Carried Baggage Detection and Recognition in Video Surveillance with Foreground Segmentation

by

Giounona Tzanidou

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

29th May 2014

Abstract

Security cameras installed in public spaces or in private organizations continuously record video data with the aim of detecting and preventing crime. For that reason, video content analysis applications, either for real time (i.e. analytic) or post-event (i.e. forensic) analysis, have gained high interest in recent years. In this thesis, the primary focus is on two key aspects of video analysis, reliable moving object segmentation and carried object detection & identification.

A novel moving object segmentation scheme by background subtraction is presented in this thesis. The scheme relies on background modelling which is based on multi-directional gradient and phase congruency. As a post processing step, the detected foreground contours are refined by classifying the edge segments as either belonging to the foreground or background. Further contour completion technique by anisotropic diffusion is first introduced in this area. The proposed method targets cast shadow removal, gradual illumination change invariance, and closed contour extraction.

A state of the art carried object detection method is employed as a benchmark algorithm. This method includes silhouette analysis by comparing human temporal templates with unencumbered human models. The implementation aspects of the algorithm are improved by automatically estimating the viewing direction of the pedestrian and are extended by a carried luggage identification module. As the temporal template is a frequency template and the information that it provides is not sufficient, a colour temporal template is introduced. The standard steps followed by the state of the art algorithm are approached from a different extended (by colour information) perspective, resulting in more accurate carried object segmentation.

The experiments conducted in this research show that the proposed closed foreground segmentation technique attains all the aforementioned goals. The incremental improvements applied to the state of the art carried object detection algorithm revealed the full potential of the scheme. The experiments demonstrate the ability of the proposed carried object detection algorithm to supersede the state of the art method.

Acknowledgements

I would like express my sincere gratitude to my supervisor Prof. Eran A. Edirisinghe for his patient guidance and support through the long years of research. His continuous encouragement and trust in my ideas and judgment, when something seemed to be infeasible, was the most vital contribution into the accomplishment of this major challenge.

I would like to acknowledge the department of Computer Science, Loughborough University for the financial support provided through my research scholarship. My sincere thanks also go to Dr Helmut Bez and Dr Lin Guan for their annual review on my research progress and advice given. I would like to extend my gratitude to the academic and secretarial staff of the department of Computer Science for their kind assistance.

My heartiest thanks go to all the colleagues of my research group. In particular, the exciting discussions on various research topics with Dr Iffat Zafar, Dr Dhammike Wickramanayake, Niraj Doshi, Muhammad Fraz, Andrew Leonce and Dr Nesreen Otoum are greatly valued. I could not forget the kind help and advice they offered each in area of their expertise. My best thanks are dedicated to my dear friends Anoud Bani-Hani, Sara Saravi, Fatma Al-Maqbali and Asila Alhajri for all the interesting conversations and moral support.

I would also like to say a big thank you to my beloved parents who with all their means and sacrifices have provided me with endless love, all the necessities to bring me up. Furthermore I am most grateful to my adored sister and brother who with their affection and laughter always managed to show me how to look at the bright side of life. Along with them I would like to thank all my relatives and friends in my longing home country.

Last but not least, all my gratefulness goes to my best friend Ramesh J.Baskaran for his immense support, patience, advices, and help.

Giounona Tzanidou, May 2014

Publications

Conference proceedings

- G. Tzanidou, E. A. Edirisinghe, "Automatic baggage detection and classification", Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp.825-830, 2011.
- G. Tzanidou, E. A. Edirisinghe, "A novel approach to extract closed foreground object contours in video surveillance", Proceedings of SPIE, vol. 9026, 902615, 2014

Journals

- G. Tzanidou, I. Zafar, E. A. Edirisinghe, "Carried Object Detection in Videos Using Color Information", IEEE Transactions on Information Forensics and Security, vol.8, no.10, pp.1620,1631, Oct. 2013.
- M. Fraz, I. Zafar, G. Tzanidou, E. A. Edirisinghe, M. S. Sarfraz, "Human object annotation for surveillance video forensic", Journal of Electronic Imaging, vol. 22, no. 4, Oct. 2013

To be submited

- G. Tzanidou, E. A. Edirisinghe, "Closed moving foreground contour segmentation in videos", SPIE Journal of Electronic Imaging.

Contents

List of Figures

List of Tables

List of Abbreviations

BTR	Bag Type Recognition
CBGS	Codebook Background Subtraction
CCM	Colour Co-occurrence Matrix
CFC	Closed Foreground Contour
COD	Carried Object Detection
CO	Carried Object
CTT	Colour Temporal Template
EEF	Edge Enhancing Filter
EM	Expectation Maximisation
FTT	Frequency Temporal Template
GMM	Gaussian Mixture Model
HoG	Histograms of Oriented gradients
ICA	Independent Component Analysis
ICP	Iterative Closest Point
LBP	Local Binary Patten
LLSQ	Linear Least Squares
MoG	Mixture of Gaussians
MRF	Markov Random Field
PCA	Principal Component Analysis
RH	Ratio Histogram
ROI	Region of Interest
SVM	Support Vector Machine
SVR	Support Vector Regression
TSE	Total Shadow Error
VD	Viewing Direction

Chapter 1

Introduction

In recent years automated video surveillance has attracted a high level of interest from the computer vision research community. CCTV camera systems have been installed in busy public spaces such as airports, train stations, and city centres collecting valuable data for real-time and post event analysis, which is mostly carried out manually, by vigilant CCTV operators. According to recent reports, the number of surveillance cameras in the UK reached 4.2 million, resulting in the impressive analogy of 1 CCTV camera being installed per 14 individuals of the population [5]. Such surveillance systems play a major role in crime detection and prevention and are an important segment of public infrastructure that protects national security at all levels.

Often a detailed description of a human appearing in CCTV footage is used in an individual's behaviour analysis. Detection and recognition of luggage being carried is one of the most widely accepted descriptions vitally used in the theft detection and criminal behaviour identification. For instance, tracking of a carried object could lead to detection of any exchange of objects that happens between people carrying them, where a violent object exchange could be characterised as theft. Another example is the immediate detection of luggage left unattended in busy places such as airports and large train stations, which facilitates early luggage contents examination by the officials.

Conceptually, automatic baggage detection in video footage can be achieved by observing the changes in human appearance and gait caused by carried objects. Usually the standard steps that are followed in computer vision for carried object detection in videos are: moving object detection and segmentation, moving object classification into human and non-human, object tracking through the frames, and analysis of the segmented silhouette to detect any human body shape violations or change in gait style or periodicity.

Moving object detection is achieved with background modelling and subtraction methods such as the C. Stauffer & W. Grimson algorithm [130] while object

Figure 1.1: Segmented foreground silhouettes of a person carrying a backpack.

tracking is successfully performed with mean-shift tracking algorithm [28]. To identify a moving object as human, Histograms of Oriented gradients (HoG) are usually used as features [32]. The implementation of all these steps results in a collection of binary or colour silhouette images of the subject that should be further processed to detect carried objects (see Figure 1.1).

1.1 Research Motivation and Goals

One of the procedures described above, i.e. moving object segmentation, is very critical for accurate Carried Object Detection (COD). The importance of moving object segmentation becomes obvious by observing Figure 1.1, which illustrates a sequence of silhouettes segmented by a standard background subtraction method. The last two silhouettes of the sequence are partially segmented, due to similarity of colour between the carried backpack and the background. Moving object segmentation in visual surveillance is of vital importance as beyond COD many other tasks such as vehicle type recognition, number plate recognition, object tracking, action recognition and many other such tasks, rely on it.

The moving object segmentation is usually performed by background subtraction, which means subtracting the current frame from a reference background frame or a background model. The subtraction itself might be a trivial task but background modelling and maintenance requires careful design since it should cope with a variety of practical situations that include: illumination changes in the scene, the presence of dynamic background (sea waves, tree leaves, fountains, escalators) and cast shadows, foreground/background similarity, adverse weather conditions (snow, rain), objects that ceased moving, camera jitter, PTZ cameras and night time videos. For this reason, even after 20 years of ongoing research and countless publications attempting to address these issues, the background modelling remains a fascinating and challenging research field.

There is no single method known that performs real-time under all the above mention circumstances. Most of the current applications aim at tackling problems of immediate priority such as the presence of illumination changes, cast shadows, and dynamic background. These will be the three goals of this thesis, with special

attention drawn onto shadow elimination and segmentation of complete object contours as these are the facts that mostly affect the results of a COD system.

After segmenting the moving objects it is important to track them, which means establishing correspondence between the detected foreground regions. Since the available tracking algorithms have satisfactory performance and it would be impossible to accommodate such variety of topics in one thesis, this subject was left for future investigation.

Once the sequence of silhouettes of a moving subject is acquired, a COD scheme can be designed. The importance of COD in visual surveillance, expressed in the beginning of this chapter, is the reason for conducting research in this specific topic. If compared to the research progress made in the domain of background modelling over the recent years, the state-of-the-art in COD methodologies have hardly been evolved. Most of the techniques are based on body shape analysis via temporal human like templates, which encode the history of motion, some on gait analysis, and very few up to date approaches involve cues such as optical flow, bag shape analysis and colour segmentation. Naturally, the most modern techniques exhibit better results than the mainstream body shape analysis based techniques. However, one approach that attracts attention is of D. Damen and D. Hogg [34] where the authors compare a human like temporal template with unencumbered human models to detect the carried object. The MATLAB implementation of this method is available for research purposes. This fact along with the encouraging experimental results presented by the authors made this algorithm a good starting point for research. A careful examination of D. Damen and D. Hogg's system revealed that although the methodology proposed is promising, the system suffers from several weaknesses. Sometimes, body parts or clothes are detected as carried objects and at the same time some carried objects fail to be detected. Further the proposed system is not fully automatic. Hence it is possible to resolve the weaknesses of the system, in view of disclosing the full potential of the scheme. This is one primary goal of the research presented in this thesis.

1.2 Original Contributions

This thesis consists of a number of original contributions to video analytics and forensics application domains. The key contributions which are outlined below mainly focus on foreground/background segmentation and COD.

1. **Design of an edge enhancing smoothing filter**

   This thesis contributes with the novel design and implementation of a filter that enables the enhancement of edge features of an object. The proposed filter originated while designing a closed foreground contour segmentation algorithm implemented for foreground object extraction. To extract gradient features the image should be pre-smoothed for noise reduction with a carefully chosen linear filter that would not degenerate edges. It is well known that the Gaussian filter does not respect edges; whilst on the other hand a truncated *sinc* filter over-enhances them. Hence a filter that would lie between the Gaussian and the *sinc* filter was designed and implemented. Its ability to enhance edges in foreground object extraction is proved through detailed experiments.

2. **Closed foreground contour segmentation with shadow elimination**

   As it was mentioned earlier prerequisite for most of the COD systems is moving silhouette segmentation. Consequently the success of a COD system depends on the successful foreground object segmentation. A key contribution of this thesis is the novel design, implementation and testing of a closed foreground contour segmentation algorithm. The approach proposed is proven to be robust to gradual illumination changes, handles specific dynamic background scenarios, reduces significantly the cast shadows and ensures the extraction of whole silhouettes. Instead of using the conventional colour pixel values as features for background modelling, the proposed method utilises the unique properties of the pixel value gradient extracted by the use of the proposed edge enhancing filter and the phase congruency features to extract the foreground contours. These contours are reflected on the corresponding edge images to isolate the edges that belong to foreground silhouettes, thus refining the crude contours. An additional contribution of this method is the processing of the refined contours for noise reduction using colour features and a classifier. To ensure closed contours the edges are extended and closed via anisotropic diffusion, which has not been used previously in literature as a post-processing step in foreground segmentation.

3. **Viewing direction estimation of pedestrians**

   Viewing direction estimation refers to the direction that a pedestrian faces while moving in the scene. It plays an important role in low resolution

surveillance environments where tasks as gaze tracking or head orientation estimation could be assisted by body orientation estimation. By extension, interaction between two or more people could be expressed by the direction that their bodies are facing. However, in this study, the viewing direction estimation is employed for the purpose of COD. It is common sense that the location of carried objects and their visibility depends on the position of the viewer in relevance to the position, and moving direction of the pedestrian. Therefore it was vital to devise a robust method for viewing direction estimation. The most common way for direction estimation is to use histogram of oriented gradients. However, the proposed direction estimation stays away from complex feature descriptors and develops simple features based on the geometry of the upper part of human silhouette. Though seemingly simple, it is proven that the method attains a high level of accuracy in classification.

4. **Carried object detection**

   A major contribution of this thesis is a robust, novel, carried object detection approach that utilises the above mentioned key contribution on viewing direction estimation and incremental original research ideas. The proposed COD system is one of the few methods which use colour information to segment the carried objects. It follows the standard steps used in the state-of-the-art algorithm proposed by D. Damen and D. Hogg [34], namely, the estimation of moving direction of humans, construction of human-like temporal templates, and their comparison with the best matching view-specific exemplar. However each step is approached from the advantageous viewpoint that colour offers, thus improving the traditional steps used in prior literature. In the light of new information the segmentation of carried objects becomes more accurate and easier to distinguish from body parts. The experimental results show that the proposed COD is capable of achieving a higher level of accuracy than the state of the art COD approaches.

5. **Carried bag type recognition**

   A natural extension of COD is carried bag recognition. It is known that object classification is a challenging task and hence the design of a general carried object recognition algorithm would be difficult to address. For this reason it is assumed that in the environment that is monitored, people mostly appear carrying objects of a particular type, i.e. a bag / piece of luggage. Thus, it is possible to classify the bags/luggage by taking into ac-

count their position in relevance to the body of the person that caries them. This approach classifies the luggage into 5 categories and attains satisfactory performance. The method could be greatly enhanced in the future by the shape features of carried objects.

1.3 Thesis Structure

For clarity of presentation the thesis is organised into seven chapters and additional experimental results and other supporting material are included in the appendices as appropriate.

Chapter 2 introduces the reader to the topic of COD through a detailed review of current literature. The chapter also includes a review of the most popularly used foreground/background (or moving object) segmentation techniques since they provide the underlying fundamental algorithms required for robust COD.

Chapter 3 explains in detail all fundamental theoretical and mathematical concepts required for understanding the original ideas and approaches proposed in the contributory chapters of this thesis. The chapter starts with the description of two benchmark algorithms, one on COD and the other on foreground/background segmentation, and subsequently continues with a presentation of theoretical preliminaries that cover numerous topics.

Chapter 4 is the first contributory chapter of this thesis and presents original research in foreground/background segmentation. It describes a novel approach for extracting closed foreground contours and eliminating cast shadows. Within the proposed framework a novel edge enhancing filter emerged and is thus formally presented.

Chapter 5 initially presents a number of possible improvements to the benchmark COD algorithm and continues with proposals to achieve fully automated operation and grater usability of the algorithm. The latter is achieved by the introduction of an automatic viewing direction estimation algorithm and carried bag type recognition system.

Chapter 6 proposes a novel method for COD by using colour information. The method follows the standard steps taken in the state-of- the-art algorithm approaching them from the point of view facilitated by colour information.

Chapter 7 summarises the research presented within the thesis drawing overall conclusions. It also suggests possible future improvements and enhancements.

Appendices provide with additional experimental results and pseudocodes.

Chapter 2

Literature Review

The literature review focuses on the two major tasks that this thesis deals with. While the first is concerned with COD in video, the second relates to moving foreground object segmentation. Early examination of approaches for COD revealed that the results of COD are highly dependent on foreground object segmentation. Successful implementation of the latter will lead to successful performance of the former.

COD literature starts with an insight to gait and motion analysis in subsection 2.1.1 as they are employed in many COD approached described in subsections 2.1.2 and 2.1.3. The COD review continues in subsection 2.1.4 with methods based on silhouette shape analysis in conjunction with neural networks for decision making. A brief description of COD based upon an event such as an object exchange or carried object left in the scene is presented in subsection 2.1.5. Finally, subsection 2.1.6 describes other important methods for COD.

Since a major contribution of this thesis is foreground segmentation the literature review continues to a presentation of background modelling techniques. Due to the presence of a large number of background modelling techniques available in the literature, it is impossible to accommodate all within this review. Therefore, only the most popular and efficient schemes based on statistical background modelling are presented in his chapter. A popular background modelling approach based on a Mixture of Gaussians [130] is described in subsection 2.2.2 followed by a summary of the proposed improvements. Several strategies presented in subsection 2.2.3 if followed, can improve the segmentation accuracy and robustness of background modelling techniques. These involve Markov Random Fields (MRF) for segmentation improvement, background modelling with multiple complementary models to maintain the background when new objects are inserted into the scene and background modelling with texture features to address dynamic background scenarios. Lastly the most recent approaches are summarised in subsection 2.2.4

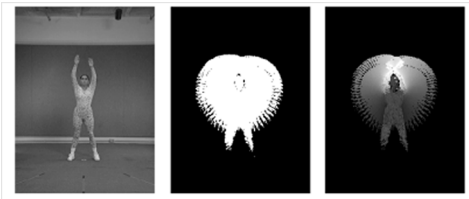


Figure 2.1: An example of Motion Energy Image and Motion History Image in second and third image respectively [8].

2.1 Carried Object Detection (COD)

2.1.1 Gait and Motion Analysis

Motion analysis, gait and action recognition are the forerunners of COD used in many succeeding applications to detect the carrying status of a pedestrian. Motion analysis can reveal the periodic attributes of motion exhibited by humans or animals which separated them from objects with constant motion such as vehicles [30, 31]. Further the motion of different parts of the same object could be evaluated: for instance separate examination of motion of upper and lower limbs could show if the periodic characteristic of any of them ceased to exist because of the presence of a carried object [55]. Gait investigation could disclose information about the gender and even the identity of a person [84, 83]. An example of a carrying status detection based on gait is shown in [133] where the average human gait is convolved with a bank of Gabor filters of different scales and orientation and classified as carrying a briefcase or not with a general tensor discriminant analysis method developed by the authors. However this method has not been tested extensively on a COD dataset but only on a gait identity dataset.

Human movement and pose recognition by analysis of different types of temporal templates that encode motion over time [35, 8, 15, 64, 55] could lead to detection of carried objects as it will be shown in subsection 2.1.2. A. Bobick and J. Davis in [8] and [35] introduce the idea of temporal template as a vector image that is obtained by accumulating binary images which represent a person's movement over time. Two different types of temporal templates are proposed in [8] and [35] and are referred as Motion Energy Images (MEI) where all the binary images are united to one image or Motion History Images (MHI) where the cumulative image has a variation in intensity of the pixels based on the time of the execution of a movement (Figure 2.1). These temporal templates are used for action recognition by matching their Hu moments against a set of training examples that encode a series of aerobic movements.

Another temporal template similar to MHI named timed Motion History Images (tMHI) encodes in it actual time, making the MHI independent of frame rate [15].

This tMHI is combined with the orientation of its gradient vectors which makes it capable of identifying the direction of local and global motion.

2.1.2 COD Based on Motion Analysis, Temporal Templates and Body Main Axis

A good starting point is "W4: real-time surveillance of people and their activities" in [55], which proposes a framework for a system that: detects foreground objects, classifies the detected objects into humans and non-humans, tracks multiple humans even if they appear as a group, tracks human body parts, recognises interaction of humans with objects, and detects carried objects. Commonly it is an integral system, presented through a number of publications over a period of time [56, 30, 31, 53] (described subsequently), which follows all standard steps required to achieve carried object detection in a video sequence [64].

The initial publication is "W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People" published in 1998 by I. Haritaoglu et al. [56]. The authors proposed a tracking technique that relies on monochromatic information, particularly useful for night-time or infrared surveillance. Appearance models of people are constructed during their motion to facilitate tracking under occlusion. These models are also useful to recognise an individual's actions with reference to an object.

Initially a foreground region detection is performed by simple pixel based thresholding of the absolute differences of the current pixel from the maximum and minimum values of that pixel observed over time. All other traditional post processing methods, such are morphological processing for noise removal, connected component analysis etc., are applied. Afterwards, a motion model is constructed for each foreground object to predict their motion in the subsequent frames. If there is significant overlap between the predicted bounding box of the object and the bounding box in the current frame, then a match is found. Beyond that, edge correlation of silhouettes between two consecutive frames is performed to confirm the match. For tracking under occlusion, local versus global correlation techniques are preferred; for example for tracking of a person's head. To address scenarios of objects merging and splitting after a time interval of combined motion, appearance models are constructed. Such a model is a temporal textural template defined as [56]

$$C^t(x,y) = \frac{I(x,y) + w^{t-1}(x,y)C^{t-1}(x,y)}{w^{t-1}(x,y) + 1} \tag{2.1}$$

where $I(x,y)$ is the intensity of the foreground pixel, and $w^{t-1}(x,y)$ is the number

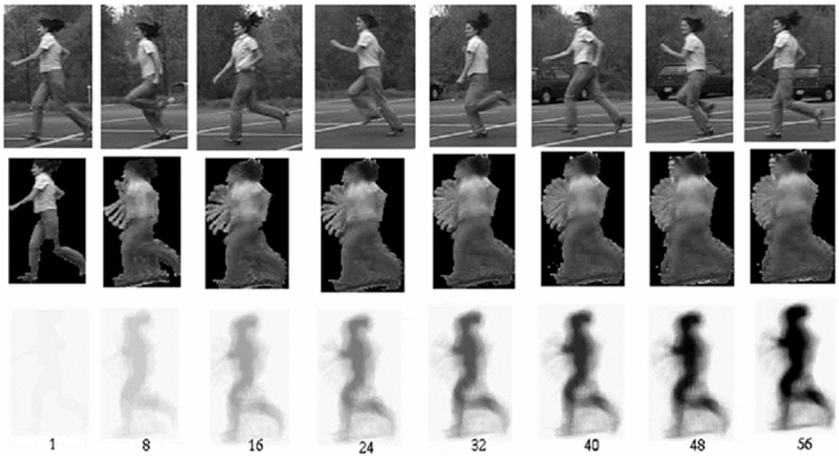


Figure 2.2: An example of temporal textural template developed over time in the second row and the corresponding frequency temporal template in the third row [55].

of times that the specific pixel in $C(x, y)$ has been classified as foreground (Figure 2.2). Having a number of temporal textural templates before merging and after splitting, it is possible to identify the tracked person by correlating its temporal templates. Aiming at action recognition, an additional cardboard human model is constructed to track the human body parts.

At the same time, another publication by R. Cutler and L. Davis in [30] proposed the detection of periodic motion by analysing the self-similarity of an object during its motion. Prerequisite for this method is the segmentation of moving objects. A self-similarity measure of a segmented object is computed as the sum of absolute differences between the pixel values of the same object at two different time points. Subsequently a self-similarity square matrix is constructed for all available combinations of time points (Figure 2.3). Simple observation and Fourier analysis of the self-similarity matrix reveal the existence of periodic motion or combination of two periodic motions with different frequencies. As it was later shown in [31] this method is especially useful for the following tasks: (a) moving object classification into humans and non-humans (Figure 2.3), (b) defining the number of people in a group based on the number of different frequencies that occur, (c) action recognition and carried object detection by analysing the periodicity of upper limbs, and (d) person recognition by their gait for tracking purpose.

Another problem that could arise in a system as W4 is the separation and tracking of people when they move in groups. Subsequent publication of I. Haritaoglu et al. proposes a complementary to W4 part "Hydra" [54], a system for multiple people detection and tracking within a group. To that end "Hydra" utilises local shape features by analysing the boundary of an object, global shape information

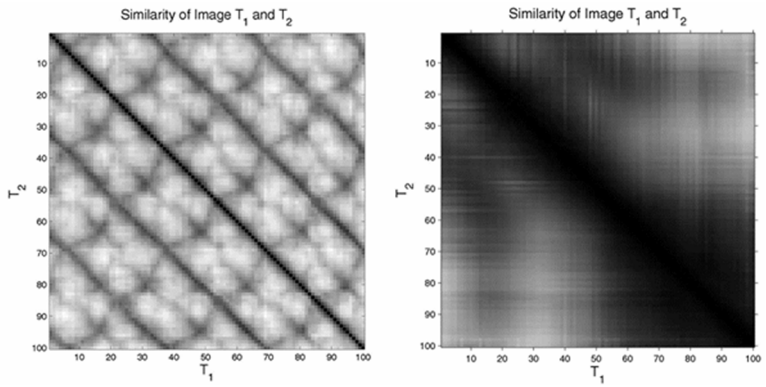


Figure 2.3: Self-similarity matrices for all combinations of available time points. In the left is the self-similarity matrix of a walking pedestrian while in the right is of a moving car [31].

(i.e. the inherent structure of human body), and appearance information (e.g. head texture). First it analyses the boundary of the group silhouette to find the heads of the people in the group, and then it takes the vertical projection histogram of the group silhouette to find the local minima that are used to separate the individuals from each other. The pixels of the group shape area are assigned to each human silhouette that is comprised of them, according to their path distance from the torso main axis. People in groups are tracked according to their heads.

Now that all the necessary steps are taken to segment and recognise moving people and ensure that they are tracked effectively even in groups, it makes sense to apply carried object detection. The project that tackles this issue is Backpack proposed in [53] of I. Haritaoglu et al. Backpack is the final touch of W4 in [55] and uses attributes of the human silhouette symmetry and periodic motion to detect carried objects. The idea applied here is, that anything that violates the symmetry around the body axes and is not part of the body comprises a foreign object [53, 55].

Initially, the background subtraction is performed to obtain the binary moving silhouette, and a major axis which traverses the centre of the body is determined by using Principle Component Analysis (PCA). Since the periodic motion of the person is taken into account, the method as in [30] and [31] is employed for the calculation of periodicity of the walking pedestrian. Horizontal and vertical projection histograms are obtained to create their self-similarity plots and calculate the frequencies. It is worth to note that, in average, 60 projections are needed for the calculations.

Next step is the recognition of regions which are not symmetric to the major axis. Usually the parts of the human body are symmetric to the axis, thus, regions that do not obey this law are marked as carried objects (Figure 2.4-left). It is a fact that sometimes body parts, like hands and feet, belong to the non-symmetric

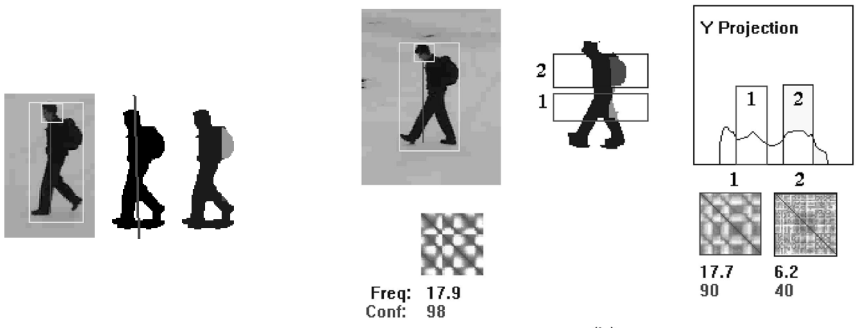


Figure 2.4: Left: Symmetry analysis for COD. Right: Calculation of self-similarity plots for two horizontal projections to determine if the non-symmetric region exhibits recursive motion. The first similarity plot, in contrast to the second, looks like the similarity plot of the whole silhouette and therefore exhibits similar motion. The second plot represents the non-periodic motion of the carried object [55].

regions. To avoid such inaccuracies, a temporal textural template is developed over time and the detected non-symmetric parts of the template undergo periodicity analysis based on two horizontal (because motion is exhibited by upper and lower limbs) projection histograms. The periodicity of the non-symmetric parts (in fact of the horizontal projection histograms that the parts belong to) is compared to that of the person that they belong to. If the periodicity of the shape is similar to the periodicity of the body then it is classified as a body part, otherwise it is considered as a carried object (Figure 2.4-right).

According to experimental results of Haritaoglu the performance of the Backpack is satisfactory, but later tests conducted by Damen and Hogg reveal some weaknesses in Haritaoglus method [33]. These include the facts that, the axis not always crosses the centroid of the body and very frequently parts of the body are detected as carried objects. Moreover the precise estimation of the frequency of the moving person needs at least 12 walking cycles, requiring around 200 frames (17 frames each walking cycle). As mentioned by Javed and Shah, the PCA method for the specification of major axis is not sufficiently efficient as large carried objects cause a significant change in the body shape and consequently the dislocation of the axis [64].

Javed and Shah in [64] proposed another integral system that involves all the prerequisite steps for COD. These are namely: moving object detection by background subtraction, cast shadow removal, object tracking and classification, and finally COD. [Note: Since an example of such a system has already been described, the rest of the paragraph concentrates on COD method only.] The authors adopt the technique of the temporal template to create the Recurrent Motion Image (RMI). In contrast to the aforementioned temporal templates, RMI aims at recording the regions of the moving object that perform kinesis. In this

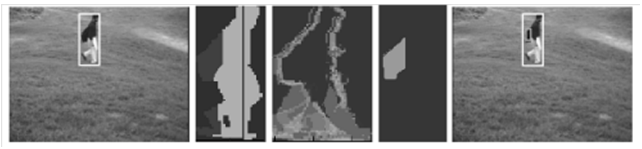


Figure 2.5: From left to right: Detection of a moving person, placement of the main axis and detection of non-symmetric regions in the silhouette, the RMI model, and the detected object after RMI consideration [64].

way the RMI is used for the classification of moving objects and for COD. Javed and Shah detect the position of the head in order to draw a vertical axis which starts from the head and ends at the feet. Subsequently, symmetry analysis is performed to spot the non-symmetric regions of the silhouette, similar to the approach proposed by I. Haritaoglu in [53]. Since the RMI maps the moving parts of the body, a simple comparison of the non-symmetric region with the one in RMI will show if the non-symmetric region belongs to the moving parts or not. In the case that the detected object is non-recurrent, then it is labelled as a carried object (see Figure 2.5).

One of the latest approaches, attracted by the body main axis properties is of Y. Qi, G. Huang and Y. Wang [111]. A Support Vector Machine (SVM) classifier is trained with vectors containing the distance of each point on the contour of the binary silhouette from its main axis. Initially, a background subtraction is performed and the binary silhouette is morphologically processed so that the contour of the shape is clear enough for further analysis. Next, the main axis is computed and all the images are resized to the same height. Finally the characteristics of the contour of the silhouette are forwarded to the SVM for the classification. Once the person is classified as carrying an object or not, the location of the object (possibly a bag) is specified.

