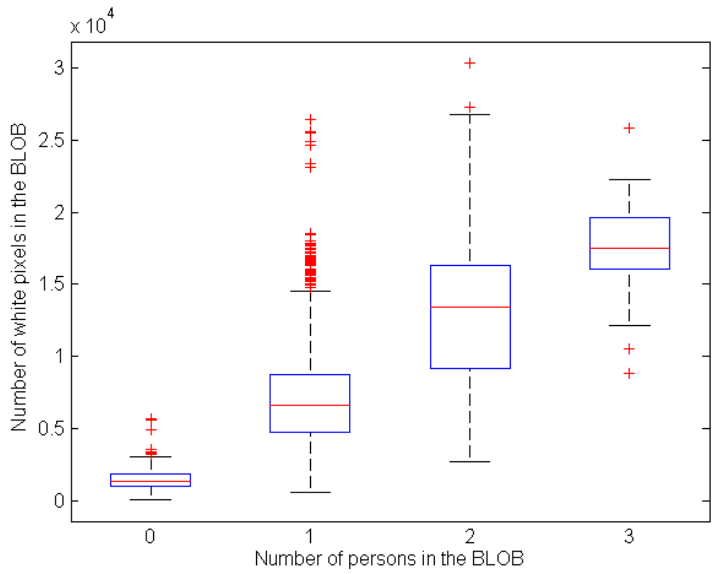


Fig. 3.25. Area of the BLOB according to the number of people in the BLOB [28]

Table 3.6. Efficiency of naïve Bayes classifier for various sets of training and test video sequences and for various *Fitcnb* function features [28]

		Video sequence (name)				
	Training	**1 p.m., 2 p.m., 3 p.m.**	**2 p.m., 3 p.m., 4 p.m.**	**3 p.m., 4 p.m., 1 p.m.**	**4 p.m., 1 p.m., 2 p.m.**	**1 p.m., 2 p.m., 3 p.m., 4 p.m.**
	Test	**4 p.m.**	**1 p.m.**	**2 p.m.**	**3 p.m.**	**RAIN**
Feature	**Feature parameters**					
Distribution	Normal (d)	74.61	89.63	92.16	87.10	67.32
	Kernel	77.53	91.50	90.73	87.63	65.79
	Multivariate multinomial	75.64	87.18	86.99	82.93	72.81
	Multinomial	61.58	63.69	80.75	71.77	38.38
Prior	Empirical (d)	74.61	89.63	92.16	87.10	67.32
	Uniform	62.09	69.31	74.15	81.99	48.03
	Vector; V=[3.68, 83.19, 11.97, 1.16]	74.96	89.63	92.34	86.96	67.32
KSSupport	Unbounded (d)	74.61	89.63	92.16	87.10	67.32
Chosen features (marked in green):		**77.18**	**91.64**	**91.80**	**87.37**	**65.79**
Chosen features (marked in green) with added weights for each attributes:		77.70	92.36	90.73	86.69	62.28

Typically, during the tests for the effectiveness of the classifier, the ratio 1:3 of the training data to the test data is used. In the research on the bidirectional people counting, to train the classifier data from three video sequences were used and in the next step, the trained classifier was evaluated using the data from the last - the fourth sequence. The above division is justified by the fact, that in some cases one class in the set occurs very rarely. For example, in the case of the sequences recorded at 1:00 and 2:00 p.m. (see Table 3.4), there is no situation, where 3 people are in a single BLOB in the binary image. Therefore, the classifier may have problem with the correct interpretation of the results. Generally, the problem of correct classification can refer to the cases, when the training set does not contain representative attributes of a rare class. In this case, the probability of effective classification equals zero.

The preprocessing steps described in Section 3 are used in the later Sections.

In Sections 4.1 and 4.3 the use of classifier is shown, which determines the moving object type ("person" vs. "vehicle") in the monitoring systems. This type of classification supports the generation of moving objects density maps (Section 4.1) and supports threat detection in urban areas (Section 4.3).

In Section 4.2 - the application of classifier, which estimates the number of people in the single BLOB, in the system for bi-directional people counting will be presented. It will be shown how strongly the use of classifier for estimation of the number of persons in the BLOB, and other methods, will affect the effectiveness of bi-directional people counting.

Chapter 4

Improving the effectiveness of macro-biometric procedures

In this Chapter, the macro-biometric modules are analyzed. The author concentrate on three issues including:

- estimation of people density - this is analysis of the number of people going through specified locations and analysis how much time these people spend in the specified location. In Section 4.1 disadvantages of commercial system - data normalization and using a non-linear reference has been resolved. Additionally, in this dissertation, the accuracy of projection the generated density maps from two different camera views was presented.
- precise people counting in both directions. Section 4.2 describes the standard and modified algorithm for bidirectional people-counting. The accuracy of counting people using the optimized GMM algorithm for moving object detection and classification the number of persons in the BLOB was shown. In the last part of the Section, the processing time of the algorithm was presented.
- detection of dangerous situations in urban areas. The aim of this part of the study (described in Section 4.3) was to create models for detection of dangerous situations from video monitoring sequences. An automatic recognition of dangerous situations is necessary in order to enhance the concentration of the video monitoring operator. The main task during the operation of the algorithm is to detect, observe and categorize objects depending on the location and the direction of view of the camera. When the camera observes, for example, roads, tracks, pedestrian crossings or areas with prohibited access, the different approaches must be implemented in the CCTV system.

4.1. Analysis of the behavior of moving people using density maps

The author of this dissertation, encountered issues of motion density estimation, within the project "Analysis of the density of people in commercial areas", conducted by the BranchBrothers company. Thus, in the research on density map generation, the author of the doctoral dissertation, together with a team, generate the density maps based on long-term observation (one-day long) carried out in indoor scenes. This research - the automatic generation of moving people density map (also called heat maps) - was published in [62, 73] papers.

In this dissertation only people movement in outdoor scenes are observed (Fig. 4.1). During the generation of density maps [25], the observation of people in the given location is done.

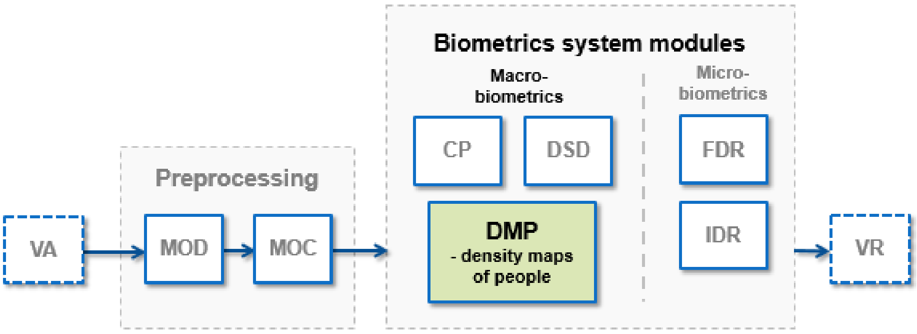


Fig. 4.1. Individual issue presented in the dissertation
– generation of density maps of people movement

4.1.1. Differences between methods of density estimation

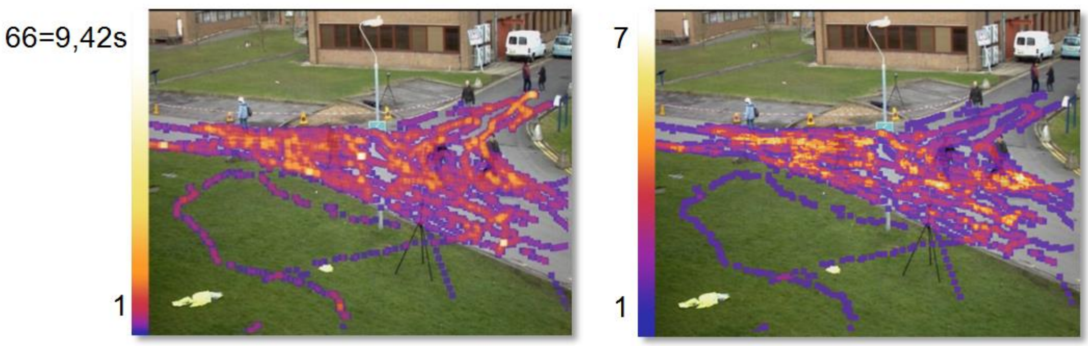
The different types of density maps were described in Section 2.2.1. These were density maps based on time and density maps based on the observation of moving objects indexes.

In this Section, the differences between these kinds of density maps were shown and tested with the use of the PETS database.

A density map based on time is presented in Fig. 4.2(a). If the person is moving fast, the trajectory (created on the basis of movement in every video frame) is marked on the density map as a dotted blue line – this is visible on the bottom part of Fig. 4.2(a). When the person stops for a while, it is shown on the density map by a lighter color - visible on the top-right part of the image from the database (Fig. 4.2(a)). The maximum value in this density map, based on time, from this set of the database was 66. It means that for this period of time (i.e. 9.42 s) people passed through the same areas.

The algorithm of density map generation based on the observation of moving objects' indexes has been also tested – results are presented in Fig. 4.2(b). The maximum value in this density map was 7 – it means that seven people passed through the same specific point.

The recordings from the PETS database are too short to generate valuable density maps of moving people. Density maps from short sequences show only paths that people follow. Thus why, in latter analysis one-hour long and one-day long video sequences are processed.



(a) (b)

Fig. 4.2. Example of the generated density map of people (a) based on time and (b) based on the observation of the indexes of moving people with the use of images from the PETS database

4.1.2. Analysis of density map visualization

Algorithm for density map generation

The software for generating density maps was prepared [73]. The algorithm processes the test video sequence frame-by-frame (with 25fps). During the research, in order to speed up the processing time of the algorithm, adequate mask was applied only in areas where movement were not occurred.

The first step is the detection of moving objects. The optimized moving object detection model was described, in details, in Section 3.2. Next, the moving object classification into types "person" and "vehicle" was done (Section 3.3). In the rest of the algorithm only person are detected as the moving object.

Each person is represented by coordinates of vertices of rectangles surrounding it. In the case when people (vertical objects) are analyzed, the middle bottom part of the rectangle is calculated. The bottom part of the persons was chosen in order to accurately image the paths of people. Vehicles are excluded from the analysis in the previous step of algorithm – moving object detection, but if vehicle (object in horizontal shape) was analyzed, the geometric center of the object (in the binary image) would be taken into account as a point, that generates the density map. It is worth noting that individual moving object are indexed with same values starting with their appearance on the scene until they disappear from the scene or until they stop for a specified period of time – in this case they are blended into the background in the binary image.

In the two dimensional accumulation matrixes, (with the same size as the resolution of processed video frames i.e. 576×720 pixels) the corresponding cells are incremented in two ways: with the used of the algorithm based on the time and observation of moving objects indexes [73]. In the results, the accumulation matrixes contain adequate numbers of objects that passed through a given point and the time in which a person was at that point.

After processing the entire video sequence, the density map is generated based on the values in the accumulation matrix. The maximum value in the accumulation matrix is noted and according to this value, the whole accumulation matrix is scaled to values ranging from <1; 255>. After accumulation matrix scaling, the values cannot be correctly read, however, a visualization of the results in the image using 2D histogram are more intuitive (the initial density map values are stored in the matrix).

The next stage is visualization of the density map - the reference image loading (Fig. 4.3), which has a resolution of 255×50 pixels; the column number is insignificant. Each value in the accumulation matrix (after scaling) corresponds to a color corresponding to a line in the reference image. Thanks to these operations, the visualization of the density of people with the use of different colors of reference maps takes place in an easy way. After scaling, the blue color on the density map indicates the minimum values in the accumulation matrix (value of "1"; the value of "0" is not color-coded, because in these points there was no movement at all) and the white color on the density map indicates the maximum value in the accumulation matrix.

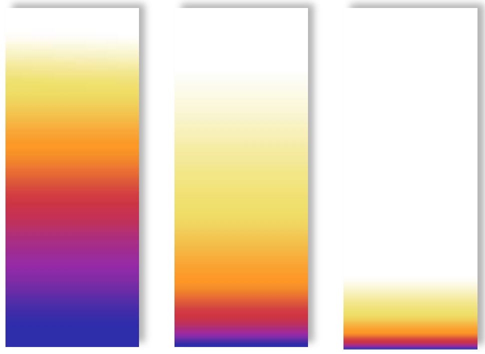
It is worth of noting, that the reference image-type should be chosen depending on the length of the video sequence or depending on the density of movement on the scene.

Various types of visualization of density maps

The literature and commercial systems have usually reference images with equal scales – they are insufficient data visualization from long video sequence. That why, there were three images prepared (Fig. 4.3) in which the color ranges are: (Fig. 4.3a) equal, (Fig. 4.3b) logarithmic-ver1, (Fig. 4.3c) logarithmic-ver2.

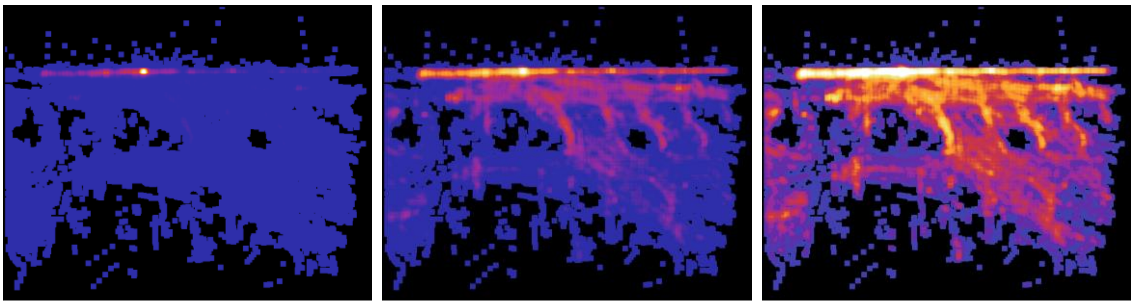
Each reference image is used for different types of situations and video sequences. The image with equal range, shown in Fig. 4.3(a), is suitable for short (up to several minutes) videos, where the distribution of motion in a video is uniform. The logarithmic ranges of colors, shown in Fig. 4.3 (b, c), are used, both for long sequences (an hour or a day long) or short sequences, where people follow the same paths (movement density is high).

A study of the appropriate choice of a reference image for long video sequences is shown in Fig. 4.4. The use of a linear reference image for long video sequences has a negative affect on the visibility of maximum values, which in this case, are located in the top part of the image (Fig. 4.4 (a)). For long recordings, when using the logarithmic reference image, the maximum values (placed in the image, where the movement was the highest) are more visible (Fig. 4.4 (c)). As a result, the areas with the largest moving people density can be, accurately, visualized and determined.



(a) (b) (c)

Fig. 4.3. Different scales used in density map generation algorithm:
(a) equal, (b) logarithmic_ver1, (c) logarithmic_ver2



(a) (b) (c)

Fig. 4.4. Example of using different scales intervals in density map visualization on the same one-hour length video sequence, (a) linear, (b) logarithmic-ver1, (c) logarithmic-ver2 scale

Normalization and visualization of density maps based on time

Commercially available algorithms have another disadvantages, they do not have normalization of results. Normalization involves scaling of an hour-long video sequence results to the highest result, for example, from the whole day or week. This solution presents results in a better way – the differences between individual hours and traffic distribution are significantly more visible.

Table 4.1 shows the examples of generated density maps based on time at Green Point shopping center parking lot (sequences named SVDMG; Section 3.1). All results are presented in Appendix C. The movement distributions of moving people are clearly visible thanks to dividing a day-long recording into hour-long video sequences.

The density of movement increases every hour (up to 6:00 p.m.) – it is clearly seen that the area close to the entrance of the shopping center is the most occupied area. From this analysis, it can be concluded that the entrance to the shopping center is the bottleneck of this scene (from the point of view of ensuring public safety).

In order to achieve better data visualization, the data from the accumulation matrix from one-hour-long video sequences was normalized to the value from the accumulation matrix from highest result of the whole day. This kind of operation has a positive impact on the visualization of data by the user of the system in a more intuitive way. The differences in data visualization are most visible in the first three density maps (Table 4.1; Appendix C). The colors on these density maps are more diversified and the user of the system can easily, distinguish results (maps) with the lowest density of moving people. This solution – data normalization - is not provided by the systems available on the market.

Table 4.1. Examples of generated density maps based on time from the parking lot of the Green Point shopping center with and without data normalization [25]

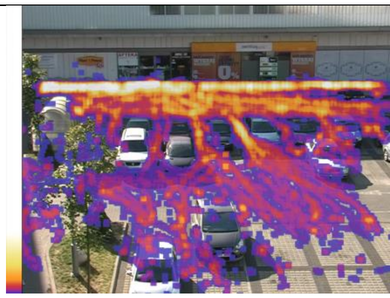
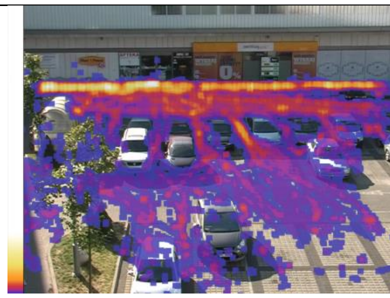
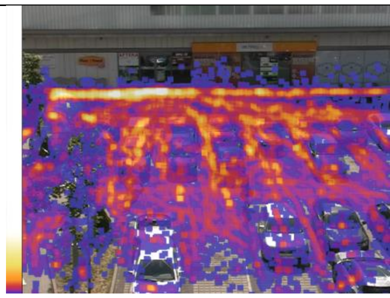
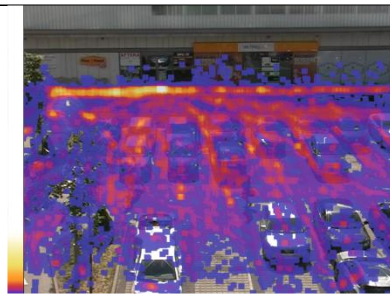
Time of recording	**Density maps based on time without data normalization**	**Density maps based on time with data normalization**
6:00 a.m. – 7:00 a.m.		
8:00 a.m. – 9:00 a.m.		
10:00 a.m. – 11:00 a.m.		

Figure 4.5 shows the density map based on time for Green Point shopping center parking lot for a whole day. The maximum value in this density map before scaling is 18634, which means that relevant people were present in 18643 frames of the video sequence. Taking the frame rate (in fps), it means that people were present there for 745.72 sec i.e. 12.42 min (this value together with some other examples are indicated in Fig. 4.5).

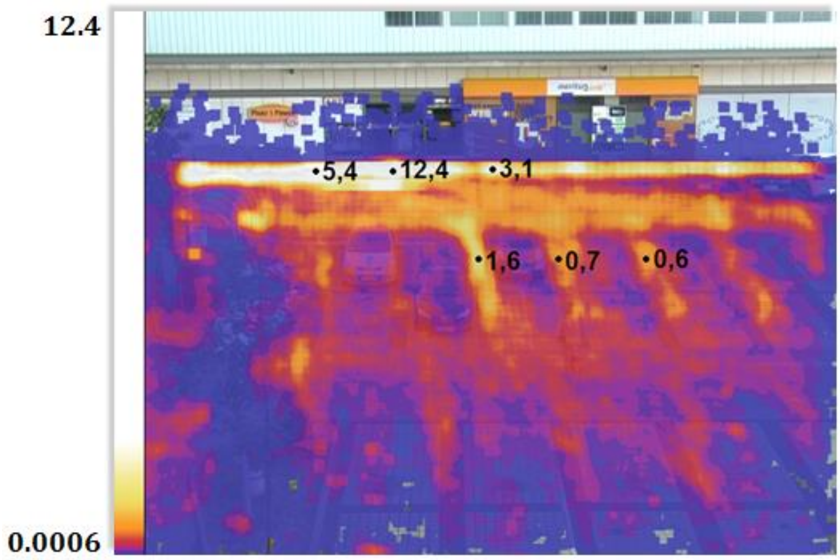


Fig. 4.5. The density map of moving people based on time from the whole day-long video sequence with marked illustrative times of the presence[min] of moving people in the specific area

Normalization and visualization of density maps based on moving objects indexes

Because each relevant moving person is tracked and because only one path for each person is generated, the number of moving people passing along a given path can be determined. Figure 4.6 shows a density map based on the observation of moving-objects indexes from Green Point shopping center parking lot for the whole day – from 6:00 a.m. to 9:00 p.m. with one-hour long intervals between each sequences. The maximum value in this density map before scaling is 871 – which means that according to the density map, there were 871 relevant people in this place. Other example of moving people counting results in the specified area is shown in Fig. 4.6. Selected density maps, based on the observation of moving-objects indexes, from the analysis of one-hour long video sequences, are presented in Table 4.2. All density maps based on moving object indexes are presented in Appendix D.

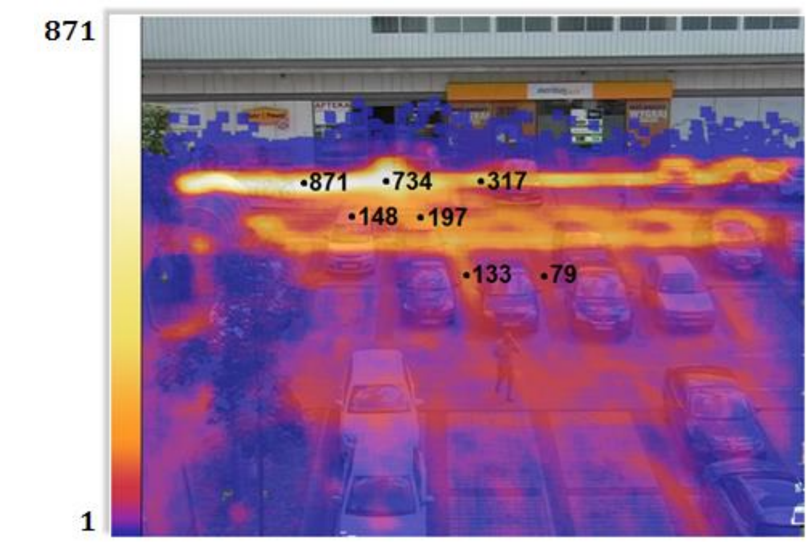
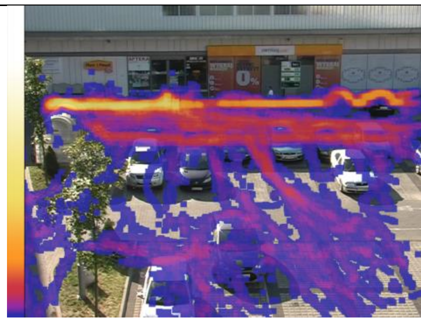
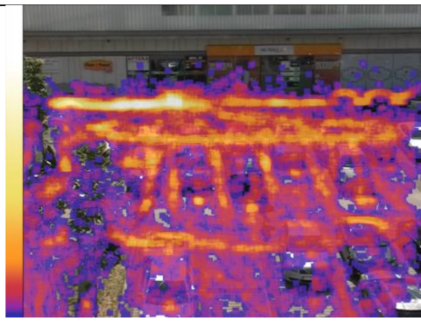
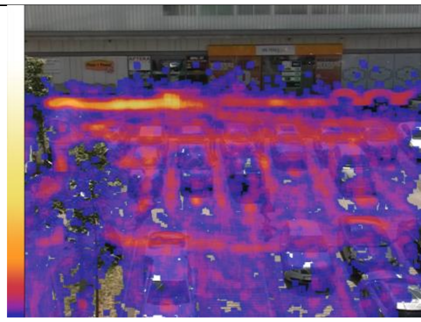


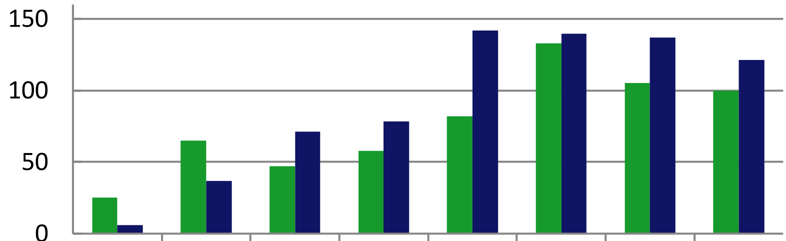
Fig. 4.6. Moving people density map based on the observation of moving objects indexes from a whole day-long video sequence with the marked numbers of moving people in the specific area

Table 4.2. Examples of density maps based on the observation of the indexes of moving people from the parking lot of Green Point shopping center with and without data normalization – part I [25]

Time of recording	**Density maps based on moving object indexes observation without data normalization**	**Density maps based on moving object indexes observation with data normalization**
6:00 a.m. – 7:00 a.m.		
8:00 a.m. – 9:00 a.m.		
10:00 a.m. – 11:00 a.m.		

The analysis of density maps obtained on the basis of subsequent recordings from one location allows to determine the "busy-hours" – the time during which the density of moving people was highest in the whole day. Fig. 4.7 shows the movement distribution changes within each hour and "busy-hour" estimation based on time and observation of moving-objects' indexes at the parking lot in front of Green Point shopping center. The maximum length of time and the largest number of people in a given point (in front of the entrance to the shopping center) was between 4:00-5:00 p.m. During that hour (between 4:00-5:00 p.m.), 133 people were present in front of the entrance to the shopping center over 140 sec - that is 2,3 min.

From the time that people remained at a given point (Fig. 4.5, Appendix C) and the number of moving people passing through a given point (Fig. 4.6, Appendix D) it can be concluded, which point has a specific density corresponding to the density map time. Statistics for a given point (in this situation this is entrance to the shopping center) can be read from a chart (Fig. 4.7).



	6 AM	8 AM	10 AM	12 AM	2 PM	4 PM	6 PM	8 PM
Count [persons]	25	65	47	58	82	133	105	100
Time [s]	5,84	36,92	71,24	78,44	141,92	139,48	137,12	121,52

Fig. 4.7. „Busy-hour" estimation based on time and observation of moving-objects indexes at the parking lot – entrance to the shopping center [25]

4.1.3. Improving clarity by the application of projective transformation

Many CCTV systems use cameras placed in such a way that the resulting recordings show the test area in a perspective view. It is necessary to transform the camera view from 'bird-view' to 'top view'. This transformation can be used in situations, where it is not possible to place a camera at a considerable height or in such a way that the camera observes the scene from top view.

The solution, used in order to achieve better visualization of density map results, is application of the projective transformation of density map image. A proposed solution for different camera angles position has, therefore, been analyzed.

To verify the accuracy of the proposed solutions for projective transformation, two experiments were performed. The first one was carried out on a micro scale with an artificial scene in order to efficiently examine the accuracy of projective transformation. The second experiment was carried out at the parking lot. The data from the side view was transformed to the top perspective view, which is preferred by security and space planning companies.

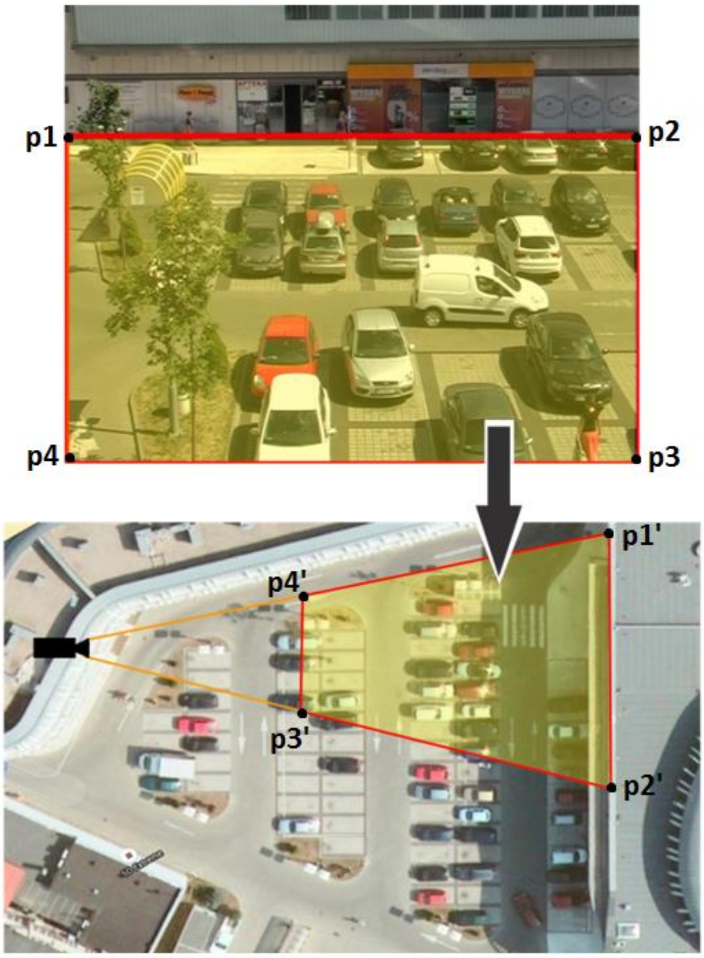
Projective transformation

Projective transformation is a mapping between two planes, e.g. between the image and the world plane. In the experiments, mappings between two images of the same scene (from side and top view) have been found. It can be realized by determination of two projective transformations from one image to the world plane and from the world plane to the second image (Figure 4.8).

Homography is defined as the following transformation on **Y** image [140]:

$$\mathbf{Y}' = \mathbf{HY} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{Y}\,, \tag{4.1}$$

where, for example, h_{11}, h_{12}, h_{21}, h_{22} represents an affine transformation (rotation and scaling) and h_{13}, h_{23} represents a translation and h_{31}, h_{32} represents two lines at infinity. These information allow to observe vanishing points, horizon in the image and allows for 8 degree of freedom (2 scale, 2 rotation, 2 translation, 2 line at infinity).



(image from GoogleMaps)

Fig. 4.8. Visualization of projective transformation from the side view ($\mathbf{Y}$) to the top view ($\mathbf{Y}'$) [141], the area to be transformed was marked together with transformation points; the transformed image is shown together with camera location and camera field-of-view

An example of visualization of projective transformation from the side view to the top view on the parking lot is shown in Fig. 4.8. In this figure the area that will be transformed and its equivalent after transformation has been marked.

Significant points in the original (non-transformed) image (side view, $\mathbf{Y}$; Fig. 4.8):

- top left: $\mathrm{p1} = (1, 164)$
- top right: $\mathrm{p2} = (720, 164)$
- bottom right: $\mathrm{p3} = (720, 576)$
- bottom left: $\mathrm{p4} = (1, 576)$.

Adequate points in the transformed image (top view, $\mathbf{Y}'$; Fig. 4.8):

- $\mathrm{p1}' = (739, 1)$
- $\mathrm{p2}' = (739, 668)$
- $\mathrm{p3}' = (1, 468)$
- $\mathrm{p4}' = (1, 163)$.

Only one transformation matrix has to be computed to map between two images of the same scene. Based on the above points from the non-transformed and transformed images (Fig. 4.8), the $\mathbf{H}$ matrix can be designated:

$$\mathbf{H} = \begin{bmatrix} 4.69 \cdot 10^{-17} & -3.39 & 1956.8 \\ 1.7585 & 1.6354 & -268.07 \\ 1.27 \cdot 10^{-19} & 5.46 \cdot 10^{-3} & 1 \end{bmatrix}.$$

Evaluation of projective transformation on the image from an artificial scene

The goal of the experiment was to show how accurate projective transformation applied to density map is, and the testing environment with two cameras (Microsoft LifeCam Studio) looking at the scene from two different angles (Fig. 4.9) was, therefore, prepared. The angle of the first camera was modified across multiple experiments from 30 degrees to 50 degrees (α angle in Fig. 4.9) with an interval of 5 degrees. The second camera was positioned in a straight downward direction (the top view). Little hexbug robots were used in the experiment as moving objects in the scene. The result of the experiment is that data of the same scene and movement are recorded simultaneously from two different views.

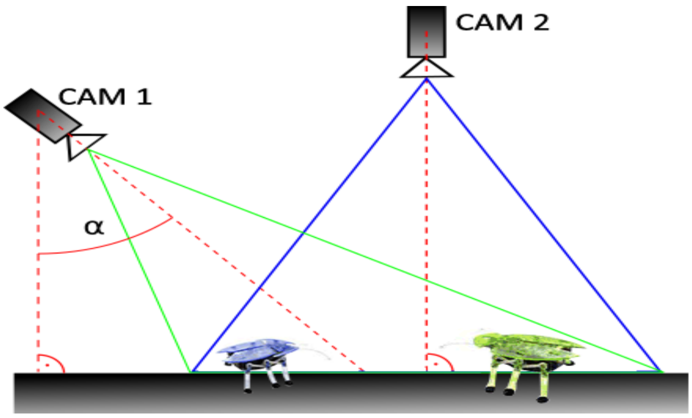


Fig. 4.9. Cameras' positioning during the experiment [141]

In this part of the experiment, only examples from the second type of density maps – based on the observation of moving-objects indexes – were shown. The accuracy of projective transformation of the density map was measured and compared using the correlation of the measure of similarity between the transformed density map recorded from the side view and the non-transformed density map recorded from the top view (Fig. 4.10).

