Background Subtraction for the Detection of Moving and Static Objects in Video Surveillance

vorgelegt von
Dipl.-Ing.
Rubén Heras Evangelio
aus Valencia, Spanien

von der Fakultät IV - Elektrotechnik und Informatik
Technische Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Olaf Hellwich
Gutachter:	Prof. Dr. Thomas Sikora
Gutachter:	Prof. Dr. José María Martínez Sánchez
Gutachter:	Dr. Lutz Goldmann

Tag der wissenschaftlichen Aussprache: 21. Februar 2014

Berlin 2014
D 83

Eidesstattliche Erklärung

Ich versichere an Eides Statt, dass ich die von mir vorgelegte Dissertation selbstständig angefertigt und alle benutzten Quellen und Hilfsmittel vollständig angegeben habe.
Eine Anmeldung der Promotionsabsicht habe ich an keiner anderen Fakultät oder Hochschule beantragt.

Rubén Heras Evangelio

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Dr.-Ing. Thomas Sikora, for giving me the opportunity to join the Communication Systems Group (Nachrichtenübertragung, NUE) at the Technical University of Berlin, for his support, for his wise advise, and for caring of having such a stimulating working environment. I also want to thank Prof. Dr. José María Martínez Sánchez and Dr. Lutz Goldmann for their detailed review of my thesis, which helped me to significantly improve its quality.

During my work at NUE I have been surrounded by the most pleasant colleagues. I would like to thank them for making such a great time out of these years. Specially I would like to thank Michael Pätzold, Tobias Senst and Volker Eiselein for the fruitful discussions and for the work that we have done together in most of the research projects in which I have been working. In these projects we have been supported by very committed students to whom I am also very thankful. Furthermore, I would also like to thank the students whose master-theses I supervised, who initiated exciting discussions and provided me with very interesting observations.

This work would have not been possible without the funding provided by the several research projects in which I have been involved, starting at the regional level with MoSensNets, going national with SinoVE and international with VideoSense and MOSAIC. These projects have given me the opportunity to work with many different people working in a broad range of fields, from whom I have learned a lot and whom I would like to thank here. I also would like to thank all the people and institutions that have made possible the existence and course of these projects, specially Birgit Boldin and Dr. Ivo Keller at NUE.

Finally, I would like to thank my friends, who have followed my work with enthusiasm and provide me with their support and with fun. Thanks to O., who cares of me with love and patience. And thanks to my family, to my parents, Maria Rosa and Pepe, and to my siblings, Susana and David, who have given me the privilege of growing up with love and respect and who are always there for me. Them I owe the luck of having a rewarding and happy life.

Rubén Heras Evangelio

Abstract

The aim of the algorithms developed in this thesis is the real-time detection of moving and new static objects of arbitrary visual appearance in unconstrained surveillance environments monitored with static cameras. This is achieved based on the results provided by background subtraction. For this task, Gaussian Mixture Models (GMMs) are used. A thorough review of state-of-the-art formulations for the use of GMMs in the task of background subtraction reveals some further development opportunities, which are tackled in a novel GMM-based approach incorporating a variance controlling scheme. The proposed approach permits an easier parametrization of the models to different environments and converges to more accurate models of the scene.

The detection of moving objects is achieved by using the results of background subtraction. For the detection of new static objects, two background models learning at different rates are used. This allows for a multi-class pixel classification, which follows the temporality of the changes detected by means of background subtraction.

In a first approach, the results provided by the subtraction of both background models are used as input of a Finite-State Machine (FSM), which is used to reason on pixel classification based on the history of the pixel. This allows for the detection of new static objects over long periods of time and for a correct classification of the uncovered background regions upon their removal.

In a further developed approach, the results provided by multi-class pixel classification are analyzed at the region level in order to distinguish between new and removed static objects. This allows for the detection of static objects without previous knowledge of the observed scene. Furthermore, it is shown that the results provided by region analysis can be used to improve the quality of the background models, therefore, considerably improving the detection results.

The results provided by the developed algorithms are proved in a novel summarization application which combines multiple analysis cues in order to provide summaries that better align with the content of the analyzed video sequences.

Kurzfassung

Das Ziel der in dieser Arbeit entwickelten Algorithmen ist die Echtzeiterkennung von neuen statischen und sich bewegenden Objekten beliebigen Aussehens in uneingeschränkten Überwachungsumgebungen, die mit statischen Kameras ausgestattet sind. Die Grundlage der Verfahren sind die Ergebnisse der Hintergrundsubtraktion. Für diese Aufgabe werden Gaussian Mixture Models (GMMs) verwendet. Eine gründliche Überprüfung der State-of-the-Art Formulierungen für den Einsatz von GMM bezüglich der Aufgabe der Hintergrundsubtraktion zeigt Möglichkeiten zur Weiterentwicklung, die zu einem neuen auf einer Varianzsteuerung basierten GMM-Ansatz führen. Der vorgeschlagene Algorithmus ermöglicht eine einfachere Parametrisierung der Modelle für unterschiedliche Umgebungen und konvergiert zu genaueren Modellen der beobachteten Szene.

Die Detektion von sich bewegenden Objekten wird durch Verwendung der Ergebnisse der Hintergrundsubtraktion erzielt. Für die Erkennung von neuen statischen Objekten werden zwei Hintergrundmodelle, die sich mit unterschiedlichen Geschwindigkeiten an die Szene anpassen, verwendet. Dies ermöglicht eine mehrklassige Pixelklassifikation, welche den zeitlichen Bezug der durch Hintergrundsubtraktion erzielte Ergebnisse berücksichtigt.

In einem ersten Ansatz werden die Ergebnisse der Subtraktion beider Hintergrundmodelle als Eingabe einer Finite State Machine (FSM) verwendet, die eine Pixelklassifizierung auf Grundlage der Pixelhistorie erzeugt. Dies ermöglicht die Erkennung von neuen statischen Objekten über lange Zeiträume und eine korrekte Klassifizierung der aufgedeckten Hintergrundbereiche bei der Entfernung dieser Objekte.

In einem weiterentwickelten Ansatz werden die Ergebnisse der mehrklassigen Pixelklassifikation in Regionen gruppiert und weiter analysiert, um zwischen neuen und entfernten statischen Objekten zu unterscheiden. Dies ermöglicht die Erkennung von statischen Objekten ohne Vorkenntniss der beobachteten Szene. Weiterhin wird gezeigt, dass die Ergebnisse der Regionanalyse verwendet werden können, um die Qualität der Hintergrundmodelle und somit auch der Detektionsergebnisse erheblich zu verbessern.

Die Ergebnisse der entwickelten Algorithmen werden anhand eines neuartigen Verfahrens zur Zusammenfassung von Videoinhalten demonstriert. Die neu entwickelte Methode fusioniert die Ergebnisse mehrerer Videoanalysealgorithmen, um Zusammenfassungen zu erstellen, die sich besser an dem Inhalt der analysierten Videosequenzen ausrichten.

Resumen

El objetivo de los algoritmos desarrollados en esta tesis es la detección en tiempo real de objetos en movimiento y de nuevos objetos estáticos de apariencia visual arbitraria en entornos no controlados de video vigilancia monitoreados con cámaras estáticas. Para ello se han utilizado como base los resultados proporcionados por sustracción de fondo. Para la sustracción de fondo se han utilizado modelos de mezcla de Gaussianas (GMMs por las siglas en inglés). Un examen exhaustivo del estado del arte con respecto al uso de GMMs en la tarea de sustracción de fondo revela algunas deficiencias de las actuales formulaciones, las cuales han sido abordadas en un nuevo GMM que incorpora un esquema para el control de la varianza. El sistema propuesto permite una parametrización más sencilla del modelo en diferentes entornos y converge a modelos más exactos de la escena observada.

La detección de objetos en movimiento se consigue mediante el uso de los resultados de la sustracción de fondo. Para la detección de nuevos objetos estáticos, se utilizan dos modelos de fondo que se adaptan a la escena a un ritmo diferente. De este modo, se puede realizar una clasificación de los pixels atendiendo a multiples clases que se corresponden con la relación temporal de los cambios detectados.

En un primer sistema, los resultados proporcionados por la sustracción de los dos modelos de fondo se utilizan como entrada de una máquina de estados finitos (FSM por las siglas en inglés), que sirve para razonar sobre la clasificación de cada píxel en función de su historia. Esto permite la detección de nuevos objetos estáticos durante largos períodos de tiempo y la correcta clasificación de las regiones de fondo descubiertas cuando estos son retirados.

En un sistema más desarrollado, los resultados proporcionados por la clasificación de pixel multi-clase se analizan a nivel de región con el fin de distinguir entre objetos estáticos nuevos y retirados. Esto permite la detección de nuevos objetos estáticos sin conocimiento previo de la escena observada. Además, se demuestra que los resultados proporcionados por el análisis de región se pueden utilizar para mejorar la calidad de los modelos de fondo, por lo tanto, mejorando considerablemente los resultados de la detección.

Los resultados proporcionados por los algoritmos desarrollados se presentan en un nuevo sistema de creación automática de sinopsis de secuencias de video que combina múltiples colas de análisis con el fin de proporcionar resúmenes que reflejan mejor el contenido de las secuencias de vídeo analizadas.

Contents

List of Figures

List of Tables

List of Acronyms

BN	Bayesian Network
CCTV	Closed Circuit Television
CHMM	Coupled Hidden Markov Models
EFSM	Extended Finite State Machine
EM	Expectation Maximization
FSM	Finite State Machine
GEM	Generalized Expectation Maximization
GMM	Gaussian Mixture Model
HCI	Human Computer Interaction
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
IPCA	Incremental Principal Component Analysis
KDE	Kernel Density Estimation
LBP	Local Binary Pattern
PCA	Principal Component Analysis
PTZ	Pan-Tilt-Zoom
STLBP	Spatio-Temporal Local Binary Pattern
SMEM	Split and Merge Expectation Maximization
SVM	Support Vector Machine
VLBP	Volume Local Binary Pattern

1

Introduction

Computer vision, image processing and pattern recognition are vast fields of research concerning the automatic analysis of images and image sequences, with a broad spectrum of applications such as remote sensing, medical diagnosis, human-computer interaction or video compression, to mention only a few of them. Profiting from the advances in those fields, automated video-based surveillance has arisen as an own research topic which has gained a lot of attention in the recent years, due to the increasing threats to the security in public places such as railway stations or airports. The aim is to assist human operators in monitoring Closed Circuit Television (CCTV) camera networks, by alerting them on deviation from the normal behavior observed in the area under surveillance. This provides the main benefit, that an operator may monitor a larger amount of cameras by concentrating his attention to the critical points in space and time, while the system assumes the tedious task of monitoring areas where non-interesting events are happening. Furthermore, the knowledge acquired by means of automatic video analysis techniques can be used in order to assist video operators and legal authorities in the retrieval of evidence proofs from recorded video data, to administer large area video networks in tasks such as panning and zooming in and out of Pan-Tilt-Zoom (PTZ) cameras, and even for less technical issues as helping to protect the privacy of individuals in public places.

Video surveillance systems have experienced a rapid development in the last decades, especially after the attacks on the 11th of September 2001 in New York, 11th of March 2004 in Madrid and 7th and 21st of July 2005 in London, leading them to become a part of our daily life. But the use of video surveillance systems is not restricted to safety and security applications. Nowadays, video surveillance systems are also being deployed at department stores in order to provide advertising assessment and quality of service, on highways for traffic monitoring

purposes, and even on houses for elderly people to assist them in a non-invasive manner. This success has been supported by the decaying prices in the sensor industry, which is able to provide higher quality cameras of ever smaller sizes at low prices. Moreover, the introduction of wireless networks has connoted a drastic reduction in the deployment costs. With the transition to IP camera networks, large camera networks can be both local and remotely controlled.

The rapid growth of video surveillance systems results in an increasing number of video feeds which should be monitored and stored in a control room. This results in a continuously growing workload for CCTV operators, who are overwhelmed by the huge sets of cameras. To alleviate this problem, automatic video analysis techniques aim at understanding actions and human behaviors in video sequences in order to alert CCTV operators upon the occurrence of threatening situations. This scenario corresponds to the proactive side of crime prevention. Besides that, video surveillance systems can also be used for crime investigation and offenders prosecution. Video indexing and summarization can be used in order to effectively accomplish this last task. Furthermore, automated video surveillance systems have given raise to the paradigm of bringing intelligence to the edge of the network. This allows for the design of distributed surveillance networks, which require a lower bandwidth for the transmission of the captured information.

Nevertheless, as video surveillance systems have become ubiquitous, some aspects of the deployed systems have been questioned. One of the aspects is the effectiveness regarding crime prevention [Sasse, 2010]. Another is the need of protecting the privacy and security of personal information, which has gained increasing attention in the recent years. The Telegraph claimed that an individual will appear on average on 300 CCTV cameras during a day [Gray, 2008].

All of these aspects together have attracted the attention of both the academy and the industry, and is expected to continue growing in the next years. A recent report of Homeland Security Research Corporation [HSRC, 2013] estimates the revenue of the global Intelligent Video Surveillance (IVS) & Video Analytics (VA) industry as $13.5 billion in 2012, and predicts a rapid growth until 2020, where it is expected to reach $39 billion.

The technical conception and deployment of automated video-based surveillance systems involve a number of key issues to be addressed. The lowest level of the system design concerns hardware issues, including video acquisition (cameras), storage devices and networks. At this level, decisions are taken like network topology and communication protocols. Upon this level, the information gathered by the cameras is analyzed by means of image and video processing techniques, so as to extract useful information out of the video sequences. This is the level providing the semantic capabilities of the system. Finally, at the top level, the extracted information is presented to the user and eventually stored in a database for further usage. At this level, considerations on the ergonomics of the system as a whole and human-

computer interaction should be taken into account. Obviously, decisions made at the different levels of design might affect the decisions to be made at the other levels; even more in the case of bringing intelligence to the network. The main focus of this thesis is set on the video processing and understanding chain.

1.1 Automated Video-Based Surveillance Systems

Automated video-based surveillance systems, in this thesis referred to as surveillance systems for brevity (otherwise explicitly indicated), rely on the automatic detection of events of interest by means of several analysis techniques mainly stemming from the fields of computer vision, image processing and pattern recognition. Detecting events of interest is an application dependent task and can be approached in very different manners. Nevertheless, there is a common number of steps that a general surveillance system usually goes through, namely, object detection, object association, commonly referred to as tracking, and scene understanding, often accomplished by the less ambitious task of event detection. In order to successfully accomplish these tasks, the cameras have to be calibrated with respect to an extrinsic Cartesian reference space, therefore allowing for a measurement of the size and position of the detected objects. These main building blocks of an automated video-based surveillance system are depicted in Figure 1.1 and briefly introduced in the following subsections.

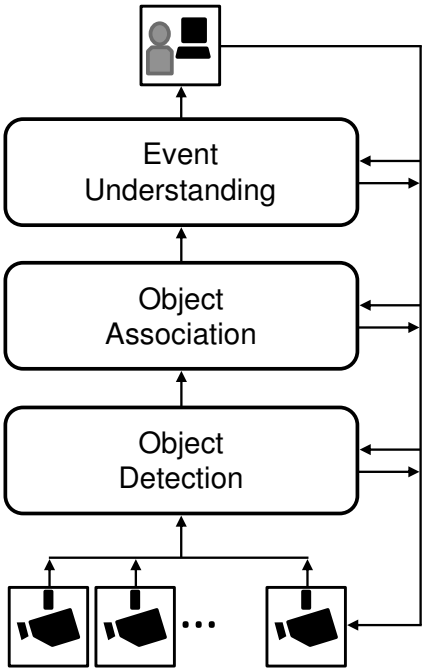


Figure 1.1: *General video-based surveillance system.*

1.1.1 Camera Calibration

Camera calibration is the process of estimating the parameters required for projecting the three dimensional world coordinates into the two dimensional image coordinates. To that aim it is necessary to determine internal camera geometric and optical characteristics (the intrinsic parameters) and the position and orientation of the camera in relation to the observed scene (extrinsic parameters). This set of parameters is needed for obtaining measurements concerning the position and size of the detected objects in the observed scene, which is an essential information in order to classify objects and associate them between consecutive video frames and between several camera views in multi-camera setups. Furthermore, the position of the detected objects can also be a decision factor in some surveillance applications as, e.g., perimeter protection. In static camera setups, camera calibration can be performed once at the system deployment. A seminal method for the calibration of cameras was presented in [Tsai, 1987], where a two-stage technique aiming at efficiently and accurately compute the set of transformation parameters is introduced. However, this method was envisioned in order to provide high accuracy camera calibration and requires an elaborated setup, including a calibration pattern and accurate 3D coordinates of the calibration points. A more flexible calibration technique, which only requires a calibration pattern printed in a planar surface and a sequence of (at least two) frames where the pattern is depicted from different orientations (either the camera or the pattern can be moved, whereas the motion does not need to be known), was presented in [Zhang, 1998]. In the case of needing a camera calibration without the possibility of using a known calibration object, alternative approaches based on the computation of vanishing points for orthogonal directions can be employed [Caprile and Torre, 1990]. Some exemplary approaches proposed for the surveillance domain can be found in [Krahnstoever and Mendonça, 2006], where the position of the head and feet of the detected moving people are used for the computation of these vanishing points, and in [Liebowitz et al., 1999], where the parallelism and orthogonality in architectural scenes are used for the computation of camera calibration from a single image (the scaling factor can be estimated by using any object in the scene whose dimension is known). In systems involving moving cameras, on-line self-calibration techniques [Maybank and Faugeras, 1992], which exploit point correspondences along the camera path for the computation of the camera intrinsic parameters, have to be employed. A survey on camera self-calibration techniques can be found in [Hemayed, 2003]. It should be pointed out, that there is not a calibration technique which better fits for all application scenarios. Nevertheless, generally speaking it can be said that calibration techniques using calibration patterns provide a higher accuracy than self-calibration techniques. Accordingly, a higher accuracy can be achieved by using 3D instead of 2D calibration patterns [Zhang, 2004].

1.1.2 Object Detection and Classification

Generic object recognition, also known as category-level object recognition, is considered to be one of the most challenging visual tasks in computer vision [Szeliski, 2010]. Given any instance of a particular general class as, e.g., 'person', 'car' or 'bicycle', the task is to correctly localize and classify it by means of visual features. An exhaustive search over all object models and image locations can be too time-consuming for many computer vision applications. In order to reduce the complexity of the problem, surveillance systems usually divide the problem into two steps: first, the objects of interest are detected and, second, the detected objects are classified. Objects of interest are usually defined as those objects introducing some kind of change in the observed scene and are generally associated to moving objects.

In this context, object detection can be approached by means of three different techniques: temporal differencing, background subtraction, and optical flow. These three techniques provide a low-level pixel classification. In order to build objects, pixels are then clustered attending to this classification and their spatial configuration. Temporal differencing is based on computing the difference of consecutive video frames at every pixel position and classifying as changed pixels those which absolute difference exceeds a given threshold. Temporal differencing is highly adaptive to dynamic environments and low demanding in computational terms, but it fails to extract the whole set of pixels corresponding to the objects in motion. Early works based on temporal differencing can be found in [Jain and Nagel, 1979] and references therein. Background subtraction is the most commonly used approach in setups with static cameras. It consist in using a model of the scene background in order to detect foreground objects by differencing incoming frames with the model. Background subtraction is mostly fast and has low computational demands. However, it can be sensitive to sudden illumination changes and small camera motions as, e.g., vibrations. A good introduction to background subtraction, including the main issues that a background subtraction approach has to deal with, can be found in [Toyama et al., 1999] and later on in this thesis (see Chapter 2). An overview of state-of-the-art approaches and their respective performance can be consulted online in the CDnet dataset website[1] [Goyette et al., 2012]. Optical flow is an estimation used to determine corresponding points between two images. Optical flow based methods can be used to detect independently moving objects even in the presence of camera motion. Nevertheless, even in their most efficient implementations, they are highly demanding in computational terms. Furthermore, depending on the smoothness constraint, the corresponding points in the considered frames might not be allowed to be more than a few pixels away, therefore, being constrained the speed of movement of objects and camera. A good introduction to the topic of optical flow computation can be found in [Barron et al., 1994; Beauchemin and Barron, 1995]. An overview of state-of-the-art approaches and their respective performance can be

[1] www.changedetection.net

consulted on-line in the Middlebury dataset website[2] [Baker et al., 2011], and in the website of the more recently created KITTI dataset in[3] [Geiger et al., 2012].

Figure 1.2: *Temporal differencing. From left to right: first image of a pair containing one moving person, second image of the same image pair, and difference mask.*

Figure 1.3: *Background subtraction example for two frames of the sequence 'office' from the CDnet dataset. From left to right: frame number 001835, background of the scene, and ground-truth foreground mask (source, www.changedetection.net).*

Figures 1.2 to 1.4 provide some exemplary results for the above presented moving object detection approaches. Figure 1.2 depicts two consecutive frames of a scene in a laboratory where one person is moving and the difference mask, which has been obtained by thresholding the absolute value of the difference between the frames. Figure 1.3 depicts a scene in an office where a person enters and consults a book. Frame number 500, which depicts the empty office, has been taken as an exemplary depiction of the background. The foreground mask has been taken from the manually generated ground-truth provided with the CDnet dataset. Figure 1.4 shows two consecutive frames of the synthetically generated 'Grove3' sequence from the Middlebury dataset and the color coded ground-truth motion map. These pictures have been obtained from the dataset's website [Baker et al., 2011].

[2] http://vision.middlebury.edu/flow
[3] www.cvlibs.net/datasets/kitti

Figure 1.4: *Optical flow example for two frames of the sequence 'Grove3' from the Middlebury dataset. From left to right: frame number 9, frame number 10, ground-truth flow (source, http://vision.middlebury.edu/flow/).*

Object classification is a very hard problem to solve due to the high amount of objects which appear in different poses and occluding each others in the natural world. Furthermore, the large intra-class variability associated to the often small inter-class differences, makes this problem even more difficult to handle. However, if the searched object is known, the problem can be broken down to a single class recognition problem. In that case, special purpose detectors can be trained by means of machine-learning techniques such as neural networks [Rowley et al., 1998], support vector machines [Papageorgiou et al., 1998] or adaptive boosting [Viola et al., 2003]. These learning methods compute a hyper-surface that is used to separate the trained object classes from each other in a high dimensional features space. These detectors can then be used to exhaustively search a given image, or to classify previously detected objects as belonging to the trained class or not. Two of the most widely used detectors in the video surveillance domain are the face detector in [Viola and Jones, 2004] and the Histogram of Oriented Gradients (HOG) for the detection of humans, first introduced in [Dalal and Triggs, 2005]. In [Viola and Jones, 2004], the concept of boosting, which consists of a series of increasingly discriminating simple classifiers, is introduced to the computer vision community. In [Dalal and Triggs, 2005], a set of overlapping histograms of oriented gradients descriptors fed into a Support Vector Machine (SVM) are used to robustly detect humans.

If training a special purpose detector is not feasible because of the application scenario constraints (real-time, computational availability, etc.) or because the search class includes too many different possible appearances, more generic approaches can be taken into account for the object classification task. For instance, in [Lipton et al., 1998], shape information of the detected objects and temporal consistency are used to classify all moving objects into either humans, vehicles or clutter. The shape features used are the size of the object and the so called *dispersedness*, which is defined as the relation between the perimeter of the detected objects and the object's area size (the *dispersedness* of humans is usually higher than that of cars since humans have more complex shapes). Shape features (*dispersedness*

and size) are used to distinguish between persons or cars on a frame basis. The temporal consistency is used to classify objects upon a given statistical certainty on their class, which is evaluated by computing a classification histogram for each motion region. Other simple frequently used shape features include, e.g., the aspect ratio of the objects bounding boxes, the eccentricity (which is computed as the ratio between the length of the major axis and the length of the minor axis) and the major axis orientation. A thorough review of shape representation techniques is provided in [Zhang and Lu, 2004]. Shape features are in general sensitive to the presence of occlusions. Therefore, some other features based on color, texture and even motion are also used for the object classification task. For example, in [Cutler and Davis, 1998] a technique is described to detect and analyze periodic motion; moving objects can then be discriminated based on this analysis (humans exhibit a periodic motion by walking, while cars not).

If training special purpose detectors is possible but the assumption that the objects of interest are in motion cannot be met, an exhaustive search of the objects of interest can be done. The main drawback of the exhaustive search is the high computational effort and, consequently, time required. To alleviate this problem, some techniques have been developed to accelerate either the localization task by using local features [Chum and Zisserman, 2007; Leibe et al., 2008], or the classification by using tree-based data structures in the case of multi-class problems [Bosch et al., 2007], or both steps concurrently [Yeh et al., 2009]. A recent survey covering most of the topics involved in the tasks of object detection and classification can be found in [Zhang et al., 2013].

1.1.3 Object Tracking

Object tracking is the task of establishing correspondences between the detected objects across the frames of a video sequence. In order to accomplish this task, a model for the objects and the motion they exhibit is used. Typical object models are points, primitive geometric shapes, as, e.g., ellipses and rectangles, silhouettes, articulated shape models and skeletons. Depending on the selected object model used, the motion model can be delimited. For example, if an object is represented by a point, then, only a translational model can be used, whereas in the case of more elaborated object models as, e.g., silhouettes, parametric and non-parametric motion models can be used. Depending on the application domain, assumptions are made in order to constrain the tracking problem. In the surveillance domain, point-based tracking models are a popular choice to solve the tracking problem. Thereby, Kalman [Broida and Chellappa, 1986] and Particle Filters [Tanizaki, 1987] are commonly state estimation methods used for computing the cost of a given object association. An excellent introduction into the tracking topic and important related issues including the use of appropriate image features, selection of motion models, and detection of objects, can be found in [Yilmaz et al., 2006].

1.1.4 Scene Understanding

The ultimate aim of the video surveillance analysis chain is the interpretation of the observed scene. Based on the knowledge acquired at this step, alarms can be triggered in order to assist human operators in CCTV control rooms, indexes and summaries can be generated in order to provide non-linear access to video contents in forensic investigations, the field of view of the cameras in a network can be automatically changed in order to better follow the situations of interest, etc. This output is usually done in form of events. The semantic interpretation of a video sequence can be done by means of either learned or imposed knowledge. Some of the video surveillance applications of interest, such as the detection of persons at certain locations in perimeter protection applications, require of imposed knowledge, while other applications, such as the detection of abnormal traffic flows, can be learned on-line. Imposed knowledge can be provided in form of areas of interest, which can be used to raise events upon the appearance of certain type of objects inside them, or more elaborated representations such as a Finite State Machine (FSM), which can be used to define simple behavior templates [Cupillard et al., 2002], or a Bayesian Network (BN) [Park and Aggarwal, 2003]. Learned knowledge is usually acquired by training a system with reference sequences representing typical behaviors. To that aim, dynamic graphical models, such as Hidden Markov Models (HMM) and Coupled Hidden Markov Models (CHMM) [Oliver et al., 2000], which allow for a more sophisticated analysis of data with spatio-temporal variability, have been extensively used. Self-organizing neural networks [Johnson and Hogg, 1996] can be used in unsupervised learning scenarios, where the object motions are unrestricted.

1.1.5 Overall System Design Considerations

Up to now, the building blocks of an automated video surveillance scenario have been described in a bottom-up approach, where each of the components takes the results of its previous analysis module as input and provides its output for further processing. This approach has the main advantage of providing a high isolation of problems. Nevertheless, low level analysis building blocks do not take advantage of the knowledge acquired at higher analysis levels. Therefore, more involved systems usually include some kind of feedback in order to profit from high-level knowledge, what corresponds to a top-down design. Moreover, Human Computer Interaction (HCI) can also be considered in order to better fit the results of an automated surveillance system to a given application scenario.

Further considerations which need to be taken into account by the design of an automated surveillance system concern the application domain and hardware configuration. Depending on that, different requirements, constraints and capabilities have to be attended; for instance, a forensic application must not provide real-time ability while a pro-active surveillance system must unavoidably attend it.

In this thesis, the considered building blocks are first analyzed separately, providing a deep analysis of the considered problems. Later, feedback is introduced in a top-down fashion, showing that carefully coupling the analysis done at different levels of abstraction can be used in order to considerable improve the results both at the overall system and at the layer level. Thereby, a deep understanding of the considered problems in isolation is of crucial importance in order to fully exploit the possibilities offered by the introduction of feedback. By the design of the developed algorithms, a special attention has been paid to the lightness in terms of hardware and computational demands, since real-time is one of the requisites that the application domain in mind imposed. Furthermore, a very important aspect which has been considered is the ability to gracefully incorporate the knowledge provided by the users of the system. The incorporation of user interaction is a topic which had been largely ignored in the development of surveillance systems. Nevertheless, it has gained attention in the recent years, as it is easy to observe by the appearance of research directions including the *human-in-the-loop* factor and by the proliferation of workshops dedicated to the topic as, e.g., 'Person-Oriented Vision' and 'User-Centred Computer Vision'.

1.2 Thesis Overview

1.2.1 Thesis Objectives and Contributions

The main focus of this thesis is the real-time detection of objects in unrestricted environments monitored with static video cameras. The objects of interest are moving as well as new static objects. The video analytics system is not provided with any previous knowledge neither of the observed scene nor of the visual appearance of the objects to be detected. The main application in mind of the developed algorithms is the detection of abandoned objects in public spaces, which has gained an important attention in the security domain, since abandoned objects might be often considered as a threat to the public security. The final system has to provide on-line alerts to human operators. Furthermore, the detected moving objects should be provided to higher-level analysis tools in order to recognize further actions and behaviors of interest typical of surveillance systems for public spaces.

Given the problem statement, background subtraction is the most robust low-level analysis cue. Other analysis cues relying on motion information are not adequate for the considered task since abandoned objects remain static. Furthermore, since the appearance of the objects is unknown, it is not possible to detect them by means of models. Therefore, the low-level analysis cue used for object detection within the work presented here is background subtraction, which provides information of the image positions where a change with respect to the background model has been observed. The combination of the results provided by two on-line generated statistical background models, which attend to different temporal configurations, is exploited in order to perform a multi-class pixel classification, which distinguishes between

several kind of changes as e.g. moving or static foreground. A first system is presented which, based on this classification, raises alarms upon the detection of new static foreground regions. In a further developed approach, static foreground detections are classified by means of region analysis as either removed or new static objects. This classification is then used by means of feedback in order to manage the update process of the background models. As shown in this thesis, the resulting system is not only able to efficiently detect new and removed static objects, but, also, to improve the performance of background subtraction.

The main contributions of the work presented are as follows:

- An enhanced Gaussian Mixture Model (GMM) for video surveillance applications, which incorporates recent proposals for the improvement of the system performance and system convergence, and a novel heuristic for:
  - better initializing the parameters of new created modes, and
  - avoiding the emergence of over-dominating modes.

  The resulting overall system improves the performance of state-of-the-art background subtraction approaches, and in particular of GMM-based approaches, in terms of segmentation accuracy and is more appropriate for systems which need to incorporate feedback information into the background model of the scene.

- A finite state machine (FSM) for multi-class pixel classification, which classifies pixels attending to different stages as background, moving, stationary or uncovered background, and further stages. The FSM is used for hypothesizing on pixel classification based on the results obtained from the subtraction of two background models and on the history of the pixel, which is implicitly recorded by the FSM.

- A system for the detection of new static objects in crowded environments based on a complementary background model, a FSM and region analysis. The proposed system does not need previous knowledge of the scene background, is robust to the main problems affecting the detection of new static objects like occlusions and ghosts, and improves the results obtained by background subtraction.

- A novel indexing and summarization approach which combines the results provided by several video analysis cues. The proposed system computes a speed associated to each of the connected analysis cues depending on their provided results. Based on these associations, a final adaptive speed based on the extracted video content is computed for the summary video.

These contributions have resulted in seven scientific publications, one international and two US patent applications. A detailed publication list is provided in Section 1.3.

1.2.2 Thesis Structure

The next chapters of this thesis are organized as follows:

Chapter 2 provides a comprehensive review of background subtraction, which is one of the main pillars of the work presented here. After briefly introducing the main goals and challenges of a general background subtraction system, a series of representative methods is presented. These methods have been selected so as to provide an overview of the whole range of techniques which have been used in the surveillance domain for the task of background subtraction and the goals addressed by each of them. Therefore, the presented approaches refer to the seminal papers where the presented techniques were first introduced and provide pointers to further developments. At the end of the chapter, current trends in the background subtraction literature are presented and the conclusions of the presented analysis are drawn.

In Chapter 3 the proposed improved Gaussian Mixture Model (GMM) is presented. The chapter first presents the Expectation Maximization (EM) algorithm, which is a fundamental machine-learning tool used for the estimation of the parameters of the most common GMM approaches for the task of background subtraction. Out of the properties of the EM algorithm and the several GMM approaches of the surveillance literature, the main deficiencies of the state-of-the-art GMM approaches are identified and addressed by the proposed method. Experimental results proof the achieved improvement.

Chapter 4 tackles the problem of the detection of new static objects. After providing a brief review of state-of-the-art approaches, a dual background based system is proposed. The results provided by background subtraction are used as input of a novel FSM, which is used to reason on pixel classification. The main advantage of this system with respect to a plain dual background based system is that the proposed system is able to detect new static objects over long periods of time without sacrificing the adaptability of the background models and to correctly classify the uncovered background regions upon their removal. The proposed system is evaluated by using several datasets.

Based on the observations made by the design and evaluation of the system in Chapter 4, in Chapter 5 an improved system is proposed which incorporates a region classification step after the multi-class pixel classification and a feedback loop used to build complementary background models based on the information gathered at the region level. The proposed system presents the main advantage of being able to detect new static objects without previous knowledge of the scene background; furthermore, the feedback loop allows for a significant improvement at the pixel classification level. Therefore, experimental results are presented for both tasks, new static objects detection and foreground segmentation.

The main two tasks accomplished by the systems proposed in this thesis, background subtraction and static object detection, are brought together in an exemplary automated video

surveillance application, video indexing and summarization, in Chapter 6. The chapter provides a thorough review of summarization techniques and state-of-the-art approaches. The pros and cons of the presented approaches motivate the proposal of a novel indexing and summarization system, which is then experimentally evaluated. Therefore, beyond providing a mere exemplary application, this chapter constitutes on its own a contribution in the field of automated video-based surveillance.

Chapter 7 concludes this thesis, highlight the main findings done during the elaboration of the presented work and summarizes the main achievements.

An overview of the datasets and performance metrics used to evaluate the proposed algorithms is provided in the Appendix.

1.3 List of Publications

Several methods and results presented in this thesis have been published in the scientific publications listed below, which are sorted in chronological order and properly cited in the corresponding chapters. Furthermore, three patent applications have been submitted to protect the intellectual property of the systems described in Chapter 4 and Chapter 5.

Conference Publications:

Heras Evangelio, R., Senst, T., and Sikora, T. (2011a). Detection of static objects for the task of video surveillance. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, pages 534–540, Kona, HI, USA.

Heras Evangelio, R., Pätzold, M., and Sikora, T. (2011b). A system for automatic and interactive detection of static objects. In *Proceedings of the IEEE Workshop on Person-Oriented Vision (POV)*, pages 27–32, Kona, HI, USA.

Heras Evangelio, R. and Sikora, T. (2011c). Complementary background models for the detection of static and moving objects in crowded environments. In *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 71–76, Klagenfurt, Austria.

Heras Evangelio, R., Pätzold, M., and Sikora, T. (2012). Splitting gaussians in mixture models. In *Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 300–305, Beijing, China.

Heras Evangelio, R., Senst, T., Keller, I., and Sikora, T. (2013a). Video indexing and summarization as a tool for privacy protection. In *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Santorini, Greece.

Heras Evangelio, R., Keller, I., and Sikora, T. (2013b). Multiple cue indexing and summarization of surveillance video. In *Proceedings of the 10th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Kraków, Poland.

Journal Publications:

Heras Evangelio, R. and Sikora, T. (2011d). Static object detection based on a dual background model and a finite-state machine. *EURASIP Journal on Image and Video Processing*, 2011:858502.

Heras Evangelio, R. Pätzold, M., Keller, I, and Sikora, T. (2014). Adaptively splitted GMM with feedback improvement for the task of background subtraction. *Accepted for publication in IEEE Transactions on Information Forensics & Security*.

Patent Applications:

Heras Evangelio, R., Sikora, T., and Keller, I. (2013c). Method and device for video surveillance. US Patent Application, Number US 2013/0027549 A1

Heras Evangelio, R., Sikora, T., and Keller, I. (2013d). Method and device for video surveillance. US Patent Application, Number US 2013/0027550 A1

Heras Evangelio, R., Sikora, T., and Keller, I. (2013e). Method and device for video surveillance. International Patent Application, Number WO 2013/017184 A1

During the development of the work presented here I had the privilege of working together with Michael Päzold and Tobias Senst, who involved me in their interesting research topics and brought their useful insights into mines. Out of these collaborations, further interesting results, which are not part of this thesis, were obtained and published in the proceedings of international conferences. A list of the resulting publications in chronological order is provided below.

Senst, T., Heras Evangelio, R., Eiselein, V., Pätzold, M., and Sikora, T. (2010). TOWARDS DETECTING PEOPLE CARRYING OBJECTS: A Periodicity Dependency Pattern Approach. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Angers, France.

Pätzold, M., Heras Evangelio, R., and Sikora, T. (2010a). Counting people in crowded environments by fusion of shape and motion information. In *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), PETS Workshop*, pages 157–164, Boston, USA.

Pätzold, M., Heras Evangelio, R., and Sikora, T. (2010b). Counting people in crowded environments: An overview. In *Hands-on Image Processing (HOIP). Security, Surveillance and Identification in Everyday Life*, TECNALIA. Bizkaia, Spain. invited paper.

Senst, T., Eiselein, V., Heras Evangelio, R., and Sikora, T. (2011a). Robust modified l2 local optical flow estimation and feature tracking. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC)*, pages 685–690, Kona, USA.

Senst, T., Heras Evangelio, R., and Sikora, T. (2011b). Detecting people carrying objects based on an optical flow motion model. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, pages 301–306, Kona, USA.

Senst, T., Pätzold, M., Heras Evangelio, R., Eiselein, V., Keller, I., and Sikora, T. (2011c). On building decentralized wide-area surveillance networks based on onvif. In *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 420–423, Klagenfurt, Austria.

Pätzold, M., Heras Evangelio, R., and Sikora, T. (2012). Boosting multi-hypothesis tracking by means of instance-specific models. In *Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 416–421, Beijing, China.

Senst, T., Heras Evangelio, R., Keller, I., and Sikora, T. (2012). Clustering motion for real-time optical flow based tracking. In *Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 410–415, Beijing, China.

2

Background Subtraction State of the Art

2.1 Introduction

The detection of change is a low-level vision task used as a first step in many computer vision applications such as video surveillance, low-rate video coding, human-computer interaction, augmented reality or medical diagnosis to mention only a few of them. Given an image sequence, the goal is to identify for each frame the set of pixels that are significantly different from the previous frames. Depending on the application, the requirements and constraints of the detection algorithm are different. Likewise, the definition of what is significantly different, may depend on the application domain.

In the video surveillance domain, change detection has been frequently used in order to segment foreground objects from the background. Foreground objects are the objects of interest in an automated surveillance system. The segmented foreground objects are then associated between frames in order to perform a scene analysis and detect events of interest. As background is understood what is normally observed in the scene. Therefore, it is assumed that the background can be well described by means of a statistical model, the background model. Nevertheless, there are some background characteristics as moving foliage or sudden illumination changes, which might make difficult the task of background modelling and maintenance. A comprehensive study of the main challenges and some principles that might be used to tackle them can be found in [Toyama et al., 1999]. The segmentation of foreground objects by means of detecting the changes with reference to a background model is commonly known as background subtraction. Figure 2.1 depicts a basic schema of a general background

subtraction system. The main challenges a background subtraction algorithm has to deal with are [Toyama et al., 1999; Brutzer et al., 2011]:

- Gradual illumination changes, which are mainly experienced in outdoor environments along the different times of the day and affect the appearance of the objects in the observed scene.
- Sudden illumination changes, which are mainly experienced in indoor environments by the switching on and off of artificial light sources, and in outdoor environments by unstable weather conditions when clouds suddenly hide the sun.
- Shadows, which are mainly casted by moving objects and complicate the accurate segmentation of objects (static objects belonging to the background also cast shadows; nevertheless, these are not that problematic for the background subtraction process since they are always casted at the same position -or at slow moving positions in outdoor scenarios depending of the sun position- and can be more easily accommodated in the background model).
- Dynamic background, which are those parts of the background exhibiting different appearances because of containing some kind of moving objects as waving trees, rippling water, escalators and so on, which are not of further interest for a scene interpretation.
- Camouflage, produced by objects whose appearance is difficult to differentiate from the appearance of the background.
- Bootstrapping, which is required because of the general unfeasibility of training a background model with a completely empty scene.

Actually, in [Toyama et al., 1999] the authors also pointed out some challenges which they claimed that a background maintenance system should be able to handle:

- Moved objects, which refers to the detections corresponding to background objects that have been moved.
- Sleeping person, which refers to foreground objects appearing in the scene and remaining motionless after a while.
- Walking person , which refers to objects that have been learned as part of the background and at some point in time start moving and leave the scene.

Nevertheless, these three challenges have not been considered in this thesis as inherent to the background subtraction problem, since these problems should be considered in accordance to the application in mind. In fact, the point in time from which, e.g., a person falling asleep is

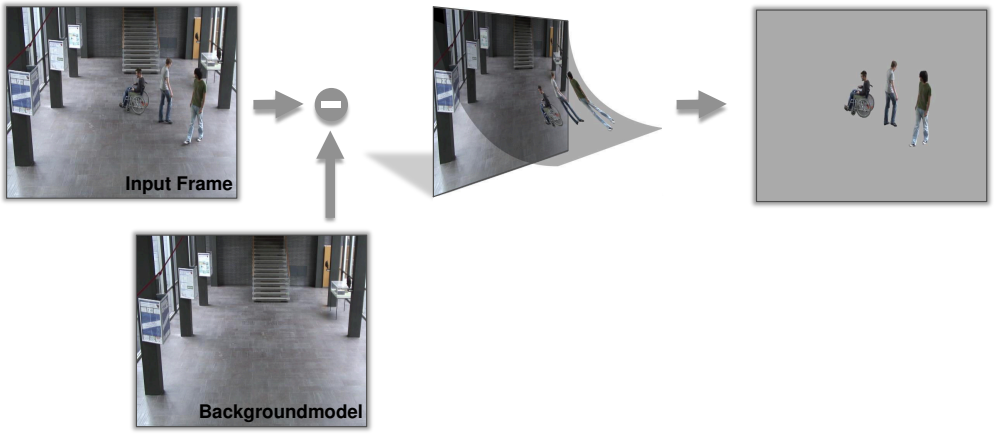


Figure 2.1: *General background subtraction system.*

not interesting anymore should be defined by a given application and, therefore, should not be considered as a general background maintenance problem. It is, moreover, remarkable that these three problems can be also considered as three singularities a good bootstrapping strategy should handle. Finally, the foreground aperture problem, also mentioned in [Toyama et al., 1999], which consists in the unfeasibility of detecting interior object pixels because of color homogeneity, has neither been considered in this work as a general change detection problem, as mostly concerns frame differencing based approaches.

2.1.1 Taxonomy

The number of background subtraction approaches which have been proposed in the literature is large, and so are the different taxonomies which can be used to classify them. Attending to the spatial level considered, background subtraction approaches can be divided into three classes:

- Pixel-level algorithms only use features gathered at each single pixel position. These methods are very fast, but they do not use any kind of inter-pixel relationships. There have been many proposals in the literature for these kind of methods; among them, Running Gaussian Averages [Wren et al., 1997], Median Filtering [McFarlane and Schofield, 1995] and Gaussian Mixture Models [Stauffer and Grimson, 1999], have been of special relevance and have originated a vast number of derived systems.

- Block-level based approaches divide an image into blocks and compute block related features to describe the background. Block-level approaches are usually more robust against noise than pixel-level approaches, on the other hand they provide courser detec-

tions of the foreground objects and are computationally expensive. Some example of these kind of approaches are the Normalized Vector Distance based approach in [Matsuyama et al., 2000] and the Local Binary Pattern texture based approach in [Heikkilä et al., 2004].