The latest proposition for temporal template employment for COD and probably the one with the best performance is of D. Damen and D. Hogg [33, 34] which is described in detail in chapter 3.

2.1.3 Gait Analysis Based Approach

An example of an approach that bases baggage detection on motion, is the method proposed by C. BenAbdelkader and L. Davis [7]. The concept is the analysis of human gait and body shape to determine the presence of carried object, given the assumption that the walking manner and the body shape changes under a carried load. The authors concentrate on two types of carried objects: objects situated in the region of arms and in the region of legs. This fact prompts the segmentation of body into four regions with one of them located at the upper part of the body and

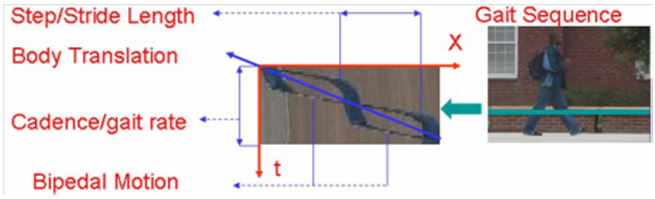


Figure 2.6: Helical signature construction by stacking samples that lie on the same vertical position through the time [112].

the remaining three at the lower part. The periodicity and amplitude of the width of segmented parts are obtained after analysing a sequence of binary silhouettes. After the characteristics (periodicity and amplitude) of naturally walking person are recorded, several constraints, in terms of the period of gait and the recurrent motion of arms and legs, are imposed for the four regions. If any of the constraints is violated then the person is assumed to carry an object. The above method differs from the one of I. Haritaoglu in [53], as it does not perform symmetry analysis.

A similar approach that uses periodicity information is described in [124] by T. Senst et al., focuses on placing people under two categories: carrying a bag or not. The bounding box of the tracked foreground silhouette is divided into $N = 25$ non overlapping blocks and self-similarity plots as in [31] are calculated for each block. To make the similarity plots gait invariant, periodicity dependency measure between two blocks is calculated as the maximum of the absolute relative correlation of the similarity plots. The periodicity dependency measures for all blocks form a feature descriptor that is forwarded to SVM for classification of individual into two categories as having baggage and not. The highest accuracy achieved was 72.5%.

The contribution of Y. Ran et al. in [112] exploits the periodical attribute of human motion to construct helical signatures that hold important information about gait rate and stride length as shown in Figure 2.6. While the accuracy of the other methods is affected by erroneous foreground segmentation and tracking the development of helical signatures does not require foreground segmentation. Analysis of the change in symmetry or frequency of the helix can reveal if a person carries a bag in one or two hands, or if an object is attached to his/her legs or upper body. Beyond carrying condition detection, the authors show how the helical signatures can be utilised for robust pedestrian segmentation when the background is complex and how to handle occlusions.

In [36] the authors detect carried backpacks in order to correct the gait curve for subsequent gait classification. The gait curve is a result of averaging individual gait curves that occur over time. Each gait curve contains a set of points which are space normalised by subtracting contour points of a binary silhouette from its

body main axis. Series of features related to the collation of the curve in the are of human back are thresholded to detect the backpack.

2.1.4 Silhouette Shape and Neural Network Exploitation

In [17] and [16] A. Branca et al. propose a method based on wavelet analysis and neural network for detecting intruders in archaeological sites and the carried by them objects. To do this, they first concentrate on people recognition among other moving objects. The sequence of images in a video facilitates the extraction of moving objects and a collation with the background gives the outline of the moving object. Next, Harr wavelet transform is applied to the segmented binary silhouette in order to decompose the image into subbands. Each of the subbands contains different characteristics/features of the image in frequency and orientation domain. After 3-level decomposition a subband of the last decomposition level is selected and forwarded to a trained, three layer neural network. The output of the network processing is true or false depending on if the moving object is a person or not. The extended edition [16] of A. Branca et al. adds to the existing neural network other two trained, neural classifiers of the carried objects that are to be recognised. The detected, greyscale human silhouette is scanned using a mask of the object in order to detect the carried item.

The work of H. Nanda et al. [99] is another example of detecting carried objects using general human appearance and neural network. The proposed neural network has two layers; one hidden with 20 sigmoid nodes and one output layer with a linear node. Scaled Conjugate Gradient training method has been chosen as the best one for training the network. Initially, foreground segmentation and object tracking is performed and a sequence of aligned and rescaled blobs are extracted for each moving object. Each blob is processed separately and is directly used as input to the neural network which classifies it as pedestrian or pedestrian with distorted shape. The majority of the classification results per subject will indicate the final result. The method is camera view point invariant and the authors report 81.3% classification accuracy.

2.1.5 Detecting People Acquiring Exchanging or Leaving Objects

N. M. Ghanem and L. S. Davis [47] approached the topic of baggage detection in terms of finding the changes in the appearance of pedestrians before entering and after leaving a Region of Interest (ROI). Similarly to other methods the subject is tracked and the foreground is extracted using the Codebook Background

Subtraction (CBGS) method. Subsequently three types of templates are created to spot the difference in the appearance "Before" and "After" entering the ROI. These templates are: the occupancy map which is the common frequency temporal template of a tracked pedestrian, the *codeword* frequency map that records the number of *codewords* (different colour values this pixel took through the time) for each pixel that belongs to foreground, and the colour histogram intersection map where the intersection of the colour histograms is calculated for the "Before"-"After" codebook templates. The next step is the calculation of the difference between each pair of maps. The result is: the Occupancy Difference Map, the Codeword Frequency Difference Map and the Histogram Intersection Map. The difference feature maps are segmented into a number of blocks of different sizes partially covering each other. Each block is represented by the averages of the difference maps values they enclose, thus achieving a grouping of the features. Consequently the number of features for each tracked person reaches high levels and the selection of the most significant features is needed. For this purpose the boosting technique, AdaBoost, is employed. After the selection of the strongest features the SVM classifier is trained to detect backpacks and suitcases. An average of 90% recognition rate was achieved.

In [27] Chi-Hung Chuang et al. addressed baggage detection when bag exchange occurs between two individuals via people tracking and calculation of Ratio Histograms (RH). The ratio histogram is a fraction of the colour histograms of each person before and after the bag exchange occurs. The numerator could be the histogram before the exchange and the denominator the histogram after the exchange and vice versa constructing it this way two different RH per person. The detection of missing colours between the before and after conditions in one of the two RH leads consequently to the detection of a carried object and answers to the question "who, out of the people involved was carrying the bag before and after the exchange".

2.1.6 Other Methods

The method of R. Chayanurak et al. in [23] utilises the star skeleton, which defines the regions representing the limbs and other protruding objects. Initially the contour points of the silhouette are defined and a Delaunay triangulation of the shape is constructed. The parent nodes of the spanning tree formed are averaged into a centroid, which is connected to the most distant points on the silhouette's contour forming a star skeleton (Figure 2.7). The limbs of the stars are tracked and the decision is made by evaluating the level of movement the identified limbs perform. The limb that exhibits the lowest motion is characterised as a carried

(a) (b) (c)

Figure 2.7: The procedure of designing a star skeleton involves the Delaunay triangulation of the contour points (c), the localisation of the centroid from the parent nodes (b) and the calculation of the most distant points from the centroid (a) [23].

object. However, the experimental results presented are not adequate to support the performance efficiency of this method. The work of A. Elgammal in [39] suggests that the dynamic shape of a human silhouette through its walking cycle can be modelled as a manifold in 3-dimensional space. This means that if the silhouette is represented as a feature vector in a high dimensional domain then the dimensionality of that vector can be reduced in such a way that it belongs to a 3-dimensional manifold. The vector that represents the shape of a silhouette is as a distance function of pixel points from the closest point on the silhouette's contour. A non-linear dimensionality reduction method has been used to embedding the manifold to 3-dimensional Euclidean space. Figure 2.8-left shows an example of embedded manifold for 10 walking cycles of a side view silhouette. Each cycle consists of sample points that correspond to shape samples taken from the image gait cycle. The nearby points on the manifold are grouped with K-means clustering to establish the representative points of each shape sample. Complementary manifolds can be constructed for other views of the silhouette.

The authors also propose a non-linear mapping function, based on Radial Basis Function interpolation that takes the points from the 3-dimensional embedding space to the higher dimensional visual space. The goal of this procedure is the reconstruction or refinement of a degenerate input silhouette shape by detecting the corresponding closest point on the embedded manifold and recovering the intrinsic body shape using the mapping function. Such an example of body shape reconstruction is shown in Figure 2.8-right.

This method is immediately related to carried object detection as it is shown in [81] and [82] by C.-S. Lee and A. Elgammal. They form a mapping function $y_t = \gamma(b_t; s, v)$ that maps the body pose b_t from the embedding space to the visual space, given the shape style s and viewing direction v. Given an input silhouette the algorithm retrieves the closest body configuration from the manifold and fills any holes in the input silhouette. Once the silhouette's shape is improved the procedure is repeated to improve the accuracy of the retrieved configuration. In

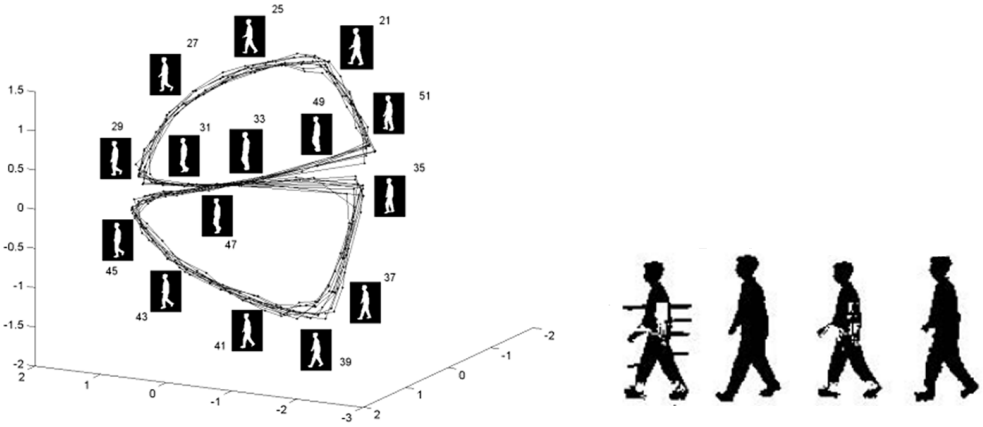


Figure 2.8: Left: An example of gait manifold for 10 walking cycles of side view of a pedestrian. Right: Two examples of shape reconstruction while preserving the viewing direction[39].

the same iterative manner the carried objects are detected as outliers of the best matching configuration retrieved.

Research that tackles the typical negative impacts of foreground-background segmentation on carried object detection is presented in [123] by T. Senst et al. They proposed modelling the motion of bounding boxes detected via optical flow as a Gaussian probability function. A bounding box exhibits a uniform motion with one main direction and so do the torso, head and any carried objects. Non uniform, periodic motion is performed by limbs which can be differentiated by the rest of the aforementioned areas by comparing their motion with the motion of the bounding box. A uniform motion model of an average person is constructed to classify the uniform motion of carried objects from the uniform motion of the torso.

One of the most recent approaches combines low level region detectors as optical flow, mean shift colour segmentation, occlusion boundary based moving blob detector [37]. The combined detected regions are classified with SVM according to a number of features related to shape of the region and its relation to the human silhouette.

Another up to date method of A. Tavanai et al. [134] combines 3 different probabilities of presence of a carried object (CO) related to, the geometric shape of a common CO, continuous spatial relationship of person and the CO, significant overlap of the CO with the found protrusions. The regions that maximise the combined probability are considered to be a CO. The experiments showed that the CO localisation achieved is more accurate than the one of D. Damen & D. Hogg.

2.2 Moving Foreground Extraction

In video analytic applications moving object segmentation constitutes the first step in a sequence of operations that aim at the analysis of moving objects. Foreground extraction can be performed by background subtraction methods, which most of the time involve background modelling, background initialisation, background maintenance and foreground segmentation modules. Background subtraction means extracting the foreground by subtracting the current frame from a reference background frame or a background model. Since the background is not static and objects are added and removed from the background scene constantly, a single reference background frame is not sufficient. Therefore, more sophisticated background models have been developed that are updated over time to handle various extended situations like scene illumination changes, cast shadows, dynamic background, camera jitter, foreground/background similarity, moving background objects, inserted background objects, PTZ cameras and night time videos. Most of the above issues are addressed in a significant number of publications, with 360 of them cited by T. Bowmans in his comprehensive survey [10].

Before starting the review it should be noted that the reader can refer to the surveys on background subtraction methods [10, 12, 9] published by T. Bouwmans who maintains a website with references to most of the influential background subtraction techniques (sites.google.com/site/backgroundsubtraction/overview [1]). According to T. Bouwmans the traditional background modelling techniques can be classified into five basic groups as presented in Table 2.1. The shaded methods and their improvements will be discussed in this section.

2.2.1 Basic Background Modelling

One of the very first examples of background subtraction was developed to track piglets in a pen [95]. The method was based on frame differencing; that is subtraction of the current frame from a reference frame, which is updated over time according to the following scheme.

$$B_t = \begin{cases} B_{t-1} + 1 & if\ I_t > B_{t-1} \\ B_{t-1} - 1 & if\ I_t < B_{t-1} \end{cases} \tag{2.2}$$

where, B_t is the value of a pixel in the reference background at a point of time t and I_t is the value of the pixel in the current frame. This is the method of a running median, where the reference image converges to a median, after a number of iterations. It is a computationally inexpensive method and only one

[1] The references given in each group might not always be accurately classified into it.

Table 2.1: Categorisation of background modelling techniques.

Basic Background Modelling		Average - Running Average [103]
		Median - Approximated Median [95]
		Histogram over time [66]
Statistical Background Modelling	Density functions	Single Gaussian [147]
		Mixture of Gaussians [130]
		Kernel Density Estimation [40]
	Machine learning	Support Vector Machine
		Support Vector Regression (SVR)
		Support Vector Data Description
	Subspace learning	Principal Components Analysis
		Independent Component Analysis
		Incremental Non Negative Matrix Factorisation
Background Clustering		K-means
		Codebook
		Basic Sequential Clustering
Neural Network Background Modelling		General Regression Neural Network
		Self Organizing Neural Network [88]
Background Estimation		Wiener Filter [136]
		Kalman Filter
		Chebytchev Filter

image needs to be stored every time. The median can also be calculated by the Least Median of Squares method [141] or the M-estimators technique [153] that both require the storage of $N \geq 3$ frames. A recently proposed histogram based median estimation requires the storage of a significantly higher number of frames (15-91 frames), which however does not affect the real time performance [62].

Another background subtraction method of the same category is the estimation of the background model as an average or a running average of frames. The simplest calculation of the average background model, which can be updated over time is expressed by the following equation:

$$B_t(x, y) = aI_t(x, y) + (1 - a)B_{t-1}(x, y) \tag{2.3}$$

where a is the learning rate, I_t is the current frame and B_t is the background model at time t. The foreground F can be segmented by thresholding the difference between the current frame and the background model with a threshold T

(Equation 2.4) [103, 105, 158].

$$F(x,y) = \begin{cases} 1 & if\ |I_t(x,y) - B_t(x,y)| < T \\ 0 & otherwhise \end{cases} \tag{2.4}$$

The average background model could be conveniently applied in colour videos as well, where each colour component is updated separately. If the HSV colour space is employed then it is possible to detect the shadow pixels by thresholding the ratio I_t^V/B_t^V of the value component [105]. Another factor that could impair the performance is brightness changes that may occur. To reduce that impact, the distance of the mean brightness μ_{I_t} of a region of I_t from the mean brightness μ_{B_t} of the region of B_t can be added to each colour channel c according to the following equation [158]

$$B_t(x,y,c) = aI_t(x,y,c) + (1-a)B_{t-1}(x,y,c) + (\mu_{I_t} - \mu_{B_t}) \tag{2.5}$$

This holds if the RGB colour space is used for background modelling, where each colour component includes brightness information. Two recent methods which follow the same baseline attempt to improve the method by incorporating spatial information [25] or allow the learning rate a adapt whenever a new object is introduced to the scene or when the background illumination changes fast [69].

Histogram based background modelling is the last example of basic background modelling methods. In its simplest approach, the history of a pixel values form a histogram where the value with the highest frequency is accounted for by the background [66, 128]. Because the history of values of the pixel that belongs to background vary (e.g. due to illumination changes) sometimes it is hard to find a local maximum in the histogram. Therefore, the frequencies of nearby values are organised into one bin [66, 128, 65]. Since the noisy video input affects the histogram appearance some methods suggest frame filtering with homomorphic [79] or a simple smoothing filter [78]. The homomorphic filter clearly outmatches the smoothing as it moderates the illumination and reflectance variation across the scene. The number of fames that should be stored to successfully model the background varies from 45 frames reported in [78] to 750 frames reported in [66]. A common way to update the background is to update the stored frames in a first-in-first-out order and reconstruct the histogram to find the new local maximum [78].

These are the basic background modelling techniques which are computationally efficient and were mostly developed for traffic surveillance systems where challenging scenarios such as dynamic or cluttered background are not so frequent. For the running median and average background modelling methods the back-

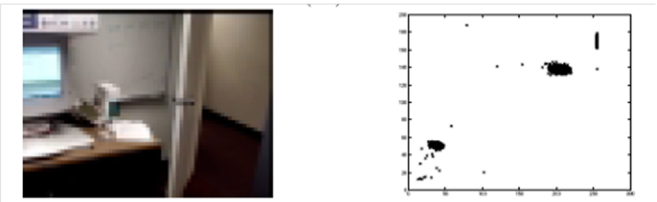


Figure 2.9: An image with a monitor that flickers and scatter plot of red and green components of a pixel on the screen [130].

ground is usually initialised with the first frame of the video sequence, which means that the foreground detection begins from the first frame. This brings up the issue of background initialisation to solve the problems that arise from an inaccurate initial background. To make the background model more robust to dynamic background scenarios and scene illumination changes, statistical background modelling methods have been developed and are described in the next section.

2.2.2 Statistical Background Modelling

The statistical background modelling is a reliable and well studied method for modelling the background, as it collects and updates statistical information over time. It is comprised of methods involving background modelling with single or mixture of Gaussian distributions and their improved and generalised versions, non-parametric density estimation for background modelling, usage of machine learning (SVM or SVR) to determine the function that models the background, subspace learning using PCA or Independent Component Analysis (ICA). Since each of these approaches features a large number of publications, it would consume many pages and effort to describe them all. Therefore only the techniques directly related to this research will be presented in the following sections.

Background Modelling with Gaussian Distributions

The background modelling by a single Gaussian is closely related to background modelling with a running average of N frames. C. Wren et al. [147] proposed to model the history of each colour component for a pixel by a single Gaussian distribution, to handle unstable backgrounds. To avoid storing N frames at all times, the parameters (mean and variance) of the Gaussian distribution are updated according to the following equations,

$$\mu_t(x, y) = aI_t(x, y) + (1 - a)\mu_{t-1}(x, y) \tag{2.6}$$

$$\sigma_t^2(x,y) = a(I_t(x,y) - \mu_t(x,y))^\top (I_t(x,y) - \mu_t(x,y)) + (1-a)\sigma_{t-1}^2(x,y) \quad (2.7)$$

where I_t is the current pixel value, μ_t is the current mean, σ_t^2 is the current variance and a is the learning rate. The higher the a, the faster the background will be updated. A pixel belongs to the background if its current value I_t is within a certain distance from the mean, otherwise it belongs to foreground. This method can be applied on various colour spaces and covers backgrounds where gradual illumination changes take place.

Imagine a situation where the background is dynamic and involves more than one colour, such as waving tree leaves, where the green leaves and blue sky occur interchangeably. Another example is a flickering screen as shown in Figure 2.9. The scatter plot of the red and green channels of a pixel for a period of time show that there are two separate distributions that describe the history of pixel values. This leads to the conclusion that a single Gaussian is not sufficient to model the background. N. Friedman and S. Russell [46] first noticed that modelling the background as a Mixture of Gaussian (MoG) distributions would solve shadow problems in traffic surveillance. They assigned shadows, road and cars to three different Gaussian distributions where their parameters were updated with incremental Expectation Maximisation (EM) method to avoid storing N recent frames that a traditional EM requires.

C. Stauffer and W. Grimson (SG) [130, 129] generalised the approach of N. Friedman and S. Russell to cover dynamic background scenarios. The background is modelled with up to 5 Gaussian distributions that are updated with on-line K-means approximation instead or EM. The Gaussian Mixture Model (GMM) exhibits a learning capability that would assist except for modelling backgrounds changing lighting conditions, dynamic background scenes with periodic motion, such as waving trees or sea waves. Their algorithm became a state-of-the art that has been improved in various ways by the research community. A detailed description of their approach is given in section 3.2.

A summary of Most Important Methods for Statistical Background Modelling

Following a chronological order P. KaewTraKulPong and R. Bowden in [68] reintroduced the EM algorithm to update the parameters of the MoG and suppressed the shadow by thresholding a colour distortion metric. A similar shadow removal technique was presented by T. Horprasert et al. in [60] who proposed a statistical non Gaussian background learning method. Later the same authors with A. El-

gammal proposed no-parametric background modelling by using a kernel density estimation [40], which became very popular and was improved by Z. Zivcovic in [161] in terms of memory efficiency and accuracy.

T. Bowmans in his survey on "Background Modelling using Mixture of Gaussians for Foreground Detection" summarises and compares the aforementioned methods and many others which aim to improve the GMM [12]. Among them the most important ones will be discussed in the next lines. Similar to A. Elagmmal, A. Mittal and N. Paragios used an adaptive kernel density estimation and optical flow for motion based background subtraction [96]. They introduced normalised colour representation, which was later adopted by H. Wang and D. Suter in [139], where the authors tackled some practical issues of the GMM and developed in [140] the first integrative system, SACON, that handles a variety of situations where the original GMM fails. In [63] O. Javed et al. introduced gradient information into GMM to remove spurious foreground objects created by illumination change, ghosts or shadows. The paper of F. Kristensen et al. [77] studied the impact of colour space selection on background modelling, and concluded that YCbCr colour model gives the best results. Spatial and colour coherency was taken into account by authors in [150] and in ASTNA system of M. Cristani, V. Murino in [29]. MRF based approach described in [121] was tested by T. Bowmans in his survey and was found to cause the least errors.

However, after resolving all the primary weaknesses of the GMM, the need for more complex methods that would deal with more challenging dynamic scenes became more prominent. Two of the early approaches for dynamic scene modelling are presented by H. Yang et al. in [149], and J. Zhang and C. Chen in [154]. The former paper suggests the combination of two GMMs for modelling the gradually and suddenly changing pixel values, while the latter one utilises a support vector machine (SVM) classifier over a set of statistical features such as mean, standard deviation and correlation to decide if the pixel belongs to the dynamic background or not. Other methods for background modelling of dynamic scenes is of Z. Wang et al. in [142], a generalised SG background subtraction of A.B. Chen and V. Mahadevan in [20], and Type-2 fuzzy GMM of T. Bowmans and F. El Baf in [11].

2.2.3 Strategies Followed to Improve the Results of Background Modelling

Several strategies can be followed to improve the original results of the GMM. These are the employment of MRFs for accurate noise free segmentation, maintenance of multiple backgrounds to recover quickly the background when the inserted static objects start moving and the utilisation of texture features to model

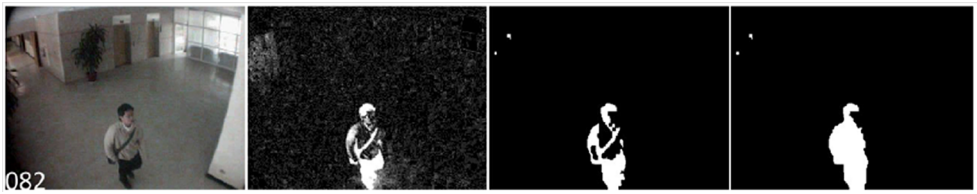


Figure 2.10: From left to right: the first image is the current frame, the second image is the probability map as a result of MoG background modelling, the fourth image is the thresholded probability map, and the fourth image is the segmentation result from MRF framework described in [22].

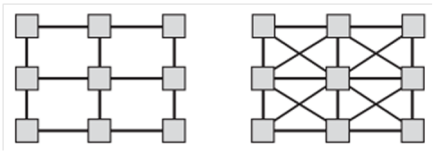


Figure 2.11: First and second order MRF with 4-neighbour and 8-neighbour connectivity respectively.

dynamic backgrounds.

MRF for Accurate Foreground Segmentation

The reason behind using the MRF method is because it imposes a kind of spatial interactions between the pixels which are modelled as isolated entities. Spatial dependence favours smooth foreground/background segmentation without noise specks and with complete foreground shapes. The MRF framework can be applied to almost any type of background model which is expressed in terms of probabilities. An example image, taken from [22], that displays the effects of MRF on foreground/background segmentation in contrast with a simple thresholding is shown in Figure 2.10.

The process of applying MRF for foreground/background segmentation is well explained by K. Schindler and H. Wang [121], and by Y. Sun et al. [131]. A current frame can be considered as an MRF which consists of an undirected graph with frame pixels $X = \{x_1 ... x_n\}$ considered as nodes and a neighbourhood system $\{N_1 ... N_n\} = N \subset X$ with N_i the set of neighbours that surround each node x_i. A set of labels $L = \{l_1 ... l_k\} = \{\text{“}bg\text{”}, \text{“}fg\text{”}\}$ for a background/foreground segmentation problem are the potential assignments to each node. Supposed $f = \{f_1, ..., f_n\}$ is a set of labelling configurations defined on the lattice X such that $f_i = f(x), \forall x \in X$, then f_i can be regarded as a function that assigns a label $l \in L$ to the pixel $x \in X$, [86, 71]. Let $F = \{F_1, ..., F_n\}$ be a set of random variables defined in the set X where each variable F_i can take a value f_i in L. The probability that a random variable F_i takes the value f_i is denoted by $P(F_i = f_i)$

or else $P(f)$. The $P(f)$ follows Gibbs distribution and as such $P(f) \propto e^{-E(f)}$, where $E(f)$ can be written as follows:

$$E(f) = \sum_{x_i \in X} D_i(f_i) + \sum_{x_i, x_j \in N} V_{i,j}(f_i, f_j) \tag{2.8}$$

where D_i is the data cost function, and $V_{i,j}$ is the smoothness cost or clique potential function that defines the interaction between the neighbouring pixels. The neighbourhood of a pixel can be regarded as 4-connectivity or 8-connectivity with cliques defined over 2 or three pixels respectively as shown in Figure 2.11. The configuration that minimises the energy function $E(f)$ is the one that solves the segmentation problem by labelling.

Let us take the basic example of background modelling with a running average, where the foreground F can be segmented by thresholding the difference between the current frame I_t and the background model B_t with a threshold T as follows:

$$F(x,y) = \begin{cases} 1 & if\ |I_t(x,y) - B_t(x,y)| < T \\ 0 & otherwhise \end{cases} \tag{2.9}$$

The smoothness cost in its simplest form can defined as follows:

$$V_{i,j}(f_i, f_j) = \begin{cases} s & if \quad f_i \neq f_j \\ 0 & if \quad f_i = f_j \end{cases} \tag{2.10}$$

where the cost is $s = aT$ constant for all neighbouring nodes with different labels and 0 otherwise. The cost could also vary depending on colour similarity between the neighbouring pixels as in the approach of L.-Y. Chang and W.H. Hsu [22]. The data cost function can be defined as class conditional probability function as follows

$$D_i(f_i) = \begin{cases} -\ln(p(I_t|f_i = \text{“}bg\text{”}) = |I_t(x,y) - B_t(x,y)|) \\ -\ln(p(I_t|f_i = \text{“}fg\text{”}) = T) \end{cases} \tag{2.11}$$

Sometimes to define the energy function several data costs and smoothness costs are accumulated [107, 22]. For example Y. Zhou et al. [159] formed an image pyramid and defined smoothness cost functions on cliques that are formed between two neighbouring pixels at the same scale, two corresponding pixels at adjacent resolution scales, and two corresponding pixels at two consecutive frames. S.-Y. Yang and C.-T. Hsu [150] combined a hybrid feature vector for background modelling with MRF and V. Reddy et al. [113] used MRF to define the cliques on an 8-neighbour connectivity node in a process to initialise a background model. A recent approach for foreground post-processing with probabilistic super pixel MRF is proposed by A. Schick in [120].

Background Modelling using Texture Features

Another possible way to model the background is using texture information. M. Heikkila and M. Pietikainen were the first to introduce texture features like Local Binary Paterns (LBP) for background modelling [58]. Later, the authors in [85] used the LBP in conjunction with a codebook representation and a single Gaussian in an attempt to model dynamic background. Other LBP implementations include the method proposed by S. Zhang et al. [155] where the authors employ a spatio-temporal LBP for dynamic background. Further a local dependency histogram based method proposed by S. Zhang et al. exhibits robustness in scenes with noise and dynamic background [156]. In [157] the same authors incorporated the LBP feature along with pixel coordinates and intensity values into a covariance matrix to ensure local dependency. R. Yumiba et al. employed a spatio-temporal texture named a Space-Time Patch to address illumination changes and dynamic background [152].

Multi Layer and Multi-modal Background Modelling

The traditional GMM approach treats all pixels equally and the mean and variance are updated with the same learning rate. H. Yang et al. in [149] presented the idea of maintaining two background models with different learning rates, with one monitoring gradual changes while the other sudden and quick changes. The model learns accurate means and variances, which help in reducing the holes in foreground when the foreground/background similarity is high. Following the above idea, R. Evangelio and T. Sikora maintained two complementary background models to detect the static foreground such as left-objects that would otherwise merge into the background model after some time [43]. A similar multi-layer background model was adopted by J. Yao and J.-M. Odobez [151] in combination with colour and LBP features to handle changes in background due to inserted objects.