At first, the correlation coefficient between whole-marked regions was computed. The results are presented in Table 4.3. The overall values of the measures dropped from 0.792 to 0.671 across the ascending angle of the side camera, which demonstrates the difficulty of view transformation, when the image is taken at a large angle. In the subsequent step, each density map was divided into four equal sectors and compared respectively (see Table 4.4).

Additionally, to show a distortion, cross correlation between sectors and the whole region was computed and the best fitting regions were founded. Figure 4.11 and Table 4.4 show that the best fitting sectors are shifted. The values of the correlation coefficient of the best-fitted sectors and the whole region can be found in Table 4.4.

Table 4.3. Correlation coefficients between transformed density maps (and their sectors) recorded from the side view and non-transformed density maps recorded from the top view [141]

Angle (degrees)	Density map area				
	Sect. 1	Sect. 2	Sect. 3	Sect. 4	Whole region
30	0.645	0.767	0.786	0.883	0.792
35	0.711	0.855	0.723	0.814	0.795
40	0.802	0.725	0.739	0.804	0.767
45	0.686	0.624	0.766	0.682	0.672
50	0.674	0.680	0.683	0.692	0.671

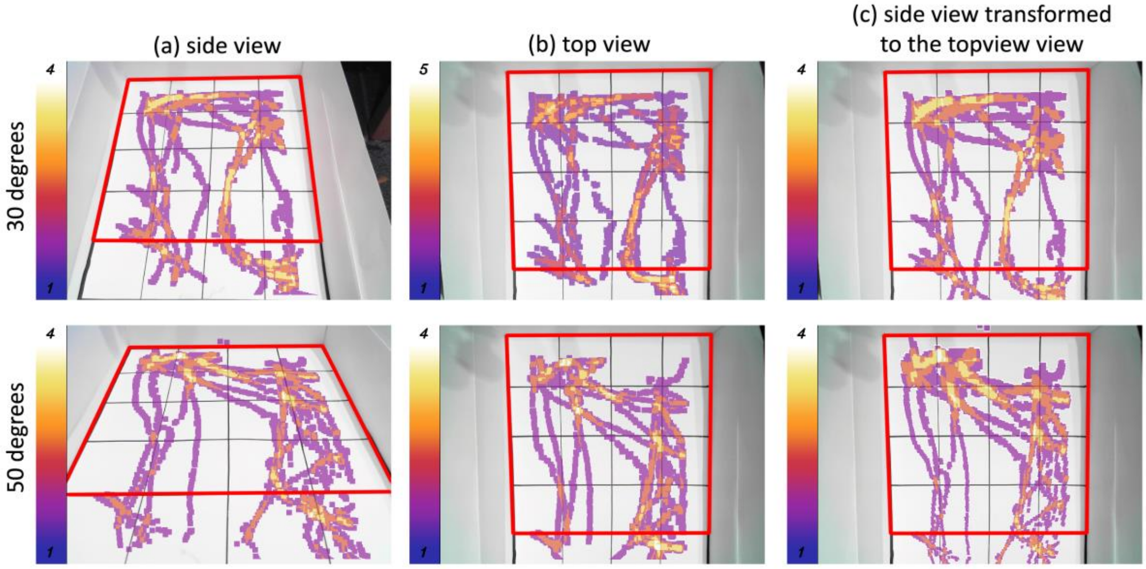


Fig. 4.10. Density maps from camera no 1 (a) side view positioned at an angle of 30 and 50 degrees; density maps from camera no 2 - (b) top view; density maps from the side view transformed to the top view (c) with the use of projective transformation [141]

Table 4.4. Selected cross correlation coefficients and offsets between transformed density maps (and their sectors) recorded from side view and non-transformed density map recorded from top view [141]

Angle (degrees)	Density map area	Cross correlation coefficient	Horizontal offset [px] (X axis)	Vertical offset [px] (Y axis)
30	Sect. 1	0.768	5	9
	Sect. 2	0.791	3	-1
	Sect. 3	0.798	0	4
	Sect. 4	0.889	-2	1
	Whole region	0.800	2	2
35	Whole region	0.825	0	6
40	Whole region	0.806	2	8
45	Whole region	0.720	2	7
50	Sect. 1	0.810	-5	14
	Sect. 2	0.825	-6	12
	Sect. 3	0.744	3	12
	Sect. 4	0.853	4	17
	Whole region	0.777	0	14

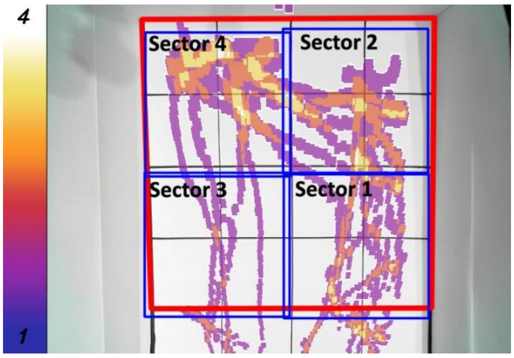


Fig. 4.11. Example of density map sectors offset with highest correlation [141]

The coordinate system for density map images and the individual sector of density maps is defined in the upper left corner. Table 4.4 shows that between the transformed density map (and their sectors) recorded from the side view and the non-transformed density map recorded from the top view there is a vertical offset (Fig. 4.12) and a horizontal offset, which is insignificant. The average vertical offsets in the sectors from 1 to 4 for all α angles are, respectively, 11.2 (for sector 1), 7.2 (for sector 2), 9.6 (for sector 3) and 6.4 (for sector 1). This underlines the fact that the areas, which are closer to the camera (sectors 1 and 3) have a greater offset than the regions, which are further from the camera (sectors 2 and 4). From Fig. 4.12, it can also be seen that the vertical offset value increases with a rise of the α angle (side view, camera no. 1, Fig. 4.10), but for the video sequence resolution of 640×480 pixels, this offset can be omitted, because it is within 2.9% of error.

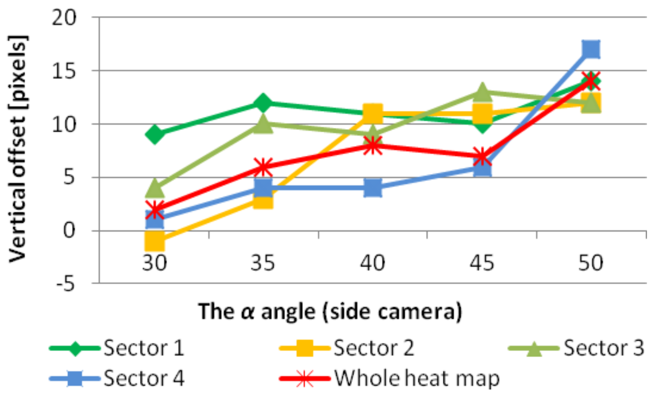
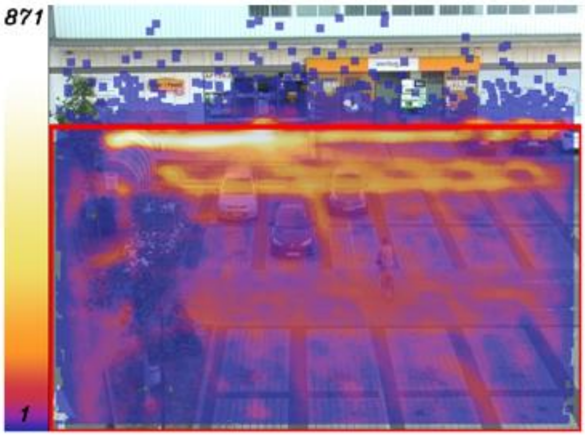


Fig. 4.12. Vertical offset between transformed density maps (and their sectors) recorded from side view and non-transformed density maps recorded from top view [141]

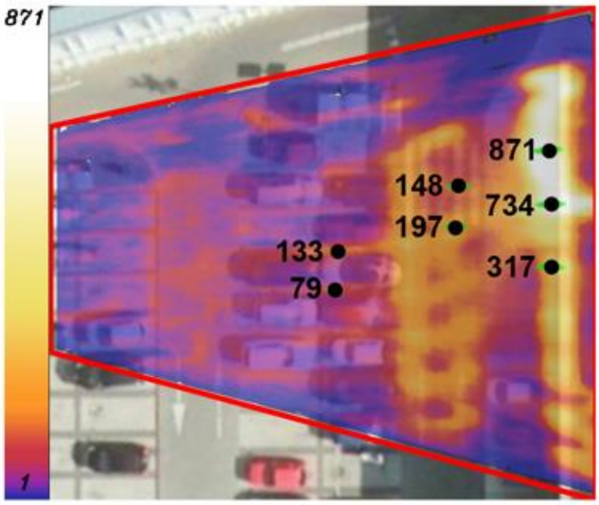
Application of projective transformation on the image from the real scene

The next experiment deals with the generation of density maps from CCTV data. Figure 4.13 (a) shows an illustrative image from video sequences used during tests for the estimation of the density of moving objects. The recordings were carried out from positions opposite to the shopping center under real-life conditions (name of video sequences set: SVDMG).

Figure 4.13 (a) shows density maps based on observations of moving objects indexes from the shopping center parking lot for the whole day. The next step is to transform the density map to show data from the top view. The results of the projective transformation can be seen in Fig. 4.13 (b).



(a)



(b)

Fig. 4.13. Density map of moving-objects based on the observation of moving-objects' indexes from the whole-day long video sequence; side view (a) of the scene and transformed top view (b) with the marked numbers of moving objects (originally shown in Fig. 3.6) [141]

In this Section, two ways of improving the density map of moving people are shown.

The first, data normalization to the highest result from the whole-day analysis was presented. This kind of improvement helps to determine, more accurately, the "busy-hour" for the analyzed space or specific localization in an intuitive way. This method can be assisted using a graph, generated from data assigned to the density maps from specific location (shown in Fig. 4.7).

The second solution is assisted by projective transformation and can be facilitated when there is no possibility of using the camera just above the observed scene. It is noted that this is a precise solution for density map of moving people visualization.

4.2. Bi-directional people counting

The next model of macro-biometric system is the model for people counting. This Section presents an analysis of a new algorithm for counting people in video surveillance systems. Gdzie jestem 4. 14.

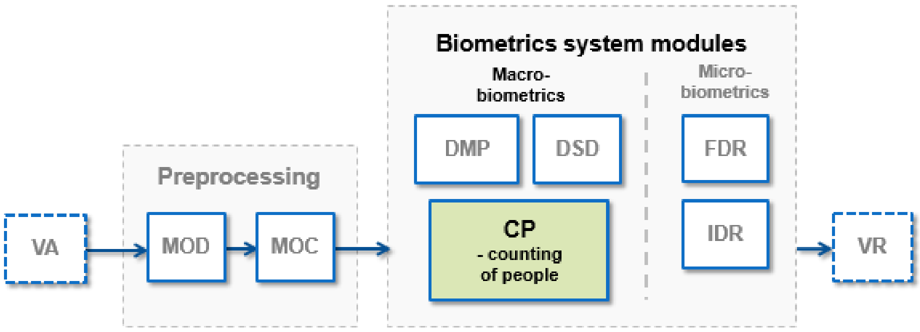


Fig. 4.14. Individual issue presented in the dissertation – bi-directional people counting

Previous author's studies dealt with the problems of real-time people counting by monitoring the entrance to the building (indoor scenes). These systems determine the number of people crossing the gate. This research was presented in [26, 61] publications. In the [61] paper, video processing operations for people counting have been realized in the Matlab/Simulink environment with the use of "People Tracking" model, which was modified and then implemented in bidirectional people counting system. In another [26] paper, a model described in [61] was implemented on the Raspberry Pi platform [131]. The developed device performs all calculations directly on the input data and transmits the video signal together with the respective metadata to the display. This makes the entire system a stand-alone.

As a consequence of previous researches and analyses of the state-of-the-art, a system for bi-directional people counting, based on data recorded in outdoor scenes is now realized. During the process of people counting (Fig. 4.15), the camera is positioned vertically downward and the observation of the center of the position of moving objects (in the binary image) is done. When a pedestrian passes through the virtual line, the adequate counter value is incremented.

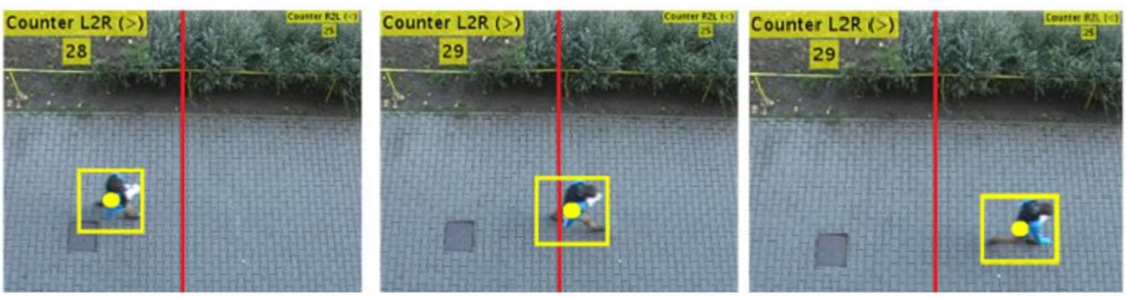


Fig. 4.15. The camera localization and the people counting example

In this thesis, an additional classifier was proposed. This classifier helps to determine the number of people in a single BLOB, when people are walking close to each other. The classifier was described in Section 3.3. Moreover, the classifier helps in the case of small objects (such as dogs or birds) – in this situations, the algorithm should not increase the counter results. Mentioned small object are also not detected by optimized movement detection algorithm.

4.2.1. Application of standard people counting model

Tested systems for bi-directional people counting are based on data recorded by a camera positioned vertically downward and above the observed scene (Fig. 4.16). The minimum height at which the camera should be mounted is 3.9 m according to [76]. The higher the camera, the greater the accuracy of the algorithm and, consequently, the lesser will be the impact of the difference in the height of the people. However, it is important to refer to PN-EN 50132-7 norm (described in Appendix A). According to which, the object in motion should occupy at least 10% of the height of the image in order to be correctly detected.

People counting model consist of the following steps.

At first, sequences of video frames are acquired (in the example) with resolution of 720×576 pixels processed at 25 fps rate.

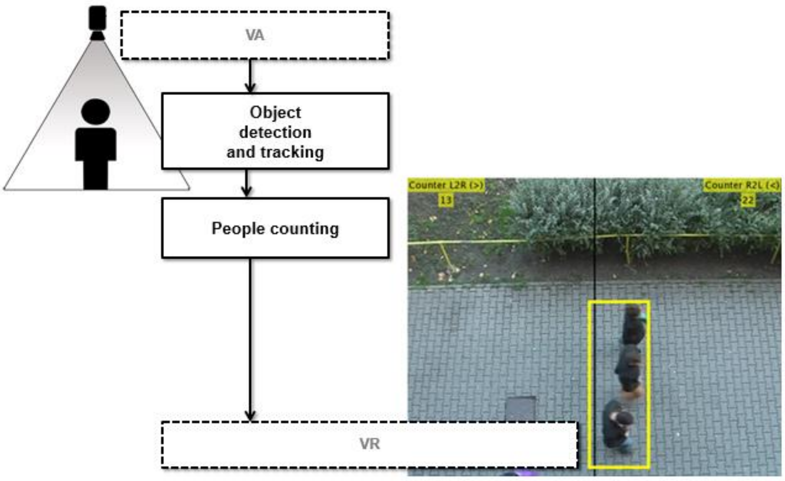


Fig. 4.16. Scheme of the standard system for people counting offered commercially

Then, moving objects were detected using the proposed and modified in this dissertation algorithm – the GMM. Thank that people counting system can be used for outdoor scenes. The motion track of each object was estimated and predicted in each video sequence frame. The system output is a set of temporal and spatial coordinates of each recognized object. In the example with the use of standard algorithm for people counting [26, 61], the moving object detection algorithm is based on the unimproved GMM (described in Section 3.2).

The next step is, dividing the image into two parts (left and right) with the use of a virtual line and determining the position of the object – an attribute is assigned to every moving object visible in the scene, determining whether the object is located on the left or on the right hand side of the frame. This division is done according to the virtual line, placed on the ground at any location of the image. Depending on a suitable camera position, it can be easy to determine how many people passed from the left to the right hand side, as well as in the opposite direction. The *Position* attribute is assigned to every moving object visible in the scene, determining whether the object is located on the left or the right hand side of the frame. Binary values of the *Position* of objects are assigned, but "0" value represents objects' positions on the left hand side of the video frame and "1" value represents objects' positions on the right hand side of the video frame. One particular condition is most important in these calculations: if the object was first detected on the left hand side and then, later, on the right hand side of the image, it means that the object is moving from the left to the right. Otherwise, the movement of the object is in the opposite direction.

The next operation is the incrementation of counters. An adequate counter is incremented, when the object has just crossed the virtual line placed vertically along the video frame (typically at the center of the image).

Finally, the results are displayed. The analyzed images are displayed together with additional metadata (algorithm marks the detected moving objects using rectangles to surround the BLOBs) and the results of the counters. The upper left corner of the output image shows the number of people that moved from the left to the right side (marked as "L2R") and the right upper corner shows the number of people who moved in the opposite direction (marked as "R2L").

During the examination of people-counting system, the set of video sequences described in Section 3.1 was used (named TVODCC). Let the reader be reminded that the set contains four hours of video sequences recorded under normal weather conditions and an hour of recording in strong rain.

As an example of the application of a standard people-counting model, video sequences named "1 p.m." and "4 p.m." were presented in the Figures. Figure 4.17 shows real value of counters in consecutive video sequence frames. Counter values in these plots were counted manually. For example, "1 p.m." video sequence contains 29 people passing under the camera from the left to the right side (of the image) and 24 people passing in the opposite direction. The total number of people was 53.

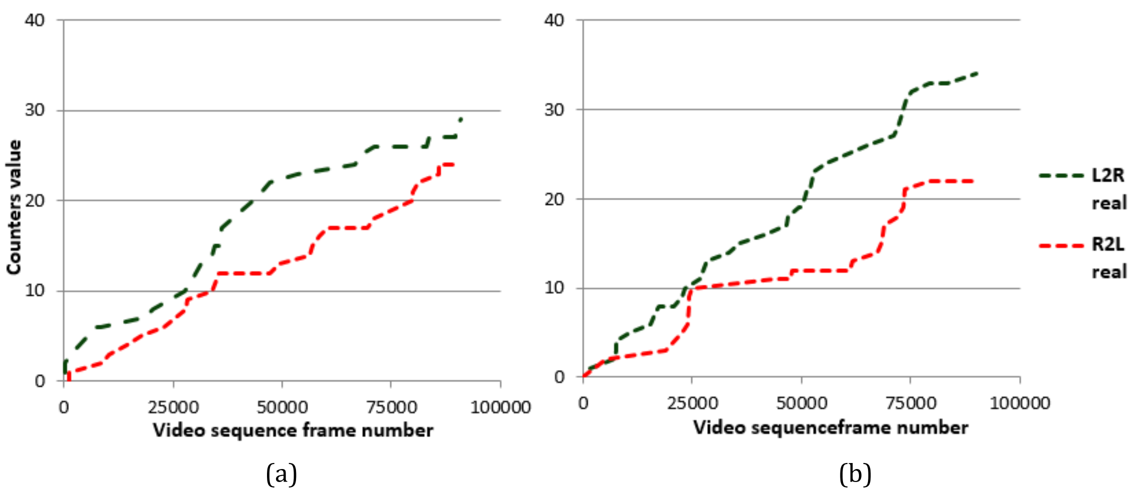


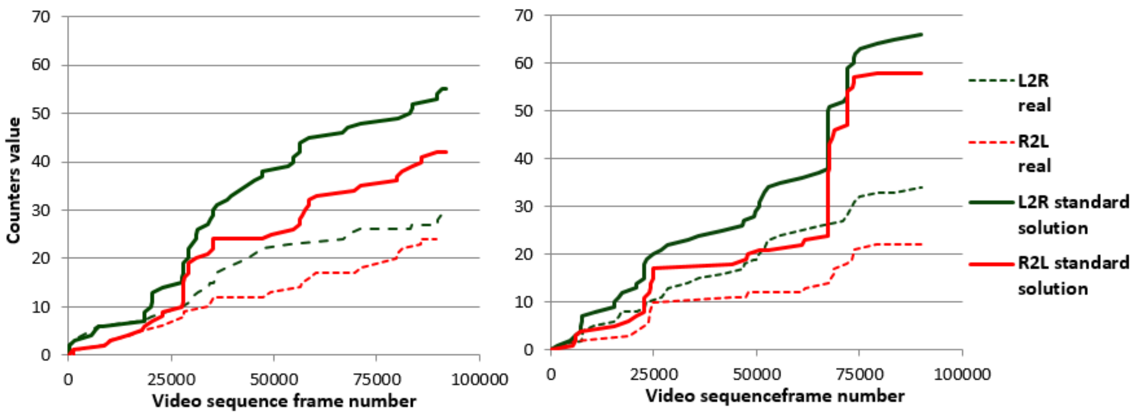
Fig. 4.17. Real, manually counted counter values in consecutive video frames on the basis of a one hour long video sequence. The names of the video sequences are 1 p.m. (a) and 4 p.m. (b) respectively, where: L2R is number of people passing from the left to the right side, and R2L is number of people passing from the right to the left side

A standard people-counting model was applied on selected video sequences. Figure 4.18 shows automatically counted results of standard model for people counting without optimization. Due to defects in the operation of standard people counting algorithm, the counters are overestimated. In the example of “1 p.m.” video sequence, the computed results are about 1.9 times higher than the actual results.

This incorrect results of standard people-counting model is caused by the following facts:

- the moving objects detection algorithm is not properly matched to the observed scene (indoor vs. outdoor scenes)
- the moving objects detection algorithm parameters are not improved upon, so the noise and other small objects are detected and affect the values of the counters
- the same objects are repeatedly counted near the virtual line
- the moving objects are not properly separated from each other. In this situation, for example, three persons will be counted as one.

This is the major disadvantages of standard algorithm for people counting and the solution for this situations is described in the next Section.



(a) (b)

Fig. 4.18. Real manually counted counter-values and automatically computed counter-values by standard algorithm (without modifications) for people counting in consecutive video frames on the basis of one-hour long video sequence; the names of video sequences are 1 p.m. (a) and 4 p.m. (b), where L2R is number of people passing from the left to the right side, and R2L is number of people passing from the right to the left side

4.2.2. Application of additional classifier in the BLOB prediction

In this Section, the proposed, modified and highly efficient model for people-counting is described. Three stages of the optimization were performed. A schema of the improved people-counting model is shown in Fig. 4.19.

The first stage of the optimization was selecting and choosing the parameters of the motion detection model. This step is briefly described in Section 3.2.

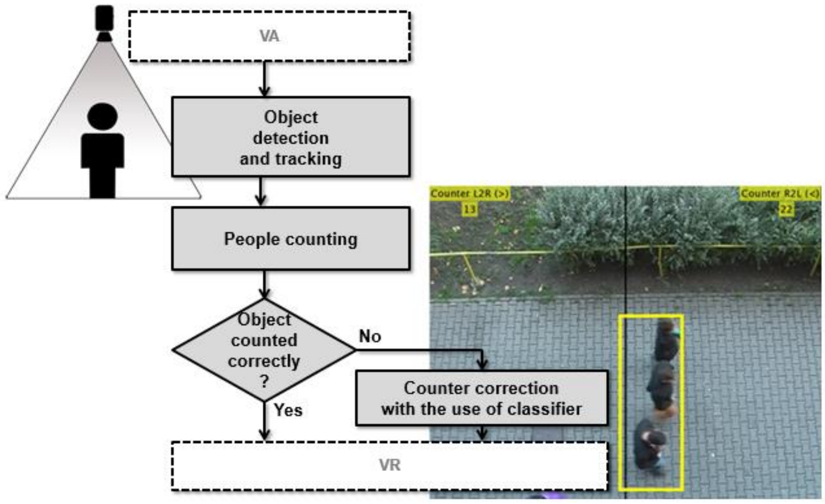


Fig. 4.19. Schema of the algorithm for people counting with modifications (modifications are marked on the gray color)

In the second stage of the optimization of the model for people counting, the problem of multiple counting for the same objects was solved. Figure 4.20 shows a situation where one object was counted a few times in consecutive video frames. In this example, the direction of the person's movement is also marked in order to better understand the counter results. This situation has occurred because of the changing of the object BLOB shape and its center in each of the consecutive frames of the video sequence. Thus, the object was detected in different parts of the image, which led to several increments of the corresponding counters. This disadvantage is eliminated in this PhD dissertation by using appropriate counters reaction delay. On the basis of video sequence observation, the time interval for the counter increments has been set to 0.16 s. The algorithm compares the position of the object from the current video frame with the position of the object from the previous video frame with an interval of four video frames.

The third stage of algorithm modification was counter correction. In order to count the objects correctly, the BLOBs should be classified in relation to the number of objects they contain. This is done with the naïve Bayes classifier. This classifier uses a multidimensional Gauss function [57]. It is easy to implement and computationally efficient. The details about processing time calculations are shown in Section 4.2.3.

This classifier type (number of objects in the single BLOB prediction) was described in Section 3.3. If the classifier predicts *n* persons in a single BLOB, the actual counter value is incremented by *n.* If the algorithm detects, for instance, a large dog, the assumption is that it will not be counted, because in the further step of the algorithm, the object dimensions using the naïve Bayes classifier, are checked and this object will be rejected from the counting process.

During the training of the classifier, BLOB height, width, and the area, all have to be analyzed. The tests showed that the best classifier effectiveness was obtained using BLOB width and height. This part of the people-counting algorithm, i.e. number of objects in the BLOB prediction, is briefly described in the Section 3.3.

Object's movement direction:

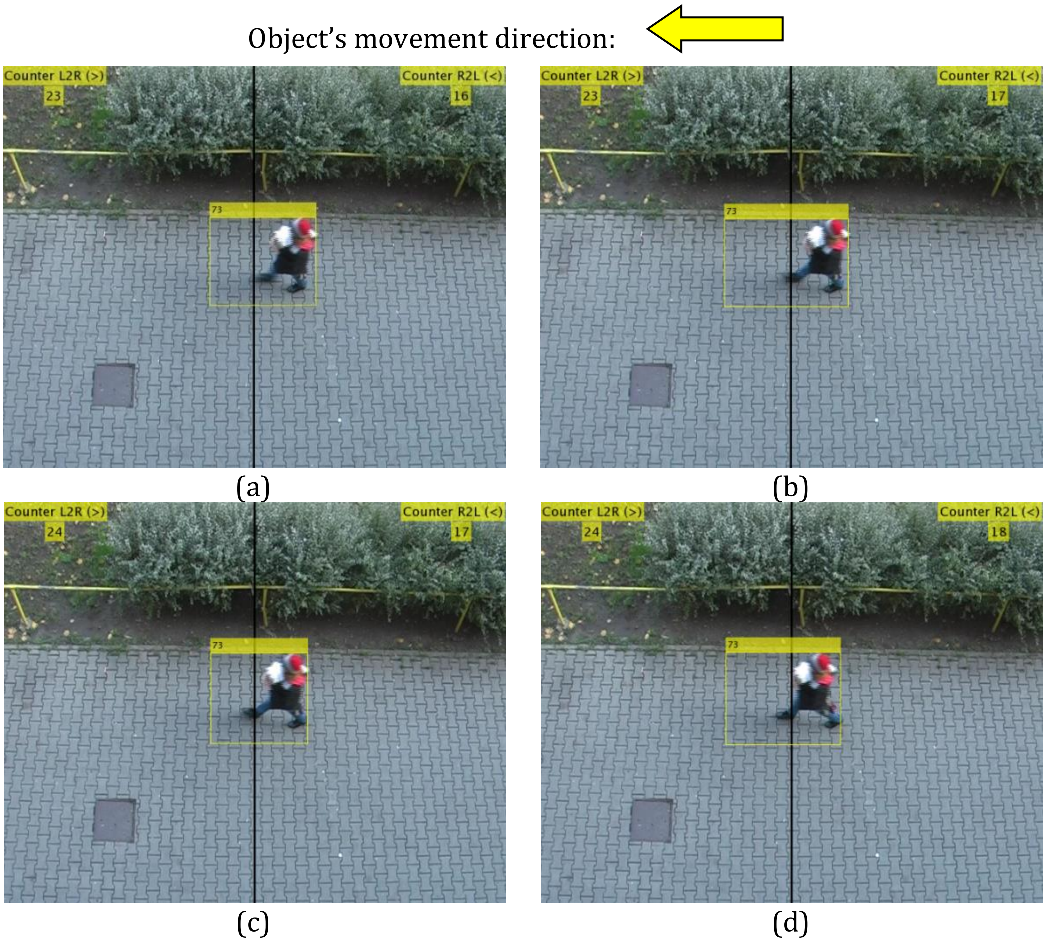


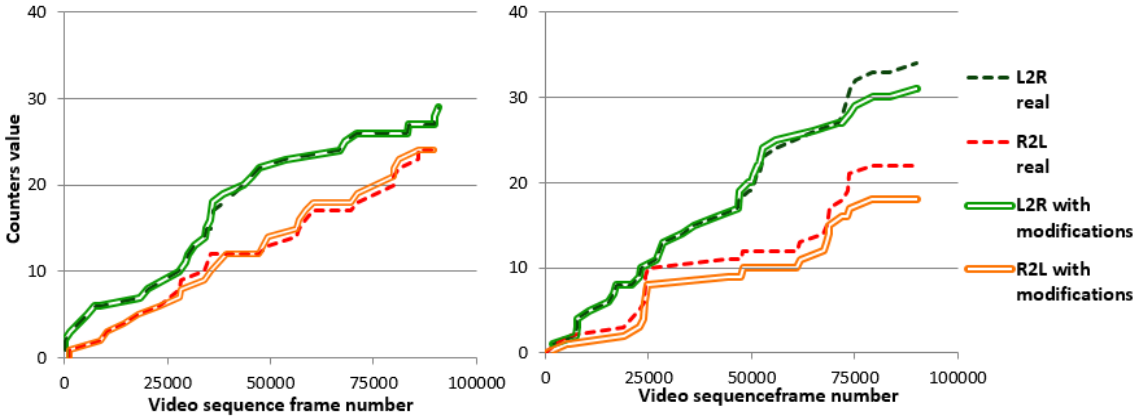
Fig. 4.20. Examples of incorrect counter incrementation in a bi-directional people counting system before algorithm modification [27]

In Figure 4.21 the results of the people-counting experiment for the one-hour long video sequence recorded at 1 p.m. and 4 p.m. is shown. The plot illustrates the results of counters for counting people in both directions ("L2R" and "R2L"). It demonstrated how well the improved algorithm worked. The differences between the actual numbers of people and those detected with the improved algorithm are so small that the respective plots nearly overlap. For example, for the "1 p.m." and "4 p.m." video sequences, the MSE (mean squared error) between the two curves (real counters values and counters values after algorithm modification) for both passing directions are:

- for "1 p.m." video sequence: $\mathrm{MSE_{L2R}} = 1.73$, $\mathrm{MSE_{R2L}} = 4.69$
- for "4 p.m." video sequence: $\mathrm{MSE_{L2R}} = 7.93$, $\mathrm{MSE_{R2L}} = 11.31$,

where :
$\mathrm{MSE_{L2R}}$ – MSE between curves for people passing from left side to the right side
$\mathrm{MSE_{R2L}}$ – MSE between curves for people passing from right side to the left side.

It is clearly shown (Fig. 4.21) that the counting quality is substantially improved in relation to the non-modified standard solution presented in Fig. 4.18.



(a) (b)

Fig. 4.21. Real manually counted counters values and modified algorithm for people counting in consecutive video frames based on a one-hour long video sequence. The names of the video sequences are 1 p.m. (a) and 4 p.m. (b),respectively. Where: L2R is number of people passing from the left to the right side, and R2L is number of people passing from the right to the left side

4.2.3. Efficiency tests of the proposed solution

In this Section, all the results obtained during the examination of the people-counting system are presented. During the entire research, 5 one-hour long video sequences were examined. The parameters of the algorithm for the detection and tracking of moving object were selected and applied in the model, in order to obtain better results with respect to the effectiveness of bi-directional people counting. Moreover, the operation of the counters incrementation was improved by not incrementing the counters several times for the same tracked objects. At the end, the correction of counters using naïve Bayes classifier was added.