- Region-level based algorithms divide an image into a set of regions which are then classified as background or foreground. There is a very limited number of purely region-level based algorithms since finding meaningful regions in an image by means of spatial consistency criteria can be computationally expensive. Therefore, region-level based approaches are usually combined with another kind of approach which is used to determine the regions followed by the region classification itself. Nevertheless, there are some examples of purely region-level based algorithms as the one presented in [Huang et al., 2004], which is based on the Partial Directed Hausdorff distance, and the more recently proposed in [Yu et al., 2007], where the authors propose to model foreground and background objects by means of Spatial-Color Gaussian Mixture Models.

It should be noticed that the differentiation between block-level and region-level is not sharp. In fact, a block can be considered to be a region. Therefore, some authors refer to region-level analysis when considering an analysis performed involving several pixels even if the unique criterion to put them together is the spatial connectivity. In the present work, the term block-level has been used to refer to groups of pixels which have been made up by following only a spatial connectivity criterion (usually a circle or a square around a given pixel position), while the term region-level is kept for referring to more general groups of pixels which have been formed using additional connectivity criteria as color, belongingness to the foreground, etc. Therefore, regions are considered to represent a higher level of semantic than blocks.

Obviously, not all of the algorithms proposed in the literature can be unambiguously classified as belonging to one of the three classes mentioned above. As a matter of fact, there is a large number of methods which can be considered to be hybrid because of using an underlying pixel based background supported by some kind of block- or region-level analysis. A very early example of such methods can be found in [Elgammal et al., 2000], where detected foreground pixels are checked against the background model of the pixels in their vicinity (block-level analysis) and the probability of displacement of the detected foreground connected components is computed (region-level analysis); both probabilities are then taken into account for the final pixel classification.

Attending to the update process of the model, background subtraction approaches can be divided into recursive and non-recursive. Such a taxonomy can be found in [Baltieri et al., 2010; Parks and Fels, 2008]. Recursive approaches update the background model as new observations arrive, therefore consuming low resources in terms of computational and memory requirements. Examples of this kind of approaches can be found in [Wren et al., 1997; Stauffer and Grimson, 1999]. On the other hand, non-recursive approaches keep a buffer of the last

incoming video frames to estimate the background. Therefore, non-recursive approaches have higher memory requirements. Nevertheless, since they have a copy of the most recent video frames, they can cope with some challenges as outlier rejection and fast convergence which cannot be easily handled with recursive techniques. Examples of this kind of approaches can be found in [Cutler and Davis, 1998; Elgammal et al., 2000].

Additional taxonomies (unimodal versus multimodal, parametric versus non-parametric approaches, etc.) can be found in the extensive background subtraction literature. Some interesting background subtraction surveys can be found in [Cheung and Kamath, 2004; Piccardi, 2004; Karaman et al., 2005; Parks and Fels, 2008; Benezeth et al., 2010; Brutzer et al., 2011], which are commented in Section 2.4 of this chapter.

2.1.2 Chapter Overview

The following section provides a brief review of some relevant background subtraction approaches. The selection of the presented approaches has been made so as to set a basis of understanding of the underlying techniques employed for the task of background subtraction while providing an overview of the wide field of background subtraction. The focus has been put on the original formulation of the respective techniques. Therefore, the provided references date back to these first formulations.

Section 2.3 provides an overview of background subtraction approaches for environments monitored with PTZ cameras. Whereas this topic has not been on the focus of attention of the work presented here, the extrapolation to PTZ cameras is a question that rapidly arises when approaching the background subtraction topic. Therefore, briefly depicting the general problem and providing some pointers to the relevant literature was unavoidable in this chapter.

Section 2.4 provides an overview of the current state-of-the-art of background subtraction, thereby paying special attention to the trends followed in the surveillance domain and the needs identified, which motivate the work presented here.

2.2 Relevant Approaches

2.2.1 Running Average Model

The most basic approach for modelling the background of a video sequence is to average the value X_t observed at time t at every pixel position (x, y) in the consecutive video frames (this notation will be followed in the rest of this thesis). This average can be approximated by using the following recursive computation:

$$B_{t+1}(x, y) = B_t(x, y) + \alpha(X_t - B_t(x, y)), \tag{2.1}$$

where B_t is the resulting background image and α is a learning rate that controls how fast the background model is adapted to the changes in the scene.

The background image, B_t, can be used to compute a difference image D_t, which contains the value of the difference between each pixel in the incoming images $I_t(x,y)$ and their corresponding values in B_t. The foreground mask F_t is generated according to the following decision rule:

$$F_t(x,y) = \begin{cases} 1 & \text{if } |D_t(x,y)| > \tau, \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where τ is the selected thresholding value. The value of τ should be chosen dynamically so as to adapt to changing viewing conditions, which might affect the noise introduced by the camera. An overview of the general approaches that can be adopted in order to compute a proper value for τ is provided in [Rosin, 2002], where a representative approach for each of the identified categories is evaluated. As an extension to the global thresholding procedure formulated in Equation 2.2, there are several procedures that can improve the detection of foreground, such as, e.g., local thresholding or hysteresis thresholding.

An early approach using this kind of background model is presented in [Kilger, 1992], where a selective updating strategy is used in order to allow for a fast adaptation of the background model while being able to maintain moving objects in the foreground. The used background updating equation is as follows:

$$B_{t+1}(x,y) = B_t(x,y) + (\alpha_1(1 - F_t(x,y)) + \alpha_2 F_t(x,y))(X_t - B_t(x,y)). \tag{2.3}$$

2.2.2 Median Model

An alternative to using the average value of the pixels in a video sequence to model the background of the scene is to use the median value of the last N frames. The main advantage of this approach is that the background image is not degraded by the appearance of foreground moving objects, provided that the background is visible during a number of frames higher than $N/2$. On the other hand, its computation requires a buffer to keep the last N frames.

A method to iteratively approximate the median value, therefore, avoiding the need of a frame buffer, is presented in [McFarlane and Schofield, 1995]. For each incoming frame, the background image is updated as follows:

$$B_{t+1}(x,y) = \begin{cases} B_t(x,y) + 1 & \text{if } I_t(x,y) > B_t(x,y), \\ B_t(x,y) - 1 & \text{if } I_t(x,y) < B_t(x,y). \end{cases} \tag{2.4}$$

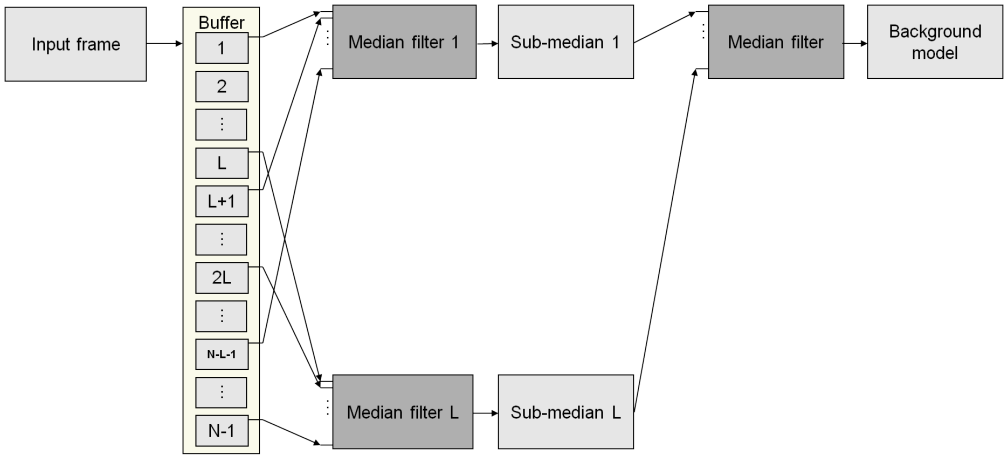


Figure 2.2: *Hierarchical median approach.*

Other alternative methods to the computation of the median value, which aim at providing more robust background models in the case of video sequences with frequently passing by foreground objects have been presented in [Cucchiara et al., 2003; Karaman, 2010]. By increasing the robustness it is possible to reduce the size of the buffer, and thus compute the background image faster. In [Cucchiara et al., 2003], the background model is computed by taking the median over a set of the last $\frac{N-1}{n}$ sub-sampled input frames I_t, where n is the sub-sampling rate, and the background past values with an adequate weight. In [Karaman, 2010] the input frames are divided into L groups of alternated $\frac{N-1}{L}$ frames. Out of these groups median values are computed which are then used to compute the overall median. Figure 2.2 depicts graphically this procedure. The advantage of sub-sampling the input values (or, alternatively, using alternated input frames) for the computation of the median value is that the probability of consecutively considering the appearance of the same moving foreground object in the computation of the background image is reduced. Furthermore, the hierarchical scheme allows for computing the median of only one of the sub-buffers at a time, therefore allowing for a faster computation of the overall median value.

Foreground objects are detected by computing the difference between the incoming video frames I_t and the background image B_t, and thresholding the difference image D_t as discussed in Section 2.2.1.

2.2.3 Running Gaussian Average Model

In order to compute a statistical driven threshold value for the computation of the foreground masks, the variances of the pixel intensities observed at every pixel position can be estimated.

The mean value accounts for the expected value of background pixels. The variance accounts for the noise introduced by the camera, which might vary according to the reflection properties and the illumination conditions at different background positions. Therefore, the set of foreground pixels can be computed on a fully statistical basis. Such a background model has been used e.g. in [Wren et al., 1997].

2.2.4 Gaussian Mixture Model

The background models presented up to now, use a single description for the appearance of the background. In order to describe more complex distributions, Gaussian Mixture Models (GMMs) can be used. The basic idea of this approach is to classify each pixel by using a model of the appearance of the pixel which consists of the combination of different classes. Figure 2.3 shows an example of the empirical distribution of the intensity values produced by a source consisting of three Gaussian modes with different prior, mean and variance values, and its corresponding GMM. The dominant mode could correspond to the background model of an observed scene, the second could be interpreted as projected shadows, and the third one to the foreground objects passing by.

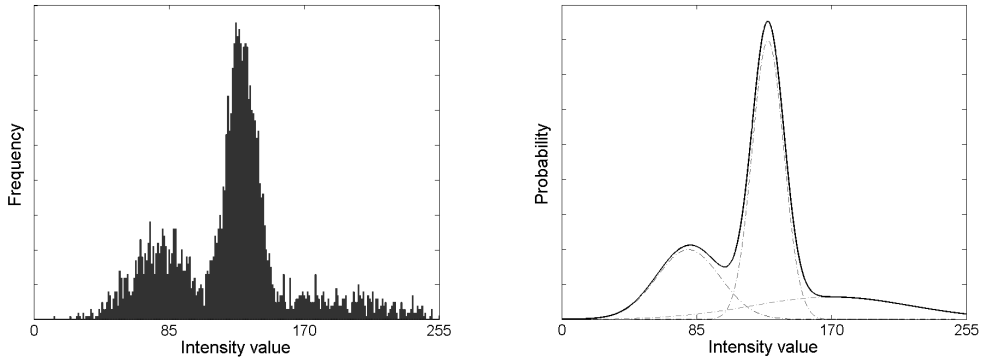


Figure 2.3: *Gaussian Mixture Model. Left: empirical distribution of intensity values. Right: corresponding three-components Gaussian Mixture Model.*

GMMs have the advantage of coping with multi-modal background appearances, as e.g. waving trees, and are able to adapt to the observed scene in real-time with low memory requirements. The original formulation of the GMM for the task of background subtraction was provided in [Friedman and Russell, 1997], where a mixture of three Gaussian distributions was used to model at a pixel level the appearance of the road, shadows and vehicles in a traffic

monitoring application. This system was generalized in [Stauffer and Grimson, 1999], and further developed in [Hayman and Eklundh, 2003] in order to cope with moving cameras.

A thorough explanation of the GMM is provided in Chapter 3.

2.2.5 Non-parametric Model - Kernel Density Estimation

In order to cope with high-frequency variations and arbitrary distributions, non-parametric background models can be used. The probability of observing a given pixel value X_t at time t using the kernel estimator K can be non-parametrically estimated based on the pixel sample $\mathscr{X} = \{X_1, X_2 \dots X_N\}$ as follows:

$$p(X_t) = \sum_{i=1}^{N} \alpha_i K(X_t - X_i), \tag{2.5}$$

where α_i are weighting coefficients (usually chosen to be uniform, $\alpha_i = \frac{1}{N}$).

The probability in Equation 2.5 can be efficiently computed by taking a Normal function $N(0, \Sigma)$ as kernel estimator, assuming independence between the different color channels, and using pre-calculated lookup tables for the kernel function given the intensity value difference $(X_t - X_i)$ and the bandwidth.

The use of non-parametric background models was first proposed in [Elgammal et al., 2000] and [Elgammal et al., 2002]. In order to alleviate the high memory requirements imposed by the need of storing the whole sample set of frames considered for the density estimation, an estimation technique based on mean-shift mode finding is introduced in [Han et al., 2004]. An approach using the balloon variable-size kernel approach, which avoids the estimation of the kernel size parameter, is proposed in [Zivkovic and van der Heijden, 2006].

However, Kernel Density Estimation (KDE) methods have a high computational cost. Moreover, in [Zivkovic and van der Heijden, 2006] it is shown that GMM seems to be a better model for simple scenes while providing a more compact representation which is suitable for further processing steps as e.g. shadow detection.

2.2.6 Codebook Model

As an alternative to statistical models, codebook models have also been proposed for the representation of background. Basically, the appearance of the background is described by means of codewords. The set of codewords describing a pixel constitute its codebook. Each codeword consists of a color vector $v = (R, G, B)$ and some auxiliary parameters. Observed values at each pixel position are compared against the corresponding codebook in order to classify them as either background or foreground.

Although the authors claim in the original formulation [Kim et al., 2004] that codebooks can capture structural background motion over a long period of time under limited memory without making parametric assumptions, the advantages of the system are not clear. For an equal number of descriptions per pixel, codebooks need a higher amount of memory than GMMs. Furthermore, the color and brightness distortion rules used for assigning an observation to an existing codeword must be empirically thresholded, therefore, requiring a parameterization.

The original system is extended in [Kim et al., 2005] by adding a layered approach in order to allow for distinguishing between several temporal characteristics of the objects detected in the scene.

2.2.7 Eigenbackground Model

In order to compensate for illumination changes at a frame level by means of considering spatial correlations, eigenspace models can be used. An eigenspace model is computed by taking a set of N frames and computing both the mean background image and its covariance matrix. The covariance matrix is diagonalized by means of eigenvalue decomposition. In order to reduce the dimensionality of the space, in Principal Component Analysis (PCA) only the M eigenvectors corresponding to the largest eigenvalues are preserved. These eigenvectors are stored in a matrix Φ_{Mb} of size $M \times p$, where p is the number of pixels in a frame. For each input frame I_t, the mean normalized image vector is projected into the eigenspace and back-projected into the image space by using the eigenvector matrix Φ_{Mb} and its transpose, respectively. Since the eigenspace provides a robust model of the background but not of the moving objects, the back-projected input image B_t should not contain moving objects. Therefore, by computing and thresholding the Euclidean distance between the input image I_t and the back-projected image B_t, moving objects can be detected.

A first approach based on eigenspace models is presented in [Oliver et al., 2000]. In [Han and Jain, 2007] this approach is extended in order to allow the process of multi-channel data, the automatic computation of the threshold value, and the adaptation to dynamic scenes by means of Incremental Principal Component Analysis (IPCA). In [Li et al., 2008] the use of an incremental rank tensor-based subspace learning algorithm is proposed in order to better capture the intrinsic spatio-temporal characteristics of a scene.

Subspace modeling is very attractive in real-time computer applications due to its low computational cost at classification time. However, the method requires the allocation of all the training images. Furthermore, the complexity of the original method is considerably increased with the extensions needed for background update, which is a primary requirement in visual surveillance applications.

2.2.8 Texture-based Model

In order to robustly cope with varying illumination conditions, the use of textures has been proposed in [Heikkilä et al., 2004] on a block-wise processing approach and extended to the pixel level in [Heikkilä and Pietikäinen, 2006]. Instead of using color or intensity features, these methods use discriminative texture measures to capture background statistics. These features are computed by using a Local Binary Pattern (LBP). To that aim, the difference of the intensity value of the pixels in the considered neighborhood with the intensity value of the pixel in the center is thresholded, and the result is considered as a binary number (the LBP code). This computation can be easily done as:

$$LBP = \sum_{p=1}^{P} s(X_i - X_c)2^p, \tag{2.6}$$

where X_c is the intensity of the center pixel, X_p the intensity value of the considered P neighboring pixels, and the $s(x)$ function is defined as:

$$s(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0. \end{cases} \tag{2.7}$$

Figure 2.4 depicts graphically the computation of the LBP code for a pixel by using a neighborhood of 3×3 pixels. The generalized LBP operator [Heikkilä et al., 2004] uses a set of P equally spaced neighboring pixels on a circle of radius R as depicted in Figure 2.5.

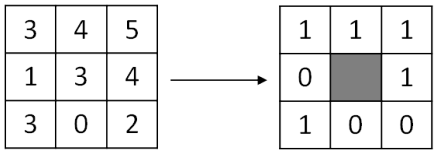


Figure 2.4: *LBP code computation.*

For each considered image position a number K of weighted LBP histograms is computed and consecutively updated following a similar updating process to the one proposed in [Stauffer and Grimson, 1999] for the case of GMM. Background subtraction is performed by computing the distance of the LBP histograms of incoming video frames with the K LBP histograms describing the corresponding image positions.

In order to also consider dynamic textures, the use of a Volume Local Binary Pattern (VLBP) operator is proposed in [Zhao and Pietikainen, 2007], which consists of concatenated LBP

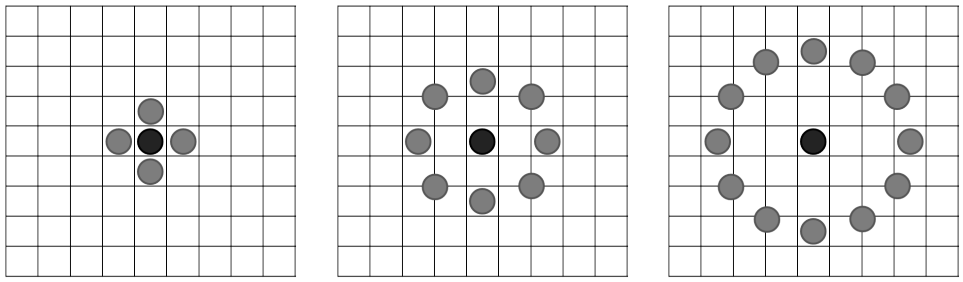


Figure 2.5: *Neighboring pixels set for several values of P and R graphically depicted as in [Heikkilä et al., 2004].*

histograms from three orthogonal planes. In [Zhang et al., 2008] the use of a Spatio-Temporal Local Binary Pattern (STLBP) operator, consisting in a weighted sum of two consecutive LBP histograms, is proposed to alleviate the computational cost imposed by VLBP.

LBP histograms provide a robust manner to cope with illumination changes in dynamic scenes provided that the textures in the observed scene are distinguishable enough. Nevertheless, they do not provide a principled manner to evaluate the distance of new observations to the background models.

2.3 Background Subtraction with Pan Tilt Zoom Cameras

As presented until now, background subtraction approaches consist in modelling the empty scene observed by the camera so as to detect foreground objects by differencing. As the background model might have a different appearance at different image positions, the generated background models might not be valid anymore when the camera moves. Therefore, background subtraction approaches need to incorporate mechanisms in order to detect if the camera is moving and to find the correspondence of the current image plane with respect to the background model, if they have to deal with moving PTZ cameras.

Most of the approaches proposed in the literature to tackle this problem are based on the generation of a mosaic (or panorama) background model representing the different positions which can be captured by the camera (see Section 2.3.1). Nevertheless, there have also been some systems reported in the literature which transform the existing background model into the current image plane so as to use the parts of the background model which are still visible after a camera movement and generate a new model for these parts of the image which were not visible before (see Section 2.3.2).

Independently of the technique used, there are a number of issues which need to be addressed by both systems:

- Image registration, which refers to the process of geometrically aligning two or more images of the same scene taken from different viewpoints. The image registration process can be divided into two fundamental steps: feature detection and matching, and image transformation, which are explained below. The image registration problem can be modeled as a system of n linear equations with p unknowns, where n is the number of considered feature points and p is the number of parameters of the selected motion model. Two excellent surveys on image registration techniques can be found in [Brown, 1992] and [Zitová and Flusser, 2003]. Figure 2.6 depicts the process of image registration.
  - Feature detection and matching, is the process of detecting a set of salient features in both the reference and the current image, and establishing the correspondence. The kind of features employed depend on the application and the kind of sensors used. In the ideal case, the features should be spread over the whole image, easily detectable and not sensitive to the expected image degradation. In the case of absence of enough distinctive features, as in medical images, area-based methods are employed for the feature detection and matching process. Nevertheless, typical video surveillance scenarios offer a rich amount of details and, therefore, feature-based methods, which are less demanding in computational terms, are usually preferred. Examples for such methods are SIFT [Lowe, 1999], SURF [Bay et al., 2008] and FAST [Rosten and Drummond, 2006]. An extensive survey on local invariant feature detectors can be found in [Tuytelaars and Mikolajczyk, 2008].
  - Image transformation, is the process of putting the different images into a common reference coordinate system by means of a transformation model. The type of the transformation model should correspond to the assumed geometric deformation and provide the required registration accuracy. The projective transformation model is the one better describing the transformation between frames captured by a PTZ camera at different camera positions. Nevertheless, it is also the most expensive in terms of computational cost. Therefore, some approaches reported in the literature have also used more simple transformation models as the pure translational, or the affine transform. Good introductions to image formation and geometric transformations can be found in [Hartley and Zisserman, 2004] and [Szeliski, 2010, Chapter 2].
- Image interpolation, which is the process used in order to compute image values in non-integer coordinate positions. Examples of image interpolation techniques are the nearest neighbor, the bilinear and the bicubic functions. Higher-order methods achieve a better performance in terms of accuracy. Nevertheless, Zitová and Flusser observe

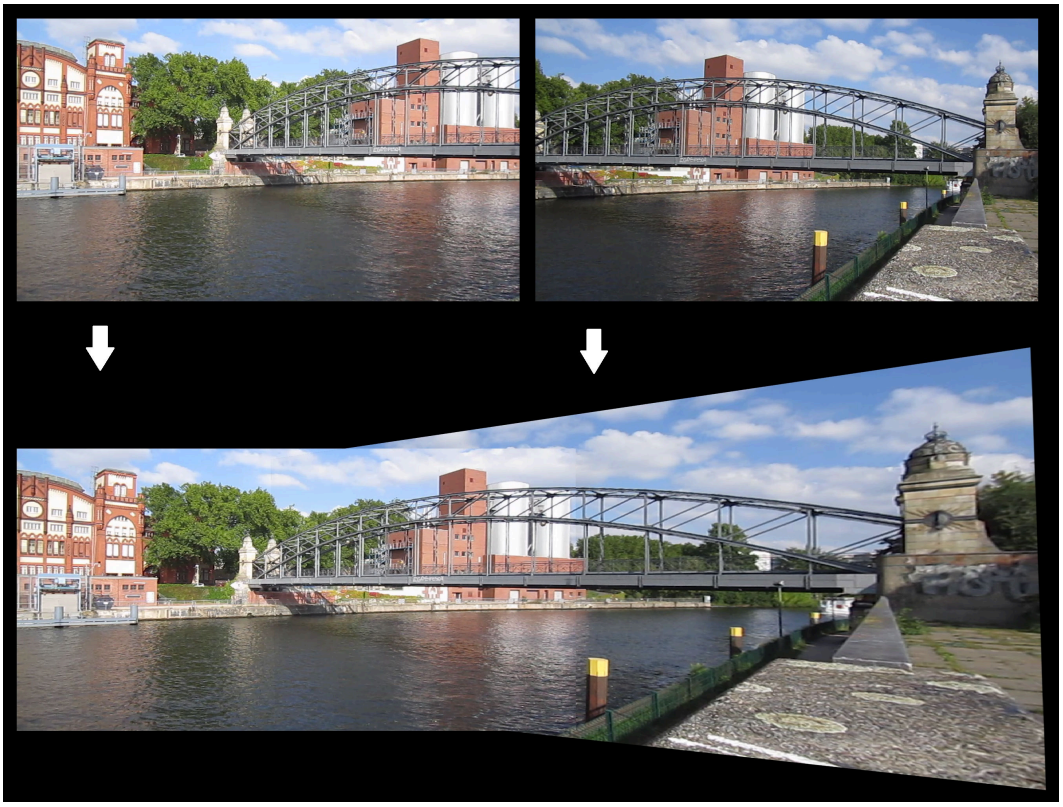


Figure 2.6: *Image registration. Two frames of a video sequence recorded with a PTZ-camera while panning to the right. Top left: First frame. Top right: Two hundred frames later. Bottom: Image registration.*

in [Zitová and Flusser, 2003] that the bilinear interpolation method achieves probably the best trade-off between accuracy and computational complexity and argue that it is, therefore, the most commonly used approach.

- Background model generation and update, which is explained in the following sections 2.3.1 and 2.3.2.

2.3.1 Background Mosaics

A mosaic is an assembled image generated by properly aligning a high number of frames and warping them into a common reference coordinate system. The mosaic contains the background model of the scene along the whole camera range of movement, which can be achieved either off-line by scanning the scene at every possible camera position at the

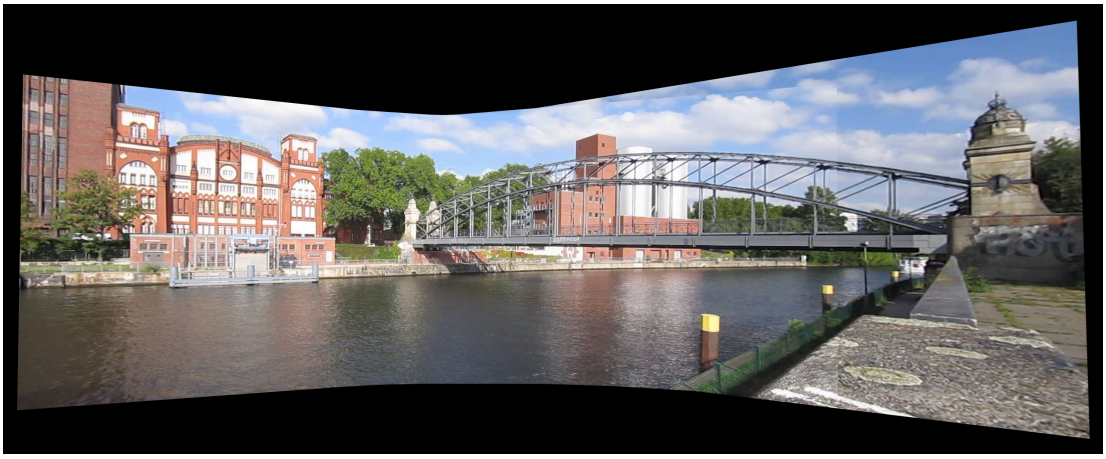


Figure 2.7: *On-line generated mosaic background of a video sequence recorded with a PTZ-camera while panning.*

initialization of the system as e.g. in [Bhat et al., 2000] and in the multi-resolution approach presented in [Sinha and Pollefeys, 2006], or on-line by generating the mosaic as new positions are being discovered, as e.g. in [Brown and Lowe, 2003] and [Bevilacqua et al., 2005]. The main advantage of having a mosaic is the availability of a background model whenever the camera moves to a position where the background has already been initialized.

Common issues regarding mosaic backgrounds are light and color mis-alignments, which arise because of several reasons involving the capturing hardware (changes in the aperture and/or exposure time) and the captured environment (changes in illumination conditions and/or time of day). This problem is closely related to the update of the background model. Mostly, the portions of the mosaic being visible at each camera position [Xue et al., 2011] are updated in a similar fashion as in static camera setups. Nevertheless, there are also some approaches in the literature, as e.g. [Azzari and Bevilacqua, 2006], which introduce an additional tonal alignment step in order to avoid color differences introduced by the hardware. Another issue that should be handled in scenarios with mosaic backgrounds is the accumulation of registration errors propagated by frame-to-frame registration when the camera moves (leading to the problem known as 'looping path' when the camera returns to a previous position). Even if this problem can be solved by global registration approaches, the availability of the whole set of frames is not feasible in real-time environments.

Mosaic representations have also been employed for many other applications [Irani et al., 1995], among them in the high-compression video coding domain [Krutz et al., 2011]. Figure 2.7 depicts an exemplary on-line generated mosaic.

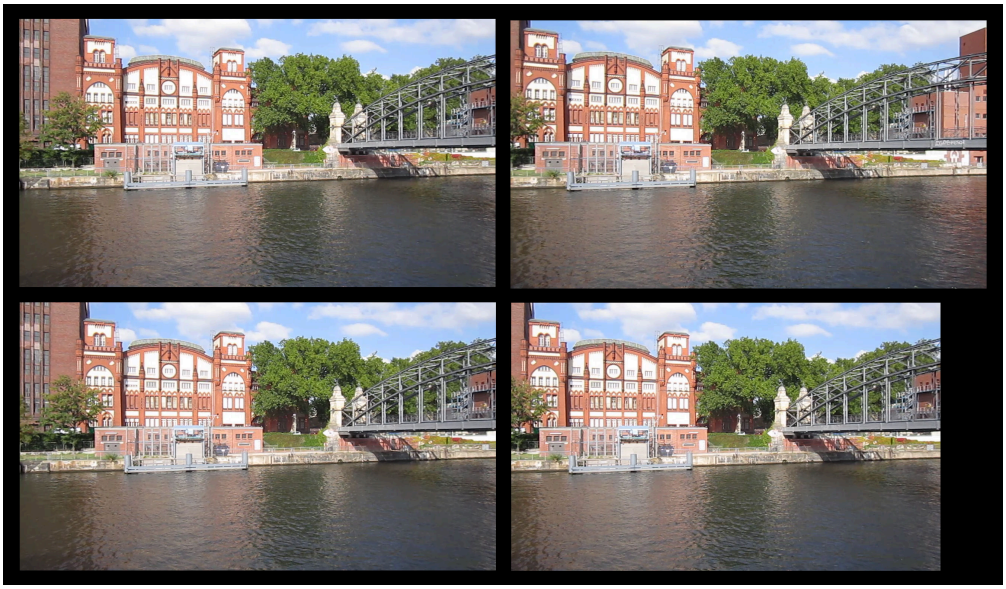


Figure 2.8: *Background model transformation. Top: consecutive frames of a video sequence. Bottom: background model corresponding to the frames on the top. As the camera pans to the right, a part of the background has to be initialized with the new observations.*

2.3.2 Background Transformation

Background transformation-based models basically operate the same as background models for static cameras incorporating a method for camera motion detection and image registration. Each time the position of the camera changes, the existing background model is projected into the new field of view of the camera, so as to use the portion of background existing in the overlapping section between the former and the current field of view. Examples for this kind of background subtraction approaches have been presented in [Kang et al., 2003] and [Robinault et al., 2009]. Although basic approaches use nearest neighbor interpolation methods in order to warp the previous background into the current position, more elaborated methods as the one presented in [Hayman and Eklundh, 2003] handle uncertainties produced by sub-pixel motions and motion blur by means of modelling mixed pixels. Figure 2.8 depicts the described transformation process.

2.4 Current Trends and Conclusions

Due to its low computational load, background subtraction is probably the most common first step in order to detect objects of interest in surveillance applications, especially in the case of using static cameras, and has generated an extensive literature. In the previous sections the

main techniques used to accomplish this task have been presented. These techniques have also been employed in many other deriving approaches which aim at better tackling some of the challenges posed to the background subtraction approach. This section provides an overview of the main trends observed in the background subtraction literature.

Obviously, depending on the application domain, including the characteristics of the observed scenes and computational constraints, the most suitable approach may vary. A study of various background subtraction algorithms in the context of urban traffic surveillance systems is presented in [Cheung and Kamath, 2004]. Special attention is paid to the trade-off between the obtained results and the computational complexity. The good compromise achieved by simple techniques such as adaptive median filtering for the considered domain is highlighted.

In [Piccardi, 2004], a more general selection of different methods covering a wide range of underlying mathematical approaches is presented. A categorization of the presented approaches attending to their speed, memory requirements and segmentation results is provided, aiming at facilitating the design/selection of a background subtraction approach depending on specific system requirements and capabilities. It is highlighted the acceptable accuracy provided by simple methods such as the running Gaussian average and the median filter, the high model accuracy of Gaussian mixture models and the sequential kernel approximation at the cost of higher memory and computation requirements, and the challenge posed by practical implementations to methods addressing spatial correlations.

2.4.1 Background Model Initialization

The first task to be solved by a background subtraction system is the initialization of the model, often referred to as bootstrapping. In controlled environments this is frequently achieved by imposing a training period during which the empty scene is visible. Nevertheless, this strategy is not applicable to general surveillance scenarios. Therefore, the background model needs to be initialized in the presence of moving objects. Even if the use of simple approaches such as a pixel-wise computation of the mean [Koller et al., 1994] or the median [Cutler and Davis, 1998] value may suffice for some applications, there is also a large number of scenarios, specially those involving crowds, where a more elaborated approach to background initialization is necessary. To that aim, usually some kind of spatial information is used. One of the earliest approaches based on this principle is presented in [Gutchess et al., 2001], where the use of optical flow information is proposed. The main idea is that using the optical flow in the vicinity of a pixel is possible to hypothesize if a background pixel is being occluded by a moving object (if the direction of the optical flow is towards that pixel) or if an occluded background pixel is being uncovered (if the optical flow is directed away from that pixel). The method proposed in [Farin et al., 2003] consist in computing the sum of absolute differences of co-located image blocks of the input frames in order to classify them as moving, static foreground or static background; the background image is computed by using a temporal median filter to combine

static background blocks. In [Colombari et al., 2006] a method is proposed which consists in dividing each input frame in patches that are clustered along the time-line in order to select a small number of background candidates, which are then incrementally deemed to be background or not by choosing at each step the best continuation of the current background according to visual grouping principles, therefore considering the spatial correlations that exist within small regions of the background image. A more recent approach which also considers the correlation of neighboring background blocks is presented in [Reddy et al., 2011], where the combined frequency response of a candidate block and its neighborhood is the selection criterion of the blocks considered as background.

A common assumption of the above mentioned methods is that the background scene is visible at some point in time during a training period used to initialize the background model. Later on in this thesis, a system is proposed which relocates the background initialization task to the appropriate point in space and time, i.e. upon the appearance of new static regions.

2.4.2 Illumination Changes and Shadows

While gradual illumination changes are correctly handled by most of the state-of-the-art adaptive approaches, sudden illumination changes and shadows casted by moving objects are still a challenge for most of them. In the case of global illumination changes, texture and, more generally, local based approaches show an improvement over pixel based approaches provided that the textures in the observed scene are distinguishable enough. For the case of casted shadows, all background subtraction approaches show deficiencies which are usually amended in a post-processing step.

Sudden global illumination changes, are usually handled in an spatial context. For instance, the system proposed in [Toyama et al., 1999] retains a representative set of scene background models attending to different lighting conditions (a minimal set would correspond to lights on and off) and chooses the model that produces the fewest number of foreground pixels. Obviously, such an approach requires a previous knowledge of the empty scene under different illumination conditions. Based on the observation that illumination changes can be better handled considering spatial information, the system proposed in [Cristani et al., 2002] combines the results provided by a GMM with spatial information provided by an off-line spatial segmentation of the background in a Bayesian framework. A more general approach which also exploits spatial relationships is presented in [Suau et al., 2009], where the observed scene is corrected by means of a multi-resolution illumination correction approach in order to bring the processed video frames to a reference luminance level. An alternative approach is presented in [Pilet et al., 2008], where the background model is defined by a statistical model of the illumination effects, instead of the pixel intensities. Furthermore, the likelihood of pixel classification also fuses texture correlation clues by exploiting texture histograms trained off-line. Although impressive results are presented, it is assumed that the background

is static and can be trained beforehand, which is a requirement that can be easily fulfilled in the scenario for which the approach is designed for, augmented reality, but not in a common video surveillance scenario.

Regarding the detection of casted shadows, most of the proposed approaches consider other color spaces than RGB which cope better with small illumination changes like HSV, YC_bC_r or the rgs information used in [Elgammal et al., 2002], where r and g are the red and green normalized chromaticity components, which are computed as $r = \frac{R}{R+G+B}$ and $g = \frac{G}{R+G+B}$, respectively, and s is a lightness measure computed as $s = R+G+B$. These color spaces are more insensitive to small changes in illumination; nevertheless, they have the inconvenient of requiring a color transformation for each pixel in the image. Therefore, some approaches produce a first classification in the RGB color space and transform only the pixels belonging to the foreground to a different color space in order to check for shadows.

A survey on shadow detection approaches is presented in [Prati et al., 2003], where the different contributions reported in the literature are classified in four classes: statistical parametric, statistical non-parametric, deterministic model-based and deterministic non-model-based. Out of the evaluated approaches, the results provided by those presented in [Horprasert et al., 1999] and [Cucchiara et al., 2002] are highlighted. The approach in [Horprasert et al., 1999] classifies pixels as foreground, background, shadowed background or highlighted background, depending on the chromaticity and brightness distortion measured by projecting the observed value into a line going through the origin of the RGB space and the expected value for every pixel position. The approach in [Cucchiara et al., 2002] classifies pixels as foreground or background depending on the distance in the HSV color space of the observed to the expected values for every pixel position, thereby exploiting the different effect that illumination conditions have on the hue, saturation and value channels.

2.4.3 Post-processing and Spatial Consistency

One of the problems faced by background subtraction techniques is the noise introduced by the camera (and, eventually, by video coding techniques) in the video sequences. As shown in Section 2.2, this problem is usually tackled by means of choosing an appropriate thresholding approach or by statistical means. Regardless of the adopted approach, there is always a part of the noise which cannot be effectively handled by background subtraction and is usually removed in a post-processing step by imposing a spatial consistency criterion.

A comparison between seven state-of-the-art algorithms with and without the application of post-processing techniques is presented in [Parks and Fels, 2008]. The post-processing chain consists in a set of common techniques comprising morphological operators for noise removal, blob thresholding by means of area size, saliency and measured optical flow, and object-level feedback. Furthermore, the article is accompanied by a software library of the

tested algorithms [1]. While the results provided by noise reduction by means of morphological operators are evidently better, the results provided by the rest of the post-processing steps are not always beneficial, especially in the case of the saliency and optical flow tests, which can even significantly decrease the performance a given algorithm if not properly parameterized.

A similar study is presented in [Benezeth et al., 2010], where the influence of three different spatial consistency criteria is evaluated. The evaluated criteria are a median filter of the classification of the pixels situated in a window of 5×5 size around the center pixel, a $close(open(F, W))$, where F is the foreground mask provided by the background subtraction algorithm and W is the size of the morphological operator (set to 5×5), and a Markovian prior. Since adding a Markovian prior leads to a Maximum a Posteriori formulation of the foreground classification, an optimization scheme is needed to that aim. In this study, the iterated conditional modes optimizer is used. The results presented show a strong increase of the performance of the evaluated algorithms with any of the tested post-processing steps. Therefore, the Markovian prior is considered a weaker solution because of its considerably higher computational load. Spatial coherency of the pixel labels has been proposed in e.g. [Migdal and Grimson, 2005; Yin and Collins, 2007].

2.4.4 Hybrid Approaches

A further method to improve the quality of the foreground masks provided by background subtraction is the use of hybrid systems where the detections provided by several systems based on different detection principles are fused. Examples of such systems can be found e.g. in [Jabri et al., 2000; Javed et al., 2002; Shen, 2004; Haque et al., 2008]. In [Jabri et al., 2000], two background models are used, a color model and an edge model. The color background model consists in a running average Gaussian obtained along the video sequence. The edge model is generated by applying the Sobel edge operator to each color channel and updating the resulting horizontal and vertical difference images along the sequence as in the case of the color model. Background subtraction is performed by combining the detections provided by the subtraction of each of the scene models. The system in [Javed et al., 2002] also combines color and gradients cues, plus multiple levels of analysis, pixel, region and frame, in order to better cope with illumination changes. Sudden illumination changes are detected at the frame level as in [Toyama et al., 1999]. In [Shen, 2004], the masks provided by means of background subtraction and temporal differencing are combined in order to more robustly cope with illumination changes because of the lower illumination sensitivity of the temporal differencing approach. In [Haque et al., 2008], the detections provided by a GMM are checked against spatial coherence, therefore combining color with spatial information.

[1] http://dparks.wikidot.com/background-subtraction

A study of several basic and hybrid approaches is presented in [Karaman et al., 2005], where a quality assessment is provided based on a set of videos comprising outdoor and indoor sequences selected to cover a wide range of the challenges posed to the task of background subtraction. The results are objectively evaluated by means of manually annotated ground truth for the selected video sequences. Color (in comparison to other features as luminance or edges) is shown to be the most robust cue for foreground segmentation. Furthermore, it is observed that the best results are achieved by optimally combining complementary feature cues. As future lines of research, it is highlighted the importance of using more sophisticated background models, whereas it should be noticed that the underlying background models of the whole set of analyzed approaches are unimodal, and the consideration of the segmentation problem as a multi-class classification problem (instead of only considering two classes).

2.4.5 Qualitative Evaluation

In order to decide what kind of background subtraction approach is more appropriate for a given scenario, several considerations have to be taken into account. The most obvious one is the processing time, which should provide real-time capabilities for an on-line surveillance system but can be relaxed in an off-line application as video coding or a medical analysis. Once the required processing time and memory needs have been guaranteed, the quality of the provided foreground masks has to be evaluated.

The qualitative evaluation of background subtraction approaches is a cumbersome task because of the need of ground truth data. Therefore, some evaluations only use a few labeled frames as [Toyama et al., 1999], where a manually annotated ground truth for only one frame of each video sequence is considered, or a small set of sequences [Karaman et al., 2005], where the evaluation is based on five sequences. Alternatively, the subjective evaluation of human experts [ITU-T, 1996], ground truth free evaluation approaches [Erdem et al., 2001; Chalidabhongse et al., 2003; SanMiguel and Martínez, 2010], or automatically generated ground truth data [Grossmann et al., 2005], can be used. A survey on the performance measure of background subtraction approaches is presented in [Elhabian et al., 2008]. A compendium of the of the performance measures most commonly used for the evaluation of background subtraction approaches is provided in Appendix B, Performance Metrics.

Still, a pixel-wise evaluation based on ground truth data seems to be the most reliable method and the one providing the most accurate insights into the merits and weaknesses of the evaluated methods, which is of crucial importance for the further development. The generation and provision of ground truth data has also seen a valuable progress in the recent years [Tiburzi et al., 2008; Brutzer et al., 2011; Goyette et al., 2012]. The dataset proposed in [Tiburzi et al., 2008] provides a set of video sequences with ground truth data based on foreground objects which were recorded in a chroma studio and segmented with chroma-key techniques. These objects were later inserted in the background sequences. The problem of this approach, is that

shadows and background to foreground occlusions are not well represented in the dataset. In order to provide an accurate ground truth of shadows, the dataset proposed in [Brutzer et al., 2011] provides a set of artificial video data generated by using high quality 3D-models and ray-tracing techniques. Nevertheless, it is commonly acknowledged that synthetic data does not faithfully represent the full range of real data [Elhabian et al., 2008]. The recently proposed CDnet dataset in [Goyette et al., 2012], which was proposed for the IEEE Workshop on Change Detection, held in conjunction with the IEEE Conference on Computer Vision and Pattern Recognition 2012, provides a set of 31 video sequences divided in six categories (Baseline, Dynamic Background, Camera Jitter, Shadows, Intermittent Object Motion, and Thermal), which cover a wide range of the challenges faced by background subtraction approaches. The existence of an annotated ground-truth foreground, background, and shadow region boundaries allows for an objective assessment of change detection algorithms. Furthermore, the dataset is accompanied by a website where the evaluated approaches are ranked attending to a set of seven pixel-based performance measures. This ranking is continuously updated with the results provided by the users of the dataset, therefore, allowing for a rapid comparison and ranking of new methods with the state-of-the-art algorithms. Due to the wide range of scenarios covered and to the good comparability offered with state-of-the-art algorithms, this dataset is extensively used in this thesis to present the results of the proposed algorithms in a compact form. For the task of detecting new static objects, additional specialized datasets are also used. A thorough description of the datasets used in this thesis and some pointers to other relevant datasets is provided in Appendix A, Description of Datasets.