2.2.4 Recent Approaches for Background Modelling

Some of the recently developed, most promising methods which are non GMM are discussed below. ViBe in [6] models the background with a set of samples rather than pixel probability distributions. The authors propose a background initialisation from a single frame where the samples of the model are taken from the neighbouring frames. Subsequently the background is updated randomly following a number of rules. Experimental results show substantial improvement compared to the state-of-the-art methods with respect to accuracy, memory, and time efficiency. In [137] M. Van Droogenbroeck and O. Paquot proposed signifi-

cant improvement of ViBe through several enhancement steps. They introduced significant noise reduction by hole filling and connected component processing, and foreground object outline maintenance by inhibiting background propagation. Furthermore, they suggest substituting the Euclidean distance measure by a colour distortion metric. The L. Maddalena and A. Petrosino in [88] successfully modelled dynamic background with a self-organizing neural network that learns the motion patterns. The method performs real time and is robust to illumination changes and cast shadows. M. Hofmann et al. proposed Pixel-Based Adaptive Segmenter (PBAS) which combines some of the characteristics of SACON and ViBe [59] while Y. Nonaca et al. integrated pixel level, region level, and frame level processing into their system [101]. They introduced Radial Reach Correlation to address the illumination changes and applied kernel density estimation for pixel based background modelling.

2.3 Summary and Discussion

This chapter summarised the state-of-the-art and most commonly used approaches for COD and background modelling for foreground object segmentation. By examining the literature on COD it became obvious that the results of COD directly depend on foreground object segmentation.

A general framework that is comprised of a number of standard steps, on how to detect a carried object given a video sequence was presented. Most of the studied COD methods, including the state of the art algorithm proposed by D. Damen and D. Hogg [34, 33] rely on a combination of silhouette and gait analysis algorithms with the use of temporal templates. Another important method explained is the representation of the human silhouettes during a walking cycle as a 3-dimensional manifold to recover their natural shape in the presence of a carried object, which is detected as an outlier.

The background modelling methods presented relate to basic and statistical background modelling techniques. Special importance was given to background modelling with a mixture of Gaussian distributions as it is the most popular and widely studied method due to its real time performance, and low memory requirement. The improvements proposed to the MoG were summarised and general strategies to improve segmentation accuracy and robustness of the algorithm were presented.

The chapter closes with a summary of the most recent background modelling methods.

Chapter 3

Theoretical Preliminaries

This chapter presents all theoretical preliminaries that are required to understand the novel concepts and methodologies proposed in the chapters that follow. It begins with introductions to the state-of-the-art COD method of D. Damen and D. Hogg [34] and the well-established foreground/background segmentation technique of C. Stauffer and W. Grimson [130], presented in sections 3.1 and 3.2, respectively. These two approaches are used as benchmark algorithms in the proposed research and many of their attributes are adopted for the development of novel algorithms. In addition, section 3.3 presents the formation of an image gradient vector and section 3.4 includes the definition of steerable filters. One further important concept introduced and used later in chapter 4 is *diffusion* in image processing, which is difficult to comprehend when not presented as originating from the foundations of Physics. Therefore, section 3.5 introduces diffusion based on its definition in Physics and extends it to the definition and applications in image processing (see section 3.6). Further topics subsequently presented relate to diffusion are the structure tensor explained in section 3.7 and a coherence enhancing anisotropic diffusion in section 3.8. Finally section 3.9 provides a summary and a discussion.

3.1 The state of the art COD Method

The work of D. Damen and D. Hogg [34] is based on creating a temporal template of a moving person and a further analysis of its properties to detect carried objects (see Figure 3.1). The concept of temporal textural template was first defined by I. Haritaoglu in [53] and it was described earlier in chapter 2. However, D. Damen prefers using a frequency temporal template also first referred by I. Haritaoglu [53]. Therefore, the baggage detector proposed by D. Damen takes as input a sequence of binary images which represent the foreground segmentation of a moving

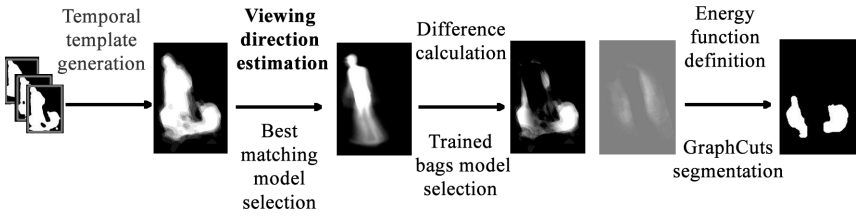


Figure 3.1: A summary of the procedure adopted by D. Damen and D. Hogg for baggage detection.

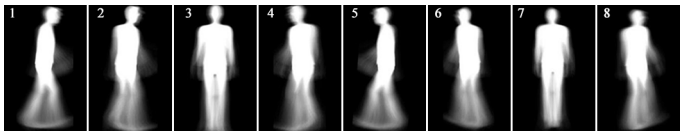


Figure 3.2: Exemplar temporal templates of 8 viewing directions.

person. These binary silhouettes are aligned with the help of an Iterative Closest Point (ICP) algorithm and averaged to create the temporal template. This temporal template is matched against a number of unencumbered temporal template models from a database to find the one that matches best. The database contains 728 unencumbered temporal template exemplars of 13 different sizes, facing at 8 different viewing directions (are shown in Figure 3.2), and 7 in-plane rotations. As expected, the difference of the two templates reveals the protruding regions which are likely to be carried objects. For further enhancement of accuracy, a trained model that maps the probable bag location is selected out of four view specific models to weigh the protruding regions. Finally, the carried objects are segmented via energy minimisation using Graph Cuts (see Appendix C).

Two important questions related to the approach described above arise, which are: How the best matching view specific exemplar is found and how the bags are segmented with the Graph Cuts.

Starting with the first question; it is common sense that the location of carried objects and their visibility depends on the position of the viewer in relevance to the position, and moving direction of the pedestrian. This fact brings the need of classifying the temporal template into 8 categories, according to its direction of motion. The procedure of direction estimation involves the transfer of motion from the image plane (the two-dimensional image surface) to the ground plane (which is at 90 degrees to the picture plane, commonly the ground that objects move on) via the homography transformation. Thus, by identifying the motion vector of the person and the position of the camera on the ground plane, it is possible to find the direction of motion. Therefore, the displacement along the x and y axis on image plane in combination with the average angle between the motion vector and the vector connecting the camera to the pedestrian on ground

plane will give the direction of motion. Knowing the viewing direction of the pedestrian the corresponding temporal template can be matched against exemplars of the same viewing direction. The exemplar that minimises the sum of absolute weighted differences between itself and the temporal template is confirmed as the best match.

The second question is associated with the segmentation of carried objects via Graph Cuts. According to Damen & Hogg [34] the difference image $v(x, y)$ between the best matching exemplar and the temporal template can be considered as a first-order Markov Random Field (MRF) and an energy function is minimised to determine which spatial locations ($\mathrm{x} = (x, y)$) on the MRF belong to the carried objects ($m_{\mathrm{x}} = 1$) and which to noise ($m_{\mathrm{x}} = 0$). They express the energy function by the following equation

$$E(m) = \sum_{\mathrm{x} \in I} (\phi(v|m_{\mathrm{x}}) + \omega(m_{\mathrm{x}}|\theta_d)) + \sum_{\mathrm{x},\mathrm{z} \in C} \psi(m_{\mathrm{x}}, m_{\mathrm{z}}) \tag{3.1}$$

where

$$\phi(v|m_{\mathrm{x}}) = \begin{cases} -\log(p(v|m_{\mathrm{x}} = 1)) & if\ m_{\mathrm{x}} = 1 \\ -\log(p(v|m_{\mathrm{x}} = 0)) & if\ m_{\mathrm{x}} = 0 \end{cases} \tag{3.2}$$

are the class-conditional probability functions based on $v(x, y)$ that express the cost of assigning a label to location $\mathrm{x} = (x, y)$ and

$$\omega(m_{\mathrm{x}}|\theta_d) = \begin{cases} -\log(\theta_d(\mathrm{x})) & if\ m_{\mathrm{x}} = 1 \\ -\log(1 - \theta_d(\mathrm{x})) & if\ m_{\mathrm{x}} = 0 \end{cases} \tag{3.3}$$

are the prior probabilities θ_d given the direction d. The smoothness cost function $\psi(m_{\mathrm{x}}, m_{\mathrm{z}})$ that defines the interaction between pairs of neighbouring pixels C in the image I is expressed as,

$$\psi(m_{\mathrm{x}}, m_{\mathrm{z}}) = \begin{cases} \lambda & if\ m_{\mathrm{x}} \neq m_{\mathrm{z}} \\ 0 & if\ m_{\mathrm{x}} = m_{\mathrm{z}} \end{cases} \tag{3.4}$$

The class conditional p.d.f are defined by Damen & Hogg [34] as,

$$p(v|m_{\mathrm{x}} = 1) = \gamma \mathcal{N}(\nu; 0.6, 0.3) + (1 - \gamma)\mathcal{N}(\nu; 1.0, 0.05) \tag{3.5}$$

$$p(v|m_{\mathrm{x}} = 0) = \frac{1/(\nu + 0.01)}{\log(1 + 0.01) - \log(0.01)} \tag{3.6}$$

In their experiments $\gamma = 0.65$ and $\lambda = 2.3$ which are optimised over two different video sequences. Subsequently, the energy function is minimised via Graph Cuts and labels are assigned to every pixel $\mathrm{x} = (x, y)$. The intermediate

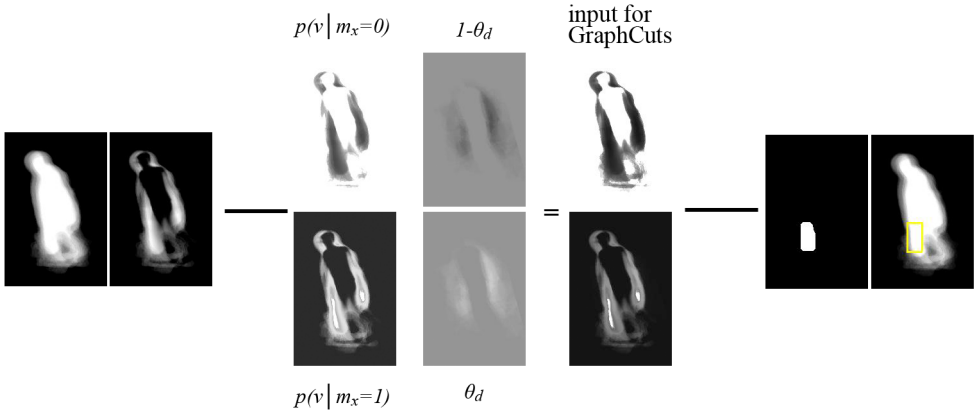


Figure 3.3: The assignment of labelling costs on spatial locations x = (x, y) of the difference image $v(x, y)$ and final carried object segmentation. Starting from left to right are the temporal template, the difference image, the costs assigned to the spatial locations, the prior probability view specific models, the input to Graph Cuts as noise probabilities and carried object probabilities, the output of Graph Cuts and finally the labelled temporal template.

and final results are illustrated in Figure 3.3.

3.2 Background Modelling with Mixture of Gaussian Distributions (Stauffer & Grimson method)

The revolutionary approach for background modelling with a mixture of Gaussian distributions of C. Stauffer and W. Grimson in [130, 129] has been used extensively throughout the years as a stepping stone for the development of novel algorithms. The stable performance of the method under various scenarios such as dynamic background and gradual illumination changes made it extremely popular. According to Stauffer & Grimson each pixel in the frame is modelled by weighted Mixture of Gaussian (MoG) distributions whose parameters and weights are continuously updated over time. This allows the prediction of pixel values that belong to the foreground/background in successive frames based on probability. The history $\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$ of the values of a pixel in frames I is modelled by a mixture of K Gaussian distributions. X_t could be a scalar if the pixel has one greyscale intensity value or a vector if the pixel has three colour components. The probability of observing X_t at time t is

$$P(X_t) = \sum_{k=1}^{K} \omega_{k,t} * \eta(X_t, \mu_{k,t}, \Sigma_{k,t}) \tag{3.7}$$

where $\omega_{k,t}$ are the coefficients of each distribution that act as weights, $\mu_{k,t}$,and $\Sigma_{k,t}$ are the mean value and the covariance matrix of the k^{th} Gaussian distribution. The Gaussian function η is defined as

$$\eta\left(X_t, \mu_t, \Sigma_t\right) = \frac{1}{(2\pi)^{n/2}|\Sigma_{k,t}|^{1/2}} e^{-\frac{1}{2}(X_t-\mu_t)^\top \Sigma_t^{-1}(X_t-\mu_t)} \tag{3.8}$$

Stauffer and Grimson assume that the R, G, and B components are independent and have the same variances and thus the covariance matrix is

$$\Sigma_{k,t} = \left(\sigma_k^2 \mathbf{I}\right) \tag{3.9}$$

where $\mathbf{I}$ is the identity matrix and σ_k^2 is the variance of the k^{th} distribution. Every new scalar or vector X_t is checked against the existing K distributions via absolute distance to find a match. A match is found if the following inequation is true

$$|X_t - \mu_{k,t-1}| \leq 2.5\sigma_{k,t-1} \tag{3.10}$$

If a match is found then the k^{th} Gaussian that is found to be a match, is updated following the next equations.

$$\begin{aligned} \mu_{k,t} &= (1-\rho)\mu_{k,t-1} + \rho X_t \\ \sigma_{k,t}^2 &= (1-\rho)\sigma_{k,t-1}^2 + \rho\left(X_t - \mu_{k,t}\right)^\top \left(X_t - \mu_{k,t}\right) \\ \omega_{k,t} &= (1-\rho)\omega_{k,t-1} + a(r_{k,t}) \end{aligned} \tag{3.11}$$

where $\rho = a(\eta X_t|\mu_k, \sigma_k)$ is a learning factor, a is the learning rate and $r_{k,t}$ is 1 for the model which has been matched and 0 for the remaining models. In case that no matching distribution is found, the pixel is labelled as foreground and the least probable distribution is substituted by a new one, initialised with prior parameters for $\omega_{k,t}$ and $\sigma_{k,t}^2$, and $\mu_{k,t} = X_t$.

To decide which portion of the distributions represents the foreground and which the background it is important to put them in an order according to the ratio ω/σ. Then the first B distributions, chosen according to the following formula, account for the background.

$$B = arg\min_b \left(\sum_{k=1}^{b} \omega_{k,t} > T\right) \tag{3.12}$$

where T is the threshold that defines the portion of distributions that are accounted for by the background. A pixel is labelled as background if the current X_t matches any of the first B distributions. In the opposite case the pixel is labelled as foreground.

This algorithm will be used in the next chapter by slightly modifying the equations that update the parameters and the distance measure according to the requirements of the proposed framework.

3.3 Gradient Vector

Let us begin with the definition of gradient and directional derivatives. The gradient of a scalar point function $f(\mathbf{r}) \equiv f(x, y, z)$ is a vector point function defined at each point $\mathbf{r} \equiv (x, y, z)$, where $f(\mathbf{r})$ is suitably differentiable. Thus, $grad(f(\mathbf{r})) \equiv \nabla f \equiv < \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} > \equiv \mathbf{i}\frac{\partial f}{\partial x} + \mathbf{j}\frac{\partial f}{\partial y} + \mathbf{k}\frac{\partial f}{\partial z} = \mathbf{F}$ where, $\mathbf{i}, \mathbf{j}, \mathbf{k}$, are unit vectors in the rectangular Cartesian-coordinate system. The vector $\mathbf{F}$ has magnitude $|\nabla f| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2 + (\frac{\partial f}{\partial z})^2}$ and points to the direction indicated by the unit vector $m = \frac{\nabla f}{|\nabla f|}$, where the directional derivative $D_m f(\mathbf{r})$ in direction m at the point $\mathbf{r}$ has the greatest value among all the directional derivatives $D_u f(\mathbf{r})$ at the same point [72] .

3.4 Oriented (Steerable) Filters

P. Cheng-San Teo in his thesis [135] identifies two main streams in the composition of steerable filters. These are the numerical approach and analytical approach. The former treats the problem numerically and computes the optimal number of basis filters and steering coefficients. The latter defines the functions that are analytically steerable and proposes analytical way of writing the linear combination of the basis functions derived.

The foundations of numerical approach were set by P. Perona in his work "Deformable kernels for early vision" [109]. His technique is based on singular value decomposition to compute small number of basis functions to be combined in a filter of arbitrary rotation. P. Peronas work was continued by R. Manduchi in [91, 92] and D. Shy in [125].

The analytical approach that was introduced by T. Freeman and E. H. Adelson in [45] proposes a method for design of steerable filters as linear combination of basis filters. They find analytically the minimum number of basis functions required to synthesise a filter at any orientation, by expanding the function as Fourier series and analysing the frequencies that occur.

According to [45] the steerable filters are defined as "a class of filters in which a filter of arbitrary orientation is synthesised as a linear combination of a set of basis filters". This statement is in agreement with the definition of the directional derivative, which states that the rate of change of a function of several variables in

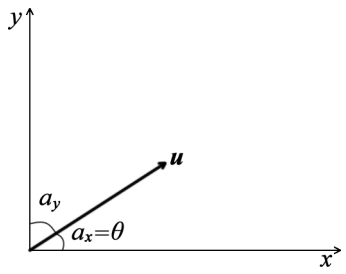


Figure 3.4: The angles between the vector and the x and y axis.

the direction pointed by unit vector $\mathbf{u} =< u_1, u_2, u_3 >$ equals to the dot product of the gradient and the vector $\mathbf{u}$. i.e.

$$D_u f = \nabla f \mathbf{u} = \frac{\partial f}{\partial x} u_1 + \frac{\partial f}{\partial y} u_1 + \frac{\partial f}{\partial z} u_3 = \cos(a_x)\frac{\partial f}{\partial x} + \cos(a_y)\frac{\partial f}{\partial y} + \cos(a_z)\frac{\partial f}{\partial z} \quad (3.13)$$

where, a_x, a_y, a_z are the angles between the vector $\mathbf{u}$ and the positive x, y and z axis, respectively [72]. In two dimensional case of a function $f(x, y)$ and a vector $\mathbf{u} =< u_1, u_2 >$ the directional derivative equation can be written respectively as

$$D_u f = \nabla f \mathbf{u} = \frac{\partial f}{\partial x} u_1 + \frac{\partial f}{\partial y} u_1 = \cos(a_x)\frac{\partial f}{\partial x} + \cos(a_y)\frac{\partial f}{\partial y} = \cos(\theta)\frac{\partial f}{\partial x} + \sin(\theta)\frac{\partial f}{\partial y} \quad (3.14)$$

where, θ is the angle between the vector $\mathbf{u}$ and the positive x axis (see Figure 3.4) [72, 119]. The partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are regarded as basis functions and $\cos(\theta)$ and $\sin(\theta)$ as the steering coefficients (according to the theory of steerable filters by T. Freeman and E. H. Adelson [45]).

Following the concept of directional derivatives the authors in [45] impose the basic steering constraint which is represented by the following formula

$$f^{\theta}(x, y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x, y) \quad (3.15)$$

where, $f^{\theta_j}(x, y)$ are M basis filters and $k_j(\theta)$ are the interpolation functions that linearly combine the basis filters.

An important attribute of spatial filters is separability. Spatially separable filters are computationally efficient since the convolution with a two dimensional filter can be substituted by a sequence of one dimensional convolutions; one in vertical direction followed by one in the horizontal direction. This means that the following statement holds:

$$I(x, y) * g(x, y) = (I(x, y) * g(x)) * g(y) \quad (3.16)$$

Every function that can be written as a polynomial in x and y has separable

basis functions that might be large in number. In [45] it is shown how to find the $x-y$ separable basis functions and functional forms of separable filters for Gaussian derivatives of up to fifth order are provided.

Assuming that the 2-D Gaussian function to be used is $G(x,y) = e^{-(x^2+y^2)}$ then the basis set for the first order derivative of Gaussian consists of two directional derivatives $G^{0^\circ} = \frac{\partial G}{\partial x} = -2xe^{-(x^2+y^2)}$ and $G^{90^\circ} = \frac{\partial G}{\partial y} = -2ye^{-(x^2+y^2)}$ and the steering condition in Equation 3.15 is expressed as:

$$G^{\theta} = \cos(\theta)G^{0^\circ} + \sin(\theta)G^{90^\circ} \tag{3.17}$$

3.5 Diffusion of Matters and Heat Conduction

Diffusion is characterised as the tendency of molecules of one substance to spread from areas of higher concentration to areas of lower concentration. At some point in time the concentrations produce equilibrium and the movement of molecules continues at constant rate. One of the first scientists to experiment with diffusion of gasses was Thomas Graham as referred to by [110]. However, A. Fick [44] spotted that in a publication of Graham on diffusion of salts in water a fundamental law for the operation of diffusion of fluids was missing. Following the Fouriers law of heat conduction, A. Fick realised a relation of heat condition, where the driving force for heat transfer is the temperature difference, with diffusion.

Let us revise the fundamental laws of heat conduction as developed by J. Fourier. The **Fouriers law of heat conduction** states that "the rate of heat conduction through a plane layer is proportional to the temperature difference across the layer and the heat transfer area, but is inversely proportional to the thickness of the layer" [41], which in one dimensional conduction can be expressed by the following equation:

$$Q = -kA\frac{T_1 - T_2}{\Delta x} = -kA\frac{\Delta T}{\Delta x} \Rightarrow^{\Delta T \to 0} Q = -kA\frac{dT}{dx} \tag{3.18}$$

In three dimensional case of heat transfer, the above equation becomes

$$\overrightarrow{\mathbf{Q}} = -k\left(A_x\frac{\partial T}{\partial x}, A_y\frac{\partial T}{\partial y}, A_z\frac{\partial T}{\partial z}\right) \tag{3.19}$$

where A is the area (normal to the direction of heat transfer), k is the thermal conductivity of the material and $\frac{dT}{dx}$ is the temperature gradient. Since the temperature gradient is the slope of the temperature curve and given that the heat is transferred from hotter to cooler areas gradient will be a negative quantity. Therefore the negative sign in Equation 3.19 ensures that the heat transfer along the

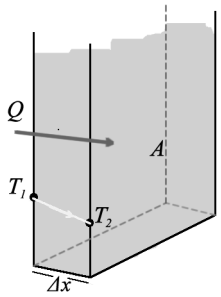


Figure 3.5: Heat transfer in a medium.

direction of positive x-axis is a positive quantity.

The heat transfer problems could be characterised as **steady** or **transient**. The term steady implies that there is no change in temperature condition of a given point within the medium over time, although there might be temperature variations across the medium. On the other hand if the temperature at any fixed point of the medium changes with time during heat conduction then it is called transient (or unsteady) heat conduction (the temperature varies with time as well as position). Here we are mostly interested in the transient heat conduction.

One could raise a question on how to determine the variation of temperature within the medium or the temperature value at a given location x at time point t, as until now no time was included in the Equation 3.18. A more complete version of the heat conduction equation in one dimensional transient case where there is no heat generation is expressed by (also called diffusion equation)

$$\frac{\partial T}{\partial t} = a\frac{\partial^2 T}{\partial x^2} \tag{3.20}$$

where a is the thermal diffusivity of the material dependant on thermal conductivity k of the material and the heat capacity (the capability of the material to store heat). Note that here k does not depend on the location x in the medium. The above equation is a result of heat energy conservation law in an element that can be expressed as "the change in rate of heat conduction over an interval Δx equals to the rate of change of heat energy content of the element over time". The complete derivation of the Equation 3.20 can be found in [41].

The heat conduction is said to be two-dimensional when conduction in the third dimension is negligible and is expressed by

$$\frac{\partial T}{\partial t} = a\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) = a(\nabla^2 T) \tag{3.21}$$

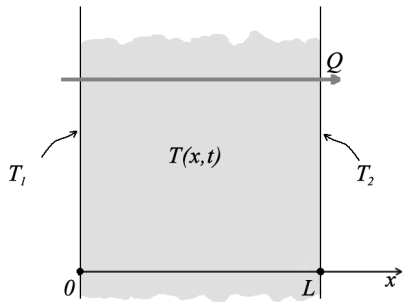


Figure 3.6: The boundary conditions at time t for a medium of thickness L are $T(0,t) = T_1$ and $T(L,t) = T_2$. If $T_1 > T_2$ then the heat is transferred towards the colder temperature which is in direction of positive x-axis.

To completely define a heat transfer problem and to ensure the uniqueness of the solution to the heat equation it is important to identify the **boundary conditions** at each side of the medium. For one dimensional problems as in Figure 3.6, the temperature $T(x,t)$ is a function of time and location and two boundary conditions are required (for two dimensional problems we would need four boundary conditions). Therefore, at time t the boundary condition at location $x = 0$ is $T(0,t) = T_1$ and at location $x = L$, which is the width of the medium, is $T(L,t) = T_2$. When the measurement of heat transfer begins the initial distribution of the temperature in the medium (condition of the medium) must also be known: So the **initial condition** at time $t = 0$ is $T(x,0) = f(x)$.

Fick spotted the analogy between heat and mass transfer [44]. As the rate of heat conduction is proportional to temperature gradient likewise the rate of mass transfer (or diffusive **mass flux**) is proportional to the concentration gradient. In one dimensional case this can be expressed by the equation

$$j = -D\frac{du}{dx} \tag{3.22}$$

where u is the concentration of matter in a mixture, x is the location and D is the diffusion coefficient (or mass diffusivity) which shows how fast the substances diffuse into each other and depends on the nature of the substances. Applying the law of mass conservation, which implies that in a closed system, mass can neither be created nor destroyed, A. Fick came up with his second law of diffusion expressed by the following equation:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} = D\nabla^2 u \tag{3.23}$$

The above equation shows how the concentration u changes with time due to the

diffusion process. Details on how the above equation is derived can be found in the "A heat transfer textbook" of J. H. Lienard [67]. If the diffusion coefficient D is not a constant but depends upon the location or concentration then it cannot be taken out of the derivative and the equation becomes

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right) \tag{3.24}$$

For two- or three- dimensional cases the derivatives can be replaced by gradient operators and thus it will be

$$\frac{\partial u}{\partial t} = \nabla\cdot(D\nabla u) = \text{div}\,(D\nabla u) = \frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(D\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(D\frac{\partial u}{\partial z}\right) \tag{3.25}$$

Similar boundary conditions are applied here to ensure unicity of the solution.

3.6 Diffusion in Image Processing

Physically the diffusion process in images resembles the heat conduction in materials. Therefore the researchers usually refer to the diffusion equation as the heat equation. As heat is transferred through a mass tending to reach a state where all the molecules have the same temperature, the image pixel values change according to their surrounding ones until homogeneity is achieved. At the same time the pixel values remain within the maximum and minimum limits, imposed by the original image, which are usually from 0 to 255. J. Weickert [143] successfully defines the intuitive perception of diffusion as "physical process that equilibrates concentration differences without creating or destroying mass".

The first researchers to use the concept of diffusion in image processing were P. Perona and J. Malic in their well-known publication "Scale-Space and Edge Detection Using anisotropic Diffusion" [108]. They proposed a non-linear diffusion process for inhomogeneous image smoothing and edge detection. The diffusion equation they adopted is as follows:

$$\begin{cases} \frac{\partial I(x,y,t)}{\partial t} = \nabla\cdot(c(x,y,t)\nabla I(x,y,t)) = \text{div}(c(x,y,t)\nabla I(x,y,t)) \\ I(x,y,0) = I_0(x,y) \end{cases} \tag{3.26}$$

where $I_0(x,y)$ is the original image, $I(x,y,t)$ is an image smoothed by a Gaussian kernel with size of variance equal to t (time or scale) and $c(x,y,t)$ is a scalar valued diffusivity factor. The simplest case of diffusivity is a constant value which leads to homogeneous diffusion or more commonly known as Gaussian blurring, uniform through the space. To achieve inhomogeneous diffusion which promotes smoothing

within the region while slowing down the diffusion across the boundaries, the diffusivity should be large within the regions and small along the edges. The simplest version of diffusivity in this case is a binary valued function with 1 within the region and 0 at the boundaries. However P. Perona and J. Malik proposed diffusivity $c(x, y, t)$ as a monotonically decreasing function (from 1 to 0) of the gradient norm of the image $I(x, y, t)$, typically defined as

$$c(x, y, t) = g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2} \tag{3.27}$$

This means that diffusion is promoted in regions with small gradients and is inhibited in areas with high gradients that corresponds to edges. Since the smoothing at edges is delayed, the noise survives for a longer period of time.

In their publication Perona and Malik wrongly defined the inhomogeneous diffusion as anisotropic, which was later corrected by J. Weikert as inhomogeneous diffusion since the diffusivity factor is a scalar valued function. He suggested that in true anisotropic diffusion the diffusivity factor should be a structure tensor. Isotropic and anisotropic diffusion along with all the related aspects of PDE based image smoothing techniques was investigated extensively by J. Weikert in [143].

3.7 The Structure Tensor

Structure tensor is a positive semi-definite symmetric matrix that holds information on the orientation and intensity of the surrounding structure of an image u. To make the tensor invariant to small variations caused by noise the image is smoothed by a Gaussian kernel K_σ with standard deviation σ producing u_σ. The diffusion tensor matrix is defined as

$$\begin{aligned} J_0(\nabla u_\sigma) = \nabla u_\sigma \otimes \nabla u_\sigma = \nabla u_\sigma \nabla u_\sigma^\top = &< \tfrac{\partial u_\sigma}{\partial x}, \tfrac{\partial u_\sigma}{\partial y} >< \tfrac{\partial u_\sigma}{\partial x}, \tfrac{\partial u_\sigma}{\partial y} >^\top = \\ &= \begin{bmatrix} \left(\frac{\partial u_\sigma}{\partial x}\right)^2 & \frac{\partial u_\sigma}{\partial x}\frac{\partial u_\sigma}{\partial y} \\ \frac{\partial u_\sigma}{\partial x}\frac{\partial u_\sigma}{\partial y} & \left(\frac{\partial u_\sigma}{\partial y}\right)^2 \end{bmatrix} \end{aligned} \tag{3.28}$$

where ∇u_σ is the gradient of image u_σ.