The algorithm efficiency statistics can be calculated with the use of true positives, false positives, true negatives and false negatives events [44, 45]. A true positive value (equivalent to the hit) is the number of correctly counted people in both directions. A false positive value (equivalent to the false alarm) is the number of false detections of moving objects, such as animals (dogs) and other moving objects which should be ommited. A true negative value (equivalent to the reject) in the counting system, is the number of objects that should be correctly rejected. For example, a person who did not fit in the camera frame (e.g., less than 50 % of a person was visible), animals (birds or dogs), big raindrops or bushes swaying in the wind (visible at the top of the scene presented in the Figures) should be rejected. A false negative value (equivalent to the miss) is the number of people who were not detected and counted by the algorithm, e.g. large BLOBs that contained two or more people.

It is important to note that the efficiency statistics of people-counting system may be checked only during instants, when the objects are passing through the virtual line on the ground.

The results of this tests are shown in Table 4.5. An average sensitivity of the people-counting algorithm under fine weather conditions was equal to 0.9, the precision under the same conditions was 0.96, the miss ratio was about 0.1 and the average accuracy was 0.89. The overall average accuracy (i.e. including the cases of bad weather – e.g. during rain is taken into account) was only slightly worse and equaled 0.87.

The effectiveness of the algorithm, defined as a ratio of the number of people detected by the algorithm to the actual number of people, who passed under the camera, was also calculated. Its overall average value, i.e. computed for all recorded video sequences in good and bad weather was 1.02. The average ratio for the good weather conditions was 0.94, i.e., the real results were underestimated and for the rain it was 1.33, i.e. the real results were overestimated.

Table 4.5. Effectiveness of bi-directional people-counting algorithm after optimization, where, * actual, ** calculated and *** calculated to actual number of people ratio [28]

	Video sequence (name)						
Training	**1 p.m., 2 p.m., 3 p.m.**	**2 p.m., 3 p.m., 4 p.m.**	**3 p.m., 4 p.m., 1 p.m.**	**4 p.m., 1 p.m., 2 p.m.**	**1 p.m., 2 p.m., 3 p.m., 4 p.m.**	**Sum or average without rain**	**Sum or average with rain**
Test	**1 p.m.**	**2 p.m.**	**3 p.m.**	**4 p.m.**	**RAIN**		
TP	52	47	47	48	36	194	230
FP	1	4	2	1	12	8	20
TN	13	13	10	10	12	46	58
FN	3	0	12	8	0	23	23
TPR	0.95	1.00	0.80	0.86	1.00	0.90	0.92
PPV	0.98	0.92	0.96	0.98	0.75	0.96	0.92
FNV	0.05	0.00	0.20	0.14	0.00	0.10	0.08
ACC	0.94	0.94	0.80	0.87	0.8	0.89	0.87
*	53	52	59	56	36	220	256
**	53	55	49	49	48	206	254
***	1.00	1.05	0.83	0.88	1.33	0.94	1.02

Figure 4.22 and Table 4.6 show the efficiency of the classifier, according to the actual number of people in the BLOB. The weights assigned to the BLOB are also listed in Table 4.6. In order to visualize the results better, the weights were aggregated from 5 values (Section 3.2) to just 3, as the weights for very low (weight no. 1) and low (weight no. 2) detection qualities were combined together, as well as those for high assessments (weight no. 4) and very high (weight no. 5). The chart in Fig. 4.22 takes the number of the examined objects into account. In this chart, the third class (assigned for three objects in the BLOB) was omitted due to a lack of data. From Table 4.6, it can be concluded that the highest efficiency of the naïve Bayes classifier was obtained for objects with the following assessments of detection quality: high and very high (weights no. 4 and 5), respectively.

It is worth noting that the current classifier is built for the system with cameras mounted at a the specified height. In the future, the author intends to expand the system with classifiers, which will be appropriate for cameras placed at various heights.

Table 4.6. Average efficiency of the naïve Bayes classifier according to actual number of persons in the BLOB and weights assigned to the BLOB [27]

		Weight assigned to the BLOB		
		1 and 2	**3**	**4 and 5**
Number of persons in the BLOB (BLOB class)	**0**	64.44	56.25	72.92
	1	53.28	82.27	99.71
	2	0	56.93	57.53
	3	no data	no data	0

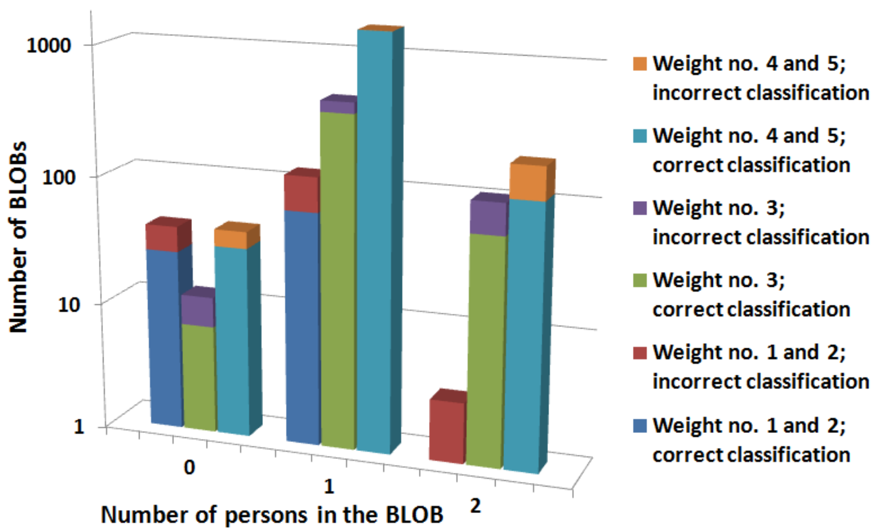


Fig. 4.22. Average efficiency of the naïve Bayes classifier, taking into account the number of examined BLOBs [27]

The processing times of the algorithm for bi-directional people counting are shown in Table 4.7. The processing times of *People counting* function without classifier and with classifier were measured. The difference between the average processing time of *People counting* function (without and with classifier) is 18.4 ms for one video frame. Differences in processing times are also visible in case of different number of BLOBs in the scene. If only one BLOB was present in the scene, the difference between execution times of both *Count* function is 10.3 ms. In case of the presence of two BLOBs in the scene that time is two times longer (21.9 ms). The average execution time of the entire program for one video sequence frame displaying results with and without classifier was 63.7 ms. The tests were performed at Intel (R) Core™ i7-2620M CPU @ 2.7GHz.

From Table 4.7, it can be calculated, that the processing time of one second of video sequence (with a resolution of 720×576, processed at 25 fps) is 1.59 times slower than real-time processing. There are two simple methods to eliminate this disadvantage. In the first approach, it is assumed that every second video sequence frame will be omitted. This solution will not affect the correct operation of the algorithm, because, currently, video sequence is processed with 25 fps. This solution accelerates the model for bi-directional people counting. The second solution is to accelerate the algorithm in downsampling the resolution of video frame. The previous research (presented in [26]) shows that the algorithm for people counting can effectively detect moving objects in the video sequence with a resolution of up to 160×128 pixels. This resolution is sufficient for proper algorithm operation. In the current system for people counting, the first solution is easier to implement, due the fact that the second solution requires changes to the classifier model and the basic parameters of algorithm in order to adjust to smaller resolutions of video frames.

In conclusion, the optimizations in standard algorithm for people counting were applied. For example, this classifier estimates the number of people in the specified BLOB, performs adequate counter correction and parameter optimization of algorithm for moving objects detection. These optimizations have a positive impact on the efficiency of counting and this improvement of quality was shown in the results.

Table 4.7. The algorithm processing time for one video sequence frame for *Count* function and the whole algorithm [28]

Frame index	No. of BLOBs in the video sequence frame	Processing time for *Count* function without classifier [ms]	Processing time for *Count* function with classifier [ms]	Execution time of entire program with classifier and without displaying results [ms]
240	2	1.24	10.49	57.43
285	2	0.78	19.40	64.95
1295	2	0.79	19.09	66.92
1515	1	0.69	10.47	60.26
5495	2	0.92	29.47	67.77
5500	2	0.74	42.22	70.12
7380	1	0.81	10.26	59.24
8615	2	0.97	19.29	67.67
8630	2	0.96	19.85	67.63
10325	1	1.04	12.71	55.72
	Average:	**0.89**	**19.33**	**63.77**

4.3. Detection of dangerous situations

The next model of macro-biometrics modules are dangerous situation detection (Fig. 4.23).

It is important to note, that the selection of algorithms should strictly depend on camera location and the type of the scene to be observed. Depending on what the camera observes, there are different events. Various algorithms for threat detection were presented in Section 2.2.3.

In this dissertation people are observed at the pedestrian crossing. The algorithm for the observation of people at the pedestrian crossing [15, 35, 16] was developed and described in the following public deliverables [19, 22].

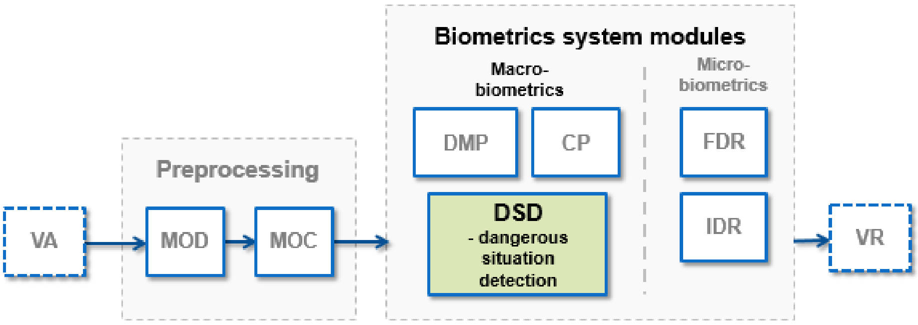


Fig. 4.23. Individual issue presented in the dissertation – dangerous situation detection

The versions of the algorithm (for the detection of people at the pedestrian crossing during red lights) presented in the above mentioned deliverables are different from the current version – object movement process in urban areas was improved (see Section 3.2). The most important features of an algorithm are: operation in real-time (720×576 pixels resolution was processed on Intel Core i7 CPU; 2,93 GHz) and the fact that the alarm function is realized during the event.

Proposed solution for threat detection at the pedestrian crossing

The schema of the tested algorithm for people detection at the pedestrian crossing at the red light is shown in Fig. 4.24. The pedestrian crossing observation algorithm consists of two parts: red light detection and a person at the area of the pedestrian-crossing detection.

A very important issue is accurate camera placement. The camera was situated at height of 2.0–2.5 m over the floor level. An advantage of such a position of the camera is the fact that the hood of the traffic lights does not block out the view. A drawback, however, is that the pedestrians can shade the lights with their bodies causing a failure of the algorithm. Although, it has been empirically proven that this is an extremely rare situation (it has not happened in any conducted experiments).

The algorithm for the traffic-lights detection is not sensitive to the light intensity or the weather (rain, strong sun etc.) changes, because the algorithm uses HSV color palette. Thanks to that, the detection of the colors of the traffic lights is very effective. Therefore, when the green light is on, the algorithm never reports false alarms of threat detection - traffic lights colors are always properly detected (Fig. 4.25). When the green light is on (and both lights are not lit) while people are going across the pedestrian pass, the algorithm does not raise an alarm due to the fact that the algorithm only verifies the presence of red light in the examined area of traffic lights.

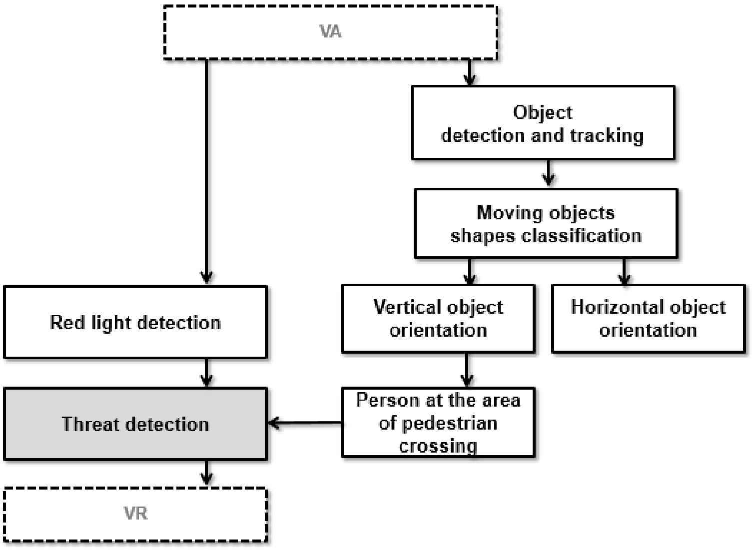


Fig. 4.24. Pedestrian crossing observation algorithm schema

Fig. 4.25. People passing at the pedestrian crossing during the green light

The second part of the algorithm is the detection of a person at the area of pedestrian crossing. Detection of moving objects is done with the use of algorithms based on optimized GMM (see Section 3.2). Thanks to an adequate choice of GMM parameters, large moving objects (like tram) are not detected. Additional classification distinguished between types of objects.

Because of the fact that the algorithm checks the presence of the bottom part of the moving objects at the area of pedestrian crossing, so people passing near the camera do not raise an alarm in case of red light detection (Fig. 4.26) – only people on the pedestrian crossing in risky situations are detected. Unfortunately, sometimes, the algorithm does not properly detect people moving close to each other as separate objects (Figures 4.25). In the future it is planned to add a classifier, that will separate the objects from each other considering that the camera is observing the scene from the side view.

As designated by the algorithm, the shapes of cars (visible form the side) are in a horizontal orientation (see Section 2.3). Hence, cars passing the pedestrian crossing do not raise an alarm even though the red light on the traffic lights was on (Fig. 4.27). Only objects crossing in a vertical orientation at the pedestrian, like persons or cyclists, during the red light can be treated as a cause of danger.

Incorrect object detections were reported in situations, where moving objects (persons) walked very quickly and stopped suddenly in front of the pedestrian crossing (because the red light was on). The persons' predicted positions (by Kalman filter) and estimated locations in the next video frame raised an alarm. This is because the locations of the persons in the following frame were directly at the pedestrian crossing area (Fig. 4.28). After a moment – up to 5 frames (of 25 fps video speed) – the algorithm refreshes the values of tracked moving objects and begins to work properly – the incorrect alarm turns off.

The problems with the incorrect threat detection can also be caused by motorcyclists, because their proportions are not as elongated horizontally as the proportion of cars (Fig. 4.29) – so they also can cause a false alarm, especially when they are visible from the front/back view (the description of types of objects classification was in Section 3.3).

Effectiveness of proposed solution for threat detection

During the experiments, TPR (true positive rate) and FPR (false positive rate) were calculated using the following data:

- TP – true positives – the number of correctly detected persons at the pedestrian crossing during the red light –number of threat detections;
- FP – false positives – false alarm, for example: a motorcyclist detected by the algorithm as a person on the pedestrian crossing during red light;
- TN – true negatives – a correct rejection, like cars during the red light and a person during the green light at a pedestrian crossing;
- FN – false negatives – situations involving people at the pedestrian crossing during the red lights and the alarm does not work.

From the mentioned data, the TPR, FPR, PPV, FNR and ACC representing efficiency of the algorithm were calculated (Table 4.8; Fig. 4.30). The true positive rate was on the level 92.85%. The false positive rate was on the level 5.5%. The accuracy of the model was 94%.

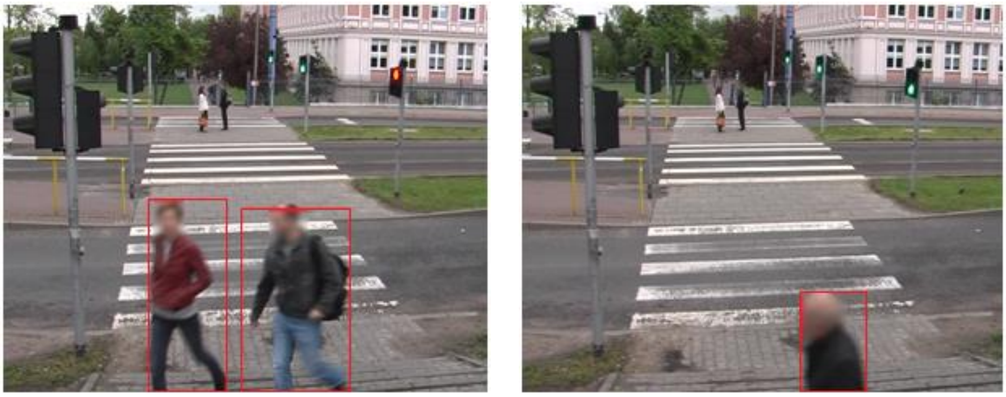


Fig. 4.26. Examples of people passing in front of camera - this situation does not raise an alarm in case of red light detection (right image)

Fig. 4.27. Examples of cars passing the pedestrian crossing during red light – these situations do not raise an alarm, even in case of covering the traffic lights (right bottom image)

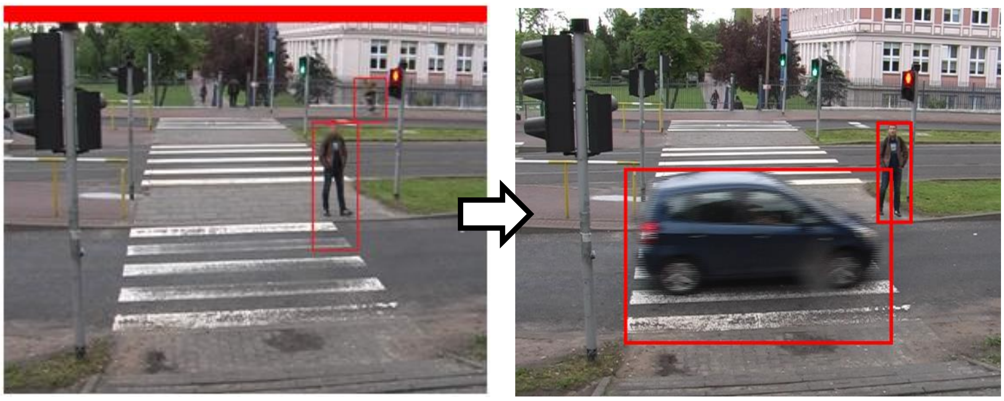


Fig. 4.28. False alarm caused by incorrect moving object tracking by Kalman filter

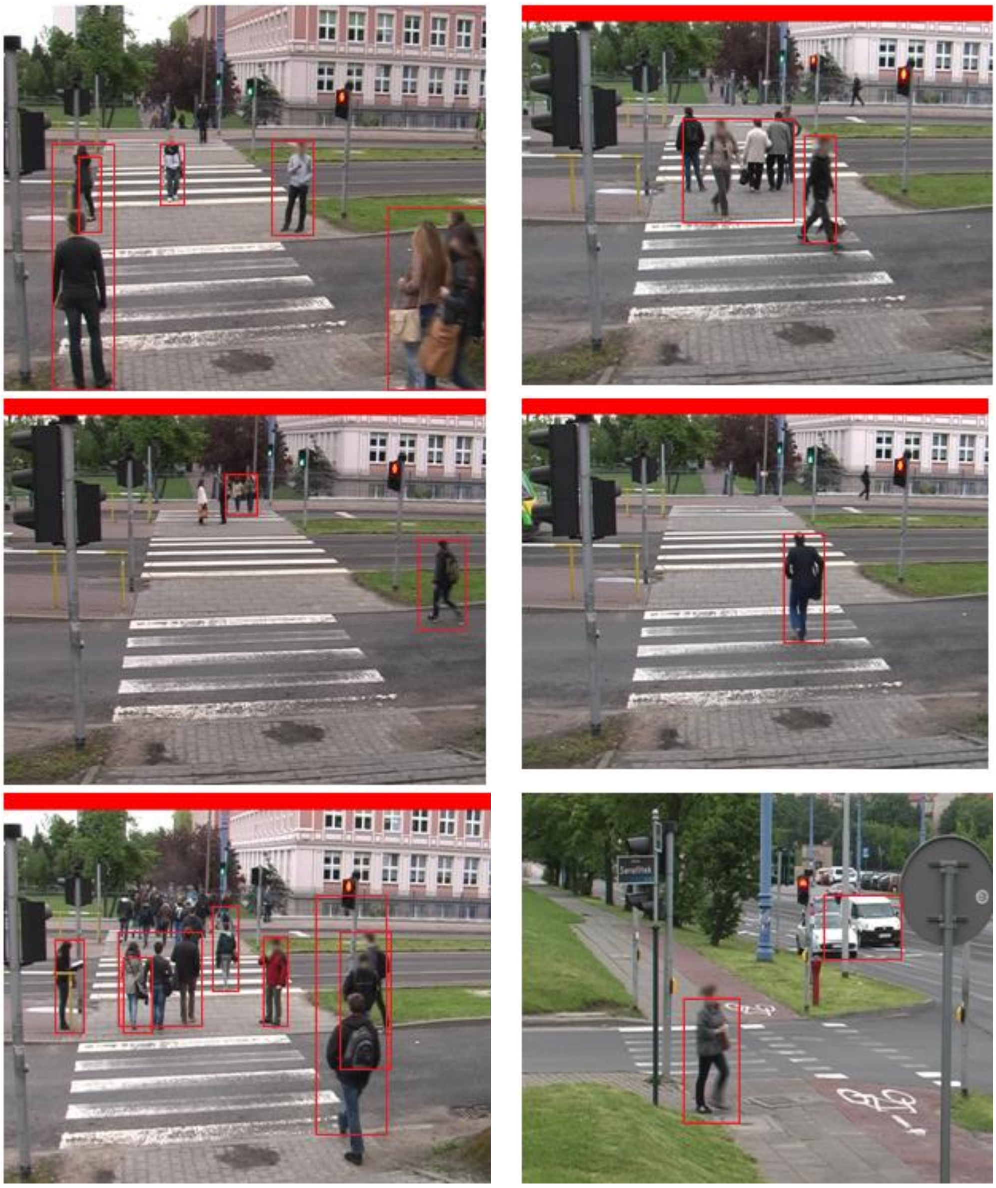


Fig. 4.29. Examples of the operation of the algorithm during red light detection

Table 4.8. Efficiency of algorithm for the detection of a dangerous situation (detection of persons on the red light at a pedestrian crossing)

Features	**TP**	**FP**	**TN**		**FN**
Value	12	3	34 (pedestrian) + 17(vehicle)		1
Features	**TPR**	**FPR**	**PPV**	**FNR**	**ACC**
Value [%]	92.30	5.55	80.0	7.69	94.02

The correct classification of objects into types person and vehicle (described in Section 3.3) results in a high value of TN. 34 persons on a green light at a pedestrian crossing and 17 vehicles passing through the pedestrian crossing on a red light are correctly classified and detected. Thanks to that, the accuracy of the whole algorithm is rated high (Table 4.8).

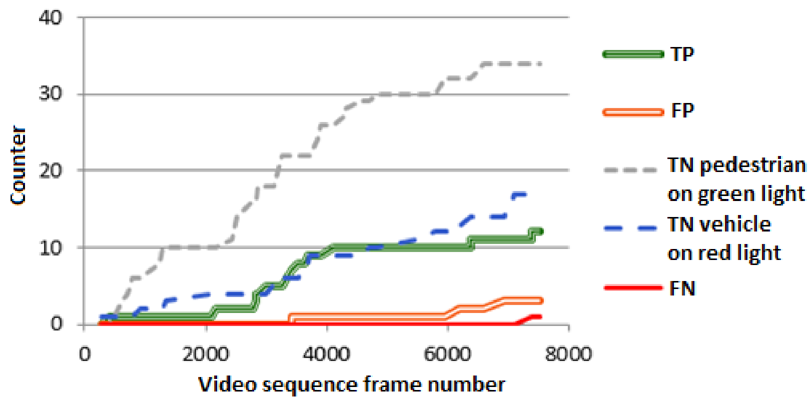


Fig. 4.30. Statistics in the consecutive video sequence frames

The detection of a dangerous situation is well reported for monitoring operator, but there were also false alarms (FP value). These situations occur because GMM sometimes do not detect a person as a whole object and the top and bottom part of the body were separate. The second example of FP value was in a situation where the person was tracked using Kalman filter and then stopped suddenly. In this situation, the position of the object was predicted in the area of pedestrian crossing and this caused an alarm. However, after a few video frames following that, the results of the object's position was refreshed and the alarm turned off .

In Section 6, an example of algorithm for the application of dangerous-situation detection in the embedded system is shown. In this Section, the DSP from Texas Instruments and the Smart Camera from National Instruments were used.

Chapter 5

Analysis of micro-biometrics techniques

In Chapter 4, macro-biometrics modules like people counting, people density estimation, etc. were described. In Chapter 5, a more precise level of biometrics will be described.

In this Chapter, a study that will be helpful in the recognition of a person from a CCTV camera, instead of a biometric device, was presented. The recognition process was tested under various conditions. For example, in the face recognition process, these conditions include non-frontal face image, different face size in the image and various lightning.

In a standard biometric system, the use of dedicated device, like reader or scanning device is needed. The system requires contact with the person and can be inconvenient. The time saving solution, which is quicker than other traditional biometric methods, is using video surveillance system for people recognition. This approach supports the comfortable acquisition of body features and it is invisible to the examined person.

5.1. Face detection and recognition

In this Section, the study about face detection and recognition are presented (Fig. 5.1). The face recognition from a video stream is a more difficult task than the recognition form a standard biometric device, because the system has to be resistant to changes in illumination, scale (size) of pictures and face positions. The recognition of this kind of images has become an important scientific issue to tackle [142, 143]. It can be also observed that the detection and recognition stages are typically analyzed separately.

Face recognition systems [103, 144] are characterized by low invasiveness of acquisition and have increasingly better reliability. The main problem that occurs in such systems is a considerably low resolution of details when taking photographs from long distances. In this dissertation, the effect of reducing resolution on both face detection and identification were examined.

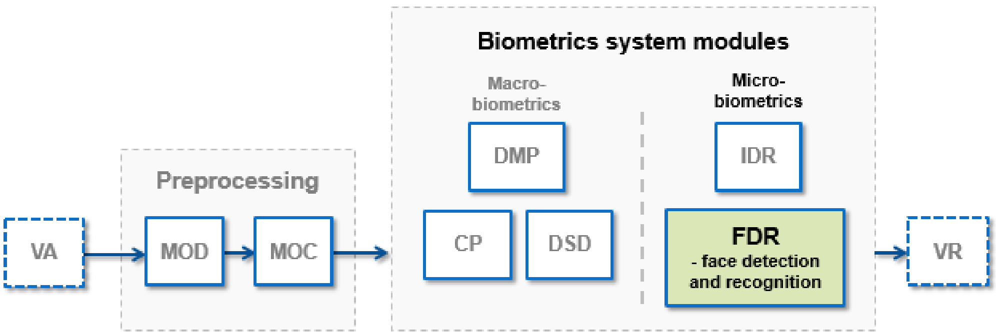


Fig. 5.1. Individual issue presented in the dissertation
– face detection and recognition

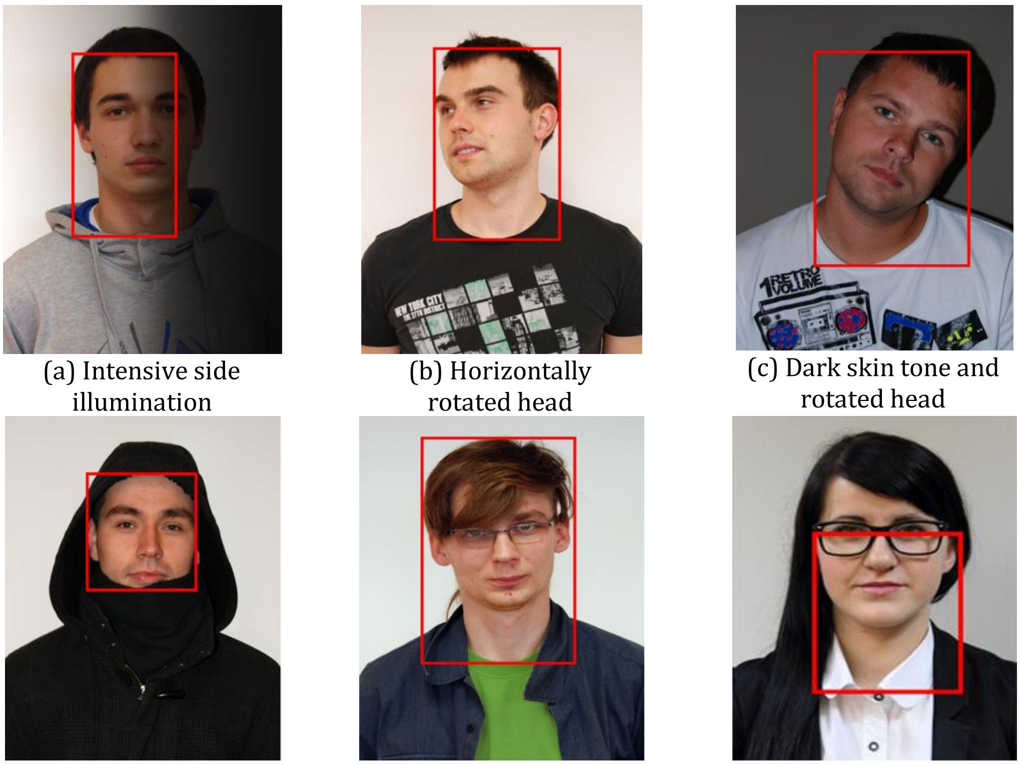
In comparing standard access verification systems using face recognition based on surveillance cameras, the second one requires taking into account several other problems. In most cases, the systems require a full-frontal view of the face to the relevant recognition process. Images acquired by CCTV (closed-circuit television) systems differ in illumination or scale, and include almost only non-frontal views of the subject [144]. Therefore, the main task of the experiments described in this article is to explore the impact of image resolution and the issues associated with various angles of face settings [30, 29, 104].

5.1.1. Comparison of the effectiveness of face detection methods

The methods for face detection, described in Section 2.3.1, are compared below.

Face detection using skin color filter

Figure 5.2 shows examples of detection using skin filter. However, a problem with this method is a small resistance to low lighting or an intensive side illumination. In this case, the algorithm incorrectly selects the face area in the image. Improper lighting conditions can be compensated for by the use of histogram equalization, which, unfortunately, may rarely decrease the detection area. It is worth mentioning, that the algorithm detects faces correctly even when the skin has a dark tone or the face is partially hidden, e.g., by a scarf, and even when the head is rotated. In Figures 5.2 and 5.5, the algorithm generally detects faces properly; however, it has significant disadvantages. As the method identifies the skin, together with face detection, the neck and sometimes even the blond hair can also be detected increasing ROI. A problem may occur with detecting the face correctly when the person is wearing thick-framed glasses.



(a) Intensive side illumination

(b) Horizontally rotated head

(c) Dark skin tone and rotated head

(d) Face partially hidden

(e) Glasses and blond hair

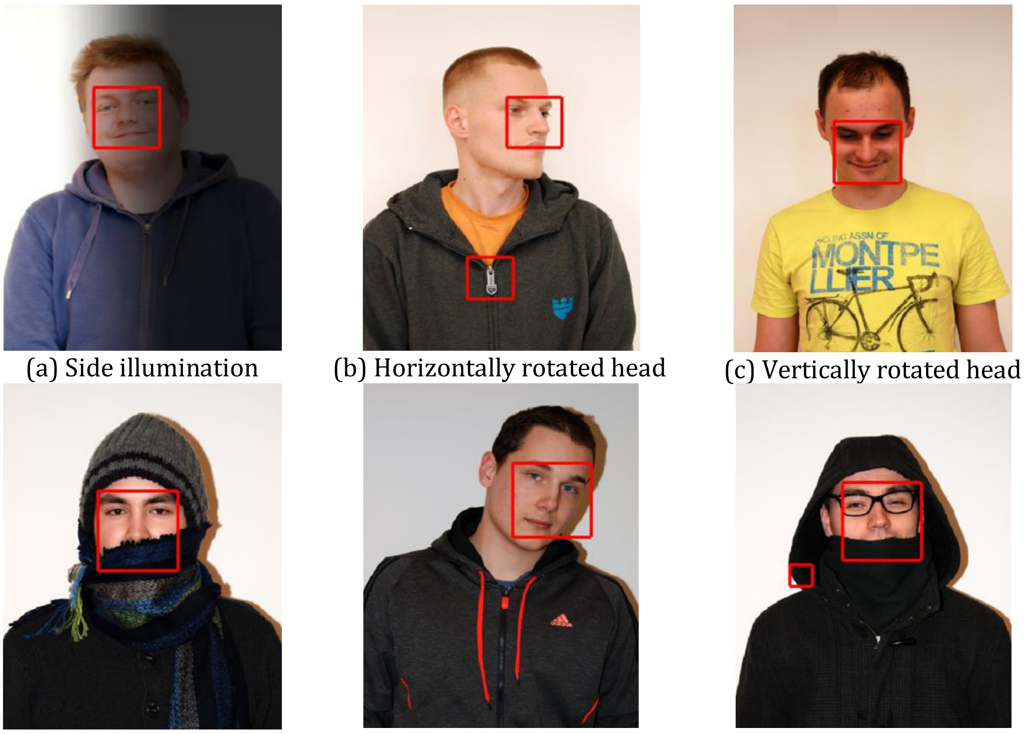
(f)Thick-framed glasses

Fig. 5.2. Examples of face detection with the use of skin filter

Face detection with the use of geometric models

The experiments have confirmed that face detection with the use of geometric models is resistant to changes of orientation or face positions (smaller than 45° in every direction) as shown in Fig. 5.3 (b) (c). In full light conditions, the program works correctly. However, problems can occur when light is directed from the side (Fig. 5.5) and from above, because of shadows falling on the face. Detection during rotations of the head, as long as the facial image is symmetrical (in the horizontal orientation), works well. Additional accessories like a hat, glasses, and a scarf generally do not influence the proper detection. An incorrect detection can occur in case of a complex background or patterned clothes.