Due to the good compromise regarding the quality of the segmentation results, the processing time and the memory requirements, GMM has been chosen as the basis method for the developed algorithms in this thesis. A thorough study of the different algorithms which have been proposed for the use of GMM for the task of background subtraction, their merits and deficiencies is provided in Chapter 3. Based on this study, an improved GMM is proposed.

3

Improved Gaussian Mixture Models

3.1 Introduction

Per-pixel adaptive Gaussian Mixture Models (GMMs) have become a popular choice for the detection of intruding objects by means of background subtraction in surveillance scenarios observed by static cameras, because of their ability to achieve many of the requirements of a surveillance system, e.g. adaptability and multimodality, in real-time with low memory requirements. In a nutshell, the basic approach consists of modelling the history of each pixel by using a mixture of K Gaussian distributions which are updated by means of an EM-like algorithm and using these models in order to classify new pixel values as either background or foreground, depending on the existence or not of a Gaussian mode which supports them with sufficient evidence.

Gaussian Mixture Models present the advantage of being able to adapt to changes in the scene, and to accommodate multi-modal background appearances in order to represent repetitive motion of scene elements. Moreover, by using multiple descriptions for each pixel, the model can be continuously updated in order to fit new observations without affecting to the existing background model. Furthermore, the fact of using Gaussian distributions to model the background at each pixel position, allows for the computation of an automatic thresholding value for the classification of the observed pixels values.

Nevertheless, as a result of the updating algorithm used for the estimation of the parameters of the underlying distribution, state-of-the-art GMM-based approaches often suffer from the problem of converging to poor solutions related to singularities and local maxima. This chapter presents a system which improves the state-of-the-art GMM-based approaches by means

of incorporating a novel variance controlling scheme, which aims to adaptively compute an appropriate value for the initialization of the variance parameter of new modes and to control the variance of existing modes so as to avoid a degeneration of the model. The proposed method achieves better background models and is low demanding in terms of processing time and memory requirements, therefore making it especially appealing in the surveillance domain.

After briefly reviewing the EM algorithm and the different variants that have been derived of it, which set the basis of state-of-the-art GMM approaches for the task of background subtraction, in Section 3.2, Section 3.3 provides an overview of some relevant state-of-the-art GMM based approaches. Thereby, their merits and weaknesses are analyzed. Section 3.4 presents the proposed model, which analogously to the Split and Merge EM algorithm, splits over-dominating modes. Therefore, an appropriate splitting operation and the corresponding criterion for the selection of candidate modes for the case of a non-stationary underlying distribution are derived. The selection criterion is based on a novel adaptive variance controlling value, which is also used in order to properly initialize new created modes. In Section 3.5 experimental results are provided, showing that the presented algorithm achieves better segmentation results than its predecessors. Section 3.6 concludes this chapter.

The content of this chapter has been partially published in *'Splitting Gaussians in Mixture Models'*, in the Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2012 [Heras Evangelio et al., 2012].

3.2 The Expectation Maximization Algorithm

The Expectation Maximization (EM) [Dempster et al., 1977] algorithm is a general approach used to iteratively compute the maximum likelihood estimate of the parameters of an underlying distribution from a given dataset when the data is incomplete (or has missing values). By incomplete data is understood the observed data $y \in \mathcal{Y}$, which is used to indirectly observe the complete data $x \in \mathcal{X}$, assuming that there is a mapping $x \rightarrow y(x)$. The EM algorithm aims at finding a parameter set ϕ which maximizes $g(y|\phi)$ given an observed y, by making use of the associated family $f(x|\phi)$.

The EM algorithm can be used both when the observed data has indeed missing values, and when the likelihood function to be maximized is analytically intractable but can be simplified by assuming the existence of missing data, which is commonly the case in pattern recognition applications.

The algorithm consists of two steps, which are repeated iteratively: the expectation and the maximization step. In the expectation step (E-step), the expected value of the log-likelihood

function of the complete data x is computed using the observed data and current estimates of the parameters:

$$Q(\phi|\phi^t) = E[\log f(x|\phi)|y,\phi^t]. \tag{3.1}$$

In the maximization step (M-step), the expectation computed in the first step is maximized with respect to the estimated parameters:

$$\phi^{t+1} = \underset{\phi}{\operatorname{argmin}}\, Q(\phi|\phi^t). \tag{3.2}$$

These two steps are repeated iteratively until convergence, which is guaranteed to a local maximum.

The set of updating equations assuming an underlying distribution described by a mixture of K Gaussian distributions of the form

$$p(x|\phi) = \sum_{k=1}^{K} \omega_k p_k(x|\phi_k) \tag{3.3}$$

is derived in [Bilmes, 1998] by means of the the observed incomplete data set $\mathcal{Y} = \{y_i\}_{i=1}^N$ for each Gaussian $k \in K$ as:

$$\omega_{k,t+1} = \frac{1}{N}\sum_{i=1}^{N} p(k|y_i,\phi), \tag{3.4}$$

$$\mu_{k,t+1} = \frac{\sum_{i=1}^{N} y_i p(k|y_i,\phi)}{\sum_{i=1}^{N} p(k|y_i,\phi)}, \tag{3.5}$$

$$\sigma_{k,t+1} = \frac{\sum_{i=1}^{N} (y_i-\mu_{k,t+1})(y_i-\mu_{k,t+1})^T p(k|y_i,\phi)}{\sum_{i=1}^{N} p(k|y_i,\phi)}, \tag{3.6}$$

where equations 3.4 to 3.6 perform the expectation and the maximization step simultaneously.

The main problems when using the EM algorithm in order to model multi-variate data by means of finite mixtures are the selection of the number of components to be used and the initialization parameters of the components. Too many components will over-fit the data, while too few components will not be flexible enough to properly describe the underlying distribution. Some references tackling the problem of model order selection can be found in [Figueiredo and Jain, 2002; Zivkovic and van der Heijden, 2004], which basically propose to start with a large number of components and introduce a prior in order to lead the algorithm

converging to more compact models. Further references on this topic can be found in these two publications. The parameter initialization problem can lead the algorithm to converge to local maxima. Local maxima arise when there are too many components in one part of the feature space and too few in another, since moving components from overpopulated to underpopulated regions is not possible without passing through positions with a lower likelihood. Obviously, this will never happen with the EM algorithm, since the model parameters are changed at each iteration step so as to increase the log likelihood function. Furthermore, in the case of using the Gaussian distribution as a basis function, the parameter initialization problem can derive in the algorithm converging to singularities, which are produced when the mean value of one of the components in the mixture model is equal to one of the data points, going therefore the log likelihood function to infinity. Thus, the maximization of the log likelihood function is not a well posed problem in the case of Gaussian mixtures. There is a huge number of publications attempting to heuristically alleviate the problem posed by local maxima and singularities. A recently simple yet effective method to overcome these problems was proposed in [Ueda et al., 2000], where an algorithm is presented which simultaneously merges two Gaussians in overpopulated regions and splits a Gaussian in underpopulated regions. To that aim, a criterion aiming at selecting the split and merge candidates is developed.

One of the premises of the EM algorithm is breaking down a difficult problem (the maximization of the likelihood function) into two simpler problems (the expectation and the maximization steps). In the case that one of these two problems remains intractable, a partial implementation can be provided, leading to the Generalized Expectation Maximization (GEM) algorithm. In [Neal and Hinton, 1998] a view of the EM algorithm in terms of a Kullback-Liebler divergence problem is presented, which justifies such incremental versions (incremental, sparse and winner-takes-all versions).

In the case of large data sets, numerical procedures as the EM algorithm can become very expensive. For these cases, stochastic approximation procedures can be considered. In [Titterington, 1984], several of such methods are developed and the link of one of such methods to the EM algorithm is made.

3.3 Gaussian Mixture Models for the Task of Background Subtraction

Most state-of-the-art GMMs follow the formulation presented in [Stauffer and Grimson, 1999], thereby modelling the history of each pixel by a mixture of K Gaussian distributions. The probability of observing a given pixel value X_t at time t is estimated as:

$$P(X_t) = \sum_{k=1}^{K} \omega_k \mathcal{N}\left(X_t, \mu_k, \Sigma_k\right), \tag{3.7}$$

where ω_k are the weights respectively associated to each of the modes $k \in \{1 \dots K\}$ describing a pixel, and $\mathcal{N}\left(X_t, \mu_k, \Sigma_k\right)$ is a normal density of mean μ_k and covariance matrix Σ_k, which is usually assumed to be the diagonal matrix $\sigma_k^2 I$, therefore, assuming that the red, green, and blue pixel values are independent and have the same variance, making thus the inversion of Σ_k easier. The mixing weights are non-negative and add up to one. The components are sorted according to their relevance and the background model is approximated by the first B components such that:

$$B = \arg\min_k \left(\sum_{k=1}^{B} \omega_k > T \right), \tag{3.8}$$

where $B \leq K$, T is a predefined threshold indicating the minimum portion of the data that should be assumed to be background, and the sorting criterion is given by the value $s_k = \omega_k / \sigma_k$ in descending order. That means, modes with a high weight and a small variance tend to the top of the list of the modes corresponding to a given pixel and, thus, to be part of the background model. The model is adapted by means of an on-line EM algorithm, used to approximate the maximum likelihood of the parameters describing the underlying non-stationary distribution in a recursive manner. The parameters of the estimated modes are updated by adopting a 'winner-takes-all' strategy. This means, that only the parameters of the distribution corresponding to the selected matching mode are updated at a time. The matching mode is selected by computing the distance of the observed pixel value X_t to the modes of the model in a descendant order and assuming the first mode m which distance is lower than τ times its standard deviation to be the best match, where τ is usually set to a value within 2.5 and 3 [Stauffer and Grimson, 1999; Zivkovic, 2004]. If none of the available modes matches the current pixel value X_t, a new mode is created with X_t as its mean, a default value for the variance and a low prior weight. If none of the modes in the model is free, this new created mode replaces the one with the lowest s_k.

For every new frame, the GMM corresponding to each pixel is updated as follows:

$$\omega_{k,t} = (1 - \alpha)\, \omega_{k,t-1} + \alpha M_{k,t}, \tag{3.9}$$

where $k = 1 \dots K$, α is a constant learning rate, and $M_{k,t}$ is a binary function with value 1 for the matched mode and 0 otherwise. The recursive computation of Equation 3.4 would require to use a variable $\alpha_t = 1/(t+1)$. By using a constant $\alpha = 1/T$, being T a pre-defined integration interval, the system introduces a forgetting factor which allows the system to better adapt to the recently observed samples. Furthermore, the μ_m and σ_m parameters of the matching distribution m are updated as:

$$\mu_{m,t} = \left(1 - \rho_{m,t}\right) \mu_{m,t-1} + \rho_{m,t} X_t, \tag{3.10}$$

$$\sigma_{m,t}^2 = \left(1-\rho_{m,t}\right)\sigma_{m,t-1}^2 + \rho_{m,t}\delta_{m,t}^T\delta_{m,t}, \tag{3.11}$$

where $m \in \{1 \dots K\}$ is the matched mode, $\delta_{m,t} = \left(X_t - \mu_{m,t}\right)$ and $\rho_{m,t}$ is a learning rate calculated as follows:

$$\rho_{m,t} = \alpha\mathcal{N}\left(X_t|\mu_m,\sigma_m\right). \tag{3.12}$$

Due to the good compromise between segmentation results, processing time and memory requirements, GMM have been extensively used in the surveillance domain. Nevertheless, there are still some improvement possibilities in the formulation of [Stauffer and Grimson, 1999]. Some of them have been addressed in numerous publications. The most relevant improvements in the surveillance domain are summarized in the following. The notation of the respective papers has been slightly modified so as to use a uniform one and thus allow for an easy comparison among the different approaches. Furthermore, the variable t is used to refer to discrete points in time associated to the consecutive frames of the analyzed video sequence and is, therefore, meant to be a member of the set of natural numbers excluding zero ($\mathbb{N}^+$).

The initialization of the background model is improved in [Kaewtrakulpong and Bowden, 2001] by introducing a learning phase of length L, where the model is updated following expected sufficient statistics update equations, followed by a steady phase where the model is updated following the L-recent window of [Stauffer and Grimson, 1999]. During the learning phase, the weights $\omega_{k,t}$ are updated as follows:

$$\omega_{k,t} = \left(1-\frac{1}{t}\right)\omega_{k,t-1} + \frac{1}{t}M_{k,t}, \tag{3.13}$$

and the μ_m and σ_m parameters of the matching distribution m are updated as in equations 3.10 and 3.11 by using a learning rate calculated as:

$$\rho_{m,t} = \frac{1}{\sum_{i=1}^{t} M_{m,t}}. \tag{3.14}$$

During the steady phase, the weights $\omega_{k,t}$ are updated as following Equation 3.9 with a constant learning rate $\alpha = \frac{1}{L}$ and the μ_m and σ_m parameters of the matching distribution m are updated as in equations 3.10 and 3.11 by using a learning rate calculated as:

$$\rho_{m,t} = \frac{\alpha}{\omega_{k,t}}. \tag{3.15}$$

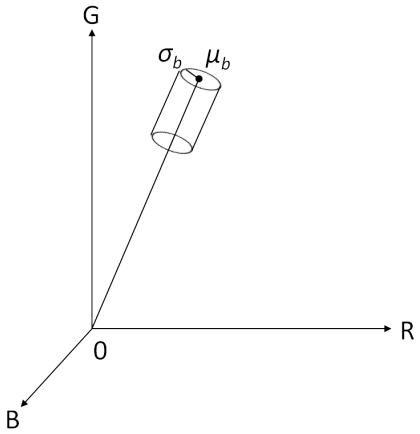


Figure 3.1: *Chromaticity and brightness distortion of a given color I_i with respect to a reference color E_i in the three dimensional RGB color space.*

In this way, the model can be more robustly initialized even in the presence of moving objects, which led to "long" duration ghosts in the model of [Stauffer and Grimson, 1999] [1]. Furthermore, modes with a lower evidence $\omega_{k,t}$ are updated more rapidly. Moreover, [Kaewtrakulpong and Bowden, 2001] introduced the use of a shadow detection algorithm, which is applied to non-background pixels. To that aim, the color model of [Horprasert et al., 1999] is used. Thereby, the brightness and the chromaticity components of a given color are analyzed separately. Figure 3.1 depicts graphically the classification process. The line passing through the origin of the RGB space and the expected color (E_i) is the expected chromaticity line. The brightness distortion α_i and the chromaticity distortion CD_i of a given color I_i with respect to a reference color E_i is computed as:

$$\alpha_i = \underset{\alpha}{\operatorname{argmin}} (I_i - \alpha E_i)^2, \tag{3.16}$$

$$CD_i = \| I_i - \alpha_i E_i \| . \tag{3.17}$$

Non-background pixels are considered as shadowed pixels if $\lambda < \alpha_i < 1$, being λ the brightness threshold, and CD_i is within the tolerated chromaticity distortion ($< \tau\sigma$). By first classifying pixels in the RGB space and then only checking non-background pixels for shadows, computational resources needed for the color model transformation are saved.

[1] The updating equations for the μ_m and σ_m parameters in [Kaewtrakulpong and Bowden, 2001] has been corrected here, since those in the original paper were erroneous.

In [Lee, 2005] the use of an adaptive learning rate calculated for each Gaussian at every frame is proposed. Thus, parameter learning of each Gaussian follows a $1/t$ schedule for the first observations and gradually approaches the basic recursive filter as in [Stauffer and Grimson, 1999]. Therefore, the convergence rate of new created Gaussians is improved without compromising the model stability. This is achieved by using a variable $\eta_{m,t}$ to count the number of assignments to a given matched Gaussian m and computing the learning rate of the matched mode as:

$$\rho_{m,t} = \frac{1-\alpha}{\eta_{m,t}} + \alpha. \tag{3.18}$$

The counter $\eta_{m,t}$ is incremented by one when a Gaussian is updated and reset to one when a Gaussian is reassigned. Therefore, the learning rate applied to each Gaussian is the same throughout all stages of the system.

In [Zivkovic, 2004] the model is provided with the capability of adaptively choosing the number of components needed for each pixel. To that aim, prior knowledge for the multinomial distribution defined by the weights $\omega_{k,t}$, with $k = 1 \ldots K$, corresponding to each GMM is introduced by using the Dirichlet prior with negative coefficients. This means that a given class k is only accepted if there is enough evidence from the data for its existence. This can be efficiently implemented by changing the updating equation of the weight corresponding to the GMM of each pixel, Equation 3.9, as follows:

$$\omega_{k,t} = (1-\alpha)\,\omega_{k,t-1} + \alpha M_{k,t} - \alpha c_T, \tag{3.19}$$

where c_T is the introduced prior knowledge. After each update, the weights $\omega_{k,t}$, with $k = 1 \ldots K$, corresponding to each GMM are normalized so that they add up to one. Modes with negative weights are eliminated. In this manner, the number of components of the mixture used for each pixel can be constantly adapted, gaining thus in computational speed at those pixels that can be modeled with a lower number of components. Furthermore, similarly to [Kaewtrakulpong and Bowden, 2001], the mean and variance values of the matched modes are updated by using a learning rate $\rho_{m,t} = \frac{\alpha}{\omega_{k,t}}$.

Further directions of enhancement have been presented in a very extensive literature. Some examples of different research directions can be found in [Cristani et al., 2002; Zang and Klette, 2004; Migdal and Grimson, 2005; Appiah and Hunter, 2005; Gorur and Amrutur, 2011; Robinault et al., 2009].

In [Cristani et al., 2002] the problem posed by sudden illumination changes is tackled by using spatial information based on an off-line generated spatial segmentation of the background. Although the formulated approach has a deep technical sound, the fact of relaying on an off-line generated spatial segmentation of the background prevents the applicability of the

system in a general scenario. In [Zang and Klette, 2004] the use of temporal frame differences and some morphological operations are proposed in order to impose a spatial pixel labeling consistency. Nevertheless, it is not specified how the temporal difference frames should be thresholded, being this a very sensitive factor for the overall system performance. A more robust form of imposing a spatial consistency in pixel classification is presented in [Migdal and Grimson, 2005], where the use of Markov random fields of binary segmentation variates is proposed. However, the use of Markov random fields for pixel classification imposes long computational times.

An implementation of a simplified version of the algorithm proposed in [Stauffer and Grimson, 1999] on a Field Programmable Gate Array (FPGA) is presented in [Appiah and Hunter, 2005]. In [Gorur and Amrutur, 2011], a windowed weight update scheme, which is also suitable for a hardware implementation, is proposed to reduce execution time of the original algorithm.

A method for using GMMs with pan-tilt-zoom cameras is proposed in [Robinault et al., 2009].

A comprehensive study of the extensive literature derivate of the system proposed in [Stauffer and Grimson, 1999] can be found in [Bouwmans et al., 2008], where the method is exhaustively analyzed from the perspectives of the problems it has to afford and the mathematical and algorithmical solutions that have been proposed to tackle them in over 150 papers.

The approach presented in the following section aims at improving the quality of the generated models, and therefore the detection results, at the pixel level. Later on in this thesis, in Chapter 5 it is shown how the achieved results can be further improved by incorporating region-based information into the model.

3.4 Splitting Gaussians in Mixture Models

The methods reviewed above use on-line variations of the EM algorithm adapted to the case of estimating the parameters of non-stationary underlying distributions. The underlying distributions are considered to be non-stationary, since the background of the observed video sequence might evolve along time due to changes in the illumination or the geometry of the scene. However, the EM algorithm is sensitive to initialization when fitting finite mixtures due to its greedy nature. That means, that depending on the initialization parameters, it might converge to different solutions. This issue is even more exacerbated in the on-line approximations used for the task of background subtraction due to two reasons. The first one is the winner-takes-all updating strategy, which assigns each observed sample to a single Gaussian of the mixture model. The second one is the matching criterion used, which takes the first matching distribution as the matched one. As a consequence, the computed GMMs can provide poor representations of the scene background in some situations.

This problem has been already observed in several publications. Nevertheless, the analysis of the problem that they offer is not complete. In [Porikli and Tuzel, 2005] it is claimed that the on-line EM usually converges to the most significant modes, therefore preventing from properly modelling multi-modal distributions. Furthermore, it is claimed that the estimated variance is always much smaller than the actual variance. Obviously, this work is restricted to the algorithm presented in [Stauffer and Grimson, 1999], whose sorting strategy of the modes favors lower variances (the sorting criterion attends to $s_k = \omega_k/\sigma_k$) and updates the matching modes proportionally to the likelihood of the sample. In opposite to that, in [Bouttefroy et al., 2010] it is claimed that the methods presented in [Stauffer and Grimson, 1999] and [Lee, 2005] can degenerate in what they call 'saturated' pixels, which are pixels with a GMM dominated by a mode with a too large variance, so that the same mode is able to accommodate values which should be modeled by different modes. Nevertheless, the updating equations proposed in [Bouttefroy et al., 2010] flawed.

A careful analysis of the methods presented in [Stauffer and Grimson, 1999; Kaewtrakulpong and Bowden, 2001; Lee, 2005; Zivkovic, 2004], reveals that both observations might hold in practical systems. In fact, a too low value for the initialization of the variance of new created modes may lead the model to over-fit some boundary of the feature space, while a too large value may lead the model to under-fit the underlying distribution. Furthermore, depending on the specific updating strategy, the model can converge to different solutions. Using the sorting criterion and matching mode parameters update in [Stauffer and Grimson, 1999] and Equation 3.10 to 3.12 the system might tend to under-estimate the variance of the modes, while using those proposed in [Zivkovic, 2004] the main mode might stretch and thus over-dominate weaker distributions.

In order to tackle these problems, the method presented in this section incorporates a strategy to adaptively choose an appropriate value for the initialization of new created modes. Furthermore, following the same guiding principle as in [Ueda et al., 2000], this value is used in order to define a splitting operation and the corresponding selection criterion to avoid over-dominating modes.

3.4.1 Background Initialization

The GMM for each pixel is initialized at system start. To that aim, the observed value at each pixel is used as its mean value and a guess for the initialization of the variance parameter is made, therefore initializing each GMM with an unique mode, which describes the background of the scene at this point in time. The variance term of each mode accounts for the variation of the values corresponding to the given distribution. These variations are introduced by the camera noise, the kind of surface and the kind of object (moving objects usually exhibit a higher variance than static ones). Correctly initializing this parameter is of crucial importance since it has a significant implication on the behavior of the model. A too low value may

lead the system to generate several modes to model a unique on-time distribution, therefore over-fitting some boundary of the feature space. Conversely, a too large value may lead the model to accommodate samples from different distributions into a unique mode, therefore under-fitting an underlying multi-modal distribution.

For the estimation of the variance parameter, the deviation of each pixel value from the first to the second frame is used ($X_{t=1}$, $X_{t=2}$). Thereby, it is assumed that most of the pixels in consecutive frames, respectively, belong to the same distribution. Furthermore, it is assumed that most of the pixels belong to the background and can, therefore, be described by Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ with similar covariance matrices $\sigma_b^2 I$. If both assumptions hold, then the distribution of the deviations is also Gaussian $\mathcal{N}(0, 2\sigma_b^2 I)$. Therefore, the median of the absolute deviations med can be used to estimate the standard deviation of the former distributions as:

$$\hat{\sigma}_b = \frac{med}{0.68\sqrt{2}}. \tag{3.20}$$

A similar method was used in [Elgammal et al., 2002] in order to estimate the bandwidth of the kernel for each pixel independently. In the estimation presented here, the computation has been extrapolated to the frame level by assuming that most of the pixels belong to the background. While this certainly is not always the case, the only consequence of including some foreground pixels in this computation would be an over-estimation of the variance corresponding to background pixels. The higher the number of moving objects in the scene, the higher the over-estimation. In practice, this does not affect much further detection results since, after this first estimation, the variance of each pixel is individually updated to match the underlying distribution. As it is shown in Section 3.5, Evaluation, the described method converges to appropriate values even if this first estimation drifts because of violation of the assumptions above. Nevertheless, if this value can be better estimated, the convergence of the system to an appropriated model can be sped up. In Section 3.4.6 a method to improve the initialization of the background model is presented.

3.4.2 Background Maintenance

For every new frame, the observed pixel value X_t at each pixel position is used in order to iteratively adapt its corresponding GMM to the described on-time distribution. Depending on the sorting strategy and the update equations, the reviewed methods converge to slightly different results.

The method in [Zivkovic, 2004] has the obvious advantage of adaptively selecting the number of components used for each pixel, turning in lower processing times at pixel positions where a lower number of modes is needed. Nevertheless, its matching mode updating strategy is based

on a learning rate which is computed as $\rho_{m,t} = \alpha/\omega_{m,t}$, with $m \in \{1 \ldots K\}$. Such a learning rate can be considered to be consistent with the updating equations derived in [Bilmes, 1998] for the case of fitting finite mixtures (Equation 3.4 to 3.6) with a 'winner-takes-all' updating strategy adapted to the recursive computation of the maximum likelihood in the case of an underlying non-stationary distribution. Nevertheless, by computing the learning rate in that manner, estimates of modes with low weights become very sensitive to noise and can even derive in singularities.

The method in [Lee, 2005] uses an adaptive learning rate calculated for each Gaussian which depends on the age of the Gaussian (instead of depending on the whole GMM). Such an updating rate is not only beneficial at system initialization, where the re-normalization of the weights would affect the learning rate applied to the first created modes, but also in extrinsically managed GMMs as will be introduced in Chapter 5, which make use of conditional updates and mode-substitutions. Nevertheless,the method presented in [Lee, 2005] does not have any means for selecting the number of needed Gaussians per pixel.

Therefore, the method presented in this section uses the same sorting strategy and weight updating (Equation 3.19) as in [Zivkovic, 2004]. Furthermore, the learning rate for the update of the Gaussian parameters of the matched mode is computed so as to follow sufficient statistics for the first observations and to gradually approach the basic recursive filter as first introduced in [Lee, 2005], i.e.:

$$\rho_{m,t} = \frac{1-\alpha}{\eta_{m,t}} + \alpha, \tag{3.21}$$

where $\eta_{m,t}$ is a variable used to count the number of observations assigned to each mode. $\eta_{m,t}$ is set to 1 when a mode is created and consecutively incremented when the parameters of the mode are updated. Therefore, the parameters of recently created modes are updated approximately as based on sufficient statistics ($\rho_{m,t} \approx 1/\eta_{m,t}$) while older modes forget older samples in an exponentially decaying manner ($\rho_{m,t} \approx \alpha$). The mean and the variance of the matching modes are updated as in Equation 3.10 and 3.11, respectively. After updating the parameters of each matching mode, the mode is checked for application of the splitting rule as defined in Section 3.4.4 if necessary.

If the current pixel value X_t does not fit in any of the available modes, a new mode is created. New modes represent observations that were not contained in the model. Therefore, they are created with a low prior weight, a mean equal to the value X_t of the observation and an initialization value for the variance $\sigma_{i,t}$, which is adaptively computed so as to fit to the dynamic of the scene as explained in the following section.

3.4.3 Dynamic Variance Control

At system initialization, $\sigma_{i,t}$ is set equal to $\hat{\sigma}_b$. In order to update the value of $\sigma_{i,t}$, the behavior of the system is observed from two different perspectives. On one hand, the absolute deviation of the observations belonging to background pixels $\mathscr{D}_b^{abs} := \{|\delta_{p,m,t}| : p \in \mathscr{P}_b\}$, being $\mathscr{P}_b$ the set of pixels belonging to the background, with respect to $\sigma_{i,t}$ is computed. Following the arguments leading to Equation 3.20, $\sigma_{i,t}$ should have a similar value to the median of $\mathscr{D}_b^{abs}$. But, since the deviations in $\mathscr{D}_b^{abs}$ are affected by the value of $\sigma_{i,t}$ at the initialization time of the individual modes, this similarity is conditioned on past values of $\sigma_{i,t}$. Therefore, on the other hand the absolute deviation of the observations belonging to foreground pixels $\mathscr{D}_f^{abs} := \{|\delta_{p,m,t}| : p \in \mathscr{P}_f\}$, being $\mathscr{P}_f$ the set of pixels belonging to the foreground, with respect to $\sigma_{i,t}$, is considered, which shows the current behavior of the system. In order to evaluate the behavior of the system from these two different perspectives, two indicators, v and $\hat{\sigma}_f$, are needed.

The first indicator, v, is a counter of the number of positions between the median of the absolute deviation of the background pixels $\mathscr{P}_b$ with respect to their corresponding matching modes m at time t, $\sigma_{p,m,t}$, and the median of $\{\sigma_{i,t}, \mathscr{D}_b^{abs}\}$. This value can be easily computed by setting $v = 0$ for every new frame and comparing for every updated background pixel $p \in \mathscr{P}_b$ the variance of the matched mode $\sigma_{p,m,t}$ with $\sigma_{i,t}$. Attending to these comparisons, v is updated as:

$$v = \begin{cases} v+1, & \text{if } \sigma_{p,m,t} > \sigma_{i,t}, \\ v-1, & \text{if } \sigma_{p,m,t} < \sigma_{i,t}. \end{cases} \tag{3.22}$$

The second indicator, $\hat{\sigma}_f$, is an approximation of the median absolute deviation of foreground modes $\mathscr{P}_f$. To obtain this value, for every new frame $\hat{\sigma}_f$ is set equal to $\sigma_{i,t}$ and, for every foreground mode, the variance of the mode σ_m is compared with $\hat{\sigma}_f$. Depending on the result of this comparison, $\hat{\sigma}_f$ is updated as:

$$\hat{\sigma}_f = \begin{cases} \hat{\sigma}_f + 0.1, & \text{if } \sigma_m > \hat{\sigma}_f, \\ \hat{\sigma}_f - 0.1, & \text{if } \sigma_m < \hat{\sigma}_f. \end{cases} \tag{3.23}$$

Equation 3.23 is a recursive approximation on the median of a series of values similar to the one proposed in [McFarlane and Schofield, 1995].

After processing a whole frame, v and $\hat{\sigma}_f$ are evaluated and $\sigma_{i,t}$ is updated accordingly. A negative value of v means that the median of the deviation of the background modes is lower than the initialization variance $\sigma_{i,t}$. Therefore, it is hypothesized that $\sigma_{i,t}$ is too high. Conversely, a positive value means that the median of the deviation of the updated

modes is higher than $\sigma_{i,t}$. In this case, it is hypothesized that $\sigma_{i,t}$ is too low. In order to verify this hypothesis, the value $\hat{\sigma}_f$ is used. If the median of the deviation of the foreground modes $\hat{\sigma}_f$ is lower than $\sigma_{i,t}$ it can be corroborated that $\sigma_{i,t}$ is too high, otherwise it can be corroborated that it is too low. By imposing the condition that both indicators v and $\hat{\sigma}_f$ agree, $\sigma_{i,t}$ is dynamically controlled so as to make it converging to the median of the deviation of the observations corresponding to background modes without being conditioned by their respective initialization settings.

If $\sigma_{i,t}$ is too high ($v < 0$ and $\hat{\sigma}_f < \sigma_{i,t}$), its value is updated as:

$$\sigma_{i,t+1} = \sigma_{i,t} + \left(\frac{\sigma_{i,t}}{\hat{\sigma}_f} - 1\right)\frac{v}{N}, \tag{3.24}$$

where N is the total number of pixels in a frame. That means, we decrease the value of $\sigma_{i,t}$ according to $\hat{\sigma}_f$ and v.

If $\sigma_{i,t}$ is too low ($v > 0$ and $\hat{\sigma}_f > \sigma_{i,t}$), its value is updated as:

$$\sigma_{i,t+1} = \sigma_{i,t} + \left(\frac{\hat{\sigma}_f}{\sigma_{i,t}} - 1\right)\frac{v}{N}\frac{c}{u}, \tag{3.25}$$

where c is the number of created modes and u the number of updates. That means, the value of $\sigma_{i,t}$ is updated according to $\hat{\sigma}_f$ and v. The factor c/u in (3.25) penalizes higher values of $\sigma_{i,t}$, i.e., as the number of foreground modes decreases and the number of background modes increases, $\sigma_{i,t}$ grows slower.

This process is repeated for every new frame. The value $\sigma_{i,t}$ is also used to set a selection criterion for the splitting rule as explained in the next section.

3.4.4 Splitting Over-Dominating Modes

The presented algorithm uses an on-line variation of the EM algorithm to fit a GMM to a non-stationary distribution and, the same as the EM algorithm, might suffer from the problem of getting caught in some boundary of the feature space. For the case of fitting a GMM to a stationary distribution, the Split and Merge Expectation Maximization (SMEM) algorithm [Ueda et al., 2000] was introduced in order to escape from local maxima. The intuition behind is that the Gaussian modes can be better distributed over the feature space by simultaneously splitting a Gaussian in an under-populated region while merging two Gaussians in an over-populated region. The split and merge operations are followed by a *partial EM procedure* and the *full EM procedure* and repeatedly performed until convergence.

The splitting rule proposed here finds its roots in the SMEM algorithm. Nevertheless, there are two important differences that hinder a straightforward transfer of the SMEM algorithm to

the background subtraction domain. First, the underlying distribution is non-stationary. And second, the number of modes used is limited, but not fixed. Moreover, the 'winner-takes-all' updating strategy and the matching mode selection scheme favor the update of dominating modes. Therefore, it can be considered that the merging operation is implicitly done in the variant of the EM used for background subtraction. Thus, only an appropriate splitting rule is needed.

To select candidate modes for the splitting operation the value $\sigma_{i,t}$ as calculated in the former section is used to set a variance controlling value σ_c as $\sigma_c = c\sigma_{i,t}$, with $c \geq 2$. Updated modes m with $\sigma_m > \sigma_c$ are selected for splitting into the m' and the m'' Gaussians. By setting $c > 2$ it is accounted for a certain variation of the variance of background pixels. For $c \to \infty$ the behavior of the system is the same as state-of-the-art GMMs with an adaptive setting of the initialization variance. Selected Gaussians m are splitted as follows:

$$\begin{aligned} \omega_{m',t} &= \omega_{m,t}, & \omega_{m'',t} &= \alpha, \\ \mu_{m',t} &= \mu_{m,t}, & \mu_{m''} &= X_t, \\ \sigma_{m',t} &= s\sigma_{m,t}, & \sigma_{m'',t} &= \sigma_{i,t}, \end{aligned} \tag{3.26}$$

where $s \leq 1$ is a factor used to reduce the variance of the mode being splitted.

That means, m' is used to represent the background and m'' to represent the foreground. Furthermore, it is assumed that the observed value X_t at the moment of splitting Gaussian m corresponds to a foreground pixel and that the mean value $\mu_{m,t}$ can still be considered as a good description of the background. Since the initial parameter values given to m' are often poor, its counter $\eta_{m',t}$ is set to a small value.

Figure 3.2 depicts graphically the process of splitting an over-dominating Gaussian mode. It can be appreciated that the resulting modes cover a lower volume of the feature space which accommodates a large range of values in the left and upper axis and a small range of values in the right axis.

The splitting operation introduces a bias towards small values of $\sigma_{i,t}$. In order compensate this bias, a lower bound on the variance of existing modes is also set, in order to not allow variance values lower than $\frac{\sigma_{i,t}}{c}$. The factor c is therefore considered as a spanning factor of the variance values over the initialization value $\sigma_{i,t}$ of the variance of new created modes.

3.4.5 Lighting Change Detection

After the classification of pixels as background or foreground, similarly to [Kaewtrakulpong and Bowden, 2001] foreground pixels are checked for illumination changes (shadow and highlight). In [Kaewtrakulpong and Bowden, 2001], the color model proposed in [Horprasert et al., 1999] is used, which basically consists of a so called expected chromaticity line passing through the

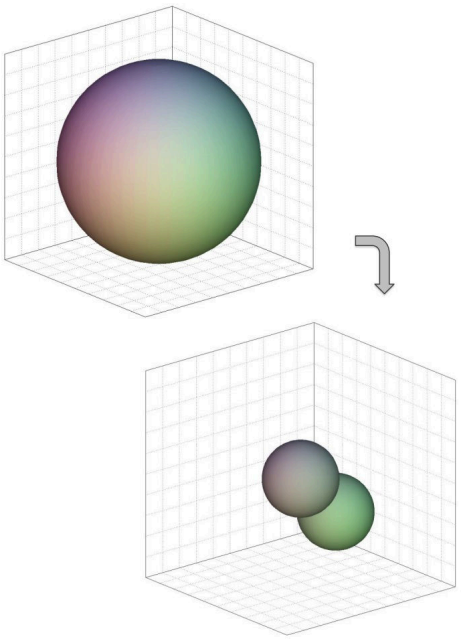


Figure 3.2: *Graphic depiction of the splitting operation.*

origin of the RGB color space and the expected background color. Foreground pixels are then compared against the current background components at their corresponding positions. RGB values falling into a cylinder with radius σ_b, where b is a considered background mode, along the expected chromaticity line and with a brightness distortion $l < \lambda < 1$, being $0 < l < 1$, are considered to be originated by shadows.

The color model used by the method presented in this section is the same, but, instead of building a cylinder along the expected chromaticity line, a cone is considered with its top vertex placed in the origin of the RGB space and a σ_b radius at the section with center in μ_b, where b is a considered background mode. RGB values falling into the cone and with a brightness distortion $l < \lambda < u$, being $0 < l < 1$ and $u > 1$, are considered to be originated by illumination changes. This method has two obvious advantages over the one proposed in [Kaewtrakulpong and Bowden, 2001]: first, by allowing a brightness distortion higher than 1, it is possible not only to detect shadow, but also highlight, and, second, by using a cone instead of a cylinder along the chromaticity line, darker colors (shadows) are forced to have a lower variance, therefore avoiding dramatic chromatic changes (as it can happen with the cylinder model). Figure 3.3 depicts graphically both methods.

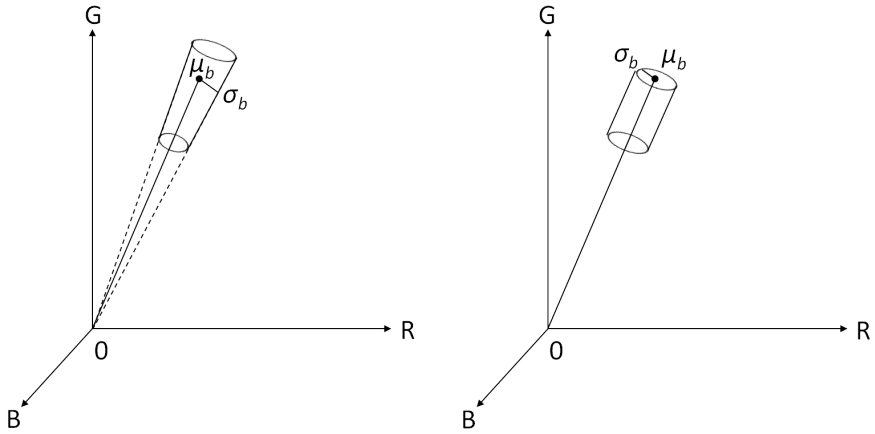


Figure 3.3: *Lighting models comparison. Left: The cone model (proposal). Right: The cylinder model as in [Horprasert et al., 1999].*

3.4.6 Improving Background Initialization

The method presented in this section has the advantage of adaptively choosing suitable values $\sigma_{i,t}$ for the initialization of the variance in new created modes. Nevertheless, as described in Section 3.4.1 and 3.4.2, the first value σ_b used to iteratively compute $\sigma_{i,t}$ is only based on the two first frames of the video sequence. Although in [Heras Evangelio et al., 2012] it is shown that the system converges to appropriate $\sigma_{i,t}$ values even in the case of a wrong initialization, which could be the fact in case of sudden camera movements or transmission problems in camera networks, it seems obvious that robustly estimating σ_b might be of importance for obtaining good segmentation results from the very beginning of the sequence. In this section, a method is presented for better estimating σ_b.

Based on the same assumptions and methodology as in Section 3.4.1, σ_b is set using the two first frames of the video sequence. Analogously, $\sigma_{b,t}$ is computed for every following consecutive two frames and checked for agreement with σ_b. $\sigma_{b,t}$ is used to update σ_b until convergence. Two variance initialization values σ_b and $\sigma_{b,t}$ are considered to be in agreement, if their absolute difference σ_{diff} is smaller than the minimum of them, $\min(\sigma_b, \sigma_{b,t})$. In this case, σ_b is updated to:

$$\sigma_b = 0.5\,\sigma_b + 0.5\,\sigma_{b,t}, \tag{3.27}$$

until convergence of σ_b. σ_b is considered to have converged to a proper value when:

$$\sigma_{diff} < \ln\left(\frac{\sigma_b}{2}\right), \tag{3.28}$$

where ln is the natural logarithm function. With this, a fast growing convergence criterion is set for low values of σ_b while the criterion for high σ_b values is set almost constant.

In the case of no agreement between σ_b and $\sigma_{b,t}$, it is assumed that the smaller of them is the right one. Therefore, σ_b is set to $\min(\sigma_b, \sigma_{b,t}) + 3$, where an offset of 3 has been added so as to avoid getting stuck at small values, which could have been produced by transmission problems in camera networks.

This process is repeated until convergence of σ_b or until the first half of a training period, which is scheduled in a similar fashion as the one described in [Kaewtrakulpong and Bowden, 2001], is finalized.

3.5 Evaluation

To assess the proposed system, in the next referred to as SGMM for brevity, the proposed technique for the estimation of $\sigma_{i,t}$ is first validated. Afterwards, the overall computational load has been measured. Finally, the segmentation results have been quantitatively evaluated and ranked against several state-of-the-art background subtraction methods.

3.5.1 Datasets

For the validation of the proposed technique for the estimation of $\sigma_{i,t}$, three video sequences exhibiting three different behaviors concerning the amount of foreground activity and lighting conditions have been used. The aim of these tests is to proof that the parameter $\sigma_{i,t}$ is able to follow the characteristics of the scene. The first sequence, *Lobby*, contains 70000 frames ($\approx$ 2 h.) recorded in the lobby of a crowded public building, which has both natural and artificial light. As it is getting darker outside, it is easy to appreciate how the camera noise raises. The second sequence, *Winter*, contains 65000 frames ($\approx$ 1 h. 50 min.) recorded in a sparsely crowded yard in winter. At the beginning of the scene it is snowing and, therefore, measurements are very noisy; at the end of the scene it stops snowing and, therefore, the noise shrinks. The third sequence, *Underground*, is a public sequence taken from the i-LIDS dataset supplied to AVSS 2007, containing 5223 frames ($\approx$ 3 min.). It contains a scene in an underground; the field of view is short and therefore the moving objects large. The noise is nearly constant during the whole scene.

To qualitatively evaluate the background subtraction results, the CDnet dataset [Goyette et al., 2012] has been used.

A thorough description of the datasets is provided in Appendix A, Description of Datasets.

3.5.2 Variance Controlling Scheme Validation

In order to evaluate the estimated $\sigma_{i,t}$ values, a ground-truth value σ_{GT} has been computed by taking the absolute deviation for consecutive values (X_t, X_{t+1}) of each pixel for each pair of consecutive frames and estimated the standard deviation of the modes representing the background and using Equation (3.20). In sparsely crowded environments, e. g. *Winter*, σ_{GT} approaches the variance of most of the pixels belonging to the background. In crowded environments, e.g. *Lobby*, σ_{GT} has a slightly higher value than most of the pixels belonging to the background. Therefore, the searched value $\sigma_{i,t}$ should be slightly higher or similar to σ_{GT}, depending on the kind of scene.

Figure 3.4 shows the results obtained for the three above mentioned sequences. The blue line, $\sigma_{i,t}$, shows the behavior of the algorithm as described in this chapter. The proposed system is able to correctly follow the dynamic of the scene and finds values near to σ_{GT}. The dashed cyan, $\sigma_{o,t}$, and green, $\sigma_{u,t}$, lines, show that the algorithm also converges to suitable values even in the hypothetic case of a wrong initialization (this case was forced, since the algorithm started well for the three sequences).

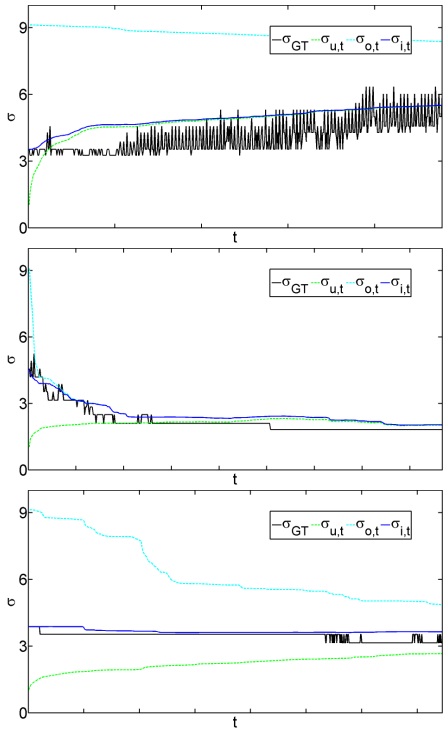


Figure 3.4: *Behavior of the proposed variance controlling scheme for the test video sequences* Lobby, Winter *and* Underground.