An $m \times m$ symmetric matrix $\mathbf{A}$, as $J_0(\nabla u_\sigma)$, is a positive semi-definite matrix if for any vector $\mathbf{x} \neq 0$ the quadratic form $\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0$. Also if $\mathbf{B}$ is an $n \times m$ matrix then $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ is positive semi-definite and possesses some important properties. This matrix is always symmetric and its eigen-decomposition always exists, and has a particularly convenient form: the eigenvalues are always positive or null and the corresponding eigenvectors are pairwise orthogonal when their

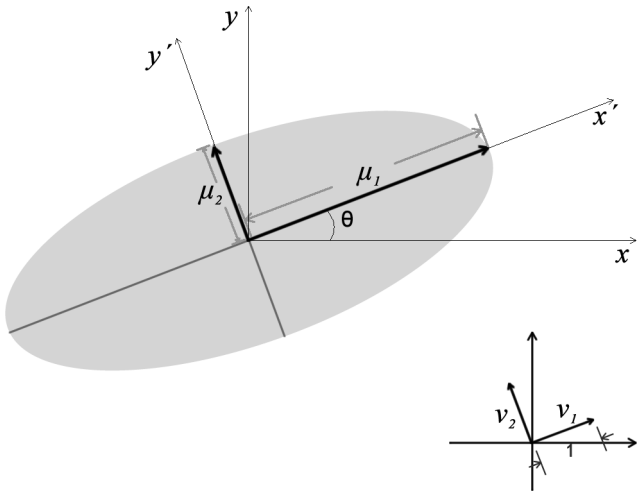


Figure 3.7: The structure tensor and its eigenvectors.

eigenvalues are different. Figure 3.7 shows an example of structure tensor with eigenvalues $\mu_1 \geq \mu_2$ and corresponding eigenvectors (unit vectors) $v_1 = [v_{1a}\ v_{1b}]$ and $v_2 = [v_{2a}\ v_{2b}]$. Supposed that

$$Q = \begin{bmatrix} v_{1a} & v_{2a} \\ v_{1b} & v_{2b} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \ and\ \Lambda = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \tag{3.29}$$

matrix $\mathbf{Q}$ is also called a rotation matrix where the angle θ shows the amount that the xy coordinate system must be rotated counter-clockwise to coincide with the system $v_1 v_2$. For a positive semi-definite matrix $\mathbf{A}$ it is also valid that:

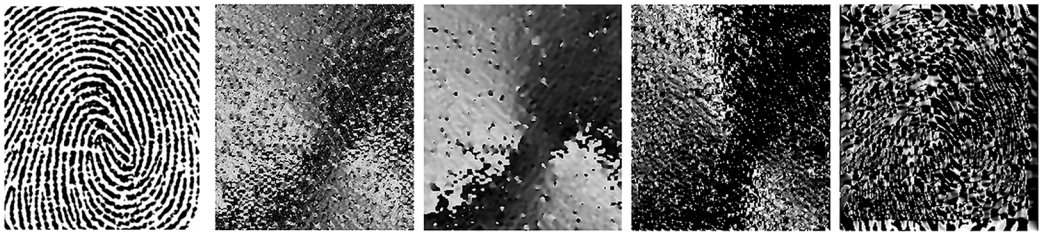
$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top \Leftrightarrow \mathbf{\Lambda} = \mathbf{Q}^\top\mathbf{A}\mathbf{Q} \tag{3.30}$$

This shows that the matrix $\mathbf{A}$ can be transformed into an equivalent diagonal one. This process is often referred as diagonalisation [57].

3.8 Coherence Enhancing Anisotropic Diffusion

A complete guide on anisotropic diffusion in image processing was written by J. Weikert in his book "Anisotropic Diffusion in Image Processing" [143].

One of the most important publications of J. Weickert that is in particular interesting is the one that studies the enhancement of flow-like patterns (e.g. fingerprints) via coherence enhancing filtering [144]. The author proposes the use of non linear tensor diffusion process with a structure tensor as diffusivity function, thereby promoting anisotropic behaviour. When Ω is an image space



(a) (b) (c) (d) (e)

Figure 3.8: The average orientation of eigenvectors (in (b) $\sigma = 1$ and in (c) $\sigma = 5$) and the average direction of gradients (in (d) $\sigma = 1$ and in (e) $\sigma = 5$) of the fingerprint image.

and $\partial\Omega$ is the boundary of the image, the diffusion process is described by the following equations [144]

$$\begin{aligned} \frac{\partial u(\mathbf{r},t)}{\partial t} &= \operatorname{div}\left(D(u_\sigma(\mathbf{r},t),\mathbf{r})\nabla u_\sigma(\mathbf{r},t)\right) && \text{on } \Omega \times (0,\inf) && \text{Diffusion equation} \\ (D(\nabla u_\sigma)\nabla u_\sigma)\cdot\mathbf{n} &= 0 && \text{on } \partial\Omega && \text{Neumann boundary} \\ u(\mathbf{r},0) &= f(\mathbf{r}) && \text{on } \Omega && \text{Initial condition} \end{aligned} \tag{3.31}$$

where u_σ is the image (density) smoothed with a Gaussian kernel of scale σ, $\mathbf{r} = (x, y)$ is the location and t is time. $D(\nabla u_\sigma(\mathbf{r}, t))$ is the diffusion coefficient for image u at location $\mathbf{r}$. The dot product of the diffusion tensor D with the outer normal n defines the Neumann boundary condition which holds for insulated boundaries where there could not be flux beyond the boundaries so that the density gradient must vanish. An analytical solution of the diffusion equation can be achieved via the method of separation of variables. Though, to solve the problem numerically it is required to discretise it.

Conventionally the diffusivity factor is observed to depend on the image gradient (e.g Perona-Malik case [108]). However, Weickert introduces the use of structure tensor as diffusivity to enable dissimilar smoothing in different directions [144]. The advantage of structure tensor over the simple ∇u_σ becomes obvious by observing the pictures in Figure 3.8. This example is similar to the one in [144] and it compares the average orientation of eigenvectors with the average direction of gradients for the fingerprint image. In this example the gradients and the elements of the structure tensors are averaged with a Gaussian kernel of $\sigma = 5$ (Figure 3.8 (c) and (e)). It becomes obvious that in a large window the average direction of gradient does not represent the structure of data. Because the averaging of gradients in a window might cancel out the gradients pointing at opposite directions, it is more convenient to average the tensor matrix field. Therefore the

Figure 3.9: Anisotropic diffusion of a fingerprint image. The values of the required parameters are $\sigma = 1$, $\rho = 6$ and $t = 30$.

convolution of the structure tensor J_0 with a Gaussian kernel K_ρ gives

$$J_\rho(\nabla u_\sigma) = K_\rho * (\nabla u_\sigma \nabla u_\sigma^\top) \tag{3.32}$$

This means that the eigenvectors of the resulting tensor summarize the direction of gradients within the window of size ρ. The eigenvector with the highest eigenvalue is parallel to the average direction of gradient, while the one with the lower eigenvalue moves along coherent structures and its eigenvector is perpendicular to average direction of gradient. If someone wants to smooth along the flow of structures then they should adapt the eigenvalue of the vector that has the same direction with the flow in such a way that it increases with respect to coherence $(\mu_1 - \mu_2)^2$.

Furthermore, the author did not only replace the gradient based diffusivity with a structure tensor, but he also adapted the diffusion tensor in such a way that it facilitates the diffusion along the flow-like patterns. Following the property of a positive semi-definite matrix $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ the authors simplified and substituted the eigenvalues of the tensor maintaining at the same time the same eigenvectors. Consequently, the diffusion tensor becomes $D = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where Λ is a diagonal matrix of new eigenvalues. An example of application of anisotropic diffusion is presented in Figure 3.9.

3.9 Summary and Discussion

This chapter presented the background and the theoretical preliminaries that are required to understand the concepts presented within rest of the thesis. The COD system presented here will be used as a benchmark algorithm in chapters 5 and 6 and the background modelling algorithm described will be used in chapter 4 to model the background by using improved feature vectors instead of using simple colour values. Since a novel edge enhancing filter is proposed in the chapter

4, it was necessary to review the theory behind oriented filters (see section 3.4). Sections 3.5 and 3.6 extensively discussed the theory behind diffusion starting from the diffusion of matter and heat conduction in materials, and concluding with a presentation on coherence enhancing anisotropic diffusion in images. Section 3.7 that presented the structure tensor is of high importance as the properties of tensors are exploited in chapter 4 for contour completion via anisotropic diffusion.

Chapter 4

Extraction of Closed Foreground Contours

The most widely recognised method for background modelling is via MoG [130] i.e. each pixel in the video frames is modelled by a weighted MoG whose parameters and weights are continuously updated throughout time. This allows the prediction of the pixel values that belong to the background in consecutive frames based on probability. However, this method has some weaknesses including the detection of shadows and spurious objects, inability to handle sudden illumination changes and foreground-background similarity. These shortcomings have led to a vast amount of literature attempting to improve the original concept.

To solve the above issues entirely or at least to improve the robustness against these weaknesses, this chapter approaches the background modelling from another perspective. Instead of using the colour pixel values as features the proposed method utilises the unique properties of gradient and phase congruency features. Since the gradient measures the difference between neighbouring pixels across a line segment, it is not so sensitive to colour variations and illumination changes. The same applies to phase congruency features that represent the agreement of phases of frequency components of a decomposed image signal. Before the computation of image gradient the image is pre-smoothed for noise reduction with a carefully chosen filter that would not degenerate the edges.

The collective result of the above features is a crude foreground contour that should undergo refinement to conclude with a fine object contour. To that end the crude foreground contour is reflected onto the edge contours obtained with the standard edge detection procedure proposed by John F. Canny and at different scales to ensure scale invariance.

Since the aim was to segment as many contours as possible it is inevitable to encounter edges that result due to noise. This situation requires a noise edge re-

duction techniques that would take into account the colour and texture properties of the surrounding area. For this reason colour and texture similarity measures are developed based on various colour spaces to train a classifier that will classify the edges as foreground and background. As the extracted contour does not always result in a closed curve, edge continuation and closure techniques are employed to ensure closed contours [51, 48].

4.1 The Starting Point of the Research

The inspiration for this chapter has been the work of O. Javed et al. in [63] who used the magnitude and direction of image gradient to address some of the weaknesses of the algorithm originally proposed by Stauffer and Grimson [130]. The work in [63] proposes a background modelling with a combination of colour features along with gradient features to ensure that the biggest part of the contour of the detected foreground consists of a strong magnitude of the gradient.

Following the algorithm proposed in [130], the authors in [63] model the background in RGB colour space, as a MoG which is updated over time using K-means approximation of the EM algorithm. To compute the gradient features, the colour mean of each pixel as well as the standard deviation are converted to their equivalent greyscale. The mean magnitude of gradient in x and y directions is calculated as the difference of means between neighbouring pixels, and the standard deviation as a sum of standard deviations. When an incoming frame arrives, the magnitude and direction of gradient are computed and compared against the ones of the model to decide which pixels belong to the foreground. It should be noted that no separate model is maintained for the gradient vector information but in contrast it is derived from the updated colour mixture model.

Next, the following procedure is applied in [63] to determine if the foreground found by RGB features was valid. Assume that the RGB coloured frame is I and the extracted foreground be $F(I)$. If the gradient image is I_g let the foreground edge information be $G(I_g)$. Further assume that the boundaries of $F(I)$ will be denoted as $\partial F(I)$. For a foreground region that belongs to $\partial F(I)$ there should be a corresponding foreground region in $G(I_g)$ that matches a minimum percentage (20 per cent) of the pixels of that region, as explained in [63]. If the percentage is relatively low then the region in $\partial F(I)$ is considered as a spurious object and is not included in the foreground. Except for the above criterion the paper in [63] suggests that the true foreground boundaries should also lie on some edge of image I.

At last, when sudden illumination changes are observed the foreground segmentation is only based on gradient features. The advantage of the method des-

cribed above is that it ignores the spurious objects such as isolated light blobs and undefined shadows having undefined contours and is robust to illumination changes.

However, the ideas presented in [63] were not exploited to their full extent. They only specify if a region (for example defined as a connected component) as a whole belongs to the foreground or not but do not evaluate parts of the same region. Therefore if for instance 10 per cent of an object consists of its contour-less shadow then the whole object will be considered as a foreground object. To address this issue, the foreground $G(I_g)$ could be used to dispose of these shadow regions or at least part of them. Furthermore, sometimes the colour based extracted foreground object lacks exacted contours that could be corrected with the use of edges that lie on $G(I_g)$.

The simplest approach to achieving the aforementioned goals would involve the following steps: Computation of $F(I)$, $G(I_g)$ and the edges of the incoming frame $E(I)$; morphological dilation and closing of $G(I_g)$ to obtain $M(G(I_g))$ and definition of the convex hulls that enclose the connected components of $M(G(I_g))$; removal of all the regions in $F(I)$ that lay out of the convex hull thus, achieving the elimination of shadows that do not have strong gradients. Conclusively, the smooth contour $C(I)$ of the foreground region is the edge $E(I)$ present in the combination of $F(I)$ with the $M(G(I_g))$ and can be defined by the following expression:

$$C(I) = (F(I) + M(G(I_g)))\, E(I) \tag{4.1}$$

The regions of $F(I)$ that are enclosed in the contour $C(I)$ are considered as foreground. Advantage of the above method is the detection of well-defined foreground contours, as sometimes the algorithms based on common MoG background subtraction fail to deliver an accurately defined contour. Examples of incomplete contours produced by algorithms described in [130, 68] and [151] are shown in Figure 4.1. On, the other hand the convex hull does not always enclose the whole object resulting in broken regions.

However, this process gives rise to the following questions: Are the gradient features enough on their own to detect the foreground contour? Could the gradient be computed at directions beyond the x and y (e.g. at angles $\{0°, 30°, 60° \ldots 330°\}$)? Does the convex hull not absorb the background edges along with the foreground edges that it encloses? Do the foreground edges always result in a closed contour? How tolerant is the gradient to noise? The sections that follow address all these questions through the design of a complete system that integrates all advantages of the method.

(a) (b) (c) (d) (e)

Figure 4.1: Examples of incomplete silhouettes segmented with three different algorithms based on MoG: (a) input frame, (b) ground truth, (c) GMM of Stauffer & Grimson [130], (d) GMM of KaewTraKulPong [68], (e) Multi-Layer Background Subtraction [151].

4.2 The Outline of the System

The general idea is the detection of foreground contour based on statistical background modelling with gradient features initially ignoring the colour properties of the pixels. Once the contours are obtained they are reflected on the incoming frame edges. As there is a high probability of noise occurrence, the noise edges are removed based on colour ratios. The removal of noise edges might lead to removal of healthy edges resulting in broken contours. This is resolved by an iterative contour closure method based on diffusion. The summary of the system is represented as a flow chart in Figure 4.2.

To model the background the magnitude of gradient is computed in 12 different directions and subsequently the resulting 12 images are combined into 6. To enhance the detection of contours, supplementary features are computed. These are the phase congruency features in 6 different directions as proposed in [75, 74]. The final combination of all these features constitutes the feature vector used to model the background. The same feature vector is computed for each incoming frame and compared against the background. The detected foreground is the contour of the likely foreground objects (sections 4.3 and 4.4).

At the same time the edge information is computed for the current frame on H and S components of the HSV colour space with J. F. Cannys edge detection algorithm [19]. To ensure scale invariance and given the fact that the saturation component is not a smooth image the edges are calculated at different scales with different thresholds. Another additional edge is derived as maximum moment of phase congruency covariance proposed by P. Kovesi in [76] (subsection 4.5.1).

The reflection of the detected crude contours on the combination of edges provides with refined foreground contour information. As the occurrence of noise is inevitable, the detected foreground edge segments must undergo further post-processing to remove the edges that result from noise. Among noise edge segments, there are also those that belong to shadows and should be classified as such. To achieve noise and shadow line removal, colour ratios are calculated in three

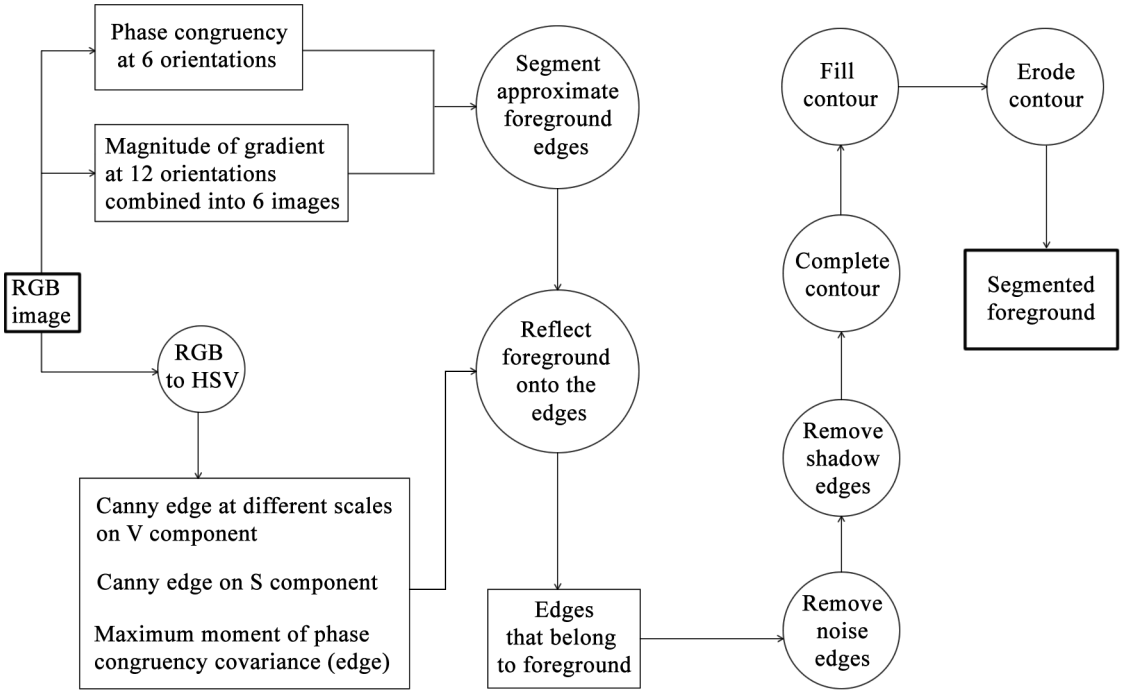


Figure 4.2: Flow chart of the foreground contour recovery system.

different colour spaces along the edge segment. This technique was successfully employed by J.-F. Lalonde [80], who utilised the colour ratios along with texture features to remove the shadows in outdoor consumer photographs. Here, the colour ratios are compacted into four representative features that are capable of performing the intended task (subsections 4.5.3 and 4.5.2).

At the final stage the contours are completed via iterative anisotropic diffusion proposed by D. Gil and P. Radeva in [48] and flood filled to recover the foreground regions (subsection 4.5.5).

4.3 Computation of Gradient and Phase Congruency Features

The directional derivative of an image is very sensitive to noise. Therefore the image should undergo smoothing in the first place. Assuming an image I and a smoothing filter h, according to the differentiation rules it is valid that [72],

$$\frac{\partial I}{\partial x_i} * h = \frac{\partial}{\partial x_i}(I * h) = \frac{\partial h}{\partial x_i} * I \tag{4.2}$$

Hence hitherto all the operations of derivation will be applied on the selected smoothing filter at a first stage and then the filter will be convolved with the image.

As it was shown in the previous section the background modelling based on the magnitude of gradient is a significantly promising method since it is stable to sudden illumination changes. The gradient information can be derived by convolving the image with the first partial derivative of a two dimensional Gaussian function. The partial derivatives are usually taken with respect to directions x and y which generate the gradient at only two directions. To improve the accuracy it was decided to compute the derivatives at multiple directions. The concept of computing the derivative of a filter at multiple directions is the same as steering any of the partial derivatives. This brings up the question of how exactly is to steer the first derivative of Gaussian filter or any other filter in general and weather a filter is steerable or not. The introductory material about steerable filters is presented in section 3.4.

For the purpose of current application, the first order derivative of Gaussian function is considered, for the reason that their convolution with an image results in gradient information. Assuming that the 2-D Gaussian function to be used is $G(x, y) = e^{-(x^2+y^2)}$ then the basis set for the first order derivative of Gaussian consists of two directional derivatives $G^{0^\circ} = \frac{\partial G}{\partial x} = -2xe^{-(x^2+y^2)}$ and $G^{90^\circ} = \frac{\partial G}{\partial y} = -2ye^{-(x^2+y^2)}$ and the steering condition is expressed as:

$$G^{\theta} = \cos(\theta)G^{0^\circ} + \sin(\theta)G^{90^\circ} \tag{4.3}$$

It is well known that the Gaussian, as a smoothing filter, does not favour the edges while reducing the noise. Therefore there is need for a smoothing function that would respect the edge structure while blurring the random noise. In the following section such a function is studied and the experiments conducted have shown that it performs better than Gaussian derivative of first order.

4.3.1 The Edge Enhancing Filter (EEF)

It is well known that the infinite sinc function $sinc(x) = \sin(x)/x$ acts as an ideal interpolation filter in spacial domain. Since the data samples available are finite this leads to the truncation of the sinc filter and consequently to approximate solutions. This results in the "ringing" effect, also known as Gibbs phenomenon while filtering with truncated *sinc*. In order to mitigate the ripple like effect, several windowing functions such as Hamming, Blackman, Kaiser, and Lanczos, that in contrast to a box filter attenuates softly at their ends, have been developed. The convolution of sinc function with any of these windowing functions results in a smoothly weakening sinc function that approximates zero as fast as the windowing function [146].

The windowed sinc function is widely used as an interpolation kernel. However

Table 4.1: Values of the sinc and the proposed synthetic EEF filter and their derivatives.

$x = [-2\pi, 2\pi]$	$sinc(x)$	$Im(H(sinc(x)))$	$f(x)$	$Im(H(f(x)))$
0.0000	1.0000	0.0000	1.0000	0.0000
1.0472	0.8270	0.4632	0.8648	0.4349
2.0944	0.4135	0.6884	0.4500	0.7020
3.1416	0.0000	0.5940	0.0000	0.5521
4.1888	-0.2067	0.3047	-0.1025	0.2511
5.2360	-0.1654	0.0280	-0.0800	0.0867
6.2832	0.0000	-0.0445	0.0000	-0.0106

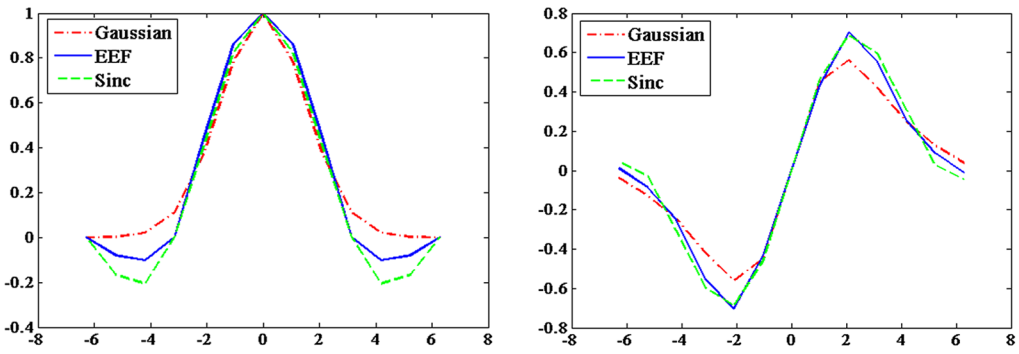


Figure 4.3: Similarity of the shapes of the proposed EEF filter, the Gaussian filter ($\sigma = 1.5$) and the sinc filter. The first image shows the initial waves and the second image shows their derivatives.

it is rarely considered for image filtering. An example of an interpolation filter evolved from the sinc function is the edge resolution enhancing interpolation filter [98]. Inspired by the work in [98] it is attempted to create a similar filter for edge enhancement. Initially the sinc function is obtained in the interval $[-2\pi, 2\pi]$ and next modified in such a way that the beneficial effects of the function are preserved, at the same time minimising the negative impact. Since the synthetic Edge Enhancing Filter (EEF) f does not have a specific function, it will be impossible to find its basis functions for steering it in an analytical manner. For the same reason the functional derivative of the filter is not known. A common way to approximate the derivative numerically is to use the Hilbert transform. The result of using the Hilbert transform is an array of complex valued numbers of the form $z = a + bi$, where $a = Re(z)$ and $b = Im(z)$. The real part $Re(z)$ is the initial function itself while the imaginary part $Im(z)$ is the derivative of the function. As a sample, 13 equally spaced points in the interval $[-2\pi, 2\pi]$ were taken. Table 4.1 shows the values of the sample points in the specified interval, as well as the results of the

(a) (b) (c) (d)

Figure 4.4: The results of filtering an image (a) with the Gaussian filter ($\sigma = 1.5$) (b), EEF (c) and a $sinc(x)$ filter (d).

Figure 4.5: From left to right in first and second row: Fingerprint image convolved with the derivative of Gaussian filter, EEF and a $sinc(x)$ filter and undergone non-maximum suppression with threshold 0.5.

functions applied to the points.

Figure 4.3 represents the EEF, Gaussian, and $sinc(x)$ functions in the interval $[-2\pi, 2\pi]$. As it appears the shape of EEF is something between the Gaussian ($\sigma = 1.5$) and the *sinc* filter. This is more clearly observed in the images of Figure 4.4 that show the original noisy image in (a) blurred by the Gaussian filter in (b), by the EEF in (c) and the *sinc* filter in (d). Another example that best illustrates the properties of the EEF is presented in Figure 4.5 where the fingerprint image has been convolved with the horizontal and vertical derivatives of the three filters under question and has then undergone non-maximum suppression with threshold 0.5 (this threshold has been selected as the best to display the results). By comparing the results it can be inferred that the EEF enhances the edge continuity along the curves in contrast to the Gaussian filter, while the sinc filter

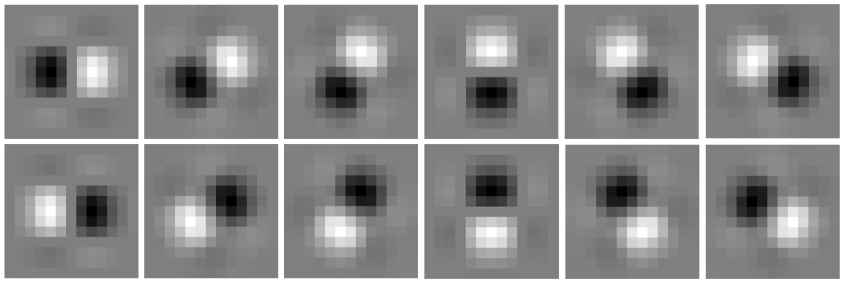


Figure 4.6: The proposed EEF filter rotated in 12 directions.

starts creating edgy effects at the boundaries of the image. More images comparing the three filters are available in Appendix A.

4.3.2 Formation of Two Dimensional EEF

It is well established that the two dimensional Gaussian and sinc filters are x and y separable, which means that the two dimensional filters can be obtained as the outer product of two one dimensional functions. If $g(x, y)$ is a two dimensional Gaussian filter and $g(x)$ is its one dimensional horizontal version and $g(y)$ is the vertical one then it is valid that

$$g(x, y) = g(x) \oplus g(x) = g(x)^{\top} g(x) = g(y)g(x) \tag{4.4}$$

where $\oplus$ is the symbol for outer product.

Similarly the EEF in two dimensions is the outer product of two one dimensional filters $f(x)$. In terms of two dimensional partial derivatives it is known that for a Gaussian function $g(x)$ and its corresponding two dimensional $g(x, y)$ it is valid that $\frac{\partial}{\partial x} g(x, y) = g(y)g'(x)$ and $\frac{\partial^2}{\partial x^2} g(x, y) = g(y)g''(x)$ and so on. In the same way the two dimensional partial derivatives of the EEF are $\frac{\partial}{\partial x} f(x.y) = f(y)f'(x)$ and $\frac{\partial}{\partial y} f(x, y) = f'(y)f(x)$.

As it was explained earlier, to obtain more accurate results the image should be convolved with a bank of rotated EEF (see Figure 4.6) at orientations $\theta = \{0°, 30°, 60°, \ldots 330°\}$. The specified rotations were selected after experiments, as they gave the best results. The rotation operator performs a geometric transform, which maps the position (x_1, y_1) of a picture element in the EEF $f^{\theta}(x, y)$ onto a position (x_2, y_2) in the EEF $f^{0°}(x, y)$ by rotating it through a specified angle θ about an origin O at (x_0, y_0), which is normally the centre of the image.

$$\begin{aligned} x_2 &= \cos(\theta)(x_1 - x_0) - \sin(\theta)(y_1 - y_0) + x_0 \\ y_2 &= \sin(\theta)(x_1 - x_0) + \cos(\theta)(y_1 - y_0) + y_0 \end{aligned} \tag{4.5}$$

The rotation algorithm can produce coordinates (x_2, y_2) which are not integers. In order to determine the value of the filter at these positions the Gauss' interpolation method is used. The value of the rotated filter at the position (x_1, y_1) is $f^{\theta}(x_1, y_1) = f(x_2)f'(y_2)$, where $f(x)$ is the one dimensional EEF.