Fig. 5.4 shows the difference in ROI designation (area of detected face) between two methods: the skin color filter and the geometric model. The second method is more precise in face location than the use of skin color filter. This advantage has a positive effect on the face recognition process.



(a) Side illumination (b) Horizontally rotated head (c) Vertically rotated head

(d) Face partially hidden (e) Head tilted (f) Thick-framed glasses and face partially hidden

Fig. 5.3. Examples of face detection with the use of Hausdorf distance

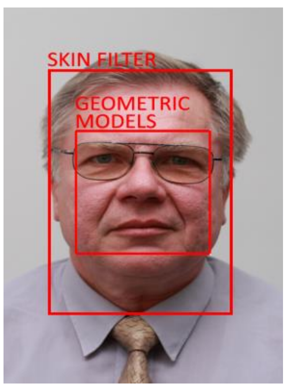


Fig. 5.4. Example of difference in the ROI designation (area of detected face) between two methods: skin color filter and geometric model

Haar-Like method for face detection

The Haar-like method uses classifiers for detection of eyes (in previously determined ROI containing the face), thus, if the eyes are hidden (as in Fig. 5.5 —an example of *the top part of the face hidden*) the face will not be detected.

In summary, the Haar-like method precisely determines the location of the face in both color and grayscale images. It is resistant to changes in lighting (thanks to histogram equalization) and to insignificant rotations of the head.

In Fig. 5.5, the differences in the operation of the Haar-like algorithm along with other methods, which make use of skin color filter and geometric models, were shown.

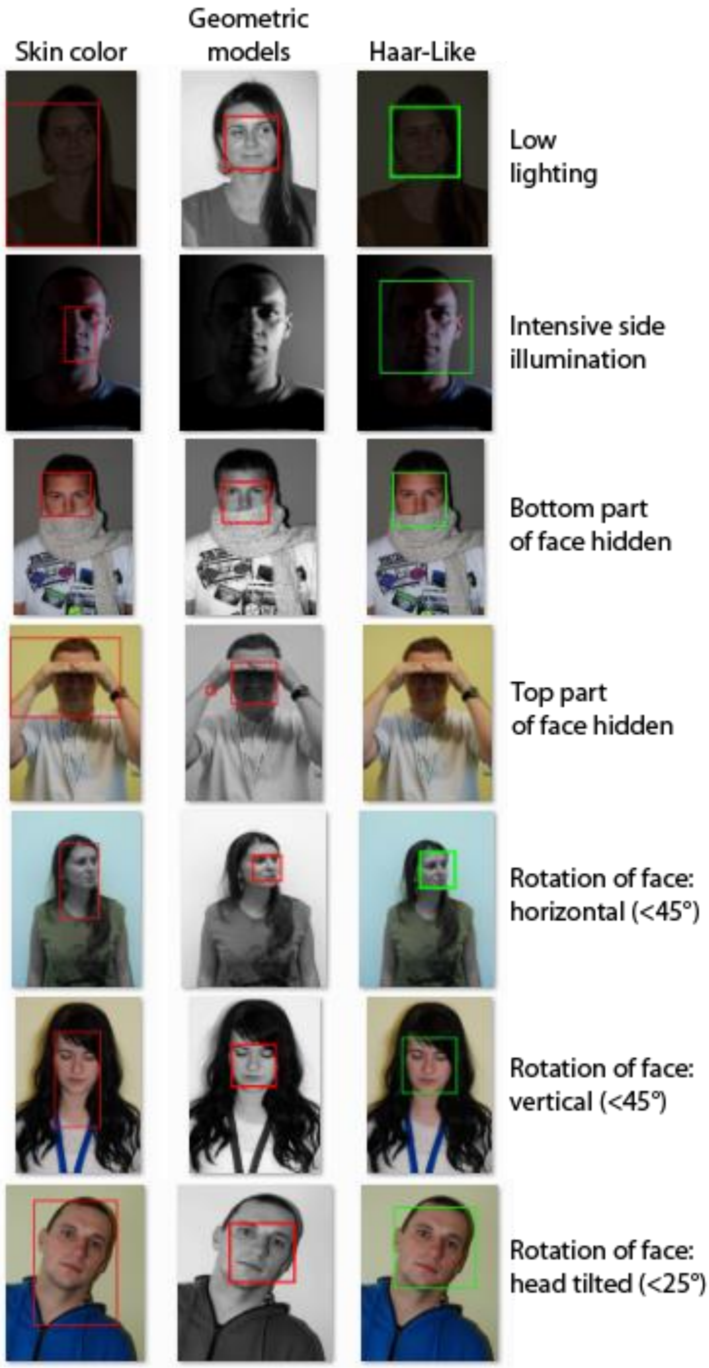


Fig. 5.5. Examples of face detection with different methods [29]

5.1.2. Effectiveness of face detection in low image resolution

In the author's preliminary research, the minimum size of face in the correctly detected image was examined. The possibility of detection using skin filter, geometric models and Haar-like features for very low-resolution images is shown in Table 5.1. In these images, the face was set in the frontal position and evenly lighted.

Table 5.1. Face detection from very low resolution image

<table>
<tr><th rowspan="2">No. of downsamplings (image resolution [px])</th><th colspan="3">Face detection method</th></tr>
<tr><th>Skin filter</th><th>Geometric models</th><th>Haar-Like</th></tr>
<tr><td>0 (480×720)</td><td rowspan="7">Face detected</td><td rowspan="5">Face detected</td><td rowspan="7">Face detected</td></tr>
<tr><td>2 (240×360)</td></tr>
<tr><td>4 (120×180)</td></tr>
<tr><td>8 (60×90)</td></tr>
<tr><td>12 (40×60)</td></tr>
<tr><td>16 (30×45)</td><td>No detection</td></tr>
<tr><td>32 (15×23)</td><td>No detection</td></tr>
</table>

During the latter study of the influence of image resolution on the detection process, 5,760 facial images from Yale Face Database were examined. The images were in 5 resolutions, starting at 640×480 pixels and then the image size was reduced 2, 4, 8 and 12 times (the smallest image resolution was 54×40 pixels – an example of a downsampled image is shown in Fig. 5.6). As described in Appendix B, the Yale database contains images of 10 people. Every person has their head in nine positions. Every position has 64 different light conditions. Thus, 28,000 photos were examined using the Haar-like method of the face detection.

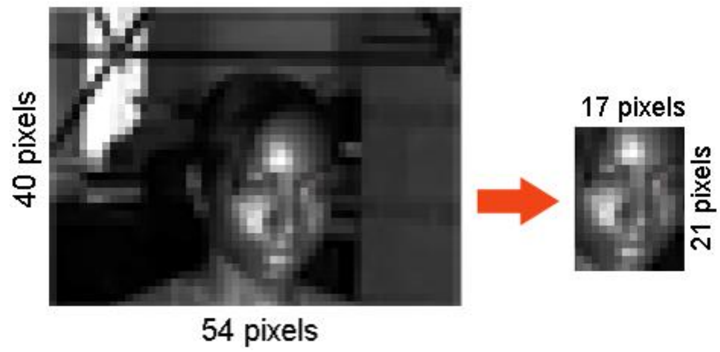
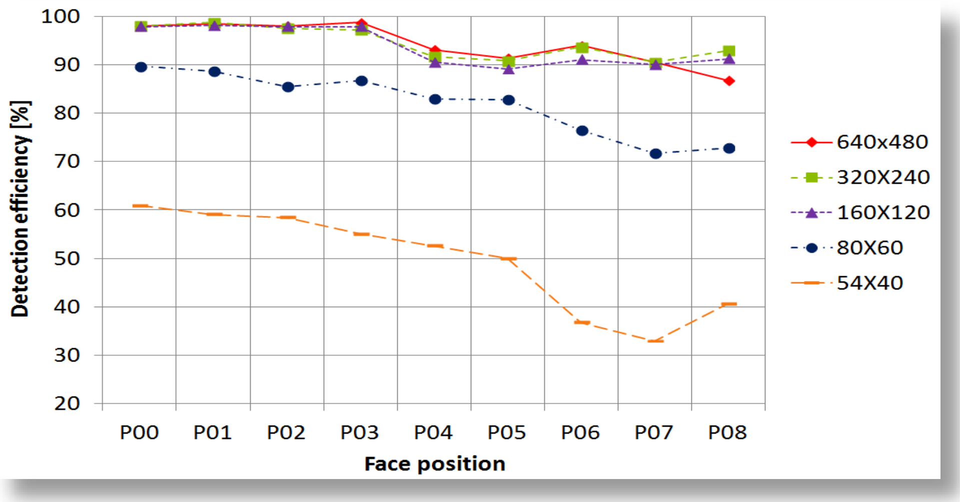


Fig. 5.6. An example of face detection in image with low resolution of 54×40 pixels taken from Yale database

Figure 5.7 shows the influence of image resolution on face-detection process regarding various face positions, i.e. for each individual position from the database. It should be noted that the reduction of image resolution even by 4 times does not affect the detection process. When the image is downsampled 8 times, the detection efficiency is reduced approximately to 10% for P00 position and about 20% for face in position P08 — head directed down (by 45°) and to the left (by 30°). Figure 5.7 indicates that the Haar-like method has the best detection results for positions P00 – P03. If the face is rotated away from the frontal position, the face detection efficiency becomes significantly lower.

It can be observed (Fig. 5.7) that a face can be detected (although with lower efficiency) even from an image with dimensions of 54×40 pixels. As shown in Fig. 5.6, the face in the photo with a resolution of 54×40 pixels has dimensions of about 17×21 pixels. Other faces in the images were not exactly the same size, but similar. This example has been included to illustrate that the detection limit of 54×40 pixels does not apply to the size of the face, but the whole picture. The facial area is much smaller and is approximately 20–30 pixels.



P00 – the front face
P01 – head directed slightly upward (25°)
P02 – head directed slightly upward (25°) and to the left (20°)
P03 – head directed slightly to the left (20°)
P04 – head directed slightly down (25°) and to the left (20°)
P05 – head directed slightly down (25°)
P06 – head directed upward (45°) and to the left (30°)
P07 – head directed to the left (45°)
P08 – head directed down (45°) and to the left (30°)

Fig. 5.7. Influence of image resolution and various face positions on face detection process [29]

Figures 5.8, 5.9 and 5.10 show the influence of light source angle on the process of face detection in different image resolutions. Like in the case of detection efficiency of rotated faces, resolution does not have any significant impact on the detection of images with 640×480 (Fig. 5.8) and 160×120 (Fig. 5.9) pixels – the graphs are similar to each other and the face detection efficiency is between 70-100%. As shown in Figs. 5.8-5.10, the effectiveness of face detection is high (70-100%) for the face illuminated from the front side. When the face is illuminated from the bottom, the effectiveness of detection is reduced to 80% for 640×480 and 160×120 image resolutions, and 20% for 54×40 resolution. The worst detection results occur when the face is illuminated from the bottom right side. In this particular case, the detection efficiency is 70–90% for 640×480 and 160×120 image resolutions and 0–19% for resolution of 54×40 pixels.

Figures 5.11-5.14 show that the impact of face detection efficiency is dependent on the angle of light emission (separately in a horizontal and vertical direction) and head positions: P00, P01, P03, P05 and P07. The abbreviations for each face position from Yale database are shown in Fig. 5.7.

The above described study may be useful for intelligent surveillance where the choice of camera location is significant in relation to the prevailing interior lighting conditions in order to achieve the highest possible efficiency of detection.

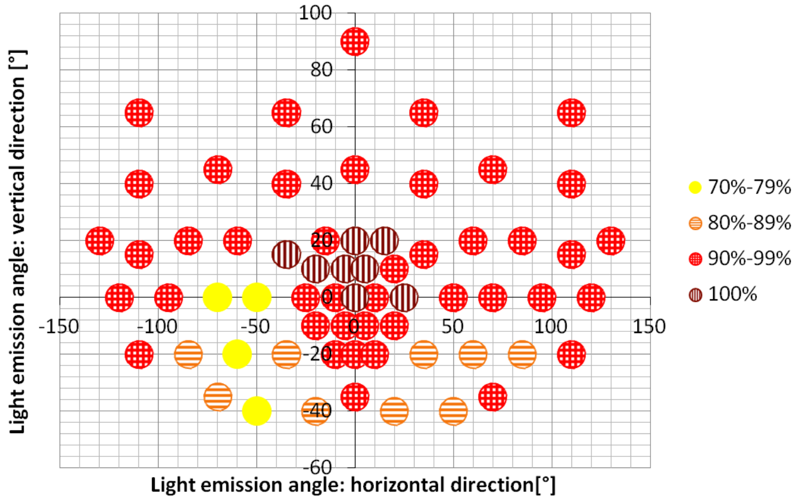


Fig. 5.8. The relation of the angle of light emission to the process of face detection of an image with a resolution of 640 × 480 pixels [29]

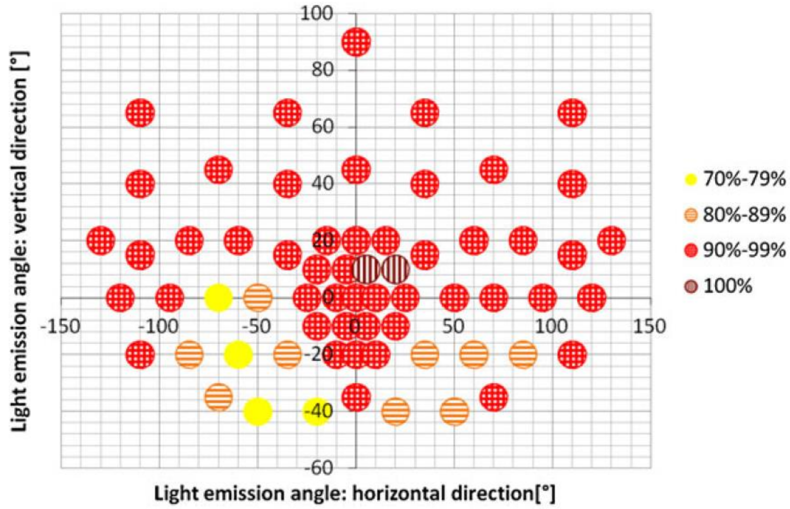


Fig. 5.9. The relation of the angle of light emission to the process of face detection of an image with a resolution of 160 × 120 pixels [29]

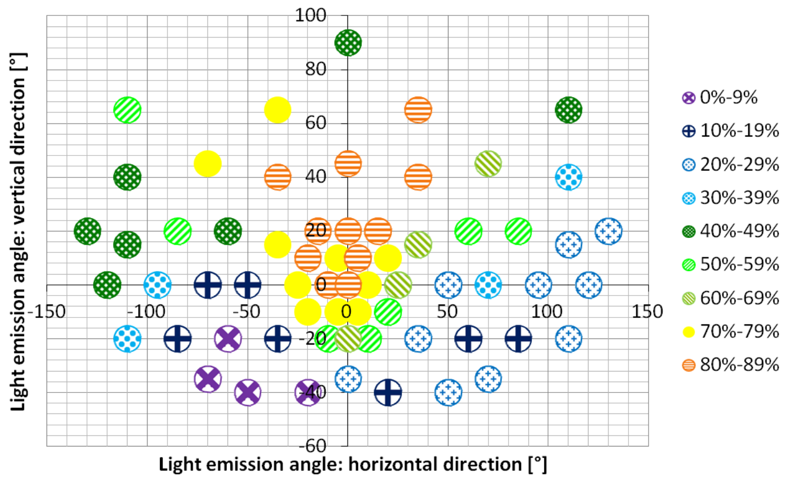


Fig. 5.10. The relation of the angle of light emission to the process of face detection of an image with a resolution of 54 × 40 pixels [29]

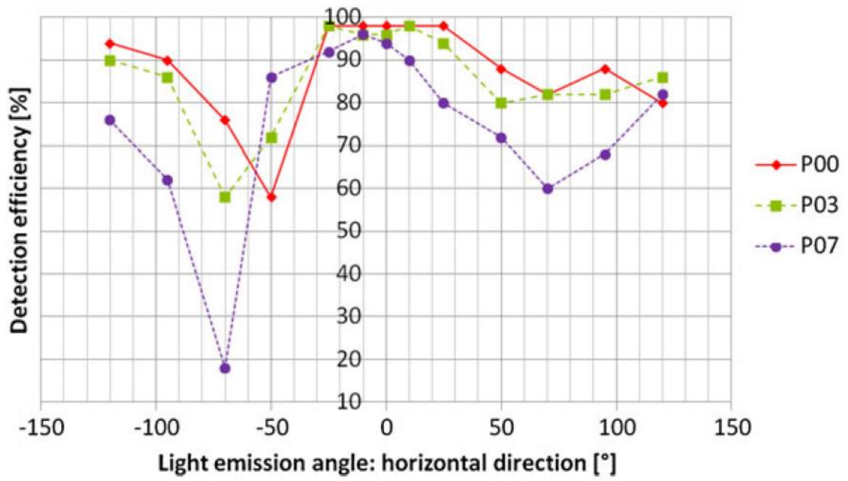


Fig. 5.11. A study of the impact of face detection efficacy depending on the angle of light emission (in the horizontal direction) and head positions (P00, P03, P07) [29]

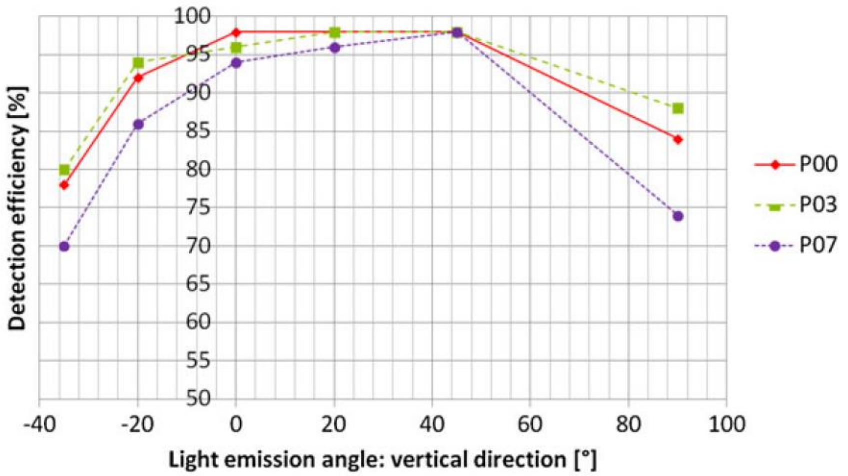


Fig. 5.12. A study of the impact of face detection efficacy depending on the angle of light emission (in the vertical direction) and head positions (P00, P03, P07) [29]

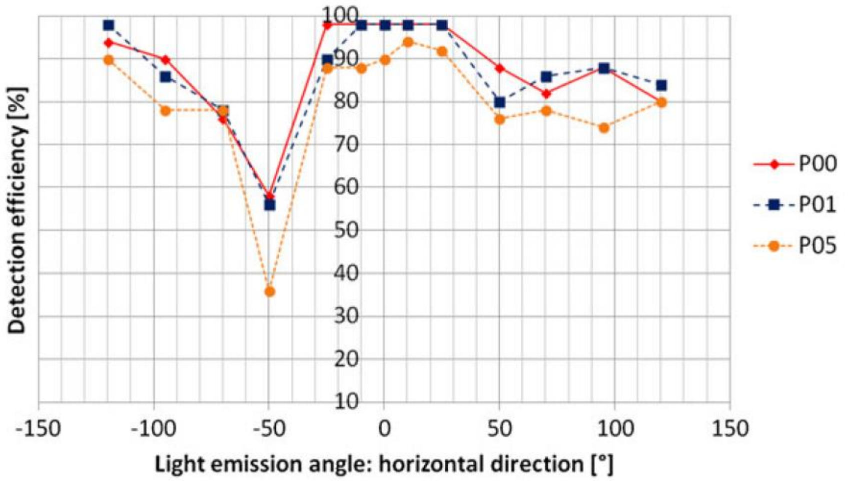


Fig. 5.13. A study of the impact of face detection efficacy depending on the angle of light emission (in the horizontal direction) and head positions (P00, P01, P05) [29]

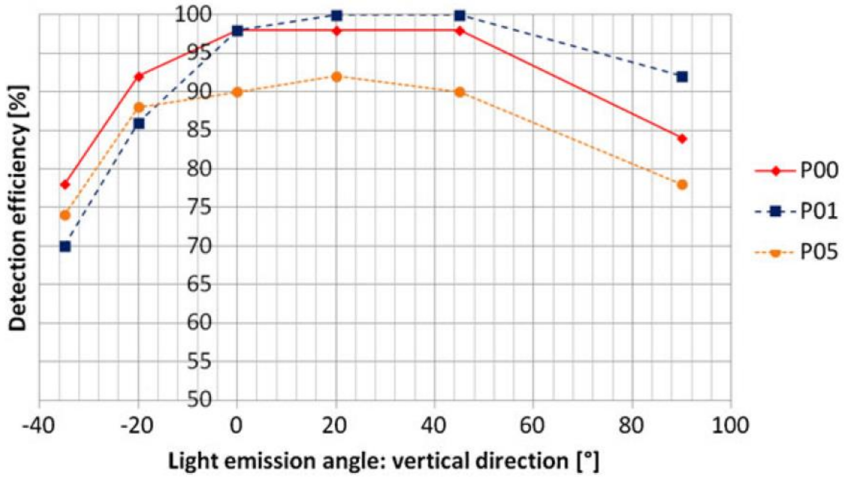


Fig. 5.14. A study of the impact of face detection efficacy depending on the angle of light emission (in the vertical direction) and head positions (P00, P01, P05) [29]

5.1.3. Effectiveness of face recognition in low image resolution

Face recognition application

One of the methods for face recognition is *Face Recognition System 2.1* [145]. In the paper [30], the authors modified this algorithm and a user interface was created. This program uses an algorithm based on PCA.

The software allows for the operation in two modes: continuous and batch processing. A simplified block diagram of the recognition software is shown in Fig. 5.15.

In the first mode (continuous processing), an image from an IP wireless webcam (e.g. D-Link DSC-930L [146]) or a standard USB camera can be acquired. Then, the recognition of the face belonging to the person that is in front of the camera is performed. If needed, the image acquired in the YUV color palette is automatically converted from YUV to the RGB color space. Other possible operations in this mode are noise reduction, face detection in the image (the skin color filter was used) and also background removal—in order to reduce the processing area and the calculation time.

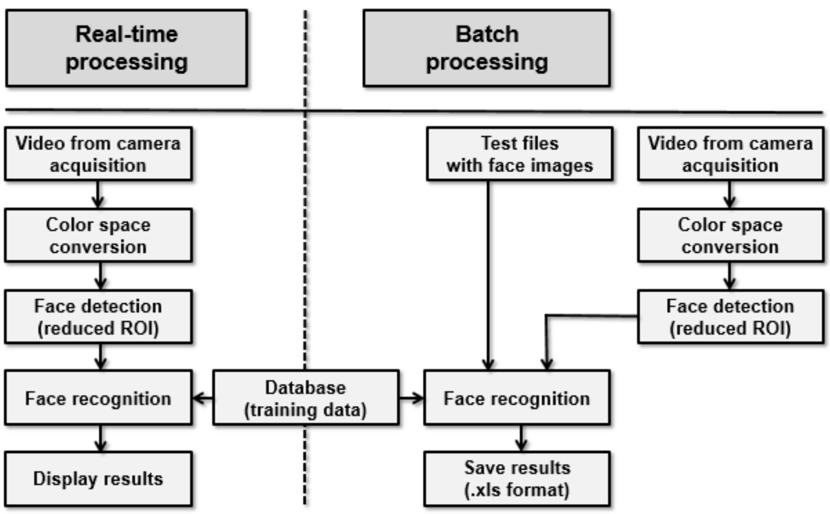


Fig. 5.15. Simplified block diagram of the face recognition application

The batch processing mode gives two possibilities:

- loading images from the database and saving results in *.xls* format
- loading a specified number of frames that can be recorded from a camera and saving the results in *.xls* format.

An experimentation on the influence of image resolution on the process of face recognition has been tested. During the analysis of face recognition from a low-resolution image a modified software [145] was used. This program uses an algorithm based on PCA.

In this particular experiment, the Yale Face Database, FullFaces and MUCT databases were used. These databases were briefly described in Appendix B. The research included both the detection and recognition processes.

The first database used (Yale Face Database) contains several pictures of rotated faces (9 positions) of 10 individuals under varying light conditions (64 different illumination angles for each position)—each of the photographed persons appears in 576 images. Due to the difficulty in detection of images with the light source at a high angle, the database was limited to 25 % for the recognition process. The person number 5, was rejected from the experiment because of very poor face detection (average of 50 % for all face positions). From the original pictures with a resolution of 640×480 pixels, the faces have been extracted using the Haar-like algorithm and then resampled to a resolution closest to the power of 2, which in this case is 256×256 pixels. All of the pictures have been downsampled 2, 4, 8, and even 12 times corresponding, respectively, to the resolutions 128×128, 64×64, 32×32, and 21×21 pixels.

For the experiment with Yale database, 48 random face pictures of every individual was chosen as reference images. Another set of 95 random face images (for every person) was used for testing.

Figure 5.16 shows FAR/FRR (false acceptance rate/false rejection rate) plots for resolutions from 256×256 down to 21×21. As can be seen, downsampling 2 and 4 times does not influence the recognition accuracy. Downsampling 8 and 12 times decreases recognition by about 8%. The EER in the first three cases is about 30 %. The presented results show that even 4 times downsampling does not influence the accuracy of recognition, irrespective of face rotation and light conditions. The combination of various light conditions and face rotation, however, cause some deterioration of results in comparison to the previous experiments described in [30, 104]. The previously shown results of front view images [30], i.e., with no face rotated (which is the main requirement in the norms) can be very well distinguished from each other. It should be noted that in previous researches [30, 104], the processess of face detection and recognition were considered separately.

The second database (FullFaces) includes 10 (horizontally and vertically) rotated face pictures of 30 individuals under constant light conditions, and saved with a resolution of 512×342 pixels. For the experiment, all of the pictures were chosen. After the detection stage, the face images were saved with resolution 256 × 256 and then downsampled 2, 4, 8, and 12 times.

In case of the FullFaces database, 4 random pictures of each individual have been used to create models, and other 6 files were used in the recognition stage. In contrast to the Yale database, the face pictures are only rotated horizontally and vertically. This noticeably influences the recognition accuracy, which can be seen in Fig. 5.17.

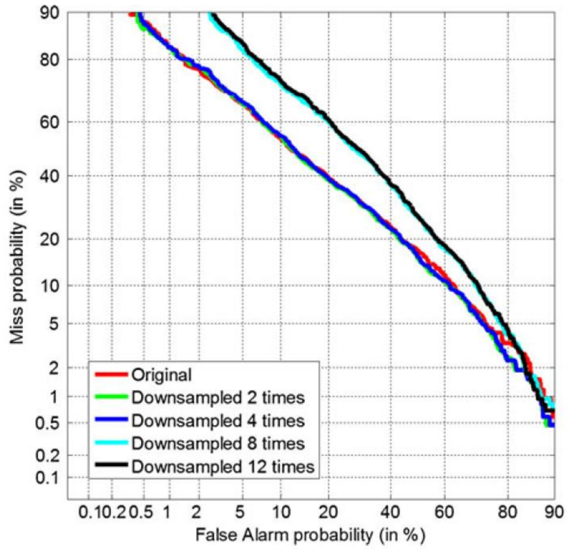


Fig. 5.16. FAR/FRR plot of face recognition accuracy for original and downsampled Yale database [29]

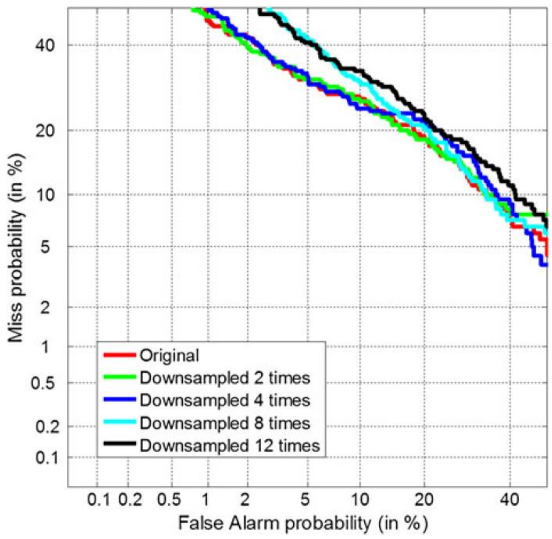


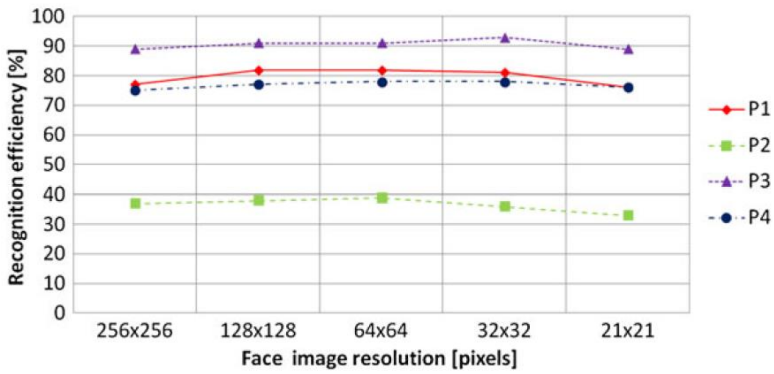
Fig. 5.17. FAR/FRR plot of face recognition accuracy for original and downsampled FullFaces database [29]

The EER in all cases is in the range of 18–23%. It can be noticed that even downsampling up to 8 or 12 times does not decrease, significantly, the recognition accuracy in contrast to the Yale database. Major differences can be seen at the ends the lines.

The presented results show that in stable light conditions, face rotation and decimation do not significantly decrease face recognition accuracy using the PCA algorithm.

To make previous results more visible, the MUCT database with color images and a vast number of people has been used. Unfortunately, this database contains different quantities of images for particular individuals. As a result, the research was divided into two separately stages.

The first stage of the MUCT database recognition research was the analysis of its effectiveness when the head position was changed. Two hundred and seventy-six subjects photographed in five different positions were selected for the designation of the effectiveness of recognition. One face image from each person was taken for the training stage and the remaining four were used for the recognition stage. The recognition efficiency for the original and downsampled *MUCT database* for different face positions is shown in Fig 5.18. This figure shows that upward and downward directed head positions have the highest recognition efficiency of up to 90% and 75%, respectively. Generally speaking, face recognition algorithm is resistant to any face resolution changes. The recognition efficiency for the resolution of 21×21 pixels is reduced only up to 10% for each tested position.



where:
P1 – head directed slightly to the left (21°)
P2 – head directed to the left (47°)
P3 – head directed upward (21°)
P4 – head directed down (21°)

Fig. 5.18. Recognition efficiency for original and downsampled MUCT database for different face positions [29]

The second stage of the MUCT database recognition research was the examination of the effectiveness in lighting changes. One hundred and ninety-nine differently illuminated individuals (2 lighting sets) were taken for the recognition process. The first illumination set (marked as 'q', 'r', 's') was used in the experiment and consists of 91 individuals. The second illumination set (marked as 't', 'u', 'v') consists of 108 individuals. The first of the two illumination sets ('q' and 't') was used for two different training stages, separately. The remaining illuminations ('r', 's' and 'u','v', respectively) were used for the recognition stage.