3.5.3 Computational Load

Table 3.1 shows the processing time for the above mentioned sequences (each frame containing $720 * 576$ RGB pixels) in a 3GHz PC without software optimization. For comparison, the processing time needed by the system in [Zivkovic, 2004], in the next AGMM, has also been measured. AGMM is able to automatically select the number of needed components per pixel in order to adapt to the observed scene, but does not have any means to initialize and control the variance parameter of the Gaussian modes.

Sequence	SGMM	AGMM
Lobby	43,96	37,52
Winter	34,15	33,65
Underground	34,84	35,45

Table 3.1: *Processing time in ms. of the three compared GMM systems.*

The processing times of both systems are very similar for sparsely crowded scenarios. In fact, the SGMM method converges to similar background models as AGMM in sequences of sparsely crowded scenarios, where over-dominating modes rarely appear. In crowded scenarios SGMM needs more processing time than AGMM. This is not a surprise, since AGMM often converges to models where over-dominating modes cover a wide range of the possible pixel values. Over-dominating modes usually occupy the first position of the modes list. Therefore, for most of the pixels matching an over-dominating mode, only one Gaussian distribution needs to be checked, turning into a lower computing demand. Particularly, in the case of the *Lobby* sequence, many of the GMMs obtained by the AGMM converged to unimodal mixtures, therefore not being able to properly segment foreground objects. Contrarily, the proposed system was able to correctly select and split over-dominating modes and thus provided useful segmentation results. To summarize, in comparison to the reference system, the method presented in this chapter achieved similar segmentation results at similar processing times in sparsely crowded environments, while achieving significantly better results in crowded scenarios at the price of a slightly higher processing time.

3.5.4 Qualitative Evaluation

The SGMM algorithm has been tested through the whole CDnet dataset and ranked against the algorithms provided as benchmark at the time of the workshop proposal: SOBS [Maddalena and Petrosino, 2008], ViBe [Barnich and Van Droogenbroeck, 2011], KDE [Elgammal et al., 2000], the seminal GMM formulation in [Stauffer and Grimson, 1999] (in the table referred to as GMM), a GMM with a two phases kind of learning and shadow detection as proposed in [Kaewtrakulpong and Bowden, 2001] (in the table, TPGMM-SD), a GMM with automatic

selection of number of components per pixel as proposed in [Zivkovic, 2004] (in the table, AGMM), Mahalanobis distance [Benezeth et al., 2010] (in the table, MD) and Euclidean distance [Benezeth et al., 2010](in the table, ED). For the computation of the performance metrics used for ranking, the provided toolkit was used. The results of the benchmark methods were taken from the website of the workshop.

The parameters chosen for the SGMM allow for a straightforward comparison of the provided results with those GMM-based approaches already evaluated with the CDnet dataset. Therefore, the learning factor α has been set to 0.001, as for the rest of already evaluated GMM-based approaches, and a maximum number of five Gaussians per pixel has been used. Furthermore, the sigma spanning factor c has been set to 3, and the brightness distortion $l < \lambda < u$ has been configured with $l = 0.85$ and $u = 1.10$ (a lower brightness distortion for highlight detection compensates the higher variance that is allowed to highlighted pixels). A 5x5 median filter has been applied in a post-processing step, as the organizing committee had done with the results provided by the methods proposed for the benchmark.

The sigma spanning factor c controls the splitting operation. A value of $c = 2$ means that an equal variance is expected for all the background modes. Therefore, those modes which variance is bigger than twice the estimated value for background modes, $\sigma_{i,t}$, should be described by two different modes. In practice, a slightly bigger value than two is recommended in order to accommodate for a certain uncertainty in the estimation of $\sigma_{i,t}$. Furthermore, by setting $c > 2$, a higher range of variance values for the background modes is allowed. This might be required in scenes depicting different kinds of background as, e.g., watter surfaces and still background areas. In the conducted experiments, values ranging from two to four provided a similar performance. For values larger than four the performance decays, because the splitting rule is rarely applied and some over-dominating modes appear. Obviously, setting this parameter is much less critical than setting the initialization of the variance parameter in the original GMM formulation. In fact, as $c \to \infty$, the proposed SGMM tends to behave as a standard GMM without splitting rule, but still with the advantage of adaptively setting the initialization value of the variance for new modes. For the shake of compactness, only the results selected for publication on the 'changedetection' website are presented here.

Table 3.2 shows the average results along the dataset, the ranking considering the average results, and the average ranking across the six different categories by the date of the evaluation of the proposed method (11.05.2012). The overall results provide an average of the seven used performance metrics (Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), F-measure and Precision) over the six categories. The average performance metrics of the evaluated methods are obtained by averaging the computed value for each of the corresponding metrics at each of the six categories. The average ranking corresponds to the average over the ranking obtained by each of the evaluated methods attending to each of the averaged metrics, therefore providing

Table 3.2: *Overall segmentation results and ranking of SGMM (11.05.2012).*

Method	Average ranking across categories	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SGMM	3,00	2,86	0.7074	0.9910	0.0090	0.0191	2.5299	0.7009	0.7813
SOBS	3,00	3,00	0.7854	0.9805	0.0195	0.0097	2.7049	0.7039	0.7040
TPGMM-SD	4,17	4,86	0.5075	0.9946	0.0054	0.0294	3.1296	0.5871	0.8182
KDE	4,17	5,29	0.7371	0.9749	0.0251	0.0147	3.5974	0.6607	0.6749
ViBe	4,33	5,00	0.6758	0.9825	0.0175	0.0182	3.2035	0.6599	0.7301
GMM	5.50	4.57	0.7070	0.9864	0.0136	0.0206	3.0962	0.6561	0.6987
AGMM	6,17	5,57	0.6942	0.9846	0.0154	0.0194	3.1498	0.6542	0.7045
MD	7,00	6,71	0.7584	0.9576	0.0424	0.0112	4.8771	0.6143	0.5904
ED	7,67	7,14	0.7020	0.9683	0.0317	0.0173	4.4509	0.6016	0.6110

an indicator of the performance of a giving method under a broad range of application scenarios. The average ranking across categories corresponds to the average of the ranking obtained by each of the methods at each individual category, giving therefore an indication of the performance of a method at each of the individual categories. A detailed description of the performance measures used and the ranking procedure is provided in Appendix B, Performance Metrics. The SGMM method outperformed not only the GMM methods already evaluated as benchmark, but also every other of the benchmark methods. The detailed results obtained for the individual categories have been provided to the organizers of the workshop and have been already made publicly available[2].

Two of the evaluated GMM-based methods (GMM and AGMM) use a maximum number of three Gaussians, while the other two (TPGMM-SD and SGMM) use five. It can be observed that the behavior of state-of-the-art GMM-based approaches is severely affected by the increase of the maximal number of Gaussians, leading to a trade-off between the recall and precision of the provided foreground masks. This is reflected by the unbalanced recall and precision values achieved by the TPGMM-SD approach, which provides the lowest recall and F-measure values of the whole set of evaluated approaches. On the contrary, the proposed method (SGMM) is able to achieve a balanced precision-recall behavior, which results in one of the highest F-measure values and the lowest percentage of wrong classification of the whole set of evaluated methods. This is a consequence of the variance controlling scheme introduced by the proposed system, which allows for the creation and maintenance of background modes with an adequate variance value of the observed scene.

[2] http://www.changedetection.net

Table 3.3: *Segmentation results and ranking of SGMM for the 'Baseline' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SOBS	1.7143	0.9193	0.9980	0.0020	0.0807	0.4332	0.9251	0.9313
KDE	2.8571	0.8969	0.9977	0.0023	0.1031	0.5499	0.9092	0.9223
ViBe	4	0.8204	0.9980	0.0020	0.1796	0.8869	0.8700	0.9288
MD	4.2857	0.8872	0.9963	0.0037	0.1128	0.7290	0.8954	0.9071
ED	5.4286	0.8385	0.9955	0.0045	0.1615	1.0260	0.8720	0.9114
TPGMM-SD	5.5714	0.5863	0.9987	0.0013	0.4137	1.9381	0.7119	0.9532
SGMM	6.2857	0.8680	0.9949	0.0051	0.1320	1.2436	0.8594	0.8584
AGMM	6.7143	0.8085	0.9972	0.0028	0.1915	1.3298	0.8382	0.8993
GMM	8.1429	0.8180	0.9948	0.0052	0.1820	1.5325	0.8245	0.8461

Tables 3.3 to 3.8 show the results provided by the proposed method for the individual categories of the CDnet dataset, respectively. In comparison with the other three GMM-based methods evaluated (GMM, TPGMM-SD, AGMM), the proposed method provides the best results in four of the six evaluated categories (Camera Jitter, Intermittent Object Motion, Shadow and Thermal) and the second best results in the other two (Baseline and Dynamic Background), in which the proposed method is outperformed by TPGMM-SD.

The better results of the TPGMM-SD method in the Baseline category are actually due to the unbalanced solution provided by the method, which yields to a kind of switch-ranking (alternating from the top to the bottom of the table) of the method according to the several performance measures. Therefore, these results are not relevant and will not be further discussed here.

The better performance of the TPGMM-SD method in the Dynamic Background category is due to a slightly too high value for the initialization of the variance parameter $\sigma_{i,t}$ of the proposed method for $t = 1$. Nevertheless, it has been observed that $\sigma_{i,t}$ decreases along the sequence, therefore increasing the performance of the proposed method.

The categories where the proposed method exhibits the greatest advantage with respect to the other GMM-based methods are Shadow and Thermal, where the optimal value for the initialization of the variance parameter in new modes is more different with respect to the rest of the sequences.

Table 3.4: *Segmentation results and ranking of SGMM for the 'Camera Jitter' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SOBS	2.1429	0.8007	0.9787	0.0213	0.1993	2.7479	0.7086	0.6399
SGMM	3	0.7088	0.9869	0.0131	0.2912	2.3761	0.7251	0.7752
TPGMM-SD	4.2857	0.5074	0.9888	0.0112	0.4926	3.0233	0.5761	0.6897
ViBe	4.4286	0.7112	0.9694	0.0306	0.2888	4.0150	0.5995	0.5289
GMM	4.5714	0.7334	0.9666	0.0334	0.2666	4.2269	0.5969	0.5126
KDE	5.4286	0.7375	0.9562	0.0438	0.2625	5.1349	0.5720	0.4862
AGMM	6.7143	0.6900	0.9665	0.0335	0.3100	4.4057	0.5670	0.4872
MD	7	0.7356	0.9431	0.0569	0.2644	6.4390	0.4960	0.3813
ED	7.4286	0.7115	0.9456	0.0544	0.2885	6.2957	0.4874	0.3753

Table 3.5: *Segmentation results and ranking of SGMM for the 'Dynamic Background' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
TPGMM-SD	3.2857	0.6303	0.9983	0.0017	0.3697	0.5405	0.6697	0.7700
SGMM	3.5714	0.7715	0.9933	0.0067	0.2285	0.9132	0.6380	0.6665
AGMM	3.5714	0.8019	0.9903	0.0097	0.1981	1.1725	0.6328	0.6213
GMM	3.7143	0.8344	0.9896	0.0104	0.1656	1.2083	0.6330	0.5989
SOBS	4.1429	0.8798	0.9843	0.0157	0.1202	1.6367	0.6439	0.5856
KDE	5.8571	0.8012	0.9856	0.0144	0.1988	1.6393	0.5961	0.5732
ViBe	6.1429	0.7222	0.9896	0.0104	0.2778	1.2796	0.5652	0.5346
MD	7	0.8132	0.9698	0.0302	0.1868	3.1407	0.5261	0.4517
ED	7.7143	0.7757	0.9714	0.0286	0.2243	3.0095	0.5081	0.4487

Table 3.6: *Segmentation results and ranking of SGMM for the 'Intermittent Object Motion' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SGMM	3.4286	0.5013	0.9853	0.0147	0.4987	4.9180	0.5397	0.6993
GMM	3.5714	0.5142	0.9835	0.0165	0.4858	5.1955	0.5207	0.6688
AGMM	3.8571	0.5467	0.9712	0.0288	0.4533	5.4986	0.5325	0.6458
SOBS	4	0.7057	0.9507	0.0493	0.2943	6.1324	0.5628	0.5531
TPGMM-SD	5	0.3476	0.9892	0.0108	0.6524	5.9854	0.3903	0.6953
ViBe	5.2857	0.5122	0.9527	0.0473	0.4878	7.7432	0.5074	0.6515
ED	5.8571	0.5919	0.9336	0.0664	0.4081	8.9975	0.4892	0.4995
MD	6.2857	0.7165	0.8886	0.1114	0.2835	11.5341	0.4968	0.4535
KDE	7.7143	0.5035	0.9309	0.0691	0.4965	10.0695	0.4088	0.4609

Table 3.7: *Segmentation results and ranking of SGMM for the 'Shadow' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SGMM	2.5714	0.8580	0.9889	0.0111	0.1420	1.7965	0.7944	0.7617
KDE	2.7143	0.8541	0.9885	0.0115	0.1459	1.6844	0.8030	0.7660
ViBe	3.1429	0.7838	0.9919	0.0081	0.2162	1.6497	0.8035	0.8342
TPGMM-SD	4.8571	0.6326	0.9936	0.0064	0.3674	2.2966	0.7179	0.8577
SOBS	5.2857	0.8355	0.9836	0.0164	0.1645	2.3318	0.7717	0.7219
GMM	5.5714	0.7960	0.9871	0.0129	0.2040	2.1951	0.7370	0.7156
AGMM	5.8571	0.7774	0.9878	0.0122	0.2226	2.1908	0.7322	0.7232
ED	6.8571	0.8006	0.9783	0.0217	0.1994	2.8949	0.6786	0.6112
MD	8.1429	0.7845	0.9708	0.0292	0.2155	3.7896	0.6348	0.5685

Table 3.8: *Segmentation results and ranking of SGMM for the 'Thermal' category (11.05.2012).*

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
KDE	2.5714	0.6725	0.9955	0.0045	0.3275	1.6795	0.7423	0.8974
SOBS	3.5714	0.5888	0.9956	0.0044	0.4112	2.0983	0.6834	0.8754
ViBe	4	0.5435	0.9962	0.0038	0.4565	3.1271	0.6647	0.9363
SGMM	4.8571	0.5363	0.9970	0.0030	0.4637	3.9394	0.6481	0.9263
MD	5.1429	0.6270	0.9906	0.0094	0.3730	2.3462	0.7065	0.8617
TPGMM-SD	5.5714	0.3395	0.9993	7.4348e-004	0.6605	4.8419	0.4767	0.9709
GMM	5.7143	0.5691	0.9946	0.0054	0.4309	4.2642	0.6621	0.8652
AGMM	6.4286	0.5542	0.9942	0.0058	0.4458	4.3002	0.6548	0.8706
ED	7.1429	0.5111	0.9907	0.0093	0.4889	3.8516	0.6313	0.8877

3.6 Conclusions

In this chapter a novel method for the detection of foreground pixels by means of background subtraction has been presented. The method is based in a per-pixel Gaussian Mixture Model, which is updated by means of an on-line variation of the Expectation Maximization algorithm for the case of fitting an underlying non-stationary multi-modal distribution.

The presented method incorporates some of the proposed improvements presented individually in recent publications [Kaewtrakulpong and Bowden, 2001; Zivkovic, 2004; Lee, 2005] to the original formulation in [Stauffer and Grimson, 1999]. Furthermore, a variance control heuristic has been presented, which is based on a dynamic estimation of a proper value for the initialization of new modes and a splitting operation of over-dominating modes. This allows for an unrestrained applicability of the method to a wide range of environments without the need of parameter tweaking (the learning rate should be set according to the video frame rate, not to the specific environment). Furthermore, the proposed system scales better than state-of-the-art GMM-based approaches when using a higher number of Gaussians, leading to more accurate models of the scene background.

A thorough quantitative evaluation of the segmentation results has been provided, showing a general improvement over state-of-the-art GMM-based background subtraction approaches, which is reflected in an increase of the recall and precision measures of the provided foreground masks.

4

Dual Background Models

4.1 Introduction

A statistical background model as defined in Chapter 3 provides a description of the static scene. In order to adapt to changes in the observed scene, statistical background models are regularly updated. Nevertheless, in their standard formulation, GMMs do not provide any means to differentiate between the several kinds of changes (illumination, introduction or removal of static objects, etc.), which can be introduced in the static scene. Therefore, the adaptation is handled in a unique manner regardless of the nature of the change, namely incorporating all changes into the background model at a pace regulated by the learning rate. This imposes a limitation in the achievable results that usually leads to the choice of a compromise value for the learning rate which allows to correctly follow illumination changes while maintaining in the foreground slow moving and new static objects as long as possible.

In this chapter, a system is presented which handles the problem posed by new static objects by using two background models learning at different rates and a Finite State Machine (FSM). New and removed static objects are incorporated into the background models at different points in time, depending on their respective learning rate. The FSM is used to reason on pixel classification attending to the results provided by the background models and to the history of the pixel.

The main purpose of the proposed system is the detection of new static objects, which is a relevant topic in many security applications. This topic is introduced in Section 4.2, which provides a brief overview of the main techniques employed in state-of-the-art approaches and motivates the proposed system. Section 4.3 expounds how dual background models can be

used in order to detect new static objects. The limitations inherent to dual background models are addressed by the FSM introduced in Section 4.4. The results of the proposed system are presented in Section 4.5. Section 4.6 concludes this chapter.

The content of this chapter has been partially published in *'Detection of Static Objects for the Task of Video Surveillance'*, in the Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV), 2011 [Heras Evangelio et al., 2011], and in *'Static Object Detection Based on a Dual Background Model and a Finite-State Machine'*, in the EURASIP Journal on Image and Video Processing, 2011 [Heras Evangelio and Sikora, 2011b].

4.2 Static Objects Detection

Detecting static objects in video sequences has several applications in surveillance systems such as the detection of illegally parked vehicles in traffic monitoring or the detection of abandoned objects in public safety systems and has attracted the attention of a vast research in the field of video surveillance.

Most of the proposed techniques aiming to detect static objects are based on the detection of change, which is usually achieved by means of background subtraction, and an additional approach used to handle the fact that static objects get incorporated into the background model of the scene when the model is updated. Commonly, this additional approach relies on tracking information [Guler et al., 2007; Venetianer et al., 2007; Singh et al., 2009; Bayona et al., 2009]. Nevertheless, these methods can find difficulties in real-life scenes involving crowds due to the large amounts of occlusions and to the shadows casted by moving objects, which turn the object initialization and tracking into a hard problem to solve. Many of the applications where the detection of abandoned objects can be of interest, like safety in public environments (airports, railway stations, etc.), impose the requirement of coping with crowds.

In order to address the limitations exhibited by tracking-based approaches, the use of dual foregrounds is proposed in [Porikli et al., 2008]. The system is based on two background models with different learning rates, a short-term and a long-term background model, which, consequently, incorporate new observations into the background at different speeds. Groups of pixels classified as background by the short-term but not by the long-term background model are then classified as static objects.

A drawback of this system is that temporarily static objects may also become absorbed by the long-term background model after a given time, which depends on its learning rate. This leads the system to not detect those static objects anymore, since both background models detect them as part of the background. Moreover, these absorbed static objects give raise to new detections when they are removed from the scene, therefore originating false positives. Furthermore, in the case that the time defined for raising new static object alarms is longer

than the time needed for incorporating new distributions into the long-term background model, a plain dual background based system will fail. To tackle this situation, a lower learning rate can be set for the long-term background model; nevertheless, this has the disadvantage, that the adaptation capability of the background model will be weaken, therefore, affecting to the background subtraction results.

To solve these problems, the system presented in this chapter uses the results obtained from a dual background subtraction to classify the pixels according to a finite-state machine. The finite-state machine is used to explain the results provided by background subtraction based on the sequence of states that a given pixel is gone through in the steps before. Thus, the system is able to differentiate between background and absorbed static objects. Furthermore, by adequately designing the states and transitions of the finite-state machine, the system can be used either in a full-automatic or in an interactive manner, making it extremely suitable for real-life surveillance applications.

4.3 Dual Background Models

Dual background models can be used to describe a given scene attending to different time courses, therefore providing different pixel classifications, which can be exploited either to improve the quality of the provided foreground masks or to classify pixels according to different temporal scales.

In [Elgammal et al., 2000], dual background models are exploited to provide foreground masks exhibiting a sensitive detection and low false positive rates. The short-term background model is used in order to quickly adapt to illumination changes in the scene. The long-term background model is used to provide a more stable representation of the scene background. The generated foreground masks result from the intersection of the masks provided by the short-term and the long-term background models, therefore eliminating false positives of the long-term foreground mask. These resulting foreground masks are also used in order to selectively update the short-term background model. Therefore, stationary or slowly moving objects are incorporated in both background models as soon as they become relevant enough in the set of samples used to generate the long-term background model.

In [Porikli et al., 2008], dual background models are used to detect new static objects. The short-term and long-term foreground masks are used in order to postulate hypotheses on pixel classification as shown in Table 4.1, where $F_L(X_t)$ and $F_S(X_t)$ denote the value of the long-term and short-term foreground mask at pixel X_t, respectively (this notation is used in the rest of this chapter). The limitation of this approach is imposed by the learning rate of the long-term background (see Section 4.2).

Table 4.1: *Hypotheses on pixel classification based on the long-term and short-term foreground masks as in [Porikli et al., 2008].*

$F_L(X_t)$	$F_S(X_t)$	Hypothesis
1	1	Moving object
1	0	Candidate abandoned object
0	1	Uncovered background
0	0	Scene background

In the system presented in this chapter, two improved GMM as described in Chapter 3 initialized with identical parameters except for the learning rate, a short-term background model B_S and a long-term background model B_L, are used as the underlying background subtraction approach. Actually, any multi-modal background model (see [Zivkovic and van der Heijden, 2006] and [Porikli and Tuzel, 2005], for example) that does not update the modes of the distributions corresponding to the background when a foreground object hides them, can be used. The foreground masks provided by background subtraction are used as input for the FSM introduced in the following section. Thereby, it is assumed that a reasonably good model of the empty scene can be achieved in a training period at system start (this requirement should also be imposed to the system in [Porikli et al., 2008] in order to reduce the number of false alarms at system start).

4.4 Multi-class Pixel Classification

In order to illustrate the limitation imposed by the learning rate of the long-term background for the detection of static objects by means of dual background models (see Section 4.2), a pixel X_t classified as background at time t is considered. Furthermore, it is assumed that this same pixel X_{t+1} at the next time step $t+1$ is occluded by a foreground object. Therefore, the value of both foreground masks $F_S(X_{t+1})$ and $F_L(X_{t+1})$ at $t+1$ will be 1. If the foreground object stays static, it will be learned by the short-term background model at first (assume at $t+\alpha$, $F_S(X_{t+\alpha}) = 0$ and $F_L(X_{t+\alpha}) = 1$) and afterwards by the long-term background (assume at $t+\beta$ $F_S(X_{t+\beta}) = 0$ and $F_L(X_{t+\beta}) = 0$). This process can be graphically described as shown in Figure 4.1.

By further observing the behavior of the background model of this pixel in time, it is possible to transfer the meaning of obtaining a given result from a dual background subtraction after a given history into multi-class pixel classification hypotheses and establish which further hypotheses can be postulated at the subsequent time steps. This knowledge can be used to define a FSM [Gibson, 1999], which can be used to hypothesize on pixel classification.

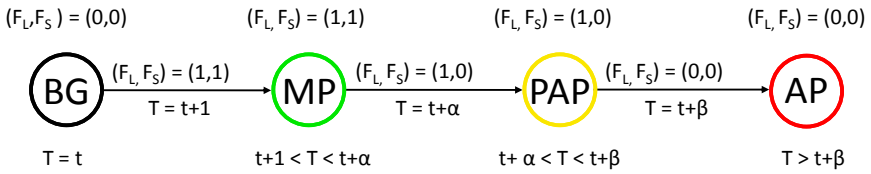


Figure 4.1: *Graphical description of the states a pixel goes through when being incorporated into the background model. BG indicates a pixel that belongs to the background model, MP a pixel that belongs to a moving object, PAP a partially absorbed pixel and AP an absorbed pixel.*

In the following subsection (4.4.1) the proposed FSM is introduced. Subsection 4.4.2 presents how this FSM can be further enhanced in terms of robustness and efficiency. Subsection 4.4.3 outlines how the multi-class pixel classification provided by the FSM can be used by higher layers in a computer vision system. By the introduction of the proposed FSM, Subsection 4.4.1, it is shown that some states require additional information in order to determine what is the next state for a given input. This is the case of some states in which it is necessary to know if any of the background models gives a description of the empty scene and, in affirmative case, which of them. Therefore, a copy of the last background value observed at every pixel position is kept. This value is used e.g. to distinguish when an absorbed static object (in the following, a long-term static object) is being removed or when it is being occluded by another object. In this sense, the FSM presented in the following can be considered as an Extended Finite State Machine (EFSM), which is a FSM extended with input and output parameters, context variables, operations and predicates defined over context variables and input parameters [Petrenko et al., 1999]. An EFSM can be viewed as a compressed notation of a FSM, since it is possible to unfold it into a pure FSM, assuming that all the domains are finite [Petrenko et al., 1999], which is the case in the state-machine presented here. In fact, context variables are used in a very limited number of transitions. Therefore, for clarity in the presentation, the proposed state-machine is first introduced as a plain FSM and special remarks are issued where the EFSM features are exploited.

4.4.1 A finite-state machine for hypothesizing on the pixel classification

A finite-state machine describes the dynamic behavior of a discrete system as a set of input symbols, a set of possible states, transitions between those states, which are originated by the inputs, a set of output symbols and sometimes actions that must be performed when entering, leaving or staying in a given state. A given state is determined by past states and inputs of the system. Thus, a FSM can be considered to record information about the past of the system it describes. Therefore, by defining a state machine whose states are the hypotheses on the pixels

and whose inputs are the values obtained from dual background subtraction, information about the pixel history can be recorded, and, thus, hypothesize on the classification of a pixel given a new dual background subtraction result, depending on the state where it was before.

A FSM can be defined as a 5-tuple $(I, Q, Z, \delta, \omega)$ [Booth, 1967], where:

- I is the input alphabet (a finite set of input symbols).
- Q is a finite set of states.
- Z is the output alphabet (a finite set of output symbols).
- δ is the next-state function, a mapping of $I \times Q$ into Q.
- ω is the output function, a mapping of $I \times Q$ onto Z.

The proposed FSM is defined as follows:

- I is the set of possible combinations of the results obtained from background subtraction. By defining the pair (F_L, F_S), the input alphabet reduces to $I \equiv \{(0,0),(0,1),(1,0),(1,1)\}$.
- Q is the set of states a pixel can go through as described below.
- Z is either a set of numbers indicating the hypotheses on the pixel classification $Z \equiv \{0,1,\ldots|Q|-1\}$, with $|Q|$ being the cardinality of Q, or a Boolean output $Z \equiv \{0,1\}$ with the value 0 for pixels not belonging to a static object and 1 for pixels belonging to a static object. Choosing the output alphabet depends on whether the hypotheses of the machine are to be further interpreted or not.
- δ is a next-state function as depicted in Figure 4.2.
- ω is the output function. This can be either a multivalued function with output values $z \in \{0,1,\ldots|Q|-1\}$ corresponding to the state of a pixel at a given time, or a Boolean function with output 0 for pixels not belonging to a static object and 1 for pixels belonging to a static object.

Additionally, a copy of the last background value observed at every pixel position is kept.

In the following, the states of the state machine, their hypothetical meaning, the condition that must be met to enter them or to stay in them and a brief description of their meaning are listed:

- 0 (BG), *background*, $(F_L, F_S) = (0,0)$.
  The pixel belongs to the scene background.

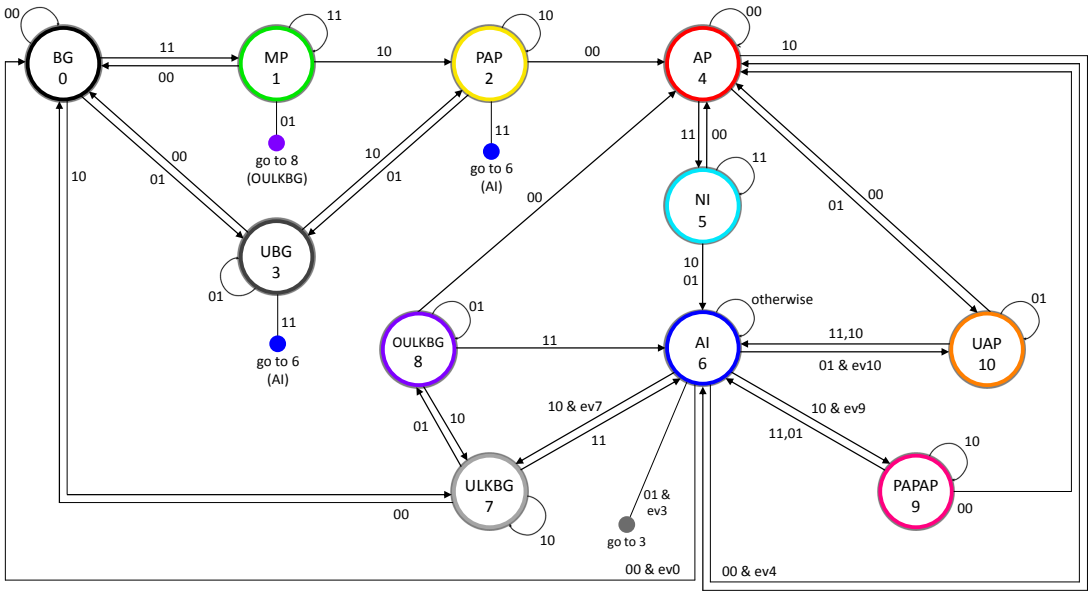


Figure 4.2: *Next-state function of the proposed FSM.*

- 1 (MP), *moving pixel*, $(F_L, F_S) = (1,1)$.
  The pixel belongs to a moving object. This state can be reached as well by pixels belonging to the background scene being affected by spurious noise not characterized by the background model.

- 2 (PAP) *partially absorbed pixel*, $(F_L, F_S) = (1,0)$.
  The pixel belongs to an object that has already been absorbed by B_S but not by B_L. In the following, these objects are called short-term static objects.

- 3 (UBG), *uncovered background*, $(F_L, F_S) = (0,1)$.
  The pixel belongs to a background region that was occluded by a short-term static object.

- 4 (AP), *absorbed pixel*, $(F_L, F_S) = (0,0)$.
  The pixel belongs to an object that has already been absorbed by B_S and B_L, i.e., a long-term static objects.

- 5 (NI), *new indetermination*, $(F_L, F_S) = (1,1)$.
  The pixel cannot be classified as background neither by B_S nor by B_L. It is not possible to ascertain if the pixel corresponds to a moving object occluding a long-term static object or if a long-term static object has been removed. The pixel is not classified as moving or static at this moment. If the pixel belongs to a moving object occluding a

long-term static object, the state machine will jump back to AP when the moving object moves out. If not, a new appearance will be learned by B_S and the state machine will jump to AI, where a decision will be taken.

- 6 (AI), *absorbed indetermination*, $(F_L, F_S) = (1,0)$.
  The pixel is classified as background by B_S but not by B_L. Given the history of the pixel it is not possible to ascertain if any of the background models gives a description of the actual scene background. To solve this uncertainty, the current pixel value is compared to the last known background value at this pixel position. A discussion follows below on how to obtain and update the last known background value.

- 7 (ULKBG), *uncovered last known background*, $(F_L, F_S) = (1,0)$.
  The pixel is classified as background by B_S but not by B_L and identified as belonging to the scene background.

- 8 (OULKBG), *occluded uncovered last known background*, $(F_L, F_S) = (0,1)$.
  The pixel is classified as background by B_L but not by B_S, and B_S is known to contain a representation of the scene background. This state can be reached when a long-term static object has been removed, the actual scene background has been learned again by B_S and an object whose appearance is very similar to the removed long-term static object occludes the background.

- 9 (PAPAP), *partially absorbed pixel over absorbed pixel*, $(F_L, F_S) = (1,0)$.
  The pixel is classified as background by B_S but not by B_L and could not be identified as belonging to the scene background. Therefore, it is classified as a pixel belonging to a short-term static object occluding a long-term static object.

- 10 (UAP), *uncovered absorbed pixel*, $(F_L, F_S) = (0,1)$.
  The pixel is classified as background by B_L but not by B_S, and B_L could not be interpreted to contain a representation of the actual scene background. This state can be reached when a short-term static object was occluding a long-term static object and the short-term static object gets removed.

In order to determine the transitions from state 6 additional information is needed. This is due to the fact that it is not possible to ascertain if any of the background models gives a good description of the empty scene. To illustrate this, two different cases are considered: a long-term static object being removed, and a long-term static object being occluded by a short-term static object. In both cases, when the long-term static object is visible B_S and B_L classify it as background (state 4, $(F_L, F_S) = (0,0)$). Afterwards, when the long-term static object is removed or occluded, a new color is observed. The new color persists at this pixel position and it gets first learned by B_S (state 6, $(F_L, F_S) = (1,0)$), causing an uncertainty, since it is not possible to distinguish if the new color corresponds to the scene background or to a short-term static object occluding the long-term static object. To solve this uncertainty, the

current pixel value is compared with the last known background value at this pixel position. In this state the FSM is actually behaving as an EFSM and the copy of the last background value observed at this position is a context variable. This is the unique state where the FSM explicitly makes use of extended features.

The last known background value is initialized for each pixel after the initialization phase of the background models, which is performed by updating the background models so as to follow sufficient statistics (see equations 3.13 and 3.14, Chapter 3). This value is subsequently updated for every pixel position when a transition from BG (state 1) is triggered as follows:

$$\begin{cases} if\ (F_L, F_S) = (0,1), & b_{LK}(X) = B_L(X), \\ otherwise, & b_{LK}(X) = B_S(X), \end{cases} \tag{4.1}$$

where b_{LK} denotes the last-known background value.

The output function of the FSM can have two forms:

- A Boolean function with output 0 for non-static pixels and 1 for static pixels. In this case, it has to be decided which subset O of Z designates a static pixel. There are many possibilities, depending on the desired responsiveness of the system. The lower and higher levels of responsiveness are achieved by $O \equiv \{4\}$ and $O \equiv \{2,4,5,6,8,9,10\}$, respectively.

- A multivalued function with output values $q \in \{0,1,\ldots|Q|\}$ corresponding to the state where the pixel is at a given time.

A Boolean function mapping a subset of Z can be used in order to classify groups of pixels belonging to the considered classes as static objects, while the results obtained by using a multivalued function can be used to feed up a higher analysis layer which groups pixels by means of their corresponding classes and builds objects.

4.4.2 Robustness and efficiency issues

The FSM introduced in Subsection 4.4.1 provides a reliable tool to hypothesize on the meaning of the results obtained from a dual background subtraction. However, there are some state-input sequences where an additional computation must be done in order to decide on the next state. This is the case of the state AI (6). A state-input sequence entering the state AI is AP-(1,1) → NI-(1,0) → AI, which corresponds to a pixel of a long-term static object being removed or getting occluded by a short-term static object. In this situations, it is necessary to disambiguate the results obtained from background subtraction.

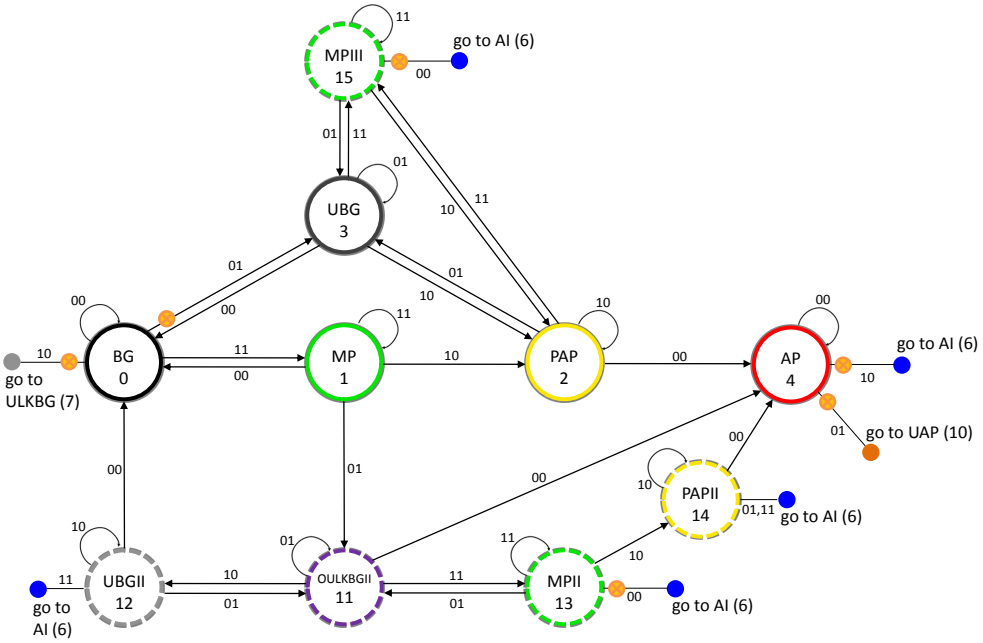


Figure 4.3: *Enhancements for the proposed FSM. Five additional states and six additional conditions on transitions to enhance the robustness and the efficiency of the FSM shown in Figure 4.2.*

There are three more state-input sequences entering the state AI, where this extra computation can be eventually avoided. These are MP-(0,1) → OULKBG-(1,1) → AI, UBG-(1,1) → AI and PAP-(1,1) → AI. In fact, these sequences enter the state AI because they can derive in a state-input where a disambiguation is necessary, given the pixel history. Therefore, defining known sequences which start at the first state-input pair of the three sequences mentioned above, reaching AI can be avoided for these known sequences. In order to do that, five more states need to be added to the FSM:

- 11 (OULKBGII), *occluded uncovered last known background ii*, $(F_L, F_S) = (0,1)$.
- 12 (UBGII), *uncovered background ii*, $(F_L, F_S) = (1,0)$.
- 13 (MPII), *moving pixel ii*, $(F_L, F_S) = (1,1)$.
- 14 (PAPII), *partially absorbed pixel ii*, $(F_L, F_S) = (1,0)$.
- 15 (MPIII), *moving pixel iii*, $(F_L, F_S) = (1,1)$.

These states are specializations of the states they inherit their name from and have the sense of avoiding to enter the state AI in these situations where the meaning of the state-input

Figure 4.4: *Multi-class pixel classification. From left to right: Frame number 2637 of the i-LIDS (AVSS 2007) AB-Easy sequence, detail of pixel classification using the FSM depicted in Figure 4.2, and using specialized states (Figure 4.3). Pixels are colored attending to their classification.*

sequence is non-ambiguous. Therefore, theses sequences are called known sequences. Their meaning can be inferred out of the transitions shown in Figure 4.3 and of the state that they specialize. In this fashion some additional specialized states can be defined.

Figure 4.3 also shows six additional conditions on six transitions marked as an orange point on the respective transition arrows. The reason for these conditions is that the tuple $(F_L(X_t), F_S(X_t))$ at time step t does not really make sense if being in the state where the transitions are conditioned. Thus, it is considered as noise and the classification state remains equal. If the tuple $(F_L(X_{t+1}), F_S(X_{t+1}))$ in the next time step $t+1$ is equal to $(F_L(X_t), F_S(X_t))$, then, the conditioned transition is done. In practice, these additional conditions are implemented as replica states with identical transitions as the state being replicated except for the transition being conditioned, which is only done in the corresponding replica state. These replica states are not depicted in the next-state function graph for clarity.

Introducing this additional states enhances the robustness of the state machine, since there are less input sequences deriving in state AI. Thus, there are less pixels that have to be checked with an eventually old version of the scene background (the last known background value is updated when leaving the state BG). Furthermore, because of avoiding this additional computation, a gain in efficiency is achieved. Replica states also contribute to enhance the performance of the system, since they filter out noisy inputs.

Figure 4.4 provides an example of the proposed multi-class pixel classification. Pixels classified other than BG have been colored according to the colors assigned to the states in figures 4.2 and 4.3.

4.4.3 Grouping pixels into regions

The state of each pixel at a given time t provides a rich information that can be further processed to refine the hypotheses. Pixels can be spatially associated depending on their states and their connectivity. In order to detect new static objects, pixels in the states 4 and 5 and those that have been in the states 2 or 9 for more than a given time threshold T can be considered as potentially belonging to the same object. Therefore, pixels fulfilling one of those criteria are taken into account to build groups of pixels by means of a connected components algorithm. Groups of pixels bigger than a fixed size are then classified as static objects.

In order to find the connected components, the algorithm proposed in [Chang et al., 2004], which provides linear time performance with respect to the size of the image, is used. This is achieved by using a contour tracing technique to detect the external contour and possible internal contours of each blob, and using the labels assigned to the external contours to identify and label the interior area of each blob. Therefore, the labeling of interior blob areas can be done in a single pass, while contour points are revisited a maximum number of four times, which is the maximal number of contours a pixel can lie on. Since the number of times a pixel can be visited is limited to four times and the processing of each pixel takes a constant amount of time, the complexity of the algorithm is linear.

The algorithm proposed in [Chang et al., 2004] has the advantage of providing one of the lowest processing time among the state-of-the-art connected components labeling techniques, specially in the case of images with a small number of labels, as is the expected case in the domain of video surveillance applications. Furthermore, as no re-labeling is needed, it does not impose additional memory requirements. A good review on state-of-the-art connected components labeling techniques can be found in [Grana et al., 2009].

4.4.4 Embedding user knowledge

As shown in Figure 4.2, the condition for the FSM to stay at the state AP at a given pixel X_t is that $(F_L(X_t), F_S(X_t)) = (0,0)$, which is the same condition as the one to stay at BG. That means, if a piece of the scene background was wrongly classified as static object, an operator could interactively correct this mistake with no need for the system to correct any of the background models, since those are updated regularly in a blind fashion. This could happen, for example, if a static object has occluded the background a long time and the lighting conditions have changed during this period. In that case, the uncovered scene background might not be similar to the last known background when the object is removed, therefore giving raise to the detection of a new static object. Since the conditions for staying at AP and BG are the same, such a situation can be easily corrected, with no need for modifying the background models, by incorporating user interaction. This is a huge advantage of the proposed system in comparison to other systems based on selectively updating the background model, since

deadlock situations caused by wrong update decisions are avoided. In the proposed system, the background model remains as a pure statistical information.

The same applies for static objects that an operator can consider non-interesting, which is a common issue in public spaces, where waste containers and other static objects are moved in the scene but do not represent a dangerous situation. Static object detection approaches based on selective updating of the background model do not offer a principled way of incorporating such items into the background model. Since the background models of the proposed system are updated in a blind fashion, these objects do get incorporated into the background models. Only the state of the FSM has to be changed. This kind of interaction can be defined as well with other layers in a complex computer vision system.

4.5 Evaluation

This section presents some experimental results. The proposed system is compared with a dual background based system that does not use a FSM (pixels are classified by using the hypotheses shown in Table 4.1 and an evidence value in a similar way as proposed in [Porikli et al., 2008]), which is used as reference system. To abbreviate, those systems are called DBG-FSM and DBG-T, respectively.

4.5.1 Datasets

The results have been obtained by using three public datasets: i-LIDS, PETS2006 and CAVIAR. The sequences AB-Easy, AB-Medium and AB-Hard from the i-LIDS dataset show a scene in an underground station. In PETS2006, there are several scenes from different camera views of a public railway station; scene S1-T1-C-3 has been used. CAVIAR covers many different actions of interest in a typical surveillance applications (people fighting, people/groups meeting, walking together and splitting up, or people leaving bags behind...), taken from a high camera view; from this dataset the scene LeftBag has been used. A thorough description of the datasets used throughout this work is provided in Appendix A, Description of Datasets.