Gauss' data interpolation method [18] is described as follows: Since the filter is synthesised in the spatial domain, we are interested to know the values of the function at n positions. Assuming that we have the real data pairs (x_n, y_n) given for $n = \{0, 1, \ldots, N-1\}$, where $x_n = 2\pi n/N$. The interpolating condition for $n = \{0, 1, \ldots, N-1\}$ is

$$y_n = a_0 + \sum_{k=1}^{\frac{N}{2}-1} \left[a_k \cos\left(\frac{2\pi nk}{N}\right) + b_k \sin\left(\frac{2\pi nk}{N}\right) \right] a_{N/2} cos(\pi N) \tag{4.6}$$

where $a_0 = c_0$, $a_{N/2} = c_{N/2}$, $a_k = 2Re\{c_k\}$, $b_k = -2Im\{c_k\}$ and where the coefficients c_k are calculated by using fast Fourier transform as

$$c_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \tag{4.7}$$

where, $k = \{1, \ldots, \frac{N}{2} - 1\}$

Once the two dimensional filter has been rotated it is imperative to check if the filter can be decomposed into several separable basis filters. Since there is no specific function for the EEF it is impossible to identify the basis filters in an analytical manner. Therefore, numerical methods such as those proposed in [**?**, 91, 92] will be suitable. Application of singular value decomposition (as proposed by P. Perona in [109]) to the bank of rotated filters reveals that the minimum number of non separable basis functions that can rotate the filter with minimum error is 6. If the 6 basis functions are separable then to perform convolution with each basis will cost $2 \times 13 = 26$ operations; $26 \times 6 = 156$ operations in total. Since the convolution with one two dimensional filter requires $13 \times 13 = 169$ operations, it was decided to not utilise any of the aforementioned numerical approaches to find the separable basis functions but use the pre-computed rotated filter bank instead. The MATLAB implementation of the method based on SVD is available at http://www.vision.caltech.edu/manduchi/def.tar.Z and is cited in the paper [92] authored by R. Manduchi et al.

4.3.3 Formation of the Final Feature Vector

The result of convolving a greyscale image with the bank of precomputed kernels will give gradient information at multiple orientations. In particular the

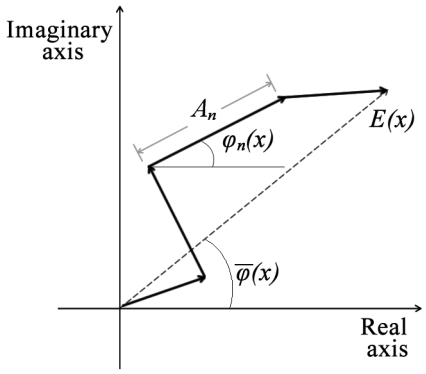


Figure 4.7: The total energy $E(x)$ is computed as sum of amplitudes A_n.

magnitude of gradient is of high interest and it is derived in the following way. Supposed $I^\theta = I * f^\theta$ is the convolution of the image with a filter and also $\theta_j = \{0°, 30°, 60°, ..., 330°\}$ with $j = 1, ..., N$. Then the magnitudes that are derived are expressed as

$$F_j = \sqrt{(I^{\theta_j})^2 + (I^{\theta_{(N-j+1)}})^2}, \quad j = 1, ..., N/2 \tag{4.8}$$

Thus, F_j is a collection of image features for background modelling. Since the goal is to maximise the detection of approximate contours it was decided to utilise image features from phase congruency. These were first proposed by P. Kovesi in [74] and [76] and it is described as a measure invariant to brightness or contrast changes and signifies the points of high interest. The author mentions that feature realisation in an image occurs at points of maximum phase congruency which, while speaking about a one dimensional signal, means that when the phases of all detected frequencies are in agreement then the point is of high interest. This is best illustrated in Figure 4.7, which shows the amplitudes A_n of all different frequencies of different phases added to compute the total length of the path from the centre of axes to the end of vector $E(x)$. At the same time the vector connecting the start point and the end point of the path is expressed as energy $E(x)$. It is obvious that if all the frequencies are in phase then the sum of amplitudes has the maximum value and it is when the local energy $E(x)$, attains the maximum. The above concept can be expressed by the following equation.

$$PC(x) = \frac{|E(x)|}{\sum_n A_n(x)} = \frac{\sum_n A_n \ cos(\phi_n(x) - \bar{\phi}(x))}{\sum_n A_n(x)} \tag{4.9}$$

P. Kovesi further enhanced the Equation 4.9 to make it insensitive to noise; for more information refer to [75]. In the practical case of two dimensional images the computation of phase congruency is not as straightforward as defined by the

above equation. To identify the amplitudes, the image is convolved with oriented Gabor wavelets of different scales. The Equation 4.9 is calculated separately at 6 orientations and the resulting phase congruencies $PC(\theta)$ are used as features for the background model.

4.4 Background Updating and Foreground Segmentation

After combining the 6 image features of gradient information with the 6 phase congruency images, a 12 dimensional feature vector is obtained. This feature vector is used to perform background modeling based on a mixture of Gaussian distributions as proposed by C. Stauffer and W. E. L. Grimson [129, 130]. If the size of the frames of the under processing video is $m \times n$ then we can assume a feature cube of size $m \times n \times 12$. Each pixel of the frame could be represented by the variable $F(x, y, i)$, where $i = \{1, ..., 12\}$ is the number of features. The history of feature values for each pixel is modelled by K Gaussian distributions. In the current application the minimum number of $K = 2$ gives the best results. The probability of observing $F(x, y)$ at time t is [129]

$$P\left(F_{(x,y),t}\right) = \sum_{k=1}^{K} \omega_{k,t} * \eta\left(F_{(x,y),t}, \mu_{k,t}, \Sigma_{k,t}\right) \tag{4.10}$$

Where $\omega_{k,t}$ are the coefficients of each distribution that act as weights, $\mu_{k,t}$, and $\Sigma_{k,t}$ are the mean value and the covariance matrix of the k^{th} Gaussian distribution. The Gaussian function is defined as [129]

$$\eta\left(F_{(x,y),t}, \mu_{k,t}, \Sigma_{k,t}\right) = \frac{1}{(2\pi)^{n/2}|\Sigma_{k,t}|^{1/2}} e^{-\frac{1}{2}\left(F_{(x,y),t}-\mu_{k,t}\right)^{\top}\Sigma_{k,t}^{-1}\left(F_{(x,y),t}-\mu_{k,t}\right)} \tag{4.11}$$

Stauffer and Grimson assumed that the R, G, and B values are independent and have the same variances and thus

$$\Sigma_{k,t} = \left(\sigma_k^2 \mathbf{I}\right) \tag{4.12}$$

where $\mathbf{I}$ is the identity matrix and σ_k^2 is the variance of the k^{th} distribution. The same assumption applies for the 12 features. Every new vector $F_{(x,y,i),t}$ is checked against the existing K distributions via absolute distance to find a match. This distance is defined as:

$$D_k = \left|F_{(x,y),t} - \mu_{k,t-1}\right| \tag{4.13}$$

A match is found if the dot product of the distance vector of each distribution is

smaller than 2.5 standard deviations, i.e.:

$$D_k \cdot D_k \leq 2.5\sigma_{k,t-1} \tag{4.14}$$

If a match is found then the k^{th} Gaussian that is found to be a match, is updated following the next equations.

$$\begin{aligned} \mu_{k,t} &= (1-a)\mu_{k,t-1} + aF_{(x,y),t} \\ \sigma^2_{k,t} &= \max\left((1-a)\sigma^2_{k,t-1} + a\left(F_{(x,y),t} - \mu_{k,t}\right)^\top \left(F_{(x,y),t} - \mu_{k,t}\right), \sigma^2_{\min}\right) \\ \omega_{k,t} &= (1-a)\omega_{k,t-1} + a(r_{k,t}) \end{aligned} \tag{4.15}$$

where a is the learning rate and $r_{k,t}$ is 1 for the model which matched and 0 for the remaining models. In the case that no matching distribution is found then the pixel is labeled as foreground and the least probable distribution is substituted by a new one, initialised with prior parameters for $\omega_{k,t}$ and $\sigma^2_{k,t}$ and $\mu_{k,t} = F_{(x,y),t}$. To decide which portion of distributions represents the foreground and which portion defines the background, it is important to put them in an order according to their weight $\omega_{k,t}$. Then the first B distributions, chosen according to the following formula, account for the background.

$$B = arg\min_b \left(\sum_{k=1}^{b} \omega_{k,t} > T\right) \tag{4.16}$$

where T is the threshold that defines the portion of distributions that are accounted for by the background. A pixel is labeled as background if the current $F_{(x,y),t}$ matches any of the first B distributions. In the opposite case the pixel is labeled as foreground. For the current implementation the parameters are $T = 0.4$, $a = 0.005$, $\sigma = 0.3$, $\sigma_{\min} = 0.7\sigma$ and $K = 2$.

4.5 Post Processing of the Segmented Foreground Contour

The result of foreground segmentation using gradient and phase congruency features is the derivation of approximate foreground contours that should undergo a series of post processing steps to achieve an accurate foreground contour definition. These steps include the reflection of the raw foreground contours on edges derived from the incoming frame and further refinement by removing the line segments that occur as a result of noise or shadow. As some edge segments are removed it is imperative to make use of contour completion methods to ensure closed contours. A sophisticated algorithm, developed by D. Gil and P. Radeva in [48] based on

anisotropic diffusion is employed to recover smooth edges. Due to the complexity of the method, its explanation will consume a significant part of this section. In spite of the good results produced the methods intensity makes it questionable for real time applications. An outline of the post processing algorithm is presented in pseudocode in Appendix B.

The key algorithms to be explained in this section are:

- Edge detection
- Noise line removal
- Shadow line removal
- Simple edge completion method
- Anisotropic diffusion to recover smooth edges

4.5.1 Edge Detection and Accumulation

A credible raw foreground contour refinement should include the reflection of the contour on credible edges. The edge detection algorithm should be scale invariant or at least scale conscious. Hence, Canny edge detection routine has been used at different scales and the results have been accumulated into one aggregate image.

Usually the Canny algorithm is applied on a greyscale image. Nonetheless, it was decided to collect edges from the HSV colour space since the saturation component encloses valuable segmentation information, when the value component fails. The scales chosen for gradient estimation are $\sigma = 2$ for the saturation component and $\sigma = \{1, 2, 3\}$ for the value component. After non-maximal suppression the corresponding high and low thresholds for hysteresis thresholding are respectively, $thresh_{high} = \{0.3, 0.08, 0.05, 0.3\}$ and $thresh_{low} = 0.4thresh_{high}$.

Except for gradient based edge detection methods an important edge detection method based on phase congruency, proposed by P. Kovesi in [76], was employed. The main strength of phase congruency is that it provides an absolute measure of significance of features. This allows the definition of one global threshold for the entire image or a range of images. P. Kovesi identifies a measure of edges from phase congruency based on moment analysis. Hence, non-maximal suppression of maximum moment of phase congruency is considered as a complementary edge. The complementarity of this method lies in the fact that in contrast to Canny filter that produces response on each side of a line feature, the phase congruency produces one centralised response. Here, the thresholds for hysteresis thresholding are $thresh_{high} = 0.3$ and $thres_{low} = 0.01$.

The total of 5 edge images undergo noise and shadow line reduction separately, and the resulting edges are combined into one final edge image ready for contour completion

4.5.2 Shadow Line Removal

Ignoring the order the algorithms are executed, it is good to start with the shadow edge removal. This part if applied is the actual bottleneck of the whole algorithm. The existing literature on edge based shadow removal suggests a variety of methods. However, the more they increase in accuracy, the more computationally expensive they become. The currently available methods implemented in MATLAB are significantly far from being real time. The methodology proposed by J. F. Lalonde et al. [80] for shadow removal in consumer photographs in outdoor scenes is the one that best suits the proposed foreground segmentation process. This is because the authors approach the shadow detection problem not at region level as usually happens, but at edge level. They perform watershed segmentation to detect object boundaries and gradient estimation to localise strong candidate shadow edges. Since in our case the edges are already known the first stage of the algorithm could be omitted.

The basic concept of the algorithm is the detection of features across the boundaries and the use of Adaboost classifier to categorise the edges. The features computed across the edges include colour ratios at 3 colour spaces (RGB, LAB, ILL [26]), texton features [94], and skewness of pixel intensities [160]. To compute the colour ratios at each pixel location across the line it is important to know the orientation of the line at that point. Once the orientation is determined, oriented first derivative of the Gaussian filter is constructed at 4 different scales and is matched with the orientation of the line. Subsequently the weighted average $fl(p)$ of pixels under the positive side of the filter can be calculated. If the filter is rotated by 180° then the weighted average $fr(p)$ of pixels at the other side of the line can be calculated. The ratios for each pixel and colour space are computed as fractions of the minimum average to the maximum, as follows:

$$\frac{\min\left(fl(p), fr(p)\right)}{\max\left(fl(p), fr(p)\right)} \tag{4.17}$$

Once all the ratios along the line are obtained, they are averaged in such a way that there is one ratio for each colour space component resulting in 9 colour ratios computed for each scale. The colour ratios along with the texture features are fed to Adaboost classifier which delivers the final classification results. More details about this technique can be found in [80]. One more contemporary approach for

shadow detection that partially uses the concepts proposed by Lalonde is the one of X. Jiang et al. in [148]. The authors suggest incorporating local correlations between the local luminance contrast and average local luminance to enhance the output of classification based only on colour ratios. However the procedure is very long and complicated to be adopted for the purpose of this application.

4.5.3 Noise Line Removal

As it was expected, the occurrence of noise lines due to inaccurate initial foreground segmentation is high, therefore they should be removed. Following the concept of colour ratios in different colour spaces it becomes possible to remove the noise lines since their average ratios are approximate to one. A high average ratio means that there is a considerable colour similarity between the areas on either side of the line. This feature is in particular useful in dynamic background scenarios with consistent texture such as sea waves or even weak shadow lines.

The preferred colour spaces are RGI (where I=R+G+B), LAB and HSV and the selected scales for Gaussian filter are $\sigma = \{2, 3, 6\}$. The first derivative of Gaussian is steered as described in the subsection 4.3.2. Similar to shadow line removal the weighted average of pixel values is calculated at either side of a line for each point (Figure 4.8). Therefore it will be $3 \times 9 = 27$ ratios per pixel in total. However the colour ratios are not used directly for classification, on the contrary they are compacted into 4 representative features that will be used with SVM classifier to perform the intended separation into 2 classes (foreground and background).

The first feature to be computed is S_1 defined as the average of the colour ratios as they were proposed by J. F. Lalonde.

$$RL_{i,j} = \frac{\min\left(fl(p_{i,j}), fr(p_{i,j})\right)}{\max\left(fl(p_{i,j}), fr(p_{i,j})\right)} \quad i = 1...N, \quad j = 1...M \tag{4.18}$$

where N is the length of the line in pixels and M is the number of colour components per pixel.

$$S_1 = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} RL_{i,j}}{MN} \tag{4.19}$$

It is common sense that the purpose of using ratios is to achieve a measure which will be invariant to the scale of colour spaces. Working towards the same direction, the ratios between consecutive partial averages of pixels at the same side of a line will further reduce the dependence to colour information. This can be expressed

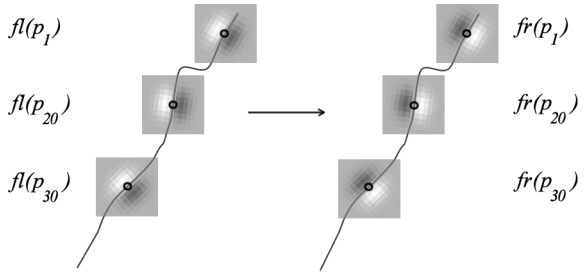


Figure 4.8: Colour ratio calculation with oriented Gaussian derivative filters along a line segment. The filter is rotated to obtain the average of pixel values at either side of the line.

by the equations

$$rl_{i,j} = \frac{fl(p_{i,j})}{fl(p_{i+1,j})}, \quad rr_{i,j} = \frac{fr(p_{i,j})}{fr(p_{i+1,j})}, \quad i = 1...N-1, \quad j = 1...M \tag{4.20}$$

Then the final ratios and the total score S_2 can be computed as

$$R_{i,j} = \frac{\min\left(rl(p_{i,j}), rr(p_{i,j})\right)}{\max\left(rl(p_{i,j}), rr(p_{i,j})\right)} \tag{4.21}$$

$$S_2 = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N-1} R_{i,j}}{M(N-1)} \tag{4.22}$$

For the third feature the ratios of partial averages at the same side of a line are computed with reference to one of the partial averages. This will give an indication of how much the pixel values along a line segment deviate from a fixed point of reference. Hence, the ratios are expressed by the follows equations as:

$$rl_{i,j} = \frac{fl(p_{1,j})}{fl(p_{i+1,j})}, \quad rr_{i,j} = \frac{fr(p_{1,j})}{fr(p_{i+1,j})} \tag{4.23}$$

The total score S_3 is calculated in the same way as S_2. The final feature to be included was widely used for shadow detection in the early foreground segmentation algorithms. The word is about the intensity component $I = R+G+B$ where the ratio between the intensity of background model and the intensity of incoming frame would indicate the presence of shadows [68, 140, 121, 40]. Thus the average of ratios for the intensity component is defined as the fourth feature:

$$S_4 = \frac{\sum_{i=1}^{N} RL_{i,3}}{N} \tag{4.24}$$

An example of the four features calculated for edge segments of the images in

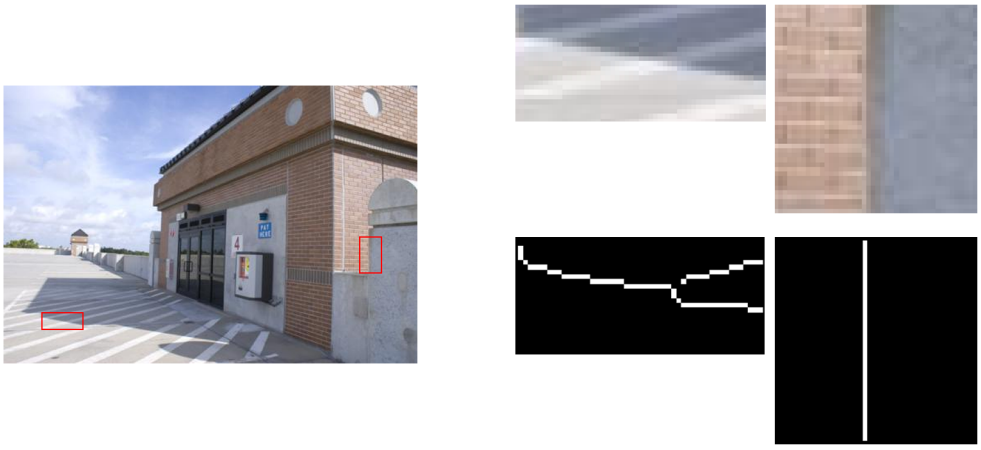


Figure 4.9: Example of two patches (right) of the same image (left) and their detected edge segments.

Figure 4.9 are: for the first case of the ground $S_1 = 0.5520$, $S_2 = 0.9652$, $S_3 = 0.6918$, $S_4 = 0.6275$, and for the second case of the wall $S_1 = 0.5627$, $S_2 = 0.8438$, $S_3 = 0.6696$, $S_4 = 0.7509$. It is obvious that feature S_2 is capable of identifying the areas where there is difference in texture and feature S_4 senses the shadow.

4.5.4 Edge Completion Methods

The edges that remain after shadow removal and noise line removal most of the time do not form a perfectly closed contour. The term "perfectly closed" means that the contour filling allows the recovery of the area enclosed in the contour. An insignificant amount of research has been conducted on edge completion techniques. Some of them, e.g., [90] and [97], concentrate on salient and closed contour extraction but not on contour completion. A considerable attempt is of M. Narayanan and B. Kimia in [100] who identify gaps for completion using constrained Delaunay triangulation and suggest completing them with Euler Spiral [70] or a straight line. Their simpler assumption in [100] that it is more likely that the edge end-points interact with the closest neighbouring edge end-points rather than with those that are far, leads to the selection of neighbouring end-points within a specified window.

The approach of completing contours with affine geodesics proposed by A. A. Handzel [51] is preferred to Eulers Spiral [70] since according to theoretical aspects it will provide with similar results via a simpler implementation. A geodesic line is the shortest line to connect two points on a curved surface; therefore, the authors obtain the geodesic as a parabolic arch that passes through two lines that are tangent to it. The contours to be completed could be assumed as two tangents. The pictures in Figure 4.10 compare contour completion with affine geodesics to

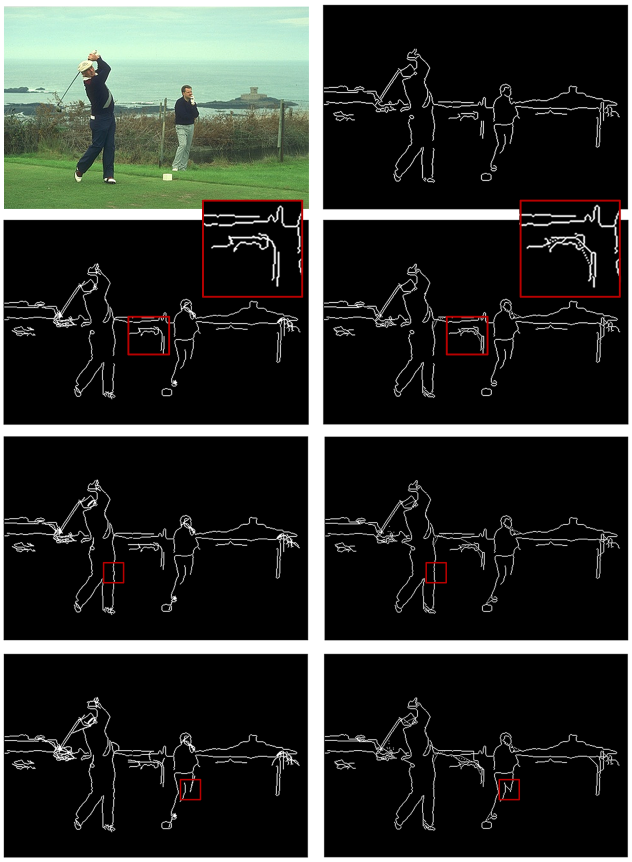


Figure 4.10: Contour completion with affine geodesics and straight lines in a neighbourhood of 11x11(second row), 15x15(third row) and 17x17(fourth row) around end-points. The parallel line segments as in the last example are not completed in the case of affine geodesics, as expected.

contour completion with straight lines in a neighbourhood of 11×11, 15×15 and 17×17 around end-points.

In order to spot edge end-point Kovesi Linker [73] was used. The edge image was obtained by applying the Canny edge detector with $threshold = 0.35$ and $sigma = 1$ on the golfer image from the Berkley segmentation dataset [93]. By zooming into detail it can be spotted that the affine geodesics complete the contour with a smooth curve in contrast to the simple straight line.

The main difficulty when using affine geodesics is the definition of the two tangent lines. Since in nature we can rarely see completely straight lines, the approximate fitting of a straight line to a contour segment is not always successful. This results in some undesirable parabolic curve definitions as it can be seen in the examples provided. Further in practical cases the above methods are not convenient for the main reason that they cannot guarantee "perfectly closed"contours.

In general a concept that has no clue on what direction a contour should follow to be completed is condemned to fail. The methods described above are good for imprinting when no information is available about the contour direction (or inten-

tion of continuation). In our case the missing contours occur because of incomplete foreground segmentation and not because of non-availability of direction relevant information. This means that following the contour towards the direction that is indicated by the gradient will lead to a smooth contour recovery. A method that successfully implements this concept using anisotropic diffusion is suggested by D. Gil and P. Radeva [48] and will be explained in detail in the following section.

4.5.5 Contour Completion Via Anisotropic Diffusion

Before proceeding to the following section the reader is advised to familiarise themselves with the concept of anisotropic diffusion presented in sections 3.5-3.8.

Smooth Contour Recovery

P. Radeva and D. Gil in their work extended the anisotropic diffusion to recover smooth contours [48]. The basic idea of their method was the extension of the objects contour in the direction perpendicular to edge gradient. Since, as explained above, the image gradient is not a sufficient measure to extend flow like structures the authors rely on the orientation of structure tensors averaged within a window of size ρ.

$$J_\rho(\nabla u_\sigma) = K_\rho * (\nabla u_\sigma \nabla u_\sigma^\top) \tag{4.25}$$

Computing the structure tensors for each pixel of an image leads to an automatic generation of a vector field defined on the whole image. The broken edges of a curve are interpolated along the defined vector field iteratively, via anisotropic diffusion. Given that the γ_0 is the set of points to be extended over the computed vector field, the diffusion process is encoded by the following equations [48]:

$$\frac{\partial u(x,y,t)}{\partial t} = \text{div } (D \nabla u(x,y,t)) \tag{4.26}$$

$$u(x,y,0) = \begin{cases} 1 \;\; if \;\; r = (x,y) \in \gamma_0 \\ 0 \;\; otherwise \end{cases} \tag{4.27}$$

To form the diffusion tensor D, the property $D = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ of a positive semi-definite matrix was used; where $Q = \begin{bmatrix} v_{1a} & v_{2a} \\ v_{1b} & v_{2b} \end{bmatrix}$. It is desired to extend the contour along the direction indicated by the eigenvector v_2 perpendicular to the image gradient and parallel to the real extension of contour. Hence, $\Lambda = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. has eigenvalues $\mu_1 = 0$ and $\mu_2 = 1$ to ensure maximum diffusion along the flow like structure. The final result of the diffusion process is the extended contour. The

following operations aim at discretisation of the problem to reach an iterative solution.

$$\partial_t u = \frac{\partial u(x,y,t)}{\partial t} = \text{div } (D\nabla u(x,y,t)) = \nabla \cdot (D\nabla u(x,y,t)) \tag{4.28}$$

If $D = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$, $\partial_x = \frac{\partial}{\partial x}, \partial_y = \frac{\partial}{\partial y}$ then by substituting the gradient operators by partial derivatives, it will be,

$$\begin{aligned} \partial_t u &= \begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \cdot \left[\begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} \partial_x u \\ \partial_y u \end{pmatrix} \right] \\ &= \begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \cdot \begin{pmatrix} a\partial_x u + b\partial_y u \\ b\partial_x u + c\partial_y u \end{pmatrix} \\ &= \partial_x(a\partial_x u + b\partial_y u) + \partial_y(b\partial_x u + c\partial_y u) \\ &= \partial_x(a\partial_x u) + \partial_x(b\partial_y u) + \partial_y(b\partial_x u) + \partial_y(c\partial_y u) \end{aligned} \tag{4.29}$$

If the symmetrical central differences are used to calculate the derivatives then

$$\begin{aligned} \partial_x f &= \tfrac{1}{2}(f_{i+1,j} - f_{i-1,j}) \\ \partial_y f &= \tfrac{1}{2}(f_{i,j+1} - f_{i,j-1}) \end{aligned} \tag{4.30}$$

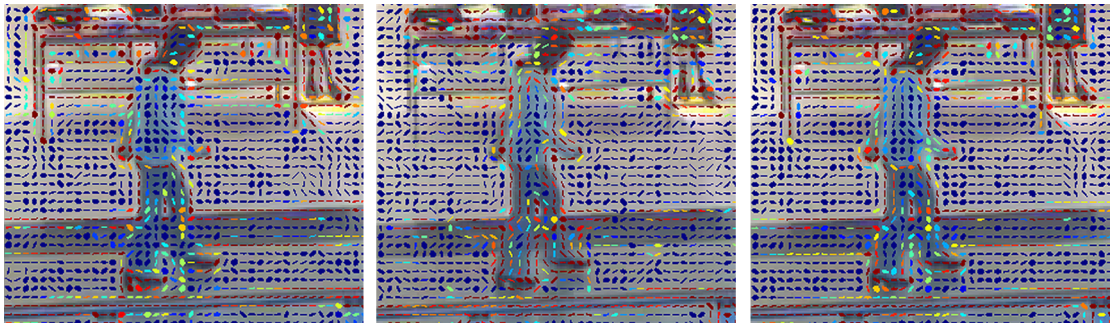
And thus the following equations express the discretised solution to the PDE,

$$\begin{aligned} \partial_x(a\partial_x u) &= \tfrac{1}{2}((a_{i,j} + a_{i+1,j})(u_{i+1,j} - u_{i,j}) - (a_{i-1,j} + a_{i,j})(u_{i,j} - u_{i-1,j})) \\ \partial_y(c\partial_y u) &= \tfrac{1}{2}((c_{i,j} + c_{i,j+1})(u_{i,j+1} - u_{i,j}) - (c_{i,j-1} + c_{i,j})(u_{i,j} - u_{i,j-1})) \\ \partial_x(b\partial_y u) &= \tfrac{1}{2}(\tfrac{1}{2}b_{i+1,j}(u_{i+1,j+1} - u_{i+1,j-1}) - \tfrac{1}{2}b_{i-1,j}(u_{i-1,j+1} - u_{i-1,j-1})) \\ \partial_y(b\partial_x u) &= \tfrac{1}{2}(\tfrac{1}{2}b_{i,j+1}(u_{i+1,j+1} - u_{i-1,j+1}) - \tfrac{1}{2}b_{i,j-1}(u_{i+1,j-1} - u_{i-1,j-1})) \end{aligned} \tag{4.31}$$

Tuning the Direction of Eigenvectors

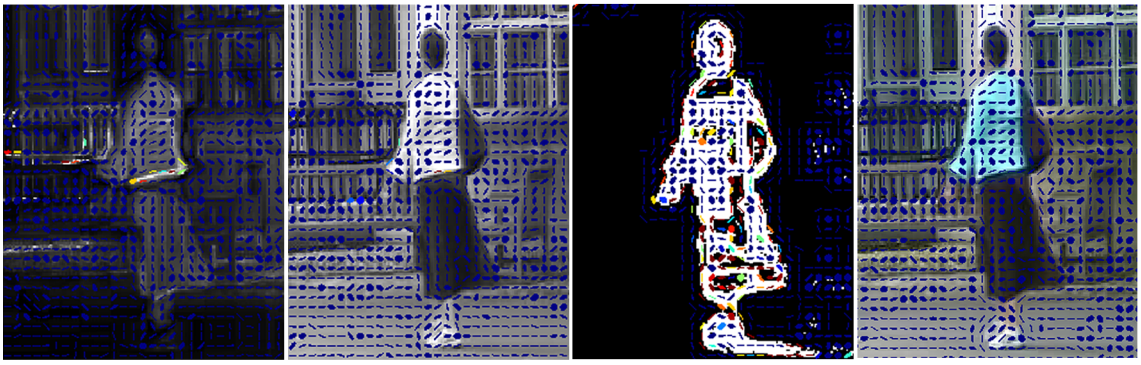
As it may have already been understood the selection of proper eigenvectors is crucial in contour completion process as they decide towards which direction the contour will be extended. Often the gradient information derived from the value component of an HSV image is not sufficient and it is required to borrow additional information from the saturation component.