Fig. 5.19 shows the recognition efficiency for the original and downsampled MUCT database for different sets of light conditions. The difference in the recognition efficiency between these two illumination sets was only about 5%. In case of 256×256 pixels resolution, the recognition efficiency has remained at the level of 75%. After the decimation, with the face image size of 21×21 pixels, the recognition efficiency decreased to about 5% and was at the level of 61%.

The achieved EER in every case (even for low resolution) was between 18–23% (for the FullFaces database). An additional advantage of the approach is that it operates correctly even with a little amount of training data. The proportions of the training data to the testing data in the experiments are up about 0.3.

Another positive feature, in relation to the EN 50132-7 norm, is the fact that face detection and recognition can take place with lower resolutions than those that are indicated in the abovementioned standards. The minimum face height in the picture in the EN 50132-7 standard and research is shown in Table 5.2.

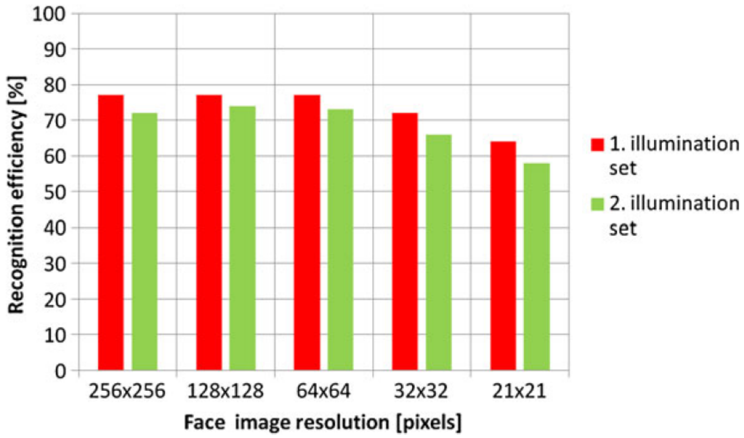


Fig. 5.19. Recognition efficiency for original and downsampled *MUCT database* for different light emission angle [29]

Table 5.2. Comparison of EN 50132-7 norm and research for face detection and recognition [29]

	The minimum face height in the picture [pixels]	
	EN 50132-7 norm	The proposed solution
Face detection	31	21
Face recognition	74	21

The presented studies show the efficiency of face detection and recognition process in various light condition, face position as well as when the face has a small resolution in the image. These studies are helpful in situations when we want to use CCTV camera in the biometric system, instead of a standard acquisition device.

5.2. Iris detection and recognition

According to the standards and norms described in Appendix A, iris detection and recognition is possible with resolutions of the iris at 100×100 pixels. Therefore, in surveillance systems, people can be recognized with the use of iris image, when there is a large zoom of camera lens of surveillance systems.

In this Section, iris image acquisition, segmentation, normalization and feature coding are analyzed (Fig. 5.20). Results of an average time of this algorithm step and also recognition accuracy for IrisBath database were also presented.

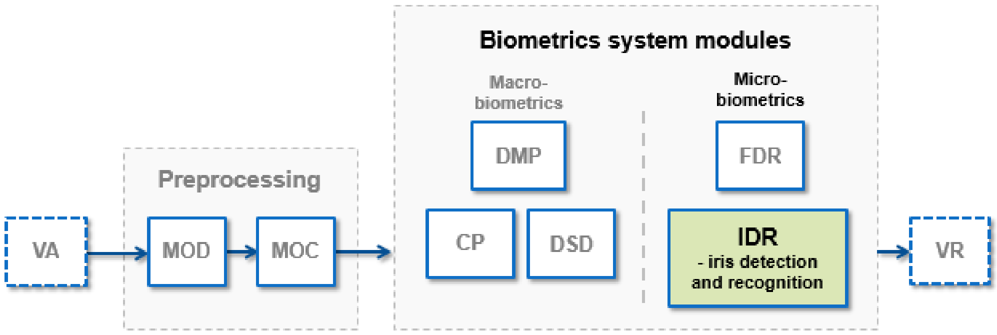


Fig. 5.20. Individual issue presented in the dissertation – iris detection and recognition

5.2.1. Analysis of problems of human iris acquisition

Proper acquisition of iris picture is a key element of the biometric identification system. Several conditions have to be met by the image of the iris, before it could serve to identify the person. There are also many problems to be faced with encoding the iris:

- change of opening angle in the pupil depending on lighting conditions
- covering a portion of the iris region by eyelids and eyelashes
- rotation of the iris due to the inclination of the head or eye movement.

Iris acquisition system

An experimental iris acquisition system with impulsive infrared lighting was designed and built. In the experiments, the acquisitions were performed both in daylight and infrared light. The experiments and the investigations were prepared according to the requirements of the European standards ISO / IEC 19794-6 [147] (Appendix A).

Standard iris acquisition systems are shown in Section 2.3.2. Image acquisition is typically done under infrared light (with wavelengths of 0.75–1.4 μm). With such light, even the iris, which has dark pigmentation, can be effectively recognized through distinctive features.

Figure 5.21 shows acquisition systems of the iris using the ophthalmological table. In this system, a good quality (image sharpness) of the iris image is obtained, i.e., through two-point support of the head, which affects the stability of the head during an image acquisition. A large monitor that shows the iris image in real time supports accurate adjustments of iris parameters (the focus and exposure time). In the system, the eye is illuminated by a set of diodes in the near infrared (with a wavelength of 940 nm), arranged in a circle. This set of diodes is very close to the camera lens and do not obscure its vision. The illuminator operates in two modes: the continuous mode and the pulse mode. While operating, the illuminator is synchronized with the camera shutter using an interface with a microprocessor.

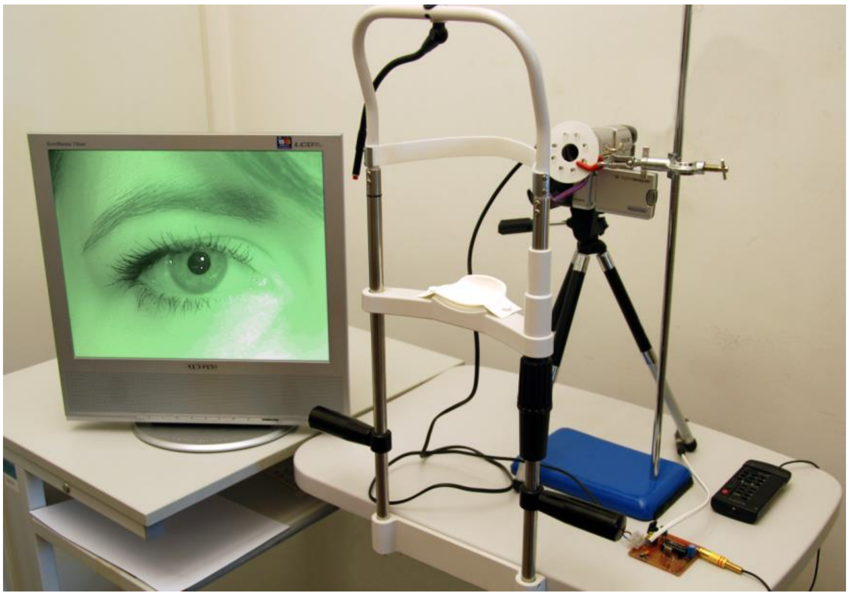


Fig. 5.21. An acquisition system of the iris with the use of ophthalmological table [32]

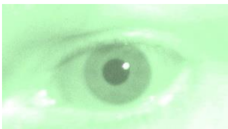
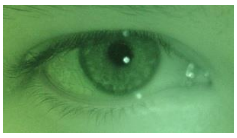
During the research, four various video cameras were used. A summary of the most important camera parameters are shown in Table 5.3.

In order to illustrate the possible problems during image acquisitions, sets of iris images were registered using various cameras and variable types of illuminations. These results are presented in Table 5.4.

Table 5.3. Parameters of the cameras used in iris acquisition [32]

Parameters	**Sony DCR-TRV33E**	**Sony HDR-XR200 VE**	**Panasonic NV-GS500**	**Modecom Venus Web Cam**
Camera standard	MiniDV	HD	MiniDV	USB
Image sensor	1Mpix	2.3Mpix	3xCCD, 3x1Mpix	1.9Mpix
Properties in the infrared range	Switchable filter and IR illuminators; high sensitivity	Switchable filter and IR illuminators; high sensitivity	Embeded IR filter	Embeded LED illuminators; no IR

Table 5.4. Example of correct and incorrect iris images registered with the use of various cameras [32]

	Sony DCR-TRV33E	**Sony HDR-XR200 VE**	**Panasonic NV-GS500**	**Modecom Venus Web Cam**
Correct				
Incorrect	 Overexposure image (IR, mono)	 A large amount of noise and reflections (IR, mono)	 Poor lighting conditions, lack of focus, reflections	 Poor lighting conditions, large shadows from the eyelashes

5.2.2. Determination of iris features

The various stages of iris analysis (segmentation, normalization, features coding) are described in Section 2.3.2. During the author's research, the program IrisCode_TK2007 [148] was used. A multi-processing was used in order to automatically create iris codes for multiple files. The study involved IrisBath database.

In the experimental program used in this research [148], the Hough transform was used for segmentation process. The edge maps were designated using a modified Kovesi algorithm [149] based on Canny edge detector. An illustration of the segmentation process with time analysis is presented in Fig. 5.22.

During tests, the implementation by Libor Masek [128] was used, which bases on the Daugman proposal of iris normalization. At the same time, an area of the iris was selected and then subjected to normalization using, both angular distribution and distribution along the radius. The processing time analysis of iris normalization is shown in Table 5.5.

A feature coding is done using a variety of logarithmic Gabor filters, the so called Log-Gabor filters [150]. These filters have certain advantages over the classic Gabor filters, i.e. by definition, they do not possess a DC component, which may occur in the real part of the Gabor filters. Another advantage of the logarithmic variation is that it exposes high frequencies over low frequencies. This mechanism approaches the nature of these filters to typical frequency distribution in real images. Due to this feature, the logarithmic Gabor filters expose information contained in the image better.

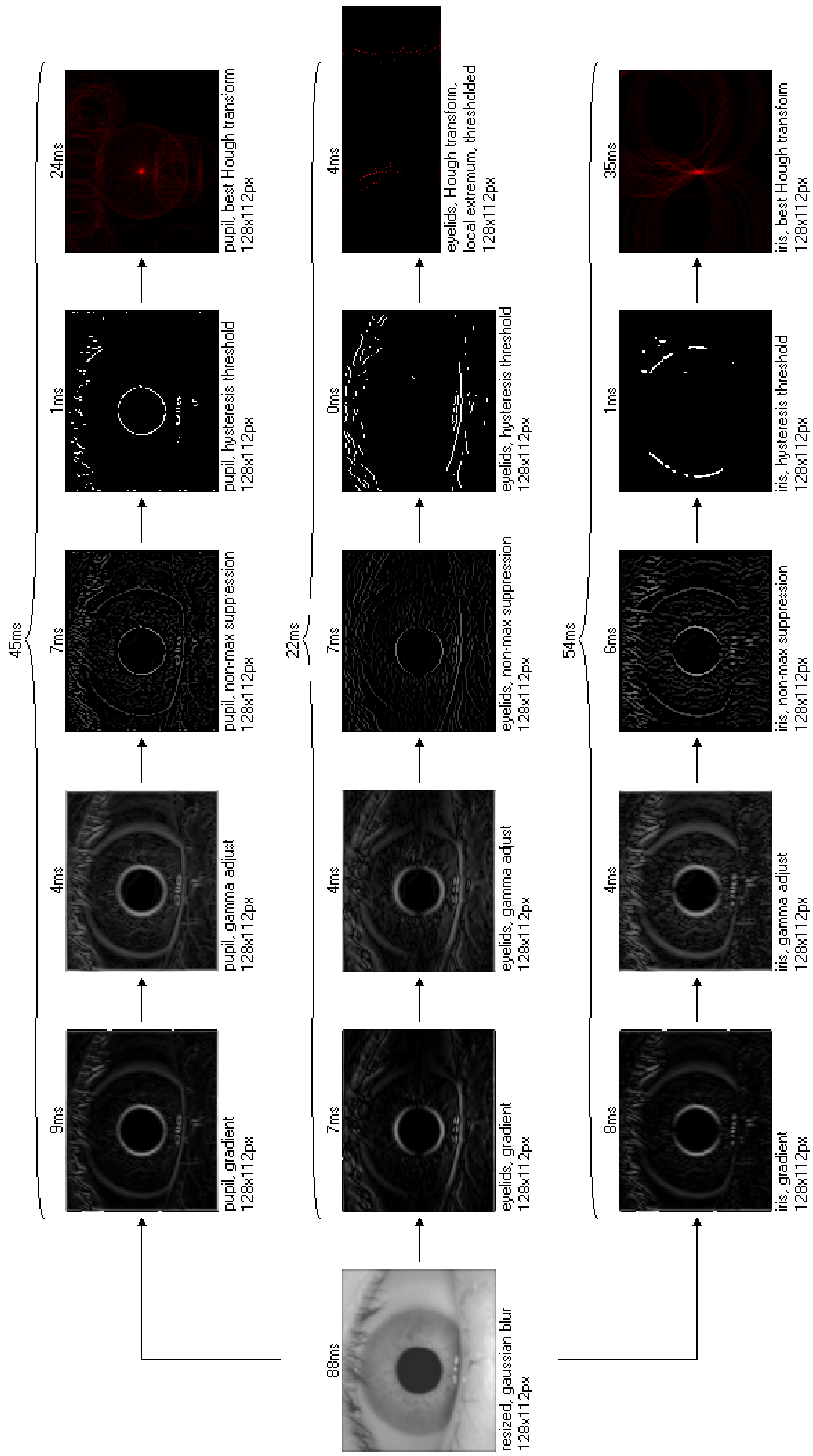


Fig. 5.22. Example time-consuming analysis of segmentation process [32]

Test results are presented in Table 5.5. The section “Information”, includes the total number of files and the number of iris classes. The section “Results”, contains the results of the processed images. These are the average times of individual stages and the total processing time for all files. Table 5.5 shows the times of individual stages, expressed in percentage, of the overall time for all tested databases (processed with Intel Core i7 CPU; 2,93 GHz).

The program also contains an option “Multithreading”, which enables multithreaded processing on multiprocessor machines. Figure 5.23 presents the comparison of the processing times of various stages, when the option “Multithreading” was used or not (processed on Intel Core i7 CPU; 2,93 GHz) for IrisBath database. The total processing time for one processor was about 17 minutes. while for two processors, it was about 9 minutes.

Table 5.5. Processing times for two databases: IrisBath and CASIA [32]

Information	**Database name**	**CASIA**					**IrisBath**	**All databases**	**Total [%]**
		Iris Image	**IrisV3**			**Iris V4 Distance**			
			Interv al	**Lamp**	**Twins**				
	Number of classes of irises	217	498	822	400	142	22	2101	
	Total number of files	756	2655	16213	3183	2572	432	25811	
Results (times)	The average time of image loading [ms]	87	86	278	283	3956	1120	968	47.02
	The average time of segmentation [ms]	103	107	320	306	4404	1249	1081	52.53
	The average time of normalization [ms]	2	2	2	2	2	2	2	0.10
	The average time of features encoding [ms]	1	1	1	1	2	1	1	0.06
	The average total time coding [ms]	193	196	601	592	8364	2372	2053	99,71
	The average time of writing results on disc [ms]	6	5	6	6	8	5	6	0,29
	Total time database processing (hour:minutes:seconds)	00:02:30	00:08:55	02:43:57	00:31:46	06:10:15	00:17:10	09:54:28	-

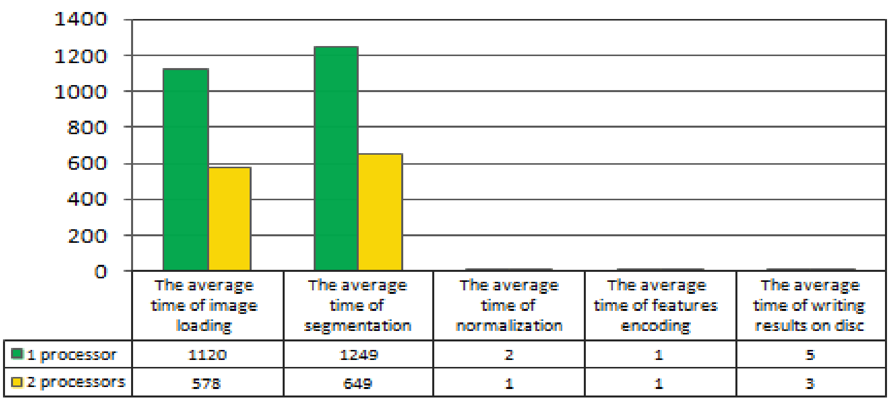


Fig. 5.23. Comparison of processing times [ms] of various stages (processed with Intel Core i7 CPU; 2,93GHz) for IrisBath database (time precision 1 ms) [32]

5.2.3. Influence of various iris part on the analysis of the recognition process

Proper selection of parts of the iris on the effectiveness of recognition process is very important. In this experiment, the dependency of iris recognition from different parts of the iris using IrisBath database, was tested.

First, the particular angular span of the iris influences the identification of a person was examined. The angular span, as a range of the iris that is used for normalization, is defined. Two semicircles were obtained by dividing the circle describing the iris with a vertical line. In each of the semicircles we define β angles oriented in opposite directions to receive areas of normalization as shown in Fig. 5.24 (the right side).

The second part of the experiment included a study of the impact that the length of the radius of a person's iris has on recognition process. The length of the radius of an iris defines a segment of the iris that is used for normalization (Fig. 5.24; the left side). Such a ring does not have to start on the edge of the pupil and does not have to end on the outer edge of the iris.

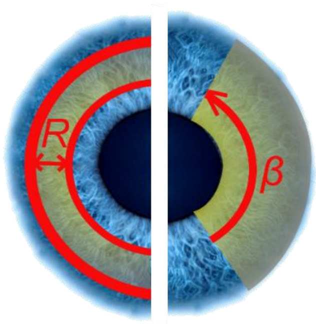


Fig. 5.24. Areas of iris image normalization

In the experiment with angular span, symmetrical areas of the iris are used. The β angles ranged from 30 to 180 degrees. Table 5.6 shows EER values of the iris recognition process for different β angles. The best iris recognition results were for β angles ranging from 120 to 180 degrees (Fig. 5.25). It can be also be inferred, that the increasing angular spanning from 120 to 180 degrees does not give f improvement. Based on this, it can be concluded that the upper and the lower parts of the iris do not include significant information and are in most cases covered by lids and lashes.

Table 5.6 The EER of iris recognition process for various β angles

Angular span [degrees]	**30**	**60**	**90**	**120**	**150**	**180**
EER [%]	0.1652	0.0408	0.0118	0.0031	0.0040	0.0031

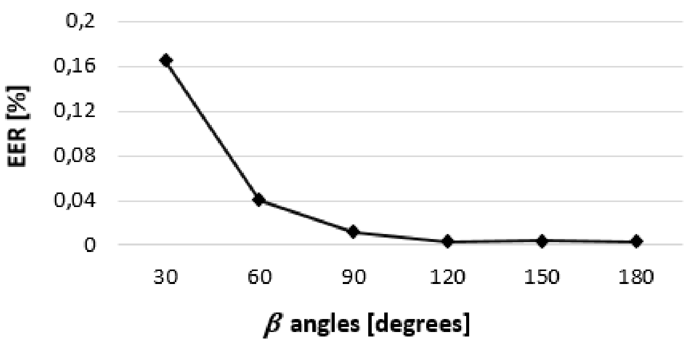


Fig. 5.25. The plot of EER of iris recognition process for various β angles

In the second step of the research, the length of the *R* radius was tested (the angular span equals to 180 degrees) in order to check the iris recognition efficiency (Table 5.7). First, the length of the radius that is increasing from the inter to the outer edge was tested. In the second step, the rings of the iris from its outer part were studied.

Table 5.7. The EER of iris recognition process for different *R* radius

Radius of iris area	r=0.1R	r=0.3R	r=0.5R	r=0.6R	r=0.8R	r=0.9R	r=R
EER [%] for increase of radius from inside to outside	0.1419	0.0329	0.0142	0.0103	0.0057	0.0019	0.0031
EER [%] for increase of radius from outside to inside	0.1453	0.0904	0.0273	0.0133	0.0034	0.0023	0.0031

The research showed that areas of iris normalization with r=0.9*R* give the best results for iris recognition. Considering the whole iris image, worse results of iris recognition were obtained.

Furthermore from Fig. 5.26 it can be deduced that the same length of radius from 0.1R to 0.5R gives better results for inner parts of the iris. This leads to the conclusion that the outer parts of iris do not contain the same amount of distinctive information as the inner parts. Another observation is the fact that far inner part of the iris can have negative impact on the person recognition based on Fig. 5.26. Such phenomenon may be caused by vicinity of pupil.

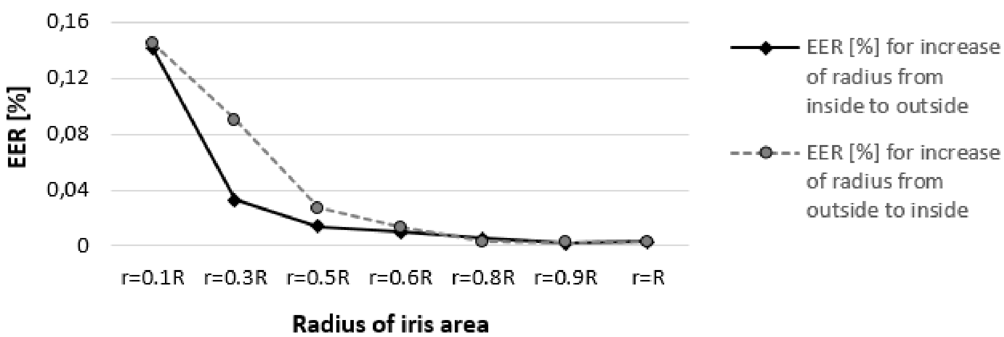


Fig. 5.26. The plot of EER of iris recognition process for various radius of iris area

Summarizing, the iris recognition process depends on acquisition precision and features extraction parameters. Data shown in this Section lead to a conclusion that the inner half of the iris area used for normalization contains more distinctive information that the outer half. Another observation is the fact that far inner and far outer parts of iris used for normalization can worsen the recognition results because of vicinity of pupil or lids and lashes.

It can be observed that the time of calculations is so short that the proposed iris recognition system can operate in real-time.

Chapter 6

Fast prototyping for intelligent video monitoring modules

This chapter presents an approach to fast prototyping of real-time event detection facilities for intelligent monitoring systems. Previously described algorithms in this dissertation can be applied in embedded devices and operated without using PC. The previously described algorithms are tested for acquisition opportunities and hardware properties and limitations.

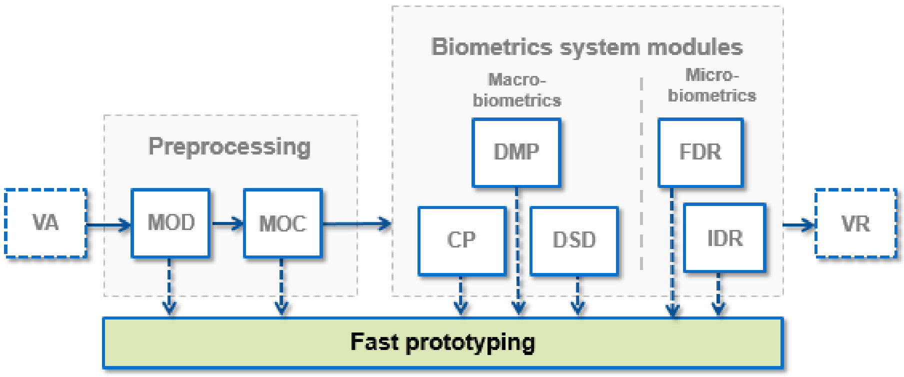


Fig. 6.1. Individual issue presented in the dissertation
– fast prototyping for intelligent monitoring modules

The aim of this study, as described in this Section, is to create systems for detecting dangerous situations in video monitoring. Two algorithms were presented as illustrative examples. The first one is responsible for the detection of a painting theft. The second one is responsible for the detection and registration of people on a pedestrian crossing at the red light [15, 16]. The real-time implementation of both models was done using the TMS320DM6437 EVM. The second model (the program for pedestrian detection) was also prototyped with the use of NI Vision Builder AI [35].

During the experiments, two cameras were used. The first one was a typical surveillance camera BOSCH LTC 0455 [151]. The LTC 0455 series cameras are compact rugged, 1/3-inch image format digital color CCD, operating with a resolution of 540 TVL. The Night Sense mode can be used for the expansion of the excellent sensitivity by a factor 3 in the monochrome operation. The video output is provided with the composite video connector. The second camera used in the experiments was Smart Camera NI 1742 (described briefly in the next Section 2.4).

6.1. Implementation of selected algorithm on Smart Camera NI 1742

6.1.1. Programming aspects of Smart Camera NI 1742

There are two methods of programming NI Smart Camera, both developed by National Instruments.

The first is NI Vision Builder for Automated Inspection [42], which is a state and model-based environment with an automatic algorithm result preview. It can also be used as a part of NI LabVIEW [152, 41].

The second method is NI IMAQdx [153] driver integrated with LabVIEW environment. It is also a model-based design development tool and it is not limited only to image processing.

The most important aspects of using different environments are stability, speed of final application and the required level of proficiency in a programming environment. The comparison of these characteristics is shown in Table 6.1.

NI Vision is an environment which is an aid to video processing, facilitates and makes image processing algorithms easier to implement. The access to functions and image transformations is very intuitive; however, it only allows the use of basic features which are sufficient to take care of basic problems. Algorithms that are more complex require extensive experience when using LabVIEW software e.g. state machine building, state comparisons, parallel processing etc. LabVIEW provides more functions and options and allows for extracting more information. The ability to write algorithms that make use of the state machines and parallel processing is a great advantage of LabVIEW over NI Vision.

Table 6.1. Comparison between NI Vision and LabVIEW basic features and some performance comparison on the basis of movement-detection algorithm [35]

Feature	NI Vision	NI LabVIEW
Image acquisition	Real time and simulated, easy to configure. Simulated acquisition can only use a sequence of pictures.	Real time and simulated. Simulated acquisition can use a sequence of pictures or video.
Storage	Real time preview, does not need configuration	Real time preview. Needs to be configured.
Stability	Good	Very good
Speed of final application	Depends on the difficulty level of inspection	Fast
Single loop execute time	Average mean time for 100 loops of movement detection algorithm: 37,57 ms	Average mean time for 100 loops of movement detection algorithm: 15,91 ms
Processor load	11%	9%
RAM memory load	178 MB	94 MB
Required experience	State and model-based design is very intuitive in most cases. Video processing knowledge is required.	Extensive knowledge of the environment is needed.

Applications developed in the NI LabVIEW environment are more complex and more powerful than in NI Vision. NI Vision allows only the use of standard components of image processing. In the LabVIEW, it is possible to create custom components or modify existing ones, rather than only to set new parameters. The LabVIEW environment enables the development of more advanced programs for a video-sequence analysis, but requires more effort, proficiency in operating the environment and design and it is more time-consuming than the design using NI Vision.

The comparison between the same model developed with two environments is presented in Table 6.1. The "Single loop execute time" was calculated as an average mean time for 100 loops of the movement detection. NI LabVIEW is 2 times faster than NI Vision, but it is also harder to implement because of the parallel processing, separated loops and more variables. The NI LabVIEW environment uses less CPU and RAM memory resources than NI Vision.

The differences and common features between these two environments are presented in Table 6.2.

Table 6.2. Comparison between NI Vision and LabVIEW's image processing algorithms [35]

Feature	NI Vision	LabVIEW
Image filtering	Filters: Smoothing, Edge Detection, Convolution	
Thresholding	Manual (threshold Range), Local Threshold (Niblack; Background Correction), Auto Threshold (Clustering, Entropy, Metric, Moments, Inter Variance)	
Morphological operation	Erode, Dilate, Open, Close (objects), Gradient In, Gradient Out, Auto Median, Thick, Thin, Remove small/large/border objects, Fill holes, Convex Hull, Skeleton, Separate objects, Label objects, Distance, Danielsson, Segment image	
Counting objects	Function: Detect Object Setup	
Measurements of objects	Measure Intensity, Measure Colors, Count Pixels, Caliper, Geometry; Impossible to measure the length of edges	
Movement detection	Only subtracting reference frame from the current movie frame.	Optical flow and differential methods
Direction and velocity detection	Not implemented	Possible, not implemented directly
Prohibited areas	Function: Create ROI and movement detection (option)	Possible, not implemented directly

6.1.2. Analysis of selected algorithm realization

In this example, the algorithm for people-detection at the red light on a pedestrian crossing is shown, however, with the use of National Instruments Automatic inspection for Vision Builder software.

In this scene, the regions of interest usually include a part of the sidewalk, the pedestrian pass, white stripes on the road, traffic lights and the background [16]. A robust detection of people passing at the red light requires a camera located in a perpendicular direction to the axis of the road (more information about camera location and moving objects classification is described in Section 3.3).

The program begins with loading images to the program environment from a camera or from a file. Then, the acquired images are analyzed.

A further step of the algorithm is the searching for traffic lights in the image. The information about whether the light is on/off is checked for each frame. Fig. 6.2 shows the result of the Pattern-Matching function, which verifies the presence of a red light in the traffic light in image. The green square in Fig. 6.2 designates the selected ROI. The red square determines whether the pattern has been established correctly. The algorithm uses monochrome images and a pattern-matching algorithm, the information about the color of the image (and the traffic lights) is not used.

The next stage of the algorithm is the detection of movement on pedestrian crossing. The program detects motion in a given region (Fig. 6.3).

Fig. 6.2. The result of a Pattern-Matching function, which verifies the presence of a lit red light in the image [35]

Fig. 6.3. Detection of moving objects [35]

Following motion detection, the shape of the discovered object is determined. Object measurement is used to distinguish two kinds of situations: the object is vertically (considered as a moving person) or horizontally oriented (considered as a moving vehicle). To calculate the coefficients of shape, two methods can be used:

- Caliper and Calculator functions [42]. The operation consists of the Feret diameters measured horizontally and vertically and dividing one by another. An example is shown in Fig. 6.4.
- Detect Object function - detects objects in a binary image in a certain region - in this case, it is a pedestrian crossing – giving the values such as: Number of Objects, their (X,Y) Position, Size, Orientation, Aspect Ratio and Number of Holes in objects. The result of Detect Object function is shown on Fig. 6.5. It is important to avoid considering small-scale objects.

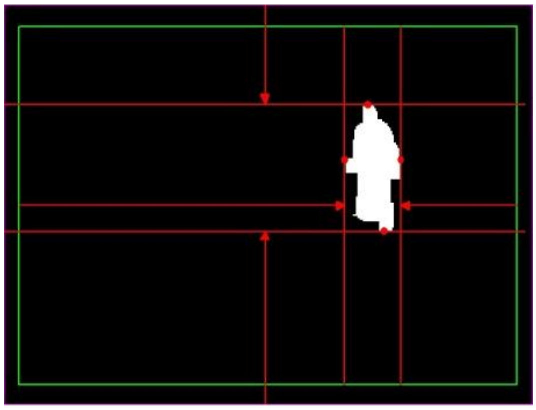
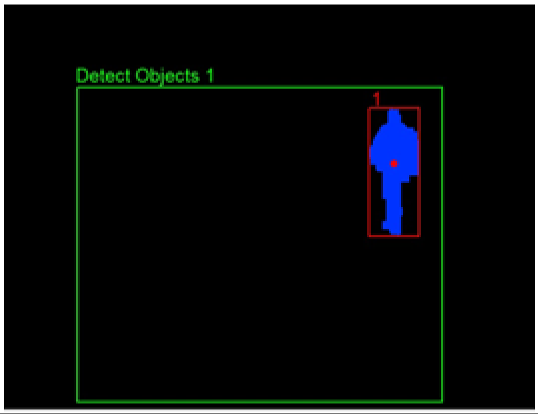


Fig. 6.4. The result of the two Caliper (horizontal and vertical) functions [35]



Objects	Number of Objects Found: 1					
No.	X	Y	Size	Orient.	Aspect	Holes
1	234,00	94,14	1226	88,32	0,395	0

Fig. 6.5. Result of Detect object function [35]

In order to verify what type of object is currently in a pedestrian area, an additional aspect of the program was developed and is shown in the Fig. 6.6. This model checks the presence of a person at a pedestrian crossing.