The scenes from i-LIDS are the most relevant for the studied problem, since they show one of the challenges tackled by the proposed system, namely static objects remaining for a long time in the scene and then being removed. However, the scenes AB-Medium and AB-Hard present the handicap that the static scene cannot be learned before the static objects come into the scene, which is a requirement for both systems (DBG-T and DBG-FSM); therefore, 10 frames showing the empty scene were added at the beginning of each sequence, respectively, in order to train the background models. In PETS2006 static objects are not removed and thus, even if the static objects have to be detected, they do not pose the problem of detecting when a static object has been removed. In the CAVIAR scene LeftBag, static objects are removed

that early, that every background model can be tuned to not absorb them without risking the responsiveness of the background model. Thus, these two last sets of scenes cannot be considered as very challenging for the task of static objects detection. Nevertheless, the three datasets have been considered for the experimental evaluation, since they are the most commonly used in the computer vision community for the presentation of systems for the detection of static objects.

4.5.2 System Configuration

The underlying background models used in both systems are Gaussian mixture models. The background models have been set up with identical parameters except for the learning rate in each dual background model configuration. The learning rate of the short-term model B_S is 10 times higher than the learning rate of the long-term model B_L. A relatively large value for α_L has been consciously chosen in order to force B_L to learn the static objects in the scene and thus being able to prove the correct operation of the proposed system both when static objects are learned by B_L and when they get removed from the scene. It is important thus to remark, that the goal of the experiments presented here is to evidence what problems double background based systems face on the detection of long-term static objects and how the proposed approach solves them. Therefore, objects are classified as static very fast. In practice, α_L can be drastically reduced. The rest of the parameters are as follows:

- $\sigma_{init} = 11$,
- $\sigma_{thres} = 3$, and
- $\mathscr{B} = 0.05$, which means, that only the first component of the background model is considered as background,

where σ_{init} is the initialization value for the variance of a new distribution and σ_{thres} is the threshold value for a pixel to be considered to match a given distribution. These are the most commonly used values in papers reporting the use of Gaussian mixture models for the task of background subtraction.

The masks obtained from background subtraction have been used without any kind of post-processing as input for the FSM. The background models learn for a period of 10 frames and, assuming that at this time the short-term background already has a model of the empty scene, the state machine starts classifying pixels.

The FSM has been implemented as a look-up table and is thus very low demanding in terms of computation time. Only at state AI extra computations are needed. At this step a voting system is used in order to decide the next state for a given input, by comparing the pixel against the last value seen for background at this pixel and imposing the condition of obtaining a

Table 4.2: *Detection results of the DBG-T and DBG-FSM systems.*

	True detections		False detections		Missed detections		Lost detections	
Scene	DBG-T	**DBG-FSM**	DBG-T	**DBG-FSM**	DBG-T	**DBG-FSM**	DBG-T	**DBG-FSM**
AB-Easy	1	**1**	0	**0**	0	**0**	1	**0**
AB-Medium	1	**1**	5	**5**	0	**0**	1	**0**
AB-Hard	1	**1**	6	**6**	0	**0**	1	**0**
cam3	1	**1**	0	**0**	0	**0**	1	**0**
LeftBag	1	**1**	0	**0**	0	**0**	0	**0**

candidate state at least five times. For this comparison, a context variable is needed; namely, the last known background model. The result of this comparison has not been defined as an input of the state machine, since it is only needed for pixels being in the state AI. Therefore, the computational effort of computing a third foreground mask based on this background model is saved.

Static object detection has been made by taking the pixels whose FSM is in the states AP and NI, or in the states PAP and PAPAP for a time longer than 800 frames and building connected components. To build connected components the CvBlobsLib[1] library, which implements the algorithm in [Chang et al., 2004], has been used. Groups of pixels bigger than 200 pixels are classified as static objects. Time and size thresholds were changed for the LeftBag sequence according to the geometry and challenge of the scene. While in the PETS and iLIDS sequences a backpack can be bounded with a 45x45 pixels box, a backpack of the approximately same size takes a box of only 20x20 pixels in the CAVIAR sequence. Moreover, the LeftBag sequence of CAVIAR poses the challenge of detecting static objects being in the scene for 385 frames, what would make no sense in a subway station (iLIDS sequences), since almost each waiting person would trigger an alarm.

4.5.3 Results

Table 4.2 presents the results obtained with the proposed system (DBG-FSM) and with DBG-T. True detections indicate that an abandoned object was detected. False detections indicate that an abandoned object was detected where, in fact, there was not an abandoned object (this includes, for example, the case of a person remaining static during a period of time longer than the time established for the detection of static objects). Missed detections indicate that an abandoned object was not detected. Lost detections indicate correctly detected static objects that were not detected anymore after a given time because of being absorbed by the long-term background model.

[1] http://sourceforge.net/projects/cvblobslib/

The proposed system successfully detects all static objects. Some false detections are reported for the AB-Medium and AB-Hard sequences. These detections correspond to persons staying static for a long time; therefore, they are rated as false detections. These detections could be ignored by incorporating an object classifier in order to discard people staying static for a long time. It should be remarked, that the learning rate for B_L has been set larger than needed in order to prove the correct operation of the FSM, which is also partially the cause of static objects being detected that fast. Furthermore, it can be observed (last two columns in Table 4.2) that the detected static objects are well maintained by the proposed system, while they get lost by by the DBG-T system when they are absorbed by the long-term background model.

Figures (4.5) to (4.7) show some examples obtained for the i-LIDS sequences. Pixels are colored according to the colors of the states shown in Figure 4.2. Pixels belonging to moving objects are painted in green, pixels belonging to short-term static objects in yellow, and so on. The second frame of each sequence shows how time can be used to mark short-term static objects as static. The third frames show how long-term static objects (in red) are still being detected (these objects are lost by a DBG-T system if not additionally using some kind of selective updating scheme for the long-term background model). In the AB-Medium and AB-Hard sequences it is also shown the robustness of the proposed system against occlusions in crowded scenes. The fourth frames show the starting of the disambiguation process when long-term static objects get removed. The fifth frames show how the static object detection stops when the scene background is again identified as such. In Figure 4.7, it can also be observed that some of the false detections could have been avoided by using some kind of region analysis, as is the case of the persons sitting at the bank, where considering the classification of the pixels surrounding the bounding boxes it can be asserted that the region is not isolated and, therefore, can not correspond to an abandoned object.

4.5.4 Computational Load

The processing time needed varies slightly depending on the scene complexity and on the configuration of the underlying background model. A very complex background scene requires more components and thus more computation time. Moreover, when long-term static objects are absorbed by the long-term background and afterwards are removed, an indetermination state has to be solved. Beyond that, the more static objects there are, the more the blobs generation costs. To provide an idea of the computational times, Table 4.3 reports the average frame processing time in milliseconds and in frames per second for the i-LIDS sequences AB-Easy, AB-Medium, AB-Hard and an average over the PETS2006 dataset, running in an Intel Core2 Extreme CPU X9650 at 3.00GHz without threading. Since each pixel is considered individually for the task of background subtraction and multi-class pixel classification, these tasks could be easily implemented in a parallel architecture, gaining considerably in speed. For the analysis of the i-LIDS sequences a region of interest, which comprises the platform and

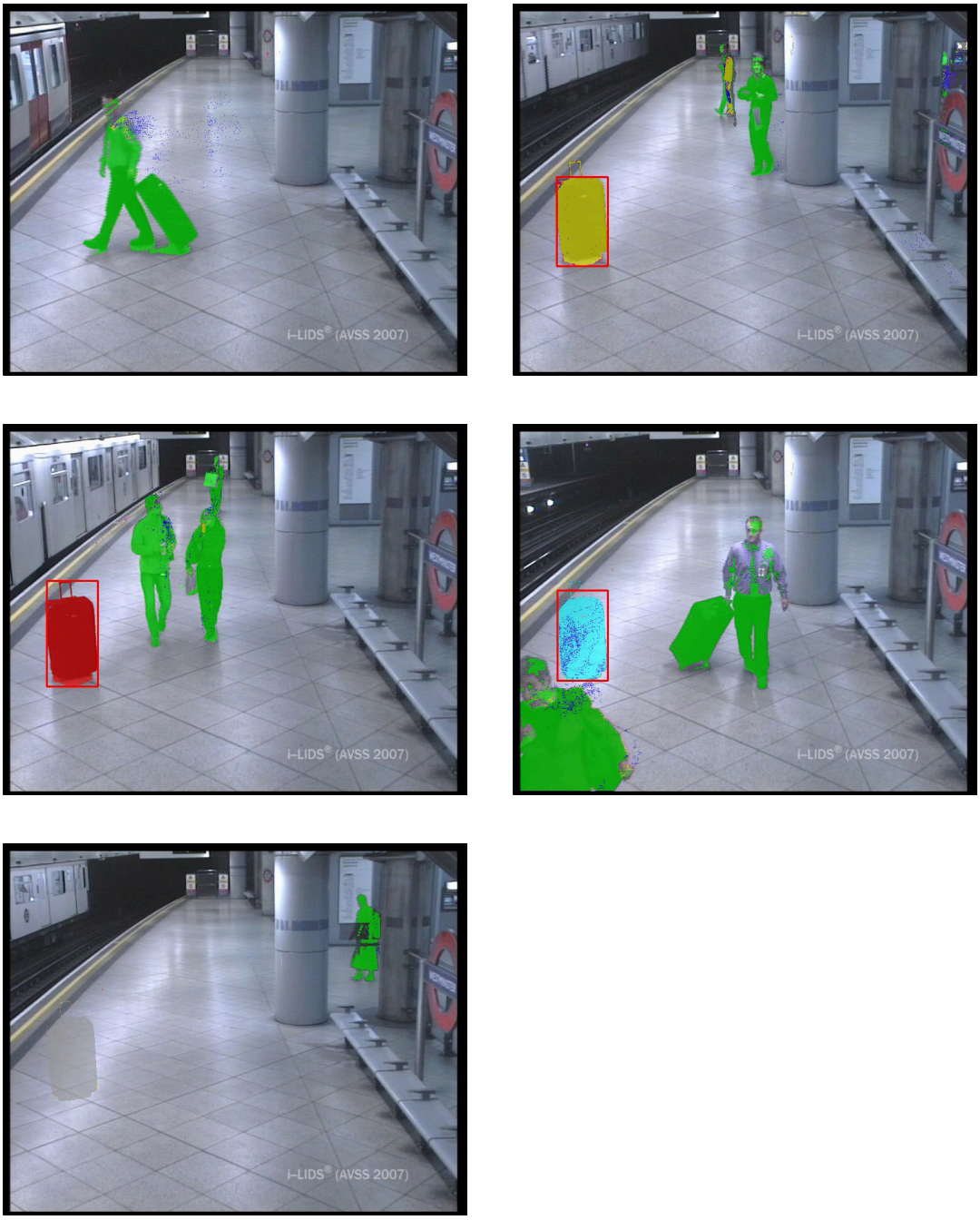


Figure 4.5: *Pixel classification in five frames of the scene AB-Easy. Frame number from left to right and top to bottom: 1967, 3041, 4321, 4922 and 5098.*

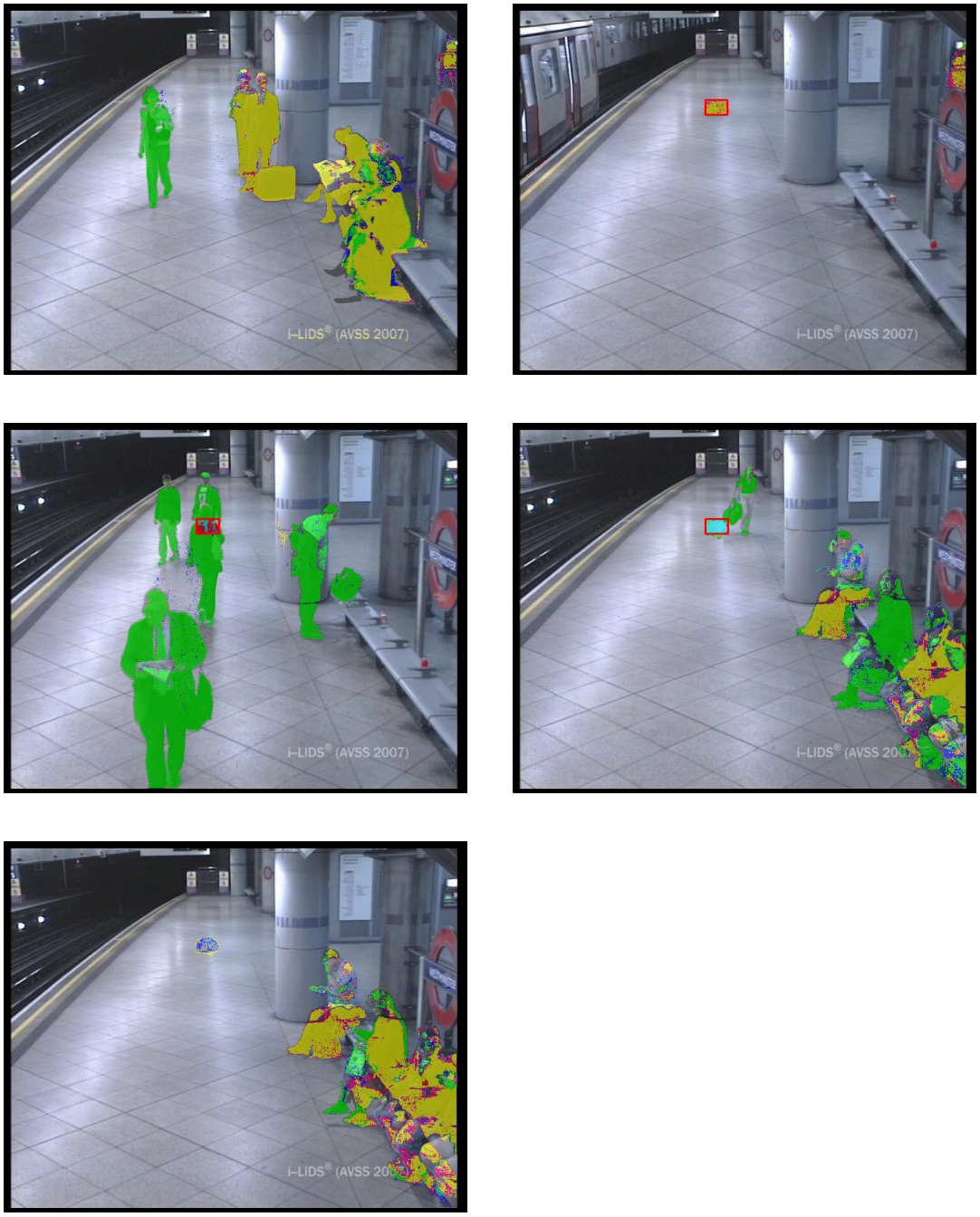


Figure 4.6: *Pixel classification in five frames of the scene AB-Medium. Frame number from left to right and top to bottom: 951, 3007, 3900, 4546 and 4715.*

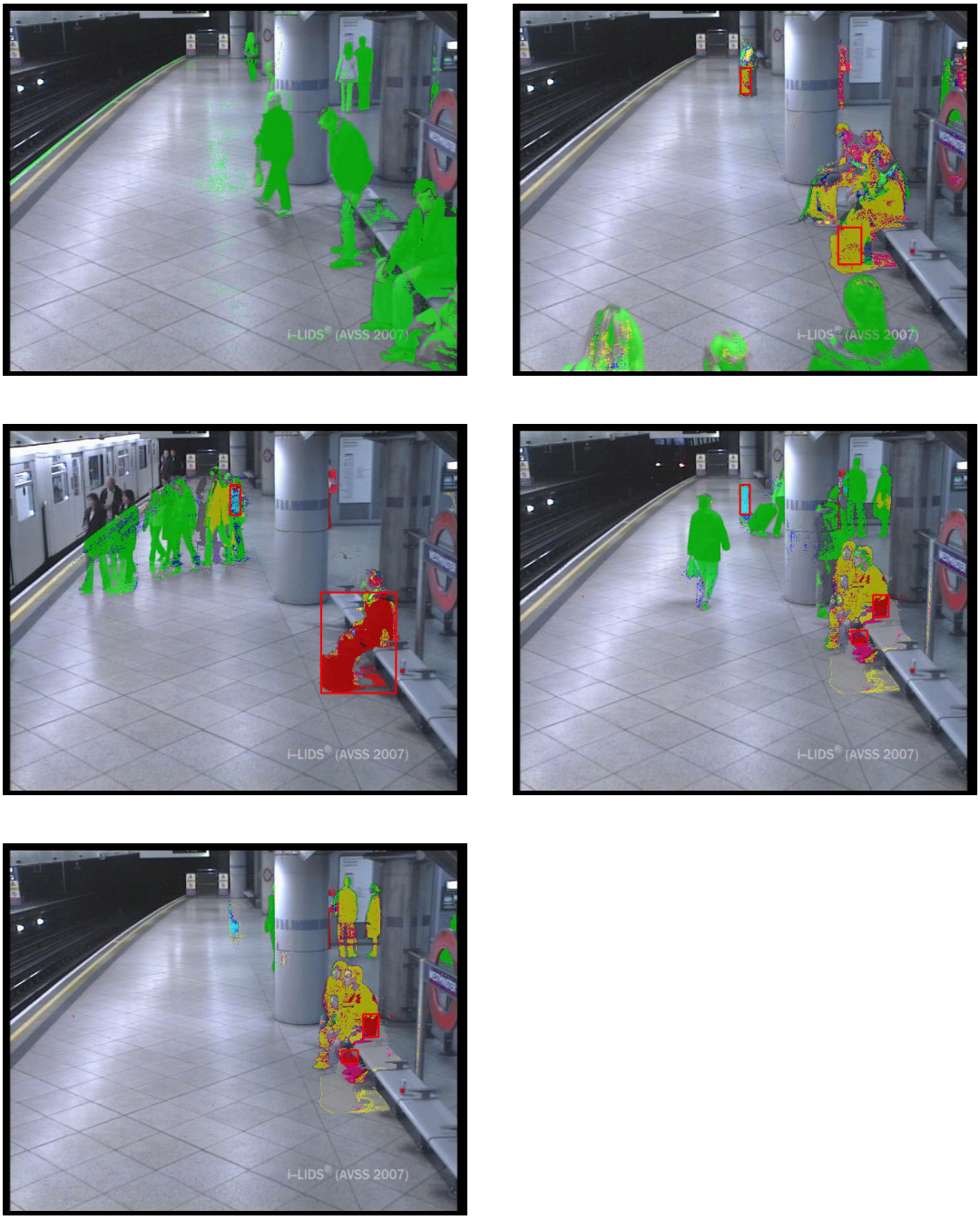


Figure 4.7: *Pixel classification in five frames of the scene AB-Hard. Frame number from left to right and top to bottom: 251, 2335, 3478, 4793 and 4920.*

Table 4.3: *Processing time needed for the update of a dual background model (DBG), for the DBG-T system and for the proposed system (DBG-FSM) in milliseconds (ms) and frames per second (fps).*

	DBG		DBG-T		DBG-FSM	
Scene	ms	fps	ms	fps	ms	fps
AB-Easy	59.30	16.86	62.18	16.08	63.27	15.80
AB-Medium	67.62	14.79	70.57	14.17	72.28	13.84
AB-Hard	68.30	14.64	71.23	14.04	71.87	13.91
PETS2006	60.77	16.46	63.41	15.77	64.58	15.48

escalators, has been defined. That means, 339.028 pixels out of 414.720 have been analyzed. The frames of the PETS2006 dataset have been analyzed without using any region of interest (414.720 pixels per frame). The processing times are provided for the update of a double background system (DBG), for the DBG-T system and for the proposed system (DBG-FSM).

4.5.5 Evaluation of the Results

It is apparent that the proposed method outperforms the DBG-T system in terms of detection accuracy while having similar processing demands. Table 4.3 shows that the computational time needed for the update of the state machine is very low compared to the time needed for the update of the background model. The processing time of the proposed system is always lower when using a state machine with the enhancements proposed in Section 4.4.2, but the most important advantage of using specialized states is the avoidance of states which require solving an indetermination, as shown in Figure 4.4. This clearly improves the quality of the classification results. The times reported show that the system can run in real-time in surveillance scenarios.

A final observation made on the experimental results is that the responsiveness of the long-term background model decreases in crowded environments. This is due to the fact that people tend to dress in similar colors. Therefore, statistically based background subtraction approaches tend to incorporate these colors into the background model. This problem is even worsen when persons or objects remain static for a while at certain positions and then move. As a result, new persons or objects placed at these positions are detected faster as part of the background. Since the proposed system uses two background models which attend to two different temporal configurations and is able to detect the point in time at which static objects are removed, the detection results could be improved by properly managing the background models upon the transitions between states.

4.6 Conclusions

In this chapter, a robust system for the detection of new static objects in crowded scenes has been presented. The proposed system is based on dual background models, which are used to compute short-term and long-term foreground masks, and a FSM that performs multi-class pixel classification based on the results provided by background subtraction and the history of the pixels. The state machine can be implemented as a look-up table with negligible computational cost, it can be easily extended to accept more inputs and can also be coupled with subsequent processing layers in order to extract semantic information out of video sequences. The system has been successfully validated with several public datasets, thereby showing a clear advantage with regard to the detection of long-term static objects.

The objects of interest of the proposed system are static objects which have been introduced in the scene along the considered video sequence and which do not belong to the empty scene. Therefore, the actual appearance of the empty scene must be known when the state machine starts working, which is not a trivial demand to be attended. This issue is managed by the proposed system by setting a background training phase before the start of the multi-class pixel classification process, and, furthermore, by allowing for the incorporation of user knowledge in a principled manner as shown in Section 4.4.4. This knowledge can also be automatically generated by a higher level of analysis.

As described in Section 4.4.3, static objects are detected by grouping pixels which belong to a certain set of classes (states of the FSM). Based on the observation that the inner pixels of slightly moving objects are learned by B_L and B_S faster than the outer pixels (especially by uniform colored objects), it can be differentiated between static objects and slightly moving objects by considering also the classification of neighboring pixels. Figure (4.8) shows a detected static object which could be discarded by following this criterion.

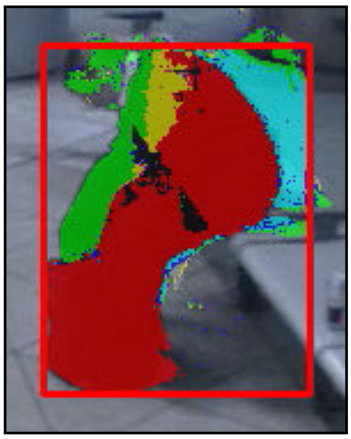


Figure 4.8: *Crop of frame nr. 4158 (i-LIDS AB-Hard).*

Furthermore, it has been observed that the information provided by the FSM and the existence of two background models which attend to different temporal configurations, could be exploited in order to improve the achieved results.

These issues are tackled by the system described in the following chapter by means of region analysis and the use of complementary background models. Furthermore, it is shown that the knowledge acquired by means of region analysis is not only beneficial for the detection of static objects but also for the improvements of the results provided by background subtraction.

5

Complementary Background Models

5.1 Introduction

The detection of change and the detection of new static objects have been studied as isolated problems in previous chapters of this thesis. Specifically, the previous chapter has introduced the main issues associated with the detection of static objects in crowded environments and how these can be addressed by means of dual background models. Moreover, it has been shown how the results provided by a dual background subtraction can be interpreted by a finite state machine in order to handle the problems posed by long-term static objects.

A drawback of dual background based systems is that they require a perfect knowledge of the empty scene in advance. While some efforts aiming to provide a model of the empty scene have been reported in the literature [Gutchess et al., 2001; Farin et al., 2003; Colombari et al., 2006; Reddy et al., 2011] (see Chapter 2, Section 2.4), a common constraint of those systems is that the empty scene must be visible at least for a short time during the initialization period, which is a very challenging demand, especially in public areas, where the background might not be visible for long periods of time.

In this chapter, a system for the detection of static objects is presented which circumvents the initialization problem by first detecting new stationary regions by means of a pixel-wise analysis, and then classifying those stationary regions as new or removed objects (empty scene background) at the region level. Therefore, the background initialization problem is relocated to the right place at the right time. Furthermore, by defining some simple operations on the models corresponding to the pixels affected by the introduction and removal of static objects,

both the foreground segmentation results as well as the static object detection results are considerably improved.

An overview of the proposed system is provided in Section 5.2. Section 5.3 provides a detailed description of the system building blocks. Experimental results are provided in Section 5.4. Section 5.5 concludes this chapter.

The content of this chapter has been partially published in *'Complementary background models for the detection of static and moving objects in crowded environments'*, in the Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2011 [Heras Evangelio and Sikora, 2011a].

5.2 System Overview

The proposed system analyzes each frame of a video sequence at two levels:

- At the **pixel level** two complementary background models are used; a fast learning background model devoted to accurately detect motion, and a more conservative model aiming to reconstruct the empty scene. Pixels are classified according to the results obtained by the subtraction of both background models.
- At the **region level** new static regions are classified as static or removed objects.

The background models are updated in a complementary fashion. While the short-term background model is updated in a blind fashion, incorporating every new observations of the scene, the long-term background model only accepts new stationary descriptions when these are classified as part of the empty scene by means of region analysis. To that aim, region level classification results are fed-back to the first level (background management). Thus, the model of the empty scene is implicitly initialized without the restriction of an initialization period. Furthermore, by using the information gathered at the region level and exploiting the existence of two complementary background descriptions, the background model of the empty scene is rapidly reestablished upon the removal of static or slowly moving objects. Figure 5.1 depicts this system.

The proposed system is able to detect new and removed static objects without previous knowledge of the empty scene and solves some problems that GMM-based approaches, and more generally statistical background models, meet in crowded environments. A first problem derives from the assumption that the scene background is mostly visible, which is not always the case, especially in rush hours, making it difficult to find a good representation of the background. Moreover, people often remain static, getting therefore included in the background model and consequently degrading the model for further detections. Finally, people often dress in similar colors, what might lead to confusing them with the background.

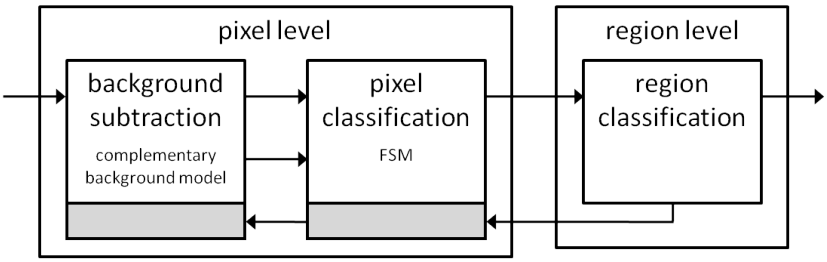


Figure 5.1: *Complementary background models based system.*

Incorporating feedback from higher-level analysis might help to alleviate these problems. One of the few frameworks aiming to incorporate high-level feedback into the update process of the background model is presented in [Harville, 2002], where the author proposes to tailor the segmentation results to the specific needs of the applications using it. Therefore, high-level modules provide feedback in form of a mask, indicating which detections should be ignored in the future and which detections should be maintained. The feedback of the several modules is added-up and the components of the model are accordingly boosted or restrained. Although the benefits of this feedback might be obvious when only one high-level module uses the results of background subtraction, the effectiveness of this approach is drastically reduced when the interests of several modules collide. Furthermore, the results obtained by background subtraction might be difficult to interpret by higher-level analysis modules, since they depend on the number and kind of applications providing high-level feedback.

The proposed system also uses high-level feedback but, instead of tailoring the segmentation results to any specific application, it is used to obtain a reliable description of the empty scene.

5.3 System Building Blocks

5.3.1 Background Modeling

The underlying background model consists of two GMMs each of them defined as in Chapter 3, a short-term model devoted to accurately segment motion and a long-term model where the empty scene is reconstructed. Both models have the same configuration parameters except for the learning rate. The short-term model B_S is fast in adapting to scene changes like illumination and the incorporation and removal of static objects. The long-term model B_L has a lower learning rate and is therefore less reactive than B_S to scene changes. When a pixel is detected as foreground by B_L but not by B_S at time t, it means that a new stable description of the scene has been found. Thus, a new mode has been raised to the background part of B_S.

At this time, B_S is propagated into B_L, the new mode from B_L is dropped and B_L stops from learning. Therefore, B_L keeps a copy of the background scene as known by time $t-1$, and B_S further learns the new description. This process is called B_S `to` B_L `by dropping`.

When a mode is dropped, its weight is divided up on the modes in the background part of the model, therefore avoiding that modes corresponding to the foreground part of the model are upgraded to the background part. That means:

$$\omega_k = \omega_k + \frac{\omega_j}{B-1}, \forall k = 1...B,\ k \neq j, \tag{5.1}$$

where $j \leq B$ is the mode to be dropped and $B > 1$ is the number of modes being part of the background in B_L after propagating B_S into B_L. After dropping the mode, $B = B-1$ in B_L. In other words, two complementary models of the static scene are generated (B_S and B_L). These models are further analyzed at the region level in order to evaluate which of them is more likely to describe the empty scene.

When a pixel is detected as foreground by B_S but not by B_L, it means that the scene background has been uncovered. Therefore, B_L is propagated into B_S in order to reestablish the background model as soon as possible (B_L `to` B_S), thus improving segmentation results.

This process is controlled at the pixel level by a FSM, which classifies pixels and triggers the above mentioned operations for background maintenance. The proposed FSM is described in the following section. New static foreground pixels are then grouped into regions and classified at the region level as shown in Section 5.3.4.

5.3.2 Pixel Classification and Background Control

Since B_L and B_S may contain different descriptions of the static scene, the foreground detection obtained by the subtraction of both models might be different as well. Specifically, at the pixel level, B_S is able to continuously incorporate new descriptions of the static scene, while B_L not. This fact is exploited for detecting pixels describing a new appearance of the static scene (new static foreground pixels). In order to do this classification and to control the corresponding actions on the background model, a FSM is used. The use of a FSM for pixel classification based on the results of a dual background subtraction and on the pixel history has been introduced in Chapter 4. The novelty of the approach presented here is that, the FSM is not only used in order to provide a multi-class pixel classification, but also to actively control the behavior of the background model.

Following the notation in Chapter 4 the FSM is defined as a 5-tuple $(I, Q, Z, \delta, \omega)$, where:

- I are the possible combinations of the results obtained from background subtraction. By defining the pair (F_L, F_S), the input alphabet reduces to $I \equiv \{(0,0),(0,1),(1,0),(1,1)\}$.

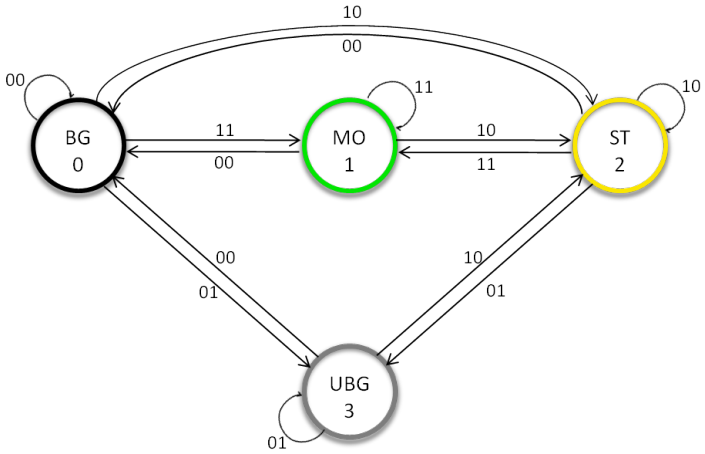


Figure 5.2: *Transition function of the proposed FSM.*

- Q is the set of states a pixel can go through, namely BG -background-, MO -moving-, ST -new static- and UBG -uncovered background-.
- Z is the set of numbers indicating the classification of a pixel $Z \equiv \{0,1,\ldots|Q|-1\}$, with $|Q|$ being the cardinality of Q.
- δ is the next-state function as depicted in Figure 5.2.
- ω is the output function, which is a multivalued function with output values $z \in \{0,1,\ldots|Q|-1\}$, corresponding to the state of a pixel at a given time.

5.3.3 Region Analysis Triggering

Pixels classified as static foreground (ST) are grouped into static foreground regions by means of a connected components labeling algorithm [Chang et al., 2004] (see Chapter 4, Section 4.4.3). These regions can correspond to new static objects or to removed objects which uncover a region of background which was not visible until this point in time. According to B_S they are static, but, according to B_L they are not part of the background as it has been known until this point in time; that means, they are still part of the foreground.

The foreground mask obtained by subtraction of B_L provides the foreground of the overall system. Therefore, the learning rate α_L should be set high enough in order to make B_L able to correctly follow gradual illumination changes. The learning rate α_S of the short-term background model B_S is set higher than α_L and, thus, slowly moving and temporarily

static objects are absorbed faster by B_S. In order to avoid unnecessarily analyzing regions corresponding to slowly moving objects, only those pixels which have been classified as static longer than a given time T_S are considered as part of a static foreground region. This time is computed based on α_S and α_L as follows:

$$T_S = \frac{(\log(T)/\log(1-\alpha_L)) - (\log(T)/\log(1-\alpha_S))}{2} \tag{5.2}$$

That means, half of the time remaining from the inclusion of a new mode in the background part of B_S until the time it should get included in the background part of B_L is allocated for pixel classification validation. The rest of the time $T_C = T_S$ is reserved for the region classification process.

The analysis of a static foreground region starts when the region has been well-formed. A region is considered to be well-formed when it fulfills the following two conditions: (i) stillness, that means that its position and size remain equal for a predefined period of time, and (ii) isolation, that means that the pixels bordering its bounding box belong to the background. Therefore, a list of the static foreground regions found in the recent frames is maintained, where, for each region, information regarding its bounding box (position and size), the first time it was seen, the last time it was seen and its classification status (new static object, removed static object, or a soft classification score) is kept. For each new frame, it is checked for associations between the detected static foreground regions and those in the list based on the position and size of their respective bounding boxes. Detected regions which do not have a match in the list are inserted in a new list element. Detected regions matching an element of the list are used to update its related information. Furthermore, the fulfillment of the stillness and isolation criteria is checked. Static foreground regions intersecting with each other will mostly do not fulfill the isolation criterion. In order to avoid indefinetily waiting for isolation at those regions, regions fulfilling the stillness condition for a period of time longer than the half of the time allocated for region classification (T_C) are deemed to be well-formed.

The process of associating the detected static foreground regions to those in the list is closely related to the standard tracking problem. Nevertheless, there are some important differences:

- Since the considered regions are not expected to move around in the scene (or at least not to much), there is no need to assume neither a motion model nor a calibration of the scene.
- Although the regions of interest might appear or disappear from the observed scene, there are not entry or exit points; therefore, some of the heuristics applied in commonly used tracking approaches cannot be applied here; the list elements corresponding to regions which have not been observed during a period of time longer than T_{max} frames are deleted.

- Merging and splitting operations are allowed, but there is no disambiguation necessity upon the splitting of regions, since no tracking or behavior recognition are performed based on the associations.

The region association algorithm used is an extension of the point correspondence paradigm which uses the bounding box delimiting the considered regions as a further constraint. The criterion used to establish correspondences is the minimum cost, being the cost function $D_{list}(R_i, L_j)$ between two regions defined as per Equation 5.3. Similar approaches are used in [Javed and Shah, 2002] and in [Johnsen and Tews, 2009].

$$D_{list}(R_i, L_j) = \begin{cases} \frac{\text{Area}(R_i)+\text{Area}(L_j)-\text{Area}(R_i \cap L_j)}{\text{Area}(R_i \cap L_j)} & if\ \text{Area}(R_i \cap L_j) > 0 \\ \infty & if\ \text{Area}(R_i \cap L_j) = 0 \end{cases} \tag{5.3}$$

where R_i is a given detected region i, L_j is a region j in the list, and $R_i \cap L_j$ is the intersecting region between R_i and L_j. That means, that an infinity cost is assigned if the regions are not intersecting. If the regions are intersecting, the cost is 1 for completely overlapping regions, and grows inversely proportional to the portion of region overlap.

For every frame, a distance matrix containing the distance of every detected region to every element in the list is computed. The rows of this matrix correspond to the elements in the list and the columns to the detected regions. Additionally, a decision matrix of the same dimensions is built with all elements set to zero. For every row, the cell corresponding to the lowest value of the distance matrix is incremented by one. The same is done for every column. Therefore, each cell of the decision matrix has a value between zero and two.

Elements of the decision matrix equal to two are directly associated and the rest of the distances in the distance matrix corresponding to the same column and row are set to infinity. This process is iteratively repeated until none of the elements in the decision matrix is equal to two. Elements of the list for which a matched could not be found are maintained in the list until the time difference between the current frame T_{cf} and the last time they were detected T_{lf} is bigger than a certain threshold T_{max}, assuming that they have been removed. Detected regions for which a match in the list has not be found, are inserted in the list.

It must be emphasized that the described association process is exclusively employed in order to collect temporal information about the detected static foreground regions, but not to build tracks out of which derive any kind of behaviors. Therefore, the described process is robust even in crowded environments, where occlusions make difficult the task of building reliable tracks due to the need of disambiguating the tracked regions upon merging and splitting operations. Furthermore, since the expected number of static foreground regions is low, the computational demands of the described algorithm can be neglected.

5.3.4 Static Foreground Regions Classification

Static foreground regions are classified as soon as possible so as to reactively update the background models at those regions classified as uncovered background. Static object alarms are triggered after a time T_A, depending on specific application requirements. Since the number of the frame at which each static foreground region has been seen for the first time is kept, triggering alarms does not depend on the time at which the region is classified (provided that $T_C \leq T_A$). Well-formed static foreground regions (see Section 5.3.3) are analyzed based on shape information when no moving objects are passing through (this can be easily checked by counting the number of pixel detections provided by B_S), therefore, avoiding clutter in the classification process. This process is defined as follows:

- Find the edges corresponding to the selected region in the input frame. In the following, the resulting image is referred to as I_R^e. Analogously, the edges corresponding to the selected region in B_L and in the mask of the static region are referred to as $B_{L,R}^e$ and M_R^e, respectively.

- If there are edges in $B_{L,R}^e$ but not in I_R^e, assume that a static object has been removed. In the opposite case, assume that a new static object has been placed. If both, $B_{L,R}^e$ and I_R^e, contain edges, continue the analysis.

- Compute the chamfer distance of M_R^e to $B_{L,R}^e$, $D_{chamfer}(M_R^e, B_{L,R}^e)$, and to I_R^e, $D_{chamfer}(M_R^e, I_R^e)$. This is achieved by:

  - Computing the distance transform of I_R^e and $B_{L,R}^e$. In the proposed system, the Chamfer 3-4 distance [Borgefors, 1986] is used [1].

  - Summing up the pixel values of the respective distance transform image which lie in the same position as the edges in M_R^e.

- If the edges of the region mask M_R^e can be unambiguously matched either to B_L or to the input frame I_R, classify as removed object or new static object, respectively. That means,

$$\begin{cases} if\ T_{chamfer} \cdot D_{chamfer}(M_R^e, B_{L,R}^e) < D_{chamfer}(M_R^e, I_R^e) & R \text{ is a removed object} \\ if\ T_{chamfer} \cdot D_{chamfer}(M_R^e, I_R^e) < D_{chamfer}(M_R^e, B_{L,R}^e) & R \text{ is a new object} \end{cases} \tag{5.4}$$

  where $T_{chamfer}$ is an empirically set value which regulates how much near M_R^e has to be to any of the considered reference regions, $B_{L,R}^e$ and I_R^e in order to consider it a match.

[1] Using the Euclidean distance is usually not necessary, as the edge points are influenced by noise, being, therefore, a waste of effort to compute exact distances from inexact edges [Borgefors, 1988].

- If an unambiguous classification is not possible, a soft classification score is accumulated. The soft classification score is computed as the rate of the distance to the less acceptable match over the distance to the more acceptable match and indicates how much near is the region to the more acceptable match. This score is accumulated until a classification can be done.

- If a region has still not been classified in the time T_C allocated for region classification, it is incorporated in the background part of B_L, therefore, avoiding to keep in the foreground regions which cannot be reliably classified.

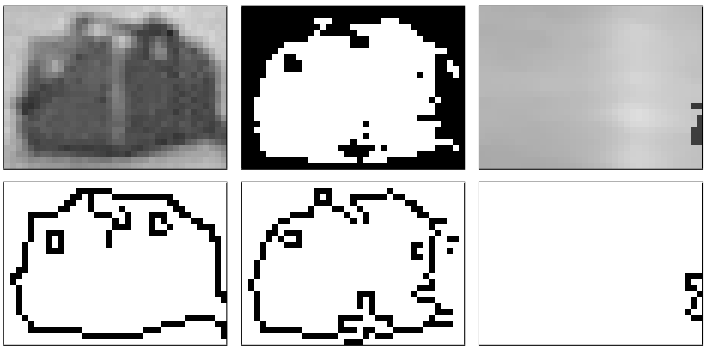
The described distance measures $D_{chamfer}(M_R^e, I_R^e)$ and $D_{chamfer}(M_R^e, B_{L,R}^e)$ correspond to the Chamfer distance and provide a value of dis-similarity between two images. The lower the value is, the better the match.

Chamfer matching has been extensively used in the computer vision literature. As the number of templates and the size of the reference image grow, an exhaustive search might result in an elevated computational cost. Nevertheless, in the described procedure, the defined distance measures are computed one time, respectively, for each classification trial. Therefore, the described classification procedure can be implemented very efficiently.

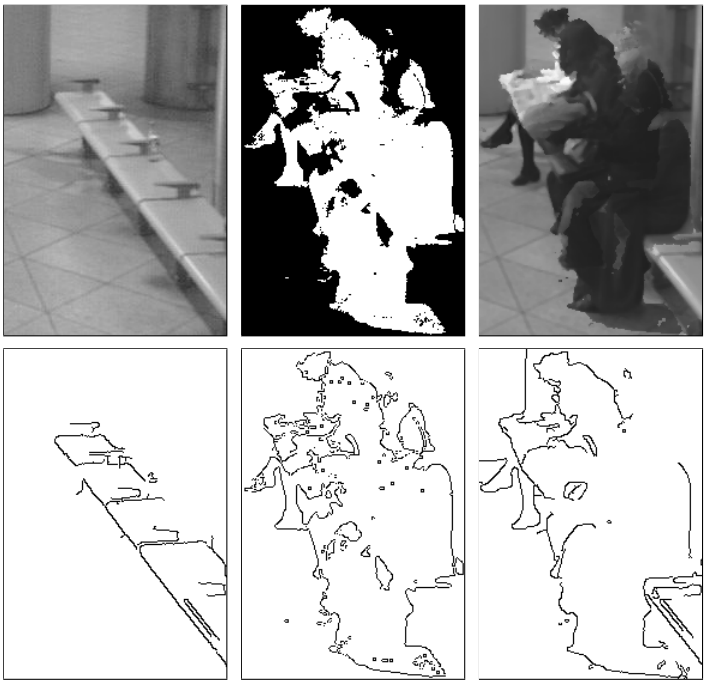
I_R^e, $B_{L,R}^e$ and M_R^e are computed by using the Canny edge detector [Canny, 1986], which is a generally well accepted robust edge detector optimal with respect to the detection (the ability to detect as many existing edges in the image as possible), the localization (the detected edges should be as close as possible to the existing edge in the image) and minimal response (the existing edges in the image should only be detected once). This is achieved in four main processing steps: noise reduction (implemented with a Gaussian filter), gradient magnitude and angle computation, non-maxima suppression and hysteresis thresholding.

The Canny detector has been selected for its robustness. Since its introduction, there have been several proposals to improve its performance in different directions, which have not been explored here for being out of the scope of the main topic of this thesis. For instance, in [Deriche, 1987] an approach based on a highly efficient recursive algorithm aiming to save the computational effort imposed by the Canny detector is presented. Furthermore, an additional surround suppression step aiming to eliminate texture edges is presented in [Grigorescu et al., 2004]. Further edge detection algorithms can be found in [Ziou and Tabbone, 1998; Basu, 2002]. While the presence of textures is not a big issue for the presented approach, since the static foreground mask is mostly expected to fit better to the region containing the object, a computational save would certainly be of advantage, provided that the quality of the detection results does not degrade.

Figure 5.3 shows examples of regions classified as new static object and removed object by following the described classification process.