Hence, the structure tensor can be first computed for the Value and Saturation components, and then averaged. The benefit of averaging the structure tensors is apparent in the images of Figure 4.11 . Attention should be drawn at the torso



(a) (b) (c)

Figure 4.11: Averaging of tensors of the value component (a) and the tensors of the saturation component (b) into one final tensor (c). The tensors are represented as ellipses with their orientation aligned with the orientation of the boundary.



(a) (b) (c) (d)

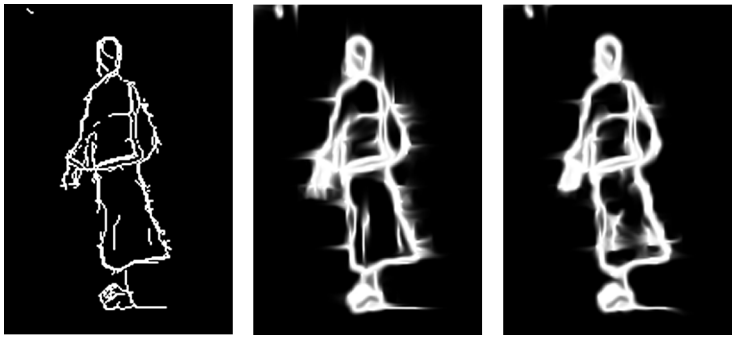
Figure 4.12: Tensor fields for the saturation (a), value (b) and foreground (c) images. The weighted average of the three images is shown in (d).

and the legs of the pedestrian, where at several points the value component fails in producing accurate structure tensors. This failure is compensated by tensors produced from the saturation component. The resulting average combines the advantages of both colour components.

Except for that, to inhibit the diffusion towards the areas that deviate from the moving object the structure tensor is further weighted by the tensor derived from the crude foreground image, result of background subtraction. Therefore the final structure tensor is expressed as follows:

$$J^A = w_1 J_{\rho_1}(\nabla S_{\sigma_1}) + w_2 J_{\rho_2}(\nabla V_{\sigma_2}) + w_3 J_{\rho_3}(\nabla F_{\sigma_3}) \tag{4.32}$$

where S is the saturation image, V is the value image, and F is the resulting foreground from the background subtraction process. The elements of tensor matrices are added element-wise after being weighted by the factors w_1, w_2, and w_3. The images in Figure 4.12 show the tensor field for saturation and value components, as well as the tensors for the foreground image. It can be easily noted that



(a) (b) (c)

Figure 4.13: Diffusion results for the edge image (a). (b) is the diffusion result for weighted average tensors of saturation and value components and (c) is the result for weighted average tensors of saturation, value and foreground contour images.

the tensors of the foreground image are well stretched around the contour of the silhouette inhibiting in this way the diffusion towards irrelevant directions. This additional property is revealed by the resulting diffused image as illustrated in Figure 4.13 , where the initial edge image Figure 4.13-(a) is diffused to achieve closed contours. On one hand is the diffused contour without the foreground image tensor field Figure 4.13-(b) where the diffusion expands to the objects that belong to background and on the other hand in Figure 4.13-(d) is the diffused contour which, due to the tensor field of foreground forms well rounded shape.

Once the closed contour is obtained it is filled and slightly eroded to thin the contour and eliminate any extended edges, which lie out of the object contour.

Another additional use of the structure tensor throughout the whole algorithm, in various steps, is the calculation of the direction of the gradient, which is given by the following formula.

$$\theta = \frac{180}{\pi}\text{atan2}\ (v_{1a}, v_{1b}) \tag{4.33}$$

where θ is the angle in 2D space and v_1 is the eigenvector with the maximum eigen-value.

4.6 Experimental Results and Analysis

To test the proposed system, ChangeDetection.net [49] video database was used as the benchmark dataset. All possible scenarios were included to ensure an overall objective evaluation of the system. The shadow line removal as proposed by J. F. Lalone was not applied since it was computationally intensive for this kind of application.

The code was developed in MATLAB and the experiments were conducted on a

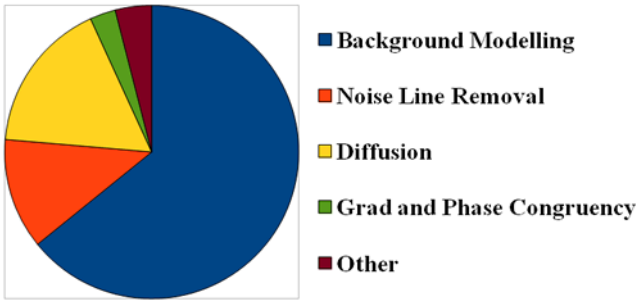


Figure 4.14: Distribution of processing time for the different parts of the system.

machine with $2.4GHz$ processor and $4GB$ memory. For the reason that MATLAB is a prototyping language, its code execution is not efficient in computationally intensive applications. Hence, the average processing time for a 320×240 frame is 30 seconds of witch, approximately 19.25 seconds are used for background modelling and updation, 0.85 seconds for phase congruency feature calculations, 3.66 seconds for noise line removal, 5.04 seconds for diffusion, and the rest of the time is used for other minor tasks (see Figure 4.14). It was found from the experiments that, if coded in C++, the background modelling based on GMM works at a rate of up to 30 frames per second. This leads to the conclusion that if the complete proposed system was coded in a compiled programming language, the worst case scenario performance would be less than 10 seconds per frame.

4.6.1 Experiments Set Up

Before presenting the experimental results it is important to specify the experimental set up, which includes some details on implementation and the choice of important parameters.

To find separate edge segments for post processing the MATLAB function "edgelink.m" developed by P. Kovesi was used [73]. His work on phase congruency is also available as a MATLAB implementation "phasecong3.m" from the same source. As input to "phasecong3.m", it is required to provide with a number of variables; the most important of which are: a greyscale image, number of wavelet scales = 4, number of filter orientations = 6, wavelength of smallest scale filter = 3 and the scaling factor between successive filters = 1.8. The values of parameters were selected as those which give the best results.The resulting phase congruency in 6 orientations are linearly adjusted to the range [0,1].

For fast implementation purposes the EEF was not employed for edge detection but only for background subtraction; instead, the traditional Canny edge detector was used. After segmenting the foreground contour with the use of background subtraction, raw foreground was morphologically closed before being reflected onto

the edges.

All edge segments at different scales are processed separately and the four features obtained are forwarded to an SVM classifier. The LIBSVM library [21] was used as a tool to implement the classification. The classifier was trained with features computed from a pair of 2 different frames for each of the following datasets: 'PETS2006', 'pedestrians', 'backdoor', 'busStation', 'canoe', 'winterDriveway', and 'bungalows'. Foreground and background samples were taken to train the classifier, with features of weak shadow also included in the background. From the last mentioned dataset only the foreground information was considered for training as the shadow conditions were extreme and thus more sophisticated features are needed to handle them. Figure 4.15 shows the layout of foreground-background samples with respect to one of the four features S_1, S_2, S_3 and S_4.

Foreground is represented by red coloured points while background by blue. It can be noticed that the features are partially complementary and each covers the other there where it fails. The features that seem to separate better the clusters are S_2 and S_3 confirming that the simple colour ratios measure S_1 is not sufficient on its own. A 3D view of features is visualised in Figure 4.16 where the separation between the two clusters is more obvious.

The classifier was trained with 3-fold cross validation to select the kernel and the training parameters. Among linear, polynomial and Radial Basis kernel functions, the polynomial kernel of 3^{rd} degree produced the highest average accuracy of 87.72% with training parameters $C = 7.4643$ and $\gamma = 1.8661$. The final model was trained with the entire set of samples. In the future it is possible to enrich the training set and even expand the set of features for more accurate results. To remove noise edges in thermal videos only the measure S_1 was used, with the threshold 0.95.

Once all the edges are classified, the ones that belong to background are removed. The rest of the edges are combined into one edge image that will be defused for closed contour recovery. As it was explained earlier in this chapter three structure tensor fields are combined into one according to Equation 4.32

For colour videos the weights are $w_1 = w_2 = 0.46$ and $w_3 = 0.08$ and the scale parameters are $\rho_1 = \rho_2 = \rho_3 = 4$, $\sigma_1 = \sigma_3 = 2$, and $\sigma_2 = 1$. For the thermal videos, since the saturation component is not available the structure tensors are computed over the greyscale frame and the segmented raw foreground. In this case the weights become $w_1 = 0.97$ and $w_2 = 0.03$, and the scale variables are $\rho_1 = \rho_2 = 4$, $\sigma_1 = 1$ and $\sigma_2 = 2$. The parameters were selected in such a way so that they facilitate acceptable results. The diffusion process is completed in 70 iterations, which are enough to recover closed contours. A longer period of diffusion, in case of noisy complex background, would result in contours extended

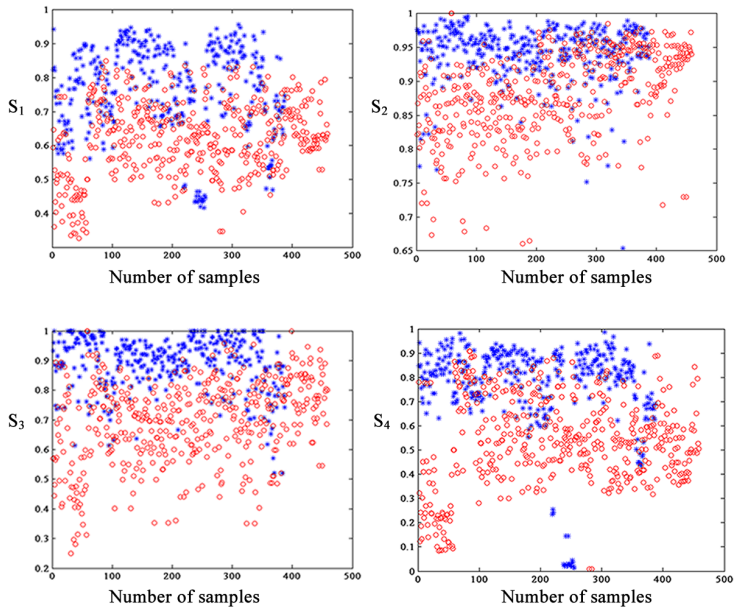


Figure 4.15: One-dimensional representation of feature clusters. From left to right are visualised the features S_1, S_2, S_3 and S_4. The horizontal axis represents the number of samples while the vertical axis the feature value for each sample point. Foreground is represented by red coloured dots while background by blue coloured dots.

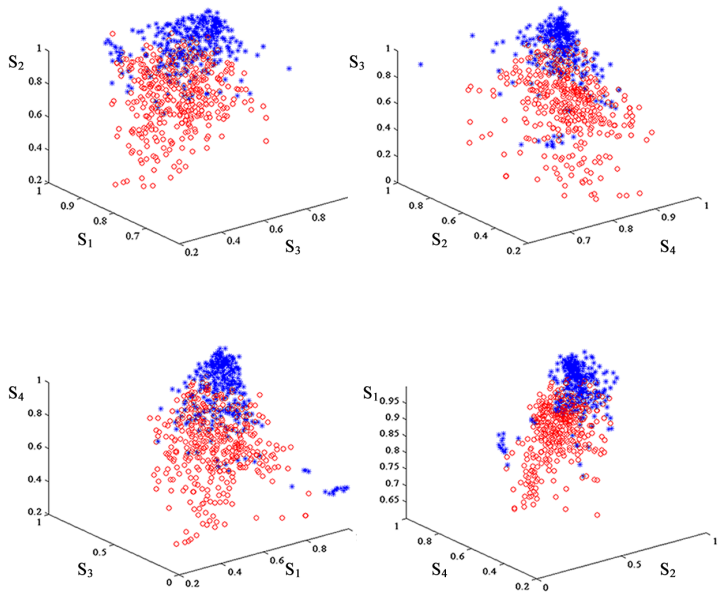


Figure 4.16: Three-dimensional representation of features. Each graph combines three different features. From left to right are visualised the feature triplets $S_1S_2S_3$, $S_2S_3S_4$, $S_3S_4S_1$, and $S_4S_1S_2$. Each axis corresponds to one of the three features. Foreground is represented by red coloured dots while background by blue coloured dots.

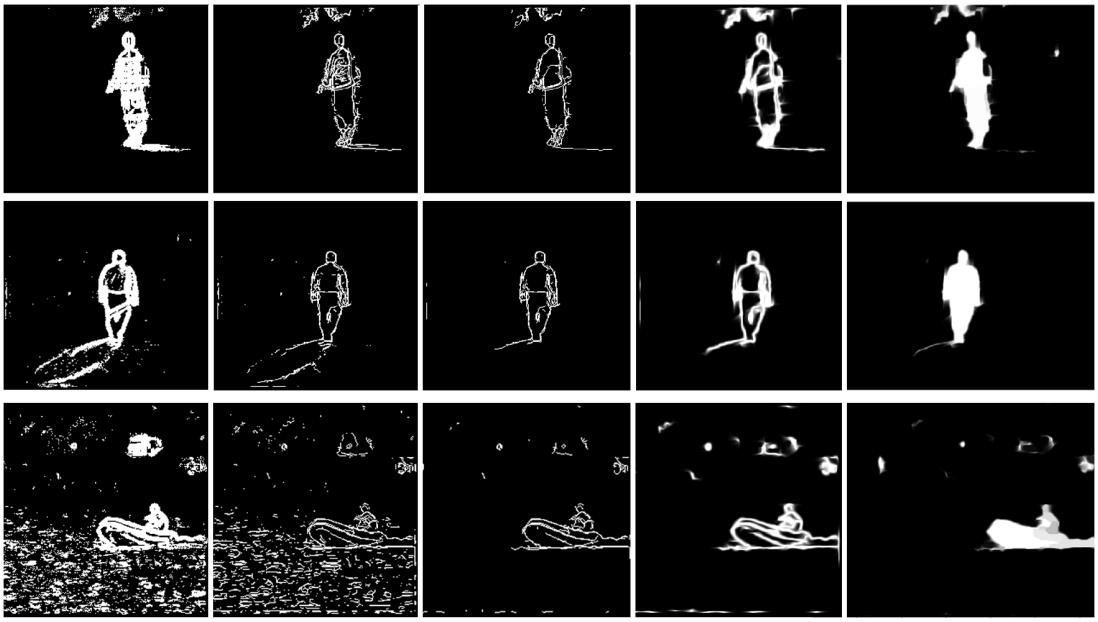


Figure 4.17: Some examples of intermediate results from the proposed foreground segmentation procedure. From left to right, the first column shows the background subtraction result, the second column the recovered foreground edges, the third column the edges after noise line removal, the forth column the results of edge completion and the last one, the filled and eroded contours.

away from the object into the background and would add more thickness to the contours. The diffused contour is flood filled to segment the foreground silhouette and morphologically eroded by a square structuring element of $side = 4$ pixels to attenuate any occurring noise extensions. Figure 4.17 shows some example results for all important stages of the algorithm starting with the raw foreground image, followed by its reflection on the edges, then the refined edge image, the closed contour and finishing with the filled and eroded image. In the example of the image boat, it is displayed how well the method removes the edges that belong to waves, while in the example from the 'backdoor' dataset, the shadow edge is partially removed. Further the noise and some parts of shadows in the cubicle image are effectively eliminated.

4.6.2 Experiments Analysis

The results of benchmark algorithms selected for comparison with the proposed method were taken from the ChangeDetection.net website [49]. The scenarios that the ChangeDetection.net covers include baseline simple cases, shadow sequences, dynamic background sequences, intermittent object motion, camera jitter and thermal video sequences. Each scenario includes a number of videos with tempo-

ral ROIs varying from 1000 to 5000 frames. ChangeDetection.net project features the results from most of the influential publications of first generation algorithms on foreground segmentation in video. It is also the home of the IEEE "change detection workshop"displaying a range of applications of different levels of performance. Therefore, it was easier and more reliable to derive the experiments results for comparison from there.

Specifically, the algorithms chosen spread across the whole range of overall performance levels; starting with the standard Stauffer-Grimson GMM (GMM-SG) [130] and its improved version by P. KaewTraKulPong (GMM-KAEW)[68], followed by a non-parametric density estimation algorithm which integrates spatio-temporal features by Y. Nonaka et al. (KDE-IST)[101], a multi-layer background subtraction technique based on colour and texture features by J. Yao and J.-M. Odobez (MLBS) [151], a self-organizing, through artificial neural networks, background subtraction algorithm by L. Maddalena and A. Petrosino (SOBS-SC)[89], a visual background extractor (ViBe+)[137] of M. Van Droogenbroeck and O. Paquot, a pixel based adaptive segmenter of M. Hofmann et al.(PBAS)[59] and finally ending with the method with the best overall performance which maintains complementary GMM background models (SGMM-SOD) of R. Evangelio and T. Sikora[43].

To measure the accuracy of the methods we employ the following metrics:

- Recall: Rec= TP / (TP + FN)
- Precision : Prec= TP / (TP + FP)
- Specificity: Spec= TN / (TN + FP)
- False Positive Rate: FPR= FP / (FP + TN)
- False Negative Rate: FNR= FN / (TP + FN)
- Percentage of Wrong Classifications: PWC=100 * (FN + FP) / (TP + FN + FP + TN)
- F-Measure : F= (2 * Prec* Rec) / (Prec + Rec) which is the weighted harmonic mean of Precision and Recall, so it can be regarded as an overall accuracy measure.

where, TP: True Positive FP: False Positive FN: False Negative TN: True Negative

Tables 4.2 to 4.7 display all the above defined metrics for the proposed Closed Foreground Contour (CFC) method for the video sequences contained in each scenario separately. Furthermore, the average results achieved by the CFC for

Table 4.2: Metrics for the baseline sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	PETS2006	0.9191	0.9976	0.0024	0.0011	0.3437	0.8335	0.8742
	highway	0.9474	0.9774	0.0226	0.0033	2.4334	0.7257	0.8219
	office	0.5266	0.9947	0.0053	0.0351	3.7627	0.8803	0.6590
	pedestrians	0.9377	0.9992	0.0008	0.0006	0.1393	0.9220	0.9298
Average results for the above video data.								
algorithms	CFC	0.8327	0.9922	0.0078	0.1673	1.6698	0.8404	0.8212
	GMM-SG	0.8180	0.9948	0.0052	0.1820	1.5325	0. 8461	0.8245
	MLBS	0.8456	0.9984	0.0016	0.1544	0.8993	0. 9655	0.9004
	SGMM	0.9334	0.9974	0.0026	0.0666	0.5494	0. 9113	0.9212

Table 4.3: Metrics for the shadow sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	backdoor	0.9358	0.9942	0.0058	0.0013	0.7006	0.7645	0.8415
	bungalows	0.5669	0.9884	0.0116	0.0276	3.6862	0.7574	0.6484
	busStation	0.9011	0.9894	0.0106	0.0038	1.3820	0.7658	0.8280
	copyMachine	0.6843	0.9880	0.0120	0.0235	3.3056	0.8092	0.7415
	cubicle	0.9455	0.9959	0.0041	0.0011	0.5123	0.8209	0.8788
	peopleInShade	0.8718	0.9924	0.0076	0.0077	1.4397	0.8729	0.8724
Average results for the above video data.								
algorithms	CFC	0.8176	0.9914	0.0086	0.0108	1.8377	0.7985	0.8018
	GMM-SG	0.7960	0.9871	0.0129	0.2040	2.1951	0.7156	0.7370
	MLBS	0.8588	0.9912	0.0088	0.1412	1.5621	0.8099	0.8216
	SGMM-SOD	0.9191	0.9902	0.0098	0.0809	1.2534	0.8226	0.8646

each scenario are compared at the end of each table with the average results for the same scenario of three other algorithms. These three algorithms are SGMM-SOD of highest overall F-measure, MLBD of medium overall F-measure, and the traditional GMM-SG with low F-measure.

It is observed that the proposed method achieves high average results for the baseline, shadow, and thermal sequences. This is well understood if the design of the system is taken into account. Specifically, the precision and recall metrics attained for the 'PETS2006' and the 'pedestrians' video sequences are over 90% (Table 4.2). One of the primary goals was the shadow reduction, which explains the good average performance for the shadow sequence and the lowest FPR (Table 4.3). The high average F-measure value for the thermal sequence is a result of the fact that the primary background modelling is based on gradient and phase congruency features that do not depend on colour (Table 4.7). Also the method performs well for the 'boats' and 'canoe' video sequences of the 'dynamic

Table 4.4: Metrics for the dynamic background sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	boats	0.7721	0.9982	0.0018	0.0014	0.3176	0.7349	0.7531
	canoe	0.9359	0.9829	0.0171	0.0024	1.8762	0.6677	0.7794
	fall	0.8069	0.7535	0.2465	0.0035	24.5575	0.0557	0.1043
	fountain01	0.8944	0.9359	0.0641	0.0001	6.4178	0.0115	0.0226
	fountain02	0.8397	0.9932	0.0068	0.0003	0.7086	0.2113	0.3377
	overpass	0.7845	0.8973	0.1027	0.0029	10.4167	0.0940	0.1679
Average results for the above video data.								
algorithms	CFC	0.8389	0.9268	0.0732	0.0018	7.3824	0.2959	0.3608
	GMM-SG	0.8344	0.9896	0.0104	0.1656	1.2083	0.5989	0.6330
	MLBS	0.7584	0.9912	0.0088	0.2416	1.0758	0.6466	0.6278
	SGMM-SOD	0.7786	0.9966	0.0034	0.2214	0.6041	0.7044	0.6883

Table 4.5: Metrics for the intermittent object motion sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	abandonedBoxx	0.4546	0.9896	0.0104	0.0276	3.6146	0.6879	0.5475
	parking	0.5772	0.9964	0.0036	0.0354	3.5998	0.9311	0.7126
	streetLight	0.4001	0.9979	0.0021	0.0274	2.8207	0.8966	0.5533
	sofa	0.2691	0.9967	0.0033	0.0373	3.8559	0.8081	0.4037
	tramstop	0.2494	0.9832	0.0168	0.1642	14.851	0.7641	0.3760
	winterDrivewayy	0.7155	0.9970	0.0030	0.0021	0.5073	0.6454	0.6786
Average results for the above video data.								
algorithms	CFC	0.4443	0.9935	0.0065	0.0490	4.8750	0.7889	0.5453
	GMM-SG	0.5142	0.9835	0.0165	0.4858	5.1955	0.6688	0.5207
	MLBS	0.5012	0.9629	0.0371	0.4988	7.0245	0.6024	0.4816
	SGMM-SOD	0.7363	0.9909	0.0091	0.2637	2.5238	0.8141	0.7151

background' category (Table 4.4). This is due to the application of noise line removal in post processing stage. Since the edges that lie on the waves have great colour and texture similarity across them, they are successfully removed. However this is not the case for the 'fountain' videos as although the edges that belong to waves are removed, the ever running fountain jet is not possible to be removed as the surrounding area differs from the fountain jet. The method also performs poorly for scenes with waving trees such as 'fall', 'overpass' and some frames from 'highway' where the precision levels for 'highway' are relatively low and for the other two cases minimal (Table 4.4 and Table 4.2). It is also expected that the precision results for camera jitter will be very low (see Table 4.6). This is justified by the fact that the primary foreground segmentation via background subtraction is based on features that occur from neighbouring pixel differencing. Hence, if the

Table 4.6: Metrics for camera jitter sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	badminton	0.9081	0.9326	0.0674	0.0033	6.8208	0.3237	0.4773
	boulevard	0.7344	0.8592	0.1408	0.0131	14.6675	0.2044	0.3198
	sidewalk	0.9290	0.6468	0.3532	0.0019	34.5842	0.0658	0.1229
	traffic	0.4230	0.9129	0.0871	0.0383	11.7641	0.2437	0.3092
Average results for the above video data.								
algorithms	CFC	0.7486	0.8379	0.1621	0.0142	16.9592	0.2094	0.3073
	GMM-SG	0.7334	0.9666	0.0334	0.2666	4.2269	0.5126	0.5969
	MLBS	0.6903	0.9905	0.0095	0.3097	2.1628	0.7905	0.7311
	SGMM-SOD	0.6113	0.9907	0.0093	0.3887	2.3608	0.8040	0.6724

Table 4.7: Metrics for thermal video sequences.

	video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
CFC	park	0.6452	0.9866	0.0134	0.0122	2.4700	0.6230	0.6339
	diningRoom	0.6341	0.9929	0.0071	0.0344	3.7881	0.8940	0.7419
	corridor	0.2976	0.9983	0.0017	0.0137	1.5093	0.7772	0.4304
	library	0.3477	0.9970	0.0030	0.1558	12.8179	0.9649	0.5112
	lakeSide	0.7879	0.9981	0.0019	0.0044	0.6138	0.8974	0.8391
Average results for the above video data.								
algorithms	CFC	0.5425	0.9946	0.0054	0.0441	4.2398	0.8313	0.6313
	GMM-SG	0.5691	0.9946	0.0054	0.4309	4.2642	0.8652	0.6621
	MLBS	0.5072	0.9986	0.0014	0.4928	3.8704	0.9611	0.6331
	SGMM-SOD	0.6396	0.9971	0.0029	0.3604	1.6846	0.9471	0.7353

directions of object and camera motion coincide, it will be difficult to recover the moving contours and at the same time all the stationary contours will appear as exhibiting motion. In this case the noise line removal will not induce any change as the stationary objects differ from the background.

Since the system does not have a long term memory to store the moving objects that stopped moving, the recall values for the 'office' video from the baseline category (Table 4.2) and all the videos, except for the 'winterDriveway', from the intermittent object motion category (Table 4.5), are very low. To handle this kind of situation complementary background models should be maintained as in SGMM-SOD of R. Evangelio and T. Sikora [43] and in MLBS of J. Yao and J.-M. Odobez [151] or the background should be updated selectively as in SOBS-SC of L. Maddalena and A. Petrosino [89]. A significant reason for the success of these methods is due of their ability to store the foreground objects that ceased moving. For the category of camera jitter, algorithms that use spatial mechanisms like

Table 4.8: Comparative averages of all metrics of available scenarios for the selected methods.

	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
Proposed CFC	0.7041	0.9561	0.0439	0.2959	6.1606	0.6274	0.5779
Proposed CFC no camera jitter	0.6952	0.9797	0.0203	0.3048	4.0009	0.7110	0.6321
Proposed CFC no camera jitter, overpass, fall, fountain01	0.6973	0.9926	0.0074	0.2729	2.7180	0.7594	0.6846
GMM-SG	0.7108	0.9860	0.0140	0.2892	3.1037	0.7012	0. 6624
GMM-KAEW	0.5072	0.9947	0.0053	0.4928	3.1051	0.8228	0.5904
KDE-IST	0.6507	0.9932	0.0068	0.3493	2.8905	0.7663	0.6418
PBAS	0.7840	0.9898	0.0102	0.2160	1.7693	0.8160	0.7532
MLBS	0.6936	0.9888	0.0112	0.3064	2.7658	0.7960	0.6993
SGMM-SOD	0.7697	0.9938	0.0062	0.2303	1.4960	0.8339	0.7661
SOBS-SC	0.8017	0.9831	0.0169	0.1983	2.4081	0.7315	0.7283
ViBe+	0.6907	0.9928	0.0072	0.3093	2.1824	0.8318	0.7224

SOBS-SC in [89], and ViBe+ in [137] perform best. This is visible in Figure 4.21 for the 'badminton' case.

To continue with the comparison the average values of all metrics in Tables 4.2-4.7 for the proposed CFC method and all the benchmark algorithms have been summarised in Table 4.8. In general the proposed method works relatively well after removing the 'camera jitter' category and other three specific video sequences from the 'dynamic background' category.

For better comparison of the methods, the averages of all metrics have been computed over some selected scenarios. These video scenarios and the average

Table 4.9: Comparative averages of all metrics for the specific video sequences: pedestrians, PETS2006, highway, office, fountain02, boats, canoe, winterDriveway, backdoor, busStation, bungalows, peopleInShade, cubicle, copyMachine, park, lakeside, corridor, dinindRoom, library.

Average	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure	TSE
CFC	0.7429	0.9927	0.0073	0.0184	2.3114	0.7976	0.7480	1977914
GMM SG	0.7349	0.9912	0.0088	0.0189	2.4858	0.7648	0.7170	3624078
GMM-KAEW	0.5446	0.9960	0.0040	0.0268	2.7757	0.8811	0.6313	3158727
KDE-IST	0.6439	0.9954	0.0047	0.0263	2.7546	0.8110	0.6655	2477131
PBAS	0.8057	0.9922	0.0078	0.0059	1.2971	0.8351	0.7742	3484131
MLBS	0.7274	0.9938	0.0062	0.0172	2.1027	0.8632	0.7480	3438928
SGMM-SOD	0.8017	0.9929	0.0071	0.0070	1.3302	0.8530	0.7937	3830332
SOBS-SC	0.7955	0.9890	0.0110	0.0082	1.8090	0.8024	0.7689	3771922
ViBe+	0.7107	0.9940	0.0060	0.0133	1.7671	0.8452	0.7439	3547007

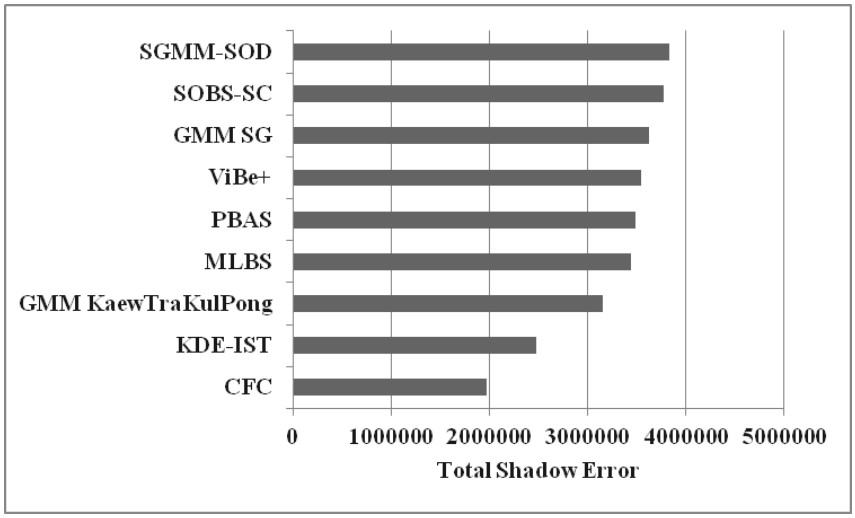


Figure 4.18: Total Shadow Error for the 9 background subtraction techniques for the video sequences specified in Table 4.9.