The algorithm reports the "danger" state when a person is detected on the pedestrian crossing ("Calculator 1 – Step Status" calculated with the use of Feret diameters) and the red light is detected on the traffic lights ("Match Pattern") at the same time (Fig. 6.7).

The entire fast prototyped model of the presented algorithm for detection of people on the red light at a pedestrian crossing developed in Vision Builder is shown on Fig. 6.8.

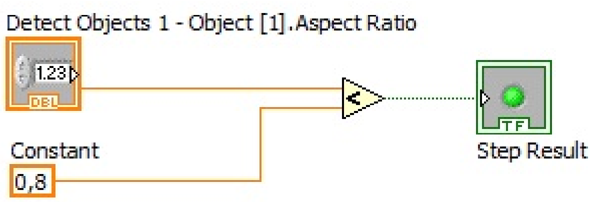


Fig. 6.6. Model based design for calculating shape criterion of an object [35]

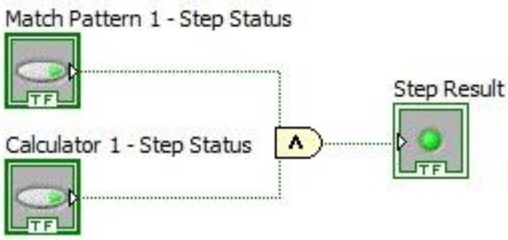


Fig. 6.7. Comparing state diagram contained in the Calculator block – helps to decide on the type of moving objects located on the pedestrian crossing during the red light [35]

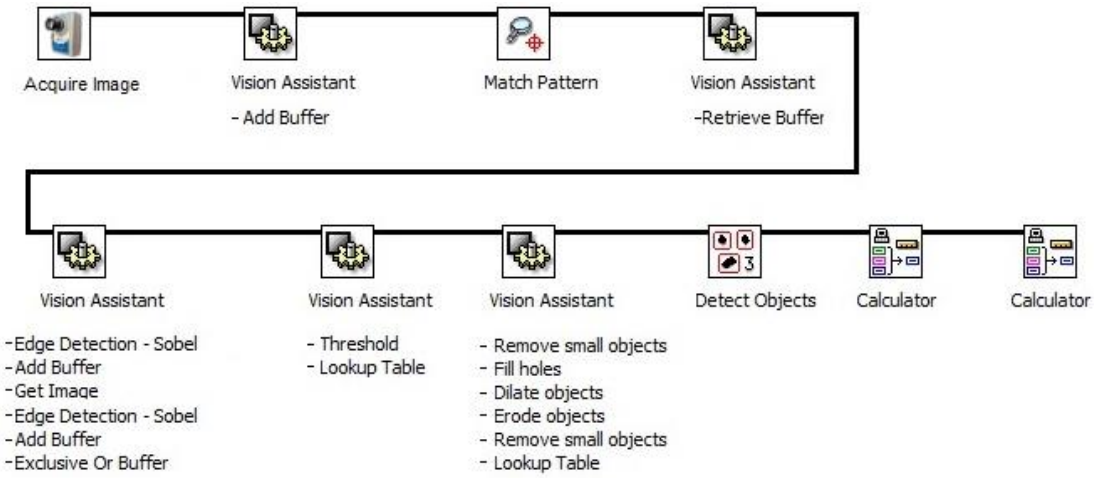


Fig. 6.8. Block diagram of a "pedestrian crossing at the red light" algorithm, realized with NI Vision Builder [35]

6.2. Fast prototyping with the use of TMS320DM6437

6.2.1. Programming aspects of TMS320DM6437

In this part of the research, the tools for rapid prototyping: Matlab / Simulink environment and C6EZFlo software from Texas Instruments are examined in conjunction with the Code Composer Studio environment. An illustrative example of algorithm implementation is also described.

For a proper implementation of the algorithms for TMS320DM6437 EVM (evaluation module) [130, 154] it is necessary to use additional libraries and blocks in the Matlab / Simulink environment that support the connection between DM6437 and CCS (Code Composer Studio) v. 3.3 software (Fig. 6.9).

The Matlab/Simulink environment provides Video and Image Processing Blockset library and other functions for processing images and video sequences. In order to communicate appropriately with TMS320EVM and C code generation, the Target Support Package C6000 is required. Basic blocks and functions allow, among others, for simple and fast changes of color spaces in the image, performing morphological operations, edge detection, color segmentation, motion detection or displaying the OSD (on screen display) messages (see Table 6.3).

The C6EzFlo 2.0 graphical environment from Texas Instruments is an embedded component of CCS ver. 5. This environment enables the generation of C code based on flowcharts. Blocks and functions available in the system are shown in Table 6.3. Most image processing operations are available for both 8 and 16-bit data.

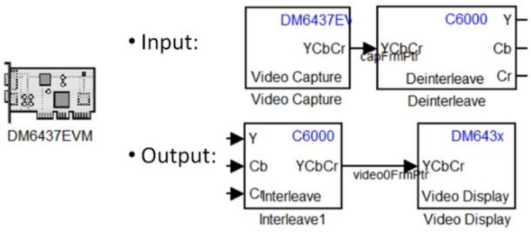


Fig. 6.9. Target Support Package C6000 library enables connection with EVM and C-language code generation

The comparison of these two mentioned environments is summarized in Table 6.3. The Table shows the availability of functions and blocks in Matlab/Simulink and C6EzFlo environments. Figures 6.10 and 6.11, respectively, show morphological operation (closing) on the binary image with the use of these two environments. The morphological closing operation is a combination of two consecutive morphological operations: dilatation and erosion.

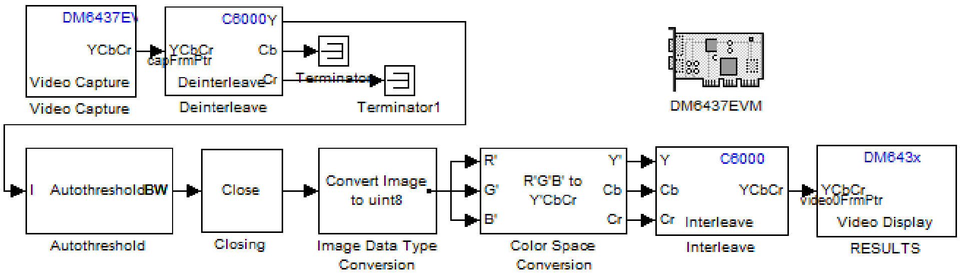


Fig. 6.10. Examples of realization of morphological closing operation built with Matlab/Simulink environment [13]

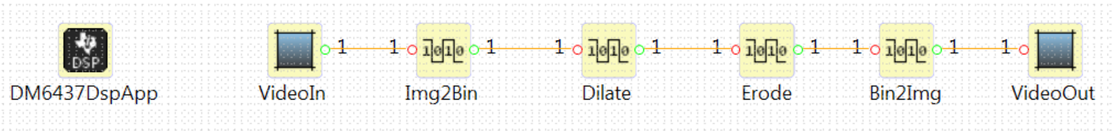


Fig. 6.11. Examples of realization of morphological closing operation built with C6EZflo environment [13]

In the Matlab/Simulink environment, in contrast to the C6EzFlo environment, it is possible to process color images. The video data is acquired in the YCbCr format. In order to convert the image to grayscale, the luminance component from the YCbCr color space was used (Fig. 6.10). In the next step, the thresholding operation is used in the monochrome image. Thanks to that, values in the image, which are above the threshold, turn on logic “1” and the values below the threshold are set to logic “0”. The binary closing operation is, then, performed using a mask (a structural element) with the dimensions of 3×3 pixels. To send the image correctly from EVM to the screen, the data format of the binary image must be changed to uint8 (unsigned integer), and then, the data must be converted from RGB to the YCbCr image color palette.

As far as the C6EzFlo environment is concerned, morphological closing in the binary image is performed on grayscale image, which is thresholded later (see Fig. 6.11). Video In module is used to acquire a monochrome image. In this environment, there is no block responsible for morphological closing operation, and consequently the dilatation and erosion must be performed one after the other. In the final stage, the data is converted from binary to uint8 format using Bin2Img block and then displayed in real-time using the Video Out module.

Table 6.3. Comparison of selected features of Matlab / Simulink and C6EZFlo environments [13]

Functions		Matlab / Simulink	C6EZFlo
Analysis	Calculation of perimeter and edge of the objects	Available	available
	Morphological operations:	dilatation, erosion, closing, opening, label	dilatation, erosion
	Edge detection with the use of:	Sobel; Prewitt; Roberts; Canny	Sobel
	Histogram equalization	Available	not available, histogram display only
	Thresholding	available; additionally auto-threshold	available
Filtering, conversions	Color space conversion	YCbCr to RGB; RGB to HSV; sRGB to XYZ; sRGB to Lab; RGB to intensity	no color space conversion – available processing only in grayscale
	Correlation	Available	available
	Filtering:	median; Kalman; FIR*	median
Other	Geometric transformations	resize; rotate	not available
	OSD	text; markers; shapes	not available
	Statistics	BLOB analysis; minimum; maximum; STD**; variance	not available

*FIR – finite impulse response; **STD – standard deviation

6.2.2. Analysis of selected algorithms realization

Detection of people at the red lights on a pedestrian crossing using Matlab/Simulink

The analysis of the detection of moving people through a pedestrian crossing at the red light is based on the model shown in Fig. 6.12. This model was also developed in the Matlab / Simulink environment.

A rule for detecting the red light on the traffic lights is based on the ROI designation and then the thresholding and binarization of the HSI (hue saturation intensity) color palette image. In order to provide a much higher accuracy of the detection of moving people at the pedestrian crossing, tracking in the appropriate proportions and sizes has been used in order to classify moving object (see Section 3.3). For threat detection, both conditions mentioned above must be fulfilled simultaneously. More information about this algorithm can be found in Section 4.3.

For better usage and visualization, the algorithm is additionally equipped with features responsible for notifying the user's system about changes in the colors of the traffic lights. The user is informed about the current color of the light and the exact count of the number of people on the pedestrian crossing while the red light was on. When people pass through the pedestrian crossing in an illegal instant, the algorithm will insert the text messages: "RED LIGHT" and "NUMBER OF PEOPLE: x".

Theft detection algorithm (at the museum)

The research related to the detection of theft is illustrated by the picture 'theft at the museum'. Fig. 6.13 presents this model designed in the Matlab / Simulink environment. The main task of this algorithm is the indication of object disappearance from the camera view.

The algorithm processes the image in the following steps:

- video capturing from the camera with PAL resolution and YC_bC_r color space
- the Y color space component from the original video sequence frame is chosen in order to designate the location of the monitored object — the ROI
- next, a filtering operation is performed using the Prewitt mask – the relevant horizontal and vertical edges are detected
- then, using the Multiport selector block, the top horizontal edge from the picture was chosen and the morphological closing operation is used in order to improve the edge detection
- during the final step of the algorithm, the sum of all values is calculated and if the sum is smaller than the given threshold, the theft is assumed to be detected.

It should be pointed out that for a correct detection of theft, the upper edge of the picture was selected, due to the possibility of it being covered by a passer-by. The camera should be placed high, near the ceiling; thus, the people passing in front of the picture would not raise an alarm.

The result of the program is a real-time video with OSD messages located in the top left corner. If the object in the ROI disappears, the algorithm will show a black rectangle in the place where the picture should have been and in the top left corner of the output image the text: "THEFT - ALARM!" will be inserted. The EVM also allows to play an audio signal, thus, there is a possibility of a sound-alarm emission. In order to verify the presence of the image in the camera view and the proper algorithm operation (when the image is in its place), the text: "PICTURE DETECTED" is inserted.

Execution time analysis

The execution times on the TMS320DM6437 EVM were analyzed for four various input video signal resolutions, like:

- 720×576 pixels – PAL resolution acquired directly from the camera
- 240×192 pixels – image downsampled 3 times
- 144×116 pixels – image downsampled 5 times
- 80×64 pixels – image downsampled 9 times.

Five models have been taken into consideration in this research: People detection and tracking, Color segmentation, Edge detection, Color space conversion and Median filtering. Some of these models and their implementations on the DSP are shown in Appendix E.

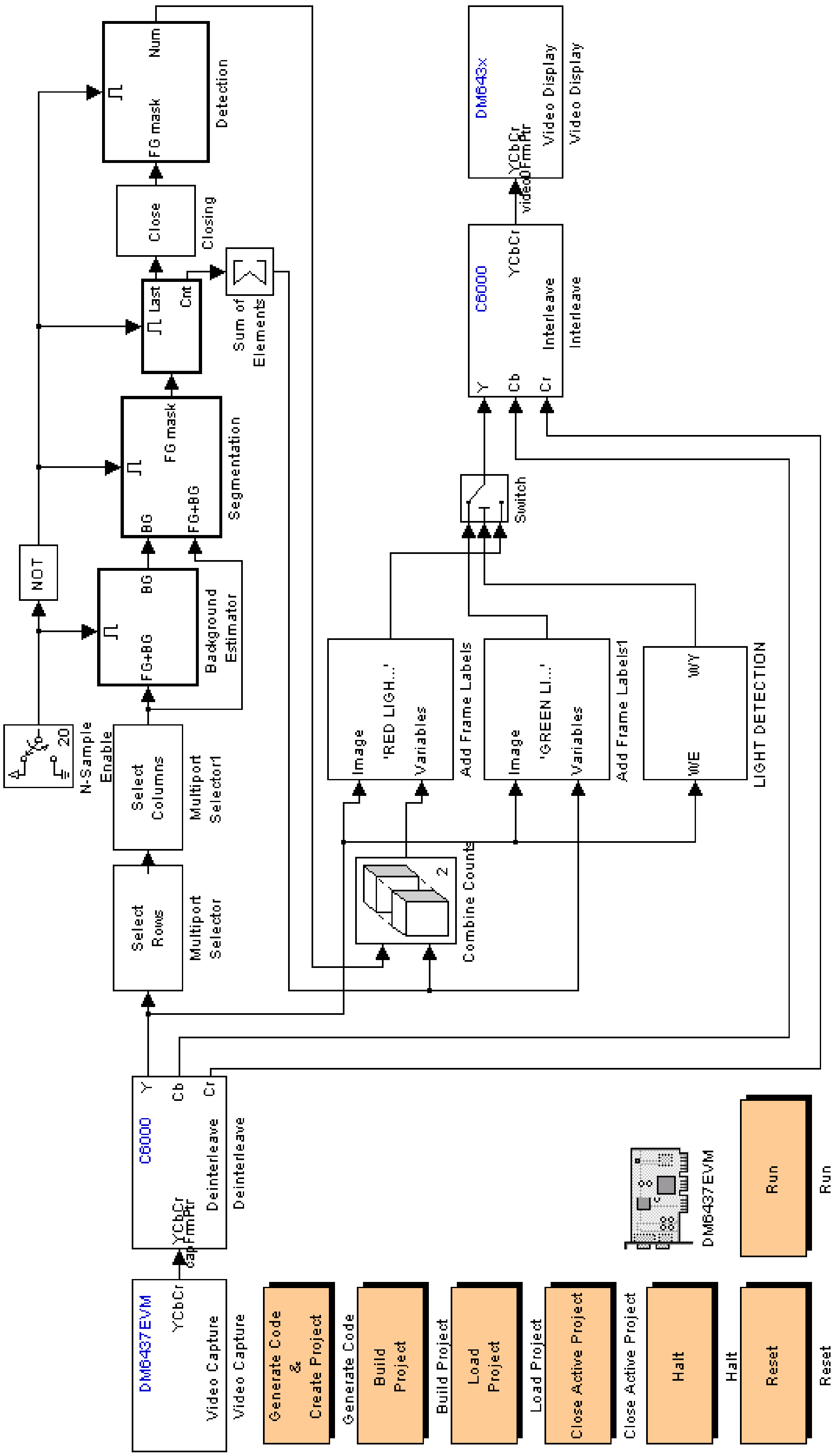


Fig. 6.12. The model of People-detection at the red light on a pedestrian crossing – Matlab/Simulink environment [34]

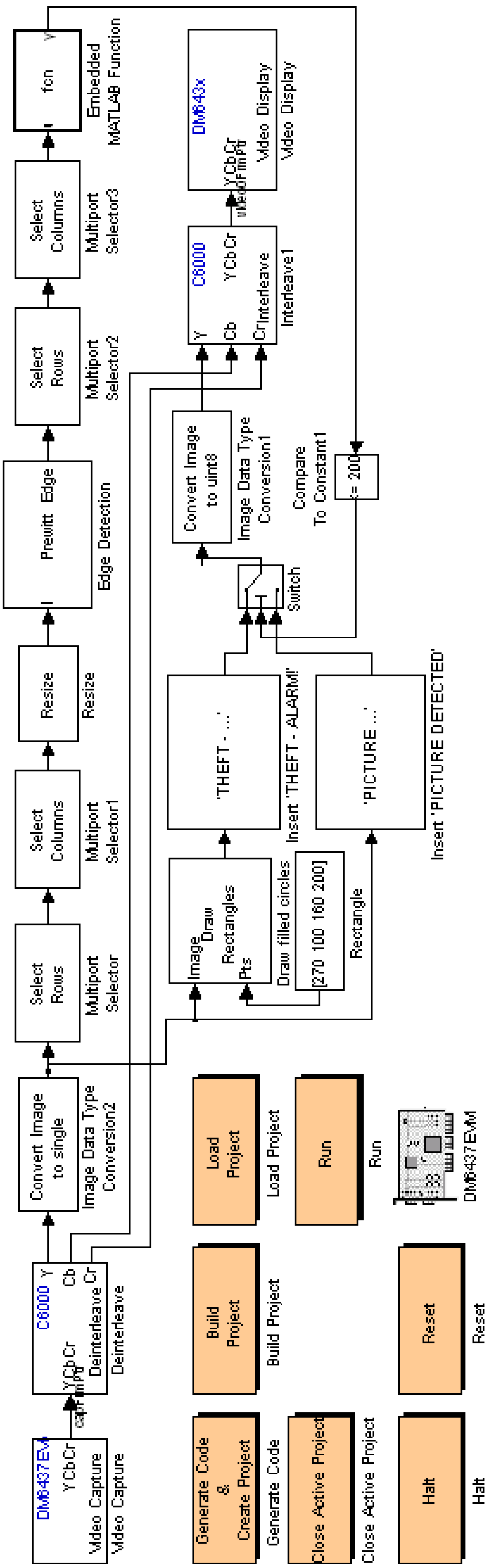


Fig. 6.13. Detection of a Picture-theft-detection at the museum – Matlab/Simulink environment [34]

The time measurements were done in the Code Composer Studio v.3.3 environment with the numbers of clock cycles counted for subsequent models. The numbers of cycles were divided by 594×10^3, which is the operating speed of the EVM module. The results are presented in Fig. 6.14. The Y axis is scaled logarithmically, which proves that changing the resolution of the image affects the execution time exponentially.

Fig. 6.14 shows which model can be performed with the DSP with a given resolution in real-time. For example, the Edge detection model could be evaluated with 38 fps for 240×192 pixels resolution, and the People detection and tracking algorithm can be evaluated with 8 fps for 144×116 pixels resolution. It should be noted that the above-mentioned models have different computational complexities.

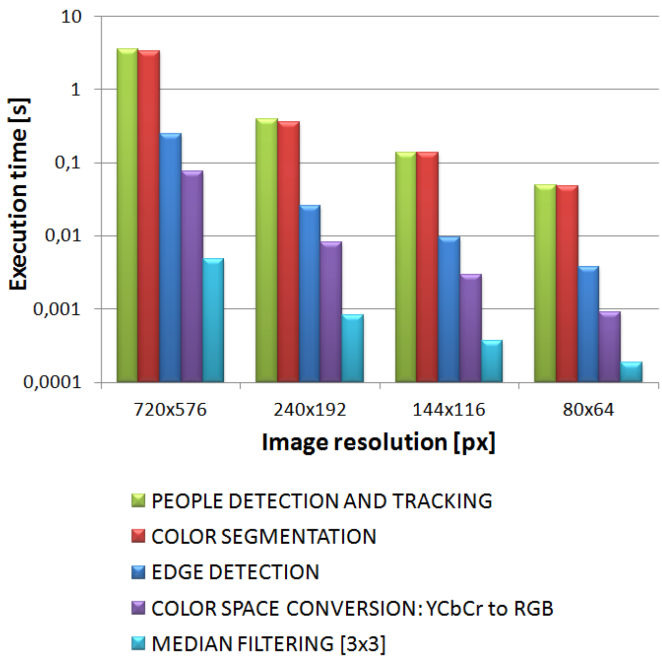


Fig. 6.14. Execution time analysis for models performed with DM6437 EVM [34]

The implementation of algorithms that support security in urban areas with the use of embedded devices, will facilitate the work of monitoring operators. Additional advantages include the prototyping speed of these embedded devices, the availability of functions and the ease of implementation. These features have a positive impact on the development of intelligent monitoring.

Chapter 7

Summary and conclusions

This PhD thesis was aimed at improving algorithms for intelligent monitoring systems used in urban areas. The research was concentrated on video processing algorithms supporting recognition of moving objects and analysis of biometric features in video surveillance. During the research, a multi-level system was designed, improved, and evaluated.

In result, solutions for the following issues have been proposed:

- adjustment of parameters of moving objects detection model in order to operate accurately in outdoor scenes as well as in the rain
- classification of moving objects into "pedestrian" and "vehicle" types, based on camera location and angle of view
- estimating the number of objects in a single BLOB, using a classifier, in situations in which people walk in close proximity to each other
- improvement of visualization quality of density maps for moving people using data normalization as well as improvement in visualization in the application of projective transformation in cases when the camera cannot be located above the observed scene
- correction of bi-directional people counting results with the use of the optimization of moving object parameters, classifier application and observation of moving objects indexes
- detecting dangerous situations with an illustrative application of the algorithm for people at a pedestrian crossing during the detection of the red light. This model uses optimization of moving object detection parameters and classification based on the type of object
- examination of the efficiency of face detection and the recognition process in various light conditions, face positions and when the face, in the image, has a small resolution
- selection of parameters of iris recognition processes
- application of innovative solutions for recognition of threats in urban areas based on fast prototyping with the use of embedded systems like smart camera and DSP.

Proper observation of objects in the CCTV is the key element in intelligent video analysis. Thus, the first issue that has been solved in this dissertation is the detection of moving objects in video sequence. It is observable, that to obtain precise results of detecting moving objects, it is required to select appropriate algorithm and their parameters. All the appropriate stages of the movement detection like the foreground/background estimation, the BLOB analysis and the tracking are important. The author of the dissertation has chosen GMM model as the method of detection of moving object. It is the best choice for a correct detection of moving object in outdoor scenes and, more so, minimize the impact of changes in lighting or weather conditions, like rainfall, slight movement caused by swaying of trees in the wind. The proposed solution operates in real-time (with 25 fps). Additional improvement of algorithm parameters has a high effect on the efficiency of detection moving objects – 94% of detected moving objects have been marked with average, good and very good detection effectiveness.

The next step of the multilevel system, analyzed in this dissertation, is classification of moving objects. This process supports the effectiveness of the algorithms in later analysis. The author concentrated on two kinds of classification. The first kind is classification into "pedestrian" and "vehicle" types. Such a type of classification allows to distinguish between two types of objects in an urban area. The parameter of this classification is strongly dependent on the camera location. Various angles of camera location (in relation to the ground) and localization in the urban area (various scenes) affects the efficiency of classification of objects. The research of these dependencies is presented in the dissertation. The second kind of classification is the prediction of the number of persons in a single BLOB in situations where people walk close to each other. The author of the research shows that a situation when 2 or more people are connected in a single BLOB, after the movement detection step, is common. From the tested database, more than 300 of 2,500 BLOB's in binary image contained 2 or 3 people. Thus, various non-linear characteristics of binary objects were examined during classifier training. Characteristics with the highest classification efficiency were selected. Additionally, parameters of the classifier was chosen to ensure the highest possible classification of objects in a single BLOB – for some cases even 92%.

The simplest way of an automatic analysis of object movement is density map generation. Thanks density mapping, the rapid analysis of specific people behaviors, e.g., evacuation ways, marketing statistics, etc., is possible. The proposed solution in the dissertation provides information about the number of people in specified location and the exact time the people were in the specific location. The processed data are normalized (for example all through the day) and adequate reference scale is selected. These modifications help the end-user of the system to, accurately, visualize the objects' movement data. The second solution of visualization of density maps data is the correction of camera positioning angle in order to obtain the best results in case of the transformation of the image from the side view to the top view of camera positioning. This solution can be very useful in areas where there is no possibility to mount the camera directed vertically downward. In the dissertation, the accuracy of projective transformation applied to the density map according to various camera angle was shown. For example, the author showed, that the shift in the vertical direction between data from cameras located at 30 degrees and 90 degrees (in relation to the ground) is only 2 pixels for images with high equals 480 pixels. This shift between the two camera views is very small and the error is 0,004 %.

The objects in motion can be also be counted. The improved system for bidirectional people counting in various (e.g., public) places is presented. The proposed modification, like counters-increment mechanism avoids multiple increments of the counters for the same tracked object. An additional algorithm based on the naïve Bayes classifier, prepared for numbering people in a single BLOB, possibly containing multiple objects, had a very important positive effect on the final people counting algorithm accuracy. The counting accuracy is at the same level as the commercial systems, i.e., about or even greater than 94% (computed using the results with the skip of video sequence containing rain). The solution regarding the location of a camera above the observed people, currently often discussed, eliminates the problem of collecting personal information. Thanks to that, the problem of personal data acquisition by vision systems, which is questionable as it could be used to identify people, becomes irrelevant. The module for counting objects can also be used on the road for vehicle counting (with the use of a classifier adequate for this purpose). This solution may have a positive impact on traffic flow and a communication infrastructure between sub-centers.

This thesis also contains the description of models for the detection of dangerous situations in urban areas. These algorithms draw attention of the monitoring system operator for dangerous situations using metadata. As there are no universal algorithms, which detect all dangerous situations in any urban area. This dissertation shows an example of detection of pedestrians during red lights at a pedestrian crossing. The algorithm accurately determines the color of the traffic light and the type of moving object (person or vehicle) at the pedestrian crossing. The combination of these data results in a low false positive rate (i.e. 5%) and a high accuracy of threat detection.

In case of the analysis of video sequence in micro-biometrics modules, the analyzed objects can have small resolution. In this dissertation, the problem of reducing resolution on both face detection and recognition stages were examined. The obtained results may find widespread usage in CCTV image analysis. The plots discussed in Section 5.1 indicate that face recognition is correct even for images as low as 21×21 pixels resolution, which means that people can be recognized from a long distance (of several meters) by using basic and standard monitoring systems. A confident image acquisition of the frontal face position can be realized, additionally, by placing the camera, for instance, on the top of a straight stairways. In the case where a face image is acquired from the camera at a short distance, a person can be recognized using an iris image.

In this dissertation, real-time implementation of algorithms in embedded systems was done as illustrative examples. The first algorithm is responsible for the detection of a painting theft. The second one deals with the detection and registration of people on a pedestrian crossing at the red lights. Implementation of both models was done using TMS320DM6437 EVM. The second model was also performed using Smart Camera NI 1742. The results of multiple tests allow to conclude that the use of the Matlab / Simulink environment and the Code Composer Studio require continuous code optimization to implement the system with the highest possible processing rate. It is important that the embedded system, which processes the image, runs at least at 4 fps, which is adequate to the speed of image registration in contemporary surveillance systems. For example, people detection and tracking algorithms can be evaluated with 8 fps for a resolution of 144×116 pixels in the EVM. The second tested environment, from National Instruments, gives vast capabilities for real time video processing. The NI Vision Builder may be used for a quick design of a simple algorithm construction based on basic processing methods. However, in some cases, using this environment may be unsatisfactory due to its limited functions. In which case, the NI LabVIEW environment can be used in order to develop advanced algorithms.

Summing up, the above-mentioned methods and algorithms presented in this dissertation contribute to providing a comprehensive monitoring system. Proposed and modified algorithms, for intelligent video analysis, has improved the effectiveness of CCTV systems in urban areas.

References

[1] L. W. Kille and M. Maximino, "The effect of CCTV on public safety: Research roundup," [Online]. Available: http://journalistsresource.org/studies/government/criminal-justice/surveillance-cameras-and-crime. [Accessed 17 02 2016].

[2] "IEEE Signal Processing Magazine," *Special Section - Video analytics for surveillance,* vol. 27, 2010.

[3] F. Porikli, F. Bremond, S. L. Dockstader, et al., "Video surveillance: past, present and now the future," *IEEE Signal Processing Magazine,* vol. 30, no. 3, 2013.

[4] European Forum for Urban Security, "Citizens, Cities and Video Surveillance," 2010.

[5] "The Video Surveillance Industry – Market size," [Online]. Available: cameramanager.com. [Accessed 17 01 2016].

[6] "Description of Cisco IP Network-Centric monitoring system," [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/physical-security/video-surveillance-services-platforms/prod_white_paper0900aecd804a3e89.pdf. [Accessed 02 12 2015].

[7] G. C. Robb, "Police Use of CCTV Surveillance: Constitutional Implifications and Proposed Regulations," *University of Michigan Journal of Law Reform,* p. 572, 1980.

[8] A. Chellappa, A. Roy-Chowdhury and S. Zhou, "Recognition of Humans and Their Activities Using Video," Morgan & Claypool, 2006.

[9] M. Ramanathan, W. Y. Yau and E. K. Teoh, "Human Action Recognition with Video Data: Research and Evaluation Challenges," vol. 44, no. 5, pp. 650-663, 2014.

[10] J. Velten, A. Kummert, A. Gavrilidis, et al., "D. 7.6 – Preliminary multimodal features detection system prototype," European Seventh Framework Programme, FR 7: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, 2012.

[11] C. S. Regazzoni, A. Cavallaro, Y. Wu, et al., "Video Analytics for Surveillance: Theory and Practice," *IEEE Signal Processing Magazine,* vol. 27, pp. 16-17, 2010.

[12] A. Aradas, "US-Mexico border: Efforts to build a virtual wall," [Online]. Available: http://www.bbc.com/news/technology-19409682. [Accessed 30 03 2016].

[13] A. Chmielewska, T. Marciniak and A. Dabrowski, "Fast prototyping of video processing algorithms in a digital signal procesor system (Szybkie modelowanie algorytmów przetwarzania sekwencji wideo w systemie z procesorem sygnałowym)," *National Conference of Electronics (Krajowa Konferencja Elektroniki),* pp. 864-869, 2013.

[14] A. Dąbrowski, "From micro-biometrics to macro-biometrics: methods, electronics and application (Od mikrobiometrii do makrobiometrii: metody, układy elektroniczne i zastosowania)," *National Conference of Elektronics (Krajowa Konferencja Elektroniki),* pp. 539-545, 2015.

[15] A. Chmielewska, P. Pawłowski and A. Dabrowski, "Application of video sequence processing methods to support the CCTV (Zastosowanie metod przetwarzania sekwencji wideo do wspomagania monitoringu miejskiego," *The theory and*

application of computer science (Teoria i zastosowanie informatyki), vol. 9, no. 3, pp. 65-83, 2011.

[16] J. Balcerek, A. Chmielewska, A. Dabrowski, et al., "Recognition of threats in urban areas by means of the analysis of video sequences," *Multimedia Communications, Services and Security,* pp. 41-48, 2010.

[17] T. Marciniak, A. Chmielewska, R. Weychan, et al., "Fast prototyping of automatic real-time event detection facilities for video monitoring using DSP module," *20th International Conference Mixed Design of Integrated Circuits and Systems,* pp. 473-478, 2013.

[18] "Description of INDECT project," [Online]. Available: http://www.indect-project.eu. [Accessed 15 06 2016].

[19] Members of the 7th framework program of the UE: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, "D.7.2 - Creation of event model in order to detect dangerous events," 2010.

[20] Members of the 7th framework program of the UE: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, "D. 7.3 – Biometric features analysis component based on video and image information," 2010.

[21] Members of the 7th framework program of the UE: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, "D. 7.4 – Multimodal features analysis component," 2011.

[22] Members of the 7th framework program of the UE: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, "D. 7.5 – Prototype of automatic event detection system," 2012.

[23] Members of the 7th framework program of the UE: Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment, "D. 7.6 – Preliminary multimodal features detection system prototype," 2013.

[24] A. Chmielewska, A. Dąbrowski, T. Marciniak, et al., "The automatic counting of objects in the urban area (System automatycznego liczenia obiektów w ruchu miejskim)," *The theory and application of computer science (Teoria i zastosowanie informatyki),* vol. 10, no. 2, pp. 83-93, 2012.

[25] A. Chmielewska, M. Parzych, T. Marciniak, et al., "New approach to traffic density estimation based on indoor and outdoor scenes from CCTV," *Foundations of Computing and Decision Sciences,* vol. 40, no. 2, pp. 119-132, 2015.