(a) *Region classified as new static object. Sequence AVSS 2007 AB Medium, frame nr. 1605. Top row, from left to right are input frame, pixel classification mask and B_L at the analyzed region. Bottom row are the corresponding edges. The edges of the pixel classification mask can be matched to the edges in the input frame.*



(b) *Region classified as removed object. Sequence AVSS 2007 AB Medium, frame nr. 2245. Top row, from left to right are input frame, pixel classification mask and B_L at the analyzed region. Bottom row are the corresponding edges. The edges of the pixel classification mask can be matched to the edges of the corresponding region in B_L.*

Figure 5.3: *Region classification.*

Figure 5.4: *Feedback Triggered Background Update. B_L at frames nr. 2245 and 2246 of the sequence AVSS 2007 AB Medium. Four persons are sitting on the bench since the first frame of the sequence. The bench is left visible at frame 2084, arising therefore a static region. At frame 2245 the detected static region is classified as removed object and high level feedback triggers an update of B_L, thus propagating B_S into B_L.*

5.3.5 Feedback Triggered Background Update

Using the feedback provided by the described static foreground regions analysis, the proposed system prevents from including new static foreground objects in the background part of B_L, while remaining flexible to incorporate parts of the background which had been occluded along the whole sequence, not being constrained to an initialization period. Furthermore, since the most recent copy of the scene background is kept in B_L upon the appearance of new static objects, B_L can be used to rapidly heal B_S upon their removal, therefore, improving segmentation results.

The incorporation of uncovered regions of the empty scene is accomplished by feeding back the position and size of the corresponding bounding box to the pixel level. B_S is propagated into B_L for those pixels classified as new static foreground (state ST). Figure 5.4 shows an example of an uncovered background region initialized by this means.

5.4 Evaluation

The proposed system has been evaluated regarding its ability to detect and correctly classify new and removed static objects, and regarding the quality of the provided foreground masks. The results of this evaluation are presented in this section.

5.4.1 Datasets

The static object detection evaluation has been conducted by using the same set of sequences as in Chapter 4 (AB sequences of the i-LIDS dataset, the S1-T1-C-3 sequence of the PETS2006

Table 5.1: *Detection results of the DBG-T (DT), DBG-FSM (DF), and CBG-FSM (CF) systems. A manually generated perfect knowledge of the empty scene was needed for DBG-T and DBG-FSM. CBG-FSM was able to automatically generate this knowledge 'on the fly'.*

	True Detections			False Detections			Missed Detections			Lost Detections		
Scene	DT	DF	**CF**	DT	DF	**CF**	DT	DF	**CF**	DT	DF	**CF**
AB-Easy	1	1	**1**	0	0	**0**	0	0	**0**	1	0	**0**
AB-Medium	1	1	**1**	5	5	**4**	0	0	**0**	1	0	**0**
AB-Hard	1	1	**1**	6	6	**2**	0	0	**0**	1	0	**0**
cam3	1	1	**1**	0	0	**0**	0	0	**0**	1	0	**0**
LeftBag	1	1	**1**	0	0	**0**	0	0	**0**	0	0	**0**

dataset, and the LeftBag sequence of the CAVIAR dataset). These sequences have been briefly described in Section 4.5, Evaluation, of Chapter 4. A thorough description of the datasets used throughout this work is provided in Appendix A, Description of Datasets.

For the evaluation of the quality of the provided foreground masks the CDnet dataset has been used. The dataset has been briefly described in Section 3.5, Evaluation, of Chapter 3. A thorough description of this dataset is provided in Appendix A, Description of Datasets.

5.4.2 Static Object Detection Evaluation

In order to evaluate the detection of new and removed static objects, the system presented in Chapter 4 (DGB-FSM) and a dual background based system similar to the one proposed in [Porikli et al., 2008] (DGB-T) have been taken as reference systems. The system proposed in this chapter is referred to as CBG-FSM.

The reference systems and the proposed system have been compared by using GMMs with identical configuration as underlying background models. The same configuration parameters as in Section 4.5 of Chapter 4 have been taken, i.e. $\alpha_S = 0.004$ for the short-term background model and $\alpha_L = 0.0004$ for the long-term background model. The DGB-T and DGB-FSM systems need a perfect knowledge of the empty scene. Therefore, a model of the empty scene has been manually generated for the AB sequences of the i-LIDS dataset. The PETS2006 and CAVIAR sequences start with an empty frame; therefore, the first frame, respectively, has been taken as model of the empty scene.

Table 5.1 shows the results obtained with the compared systems. True detections (TD) are detected abandoned objects. False detections (FD) are objects detected as static, which do not correspond to abandoned objects (a person, for example). Missed detections (MD) are not detected abandoned objects. Lost detections (LD) indicate detected abandoned objects that are not detected anymore after a given time because of being absorbed by the long-term background model.

One of the major advantages of the proposed system is that it does not need any previous knowledge of the empty scene. In fact, the CBG-FSM system outperforms the DBG-T and DBG-FSM systems even if the DBG-X systems have been started with a perfect initialization of the background model (CBG-FSM was initialized without previous knowledge). Moreover, since object classification is done at the region level when the new static regions are stable, CBG-FSM is able to filter out many objects which are nearly static, i.e. persons waiting at a fixed position which only slightly move some parts of their body like arms or legs. Therefore, false detections are reduced. Furthermore, since every detected static foreground region is analyzed at the region level, CBG-FSM is more resilient than DBG-T and DBG-FSM to potential previous failures of the system.

Figures 5.5 to 5.7 depict some pixel classification examples for the sequences AB-Easy, AB-Medium and AB-Hard of the i-LIDS dataset.

Figure 5.5 shows (on the left) that the CBG-FSM and DBG-FSM systems are able to hold the detection of the suitcase, while the DBG-T loses it. After the suitcase has been removed, a person walks through the place where the suitcase had been staying for a long time. Thanks to the rapid healing of background areas upon removal of static foreground objects, the CBG-FSM is the unique system able to correctly segment this person. This is depicted in the frames on the right.

Figure 5.6 shows (on the left) that the CBG-FSM is able to correctly integrate on the background the area where persons were sitting at the beginning of the sequence. On the right, it is shown, that the CBG-FSM and DBG-FSM systems are able to hold the detection of the abandoned bag, while the DBG-T loses it. Furthermore, the person approaching the bench is better segmented by the CBG-FSM. This is due to the fact that this area is frequently occluded by people passing by. While the DBG-T and DBG-FSM systems slowly integrate in the background models frequently observed colors, therefore missing some foreground detections, the CBG-FSM system prevents the incorporation of these colors into the background model, therefore achieving more accurate segmentation results.

This fact is also depicted in Figure 5.7, where the person entering the scene (frames on the left) and the persons sitting on the bench (frames on the right) are more accurately segmented by the CBG-FSM.

5.4.3 Computational Load

In terms of computational demands, DBG-T and DBG-FSM exhibit a very similar behavior, requiring an average processing time of near 68 ms per frame ($\approx$ 14.7 frames per second) for the AB sequences of the i-LIDS dataset in an Intel Core2 CPU at 3.00GHz without threading, while the CBG-FSM system needed an average processing time of near 86 ms per frame ($\approx$ 11.6 frames per second). It is important though to remark, that the processing time measured for

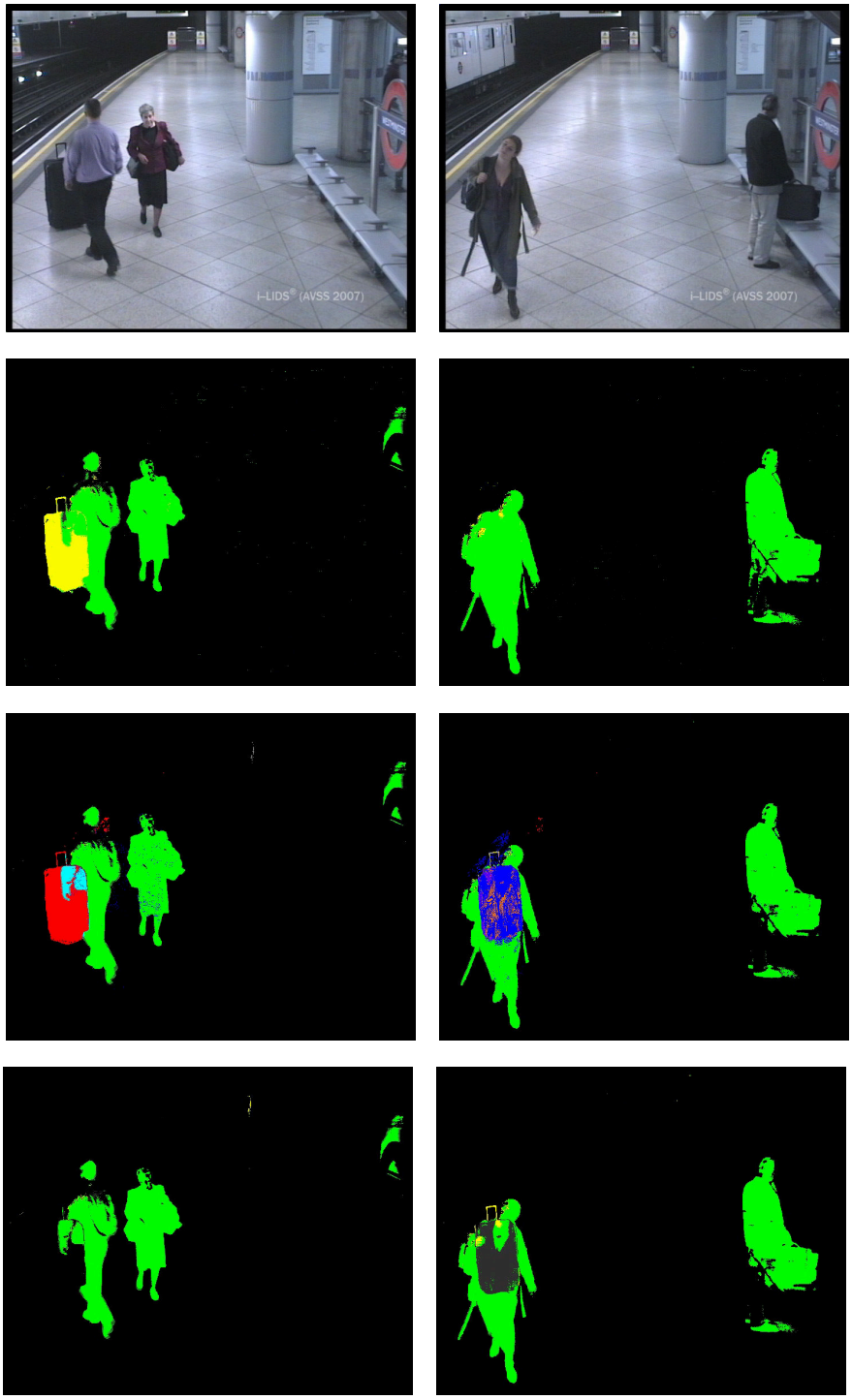


Figure 5.5: *Pixel classification in two frames of the scene AB-Easy. From left to right: frames number 4783 and 5441. From top to bottom: original frame, CBG-FSM, DBG-FSM, and DBG-T.*

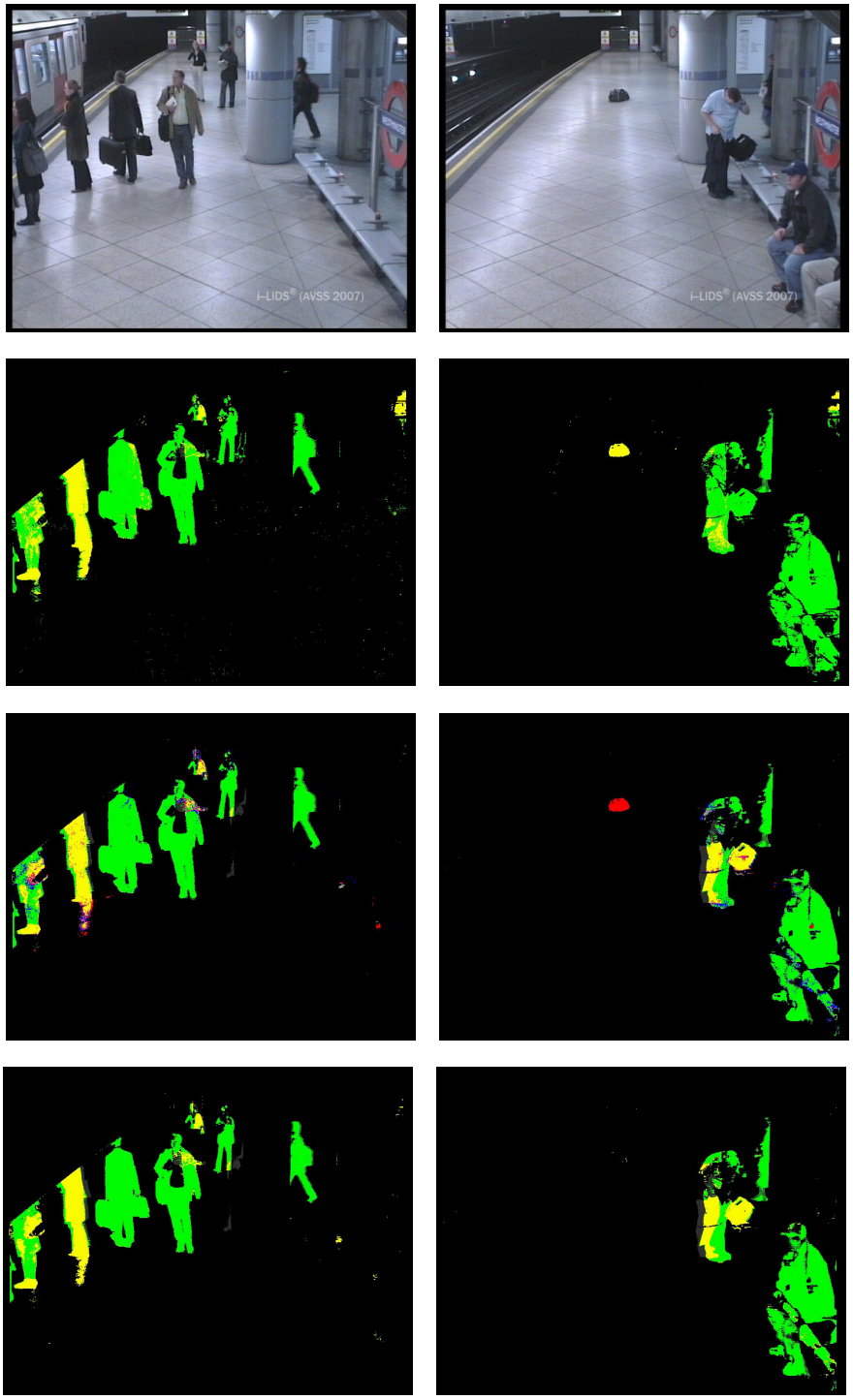


Figure 5.6: *Pixel classification in two frames of the scene AB-Medium. From left to right: frames number 2394 and 4164. From top to bottom: original frame, CBG-FSM, DBG-FSM, and DBG-T.*

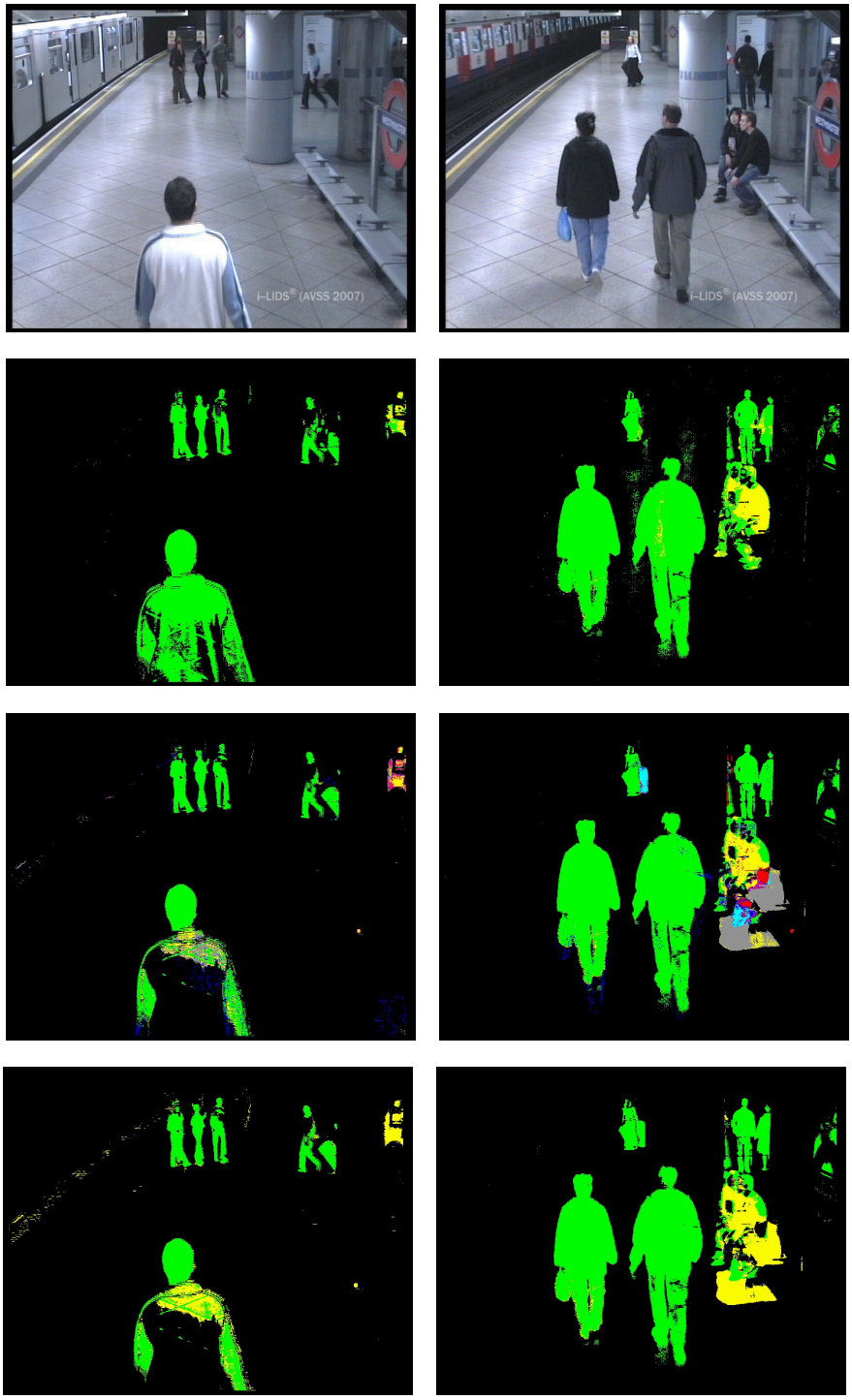


Figure 5.7: *Pixel classification in two frames of the scene AB-Hard. From left to right: frames number 1087 and 4698. From top to bottom: original frame, CBG-FSM, DBG-FSM, and DBG-T.*

DBG-T and DBG-FSM does not account for any event association by the detection of objects while this task is implicitly accomplished by the CBG-FSM in form of a list. Therefore, an application based on the CBG-FSM system would be able to raise only one alarm for each static object, while an application based on the DBG-T or DBG-FSM system would need to associate the objects detected along the frames in order to raise alarms. In fact, the time needed by the CBG-FSM system only for pixel classification and background update is slightly slower than the time needed by DBG-T and DBG-FSM. This is mainly due to the simplification of the background models achieved when temporarily static objects are removed.

5.4.4 Background Subtraction Qualitative Evaluation

For the evaluation of the quality of the provided foreground masks, the proposed system has been implemented with an underlying SGMM model, as introduced in Chapter 3. The resulting system is referred to as SGMM-SOD in the following, since this is the name that was used to commit the obtained results to the site hosting the 'changedetection' challenge.

The long-term SGMM has been configured by using the same parameter set used for the evaluation of the SGMM method in Section 3.5, Evaluation, of Chapter 3, i.e., a maximum number of five Gaussians per pixel was used, $\alpha_L = 0.005$, a sigma spanning factor $c = 3$, and a brightness distortion $l < \lambda < u$, with $l = 0.85$ and $u = 1.10$.

The short-term SGMM has been identically configured as the long-term SGMM except for the learning rate α_S, which has been set 10 times higher than α_L, i.e., $\alpha_S = 0.05$.

A 5x5 median filter has been applied in a post-processing step, as the organizing committee had done with the results provided by the methods proposed for the benchmark.

Overall, the SGMM-SOD system needs three parameters more than the original GMM formulation, one for controlling the span of sigma, one for the learning rate relation between the long-term and the short-term background models, and two for the lighting detection, minus the one needed for the initialization of the variance parameter, which is automatically set by the SGMM system.

The meaning of the sigma spanning factor c has been discussed in Section 3.5, Evaluation, of Chapter 3.

The learning rate relation between the long term and the short term background model affects the time available for region classification. Therefore, it should be set high enough so that static foreground regions classification can be reliably classified, while low enough so that the short term background model does not absorb slowly moving objects. Nevertheless, since the long term background model sets the upper bound for region classification, the segmentation

Table 5.2: *Segmentation results and ranking of SGMM-SOD (06.06.2013). Top 10 change detection algorithms as ranked by the CDnet benchmark. The proposed algorithm is SGMM-SOD.*

Method	Average ranking across categories	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
Spectral-360	3.83	3.86	0.7770	0.9920	0.0080	0.2230	1.8516	0.7770	0.8461
SGMM-SOD	4.00	4.29	0.7697	0.9938	0.0062	0.2303	1.4960	0.7661	0.8339
PBAS	4.67	5.43	0.7840	0.9898	0.0102	0.2160	1.7693	0.7532	0.8160
DPGMM	6.00	5.57	0.8275	0.9855	0.0145	0.1725	2.1159	0.7763	0.7928
PSP-MRF	7.50	8.71	0.8037	0.9830	0.0170	0.1963	2.3937	0.7372	0.7512
ChebProb-SOD	9.50	9.57	0.7133	0.9888	0.0112	0.2867	2.3856	0.7001	0.7856
SC-SOBS	9.67	9.43	0.8017	0.9831	0.0169	0.1983	2.4081	0.7283	0.7315
CDPS	10.17	9.14	0.7769	0.9848	0.0152	0.2231	2.2747	0.7281	0.7610
SGMM	11.17	9.43	0.7073	0.9910	0.0090	0.2927	2.5311	0.7008	0.7812
KNN	11.50	11.14	0.6707	0.9907	0.0093	0.3293	2.7954	0.6785	0.7882

Table 5.3: *Segmentation results and ranking of the top three change detection algorithms in the CDnet benchmark by using a 9x9 median post-filtering (06.06.2013).*

Method	Average ranking across categories	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SGMM-SOD	3.33	2.86	0.7972	0.9931	0.0069	0.2028	1.4094	0.7812	0.8390
Spectral-360	3.67	4.29	0.7770	0.9920	0.0080	0.2230	1.8516	0.7770	0.8461
PBAS	4.50	5.71	0.7840	0.9898	0.0102	0.2160	1.7693	0.7532	0.8160

performance of the system is not much affected by this parameter. A value of 10 has provided good results, given the learning rate of the long term background model.

Table 5.2 shows the overall performance achieved by the top 10 algorithms. By the time of writing this thesis, the proposed method is being ranked in the second position. It must be noted that the results of the two methods (PBAS [Hofmann et al., 2012], Spectral-360 [Sedky et al., 2008]) sharing the top three with the proposed approach were post-processed with a 9x9 median filter, while the results of the SGMM-SOD method with a 5x5 median filter. Although the SGMM-SOD method outperforms these two methods if applying a 9x9 median filter, being, therefore ranked in the first position (see Table 5.3), the results obtained with a 5x5 median post-filtering have been published, so as to allow for a straightforward comparison with the GMM approaches which had already been evaluated within the benchmark. Furthermore, a 5x5 median post-filtering is much lighter in terms of computational load.

Table 5.4: *Segmentation results and ranking of SGMM and SGMM-SOD across the six categories in the CDnet dataset (06.06.2013).*

Category	Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
Baseline	SGMM-SOD	8.00	0.9334	0.9974	0.0026	0.0666	0.5494	0.9212	0.9113
	SGMM	16.71	0.8680	0.9949	0.0051	0.1320	1.2436	0.8594	0.8584
Dynamic Background	SGMM-SOD	9.71	0.7786	0.9966	0.0034	0.2214	0.6041	0.6883	0.7044
	SGMM	13.71	0.7715	0.9933	0.0067	0.2285	0.9132	0.6380	0.6665
Camera Jitter	SGMM	8.00	0.7088	0.9869	0.0131	0.2912	2.3761	0.7251	0.7752
	SGMM-SOD	8.86	0.6113	0.9907	0.0093	0.3887	2.3608	0.6724	0.8040
Intermittent Object Motion	SGMM-SOD	3.00	0.7363	0.9909	0.0091	0.2637	2.5238	0.7151	0.8141
	SGMM	10.86	0.5013	0.9853	0.0147	0.4987	4.9180	0.5397	0.6993
Shadow	SGMM-SOD	4.71	0.9191	0.9902	0.0098	0.0809	1.2534	0.8646	0.8226
	SGMM	10.00	0.8580	0.9889	0.0111	0.1420	1.7965	0.7944	0.7617
Thermal	SGMM-SOD	7.00	0.6396	0.9971	0.0029	0.3604	1.6846	0.7353	0.9471
	SGMM	13.29	0.5363	0.9970	0.0030	0.4637	3.9394	0.6481	0.9263

Including the one presented in this chapter, there have been seven GMM approaches evaluated with the CDnet dataset until the date of writing this work, the original formulation in [Stauffer and Grimson, 1999] (GMM | Stauffer & Grimson), the GMM with a preliminary learning phase [Kaewtrakulpong and Bowden, 2001] (GMM | KaewTraKulPong), the GMM with adaptive selection of the number of components [Zivkovic, 2004] (GMM | Zivkovic), a block-based GMM [Riahi et al., 2012] (GMM | RECTGAUSS-Tex), a Dirichlet process GMM followed by probabilistic regularization [Haines and Xiang, 2012] (DPGMM), the SGMM approach presented in Chapter 3, and the approach presented in this chapter (SGMM-SOD). Among them, the one presented in this chapter provides the best performance both in the average ranking and in the average ranking across categories, followed by the DPGMM approach, whereby the DPGMM approach needs considerably more processing time although the sequences had been processed based on a OpenCL version running on a GeForce GTX 580 while the processing time of the SGMM-SOD method has been measured based on a non-threaded version. The next best performing GMM-based method is the one we presented in Chapter 3. Table 5.4 shows a comparison of the performance of the method presented in this chapter (SGMM-SOD) with the method in Chapter 3. The categories where the approach presented in this chapter has a more obvious advantage are those involving new and removed static objects, which are mostly correctly classified and accordingly preserved from being integrated in the background model or removed from it, respectively. These categories are

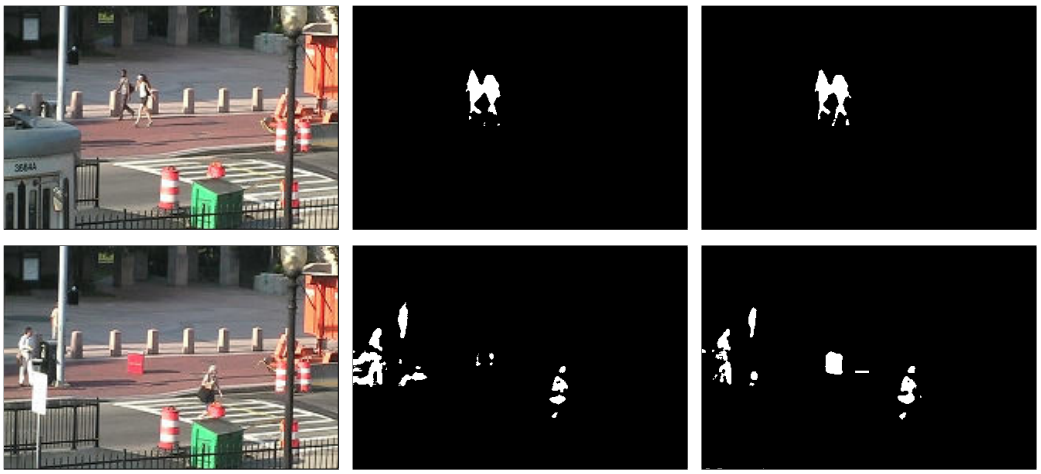


Figure 5.8: *Foreground segmentation results for two frames of the 'tramstop' sequence. From top to bottom: frames number 653 and 1530. From left to right: original frame, foreground mask of SGMM and foreground mask of SGMM-SOD.*

Baseline, Intermittent Object Motion, Shadow and Thermal. Therefore, both the Precision and the Recall of the segmentation results increase. The results for the Camera Jitter sequences are slightly worse for the SGMM-SOD approach ($\approx 5\%$ attending to the average F-Measure), due to the fact that some static foreground regions appearing when the camera changes its position after a vibration are wrongly detected as static objects by the region analysis layer and, therefore, prevented from being integrated in the background model while the SGMM approach better adapts to these changes.

Figures 5.8 and 5.9 depict two exemplary segmentation sequences which illustrate the improvement in the segmentation results achieved by means of region analysis feedback. Figure 5.8 shows two frames corresponding to the 'tramstop' sequence on the left column and the corresponding foreground masks obtained by the SGMM and by the SGMM-SOD methods on the middle and the right columns, respectively. The 'tramstop' sequence starts with a tram standing at a tramstop. Towards frame number 1000, the tram starts moving and leaves the tramstop, while a person leaves a big box on the sidewalk. Figure 5.8 shows that SGMM incorporate these two stationary changes into the background model towards frame number 1500 while SGMM-SOD is able to differentiate between the removed static object (the tram) and the new static object (the box) and correspondingly integrates them or not into the background model.

Figure 5.9 shows three frames corresponding to the 'copyMachine' sequence on the left column and the corresponding foreground masks obtained by the SGMM and by the SGMM-SOD methods on the middle and the right columns, respectively. The 'copyMachine' shows the

Figure 5.9: *Foreground segmentation results for three frames of the 'copyMachine' sequence. From top to bottom: frames number 147, 824 and 2686. From left to right: original frame, foreground mask of SGMM and foreground mask of SGMM-SOD.*

activities of different persons in a room with two machines. At the beginning of the sequence there is no person present in the room. Towards frame number 150 two persons are entering the room. One of them is staying operating at the machine until approximately frame number 1000, while the second person is waiting. As the middle row of Figure 5.9 shows, SGMM integrates the persons into the background model, while SGMM-SOD is able to hold them in the foreground. As a consequence, persons wearing similar colors operating afterwards at the machine are not properly detected by SGMM, while SGMM-SOD is able to correctly segment them. An example of this is provided in the lower row of Figure 5.9.

5.5 Conclusions

In this chapter, a robust system for the detection of new and removed static objects in crowded scenes has been presented. The proposed approach is based on a complementary background subtraction system consisting of two GMM-based models learning at different rates and with different updating mechanisms. While the short-term background model adapts rapidly to all the changes in the scene, the long-term background model only incorporates changes

into the model if they belong to the background. The results provided by the complementary background subtraction are used as input of a FSM, which performs a multi-class pixel classification. Pixels classified as static foreground are then grouped into regions, which are classified at the region level as new or removed static objects. New objects are prevented from being incorporated into the long-term background model, while the regions of the empty scene being uncovered upon the removal of static objects not. Furthermore, by propagating the model of the empty scene into the short-term background model upon the removal of temporarily new static objects, foreground segmentation results are considerably improved.

The proposed system has been thoroughly evaluated, both from the static object detection perspective and from the foreground segmentation perspective, showing considerable improvements in both tasks.

The focus in this chapter has been set on coupling region analysis with a pixel-based background subtraction approach, not on the individual analysis tasks. Thereby, it has been shown that analysis tasks performed at different levels of abstraction can profit from each other by adequately designing the interaction between layers. The most intricate decision in this design process has been the use of a selective updating mechanism for the long-term background, which leads to holding in the foreground portions of uncovered background until they can be classified. In practice, it has been observed that such regions are usually reliably classified within a low number of frames, therefore, not significantly influencing the foreground segmentation results in a negative manner. Moreover, this circumstantial excess in foreground detections during the classification of uncovered background regions is clearly outbalanced by the rapid healing of the background upon the removal of long-standing new static objects.

In overall, the presented system is able to outperform state-of-the-art approaches for the detection of static objects. Thereby, neither a previous knowledge of the empty scene nor tracking information are needed. Furthermore, the foreground segmentation results provided by the system are ranked among the bests within the most recent state-of-the-art background subtraction approaches by considering the broad range of application scenarios contained in the recently proposed CDnet dataset.

6

Application Scenario: Video Indexing and Summarization

The rapid growth of video surveillance systems results in an increasing number of video feeds which should be watched and stored in a control room. This turns in a continuously growing workload for CCTV operators, who are overwhelmed by the huge sets of cameras. In a proactive crime prevention scenario, automatic video analysis techniques, aiming at understanding actions and human behaviors in video sequences, can be used in order to alert CCTV operators upon the occurrence of threatening situations. Besides that, video surveillance systems can also be used for crime investigation and offenders prosecution. Video indexing and summarization can be used in order to effectively accomplish this last tasks. Figure 6.1 depicts the role of automatic video analysis techniques in the described scenarios. The task of video indexing and summarization is used in this chapter as an exemplary application scenario of the video analysis tools presented in the previous chapters.

Video summarization is a process which aims at providing the viewer with an overview of the content of a video. For that purpose, it is necessary to find the relevant information contained in the video to be summarized (video indexing), and to develop a proper representation method which allows the user to rapidly grasp the extracted information and to navigate through it. Furthermore, as the user is directly driven to the critical points in time, the privacy of the people recorded at irrelevant passages of the video sequences is preserved.

Depending on the type of video content being analyzed, the techniques used for video summarization may differ. Basically, it can be distinguished between low-level features, object, or event based approaches. Event based approaches provide the highest semantic level. Never-

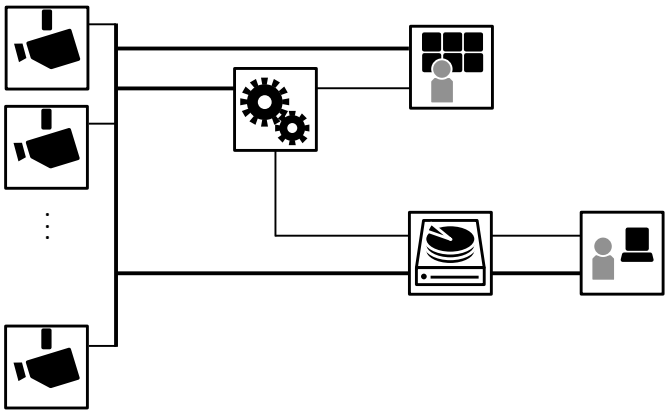


Figure 6.1: *Automated video surveillance scenario. Extracted content by means of video analysis is used for alerting control room operators in proactive crime prevention (top) and for video summarization for crime investigations (bottom).*

theless, the summaries provided based on this kind of information are very sensitive to the quality of the performed analysis. On the absence of event detections, either because the searched events do not happen in the considered video data or because of failure of the event detection algorithm, there is no basis for building up a summary. Furthermore, it is often the case that there is little information on a given event, which needs to be investigated. This requires the inspection of large hours of video data. In these cases, low-level features based approaches may be useful in driving the user to the potential points of interest.

One of the common issues, which is easy to observe in the state-of-the-art summarization approaches is that the information of interest is extracted by means of a unique level of analysis, i.e., either low-level feature extraction or mid-level object detection or high-level event detection. Therefore, the quality of the generated summaries is limited by the kind of the analysis tool used. In this chapter, a novel system that allows the combination of multiple cues of different kinds of analysis is presented. In this manner, the achievable quality of the information extracted out of the analyzed video sequences can be improved and the system is able to generate summaries that better align with the content of the original video. This system is demonstrated with the information provided by background subtraction as the low-level features extractor and the alarms produced by a static object detector as an example of event based features.

The rest of this chapter is structured as follows: Section 6.1 presents the main techniques for both indexing and representation, and reviews some representative approaches of the respective techniques. The properties of the presented systems and the requirements imposed by the selected application scenario motivate the proposed system, which is presented in

Section 6.2. In Section 6.3 experimental results are provided. Section 6.4 summarizes and concludes this chapter.

The content of this chapter has been partially published in *'Video Indexing and Summarization as a Tool for Privacy Protection'*, in the Proceedings of the IEEE International Conference on Digital Signal Processing, 2013 [Heras Evangelio et al., 2013b], and in *'Multiple Cue Indexing and Summarization of Surveillance Video'*, in the Proceedings of the 10th IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2013 [Heras Evangelio et al., 2013a]

6.1 Problem Statement and State-of-the-art

Video content has several features, ranging from the colors captured at the individual pixels, over the objects depicted at the successive video frames, to the motion described by the camera capturing the sequence. Moreover, the content of the video sequences is very broad as well, ranging from movies, over news programs, to surveillance videos.

In [Xiong et al., 2005], the authors make a distinction between *scripted* and *unscripted* video content. With *scripted* content is meant content which is structured as a series of semantic units as in the case of movies or news. On the contrary, *unscripted* content refers to this type of content which does not follow a predefined structure as in the case of surveillance or sport videos. Depending on the type of content, the techniques employed to extract the semantic information are different; while identifying the changes of scene might be sufficient in order to summarize a news program, this method would not be enough for summarizing a movie and even would fail to summarize a surveillance video sequence. In the case of scripted video content, segmenting the content can be considered a common step towards summarization. Equivalently, the extraction of highlights or relevant information can be considered the common approach for the case of unscripted content. The process of extracting the relevant information is denoted as indexing. Obviously, an index can point both to a space-time as well as to a space-lapse-of-time position within the indexed video data.

Once the relevant information has been extracted, the next step towards summarization is to structure and represent the extracted information so as to facilitate the access of the user to the content in a comfortable and efficient manner. Depending on the type of content and the application in mind, different kinds of information representation may be more appropriate than others. Regarding the access to the information, the authors in [Xiong et al., 2005] make the distinction between accessing the data with the intention of getting an idea of the information contained in it, *browsing*, or with the intention of looking for specific topics, *retrieving*. To clarify this difference, they use the analogy of consulting in a book the Table-of-Contents for *browsing* and the keywords-based Index for *retrieving*. While the content to be analyzed in the surveillance context can be considered to be of unscripted type as a whole, the

access to the content is required both in a browsing as well as on a retrieving fashion. Browsing is required in order to scrutinize unsupervised recorded video data in a preventive manner. Retrieving is required for carrying out criminal investigations.

After briefly presenting the most common techniques for indexing an representing surveillance video information, this section provides a review of some representative state-of-the-art summarization approaches.

6.1.1 Video Indexing and Summarization Techniques

The work presented here is focused on the analysis and representation of surveillance video content. Therefore, it presents techniques employed in order to extract information out of unstructured video content and to represent it to a user who is potentially carrying out a criminal investigation or needs to rapidly obtain an overview of a certain period of time with a preventive intention.

It can be distinguished between three different levels of analysis for the extraction of the relevant information:

- **Feature** based approaches compute some kind of scoring value based on low-level features as, e.g., number of foreground pixels or frame difference energy, in order to index those frames (or groups of frames) which are supposed to contain the higher amount of information.
- **Object** based approaches look for application-dependent objects of interest as, e.g., persons or cars, and index the frames containing this information.
- **Event** based approaches look for specific events as, e.g., pedestrians crossing the street from left to right or mugging situations, in order to set pointers with a high semantic level.

Event based approaches offer the highest semantic level at the cost of a higher sensitivity to the underlying analysis technique. Therefore, event based approaches are usually more application specific. The more specific the extracted semantic, the more specific the application domain. On the other hand, feature based approaches tend to be more application independent but, in the simplest case, they can only differentiate between segments of activity and segments of inactivity. However, they are a useful tool in order to segment long video sequences.

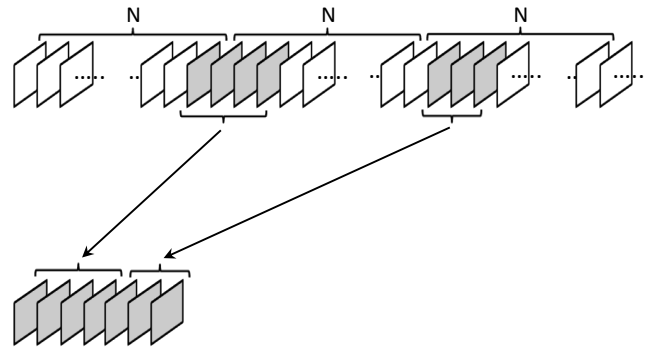
Regarding the representation, three different levels of abstraction can be observed:

- **Key frame** based representation relies on the selection of specially relevant frames to depict the content of the whole sequence.

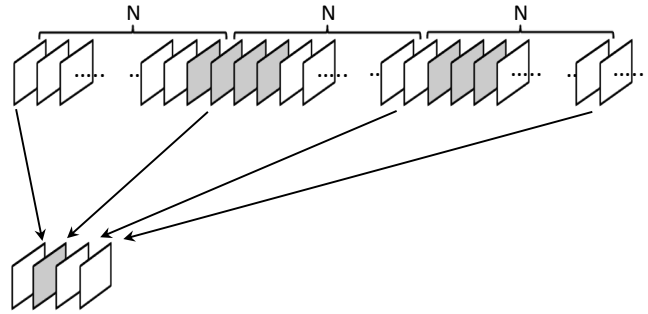
- **Frame-true time compressed video** techniques provide shortened or accelerated versions of the most relevant segments of the whole sequence by selecting a set of frames from the original sequence. Representative for this type of techniques are video editing [Smith and Kanade, 1995], fast forwarding, and adaptive fast forwarding [Petrovic et al., 2005]. Video editing techniques consist in gluing together the parts of a video sequence containing the most relevant information. Fast forward approaches depict only 1 frame out of every group of N frames, therefore, providing an accelerated version of the original video sequence. A more elaborated version of this last approach is adaptive fast forwarding, consisting in increasing the reproduction speed in less interesting parts of the video while slowing down in the parts of interest. Although the mentioned representation techniques were originally formulated for the multimedia domain, they can be applied as well in the surveillance domain. Figure 6.2 depicts an exemplary frame selection schedule for these three techniques.

- **Frame-free time compressed video** techniques aim at shortening video sequences by eliminating periods of inactivity and, furthermore, by displacing space segments in time so as to present more information at every frame. That means, that some objects may be displaced in space and time and, therefore, represented in other frames than those where they appeared in the original sequence. In this case, the relative timing between activities may change. Examples for this kind of techniques are dynamic video synopsis [Rav-Acha et al., 2006], which condenses video sequences by simultaneously showing several actions even if they occurred at different times, and video condensation by ribbon carving [Li et al., 2009], where the temporal warping is explicitly controlled so as to permit avoiding a reversal display order of the activities. Figure 6.3 depicts an exemplary top view of the space-time trajectories found in a sequence and their corresponding space-time assignment by dynamic video synopsis and video condensation by ribbon carving.

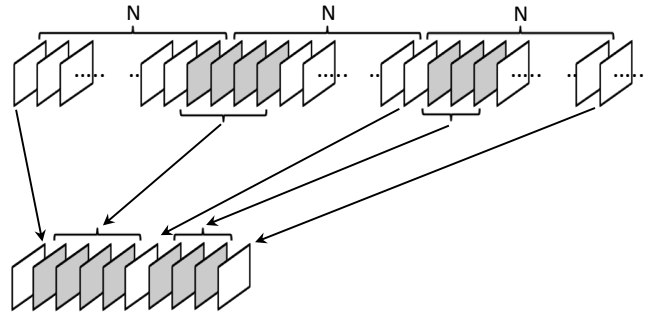
Key frame based representation techniques allow for the most condensed form of information representation, but contextual information is lost. Therefore, key frames are often used to provide non-linear access to the segments of video that they represent. Generally, the higher the abstraction, the higher the loss of information. Table 6.1 provides an overview of the capabilities that the three considered representation techniques allow considering the overall usability of the system. Four evaluation criteria have been considered. 'Information Compactness' refers to the number of frames needed to depict the content of the whole video sequence. 'Context Representation' is the capability of the system to depict the context surrounding the represented video content. 'Information Access Flexibility' is the flexibility that the system provides to the user in order to access specific pieces of the whole video sequences. 'Indexation Failure Resilience' refers to the capability of the representation system to provide informative summaries in the case that the quality of the generated indexes decreases.



(a) Video Editing.