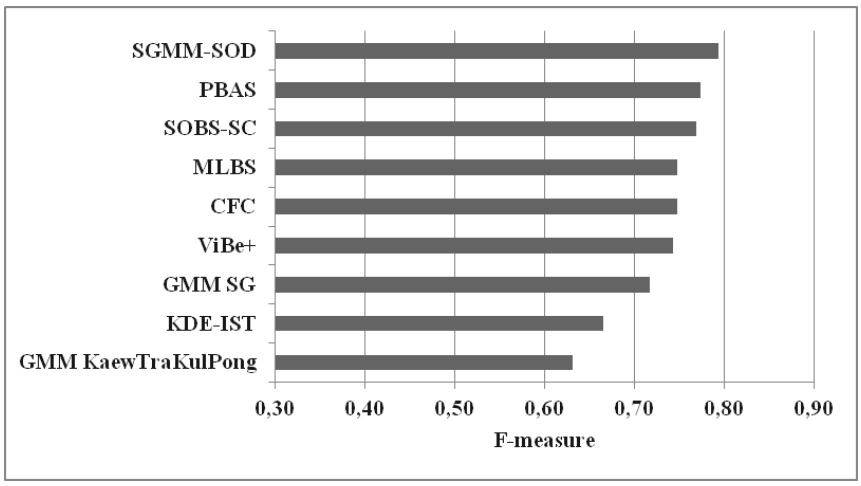


Figure 4.19: Average F-measure for the 9 background subtraction techniques for the video sequences specified in Table 4.9.

metrics obtained are shown in Table 4.9. An additional measure incorporated in this table is the Total Shadow Error (TSE), which measures the total number of false positives that occur due to shadow. As it is shown in Table 4.9 the proposed CFC achieves the lowest shadow error, which is confirmed by the qualitative examples illustrated in Figure 4.20. The corresponding bar charts that compare the TSE and F-measures of Table 4.9 are displayed in Figure 4.18 and Figure 4.19 respectively. The bar chart in Figure 4.18 demonstrates the big difference between the TSE of the proposed method and the rest methods. As it is shown in Figure 4.19 the F-measure of the proposed CFC system for the specific group of videos is ranked as average among the rest of the techniques.

A visual comparison of the methods in Figure 4.20 reveals that the proposed CFC method removes effectively the light shadows in comparison to the rest of

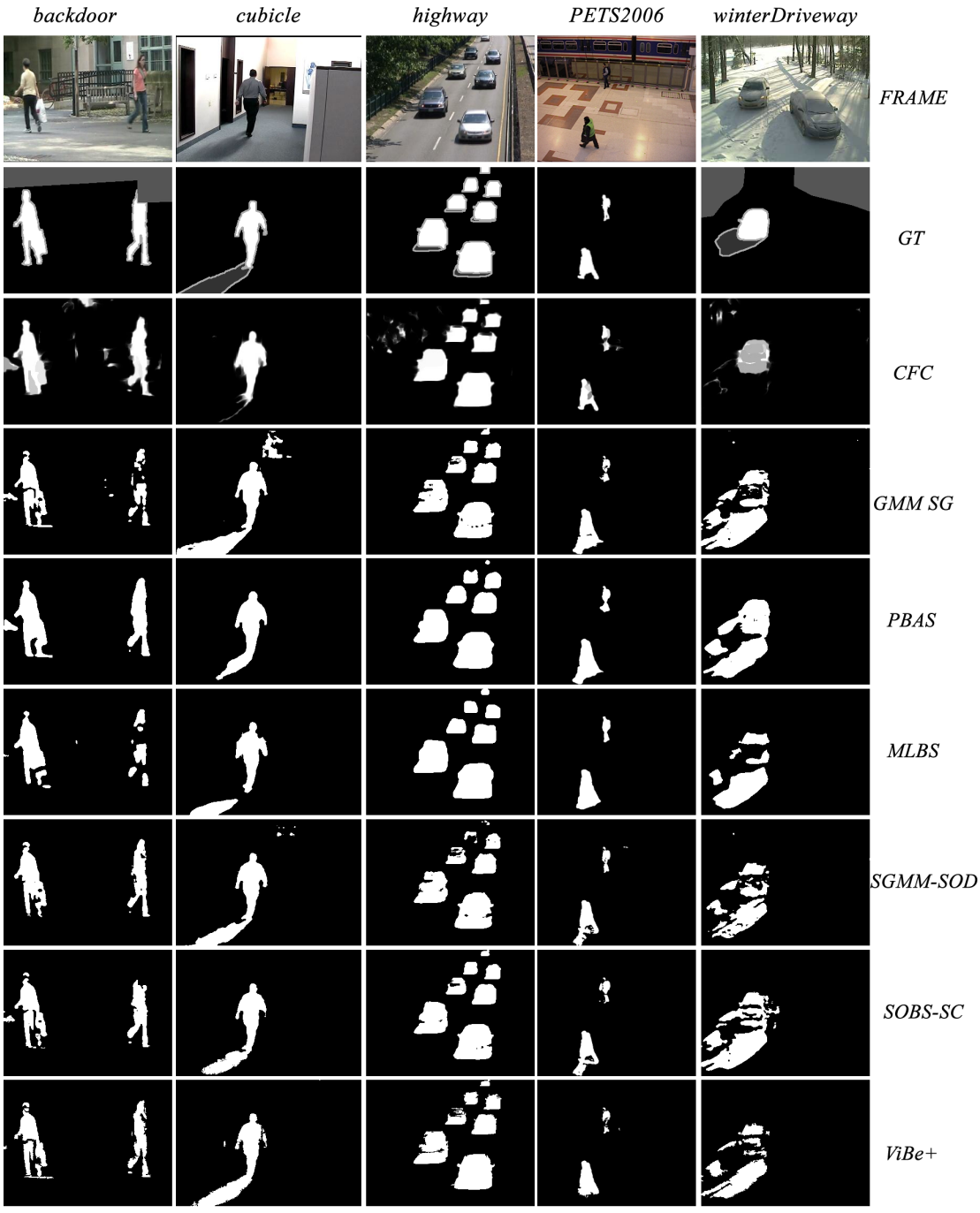


Figure 4.20: Qualitative results comparing the proposed CFC method with a selection of methods under analysis.

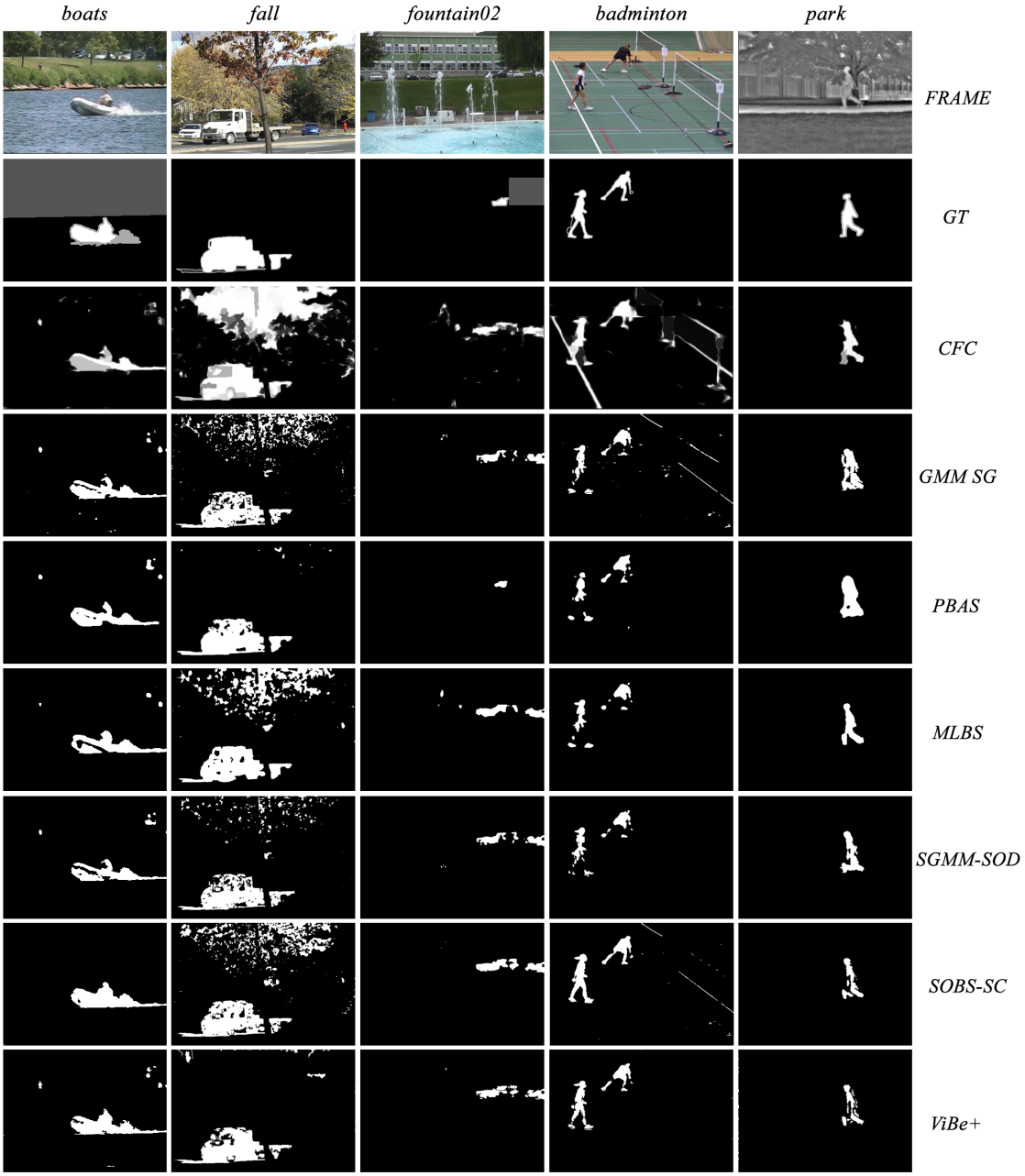


Figure 4.21: Qualitative results comparing the proposed CFC method with a selection of methods under analysis.

Table 4.10: Comparative results for the 'shadow' category while employing the EEF and the Gaussian filters.

video	Recall	Specif.	FPR	FNR	PWC	Prec.	F-measure
backdoor EEF	0.9358	0.9942	0.0058	0.0013	0.7006	0.7645	0.8415
backdoor Gaussian filter	0.9491	0.9780	0.0220	0.0010	2.2616	0.4667	0.6257
bungalows EEF	0.5669	0.9884	0.0116	0.0276	3.6862	0.7574	0.6484
bungalows Gaussian filter	0.6725	0.9856	0.0144	0.0209	3.3145	0.7491	0.7087
busStation EEF	0.9011	0.9894	0.0106	0.0038	1.3820	0.7658	0.8280
busStation Gaussian filter	0.9578	0.9756	0.0244	0.0016	2.5013	0.6011	0.7386

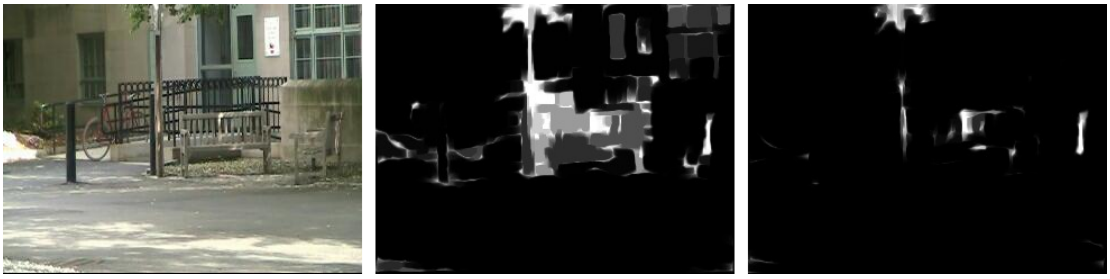


Figure 4.22: Qualitative comparative results for the 'backdoor' video sequence while employing the EEF and the Gaussian filters, respectively.

the background subtraction techniques. Special attention should be given to the 'PETS2006' case which is one of the most basic scenarios that occur in surveillance and where the other methods fail removing the ground shadow. Hard shadow examples such as those present in the 'highway' image example are not possible to be removed with the proposed method. A further observation noticeable in the examples of Figure 4.20 and Figure 4.21 is the wholeness of the achieved segmented foreground, as broken objects is a usual case for many background subtraction techniques. In the proposed CFC the contour completion via anisotropic diffusion ensures whole objects. However, this attribute results in a sub-optimal output in case of waving trees as in the 'fall' example of Figure 4.21 (explained in section 8.2.). Finally, the thermal frame example confirms the ability of the CFC system to provide with accurately defined foreground shapes.

Last but not least, Table 4.10 and Figure 4.22 are given to show why the Gaussian filter is not preferable for smoothing before computing the derivative. The table shows that although the recall values have been increased, the precision values for two cases have been decreased. As it appears in Figure 4.22, the Gaussian filter is more sensitive to illumination changes than the EEF filter causing a large

number of false positive detections. Therefore, the use of the Gaussian filter is not advisable.

4.7 Summary and Discussion

In this chapter a method for closed foreground contour segmentation and ground shadow reduction in video sequences has been proposed. The procedure begins with the acquisition of crude foreground contours via background subtraction, based on GMM, which through a number of steps that follow is refined and finally closed into an accurate foreground contour.

Specifically, a novel smoothing filter is applied to each incoming frame prior to computation of gradient features, which ensures the maximum continuation of contours and tolerance to illumination changes. Additional phase congruency features contribute to the detection of contours that the gradient omits. Subsequently after the raw foreground contours are reflected onto the edges of the incoming frame, a noise edge removal technique based on colour ratios is applied. This is an important step as it also decides whether an edge lies over a shadow contour. Finally a contour completion technique based on anisotropic diffusion is applied to achieve a closed foreground contour that is filled to define foreground regions.

The experiments have shown that the method performs very well in the presence of shadows. It attains the lowest occurrence of false positives due to shadows in comparison with the state-of-the-art techniques. Dynamic background scenes like sea waves are handled successfully since there is significant level of similarity in colour and texture across edges that lie on waves. However the noise reduction techniques fail to remove the tree leaves. Further, the contour completion technique underperforms in backgrounds containing waving trees, as the completion of undeleted contours means the detection of the entire tree. It was also shown that for a group of specific scenarios where the proposed CFC system is designed to perform well, the system is ranked as medium among the rest of the techniques.

Key step on the CFC algorithm is the noise line removal technique as it is the one that decides which of the detected edges belong to the foreground and which not. This step relies on colour ratio features and a classifier that takes the decision. This gives potential for further research into additional features and further training of the classifier, which could improve the results obtained above.

Chapter 5

Viewing Direction Estimation and Carried Bag Type Recognition (BTR) for a COD System

This chapter deals with viewing direction estimation of a walking person for the purpose of using it subsequently for COD as proposed by D. Damen and D. Hogg [33, 34]. Additionally, bag type recognition algorithm is devised based on the location of the detected object relatively to the human body. The chapter begins with section 5.1 where the COD system of Damen & Hogg is analysed to identify its shortcomings for the purpose of later proposing solutions. Section 5.2 presents the proposed viewing direction estimation and bag type classification algorithms. Section 5.3 provides experimental results and a detailed analysis. Finally section 5.4 concludes, identifying future directions of research.

The COD of Damen & Hogg is the state of the art method extensively tested on a large dataset and exhibits the best known results. Their COD algorithm is based on a temporal template analysis, as clarified in the chapter 3 and requires a moving foreground segmentation. The foreground silhouettes are segmented by a GMM based background modelling algorithm as proposed by P. KaewTraKulPong in [68]. The tracking part was performed by a simple connected component analysis, based on features such as for e.g., aspect ratio of the object, colour properties and location. The readers are advised to familiarise themselves with the COD of Damen & Hogg, described in chapter 3 before proceeding.

Several techniques have been proposed recently for person's body orientation estimation or body pose recognition; these are mostly based on the direction of gradient features, and a few are summarised below. M. Enzweiler and D. M. Gavrila begin their work with pedestrian classification to continue with body orientation estimation in a unified fashion. The classification is performed with

a linear SVM and local receptive field neural network (NN/LRF), on HOG [42]. Another approach, which utilises HOG features in conjunction with LBPs for person detection and orientation classification is presented in [145]. The authors suggest that substituting the conventional SVM with an SVM decision tree gives significantly improved results. In [24] the authors achieve body pose classification by a sparse representation of multi-level HOG features. They also show that sparse representation attains higher accuracy levels than SVM training. A different approach in [3], by M. Andriluka et al., utilises pictorial structures extended to 8 viewpoint specific models for human detection and viewpoint estimation.

In contrast to the above mentioned methods the following ones assume a detected person and an extracted silhouette. A. Agarwal and B. Triggs in [1] introduce the relevance vector regression for 3D human pose estimation. They apply a Relevance Vector Machine on a Histogram of Shape Context, previously undergone dimensionality reduction. The same features are adopted by Rybok et al. who classify the body orientation into 12 categories using SVM and Nearest Mean classification [116]. In [104] the authors apply body orientation estimation from the top camera view by shape context matching of the upper body with predetermined models.

The proposed direction estimation stays away from the complex feature descriptions and classification methods. Since primarily it is to be used with temporal templates for COD purpose, the devised features are simple and based on the geometry of the upper part of human silhouette. Though seemingly simple, the method attains high accuracy classification results.

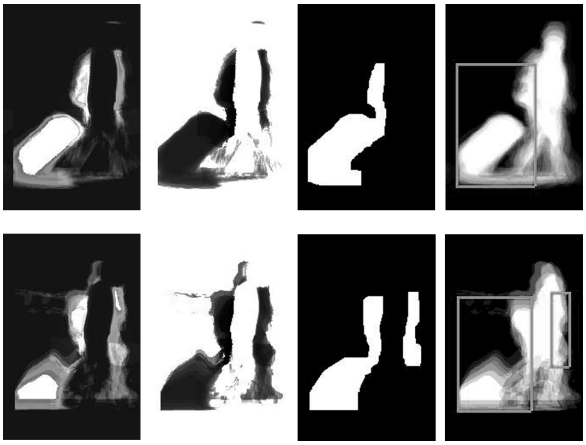
For carried object recognition no relevant literature was found. Therefore the proposed algorithm is simple in nature and exploits one of the most obvious features that characterise a bag, which is its location near the human body.

5.1 Evaluation of the Damen & Hogg COD System and the Proposed Improvements

5.1.1 Shortcomings of the Damen & Hogg COD System

A detailed examination of Damen & Hogg's approach revealed the following shortcomings, if addressed, can improve its accuracy and practical usability:

Exhaustive search: One of the shortcomings is the exhaustive search used for finding the best matching exemplar from the database. The baseline of the exhaustive search includes: the collection of different sizes and rotations of the exemplars, which share the same viewing point with the temporal template and



(a) (b) (c) (d)

Figure 5.1: The problem of connected bags as output of Graph Cuts in two examples; (a) and (b) are the input of Graph Cuts, (c) is the segmented output, (d) is the result reflected on the temporal templates.

the selection of the one that minimises the sum of difference between the exemplar and the temporal template. The system goes through all available exemplars translating them up to 30 pixels towards each direction (horizontally and vertically), 3 pixels at a time, until the minimum sum confirms the best match. It is obvious that exhaustive search is not the best option as it is a time consuming process.

Connected bags: A further problem is associated with the output of Graph Cuts algorithm (see Appendix C) which segments the shape of the detected bags. Two images are fed as input of Graph Cuts algorithm; the probability of bag location multiplied by transformed trained bags model (rescaled translated and rotated to match the temporal template) and the probable noise location, multiplied by the transformed inverse of the trained bags model. The referred probable bag and noise locations are probability estimations derived from the difference image, which is the result of subtraction between a temporal template and its matched exemplar. Provided that the alignment of the exemplar with the temporal template is accurate, Figure 5.1 illustrates the input of Graph Cuts in columns (a) and (b), and the output in column (c) reflected on the temporal template in column (d). It is observed that the person in both of the illustrated cases carries a backpack and a pulling luggage. However the detection of the bags as separate items was not successful as they appear as a one piece/item (see Figure 5.1 (c)).

Homography and direction estimation: As it was mentioned before, the mapping of motion from the image plane to the ground plane is achieved with the use of a homography transformation, which requires 8 corresponding points between the two planes. These points should be provided manually to the D. Damen's system, making the approach highly dependent on human intervention.

(a) (b) (c) (d)

Figure 5.2: The connected bag separation process takes as input the binary output of graph cuts (a) and the respective area on the temporal template (b). After the application of the thresholding the result is (c) and the outcome of morphological opening is (d).

Therefore, the system becomes impractical while trying to automatically or semi-automatically process videos from different cameras. Moreover, to estimate the position of the camera a set of vertical lines is needed to indicate the vertical vanishing point on image plane, which is a further complication. The vanishing point is transformed to world coordinates using the previously calculated homography matrix. Finally the motion vector of the person is estimated by fitting the points which represent the person's position on the ground over a walking cycle using the Linear Least Squares method (LLSQ).

5.1.2 Improving the Shortcomings of the Damen & Hogg COD System

Optimisation of the exhaustive search: The first area of improvement is the exhaustive search applied for the best model match selection. As the database of exemplars contains silhouettes of different sizes, it is not practical to scan through all the exemplars. Therefore, initially the size of the exemplar is decided in accordance to the height H of the pedestrian, i.e. the difference between the highest and the lowest non zero pixel of the temporal template. From the availability of 13 different exemplar sizes only the three closer to the height of the silhouette are selected. It is safe to assume that three closest sizes are enough to match exactly with the template or even to cover it. Next, the exemplar and the silhouette are aligned with the help of the vertical axis that passes through the centre of the head. Finally the shifts of 10 pixels or less are performed avoiding the temporary matching of all the sizes. To conclude, for the fact that sometimes the silhouettes of the pedestrians are oversized, the exemplar is being dilated if the following rule is true:

$$\left(\sum_{(x,y)=(1,\frac{H}{4})}^{w,\frac{H}{4}} |PT(x,y) - M(x,y)| \right) > 0.1 \tag{5.1}$$

where $PT(x, y)$ is the person temporal template and $M(x, y)$ is the best matched temporal model.

Separation of connected bags: As mentioned in the previous section sometimes two bags are recognised as one item. To prevent this, the binary output of the Graph Cuts algorithm (the detected bag) should be cleaned in the areas where the intensity of the pixels is low in relevance to the rest of the greyscale area occupied by the bag. The separation of bags is achieved via histogram based thresholding.

The intensity histogram of the greyscale bag area can be used to indicate which regions should be removed. Initially, the average value of intensities needs to be found (Equation 5.2).

$$\bar{M} = \frac{\sum_{i=1}^{m} I_i C_i}{\sum_{i=1}^{m} C_i} \tag{5.2}$$

where $I_i \in (0, 1)$ is the range of intensity values and C_i is the frequency of their occurrence. The next step involves the calculation of local maxima in the interval $(0, \bar{M})$. Hence $localmax = \max(C_i) : I_i \in (0, \bar{M})$. Finally, all the pixels with intensity values $I_i < localmax$ should become 0. Afterwards, to smooth out the result, morphological opening followed by closing is applied to the thresholded image. The intermediate results of the procedure are shown in Figure 5.2 .

In our case the method is applied only for separation of large objects. Advantage of this method is the accurate definition of the shape of the bag, which could serve well in bag classification process which is based on bag size and position.

5.2 Extending the Practical Usability of the COD System

5.2.1 Viewing Direction Estimation

The primary aim is the complete disengagement of the system from the need for human intervention which will be achieved by designing a reliable method for direction estimation. Thus, the motion information of the tracked silhouette as in [33] is combined with an initial body pose classification based on shoulder shape properties of the temporal template.

Assuming that the enquired moving directions are the 8 illustrated in Figure 5.3 the final classification result should place the temporal template in one of the 8 categories. However, a careful examination of Figure 5.3 reveals that pairs of the 8 exemplars look similar and their shoulder region follows a distinctive pattern.

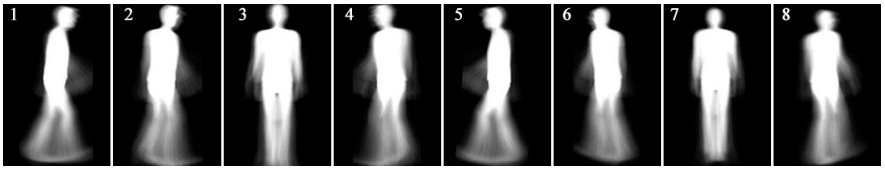


Figure 5.3: Exemplar temporal templates of 8 viewing directions.

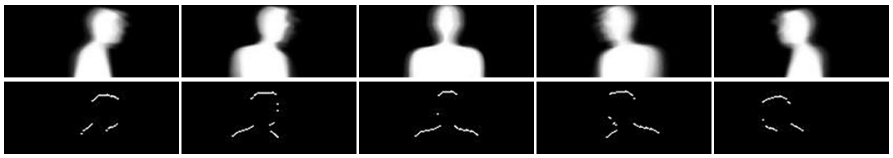


Figure 5.4: The shoulder and edge images that are used to extract information for classification. Here we have the exemplars 1 through 5.

Therefore, the classes could be reduced to five as in Figure 5.4 or even to three as follows: class 1 (directions 1 and 5), class 2 (directions 2, 4, 6, and 8), class 3 (directions 3 and 7). The information that will help refining the classification at a later stage is the motion information.

To present the features that are used for classification, it is first required to identify some landmark points that would enable the calculation of the features. Initially the temporal template is thresholded with Otsu's method (image histogram based thresholding method proposed by N. Otsu in [102]) to determine the binary template. Then, the height of the silhouette is defined as the difference of the topmost and bottommost non-zero pixels of the binary template. If the segmented silhouette in each frame is considered as a binary template then this direction estimation algorithm can be applied per frame basis as well. Since the human body is proportionate, its properties have been utilised to localize the head and shoulders. At this point it should be mentioned that although the calculations seem to be based on assumptions that probably do not apply to each and every individual, for the purpose of this method the accuracy of definition of body parts is satisfactory. The only requirement for this technique is that the camera is elevated at a height that facilitates the side (within a range) and not the top view. In the paragraphs that follow, all features are defined as variables with mathematical types.

If the height of the silhouette is H and the topmost point on the head is t then the vertical shoulder position (along the y axis) and the point that the image will be cropped is defined as $cut = t + H/4$ and the vertical centre of the head is defined as $VertHeadCent = t + H/17$. The sum of the pixels located at the horizontal line $VertHeadCent$ will give us the head width $HeadW$ and the maximum of horizontal projection of the cropped image the shoulder width $ShouldW$. Thus

the first feature is defined as

$$Head\text{-}Shoulder\ Ratio = \frac{HeadW}{ShouldW} \tag{5.3}$$

Next, the horizontal position (along the x axis) of shoulders and head should be found. Thus, the first nonzero pixel at position $x = cut$ is the left shoulder $HorShouldL$ and the last nonzero pixel is the right shoulder $HorShouldR$. Similarly the first non-zero pixel at the vertical position $VertHeadCent$ is the left head side $HeadL$ and the last one the right head side $HeadR$. The centre between two head sides is considered to be the horizontal head center $HorHeadCent$. Thus the next feature is the,

$$Distance\ Ratio = \frac{|HorShouldR - HorHeadCent|}{|HorHeadCent - HorShouldL|} \tag{5.4}$$

For the rest of the features we need the horizontal edges of Sobel derivative of the silhouette as shown in Figure 5.4. The features that are derived from the edges are the horizontal shoulder and head ranges: $HorShouldRangeL$, $HorShouldRangeR$, and $HorHeadRange$ which is right above the $VertHeadCent$. The reason that these features were selected becomes obvious from Figure 5.4 where the visible right and left shoulders width changes in accordance with the viewing direction. In addition, some vertical shoulder features should be taken into account. To calculate the vertical features it is essential to identify the pixels that are at the right and left side of the $HeadR$ and $HeadL$ and simultaneously occur only once through the vertical path from $y = VertHeadCent$ to $y = cut$. Once these pixels are isolated, their vertical mean, standard deviation, range and length are computed as $meanR$, $meanL$, $stdR$, $stdL$, $rangeR$, $rangeL$, $lengthR$, $lengthL$, respectively. It is also important to examine the slope of the shoulders; thus the found shoulder points are fitted to a line and its slope is denoted as $angleR$ and $angleL$.