[26] T. Marciniak, A. Chmielewska, A. Dabrowski, et al., "People counting vision system based on ARM procesor programmed using Simulink environment," *Electronics,* no. 6, pp. 55-59, 2014.

[27] A. Chmielewska, T. Marciniak and A. Dabrowski, "Improved video-based people counting algorithm using BLOB classification," *Bulletin of the Polish Academy of Sciences (to be published).*

[28] A. Chmielewska, T. Marciniak and A. Dabrowski, "Application of the naive Bayes classifier for bidirectional people counting," *International Journal of Applied Mathematics and Computer Sciences (to be published).*

[29] T. Marciniak, A. Chmielewska, R. Weychan, et al., "Influence of low resolution of images on reliability of face detection and recognition," *Multimedia Tools and Applications,* vol. 74, no. 12, pp. 4329-4349, 2013.

[30] T. Marciniak, A. Dąbrowski, A. Chmielewska, et al., "Face Recognition from Low Resolution Images," *Communications in Computer and Information Science,* no. 287, pp. 220-229, 2012.

[31] T. Marciniak, A. Dabrowski, A. Chmielewska, et al., "Selection of parameters in iris recognition system," *Multimedia Tools and Applications,* no. 68, pp. 193-208, 2014.

[32] T. Marciniak, P. Pawłowski, A. Dąbrowski, et al., "The selection of hardware and software elements of iris image acquisition system for people identification (Dobór elementów sprzętowo-programowalnych w systemie akwizycji obrazu tęczówki do celów identyfikacji osób)," *Electrical Review,* no. 11b, pp. 18-22, 2012.

[33] T. Marciniak, A. Dabrowski, A. Chmielewska, et al., "Analysis of Particular Iris Recognition Stages," *Communications in Computer and Information Science,* vol. 149, pp. 198-206, 2011.

[34] A. Chmielewska, R. Weychan, T. Marciniak, et al., "Fast Prototyping for Video Monitoring Systems with the Use of DSP Module," *International Journal of Electronics and Telecommunications,* vol. 59, no. 4, pp. 375-381, 2013.

[35] A. Chmielewska, A. Dabrowski, A. Namerla, et al., "Comparison of NI LabVIEW and NI Vision Builder AI environments in fast prototyping of video processing algorithms for CCTV using smart camera," *Electronics – constructions, technologies, applications,* no. 5, pp. 72-76, 2011.

[36] "Matlab/Simulink environment description, Mathworks Inc.,," [Online]. Available: http://www.mathworks.com. [Accessed 16 11 2015].

[37] "Computer Vision System Toolbox; Matlab environment," [Online]. Available: http://www.mathworks.com/help/vision/index.html. [Accessed 01 10 2015].

[38] A. Bradski, "Learning OpenCV, computer vision with opencv library; software that sees”," 2008.

[39] "Code Composer Studio IDE Overview," Texas Instruments, [Online]. Available: http://www.ti.com,. [Accessed 16 11 2015].

[40] "C6EZFlo Graphical Development Tool for TMS320C6000 Devices," [Online]. Available: http://www.ti.com/tool/c6flo-dsptool. [Accessed 08 11 2016].

[41] "LabVIEW User Manual," National Instruments, [Online]. Available: http://www.ni.com/pdf/manuals/320999e.pdf. [Accessed 20 11 2015].

[42] "NI Vision Builder for Automated Inspection Tutorial," [Online]. Available: http://www.ni.com/pdf/manuals/373379f.pdf. [Accessed 20 11 2015].

[43] "Description of boxplot function from Matlab environment," [Online]. Available: http://www.mathworks.com/help/symbolic/mupad_ref/plot-boxplot.html. [Accessed 01 10 2015].

[44] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* pp. 861-874, 2006.

[45] N. A. Macmillan and C. D. Creelman, "Detection Theory – A User's Guide," 2005.

[46] B. Karasulu, "Review and evaluation of well-known methods for moving object detection and tracking in videos," *Journal of Areonautics and Space Technologies,* vol. 4, no. 4, pp. 11-22, 2010.

[47] M. M. Azab, H. A. Shedeed and A. S. Hussein, "A new technique for background modeling and subtraction for motion detection in real-time videos," *International Conference on Image Processing,* pp. 3453-3456, 2010.

[48] Z. Lijing and L. Yngli, "Motion human detection based on background subtraction," *2010 Second International Workshop on Education Technology and Computer Science,* pp. 284-287, 2010.

[49] "Motion detection model description," [Online]. Available: http://www.mathworks.com/examples/simulink-computer-vision/mw/vision_product-vipmotion-motion-detection. [Accessed 30 05 2016].

[50] C. Linkai, Z. Pinwei and Z. Guangping, "Moving objects detection based on background subtraction combined with consecutive frames subtraction," *International Conference on Future Information Technology and Management Engineering,* pp. 545-548, 2010.

[51] P. N. Pathirana, A. E. K. Lim, J. Carminati, et al., "Simultaneous estimation of optical flow and object state: A modified approach to optical flow calculation," *International Conference on Networking, Sensing and Control,* pp. 634-638, 2007.

[52] Y. Rui, X. Song and S. Yan, "Moving object detection based on an improved gaussian mixture background model," *International Colloquium on Computing, Communication, Control, and Management,* vol. 1, pp. 12-15, 2009.

[53] H. Nallaivarothayan, D. Ryan, S. Denman, et al., "An evaluation of different features and learning models for anomalous event detection," *International Conference on Digital Image Computing: Techniques and Applications,* pp. 1-8, 2013.

[54] S. Rivollier, J. Debayle and J. Pinoli, "Shape representation and analysis of 2D compact sets by shape diagrams," *International Conference on Image Processing Theory Tools and Applications,* pp. 411-416, 2010.

[55] X. Song, L. Wang, H. Wan, et al., "Detection and identification in intelligent traffic video monitoring system for pedestrian and vehicles," *International Conference on Networked Computing and Advances Information Management,* pp. 181-185, 2011.

[56] "Inria Person database," [Online]. Available: http://pascal.inrialpes.fr/data/human/. [Accessed 24 04 2016].

[57] S. Osowski, "Methods and tools for data mining (Metody i narzędzia eksploracji danych)," 2013, pp. 85-96.

[58] A. Chmielewska, T. Marciniak and A. Dąbrowski, "Separation of moving objects for automatic people counting," *National Conference of Electronics (Krajowa Konferencja Elektroniki),* pp. 591-596, 2015.

[59] C. Thou-Ho, C. Tsong-Yi and C. Zhi-Xian, "An Intelligent People-Flow Counting Method for Passing Through a Gate," *2006 IEEE Conference on Robotics, Automation and Mechatronics,* pp. 1-6, 2006.

[60] A. Albiol, I. Mora and V. Naranjo, "Real-time high density people counter using morphological tools," *IEEE Transactions on Intelligent Transportation Systems,* vol. 2, no. 4, pp. 204-218, 2001.

[61] T. Marciniak, A. Dabrowski, A. Chmielewska, et al., "Real-Time Bi-Directional People Counting Using Video Blob Analysis," *IEEE New Trends in Audio and Video / Signal Processing Algorithms, Architectures, Arrangements and Applications,* pp. 161-166, 2012.

[62] M. Parzych, A. Chmielewska, T. Marciniak, et al., "Automatic people density maps generation with use of movement detection analysis," *6th International Conference on Human System Interaction,* pp. 26-31, 2013.

[63] "CCTV blog, Visual Analysis of Pedestrians," [Online]. Available: https://smalltown2k.wordpress.com/2014/04/10/visual-analysis-of-pedestrians. [Accessed 03 11 2015].

[64] "Software for tracking shoppers," [Online]. Available: http://www.cnet.com/videos/software-tracks-shoppers-to-find-stores-hot-zones. [Accessed 03 11 2015].

[65] "MxAnalytix tool for heat map creation and people counting," [Online]. Available: http://www.mobotix.com/eng_AU/Support/Manuals/Software2/Compact-Guides. [Accessed 03 11 2015].

[66] "Bosch IVA 5.60 datasheet," [Online]. Available: http://resource.boschsecurity.us/documents/DS_IVA_5_60_Data_sheet_enUS_11858668555.pdf. [Accessed 03 11 2015].

[67] "Heatmap tool from Cognimatics datasheet," [Online]. Available: http://www.cognimatics.com/downloads/trueview-heatmap-product-sheet.pdf. [Accessed 14 02 2014].

[68] M. H. Sharif, N. Ihaddadene and C. Djeraba, "Crowd Behaviour Monitoring on the Escalator Exits," *11th International Conference on Computer and Information Technology,* pp. 194-200, 2008.

[69] M. Rodriguez, I. Laptev, J. Sivic, et al., "Density-aware person detection and tracking in crowds," *2011 IEEE International Conference on Computer Vision,* pp. 2423-2430, 2011.

[70] H. Fradi and J. Dugelay, "Crowd Density Map Estimation Based on Feature Tracks," *2013 IEEE 15th International Workshop on Multimedia Signal Processing,* pp. 40-45, 2013.

[71] C. Stahlschmidt, A. Gavrilidis and A. Kummert, "Density Measurement from a Top-View Position using a Time-of-Flight Camera," *International Workshop on Multidimensional System,* pp. 193-198, 2013.

[72] N. Ihaddadene and C. Djeraba, "Real-time crowd motion analysis," *International Conference on Pattern Recognition,* pp. 1-4, 2008.

[73] M. Parzych, A. Chmielewska, T. Marciniak, et al., "New approach to people density estimation based on video surveillance," *IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications,* pp. 94-99, 2014.

[74] R. Surveillance. [Online]. Available: http://retail-surveillance.com/index.htm. [Accessed 01 10 2015].

[75] "People Counter from Honeywell system description," [Online]. Available: http://www.security.honeywell.com/me/products/video/ias/va/167518.html. [Accessed 03 11 2015].

[76] "PeCo system from Visual Tools," [Online]. Available: http://www.visual-tools.com. [Accessed 01 10 2015].

[77] C. p. descriptions. [Online]. Available: http://www.countwise.com/product. [Accessed 01 10 2015].

[78] "Video Turnstile – device for counting people in the building," [Online]. Available: http://videoturnstile.com. [Accessed 01 10 2015].

[79] O. Masoud and N. P. Papanikolopoulos, "A Novel Method for Tracking and Counting Pedestrians in Real-Time Using a Single Camera," *IEEE Trans. Vehicular Technology,* vol. 50, no. 5, pp. 1267-1278, 2001.

[80] W. Ye and Z. Zhong, "Robust People Counting in Crowded Environment," *Proceedings of IEEE International Conference on Robotics and Biometrics,* pp. 1133-1137, 2007.

[81] M. Honglian, L. Hunchuan and Z. Mingxiu, "A Real-time Effective System for Tracking Passing People Using a Single Camera," *Proceedings of the 7th World Congress on Intelligent Control and Automation,* pp. 6173-6177, 2008.

[82] C. Thou-Ho, C. Yin-Chan, C. Tsong-Yi, et al., "People Counting System for Getting In/Out of a Bus Based on Video Processing," *Eight International Conference on Intelligent Systems Design and Applications,* pp. 565-569, 2008.

[83] M. D. Breitenstein, F. Reichlin, B. Leibe, et al., "Online Multiperson racking-by-Detection from a Single, Uncalibrated Camera," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 9, pp. 1820-1833, 2011.

[84] A. B. Chan and N. Vasconcelos, "Counting People With Low-Level Features and Bayesian Regression," *IEEE Transactions on Image Processing,* vol. 21, no. 4, pp. 2160-2177, 2012.

[85] K. Kopaczewski, M. Szczodak, A. Czyzewski, et al., "A method for counting people attending large public events (online)," *Multimedia Tools and Applications,* 2013.

[86] M. Grega, A. Matiolański, A. Guzik, et al., "Automated Detection of Firearms and Knives in a CCTV Image," *Sensors,* vol. 16, no. 47, 2016.

[87] G. Szwoch, "Extraction of stable foreground image regions for inattended luggage detection," *Multimedia Tools and Applications,* vol. 75, no. 2, pp. 761-786, 2016.

[88] P. Kumarguru and S. Liew, "Fire detection algorithm using image processing techniques," *International Conference on Artificial Intelligence and Computer Science,* pp. 160-168, 2015.

[89] A. Chacon-Rodriguez, P. Julian, L. Castro, et al., "Evaluation of Gunshot Detection Algorithms," *IEEE Transactions on Circuits and Systems,* vol. 58, p. 363–373, 2011.

[90] "Biometrics market intelligence," 2009. [Online]. Available: http://www.acuitymi.com/hdfsjosg/euyotjtub/eUpdate%20August%202009.html. [Accessed 23 01 2017].

[91] R. W. Frischholz, "The face detection homepage," [Online]. Available: http://www.facedetection.com. [Accessed 04 01 2016].

[92] J. P. Kapur, "Face Detection in Color Images," EE499 Capstone Design Project, University of Washington, Department of Electrical Engineering,, [Online]. Available: http://www.oocities.org/jaykapur/face.html. [Accessed 02 01 2016].

[93] "Algorithm for face detection using Hausdorff distance: FDMver1.0," [Online]. Available: http://en.freesion.com/3865/4898908/30215154. [Accessed 02 01 2016].

[94] O. Jesorsky, K. J. Kirchberg and R. W. Frischholz, "Robust Face Detection Using the Hausdorff Distance," *BioID AG, Berlin,* 2001.

[95] R. Lienchart and M. Maydt, "An extended set of Haar-like features for rapid object detection," *International conference on image Processing,* vol. 1, pp. 900-903, 2002.

[96] "The Face Annotation Interface," vol. 04, no. http://faint.sourceforge.net, p. 01, 2016.

[97] "Haar Feature-based Cascade Classifier for Object Detection," *OpenCV 2.0 C Reference. Image Processing and Computer Vision,* 2009.

[98] M. Nilsson, "Face detection algorithm for Matlab," Blekinge Institute of Technology School of Engineering Department of Signal Processing, [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/13701-face-detection-in-matlab. [Accessed 04 01 2016].

[99] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision,* vol. 38, pp. 15-33, 2000.

[100] R. Lienhart, A. Kuranov and V. Pisarevsky, "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," *Pattern Recognition,* pp. 297-304, 2003.

[101] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Conference of Computer Vision and Pattern Recognition,* pp. 511-518, 2001.

[102] "Face recognition homepage, Algorithms," [Online]. Available: http://www.face-rec.org/algorithms. [Accessed 03 01 2016].

[103] T. Marciniak, S. Drgas and D. Cetnarowicz, "Fast Face Localisation Using AdaBoost Algorithm and Identification with Matrix Decomposition Methods," *Communications in Computer and Information Science,* vol. 149, pp. 242-250, 2011.

[104] T. Marciniak, R. Weychan, A. Chmielewska, et al., "Influence of Pose Angle on Face Recognition from Very Low Resolution Images," *IEEE New Trends in Audio and Video / Signal Processing Algorithms, Architectures, Arrangements and Applications,* pp. 177-181, 2012.

[105] S. Ravi, D. Mankame and S. Nayeem, "Face recognition using PCA and LDA: analysis and comparison," *International Conference On Advances in Recent Technologies in Communication and Computing,* pp. 6-16, 2013.

[106] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces," *Recognition Using Class Specific Linear Projection,* 1997.

[107] D. Huang, C. Shan, M. Ardabilian, et al., "Local Binary Patterns and its Application to Facial Image Analysis," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 41, pp. 765-781, 2011.

[108] M. Sharkas and M. A. Elenien, "Eigenfaces vs. Fisherfaces vs. ICA for face recognition; a comparative study," *9th International Conference on Signal Processing,* pp. 914-919, 2008.

[109] M. Agarwal, H. Agrawal, N. Jain, et al., "Face Recognition using principle component analysis, eigenface and neural network," *International conference on signal acquisition and processing,* pp. 310-314, 2010.

[110] "Description of iris acquisition devices: IG-AD100," [Online]. Available: http://www.neurotechnology.com/eye-iris-scanner-irisguard-ig-ad100.html. [Accessed 10 06 2016].

[111] "Oki IRISPASS-M," [Online]. Available: https://www.oki.com/en/otr/2006/n205/pdf/otr-205-R06.pdf. [Accessed 28 05 2014].

[112] "Panasonic Iris Reader BM-ET330," [Online]. Available: ftp://ftp.panasonic.com/pub/Panasonic/cctv/SpecSheets/BM-ET330.pdf. [Accessed 28 05 2014].

[113] "Iris On The Move system," [Online]. Available: https://www.sri.com/sites/default/files/brochures/iom_passport_system.pdf. [Accessed 10 06 2016].

[114] S. Sheela and P. Vijaya, "Iris Recognition Methods – Survey," *International Journal of Computer Applications,* vol. 3, no. 5, pp. 19-25, 2010.

[115] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1148-1161, 1993.

[116] "IrisAccess 4000," [Online]. Available: http://www.irisid.com/productssolutions/irisaccesssystem/irisaccess4000. [Accessed 10 06 2016].

[117] R. Wildes, "Iris recognition: An emerging biometric technology," *Proceedings of the IEEE,* vol. 85, no. 9, pp. 1348-1363, 1997.

[118] W. Boles and B. Boashash, "A human identification technique using images of the iris and Wavelet transform," *IEEE Transactions on Signal Processing,* vol. 46, no. 4, pp. 1185-1187, 1998.

[119] L. Ma, T. Tan, Y. Wang, et al., "Personal identification based on iris texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1519-1533, 2003.

[120] L. Ma, T. Tan, Y. Wang, et al., "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image Processing,* vol. 13, no. 6, pp. 739-750, 2004.

[121] C. Sanchez-Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation," *Pattern Recognition,* vol. 38, no. 2, pp. 231-240, 2005.

[122] M. Vatsa, R. Singh and A. Noore, "Reducing the false rejection rate of iris recognition using textural and topological features," *International Journal of Signal Processing,* vol. 2, no. 1, pp. 66-72, 2005.

[123] L. Yu, D. Zhang, K. Wang, et al., "Coarse iris classification using box-counting to estimate fractal dimensions," *Pattern Recognition,* vol. 38, no. 11, pp. 1791-1798, 2005.

[124] D. Monro, S. Rakshit and D. Zhang, "DCT-based iris recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, pp. 586-596, 2007.

[125] A. Poursaberi and B. Araabi, "Iris recognition for partially occluded images: Methodology and sensitivity analysis," *Journal on Advanced Signal Processing,* pp. 1-12, 2007.

[126] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, no. 1, pp. 21-30, 2004.

[127] R. Wildes, J. Asmuth, G. Green, et al., "A machine-vision system for iris recognition," *Machine Vision and Applications,* vol. 9, pp. 1-8, 1996.

[128] L. Masek, "Recognition of Human Iris Patterns for Biometric Identification," *M. Thesis, The University of Western Australia,* 2003.

[129] "Smart Cameras for Embedded Machine Vision," National Instruments, [Online]. Available: http://www.ni.com/pdf/products/us/cat_ni_1742.pdf. [Accessed 20 11 2015].

[130] "TMS320DM6437 Evaluation Module with TVP5146M2," Texas Instruments, [Online]. Available: http://www.ti.com/lit/ug/spruev6/spruev6.pdf. [Accessed 17 11 2015].

[131] "Raspberry Pi microcomputer," [Online]. Available: www.raspberrypi.org. [Accessed 01 10 2015].

[132] M. Casares and S. Velipasalar, "Light-weight salient foreground detection for embedded smart cameras," *Proc. of the CM/IEEE International Conference on Distributed Smart Cameras,* pp. 1-7, 2008.

[133] M. Haque and M. Murshed, "Panic-driven event detection from surveillance video stream without track and motion features," *IEEE International Conference on Multimedia and Expo,* pp. 173-178, 2010.

[134] M. Bramberger, R. Pflugfelder, A. Maier, et al., "Real-Time Video Analysis on an Embedded Smart Camera for Traf?c Surveillance," *10th IEEE Real-Time and Embedded Technology and Applications Symposium,* pp. 174-181, 2014.

[135] A. Dąbrowski, P. Pawłowski, P. Kardyś, et al., "Creating Automatics and Robotics Courses Using LabVIEW, NI ELVIS II, NI CompactDAQ, and NI Vision," *National Instruments Case Study Booklet, Eastern Europe,,* pp. 33-37, 2011.

[136] "The NV-GS500 camera from Panasonic specifications," [Online]. Available: http://panasonic.net/avc/camcorder/dv/global/products/nv-gs500/specifications.html. [Accessed 14 02 2014].

[137] "Description of Optical Flow for motion estimation in video model," [Online]. Available: http://uk.mathworks.com/discovery/optical-flow.html. [Accessed 30 05 2016].

[138] "Motion-Based Multiple Object Tracking model description," [Online]. Available: http://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html. [Accessed 03 11 2015].

[139] S. Mohinder and A. P. Andrews, Kalman Filtering: Theory and Practice Using Matlab, John Wiley and Sons, 2001.

[140] R. I. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," *Cambridge University Press,* pp. 39-43, 2014.

[141] A. Chmielewska, P. Walkowiak, M. Parzych, et al., "Application of the projective geometry in the density mapping based on CCTV monitoring," *IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications,* pp. 179-184, 2015.

[142] W. Zou and P. Yuen, "Very Low Resolution Face Recognition Problem," *IEEE Transactions on Image Processing,* vol. 21, no. 1, pp. 327-340, 2012.

[143] Y. Xu and Z. Jin, "Down-sampling face images and low-resolution face recognition," *The 3rd international conference on innovative computing information and control,* pp. 392-392, 2008.

[144] M. Davis, S. Popov and C. Surlea, "Real-Time Face Recognition from Surveillance Video," *Computational Intelligence,* vol. 332, pp. 155-194, 2010.

[145] L. Rosa, "Face Recognition System 2.1," [Online]. Available: http://www.advancedsourcecode.com//face.asp. [Accessed 04 01 2016].

[146] "Description of D-Link camera," [Online]. Available: http://mydlink.dlink.com/products/DCS-930L. [Accessed 04 01 2016].

[147] Standard ISO/IEC 19794-6:2011, "Biometric data interchange formats ANSI INCITS 358-2002," 2002.

[148] T. Kaminski, "Implementation and efficiency analysis of person identification using iris recognition (Implementacja i analiza skuteczności identyfikacji osób na podstawie tęczówki)," *M.Sc. Thesis, Poznan University of Technology,* 2007.

[149] P. Kovesi, "Some of my MATLAB functions," [Online]. Available: http://www.csse.uwa.edu.au/~pk/. [Accessed 28 05 2014].

[150] D. Fiels, "Relations between the statistics of natural images and the response properties of cortical cells," *Optical Society of America,* 1987.

[151] "LTC 0455 Series Dinion Color Cameras, Bosch Security Systems," Bosch, [Online]. Available: http://resource.boschsecurity.com/documents/Data_sheet_enUS_2005022731.pdf. [Accessed 17 11 2015].

[152] R. Jamal and H. Pichlik, "LabVIEW Applications and Solutions," Englewood Cliffs, NJ: Prentice-Hall, [Online]. Available: http://www.ni.com. [Accessed 20 11 2015].

[153] "Image acquisition NI-IMAQdx software user manual," [Online]. Available: http://home.hit.no/~hansha/documents/lab/Lab%20Equipment/Vision%20System/resources/NI-IMAQdx%20User%20Manual.pdf. [Accessed 20 11 2015].

[154] T. Marciniak, D. Jackowski, P. Pawlowski, et al., "Real-time people tracking using DM6437 EVM," *IEEE Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings,* pp. 116-120, 2009.

[155] *PN-EN 50132-7 norm: Alarm systems – CCTV surveillance systems for use in security applications,* 2003.

[156] ISO/IEC 1974 5, "Biometric data interchange format – Part 5: Face image data," 2005.

[157] ANSI/INCITS 385 2004, "Face Recognition Format for Data Interchange," 2004.

[158] "PETS 2009 database," [Online]. Available: www.cvg.rdg.ac.uk/PETS2009/a.html. [Accessed 14 02 2014].

[159] A. S. Georghiades, P. N. Belhumeur and D. J. Kriegma, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, 2001.

[160] "Achermann database, The face database of The University of Bern,," [Online]. Available: http://www.cs.columbia.edu/~jebara/htmlpapers/UTHESIS/node81.html. [Accessed 02 01 2016].

[161] S. Milborrow and J. N. Morkel, "The MUCT Landmarked Face Database," Pattern Recognition Association of South Africa, [Online]. Available: http://www.milbo.org/muct. [Accessed 02 01 2016].

[162] "Database of the Sheffield University," [Online]. Available: http://www.sheffield.ac.uk/eee/research/iel/research/face. [Accessed 02 01 2016].

[163] P. J. Phillips, et al., "The FERET Evaluation Methodology for Face Recognition Algorithms," *IEEE Transactions Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1090-1104, 2000.

[164] M. Grgic, K. Delac and S. Grgic, "SCface – surveillance cameras face database," *Multimedia Tools and Apllications,* 2011.

[165] Chinese Academy of Sciences, Institute of Automation, "CASIA-IrisV3," [Online]. Available: http://www.cbsr.ia.ac.cn/IrisDatabase.htm. [Accessed 09 06 2016].

[166] "Signal and Image Processing Group, University of Bath, Iris Image Database," [Online]. Available: http://www.smartsensors.co.uk/irisweb. [Accessed 28 05 2014].

[167] H. Proença, S. Filipe, R. Santos, et al., "The UBIRIS.v2: A Database of Visible Wavelength Iris Images Captured On-The-Move and At-Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 1-7, 2010.

[168] M. Dobeš and L. Machala, "Iris Database," [Online]. Available: http://www.inf.upol.cz/iris. [Accessed 28 05 2014].

[169] P. Pawlowski, K. Borowczyk, T. Marciniak, et al., "Real- time object tracking using motorized camera," *IEEE Signal Processing Algorithms, Architectures, Arrangements, and Applications,* pp. 110-115, 2009.

[170] "Description of city of Poznan monitoring system," [Online]. Available: http://www.poznan.pl/mim/main/system-monitoringu-wizyjnego-miasta-poznania,p,1443,1743.html. [Accessed 01 04 2016].

[171] A. M. Muad, A. Hussain, S. A. Samad, et al., "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," *IEEE Region 10 Conference TENCON,* vol. 1, pp. 207-210, 2004.

[172] S. Dongwook and J. Kanghyun, "Inverse Perspective Mapping based road curvature estimation," *IEEE/SICE International Symposium on System Integration,* pp. 480-483, 2014.

[173] L. Chien-Chuan and W. Ming-Shi, "Topview Transform Model for the vehicle parking assistance system," *International Computer Symposium,* pp. 306-311, 2010.

[174] S. Tuohy, D. O'Cualain, E. Jones, et al., "Distance determination for an automobile environment using Inverse Perspective Mapping in OpenCV," *Signals and Systems Conference,* pp. 100-105, 2010.

[175] H. Li, M. Feng and X. Wang, "Inverse perspective mapping based Urban road markings detection," *IEEE 2nd International Conference on Cloud Computing and Intelligent Systems,* vol. 3, pp. 1178-1182, 2012.

[176] M. Nieto, L. Salgado, F. Jaureguizar, et al., "Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation," *IEEE Intelligent Vehicles Symposium,* pp. 315-320, 2007.

[177] D. Zhang, B. Fang, W. Yang, et al., "Robust inverse perspective mapping based on vanishing point," *International Conference on Security, Pattern Analysis, and Cybernetics,* pp. 458-463, 2014.

Appendixes

APPENDIX A. Norms and standards referred in the thesis

This Appendix presents standards and norms referred to in the research. These standards concern, for example, people size in the scene in order for proper detection and recognition. The parameters regarding face and iris detection and recognition have also been described. In the second part of the Section, databases used in the research are shown.

During the detection of the moving objects the object sizes in the image are important - according to the PN-EN 50232-7:2003 (for CCTV and alarm systems) norm [155]. The object size on the screen should be in relation to the task, such as detection of people or crowd control. For the CCTV system, the image height should be at least 400 lines and the person should occupy at least 10% of the height of the image during motion detection (Fig. A.1). For crowd control, people should occupy at least 5% of the image height.

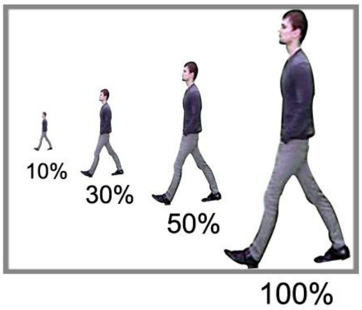


Fig. A.1. Schema of the recommended minimum size of the observed object (people) in the image according to the PN-EN 50232-7: CCTV and alarm systems norm [62]

Standards and norms for face detection and recognition

The PN-EN 50132-7 standard [155] describes development guidelines for video monitoring systems. The standard contains recommendations for the size of objects on the screen that should be related to operator tasks, such as recognition, control, detection or identification. An person in the face detection process should occupy at least 50% of the height of the screen (for CCTV systems, where the screen height is 480 lines). As far as a human body is concerned (Fig. A.2), the precise recognition of such an object is possible only when the object occupies at least 120%. Figure A2 presents a schema of the minimum size of the observed object according to this norm. The standard also suggests the resolution of face for face recognition systems, marked as VLR (very low resolution) – 32×32 or 64×64 pixels [142].

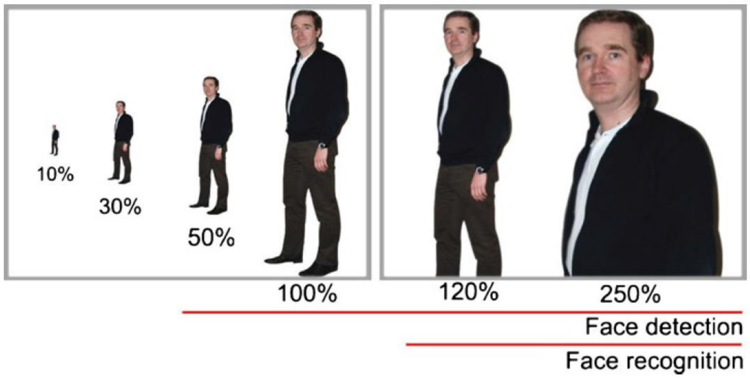


Fig. A.2. Schema of the recommended minimum size of the observed object according to European norm "EN 50132-7: CCTV and alarm systems" [29]

Most face recognition techniques and databases are prepared with the assumption that the image is normalized by ISO/IEC 19794-5:2005 [156] (Information technology – Biometric data interchange formats – Part 5: Face image data) standard or ANSI/INCITS 385-2004 [157] (Information technology – Face Recognition Format for Data Interchange) norms. These standards indicate, for example, the position, size and rotation of the face in the image or the width-to-height ratio of the image. The examples of correct and incorrect face localization in the image according to these norms are shown in Fig. A.3. The above-mentioned standards describe an example of proper face position in an image. They contain distances (in pixels) for a picture taken with 320×240 resolution. The most interesting regions have been separated (the inner region and the outer region – see Fig. A.3 (left side of the image)). The M_x line in these standards come close to horizontal midpoints of the mouth and of the bridge of the nose and the M_y line defines the line through the center of the left eye and the center of the right eye. An M point placed at the intersection of these lines defines the center of the face. The x-coordinate M_x of M should be between 45% and 55% of the image width and the y-coordinate M_y of M should be between 30% and 50% of the image height. The width of the head should be in the range between 50% and 75% of the image width, and the length of the head—between 60% and 90% of the image height. Rotation of the head should be less than about 5° from the front side in every direction—roll, pitch, and yaw. This standard also includes a width-to-height ratio of the image, which should be between 1.25 to 1.34.

Standards and norms for iris detection and recognition

The acquisition of the iris should be implemented in accordance with the standards i.e. the ANSI INCITS 358-2002 (known as BioAPI™ v1.1) recommendations. Additionally, the iris image should comply with the ISO/IEC 19794-6:2011 norm [147].

The iris picture has to be taken with a good resolution, as the photographed iris region is small, with a diameter of about 1 cm. The minimum iris diameter in the image should be 100 px and 200 px for images with high quality. Figure A.4 shows iris localization in the image. The higher the resolution, the more details are visible. The method should also not be invasive or cause discomfort to the photographed person. It has to be stressed that too much light entering the eye can cause pain. Next, the shooting area should be properly cropped in order to include only the interesting part of the eye. Finally, the acquired image should have none or just small number of reflections in the iris, as it is difficult to overcome them.