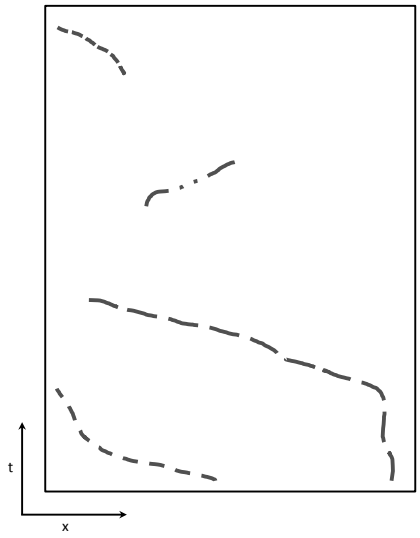


(b) Fast Forwarding.

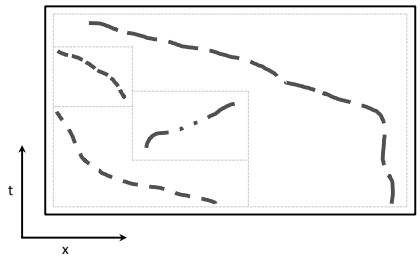


(c) Adaptive Fast Forwarding.

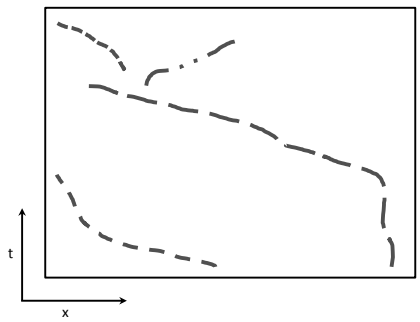
Figure 6.2: *Frame selection schedule in frame-true video representation. Grey and white are the frames with and without relevant content, respectively.*



(a) *Top view of space-time object trajectories.*



(b) *Dynamic Video Synopsis.*



(c) *Video Condensation by Ribbon Carving.*

Figure 6.3: *Frame-free video representation techniques.*

Table 6.1: *Comparison of three levels of abstraction for the representation of information extracted from surveillance video sequences.*

	Information Compactness	Context Representation	Information Access Flexibility	Indexation Failure Resilience
Key Frames	High	Low	High	Low
True-Frame Time Compression	Medium	High	Medium	Medium
Frame-Free Time Compression	High	Medium (might be confusing)	Low	Low

Depending on the level of the performed video analysis and on the application domain, some representation techniques are more appropriate than others. For instance, while low-level features can be successfully used to detect segments of activity of which a set of key frames can be selected for representation, these same features could not be employed for a frame-free time compressed video representation. On the other hand, while a compact representation as the one provided by frame-free representations can be of interest in order to provide a fast overview of the set of objects observed at a given location, such a representation would not be advisable for a crime investigation, where the context and objects interrelations are of crucial importance.

6.1.2 State-of-the-Art Summarization Approaches

A quite straightforward summarization approach can be found in [Damnjanovic et al., 2008], where the energy of the difference between consecutive frames in a video sequence is used for indexing. Thereby, it is assumed that events of interest are associated with a higher energy. Furthermore, the authors propose to use a normalized cut clustering criterion on the similarity matrix between the frames selected by the energy criterion to select frames for a key frame video representation and to build clusters of frames for a video editing based representation.

An approach based on the detection of a set of objects of interest is presented in [Cullen et al., 2012]. In this particular case, boats, cars and people, are taken as input for a video condensation algorithm able to remove inactive space-time regions by means of ribbon carving as proposed in [Li et al., 2009].

In [Li et al., 2007], an event based adaptive fast forwarding summarization approach is presented. Frames depicting the defined event of interest, which are triggered by the detection of motion with a certain speed and direction in predefined regions of interest, are played at normal speed and the rest of the frames are played accelerated.

A different approach which also provides adaptive fast forwarding has also been presented in [Höferlin et al., 2011]. In this paper, the authors propose to adapt the speed of the video

data to the temporal information contained in them. To that aim, they compute the temporal information between consecutive video frames by means of the divergence between the distribution of the absolute frame difference and the distribution of the estimated noise. As the authors observe in their paper, information based adaptive video fast-forward is not capable of pointing out relevant events on its own. Furthermore, the proposed information measure is sensitive to the absolute frame distance. Therefore, the benefit of the approach is marginal in crowded scenes.

An example of a frame-free video representation approach is presented in [Rav-Acha et al., 2006], where the authors formulate the video synopsis task as an energy minimization problem. They present two approaches. The first one uses a 3D Markov random field, where each node corresponds to a pixel in the 3D volume of the generated synopsis. The second, consists of first detecting moving objects and then performing the minimization on the detected objects. This second approach has the advantage of being much faster. An improvement for the video synopsis approach is presented in [Pritch et al., 2009], where the authors propose to cluster activities, and to display together only similar activities.

Another frame-free video representation approach which explicitly controls the temporal warping is presented in [Li et al., 2009]. To that aim, they introduce the concept of a ribbon in the space-time video volume, which allows by means of a flex-parameter to find a trade-off between the condensation ratio and the anachronism of the displayed events.

In [Ji et al., 2010] an approach based on the depiction of the detected moving objects along with their trajectories is presented. Video sequences are first segmented based on the difference in the number of foreground pixels detected in equally time-separated frames (a time difference of 10 frames is taken), therefore providing indexes corresponding to the entrance and exit of objects in the scene. The last frame of each segment is taken for video representation. The computed object trajectories are depicted in this frame. Furthermore, the authors propose to synthesize key frames of a video summary in order to provide even more compact representations of a video sequence. The problem of this approach is that it does not scale good, since in crowded scenarios it is difficult to set the limits of the segments. Moreover, the more the detected moving objects in the scene, the less visible are the depicted trajectories.

An object-based video summarization approach for multi-camera networks is presented in [Porikli, 2004]. Its aim is to change the camera-oriented videos paradigm into an object-oriented structure so as to allow to respond to semantic queries such as the places where a given object was recorded during a certain period of time. Video representation is provided in form of key frames, which are selected by minimizing the Semi-Hausdorff distance between the selected set of frames and the set of frames contained in the generated object-specific sequences.

In [Babaguchi et al., 2002], a system is proposed to summarize video captured by an omnidirectional surveillance camera by means of event based spatio-temporal indexing. The system displays the contents by using a timeline and a spatial map. Furthermore, video summarization can be provided in form of videos depicting the perspective or panoramic projections of the captured video at the times when events of interest were detected. The rest of the video material is cut-off. The reproduction speed of the generated videos can be controlled by the user.

An interesting approach from a theoretical point of view which aims at finding the optimal summary by formulating the problem as a rate-distortion optimization problem is presented in [Li et al., 2005]. The rate can be either the temporal or the bit rate, and the distortion is assumed to be introduced by missing frames and should be measured by an appropriate distortion metric. Nevertheless, as the authors show in the experimental evaluation, this summarization system would not be practicable in reality, since the computational load grows very fast. A formal computational complexity analysis is not provided, but the authors report 3 and 23 seconds to summarize 100 and 200 frame sequences, respectively.

6.1.3 Main Findings

It seems obvious that incorporating a tool for efficiently summarizing and providing access to the relevant information is of crucial importance for modern computer-aided surveillance systems, which continuously incorporate an increasing number of cameras. Furthermore, they bring the additional advantage of protecting the privacy of persons, as operators can be more directly pointed to the time spots of interest, therefore being able to skip the rest of the video data. In this sense, the more elaborated the semantic queries that the system is able to process, the higher the privacy protection.

In the ideal case, the user should be able to formulate queries based on events of interest. This means, that the system should be able to extract event information. Nevertheless, a problem of event based indexing and summarization systems is that the detection of the events of interest is mostly defined as a binary problem. This results in the lack of a basis for building up a summary in the case of the absence of event detections, whether because the considered events do not happen in the considered video data or because of failure of the algorithm.

Feature based approaches are more robust to the absence of specific events, but are only able to index points in time where relevant events might happen. Therefore, such approaches are more appropriate either for very restricted scenarios, where the detection of some video features provides a high certainty of the existence of an event, or for very generic scenarios, where the extraction of events is not feasible.

Object based approaches are appropriate for scenarios where the definition and identification of an object of interest is possible (as e.g. cars). Moreover, an approach aiming to provide

an object-oriented structure as the one presented in [Porikli, 2004], could be considered to have strong links to the privacy protection of individuals, as it provides the possibility of generating video summaries based on objects (as e.g. suspicious persons). In a more developed version, the identity of non-suspicious persons appearing in video segments where the followed person has been recorded could be hidden. Nevertheless, despite the huge challenge posed by multi-camera tracking, the question is how to choose the individuals of interest, since the generated queues are associated to individuals, but not to their actions. Furthermore, as the number of detected objects grows, the number of generated summary videos increases, deriving in the worst case scenario in a higher volume of video data as before the summarization process. Therefore, at the current level of development, this system can only be considered of theoretical interest.

There is a lack on experimentation on fusing several queues of content extraction. Especially promising is the combination of information of different nature as, e.g., low-level features with event detections. Collaborative approaches have been already proposed [Dumont et al., 2008]. Nevertheless, these are more oriented to entertainment applications and the content extraction techniques employed there are not applicable in the surveillance domain.

The suitability of a given video representation form depends on the application context. Generally speaking, frame-true approaches are more appropriate in scenarios where the relations between objects can be of relevance (as e.g. in security scenarios), whereas frame-free approaches may suit better the requirements of applications where the observation of specific objects is the center of interest, but interactions between the observed objects are not expected.

Regarding the protection of privacy, a very interesting study on the influence of the representation speed for the tasks of object identification and video recognition was presented in [Ding and Marchionini, 1997]. Among their results, the authors observed that an increased display speed has an earlier effect on the object identification than on the video comprehension task, i.e., the speed limit for successfully carrying out the task of object identification is lower than that for video comprehension. This can be explained by the fact that object identification and video comprehension correspond to different cognitive processes. While object identification requires focused attention, video comprehension implies global attention. This result could be considered as a motivation for using acceleration techniques for video summarization systems aware of privacy protection. Finally, having properly indexed the video content, different access rights can be provided to different kinds of users in order to further protect the privacy of the individuals being depicted in the recorded video material.

6.2 Multiple Cue Video Indexing and Summarization

One of the common issues, which is easy to observe in the state-of-the-art summarization approaches is that the information of interest is extracted by means of a unique level of analysis, i.e., either low-level feature extraction or mid-level object detection/classification or high-level event detection. Therefore, the quality of the generated summaries is limited by the kind of the analysis tool used. In this section, a system is presented that combines the results obtained by multiple cues of different levels of analysis. Thus, the overall quality of the information extracted out of the analyzed video sequences is improved and the system is able to generate summaries that better align with the content of the original video.

The proposed system provides indexes and summaries for security video investigations. Thereby, it is important to preserve the context surrounding the objects and events of interest in the generated summaries. Furthermore, the system should be easy to operate by a non-expert user and provide a flexible and rapid access to the gathered information. To that aim, both non-linear access to the segments of the input videos containing events of interest, and accelerated versions of the original videos, whose speed is adapted to their content, is provided. Video segments containing few relevant information are displayed at a high speed, while those with important content at a lower. The speed of the generated videos is computed by combining multiple video analysis cues. Figure 6.4 provides an overview of the proposed system. The input video is analyzed by several kinds of analysis tools which, respectively, generate an index and compute an associated speed according to their detections. The speed v_t of the generated summary at time t is computed as the minimum of the set of speeds $V_t = \{v_{c,t}\}_{c=1}^{C}$, where C is the number of cues used. t refers to discrete points in time associated to the consecutive frames of the analyzed video sequence and is, therefore, meant to be a member of the set of natural numbers excluding zero ($\mathbb{N}^+$). The generated indexes can be used both for providing non-linear access to the set of events detected, and for generating additional summaries according to different combinations of analyses and their respective computed associated speed.

In the following, the input video is assumed to have been recorded by fixed cameras. Furthermore, the proposed system is demonstrated by combining two video analysis cues: one provided by a dynamic foreground analyzer and the other by a new static objects detector. The dynamic foreground analyzer computes an associated speed $v_{f,t}$ based on low-level features extracted by means of background subtraction. The events triggered by the new static objects detector are used in order to compute an associated speed $v_{s,t}$ on an event basis. Therefore, the system combines two different levels of video analysis. Furthermore, the maximum speed of the generated summary is limited by v_{max}. The speed of the generated video is computed as:

$$v_t = min\{v_{f,t}, v_{s,t}, v_{max}\}. \tag{6.1}$$

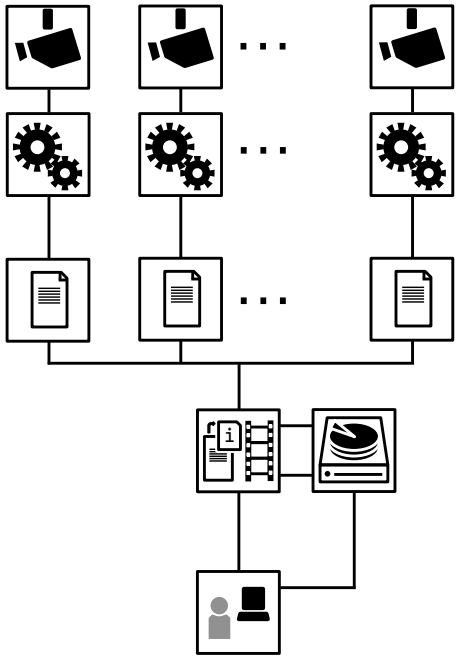


Figure 6.4: *Overview of the proposed summarization system.*

6.2.1 Low-level Features Analysis

The dynamic foreground analyzer takes for every frame the foreground mask corresponding to the input frame and computes an associated speed $v_{f,t}$ based on the absolute difference of the portion of foreground pixels $F_{diff,t}$ between consecutive frames.

This cue is used in order to rapidly direct the user to those parts of the video where the dynamics of the scene change. Thereby, it is assumed that dynamics changes are more relevant from a summarization point of view than the amount of foreground pixels itself. This can be intuitively illustrated by using the example of a crowded commercial street, where there is a large amount of moving objects, which, nevertheless, do not reveal any relevant information for a summarization system. On the contrary, the entrance of a single moving object into an empty scene can be considered as relevant. Figure 6.5 depicts graphically the analysis of the foreground masks obtained for a sequence reproducing this last example. In the analyzed sequence, a person enters an empty room at frame number 900, remains staying in the foreground for a while and then leaves the room again at frame number 4200. It is easy to see that the profile obtained by considering the difference of the portion of foreground (bottom) can be used to efficiently bring the user to the events of entering and leaving the room, while conveniently accelerating the rest of the sequence.

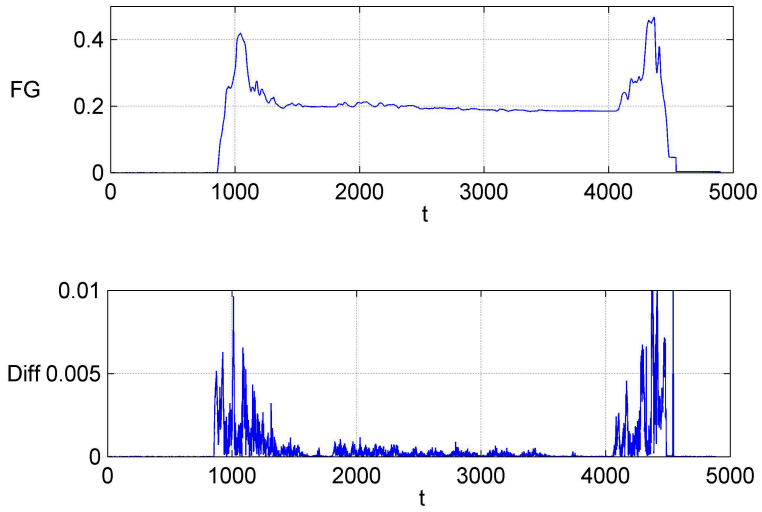


Figure 6.5: *Analysis of the foreground masks for an exemplary sequence. Top: Foreground portion. Bottom: Difference of foreground portion.*

The foreground masks are obtained by means of background subtraction. To that aim, the Gaussian mixture model developed in Chapter 3, has been used. For every frame, the amount of foreground pixels normalized to the size of the frame $\bar{F}_t$ is computed. The difference of this value $\bar{F}_{diff,t} = \bar{F}_t - \bar{F}_{t-1}$ is followed along the whole sequence. After processing each frame, a scaled version of $\bar{F}_{diff,t}$ is added to the score value D_t, which triggers a frame marker when $D_t > 1$. D_t is computed as:

$$D_t = \alpha D_{t-1} + \beta \bar{F}_{diff,t}, \tag{6.2}$$

where $\alpha \leq 1$ is a retaining factor and β is a weighting factor controlling the influence of the foreground difference into the speed of the summary. Upon the triggering of the frame marker, the value of D_t set to zero.

$v_{f,t}$ can then be easily computed as:

$$v_{f,t} = (t - t_d)\, v_i, \tag{6.3}$$

where t is the current point in time, t_d is the previous point in time in which a frame marker was triggered by the dynamic foreground analyzer and v_i is the speed of the input video.

In this way, the associated speed to the dynamic foreground analyzer gently adapts to the changes in the dynamic of the scene, associating high acceleration values to the segments of the sequence where the amount of foreground remains stable, while decreasing the acceleration for segments with high differences. By using the score value D_t the noise contained in the foreground masks is indeed filtered.

For every time t, the value of $\bar{F}_{diff,t}$ is logged into a file which can be used in order to generate alternative summaries of the analyzed video as shown later.

6.2.2 High-level Events

The second cue of the proposed system computes an associated speed $v_{s,t}$ based on the events triggered by a new static objects detector. To that aim, the system proposed in Chapter 5 has been used.

The detection of new static objects is a very important cue in safety and security applications as it advices for the presence of objects which might imperil the security of people in public spaces. Furthermore, by analyzing large archives of security video data, it could be observed that most of the events of interest were preceded by the occurrence of a new static object as, e.g., a car parked by the subjects committing an offense. Therefore, the speed associated to the static objects detector $v_{s,t}$ is set to a low value for a given number of frames N upon the detection of new static objects, and set to a high value otherwise.

$$v_{s,t} = \begin{cases} a_{s,l} v_i, & \text{for } t_e \leq t < t_e + N,\ \forall e \in \{1 \dots E\}, \\ a_{s,h} v_i, & \text{otherwise,} \end{cases} \tag{6.4}$$

where $\{a_{s,l}, a_{s,h}\} \in \mathbb{N}^+$ are the low and high acceleration factors, respectively, t_e is the time of detection of the event e, and E is the total number of events detected.

The speed associated to the static objects detector $v_{s,t}$ on the event of removal of the detected new static objects is also computed as per Equation 6.4.

Furthermore, the events raised by the occurrence of new static objects are logged into a file containing the number of frame of the detection and the bounding box associated to the object. This log-file can be used in order to provide non-linear access to the segments of the video where the new static objects appear and to generate alternative summaries.

6.2.3 Further Analysis Cues

Further analysis cues can be easily added to the proposed system by properly defining the associated speed of the video output depending on the performed analysis and feeding this value into the output speed computation as in Eq. 6.1. For practical reasons (see Section 6.2.4), the speed associated to each of the analysis cues must result from the multiplication of the input video speed with a natural number other than zero.

6.2.4 Summary Generation

The information gathered by the proposed system is provided to the user by means of two kinds of representation: a list of the detected events, which provides non-linear access to the segments of the video containing events of interest, and adaptively accelerated versions of the input videos.

The list of the events of interest (index) is generated by fusing the log-files generated by the individual analysis cues. This list can be visualized in text or in image form, by using the frame at which the event was first detected. Furthermore, the user can filter events by type or time of occurrence.

The accelerated versions of the input video are generated by using the speeds associated to each of the analysis cues. For each time t, the current output speed is computed as in Eq. 6.1. This speed is a multiple number of the input video speed, with an acceleration factor $a_t = v_{max}/v_i$, being $a_t \in \mathbb{N}^+$. The summary video generator keeps a register of the point in time corresponding to the last frame recorded t_l. If the difference between time corresponding to the current input frame t and t_l is bigger or equal than a_t, the current frame is recorded into the summary video. If not, it is skipped. Figure 6.6 depicts graphically the described procedure.

Although the indexation and the summary video generation have been described separately, these processes can be run together. In fact, the described system has been implemented for on-line generation of indexes and video summaries immediately afterwards of the video analysis process with negligible processing time for the indexing and summary video generation tasks.

Furthermore, by decoupling the tasks of indexing from the summary video generation, custom summaries can be easily generated in order to better fit individual user preferences.

6.3 Experimental Results

The proposed system has been tested using an extensive set of surveillance sequences comprising both public and private datasets. From the i-Lids dataset for AVSS 2007, the abandoned

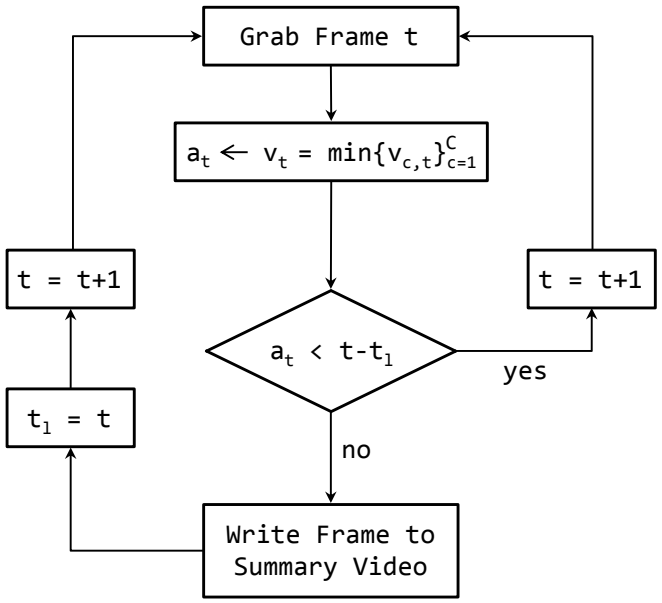


Figure 6.6: *Summary video generation.*

baggage scenario has been used, which consists of three video sequences recorded in a subway station where a piece of baggage is abandoned. Furthermore, several subways arrive and depart from the station, occasionally producing increased flows of passengers on the platform. From the CDnet dataset [Goyette et al., 2012], the sequences 'library', 'office' and 'tramstop' have been used. The two first sequences depict scenes in which a person enters an empty room, remains for a while, and then leaves the room. The sequence 'tramstop' depicts a more intricate situation involving the departure of a tram from a stop position and the abandonment of a box on a sidewalk. Table 6.2 summarizes the most important events of the described sequences and the approximated frame number of their occurrence. The private sequences depict hours of surveillance video recorded in outdoor environments. The scenes show most of the time people walking and cars driving through. The most relevant events are cars parking in an out and a very reduced set of events as mugging and a housebreak. A thorough description of the datasets used throughout this work is provided in Appendix A, Description of Datasets.

The system has been configured with the same parameters for all test sequences. The background subtraction system and the static objects detection have been configured with default parameters. The dynamic foreground analyzer has been configured with a retaining factor α equal to one and a weighting factor β equal to 25. That means, the difference score D_t is used as a pure accumulator. For the cue associated to the new static objects detector the low acceleration factor $a_{s,l}$ has been set to one and the high acceleration factor $a_{s,h}$ to 32. The

Sequence	Frame Nr.	Event Description
AB-Easy	2600	a piece of baggage is abandoned
	4600	abandoned baggage removal
AB-Medium	2250	a piece of baggage is abandoned
	4290	abandoned baggage removal
AB-Hard	2300	a piece of baggage is abandoned
	4450	abandoned baggage removal
library	900	a person enters an empty room
	4200	the person leaves the room
office	600	a person enters an empty office
	2000	the person leaves the office
tramstop	1000	tram starts moving
	1270	tram leaves scene
	1400	an object is abandoned

Table 6.2: *Main events of the summarized sequences.*

overall system maximum speed, v_{max}, has also been set to 32. This speed was chosen empirically aiming at achieving a good compromise between the compactness of the generated summaries and the comfort of the users visualizing them. However, these parameters can be easily adapted in order to fit the preferences of a particular user.

Figure 6.7 depicts the analysis results for the summarization of the sequences corresponding to the public datasets. The blue vertical lines correspond to the frames of the input video that were recorded in the summary video. Therefore, segments of time with a high density of blue lines correspond to low accelerated parts of the summary video, while segments with a low density correspond to high accelerated ones. For the sake of depiction clarity, only the frames triggered by the foreground analyzer and by the new static objects detector were depicted, but not those by v_{max}. The green vertical lines correspond to the detected events. The red curve represents the portion of foreground pixels that were detected for each input frame. It can be observed that non-complex scenes, which are indeed very usual in the security domain, as the 'library' and the 'office' sequences can be very accurately segmented by means of low-level features. In fact, the deceleration of the generated summary at the events of a person entering the room was possible based on the information provided by this analysis cue. On the other hand, more involved sequences as the i-Lids sequences and the CDnet 'tramstop' needed the information provided by the new static objects detector cue in order to decelerate the video at the segments containing the events of interest. Furthermore, it can also be observed that sequences with a higher level of semantic information show higher differences in the amount of foreground and are, therefore, summarized with a higher number of frames. In overall, it can be said that the combination of several analysis cues increases the alignment of the generated video summaries with the content of the original video.

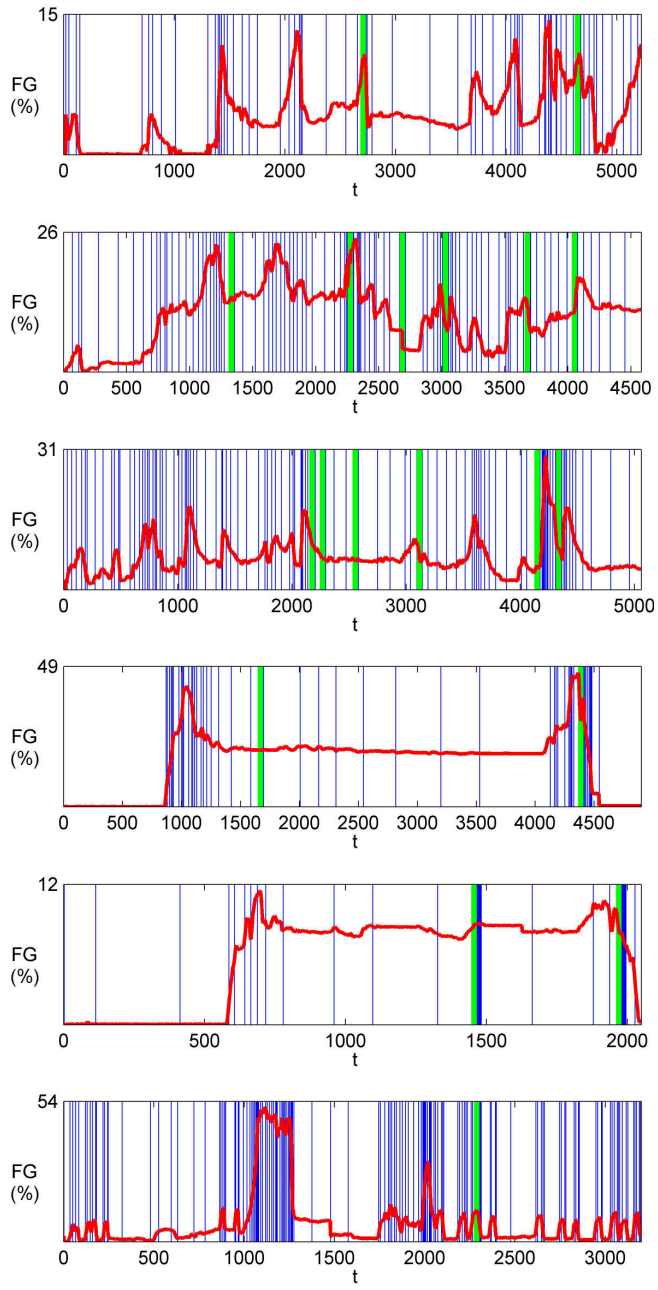


Figure 6.7: *Analysis results for the summarization of six video sequences. From top to bottom: iLids AVSS2007 AB-Easy, iLids AB-Medium, iLids AB-Hard, CDnet library, CDnet office and CDnet tramstop. Blue: frames to be added to the video summary. Green: detected events. Red: percentage of pixels classified as foreground.*

Sequence	Compression Rate		
	standard	no v_{max}	only events
AB-Easy	23.4170	43.8824	96.7037
AB-Medium	14.9251	18.4758	29
AB-Hard	15.2840	19.3092	32.0190
library	21.6770	46.6571	90.7222
office	17.6638	28.8592	37.9444
tramstop	15.0896	18.4913	114.2500
priv-01	17.5185	26.7819	37.7566
priv-02	14.1031	17.9504	25.2680
priv-03	19.0496	26.4841	67.4032
priv-04	11.6267	14.4018	17.7608

Table 6.3: *Compression rate of the generated summary videos for the test sequences by using three different configurations.*

A very useful functionality of the proposed system is that, due to the explicit decoupling of the indexing and summarization tasks, customized summary videos can be easily generated and displayed. In fact, based on the logs generated by the individual video analysis tools, the reproduction speed of the analyzed videos can be computed on-line. Furthermore, the user can preview the amount of time needed for watching a summary generated by a given configuration. In this way, the more appropriate configuration can be chosen, depending on the length of the video and the time availability of the video operator. Table 6.3 shows the compression rates, computed as the number of frames in the generated summary video divided by the number of frames in the input video, achieved for the whole set of sequences by using different summarization configurations. The configurations used are 'standard', which is the one explained in this chapter, 'no v_{max}', which corresponds to the speed computed by both analysis cues without using an upper limit, and 'only events', which corresponds to the summary video generated by using only the events cue. Sequences with a higher visual semantic content as, e.g., 'tramstop' achieve lower compression rates than sequences with lower content as, e.g., 'library'.

The proposed system has been preliminarily subjectively evaluated in the context of the MOSAIC project. In the evaluation session, several video analysis and data mining tools were demonstrated to a group of experts compound of three Intelligence Analysts, one CCTV Operator, and three higher ranking Police Officers (Chief Inspector, Detective Chief Inspector and Detective Chief Superintendent). An exemplary working scenario involving the whole set of analysis tools demonstrated was presented. The users were asked to evaluate each individual demonstrator through the completion of a User Evaluation Document, comprising their comments depending on how they felt about the performance of each of the demonstrated

tools and an individual rate in a scale ranging from 1 (poor performance) to 5 (excellent performance). The final rate for each tool was computed by averaging the scores provided by each user. In overall, a total amount of 12 analysis tools were demonstrated (4 network analysis + 8 video analysis tools). The evaluated video analysis tools were:

- Detection of loitering and unusual presence of persons at predefined locations
- Detection of unusual activities
- Detection of mugging
- Detection of illegally parked cars
- Detection of persons carrying with large objects
- Detection of suspicious left-behind items
- Tracking
- Video indexing and summarization

For the demonstration of the proposed video summarization system, the summaries of three long CCTV sequences depicting the activity in three different outdoor locations (longer than one hour each) were generated by using the above mentioned configuration parameters, that means, using the foreground difference and static object detection input cues and an overall maximum speed of $32x$. Some of the seven experts had already seen the original sequences and some not. In this way, it was possible to assess if the system is able to drive the user to the real points of interest in the recorded video material and if a user which have not seen a given sequence before is able to grasp the meaning of it by watching the summary. The proposed summarization tool was rated with the highest satisfaction score (3.86) among the whole set of tools presented. Moreover, the users commented that they were efficiently directed to the points of interest in the summarized video sequences and that such a tool would greatly assist for the review of items/offenses. In future subjective evaluation sessions, further aspects of the system including the maximum speed of the generated summaries and user interaction in the aim of generating user tailored summaries will be investigated.

6.4 Conclusions

This chapter has presented an exemplary application of the analysis techniques presented in previous chapters in the domain of video surveillance for safety and security: video indexing and summarization. To that aim, a thorough survey on existing video indexing and summarization techniques has been carried on. Thereby, the strengths and weaknesses that the presented techniques show for different surveillance scenarios have been identified. Out of

this study, a novel indexing and summarization system has been proposed. Therefore, beyond providing a mere exemplary application, this chapter constitutes on its own a contribution in the field of automated video-based surveillance.

The proposed system indexes the input video sequences attending to their content and generates video summaries which consist in adaptively accelerated versions of the input sequences. Passages with not relevant semantic content are played faster than those with relevant content. To that aim, the proposed system combines the results provided by multiple cues of different levels of video analysis. Therefore, the system accounts with a richer and more diverse amount of information, which is used to generate indexes and video summaries that better align with the content of the original video. While low-level video analysis provides the means for a coarse segmentation of the video sequences, high-level video analysis allows for an application dependent highlighting of events of interest. The proposed system is flexible and provides the capability of using an arbitrary number of analysis cues in a principled manner.

While state-of-the-art approaches usually consider the information extraction and summary generation as a closed unit, the approach developed in this chapter makes an explicit separation of the indexing and the visualization/summarization tasks. This provides the system with the ability of generating on-line customized summaries adapted to the user requirements, which is a huge advantage in safety and security scenarios, where the access to the recorded video data must provide both browsing and retrieving capabilities.

The system has been evaluated by using two analysis cues: a low-level analysis of the dynamics of foreground and the high-level events generated by a static object detector. The tests have been driven by using an extensive set of surveillance sequences, showing compression ratios ranging from 11 to 114, depending on the video content and on the configuration of the system.

7

Summary and Conclusions

This thesis has dealt with the detection of objects of arbitrary visual appearance in surveillance video data. In particular, the objects of interest were of two different natures: moving objects, which pass by through the observed scene, and static objects, which are added or removed from the scene. Moving objects should be provided to higher-level analysis layers for action and behavior recognition. Static objects should provide on-line alerts to human operators in real-time.

The absence of appearance models (and the unfeasibility to build them) and the immobility of the static objects has led to the use of background subtraction as the low-level processing tool. A thorough review of state-of-the-art background subtraction methods has been provided, thereby highlighting the main problems faced by this technique and how these problems have been approached in the extensive literature.

Due to the real-time and hardware savings requirements, Gaussian Mixture Models (GMM) have been chosen as the underlying background model. A deep analysis of the GMMs applied to the visual surveillance domain pointed out two main improvement opportunities: avoiding the convergence of the model to singularities or local maxima, and autonomously finding a configuration which allows to better adapt to the characteristics of the observed scene. The convergence problem arises from the use of variants of the EM algorithm, which is the standard method used to fit finite mixture models to unknown distributions by means of the observed data. Since the EM algorithm is a greedy method, choosing good initialization parameters is of capital importance in order to converge to meaningful models. Although the convergence problem is known and some approaches have been proposed to avoid it, these approaches have been focused in the case of fitting finite mixture models to stationary distributions and,

therefore, cannot be applied to the video processing case, where the underlying distribution is non-stationary.

In this thesis, a method has been proposed which tackles the above mentioned problems by means of incorporating a novel variance controlling scheme, which aims to adaptively compute an appropriate value for the initialization of the variance parameter by the creation of new modes, and to control the variance of existing modes in order to avoid the degeneration of the model. After guessing a proper value for the initialization of the variance controlling value at system start, two model observers are used in order to update it so as to adapt to changes in the characteristics of the observed scene. The proposed method is light in computational terms and results in GMMs which provide more accurate segmentation results than state-of-the-art GMM-based approaches. The proposed method has been thoroughly evaluated in terms of convergence, processing time and segmentation results, showing a notable improvement over the state-of-the-art. The main advantages of the method have been observed by the evaluation of the results provided by the analysis of sequences with different characteristics (illumination, noise...), where the sensitivity of the EM algorithm to the initialization parameters is made evident, and in crowded environments, where the emergence of over-dominating modes is common in state-of-the-art approaches.

While the detection of moving objects can be successfully approached by means of background subtraction, stationary objects pose an additional problem which derives from the need of adaptation of the background model to the changes in the scene. In this thesis, this problem has been tackled by using two background models learning at different rates, which allow for the detection of new stationary foreground regions as those which have been incorporated into the short-term but not into the long-term background model.

In a first approach, the results provided by a dual background subtraction are used as input of a Finite State Machine (FSM), which is used to provide a multi-class pixel classification based on the history of the pixel. Compared to a plain dual background subtraction based system, the proposed approach has the advantage of not being dependent on the learning rate of the long-term background model in order to correctly classify pixels. This results in the ability of detecting new static objects even if they have been already absorbed by both background models and, furthermore, correctly classify the uncovered background areas upon their removal. Moreover, since the ability of the system to detect new static objects is not anymore dependent on the learning rate of the long-term background model, this can be freely adjusted so as to optimally adapt to the changes in the observed video sequence. It has been shown that a FSM is an efficient method for reasoning on the results provided by background subtraction. Furthermore, it has been observed that the knowledge gained by means of reasoning could be used to improve the segmentation results and to overcome the limitation imposed by the need of a perfect knowledge of the empty scene in dual background based systems.

Based on this observation, a further developed system has been proposed, which consists of two background models learning at different rates, a FSM for multi-class pixel classification, a region analysis layer to classify new static regions, and a feedback loop to integrate the results of region classification into the background model. The background models are updated in a complementary fashion. While the short-term background model rapidly adapts to every change in the observed scene, the long-term background model is updated in a selective fashion, so as to only incorporate into the background model changes which correspond to the empty scene. The results provided by the subtraction of these complementary background models are used as input of a FSM, which classifies pixels attending to four categories: background, moving, static foreground and uncovered background. Static foreground pixels are prevented from being integrated into the long-term background model and classified at the region level as new or removed static objects. These classification results are fed-back to the pixel level in order to incorporate the uncovered background regions into the background while holding new static objects in the foreground. The system is able to correctly detect new static objects without previous knowledge of the empty scene, and to rapidly recover the background model of the empty scene upon the removal of long-term static objects, therefore, improving segmentation results. The performance of this system has been evaluated regarding its ability to detect new static objects, and regarding the quality of the provided foreground masks, showing considerable improvements in both tasks. Currently, the proposed system is ranked among the best performing systems in the publicly available benchmark CDnet.

Finally, an exemplary application scenario has been defined which combines the information gathered at different levels of analysis in order to generate indexes and summaries of surveillance video sequences. This application has been used to demonstrate the results provided by the developed algorithms in a practical scenario. The combination of several analysis cues allows for the collection of a richer and more diverse amount of information about the analyzed video sequences. Therefore, the system is able to generate indexes and video summaries that better align with the content of the original video. Low-level video analysis provides the means for a coarse segmentation of the video sequences. High-level video analysis allows for an application dependent highlighting of events of interest. Further analysis cues can be added to the system in a principled manner.

In this thesis it has been shown that properly combining the knowledge gained at different levels of analysis can bring substantial benefits. In a bottom-up scheme, it has been shown that the combination of different levels of analysis can provide a system with a diversity which can be exploited in order to better understand the observed sequences. In a top-down approach, an architecture has been defined which divides the low-level modelling of the observed scene into two complementary parts. This has allowed for building a purely statistical model and a high-level driven model of the scene. Furthermore, by defining the high-level driven model by means of low-level semantics instead of application dependent requirements, the results provided by this analysis layer can be used by a wide range of high-level analysis layers. Finally,

the fact of accounting with two complementary models allows for a graceful recovering of eventual feedback errors.

It has to be remarked, that most of the efforts in the work presented here have been devoted to the low-level part of the system. Although the provided results at the region classification level are of high quality, further investigations on the classification of static foreground regions, and, specifically, the triggering of region classification, could further improve the provided results. In this line, more elaborated strategies could be defined for handling overlapping foreground regions, which, at the moment, are individually classified and, consequently, managed with respect to their integration in the high-level driven background model. Finally, the incorporation of heterogeneous region analysis tools of low computational cost could be considered in order to more reliably classify the static foreground regions.

Despite the attention which has been paid to the computational and hardware requirements of the developed algorithms, and obviating that the whole set of proposed algorithms are able to process typical surveillance video data in real-time, a minimal hardware and computational configuration has not been provided. Investigations in this direction could include the use of alternated updating schedules or the use of more restrictive management of layers. Finding such a minimal configuration would help in assessing the interest of porting the developed algorithms to the cameras in a distributed surveillance network with the goal of bringing the intelligence to the edge of the network.

A

Description of Datasets

A fundamental step in the development of algorithms is the evaluation of the obtained results. In the ideal case, this is done by using publicly available datasets in order to allow for the comparison of the results with those provided by other research groups.

An evaluation dataset should provide the possibility of evaluating the challenges posed to the studied algorithms in the relevant application scenarios. Furthermore, the existence of a common ground-truth is of crucial importance for the comparability of the results. The ground-truth consist in an annotation of the answer expected from the evaluated algorithms for a given input.

This chapter provides a detailed description of the datasets used in this thesis for the evaluation of the proposed algorithms. Furthermore, it provides pointers to further relevant datasets.

A.1 CDnet

The ChangeDetection.net (CDnet) video dataset was proposed for comparing the detection results of change detection algorithms in the IEEE Workshop on Change Detection, held in conjunction with the IEEE International Conference on Computer Vision (CVPR), 2012. The dataset consists of 31 surveillance videos divided into six categories covering most of the challenges regarding background subtraction for the task of video surveillance. The dataset is provided with a set of human-annotated multi-class pixel classification ground-truth consisting of foreground, background, shadow and shadow region boundary. Furthermore, a toolkit to compute the performance metrics used is provided, so as to enable a quantitative comparison of foreground segmentation algorithms.

The dataset is publicly available at www.changedetection.net, where a ranking of the evaluated methods is provided. This ranking is being continuously updated with the results provided by the authors which use the dataset to evaluate additional methods. Currently, a total of 32 methods (among them 27 competing in the whole set of categories) are ranked. In this section, a brief description of the six video categories and the videos contained in them is provided. A description of the evaluation methodology followed to rank the compared change detection approaches is provided in Section B.1.

Baseline

Four videos (two indoor plus two outdoor) representing a mixture of mild challenges typical of the next 4 categories. Figure A.1 shows an example frame of each video and the corresponding provided ground-truth.

Dynamic Background

Six videos in outdoor environments with strong background motion (two videos depict boats on rippling water, two videos show cars passing near to a fountain, and the other two depict pedestrians, cars and trucks passing by in scenes with trees in the background moving because of the wind). Figure A.2 shows an example frame of each video and the corresponding provided ground-truth.

Camera Jitter

Four videos (one indoor and three outdoor) captured by vibrating cameras. Figure A.3 shows an example frame of each video and the corresponding provided ground-truth.

Intermittent Object Motion

Six sequences (five outdoor and one indoor) depicting scenarios related to new and removed static objects which pose a special challenge to background bootstrapping, maintenance and healing. Figure A.3 shows an example frame of each video and the corresponding provided ground-truth.

Shadow

Four sequences (two indoor and four outdoor) with shadows casted by moving objects and elements of the background as trees and buildings. Figure A.5 shows an example frame of each video and the corresponding provided ground-truth.

Thermal

Five videos (three outdoor and two indoor) captured by far-infrared cameras containing typical thermal artifacts as heat stamps, heat reflection on doors and windows and camouflage effects,

when a moving object has a similar temperature as its surrounding regions. Figure A.6 shows an example frame of each video and the corresponding provided ground-truth.

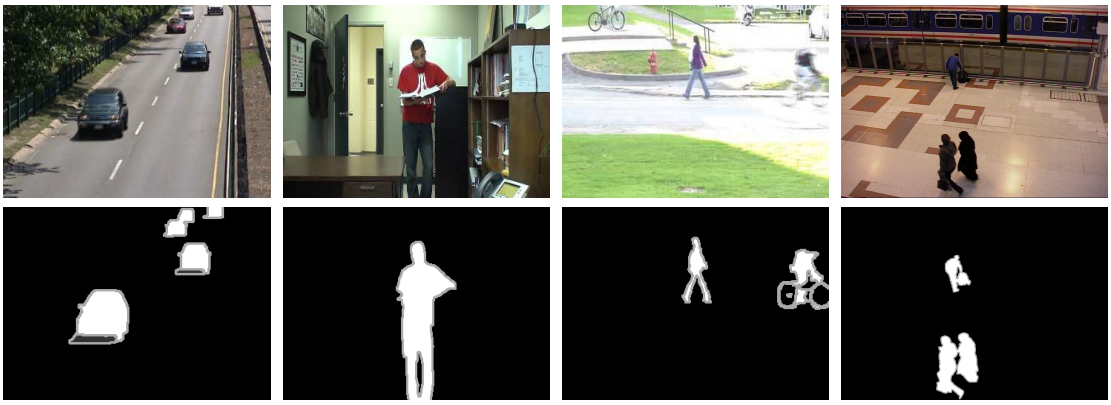


Figure A.1: *CDnet Baseline. From top to bottom: sample frame of the four video sequences and corresponding ground-truth. From left to right: 'highway', 'office', 'pedestrians' and 'PETS2006'. (Source,* www.changedetection.net*).*

Figure A.2: *CDnet Camera Jitter. From top to bottom: sample frame of the four video sequences and corresponding ground-truth. From left to right: 'badminton', 'boulevard', 'sidewalk' and 'traffic'. (Source,* www.changedetection.net*).*

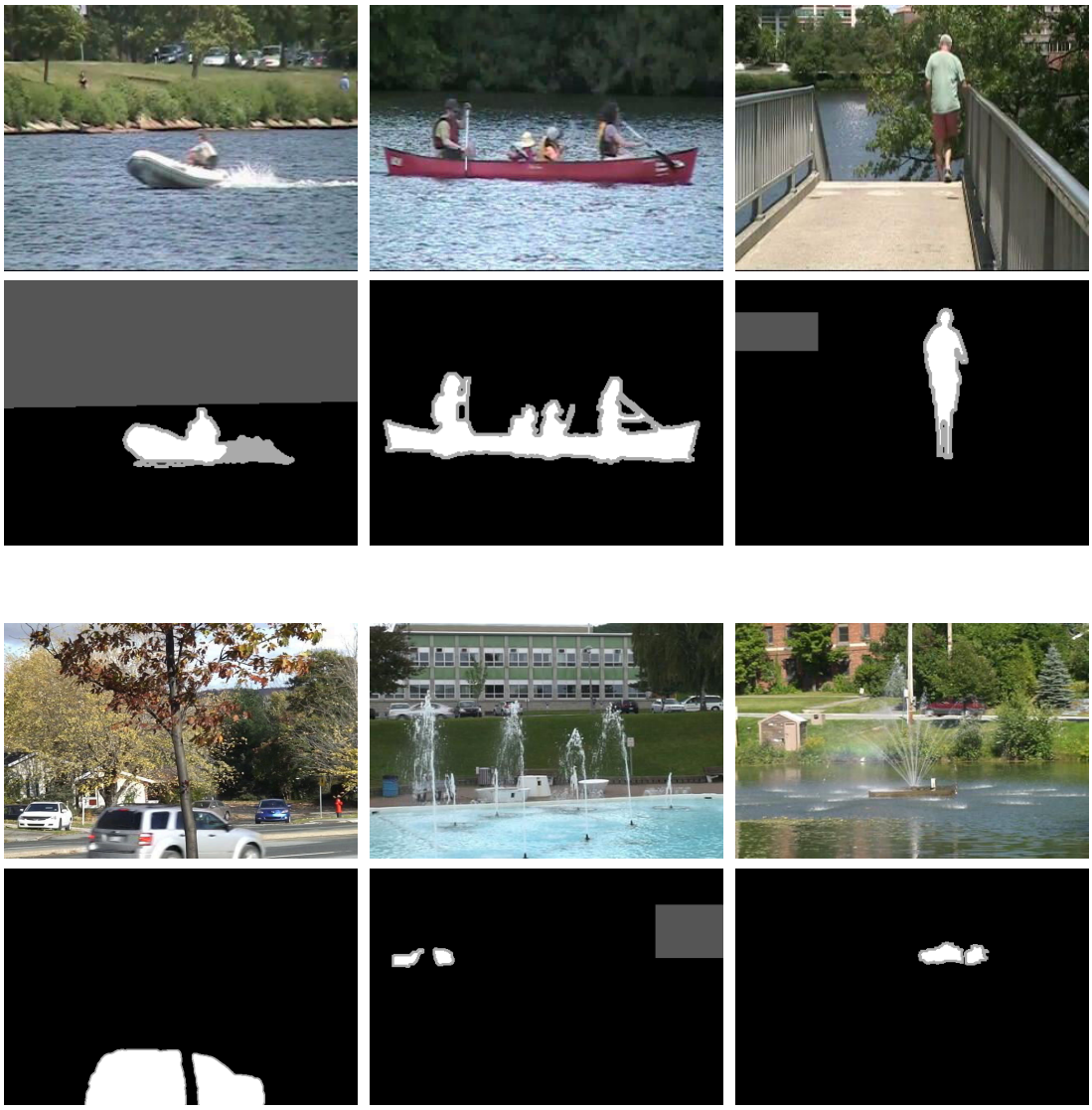


Figure A.3: *CDnet Dynamic Background. First row from left to right: sample frame of the video sequences 'boats', 'canoe' and 'overpass'. Second row: ground-truth corresponding to the frames in the first row. Third row from left to right: sample frame of the video sequences 'fall', 'fountain01' and 'fountain02'. Fourth row: ground-truth corresponding to the frames in the third row. (Source, www.changedetection.net).*

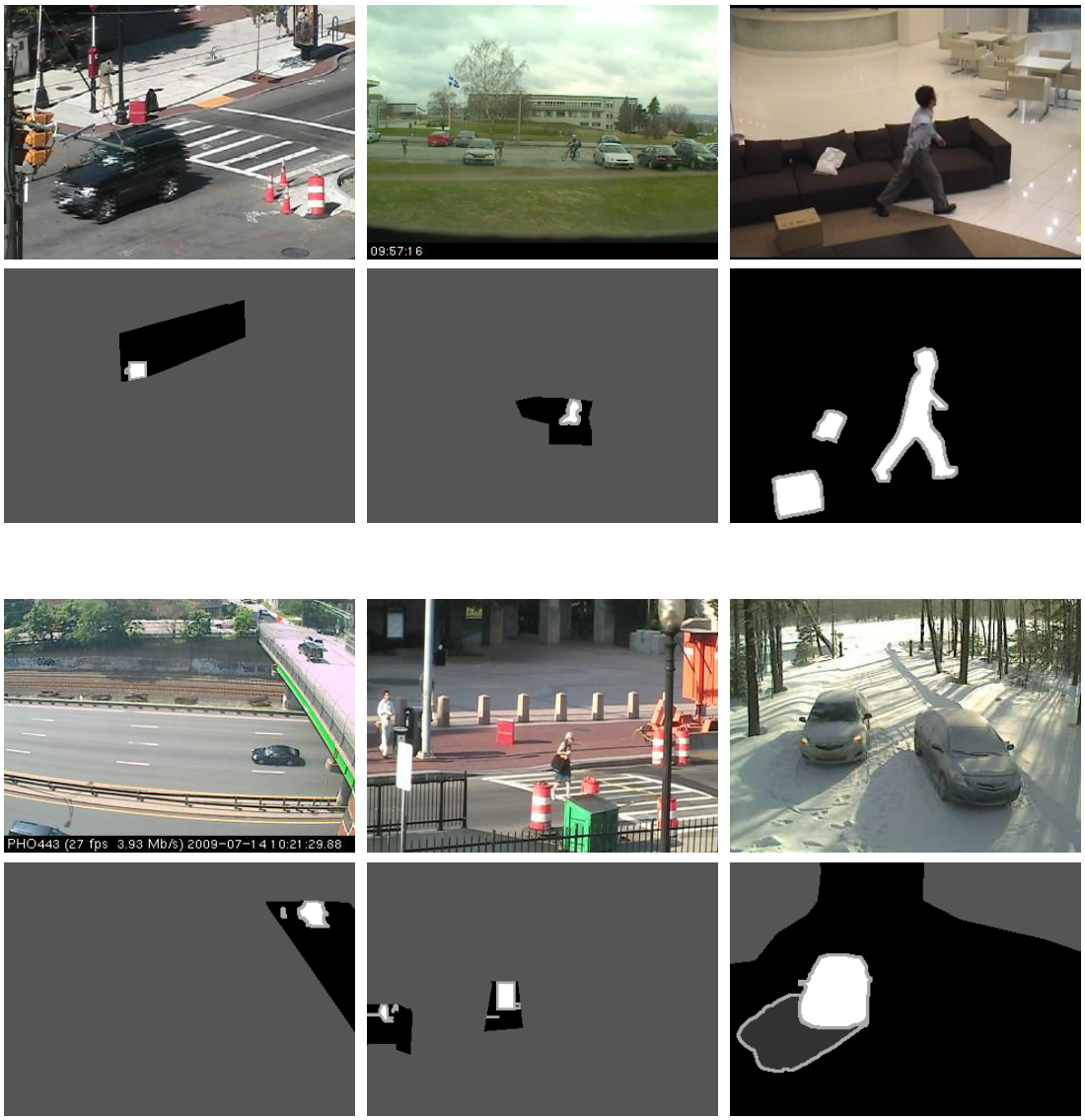


Figure A.4: *CDnet Intermittent Object Motion. First row from left to right: sample frame of the video sequences 'abandonedBox', 'parking' and 'sofa'. Second row: ground-truth corresponding to the frames in the first row. Third row from left to right: sample frame of the video sequences 'streetLight', 'tramstop' and 'winterDriveway'. Fourth row: ground-truth corresponding to the frames in the third row. (Source,* www.changedetection.net*).*

Figure A.5: *CDnet Shadow. First row from left to right: sample frame of the video sequences 'backdoor', 'bungalows' and 'busStation'. Second row: ground-truth corresponding to the frames in the first row. Third row from left to right: sample frame of the video sequences 'copyMachine', 'cubicle' and 'peopleInShade'. Fourth row: ground-truth corresponding to the frames in the third row. (Source,* www.changedetection.net*).*

Figure A.6: *CDnet Dynamic Background. First row from left to right: sample frame of the video sequences 'corridor', 'dinningRoom' and 'lakeSide'. Second row: ground-truth corresponding to the frames in the first row. Third row from left to right: sample frame of the video sequences 'library' and 'park'. Fourth row: ground-truth corresponding to the frames in the third row. (Source, www.changedetection.net).*

A.2 AVSS2007

This dataset is a subset of the i-LIDS dataset for event detection in CCTV footage. It was provided for the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) 2007 and can be publicly accessed on-line at http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html. The events of interest are abandoned baggage (Task 1) and parked vehicle (Task 2).

Task 1 consists of three sequences of increasing complexity depicting an underground station where some baggages are left unattended. Figure A.7 shows an example frame of each video sequence.

Figure A.7: *AVSS 2007. From left to right: sample frame of the video sequences 'AVSS AB Easy', 'AVSS AB Medium' and 'AVSS AB Hard'. (Source, http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html).*

A.3 PETS2006

This dataset consists of several multi-sensor sequences containing left-luggage scenarios with increasing complexity. It was provided publicly for the PETS 2006 workshop, in Conjunction with IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006 and can be publicly accessed on-line at http://www.cvg.rdg.ac.uk/PETS2006/data.html.

The whole dataset consists of seven sequences recorded from four different camera positions. Figure A.8 shows an example frame of each camera position. For the evaluation of the algorithms presented in this thesis, the camera 3 of sequence 1 has been used.

A.4 Caviar

This dataset consists of a set of video clips recorded for the evaluation of action recognition algorithms, including people walking alone, meeting with others, window shopping, entering and exiting shops, fighting and leaving a luggage abandoned in a public place.

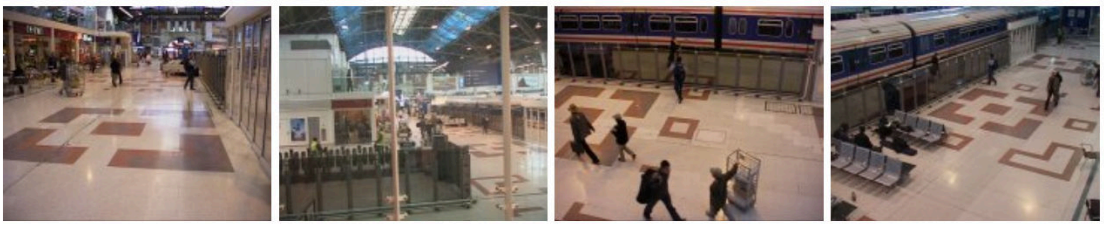


Figure A.8: *PETS 2006. From left to right: sample frame of the video sequence 1 taken from camera 1 to 4. (Source,* http://www.cvg.rdg.ac.uk/PETS2006/data.html*).*

The first set of the video sequences were filmed for the EC funded CAVIAR project with a wide angle camera lens in the entrance lobby of the INRIA Labs at Grenoble, France. The second set was also recorded using a wide angle lens in the hallway of a shopping center in Lisbon, and provides the sequences recorded from two different camera positions.

This dataset can be publicly accessed at http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. Figure A.9 shows an example frame of the sequence used in this thesis for the evaluation of the proposed algorithms.

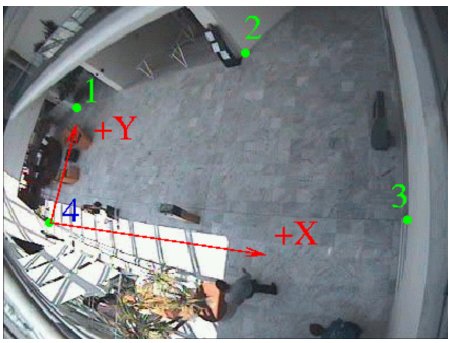


Figure A.9: *CAVIAR. Frame with calibration points for the sequences from INRIA (1st Set). (Source,* http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/*).*

A.5 Private

One of the problems encountered during the elaboration of the work presented here is the absence of long sequences, which are of crucial importance for the evaluation of the convergence of algorithms and for their evaluation in a long-term basis. In order to overcome this problem, private sequences were recorded at the courtyard in front of the EN-building of the

Technical University of Berlin. Figure A.10 shows three example frames of the TUB sequence *Winter*. At the beginning of the scene it is snowing and, therefore, measurements are very noisy; in the middle of the sequence it stops snowing and, therefore, the noise shrinks; at the ends of the sequence it gets darker.

Figure A.10: *Private. Winter sequence. From left to right: Beginning, middle and end of the sequence.*

Further private datasets, as e.g. the dataset containing the *Lobby* sequence, have been used for testing and presenting the results of the algorithms presented in this thesis. These datasets have been provided by the users of the projects for which the algorithms have been developed and cannot be made public because of the corresponding usage agreements.

A.6 Further Datasets

Apart from the datasets used in this thesis, there are a large number of datasets publicly available in the Internet. Some of them, related to the main topics handled in this thesis (foreground detection and static object detection), are listed in the following:

- **Wallflower**: The well-known Wallflower dataset consists of seven image sequences representing different problematic scenarios for background maintenance as identified in [Toyama et al., 1999]. For each sequence, only one hand-segmented image is provided as ground-truth. This dataset is publicly available at http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

- **SABS (Stuttgart Artificial Background Subtraction)**: The SABS dataset is an artificial dataset consisting of video sequences for nine different challenges of background subtraction in the context of video surveillance, and the corresponding "perfectly" labeled ground-truth. It allows for a pixel-wise evaluation of background subtraction approaches. The dataset and a matlab script for the evaluation of the foreground masks can be downloaded from http://www.vis.uni-stuttgart.de/en/research/information-visualisation-and-visual-analytics/visual-analytics-of-video-data/sabs.

html. Although the authors claim that the video sequences are realistic because of the use of ray-tracing techniques with global illumination, the quality of the footage is far from the video queues analyzed in typical surveillance scenarios. More information on this dataset and the evaluation methodology can be found in [Brutzer et al., 2011].

- **cVSG (Chroma Video Segmentation Ground Truth)**: In the same aim of allowing for an evaluation based on an accurate ground-truth, the cVSG provides a set of sequences composed of background and foreground objects which have been separately recorded and extracted by means of chroma techniques. As the authors of the dataset acknowledge in the website, although foreground and background were combined trying to obtain realistic sequences, realism was not always achieved. The dataset can be accessed for research purposes at http://www-vpu.eps.uam.es/DS/CVSG/. Further information is provided in [Tiburzi et al., 2008].

- **PETS2007**: Similarly to PETS2006, the PETS2007 dataset provides eight multi-camera sequences depicting three scenarios with increasing complexity: loitering, attended luggage removal (theft) and unattended luggage. Nevertheless, this dataset has not found a high echo in the surveillance community, maybe because of the high complexity of the sequences, which include high dense crowds, poor conditions for background learning and extreme lighting conditions.

- **Shadow Detection**: The Shadow Detection dataset, provided at the Autonomous Agents for On-Scene Networked Incident Management (ATON) project website http://cvrr.ucsd.edu/aton/shadow/index.html, consists of a set of five sequences with the associated ground-truth for one of them (the Intelligent Room sequence). It is aimed at evaluating shadow detection algorithms.

- **ASODds (Abandoned and Stolen Object Discrimination dataset)**: This dataset consists of a set of sequences extracted from public datasets aiming to provide a representative test-set for the evaluation of systems devoted to the detection of new and removed static objects, along with the manual annotation of the events of interest. The dataset is provided for research purposes at http://www-vpu.eps.uam.es/DS/ASODds/index.html.

Further datasets for the evaluation of background subtraction approaches can be found at https://sites.google.com/site/backgroundsubtraction/test-sequences. Further pointers to general computer vision datasets are provided in http://www.cvpapers.com/datasets.html and http://homepages.inf.ed.ac.uk/cgi/rbf/CVONLINE/entries.pl?TAG363.

B

Performance Metrics

Performance metrics are the basis for the evaluation and comparison of the ability of several algorithms to perform a certain task. Performance metrics should provide a fair and robust comparison of the evaluated algorithms.

The number of metrics proposed in the literature for the evaluation of background subtraction is large. In this section, the measures used in the CDnet dataset, which has been extensively used for the evaluation of the results provided by the algorithms presented in this thesis, are explained. Furthermore, it is explained how methods are ranked based on these measures. Further pointers to the literature on the topic of performance evaluation are provided at the end of this section.

The first step in quantitatively measuring the performance of a given algorithm is to compare the results provided by the evaluated method with the established ground-truth for each frame. This leads to the following measures:

True positives (TP): The number of detections which correspond to a detection in the ground-truth.

True negatives (TN): The number of non-detections which correspond to a non-detection in the ground-truth.

False positives (FP): The number of detections which correspond to a non-detection in the ground-truth.

False negatives (FN): The number of non-detections which correspond to a detection in the ground-truth (missed detections).

Out of these simple measures, more elaborated metrics are derived as follows:

Recall (Re), also called the true positive rate or sensitivity, measures the percentage of the positive class which is classified al such,

$$Re = \frac{TP}{TP + FN} \tag{B.1}$$

Specificity (Sp), also called the true negative rate, measures the percentage of the negative class which is classified al such,

$$Sp = \frac{TN}{TN + FP} \tag{B.2}$$

False Positive Rate (FPR), also called false alarm rate, measures the percentage of wrong positive classifications among the whole set of negative examples ($FPR = 1 - Sp$),

$$FPR = \frac{FP}{FP + TN} \tag{B.3}$$

False Negative Rate (FNR), also called missed detection rate, measures the percentage of wrong negative classifications among the whole set of positive examples ($FPR = 1 - Sp$),

$$FNR = \frac{FN}{TP + FN} \tag{B.4}$$

Percentage of Wrong Classifications (PWC), measures the percentage of wrong classifications,

$$PWC = 100\frac{FN + FP}{TP + FN + FP + TN} \tag{B.5}$$

Precision (Pr), measures the percentage of positive classifications which indeed belong to the positive class,

$$Pr = \frac{TP}{TP + FP} \tag{B.6}$$

F-Measure ($F1$), frequently used as a single measure of performance, is the harmonic mean of precision and recall:

$$F1 = \frac{1}{\alpha \cdot \frac{1}{Re} + (1 - \alpha) \cdot \frac{1}{Pr}} \tag{B.7}$$

where α is a parameter which should be selected according to the application scenario depending on the relative importance of precision and recall. If a high recall is required, α should be set low. On the contrary, if a high precision is required, α should be set high. In the case of giving equal importance to the precision and recall values, as is the case in the CDnet dataset, $\alpha = 0.5$ and, therefore:

$$F1 = 2\frac{Pr \cdot Re}{Pr + Re} \tag{B.8}$$

B.1 Performance Evaluation and Ranking

The ultimate aim of computing performance metrics is the evaluation and comparison of the considered algorithms. In the case of the CDnet challenge, this is made by means of a ranking, which is produced as explained in this section. The measures listed above (Re, Sp, FPR, FNR, PWC, Pr and $F1$) are computed for each video and averaged over each category. For example, the average recall $Re_{i,c}$ of a method i in a given category c is computed as:

$$Re_{i,c} = \frac{1}{|N_c|}\sum_{v=1}^{|N_c|} Re_{v,c} \tag{B.9}$$

where $|N_c|$ is the number of videos in the considered category c.

The overall metrics are computed by averaging over the metrics computed for each individual category. For example, the average overall recall Re_i of method i is computed as:

$$Re_i = \frac{1}{6}\sum_{c=1}^{6} Re_{i,c} \tag{B.10}$$

Ranking is done at the category level and across categories. The rank $RM_{i,c}$ of a method i in a given category c is computed as:

$$RM_{i,c} = \frac{1}{7}\sum_{m=1}^{7} rank_i(m,c) \tag{B.11}$$

where $rank_i(m,c)$ is the rank of method i for metric m in category c.

The average overall ranking across categories RC_i of a method i is computed by taking the average of its category rankings across all 6 categories:

$$RC_i = \frac{1}{6}\sum_{c=1}^{6} RM_{i,c} \tag{B.12}$$

B.2 Remarks

The evaluation of the results provided by the performance measures is usually application dependent. A common way to parameterize an algorithm for a given application is the use of graphical representations of the measured performance as e.g. the ROC curves or, in the case of absence of true negatives, as is the case of object detection algorithms, F-Measure based approches as the one presented in [Lazarevic-McManus et al., 2006].

Further performance metrics as well as performance evaluation methodologies for the assessment of object detection approaches based on background subtraction can be found in [Elhabian et al., 2008]. A thorough review of performance measures used in the computer vision can be found in [Goldmann, 2009].

Bibliography

Appiah, K. and Hunter, A. (2005). A single-chip fpga implementation of real-time adaptive background model. In *Proceedings of the IEEE International Conference on Field-Programmable Technology*, pages 95–102.

Azzari, P. and Bevilacqua, A. (2006). Joint spatial and tonal mosaic alignment for motion detection with ptz camera. In *Proceedings of the ICIAR*.

Babaguchi, N., Fujimoto, Y., Yamazawa, K., and Yokoya, N. (2002). A system for visualization and summarization of omnidirectional surveillance video. In *Proceedings of the 8th International Workshop on Multimedia Information Systems (MIS2002)*, pages 18–27, Tempe AZ.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, pages 1–31.

Baltieri, D., Vezzani, R., and Cucchiara, R. (2010). Fast background initialization with recursive hadamard transform. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 165–171.

Barnich, O. and Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709 –1724.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 12:43–77.

Basu, M. (2002). Gaussian-based edge-detection methods-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(3):252–260.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359.

Bayona, Á., SanMiguel, J. C., and Martínez, J. M. (2009). Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 25–30.

Beauchemin, S. S. and Barron, J. L. (1995). The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467.

Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., and Rosenberger, C. (2010). Comparative study of background subtraction algorithms. *J. Electronic Imaging*, 19.

Bevilacqua, A., Di Stefano, L., and Azzari, P. (2005). An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 511 – 516.

Bhat, K., Saptharishi, M., and Khosla, P. (2000). Motion detection and segmentation using image mosaics. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1577 –1580 vol.3.

Bilmes, J. A. (1998). A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, Berkeley CA.

Booth, T. L. (1967). *Sequential Machines and Automata Theory*. Wiley.

Borgefors, G. (1986). Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344–371.

Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *IEEE International Conference on Computer Vision*.

Bouttefroy, P. L. M., Bouzerdoum, A., Phung, S. L., and Beghdadi, A. (2010). On tne analysis of background subtraction techniques using gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Bouwmans, T., Baf, F. E., and Vachon, B. (2008). Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science*, 1:219–237.

Broida, T. J. and Chellappa, R. (1986). Estimation of object motion parameters from noisy images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(1):90–99.

Brown, L. G. (1992). A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376.

Brown, M. and Lowe, D. (2003). Recogniting panoramas. In *Proceedings of the International Conference on Computer Vision*.

Brutzer, S., Hoferlin, B., and Heidemann, G. (2011). Evaluation of Background Subtraction Techniques for Video Surveillance. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1937–1944, Colorado Spring, USA.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Caprile, B. and Torre, V. (1990). Using vanishing points for camera calibration. *The International Journal of Computer Vision*, 4(2):127–140.

Chalidabhongse, T. H., Kim, K., Harwood, D., and Davis, L. (2003). A perturbation method for evaluating background subtraction algorithms. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS*.

Chang, F., Chen, C.-J., and Lu, C.-J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206 – 220.

Cheung, S.-C. S. and Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. In *Proceedings of the IS&T/SPIE's Symposium on Electronic Imaging*, San Jose, CA, United States.

Chum, O. and Zisserman, A. (2007). An exemplar model for learning object classes. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8.

Colombari, A., Fusiello, A., and Murino, V. (2006). Background initialization in cluttered sequences. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, CVPRW '06, pages 197–, Washington, DC, USA. IEEE Computer Society.

Cristani, M., Bicego, M., and Murino, V. (2002). Integrated region- and pixel-based approach to background modelling. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 3–8, Orlando, FL.

Cucchiara, R., Grana, C., Neri, G., Piccardi, M., and Prati, A. (2002). The sakbot system for moving object detection and tracking. In *Video-Based Surveillance Systems*, pages 145–157. Springer US.

Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342.

Cullen, D., Konrad, J., and Little, T. (2012). Detection and summarization of salient events in coastal environments. In *Proceedings of the IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 7 –12.

Cupillard, F., Bremond, F., and Thonnat, M. (2002). Group behavior recognition with multiple cameras. In *Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 177–183.

Cutler, R. and Davis, L. (1998). View-based detection and analysis of periodic motion. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 495 –500 vol.1.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886 –893 vol. 1.

Damnjanovic, U., Fernandez, V., Izquierdo, E., and Martinez, J. (2008). Event detection and clustering for surveillance video summarization. In *Proceedings of the Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08*, pages 63 –66.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Deriche, R. (1987). Using Canny's criteria to derive a recursively implemented optimal edge detector. *The International Journal of Computer Vision*, 1(2):167–187.

Ding, W. and Marchionini, G. (1997). A study on video browsing strategies. Technical Report CLIS-TR-97-06, University of Maryland, College Park.

Dumont, E., Merialdo, B., Essid, S., Bailer, W., Rehatschek, H., Byrne, D., Bredin, H., O'Connor, N. E., Jones, G. J., Smeaton, A. F., Haller, M., Krutz, A., Sikora, T., and Piatrik, T. (2008). Rushes video summarization using a collaborative approach. In ACM, editor, *TRECVID BBC Rushes Summarization Workshop (TVS 2008) at ACM Multimedia 2008, Vancouver, Canada*, pages 90–94, Vancouver, BC, Canada. ACM, National Institute of Standards and Technology (NIST), Washington, DC, USA. ISBN 978-1-60558-303-7.

Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002). Background and Foreground Modeling Using Non-parametric Kernel Density Estimation for Visual Surveillance. In *Proceedings of the IEEE*, volume 90-7, pages 1151–1163.

Elgammal, A., Harwood, D., and Davis, L. S. (2000). Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision*.

Elhabian, S. Y., El-Sayed, K. M., and Ahmed, S. H. (2008). Moving object detection in spatial domain using background removal techniques - state-of-art. *Recent Patents on Computer Science*, 1:32–54 (23).

Erdem, c. E., Sankur, B., and Tekalp, A. M. (2001). Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 69–72 (2).

Farin, D., de With, P., and Effelsberg, W. (2003). Robust background estimation for complex video sequences. In *Proceedings of the IEEE International Confeference in Image Processing (ICIP)*, pages 145–148, Barcelona.

Figueiredo, M. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:381–396.

Friedman, N. and Russell, S. (1997). Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gibson, D. (1999). Finite state machines - making simple work of complex functions. Technical report, SPLat Control Pty Ltd.

Goldmann, L. (2009). *Towards an Universal Person Description Framework for Looking at People Applications*. PhD thesis, Technische Universität Berlin.

Gorur, P. and Amrutur, B. (2011). Speeded up Gaussian Mixture Model Algorithm for Background Subtraction. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*.

Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., and Ishwar, P. (2012). changedetection.net: A New Change Detection Benchmark Dataset. In *Proceedings of the IEEE Workshop on Change Detection (CDW) at CVPR*, Providence, RI.

Grana, C., Borghesani, D., and Cucchiara, R. (2009). Connected component labeling techniques on modern architectures. In *Image Analysis and Processing - ICIAP*, volume 5716 of *Lecture Notes in Computer Science*, pages 816–824. Springer Berlin Heidelberg.

Gray, R. (2008). How Big Brother watches your every move. online: http://www.telegraph.co.uk/news/uknews/2571041/How-Big-Brother-watches-your-every-move.html.

Grigorescu, C., Petkov, N., and Westenberg, M. A. (2004). Contour and boundary detection improved by surround suppression of texture edges. *Journal of Image and Vision Computing*, 22(8):583–679.

Grossmann, E., Kale, A., Jaynes, C., and Cheung, S.-C. (2005). Offline generation of high-quality background subtraction data. In *Proceedings of the British Machine Vision Conference*.

Guler, S., Silverstein, J., and Pushee, I. (2007). Stationary objects in multiple object tracking. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 248 –253.

Gutchess, D., Trajković, M., Cohen-Solal, E., Lyons, D., and Jain, A. K. (2001). A background model initialization algorithm for video surveillance. In *in proceedings of the IEEE International Conference on Computer Vision*, pages 733–740.

Haines, T. S. F. and Xiang, T. (2012). Background Subtraction with Dirichlet Processes. In Fitzgibbon, A. W., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 99–113. Springer.

Han, B., Comaniciu, D., and Davis, L. (2004). Sequential kernel density approximation through mode propagation: Applications to background modelling. In *Proceedings of the Asian Conference on Computer Vision*.

Han, B. and Jain, R. (2007). Real-time subspace-based background modeling using multichannel data. In *Advances in Visual Computing*, volume 4842 of *Lecture Notes in Computer Science*, pages 162–172. Springer Berlin Heidelberg.

Haque, M., Murshed, M., and Paul, M. (2008). A hybrid object detection technique from dynamic background using gaussian mixture models. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Harville, M. (2002). A Framework for High-level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models. In *In Proceedings of the European Conference on Computer Vision*, pages 543–560.

Hayman, E. and Eklundh, J. (2003). Statistical Background Subtraction for a Mobile Observer. In *Proceedings of the International Conference on Computer Vision*, pages 67–74.

Heikkilä, M. and Pietikäinen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:2006.

Heikkilä, M., Pietikäinen, M., and Heikkilä, J. (2004). A texture-based method for detecting moving objects. In *Proccedings of the British Machine Vision Conference*, volume 1, pages 187–196.

Hemayed, E. E. (2003). A survey of camera self-calibration. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, volume 0, page 351, Los Alamitos, CA, USA. IEEE Computer Society.

Heras Evangelio, R., Keller, I., and Sikora, T. (2013a). Multiple cue indexing and summarization of surveillance video. In *Proceedings of the 10th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Kraków, Poland.

Heras Evangelio, R., Pätzold, M., and Sikora, T. (2012). Splitting Gaussians in Mixture Models. In *Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 300–305, Beijing, China.

Heras Evangelio, R., Senst, T., Keller, I., and Sikora, T. (2013b). Video indexing and summarization as a tool for privacy protection. In *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Santorini, Greece.

Heras Evangelio, R., Senst, T., and Sikora, T. (2011). Detection of static objects for the task of video surveillance. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, pages 534–540, Kona, HI, USA.

Heras Evangelio, R. and Sikora, T. (2011a). Complementary Background Models for the Detection of Static and Moving Objects in Crowded Environments. In *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 71–76, Klagenfurt, Austria.

Heras Evangelio, R. and Sikora, T. (2011b). Static object detection based on a dual background model and a finite-state machine. *EURASIP Journal on Image and Video Processing*, 2011:858502.

Höferlin, B., Höferlin, M., Weiskopf, D., and Heidemann, G. (2011). Information-based adaptive fast-forward for visual surveillance. *Multimedia Tools and Applications*, 55(1):127–150.

Hofmann, M., Tiefenbacher, P., and Rigoll, G. (2012). Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter. In *Proceedings of IEEE Workshop on Change Detection*.

Horprasert, T., Harwood, D., and Davis, L. S. (1999). A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. In *Proceedings of the IEEE Conference ICCV*, volume 99, pages 1–19.

HSRC (2013). Intelligent video surveillance, vca & video analytics: Technologies & global market - 2103-2020.

Huang, S.-S., Fu, L.-C., and Hsiao, P.-Y. (2004). A region-based background modeling and subtraction using partial directed hausdorff distance. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 956 – 960 Vol.1.

Irani, M., Anandan, P., and Hsu, S. (1995). Mosaic based representations of video sequences and their applications. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 605–611.

ITU-T (1996). Recommendation - subjective video quality assessment methods for multimedia applications. 910.

Jabri, S., Duric, Z., Wechsler, H., and Rosenfeld, A. (2000). Detection and location of people in video images using adaptive fusion of color and edge information. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 4, pages 627–630, Barcelona, Spain.

Jain, R. and Nagel, H.-H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):206–214.

Javed, O., Shafique, K., and Shah, M. (2002). A hierarchical approach to robust background subtraction using color and gradient information. In *Proceedings of the Workshop on Motion and Video Computing*, pages 22 – 27.

Javed, O. and Shah, M. (2002). Tracking and object classification for automated surveillance. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 343–357, London, UK, UK. Springer-Verlag.

Ji, Z., Su, Y., Qian, R., and Ma, J. (2010). Surveillance video summarization based on moving object detection and trajectory extraction. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 2, pages V2–250 –V2–253.

Johnsen, S. and Tews, A. (2009). Real-time object tracking and classification using a static camera. In *Proceedings of the IEEE ICAR,Workshop on People Detection and Tracking*, Kobe,Japan.

Johnson, N. and Hogg, D. (1996). Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:583–592.

Kaewtrakulpong, P. and Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In *Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance Systems*. Kluwer Academic Publishers.

Kang, S., Paik, J., Koschan, A., Abidi, B., and Abidi, M. A. (2003). Real-time video tracking using ptz cameras. In *Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision*.

Karaman, M. (2010). *Towards Robust Object Segmentation in Video Sequences and its Applications*. PhD thesis, Technische Universität Berlin.

Karaman, M., Goldmann, L., Yu, D., and Sikora, T. (2005). Comparison of static background segmentation methods. In *Visual Communications and Image Processing (VCIP), IS&T/SPIE's Electronic Imaging 2005*, volume 5960, pages 596069–596081.

Kilger, M. (1992). A shadow handler in a video-based real-time traffic monitoring system. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 11–18.

Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2004). Background modeling and subtraction by codebook construction. In *In International Conference on Image Processing*, pages 3061–3064.

Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:172–185.

Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., and Russell, S. (1994). Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 4, pages 3776–3781 vol.4.

Krahnstoever, N. and Mendonça, P. R. S. (2006). Autocalibration from tracks of walking people. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 4–7.

Krutz, A., Glantz, A., Frater, M., and Sikora, T. (2011). Rate-distortion optimized video coding using automatic sprites. *Selected Topics in Signal Processing, IEEE Journal of*, 5(7):1309 –1321.

Lazarevic-McManus, N., Renno, J., and Jones, G. A. (2006). Performance evaluation in visual surveillance using the f-measure. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, VSSN '06, pages 45–52, New York, NY, USA. ACM.

Lee, D.-S. (2005). Effective Gaussian Mixture Learning for Video Background Subtraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):827–832.

Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vision*, 77(1-3):259–289.

Li, J., Nikolov, S., Benton, C., and Scott-Samuel, N. (2007). Adaptive summarisation of surveillance video sequences. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2007)*, pages 546 –551.

Li, X., Hu, W., Zhang, Z., and Zhang, X. (2008). Robust foreground segmentation based on two effective background models. In *Proceedings of the 1st ACM international conference on Multimedia Information Retrieval*, MIR '08, pages 223–228, New York, NY, USA. ACM.

Li, Z., Ishwar, P., and Konrad, J. (2009). Video condensation by ribbon carving. *IEEE Transactions on Image Processing*, 18(11):2572 –2583.

Li, Z., Schuster, G., and Katsaggelos, A. (2005). Minmax optimal video summarization. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(10):1245 – 1256.

Liebowitz, D., Criminisi, A., and Zisserman, A. (1999). Creating architectural models from images. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, volume 18, pages 39–50.

Lipton, A., Fujiyoshi, H., and Patil, R. (1998). Moving target classification and tracking from real-time video. In *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV)*, pages 8–14.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150 –1157 vol.2.

Maddalena, L. and Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168 –1177.

Matsuyama, T., Ohya, T., and Habe, H. (2000). Background subtraction for non-stationary scenes. In *Asian Conference on Computer Vision*.

Maybank, S. and Faugeras, O. (1992). A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151.

McFarlane, N. J. B. and Schofield, C. P. (1995). Segmentation and Tracking of Piglets in Images. *Machine Vision and Applications*, 8:187–193.

Migdal, J. and Grimson, W. E. L. (2005). Background subtraction using markov thresholds. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTIONS)*, volume 2, pages 58–65.

Neal, R. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

Oliver, N., Rosario, B., and Pentland, A. (2000). A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831 –843.

Papageorgiou, C., Oren, M., and Poggio, T. (1998). A general framework for object detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 555–562.

Park, S. and Aggarwal, J. K. (2003). Recognition of two-person interactions using a hierarchical bayesian network. In *First ACM SIGMM international workshop on Video surveillance*, IWVS '03, pages 65–76, New York, NY, USA. ACM.

Parks, D. H. and Fels, S. (2008). Evaluation of background subtraction algorithms with post-processing. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199.

Petrenko, A., Boroday, S., and Groz, R. (1999). Confirming configurations in efsm. In *In Proceedings of the IFIP Joint International Conference FORTE/PSTV*, pages 5–24. Kluwer.

Petrovic, N., Jojic, N., and Huang, T. S. (2005). Adaptive video fast forward. *Multimedia Tools Appl.*, 26(3):327–344.

Piccardi, M. (2004). Background subtraction techniques: a review. In *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104.

Pilet, J., Strecha, C., and Fua, P. (2008). Making background subtraction robust to sudden illumination changes. In *Proceedings of the ECCV*.

Porikli, F. (2004). Multi-camera surveillance: Object-based summarization approach.

Porikli, F., Ivanov, Y., and Haga, T. (2008). Robust abandoned object detection using dual foregrounds. *EURASIP J. Adv. Signal Process*, 2008:30.

Porikli, F. and Tuzel, O. (2005). Bayesian background modeling for foreground detection. In *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 55–58, New York, NY, USA. ACM.

Prati, A., Mikic, I., Trivedi, M., and Cucchiara, R. (2003). Detecting moving shadows: algorithms and evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):918 – 923.

Pritch, Y., Ratovitch, S., Hendel, A., and Peleg, S. (2009). Clustered synopsis of surveillance video. In *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'09)*, Genoa, Italy.

Rav-Acha, A., Pritch, Y., and Peleg, S. (2006). Making a long video short: Dynamic video synopsis. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 435–441.

Reddy, V., Sanderson, C., and Lovell, B. C. (2011). A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. *EURASIP Journal on Image and Video Processing*.

Riahi, D., St-Onge, P., and Bilodeau, G. (2012). RECTGAUSS-Tex: Block-based Background Subtraction. Technical Report EPM-RT-2012-03, Ecole Polytechnique de Montreal.

Robinault, L., Bres, S., and Miguet, S. (2009). Real Time Foreground Object Detection using PTZ Camera. In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 609–614.

Rosin, P. (2002). Thresholding for change detection. *Computer Vision and Image Understanding*, 86:79–95.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, pages 430–443.

Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20:23–38.

SanMiguel, J. and Martínez, J. (2010). On the evaluation of background subtraction algorithms without ground-truth. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 180–187.

Sasse, M. A. (2010). Not seeing the crime for the cameras? *Communications of the ACM*, 53(2):22–25.

Sedky, M. H. I., Moniri, M., and Chibelushi, C. C. (2008). Object Segmentation Using Full-Spectrum Matching of Albedo Derived from Colour Images. UK patent application no. 0822953.6 16.12.2008 GB, 2008, PCT patent application international application no. PCT/GB2009/002829, EP2374109, 2009, US patent no. 2374109 12.10.2011 US, 2011.

Shen, J. (2004). Motion detection in color image sequence and shadow elimination. In *Proceedings of the SPIE. Visual Communications and Image Processing*, volume 5308, pages 731–740.

Singh, A., Sawan, S., Hanmandlu, M., Madasu, V., and Lovell, B. (2009). An abandoned object detection system based on dual background segmentation. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 352 –357.

Sinha, S. and Pollefeys, M. (2006). Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Computer Vision and Image Understanding*, 103 (3):170–183.

Smith, M. A. and Kanade, T. (1995). Video skimming for quick browsing based on audio and image characterization. Technical report, Carnegie Mellon University.

Stauffer, C. and Grimson, W. (1999). Adaptive Background Mixture Models for Real-time Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, Fort Collins, CO, USA.

Suau, X., Casas, J. R., and Ruiz-Hidalgo, J. (2009). Multi-resolution illumination compensation for foreground extraction. In *ICIP'09: Proceedings of the 16th IEEE International Conference on Image Processing*, pages 3189–3192, Piscataway, NJ, USA. IEEE Press.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Tanizaki, H. (1987). Non-gaussian state-space modeling of nonstationary time series. *J. Amer. Statist. Assoc.*, 82:1032–1063.

Tiburzi, F., Escudero, M., Bescos, J., and Martinez, J. (2008). A ground truth for motion-based video-object segmentation. In *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, pages 17–20.

Titterington, D. M. (1984). Recursive Parameter Estimation Using Incomplete Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2).

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and Practice of Background Maintenance. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, page 255, Los Alamitos, CA, USA. IEEE Computer Society.

Tsai, R. J. (1987). A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3:323–344.

Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and TrendsR in Computer Graphics and Vision*, 3:177–280.

Ueda, N., Nakano, R., Ghahramani, Z., and Hinton, G. E. (2000). Split and Merge EM Algorithm for Improving Gaussian Mixture Density Estimates. *The Journal of VLSI Signal Processing*, 26:133–140.

Venetianer, P., Zhang, Z., Yin, W., and Lipton, A. (2007). Stationary target detection using the objectvideo surveillance system. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 242 –247.

Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154.

Viola, P., Jones, M., and Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision*, volume 2, pages 734–741.

Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785.

Xiong, Z., Rui, Y., Radhakrishnanz, R., Divakaranz, A., and Huang, T. S. (2005). *A Unified Framework for Video Summarization, Browsing and Retrieval*, chapter 9.2, in The Image and Video Porcessing Handbook. Academic Press, 2nd edition.

Xue, K., Ogunmakin, G., Liu, Y., Vela, P., and Wang, Y. (2011). Ptz camera-based adaptive panoramic and multi-layered background model. In *International Conference on Image Processing (ICIP)*, pages 2949 –2952.

Yeh, T., Lee, J., and Darrell, T. (2009). Fast concurrent object localization and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 280–287.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.*, 38.

Yin, Z. and Collins, R. (2007). Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yu, T., Zhang, C., Yui, Y., Cohen, M., and Wu, Y. (2007). Monocular video foreground/background segmentation by tracking spatial-color gaussian mixture models. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, Austin, Texas.

Zang, Q. and Klette, R. (2004). Robust background subtraction and maintenance. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages (2) 90–93.

Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19.

Zhang, S., Yao, H., and Liu, S. (2008). Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, pages 1556–1559.

Zhang, X., Yang, Y.-H., Han, Z., Wang, H., and Gao, C. (2013). Object class detection: A survey. *ACM Comput. Surv.*, 46(1):10:1–10:53.

Zhang, Z. (1998). A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research.

Zhang, Z. (2004). Camera calibration. In *Emerging Topics in Computer Vision*, chapter 2, pages 4–43. Prentice Hall Professional Technical Reference.

Zhao, G. and Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29.6:915–928.

Ziou, D. and Tabbone, S. (1998). Edge detection techniques - an overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559.

Zitová, B. and Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000.

Zivkovic, Z. (2004). Improved Adaptive Gausian Mixture Model for Background Subtraction. In *Proceedings of the International Conference on Pattern Recognition*.

Zivkovic, Z. and van der Heijden, F. (2004). Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26.

Zivkovic, Z. and van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27:773–780.