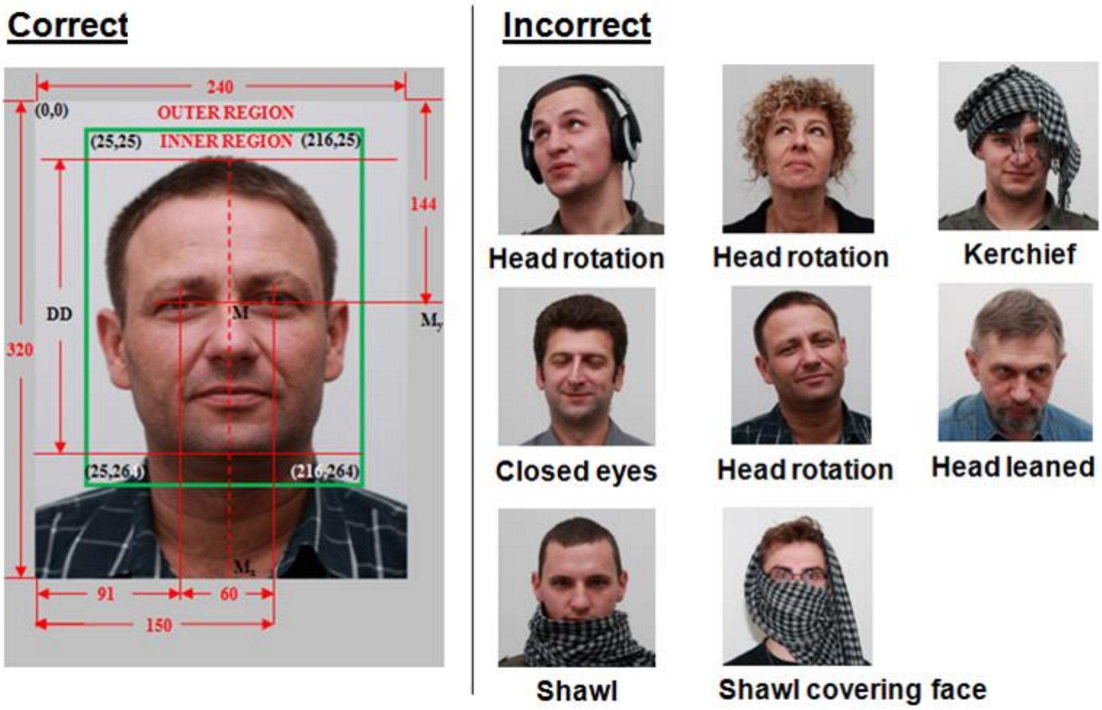


Fig. A.3. Correct and incorrect face localization in the image in the typical biometric system [30]

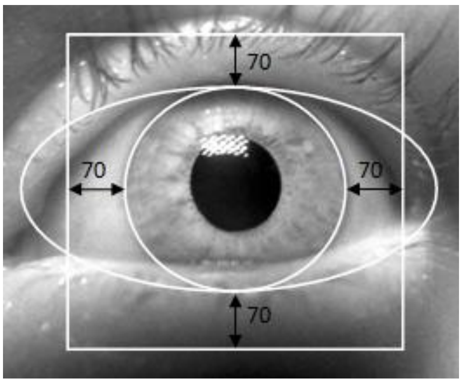


Fig. A.4. Iris localization in the image (values in pixels) [147]

APPENDIX B. Databases referred in the thesis

Database for test of moving object detection

During tests, for example, of moving objects detection, a well-known set of recordings from PETS (performance evaluation of tracking and surveillance) database [158] was used. The PETS database is used for the advanced image processing like movement detection, analysis and estimation of the human silhouette and counting the number of people in the crowd. Datasets from this database originate from multiple cameras placed in different positions and representing a variety of crowd behaviors (the number of people was about 40). The database consists of the following parts: training data, images used for person counting, people density estimation, people tracking, flow analysis of the crowd and event recognition. Each dataset contains several sequences and each sequence contains different views. Examples of scenes used in this work are shown in Fig. B.1. Cameras used for recording the datasets are Axis 223M, PTZ Axis 233D, Sony DCR-PC1000E 3xCMOS (complementary metal-oxide semiconductor) and Canon MV-1 1xCCD (charge coupled device). Resolution of the video frame is 768×576 pixels, and the frame rate is 7 fps (frames per seconds). Frames are compressed as JPEG image sequences.

Fig. B.1. Sample images from the PETS database

Face databases

In this dissertation, the public databases shared by various university research teams were used in order to test the face detection and recognition process. Below, a comparison of some performances of databases against the standards for facial biometrics is described.

One of the best-constructed databases regarding usage, ease of processing, and sorting of files in terms of features is the *Yale Face Database* [159]. This includes 5,760 images of ten people. Each of the photographed people has 576 images in 9 positions and different (64) lighting conditions. Every file in the database can be easily separated due to the clear description of file names for frontal photos and others. The grayscale pictures are of a decent quality and in high resolution (640×480 pixels). Details about databases comparison with biometric standards are shown in Table B.1.

Table B.1. Comparison of face databases in relation to the biometric standards [29]

	Biometric norm	Sheffield database	Yale database	MUCT database	FERET database
Number of individuals	-	20	10	624	1199
Total number of files	-	564	5760	3755	14126
Tested file	-	1i012.pgm	yaleB10_P00A +005E-10.pgm	i025ra-mn.jpg	00068_ 931230fb.ppm
The ratio of height to width of the image	1.25-1.34	1.09	0.75	1.131	1.48
The rotation of the head [degrees]	Smaller than 5	0	0	2	0
M_x[%]	45-55	52	53	51	50
M_y[%]	30-50	36	49	42	30
The ratio of head width to image width	0.5-0.75	0.58	0.39	0.47	0.34
The ratio of head height to image height	0.6-0.9	0.7	0.78	0.53	0.33
A statement - whether the database complies with the biometric standards	-	No. The bad aspect ratio of the picture	No. The picture is taken horizontally, so the ratio of the width of the head to the width of the image is too small	No. The face is too far from the lens	No. The face is too far from the lens and the bad aspect ratio of picture

The Achermann database [160], also called FullFaces from University of Bern in Switzerland, includes files saved in the RAS format at 512×342 resolution. This database contains grayscale images of 30 individuals, including 10 images per person (i.e., only 300 images in total). The photos were taken in constant light conditions and in various head positions – frontal, face seen in profile and face directed up and down.

The third tested database is the MUCT face database [161] from the University of Cape Town. It consists of 3755 face images of 624 subjects. Each face was photographed with the use of five different cameras at the same time. Thanks to that, five facial images with different poses were obtained. Additionally, each individual was photographed with 4 different lighting sets. The MUCT database files are saved as color images at 480×640 pixels resolution in the JPEG format.

A additional databases, concerning face detection and recognition, include the Sheffield University database (20 people with the average number of photos per person: 50) [162], Color FERET Database (14.126 images that includes 1199 individuals) [163] and SCface – surveillance cameras face database [164] (4160 static images of 130 subjects).

Basic statistics of the face databases are presented in Fig. B.2.

Iris databases

Experimental studies of iris recognition can be carried out using databases containing photos of irises prepared, for example, by scientific institutions. Two publicly available databases were used during the experiments. The first database is CASIA [165], coming from the Chinese Academy of Sciences, Institute of Automation, while the second was IrisBath [166] - developed at the University of Bath. Another iris database are UBIRIS v.2.0 database [167] and the database prepared by Michael Dobeš and Libor Machala [168].

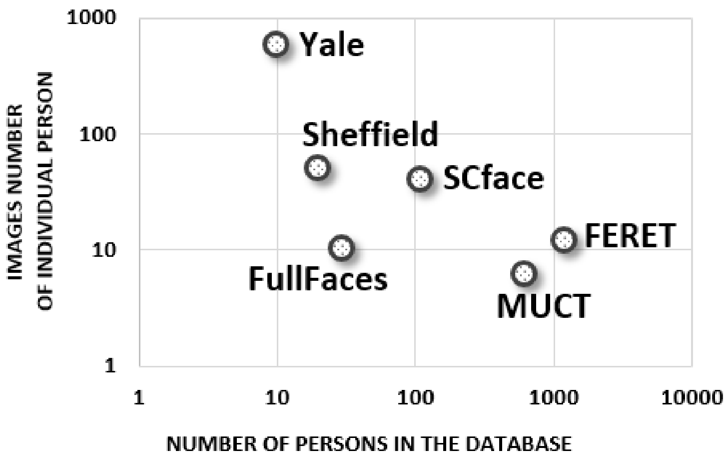


Fig. B.2. Statistics of databases used in the research for face detection and recognition

The CASIA database is presented in three versions. All the presented photographs were taken in the near IR (infrared). During the experimental research, the first and third version of this database were used.

Version 1.0 contains 756 iris images with dimensions 320×280 pixels carried out on 108 different eyes. The pictures in the CASIA database were taken using a specialized camera and saved in the BMP format. For each eye 7 photos were made, 3 in the first session and 4 in the second. The pupil area was uniformly covered with a dark color, thus eliminating the reflections occurring during the acquisition process.

The third version of the CASIA database contains more than 22 000 images from more than 700 different objects. It consists of three sets of data in the JPG 8-bit format.

Lately, a new version of the CASIA database has been created - the CASIA-IrisV4. It is an extension of CASIA-IrisV3 and contains six subsets. The three subsets from CASIA-IrisV3 are: CASIA-Iris-Interval, CASIA-Iris-Lamp, and CASIA-Iris-Twins. The three new subsets are: CASIA-Iris-Distance, CASIA-Iris-Thousand, and CASIA-Iris-Syn. CASIA-IrisV4 contains a total of 54 607 iris images from more than 1 800 genuine subjects and 1 000 virtual subjects. All iris images are 8 bit gray-level JPEG files, collected under near infrared illumination.

The IrisBath database was created by Signal and Image Processing Group (SIPG) at the University of Bath in UK. The project aimed to bring together 20 high resolution images from 800 objects. Most of the photos show the iris of students from over one hundred countries, who form a representative group. The photos were taken with a resolution of 1280×960 pixels in 8-bit BMP, using a system with camera LightWise ISG. There are thousands of free-of-charge images that have been compressed into the JPEG2000 format.

Basic statistics of the iris databases are presented in Fig. B.3.

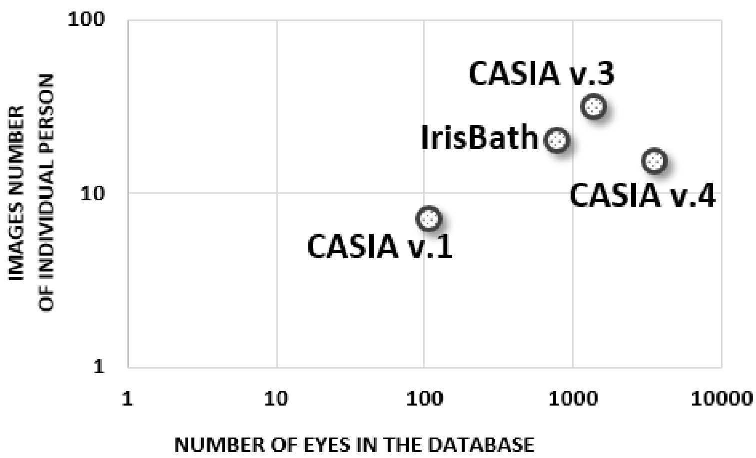
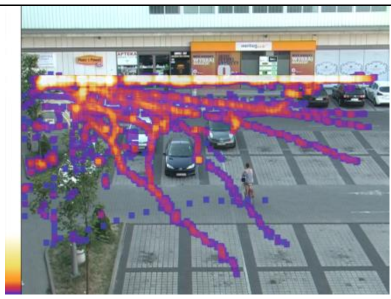
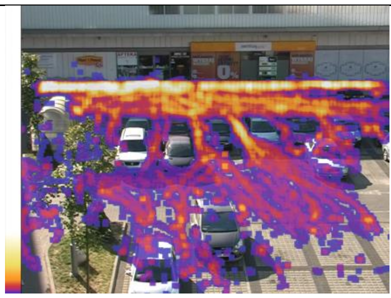
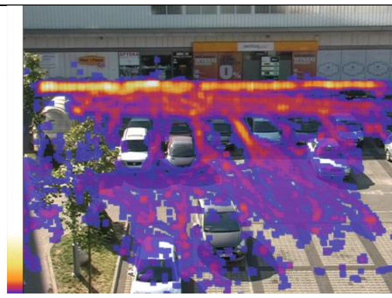
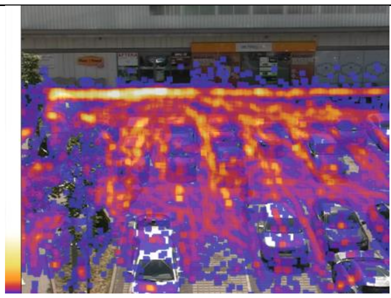
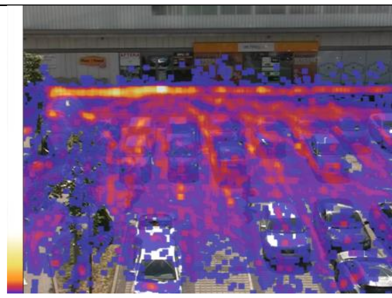
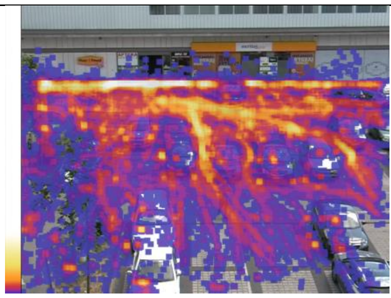
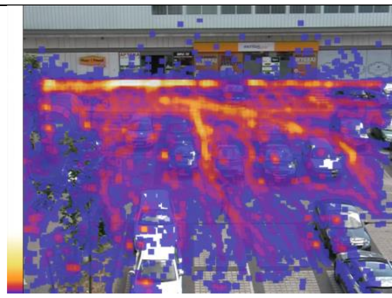
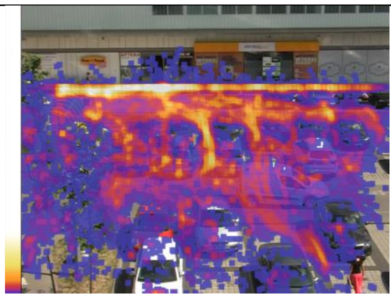
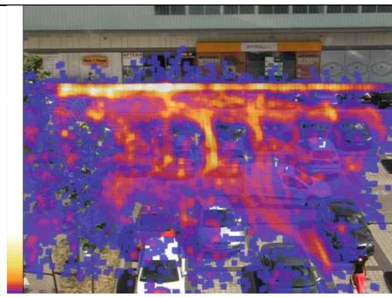
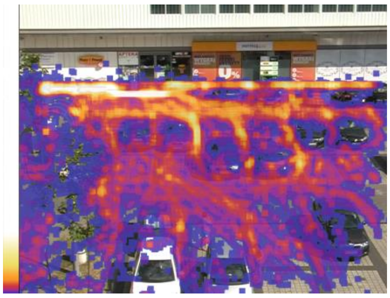
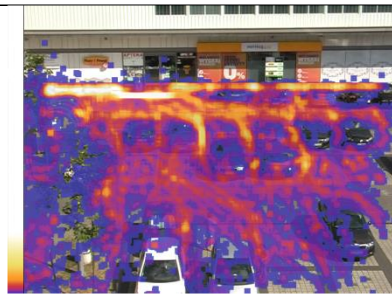
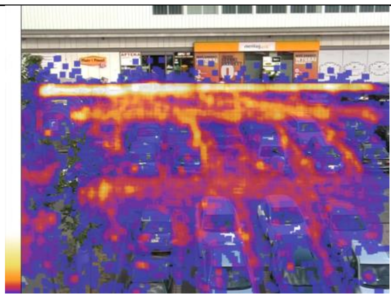
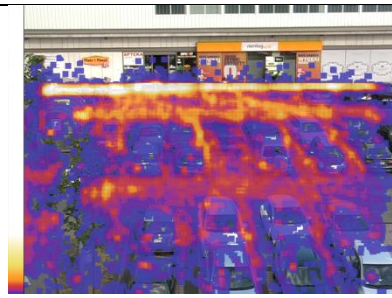
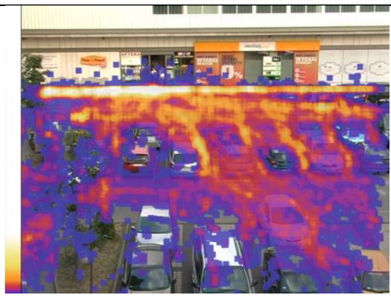
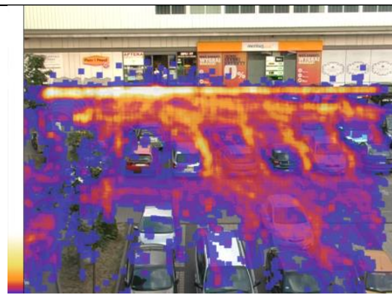


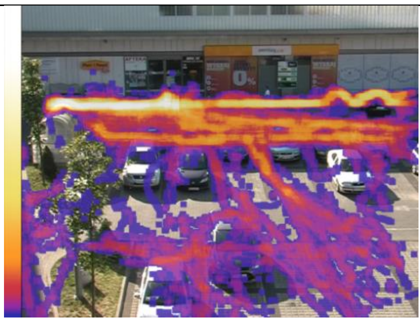
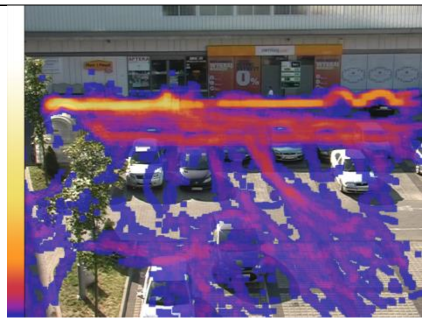
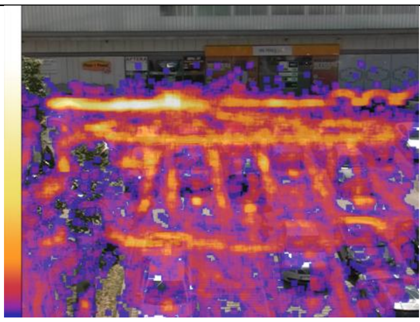
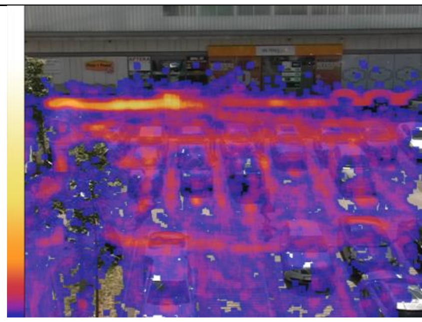
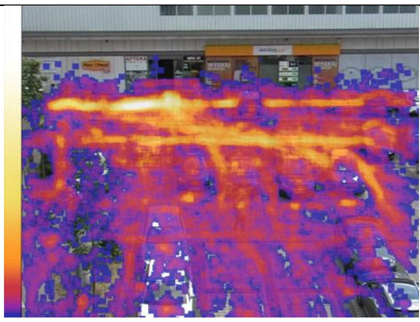
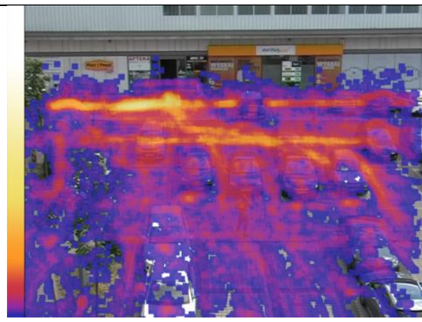
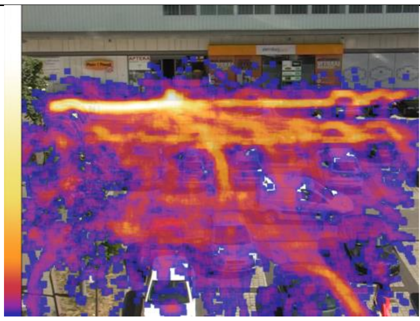
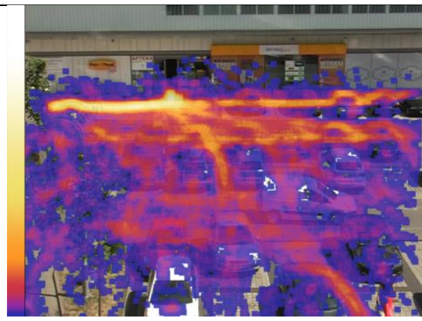
Fig. B.3. Statistics of databases used in the research for iris detection and recognition

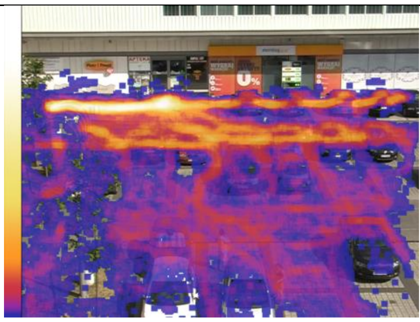
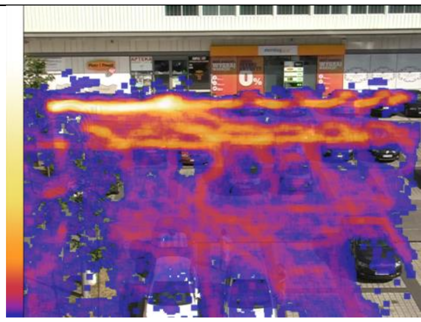
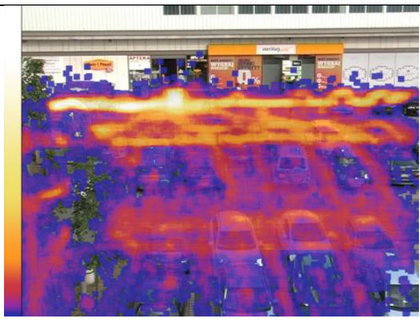
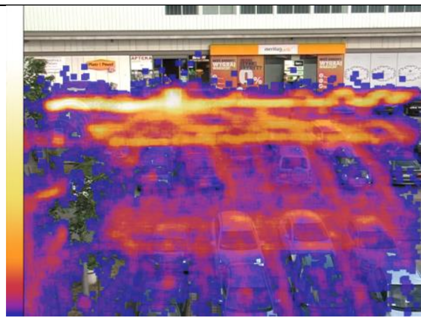
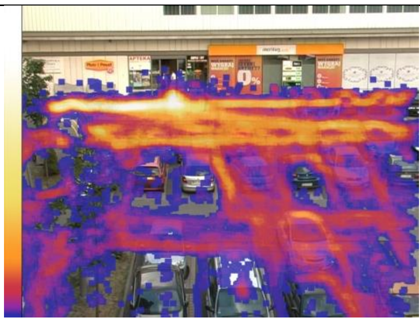
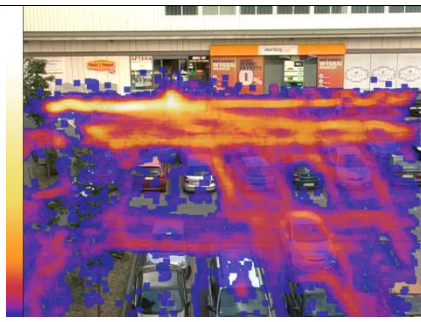
APPENDIX C. Examples of generated density maps based on time from the parking lot of the Green Point shopping center with and without data normalization [25]

Time of recor-ding	Density maps based on time without data normalization	Matrix max value [no. of frames]	Matri x max value [min]	Density maps based on time with data normalization
6:00 a.m. – 7:00 a.m.		743	0,45	
8:00 a.m. – 9:00 a.m.		2025	1,35	
10:00 a.m. – 11:00 a.m.		2442	1,62	
12:00 a.m. – 1:00 p.m.		2585	1,72	
2:00 p.m. – 3:00 p.m.		4522	3,01	

Time of recor-ding	**Density maps based on time without data normalization**	**Matrix max value [no. of frames]**	**Matrix max value [min]**	**Density maps based on time with data normalization**
4:00 p.m. – 5:00 p.m.		4232	2,82	
6:00 p.m. – 7:00 p.m.		4889	3,25	
8:00 p.m. – 9:00 p.m.		4492	2,99	

APPENDIX D. Examples of density maps based on the observation of the indexes of moving objects from the parking lot of Green Point shopping center with and without data normalization [25]

Time of recor-ding	Density maps based on moving object indexes observation without data normalization	Matrix max value	Density maps based on moving object indexes observation with data normalization
6:00 a.m. – 7:00 a.m.		30	
8:00 a.m. – 9:00 a.m.		88	
10:00 a.m. – 11:00 a.m.		78	
12:00 a.m. – 1:00 p.m.		111	
2:00 p.m. – 3:00 p.m.		134	

Time of recor-ding	Density maps based on moving object indexes observation without data normalization	Matrix max value	Density maps based on moving object indexes observation with data normalization
4:00 p.m. – 5:00 p.m.		188	
6:00 p.m. – 7:00 p.m.		187	
8:00 p.m. – 9:00 p.m.		160	

APPENDIX E. Examples of algorithms implementation on DM6437 EVM

Blocks and functions used to build the real-time software for the monitoring system, which are available in the Matlab / Simulink environment obtained from the Video and Image Processing library, can be divided into two groups: basic and advanced. Examples of these algorithms are shown in Table E.1.

Table E.1. Examples of algorithms available in the Matlab/Simulink environment [34]

Basic	Advanced
Motion detection Edge detection Color segmentation	People tracking

Motion detection model

A motion detection model from Matlab/Simulink environment makes use of the sum of absolute differences function (described in Section 2.1).

If the SAD values exceed a given threshold, the particular area (the appropriate one from the four mentioned areas) is highlighted in red color (Fig. E.1). The detected movement becomes visible in the original grayscale image. Each area is assigned with a specified color from the movement intensity chart, which is found in Fig. E.2. The threshold value is controlled in the Motion threshold block. In the example of Motion detection system, the camera is placed in a way that two bottom (from the four) areas (Fig. E.1) are directed towards the road. Thanks to that, passing cars and their speed can be easily detected. In Fig. E.2, frames with numbers around 500 show that in the video sequences, the car moved from Area II to Area IV (from the left to the right hand side in relation to the video from; areas in the image are marked as follows: Area I –top left, Area II –bottom left, Area III –top right, and Area IV – bottom right). The car movement is first noticeable in Area II and later in Area IV. The speed of the car was high (it is visible from the large SAD value). The threshold is set in such a way that the moving people in the top of the video sequence frames were undetected.

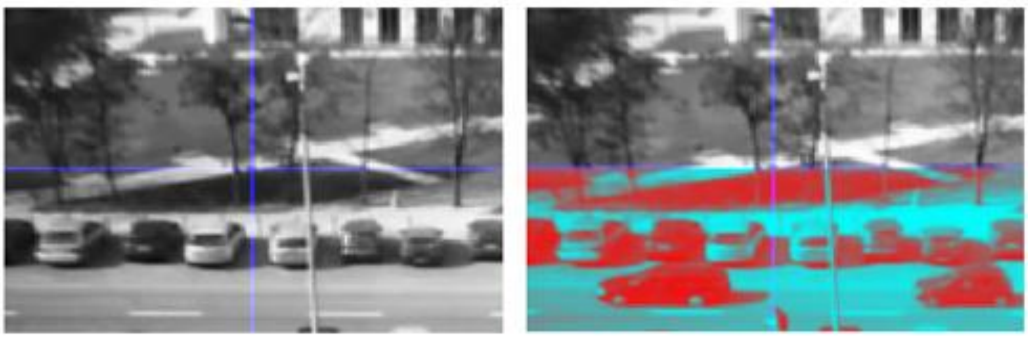


Fig. E.1. Results of Motion detection model – (from the left) the reference image and output image with the divided detection areas [34]

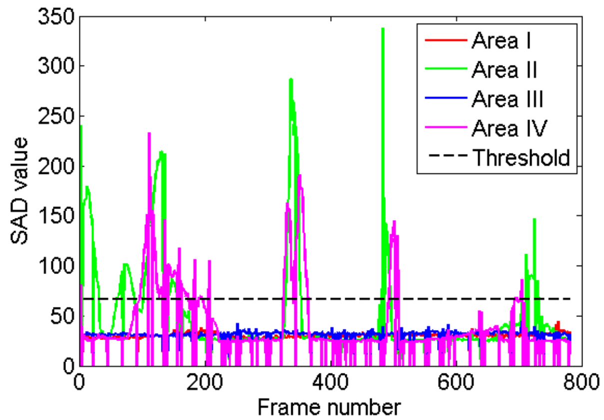


Fig. E.2. Results of the Motion detection model – the threshold and the SAD chart for the detection areas, where areas in the input image (left side of Fig. 5.17) are marked as follows: Area I – left top, Area II – left bottom, Area III – right top, and Area IV – right bottom part of the image [34]

Edge detection model

The edge detection model detects the edges of objects visible in a video sequence. During operation of the program, the image that constitutes the imposition of the input image and the binary image with detected edges are generated. In this experiment, the Prewitt method, with 15/256 threshold value and with the edge thinning option enabled, was used.

The research performed on the DSP (Section 6) shows that the algorithm can be evaluated in the real-time mode for PAL resolution (720×576 pixels) at the speed of 4 fps. However, when the original resolution was down-sampled 3 times (up to 240×192 pixels), the video speed could exceed the PAL speed (25 fps) and could be even equal to 38 fps.

Color segmentation model

The color segmentation technique is used to detect and track human faces and hands [29, 104, 30]. The algorithm is based on image processing, which includes the calculation of color to obtain the mean and covariance of color components (C_b and C_r of the YC_bC_r color palette).

Color segmentation block calculates the Mahalanobis distance (a distance between two points in a multidimensional space). The result is compared with the threshold value, enabling the detection of the skin color.

The algorithm result is a binary image, whose pixels with values equal to "1", indicate the location of a skin color in the input image (Fig. E.3). This binary skin map is compared with the original image and the face, hand or limbs are detected as marked in the output image.

Real-time face and hand detection results, performed with the DSP using the Color segmentation model, are shown in Fig. E.4. This operation (the skin detection) is evaluated with 0.3 fps for PAL resolution, thus, it is impossible to operate it in real-time for the original resolution. The input image decimated down to 240×192 pixels, gives the result of 3 fps, but the image down-sampled to 144×116 pixels allowing for a fluent image display with 8 fps.

Fig. E.3. Results of Skin detection model [34]

Fig. E.4. Results of the real-time Skin detection model performed with DM6437 EVM [34]

People tracking model

The people-tracking algorithm is designated to detect and track people in video sequences (Fig. E.5) [169]. The object tracking is executed in several stages:

- background estimation with the use of 20 consecutive frames — the Background estimator block
- separation of pixels that represent a moving object and the background — the Segmentation block
- connection of pixels, which are close to each other (morphological closing operation) and calculation of BLOB areas (Blob analysis block); adding boundary to the detected people — the Tracking block
- estimation of the location of people based on their location in previous frames — the Detection block.

The algorithm is designated to operate on sequences with 160×120 pixels resolution on TMS320DM6437 EVM with 8 fps. In order to operate on the video sequences with higher resolution, the resolution in the Reshape block (which is located in the Background estimator block) should be changed. With the TMS320DM6437 EVM, the algorithm with higher input image resolution, for example with 240×192 pixels, is performed with 2,5 fps only.

Results of real-time example of the model performed with the EVM are shown in Fig. E.6. In this example, the People tracking algorithm is executed on the grayscale image but the output data are displayed in the YCbCr color space format on the TV screen.

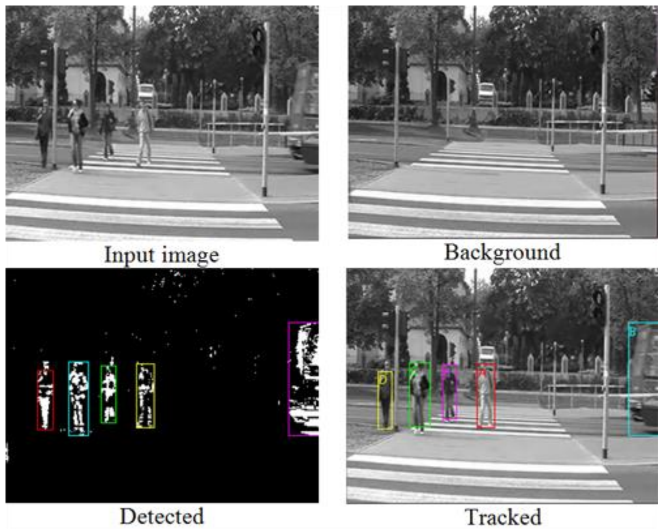


Fig. E.5. Results of the People tracking model [34]

Fig. E.6. The real-time results of the People tracking model performed with the DM6437 EVM [34]