The proposed final features are:

$$Shoulder\ Range\ Ratio = \frac{HorShouldRangeR}{HorShouldRangeL} \tag{5.5}$$

$$Head\text{-}Shoulder\ Range\ Ratio = 0.5\left(\frac{HorHeadRange}{HorShouldRangeR} + \frac{HorHeadRange}{HorShouldRangeL}\right) \tag{5.6}$$

$$Shoulder\text{-}Height\ Ratio = \frac{ShouldW}{H} \tag{5.7}$$

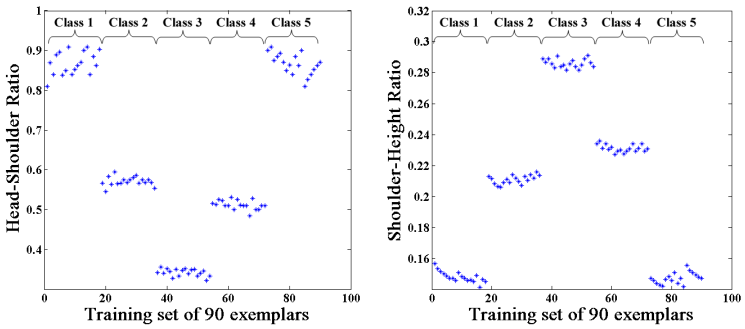


Figure 5.5: The two figures measure the values of *Head-ShoulderRatio* and *Shoulder-HeightRatio* respectively as they occur if applied on the exemplars database. Each of the 5 clusters identified consists of 18 exemplars of different sizes and in-plain rotations which are responsible for the small variation of feature values within each class.

$$Shoulder\ Vertical\ Range\text{-}Length\ Ratio = \frac{rangeR/lengthR}{rangeL/lengthR} \tag{5.8}$$

$$Right\ Head\text{-}Shoulder\ Distance = HeadR - HorShouldR \tag{5.9}$$

$$Left\ Head\text{-}Shoulder\ Disance = HorShouldL - HeadL \tag{5.10}$$

$$Mean\ Difference = |meanR - meanL| \tag{5.11}$$

$$Angle\text{-}Length\ Difference = ||angleR * lengthR| - |angleL * lengthL|| \tag{5.12}$$

$$Length\text{-}Shoulder\ Differece = \left| \frac{lengthR}{ShouldW} - \frac{lengthL}{ShouldW} \right| \tag{5.13}$$

$$Range\text{-}Length\ Differene = \left| \frac{rangeR}{lengthR} - \frac{rangeL}{lengthL} \right| \tag{5.14}$$

Some of these features, such as *Head-ShoulderRatio* and *Shoulder-HeightRatio*, are strong and capable of classifying the exemplar template models into 5 classes as in Figure 5.4 regardless the size and the in-plain rotation of the exemplar. The effectiveness of these features is demonstrated in Figure 5.5 where 5 different classes are distinguished (represented by 5 dotted clusters where each dot represents an exemplar template). Each cluster corresponds to a different viewing direction where the exemplar templates of different sizes and in-plain rotations share an approximately same feature value. Other features are comparatively weak and act as complementary when the strong ones fail in real life scenarios.

Body Orientation Estimation Using Decision Tree

The defined set of features will be used to classify the body orientation in one of the first 3 basic categories shown in Figure 5.4 according to the algorithm in Figure 5.6.

To select the threshold values for the algorithm the cluster plots of features have been constructed as illustrated in Figure 5.5. At an initial stage the thresholds were obtained from the database of exemplar models and at the second stage the thresholds were refined with data obtained from real life temporal templates. For instance the features $Head\text{-}ShoulderRatio$ and $Shoulder\text{-}HeightRatio$ have clearly defined thresholds between the 5 classes as shown in Figure 5.5. The algorithm has been simplified for presentation purpose to classify into the first 3 categories. However, the variables $DistanceRatio$ and $ShoulderRangeRatio$ indicate if the template is facing the right or left direction, attaining categorisation into 5 categories.

Body Orientation Estimation using SVM

Another possible way of classification is the exploitation of SVM classifier. However this method is not examined in this chapter. Instead, the section 6.3 and subsection 6.7.2 explain how the features:

1. $DistanceRatio$
2. $Head\_ShoulderRatio$
3. $Head\_ShoulderRangeRatio$
4. $ShoulderRangeRatio$
5. $ShoulderVerticalRange\_LengthRatio$
6. $Shoulder\_HeightRatio$
7. $RightHead\_Shoulderdistance$
8. $LeftHead\_Shoulderdistance$
9. $MeanDifference$
10. $Length\_ShoulderDifference$
11. $Angle\_LengthDifference$
12. $Range\_LengthDifference$

could be used to train the classifier and achieve high accuracy results.

Final Viewing Direction Estimation by Combining the Body Orientation with the Motion Vector

After acquiring the basic information about the pose of the temporal template, its viewing direction can be identified by combining its pose, and the angle between the image plane motion vector (unit vector) and a vertical unit vector.

BODY ORIENTATION-ESTIMATION(*all the calculated parameters*)

```
    // viewing directions 1 and 5
1   if 0 ≤ DistanceRatio ≤ 0.14 or 5 < DistanceRatio ≤ 60
2       if Head-ShoulderRatio ≤ 0.65 or Shoulder-HeightRatio > 0.23
3           pose = 2
4       else
5           pose = 1
        // viewing directions 3 and 7
6   elseif 0.60 < DistanceRatio ≤ 1.6 or 0.2 < ShoulderRangeRatio < 5
7       if (Head-ShoulderRatio ≤ 0.55 and Shoulder-HeightRatio > 0.20)
         or (0.4 < ShoulderRangeRatio < 1.9
         and Head-ShoulderRangeRatio < 5
         and Shoulder-HeightRatio > 0.20) or Shoulder-HeightRatio > 0.23
8           if MeanDifference > 7 or ((stdR > 4 or stdL > 4)
             and Angle-LengthDifference > 4.5)
9               pose = 2
10          else
11              pose = 3
12      else
13          pose = 1
        // viewing directions 2,4,6 and 8
14  elseif 0.14 < DistanceRatio ≤ 0.60 or 1.6 < DistanceRatio ≤ 5
    or ShoulderRangeRatio ≥ 5or ShoulderRangeRatio ≤ 0.2
15      if Head-ShoulderRatio > 0.65 and 0.2 < ShoulderRangeRatio < 5
16          pose = 1
17      else
18          if RightHead-Shoulderdistance ≥ 4
             or LeftHead-Shoulderdistance ≥ 4
             or (Shoulder-HeightRatio < 0.21
             and Head-ShoulderRangeRatio > 3)
             or Head-ShoulderRatio > 0.70
19              pose = 1
20          else
21              pose = 2
22  else
23      pose = 2
```

Figure 5.6: Human body orientation estimation algorithm

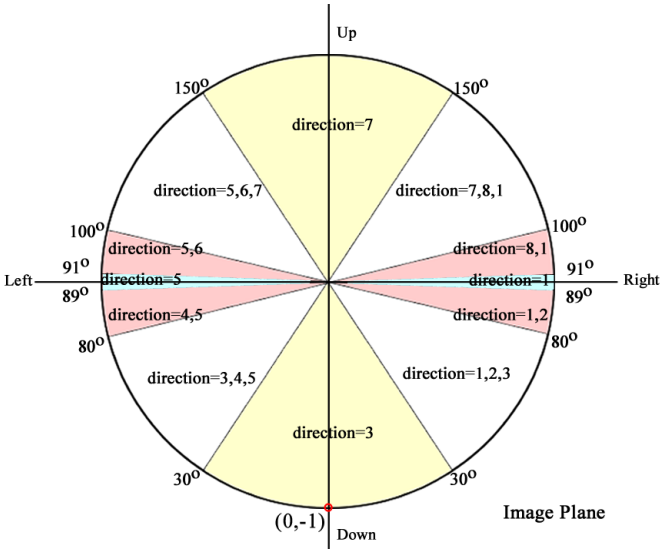


Figure 5.7: The unit circle with threshold angles on the image plane. The angles separate the circle into segments that suggest the most likely directions within them. The output of SVM classifier will decide which direction should be selected as final.

To calculate the motion vector, PCA is used for data fitting instead of the Linear Least Squares (LLSQ) method proposed by D. Damen. Let us assume that the position of the person in consecutive frames is represented by the set of points (x, y). The result of PCA on this set is the two principal components, which are eigenvectors with their eigenvalues. Thus, the two principal components are

$$\begin{aligned} \mathbf{P_1} &= a_1 x + b_1 y \\ \mathbf{P_2} &= a_2 x + b_2 y \end{aligned} \tag{5.15}$$

The eigenvector $[a_i, b_i]$ with the highest eigenvalue is the principal component, which fits the data to a single line. What makes PCA ideal for direction estimation is the fact that the eigenvector is a unit vector and the dot product of it with a vector $\mathbf{Q}$, which connects the origin with the position (0,-1) will give the angle, which defines the probable viewing direction of the person.

Therefore, if $\mathbf{Q} = -y$ then,

$$\mathbf{P} \cdot \mathbf{Q} = |\mathbf{P}| \cdot |\mathbf{Q}| \cos\theta \Rightarrow \cos\theta = \frac{\mathbf{P} \cdot \mathbf{Q}}{|\mathbf{P}| \cdot |\mathbf{Q}|} \tag{5.16}$$

A good reason to substitute the LLSQ method with PCA is the precision of data fitting. In some cases LLSQ methods fails to fit the data accurately because of the fact that the error is minimised only with respect to the dependent variable y. In contrast PCA minimises the error orthogonal to the model line.

The unit circle in Figure 5.7 is marked with the vertical unit vector $\mathbf{Q}$ and angles of high interest which define the thresholding areas. Thus, if the angle of

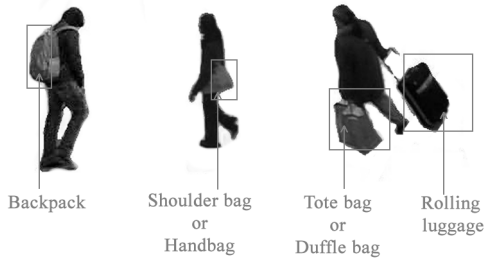
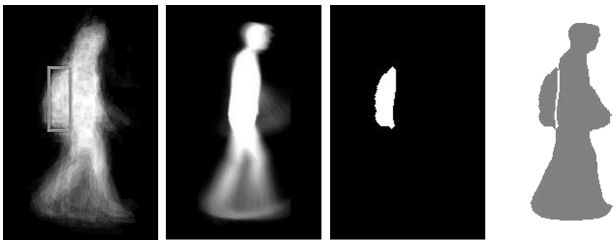


Figure 5.8: Bags type examples in BTR module.



(a) (b) (c) (d)

Figure 5.9: BTR involves: (a) the position of the detected bag, (b) the best matched exemplar, (c) the Graph Cuts bags segmentation and (d) the intersection of the exemplar with the bags.

the determined motion vector falls in one of these areas then possible directions are the ones indicated within each area in Figure 5.7. The output of the initial classifier that classifies into 3 basic poses is used to select one of the possible directions in each group. For example if the angle of the motion vector is 85° then the possible directions are 4 and 5. Subsequently, if the output of the classifier is 1 then the final viewing direction will be 5. In case the output of classifier does not match any of the available selections then the most likely direction is selected; in this example 5.

5.2.2 Carried Bag Type Recognition

Carried BTR algorithm is developed to further extend the usability of the COD system. The algorithm classifies the detected baggage into 5 categories subject to the position of the bag relative to the human body. Therefore it is important to examine the proportions of the human body. The 5 categories include *backpack, rolling luggage, tote bag or duffel bag, handbag* and *other* a category used to represent anything else which does not belong to the above four types named (see Figure 5.8). The result of this part is highly dependent on the baggage detection output as the position of the detected bag will indicate its type. Any inaccuracies of the baggage detectors output will have an impact on the BTR results. For this reason all algorithmic improvements previously made to the system, make the bag

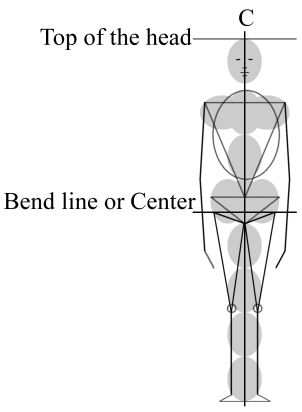


Figure 5.10: Human body proportions model. Bend line is the horizontal line through the vertical centre of the body.

position detection more accurate and render the system more suitable for the BTR module.

As it was mentioned before the type of the bag is decided by the position of the bag in relevance to the human body. Due to the fact that for baggage detection the temporal template is compared against the exemplars, the same exemplars are used for bag type recognition. Figure 5.9 (a) shows an example output of baggage detector and (b) the best matched exemplar. The Graph Cuts output (c) depicts the shape and the position of the bag. By relating the bag to the exemplar as in (d) it becomes possible for the system to recognise the bag type. The BTR algorithm involves the steps described below.

Human body parameter estimation: By examining the human body proportions it can be deduced that in average the height of a person is $H = 8h + e$, where h is the length of head and e the length of neck. In the proposed method e will be ignored as it is of minor importance. Therefore $h = H/8$. Figure 5.10 shows the proportions of the human body. Bend line B is the centre of the body and will be used as landmark for future estimations. If T is the position of the top of the head in the image, then $B = 4h + T$. The last element used is the vertical line C that traverses the centroid of the body.

Data pre-processing: The first step is the pre-processing of the best matched exemplar to extract useful information such as the height and the bending line. Second step is to merge the exemplar with the detected bags (Figure 5.9 (d)) to find if there is any intersection between them. If not then the bag in the later steps will be considered as a rolling luggage. Thirdly, the bounding boxes of the bags are obtained to record the position of the bags and their length over x and y axes in order to decide if they belong to any of the bag categories defined earlier. It is also examined if the bag is positioned in front of the human silhouette or behind. For this reason the viewing direction is taken into account. The rest of

```
BagTypeRecognition(SelectedTemplate, LabeledBags, C, direction, bag_x)
    // bag_x is center of bag on x axis
    // C is vertical line C through centroid
    // direction is viewing direction of the person
 1  if direction = 1, 2, or 8
 2      if bag_x < C
 3          apply restrictions to get one of the bag types:
            backpack, handbag, tote bag, rolling luggage, other
 4      else
 5          apply restrictions to get one of the bag types:
            handbag, tote bag,other
 6  elseif direction = 3, or 7
 7      if bag_x < C
 8          apply restrictions to get one of the bag types:
            backpack, handbag, tote bag, rolling luggage, other
 9      else
10          apply restrictions to get one of the bag types:
            backpack, handbag, tote bag, rolling luggage, other
11  elseif direction = 4, 5, or 7
12      if bag_x > C
13          apply restrictions to get one of the bag types:
            backpack, handbag, tote bag, rolling luggage, other
14      else
15          apply restrictions to get one of the bag types:
            handbag, tote bag, other
```

Figure 5.11: Bag type classification algorithm.

the algorithm is explained in the section that follows.

Bag classification: The general concept places the bag in one of the three major categories (depending on the viewing direction of the person) and then further classifies it to one of the two subcategories (depending on the position of the centre of the bag with respect to C). Afterwards the bag is classified into one of the 5 categories according to the constraints satisfied by its position. To begin with, it is very likely that some of the regions detected as bags are not actually bags. Figure 5.12 shows such regions. These are the ones that stretch over the line $T + 3(H/16)$ and under the line $T + 9(H/16)$ from both sides of the silhouette. It should be noticed that there is no constraint on the width of these regions. Once we dispose of the unwanted regions we can continue with the classification of the remaining reliable ones. The pseudocode of the algorithm is presented in Figure 5.11 and the diagrams in Figure 5.13 and Figure 5.14 specify the constraints to be applied in order to achieve the desired classification.

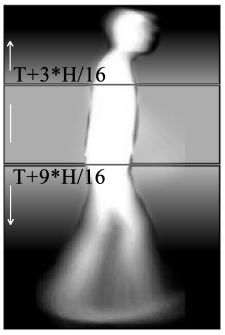
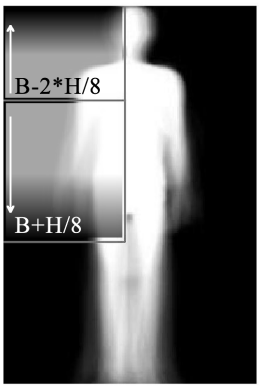
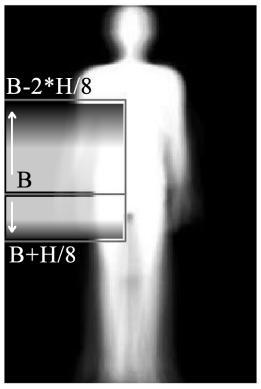


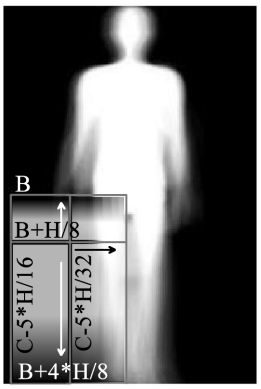
Figure 5.12: Regions that do not belong to any type of bag.



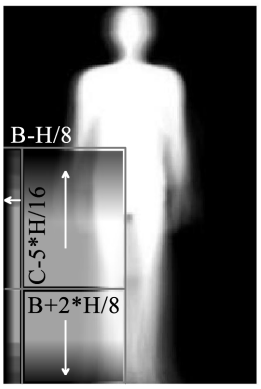
(a) Backpack

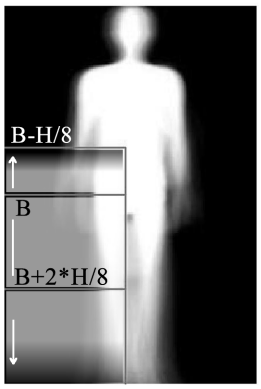


(b)Handbag



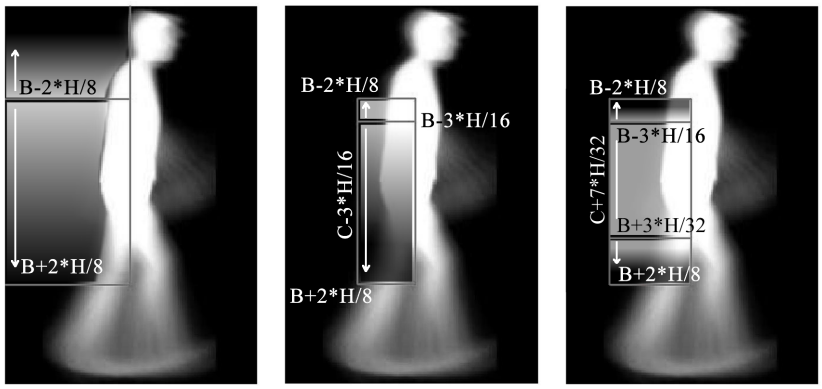
(c)Tote bag



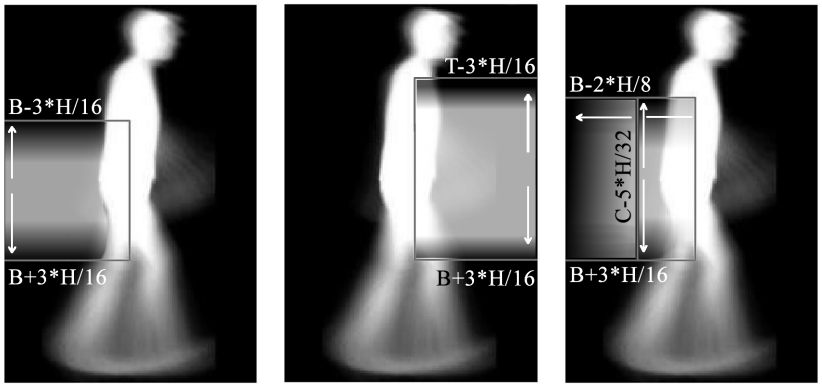


(d) Rolling Luggage

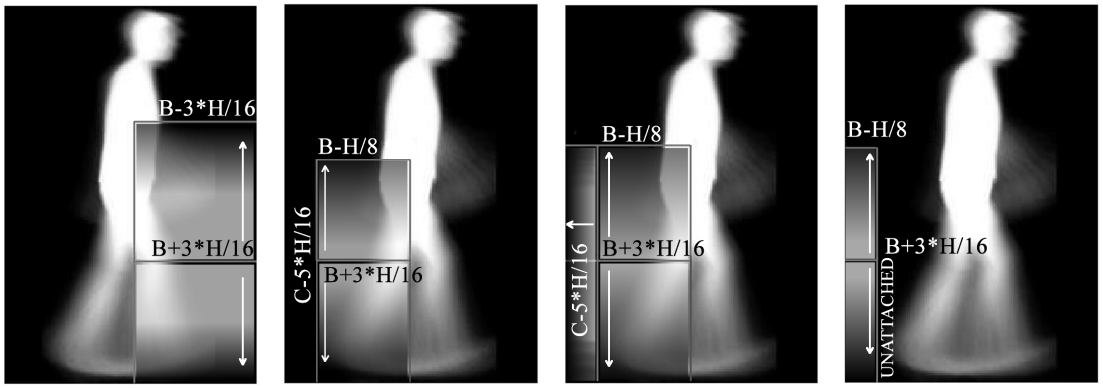
Figure 5.13: The groups of models (a)-(d) reflect the set of conditions that should be satisfied for the identification of a bag for directions of motion 3 and 7. For instance, if the dimensions of the bounding box of a bag are compliant with the restrictions depicted in (a) then the bag is classified as a backpack; if not, then it is attempted to place the bag in one of the other categories. The bag is classified as an unknown object if it fails to be placed in any of the above categories.



(a) Backpack



(b) Handbag



(c) Tote bag (d) Rolling luggage

Figure 5.14: The groups of models (a)-(d) reflect the set of conditions that should be satisfied for the identification of a bag for directions of motion 1, 2 and 8. For instance, if the dimensions of the bounding box of a bag are compliant with one of the three restrictions depicted in (a) then the bag is classified as a backpack; if not, then it is attempted to place the bag in one of the other categories. The bag is classified as an unknown object if it fails to be placed in all of the above categories.

5.3 Experimental Results and Analysis

The data used in the experiments were obtained from the third camera view of the PETS 2006 dataset, the videos AVSS_AB_easy and AVSS_AB_medium from the i-Lids 2007 dataset, and further in-house videos recorded by a commercial grade camera. Each person's trajectory has been split in such a way that it records 2 seconds of movement (approximately 2 walking cycles) as suggested by D. Damen. Therefore, the final sample size is 239 (or 179 not split) individuals for the PETS dataset, 75 (or 46 not split) for the i-Lids dataset, and 75 (or 39 not split) for the in-house videos. In spite of the fact that the last two datasets cannot be employed for the comparison of the viewing direction estimation because of the lack of calibration measurements, they are used for testing the viewing direction estimation and bag type recognition. The code of Damen & Hogg was used for the experiments and further development of the ideas as it is freely available for research purposes. The method intrinsically does not deal with partially occluded silhouettes; therefore, their trajectories have been removed manually. All experiments have been conducted on a computer with a 2.53 GHz processor and 4.00 GB memory.

5.3.1 Exhaustive Search Optimisation

To prove the efficiency of the new algorithm for acquiring the best matching exemplar from the database it was necessary to measure the execution time for different cases. Execution time was measured for a sample of 11 objects for each of the two search strategies and the average execution time was 17.07s for the primary method and 4.30s for the optimised one, i.e. the latter being 3.97 times faster. As the execution time for this part of the system does not depend on the number of frames per person, there was no necessity to make more measurements. Furthermore, by explicitly defining the approximate template size, the selection of exemplars much bigger than the temporal template is prevented, which is a regular phenomenon due to the fact that the goal is to minimise the difference.

5.3.2 Connected Bags Separation

The process of connected bag separation involves only large in size detected objects. This means that in templates of smaller sizes even if they include relatively big carried objects the separation will not be attempted, for the simple reason that the range of intensities will be very limited. Some instances of successfully separated bags are presented in Figure 5.15. The last example demonstrates the usability of the algorithm for more accurate isolation of the bag.

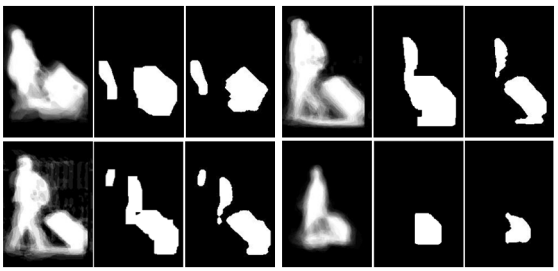


Figure 5.15: Examples of separated bags. The separation by thresholding improves the shape of the bags as well.

Table 5.1: Body orientation estimation without motion information. Classification results for the 3 basic classes.

	PETS	i-Lids	In-house
3 basic classes	85%	72%	78%

5.3.3 Evaluating the Viewing Direction Estimation

To compare the homography based direction estimation method with the proposed, the videos have been selected from the PETS dataset. Since most of the cases from the third camera view of the PETS dataset resemble viewing directions 2, 4, 6 and 8, the rest of the videos, which feature the directions 1,3,5 and 7, are used as complementary to prove the ability of the system to classify successfully. Table 5.1 presents accuracy results for the first classification stage where there are 3 classes and no motion information. Table 5.2 presents the results after including the motion vector of the person.

5.3.4 Bag Type Recognition

For BTR the experiments have been conducted on only the true positive objects recognized by the improved system. As summarised in Table 5.3 the overall accuracy for BTR is 75%. The accuracy figures obtained for the in-house videos and the i-Lids dataset are higher due to the fact that the moving people are closer to the camera and the viewing angle is optimised for baggage detection. The confu-

Table 5.2: Direction estimation comparison of the proposed method (including motion vector) with the D. Damen's method and demonstration of results for the other datasets.

	PETS	i-Lids	In-house
Homography based method	83%		
Proposed shape based method	85%	80%	82%

Table 5.3: BTR results for the PETS 2006 dataset, captured videos and the i-Lids dataset.

	PETS	Captured videos & i-Lids	Total
Accuracy	63%	92%	75%

Table 5.4: Confusion matrix for bag types recognised in all datasets.

		Predicted class				
		backpack	rolling luggage	tote bag	hand bag	other
Actual class	backpack	18	0	0	8	0
	rolling luggage	0	44	4	0	8
	tote bag	3	1	45	10	3
	hand bag	5	0	1	19	5
	other	0	0	0	0	1

sion matrix in Table 5.4 demonstrates the ability of the system to classify into five different classes.

5.3.5 Overall Performance Evaluation

After applying all of the proposed improvements it is important to examine their collective contribution to the overall performance of the system.

The graph in Figure 5.16 compares the execution time of the original system and the improved system against the number of people in the video sequence. At this point it should be mentioned that the number of frames that a person appears in affects the execution time but not significantly. Therefore, it has been ignored. The graph reveals the importance of computation time reduction for a large amount of processed data.

To test the performance of the system, Accuracy, Precision, Recall, and Specificity metrics have been employed. The ground truth box for the position of the bags on the temporal templates was obtained manually. Detection is considered as successful only if the overlap between the bounding box of the ground truth and the detected one is higher than 20%. In case the overlap is less than 20% but greater than 0% then, if the detected bounding box is inside that of the ground truth, then it is labelled as false negative, else if the ground truth bounding box is inside the bounding box detected, then it is labelled as being false positive. Any other case would suggest that the detected and the ground truth boxes are not related to each other and therefore the detections are labelled as false positive and

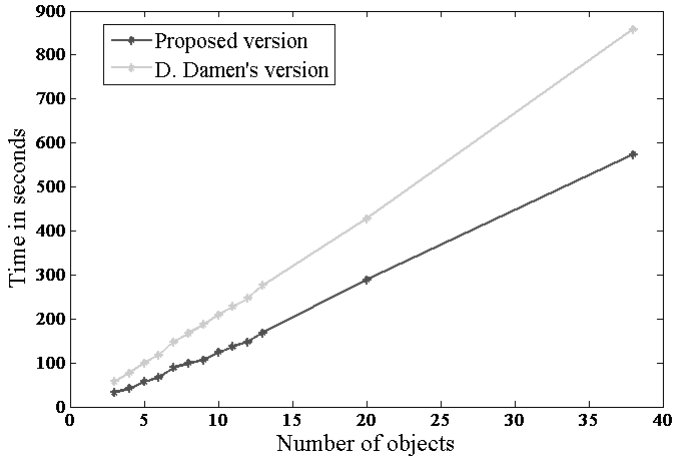


Figure 5.16: A performance comparison between the improved and the original versions.

Table 5.5: Overall results for the improved and primary system for 179 individuals from the PETS dataset.

	Accuracy	Precision	Recall	Specificity
D. Damen's version	50.3%	54.4%	55.0%	44.7%
Proposed version	56.9%	63.5%	56.9%	56.8%

false negative respectively.

In Table 5.5 the specificity and precision values of the primary system are low which shows the weakness of the system in recognizing the negative cases. However, the recall value is relatively high. Because of the fact that the improved system reduces the detection of false positives preserving the number of true positives, the precision and accuracy levels have been significantly increased. In addition the true negative detection is improved which is reflected in the new specificity obtained.

5.4 Summary and Discussion

In this chapter, a number of methods have been proposed to improve the baggage detection system originally proposed by D. Damen and D. Hogg. Two substantial extensions to the original system proposed by this research are body orientation estimation and baggage type recognition. The former will be further enhanced in the next chapter with the application of ordinary classifiers. It has also been proven that the proposed viewing direction estimation approach can successfully substitute the homography based direction estimation, discharging the system from the need of human intervention. Furthermore, due to the proposed enhancements for

the best matching model search, computational time has been significantly reduced allowing the potential for processing a large amount of data. Finally, it was shown that the proposed connected bag separation algorithm resulted in more accurate bag type classification.

Future work can focus on the enrichment of the temporal templates and usage of colour information for COD. Further, the current bag classification is based only on the location of the bag and other features like size and shape are not taken into account. Another perspective that has not been explored is the use of gradient information or shape context histograms for bag shape recognition, which can distinguish it from the body parts. In addition, frame wise colour silhouette segmentation can separate the areas belonging to clothes from the ones that belong to the carried objects. For more accurate temporal template construction the orientation estimation algorithm could be applied per frame basis, splitting the trajectory when the orientation changes. Some of the suggestions for future work will be presented in the next chapter.

Chapter 6

Carried Object Detection Using Colour Information

The preliminary information and concepts required to understand the major contribution of the work presented in the current chapter have been presented in chapters 3 and 5. The COD system proposed in this chapter is based on the COD system analysed earlier in section 3.1 with the difference that the intensity based temporal template used originally has been replaced with a colour temporal template (section 6.2). The work presented includes all approaches that are proposed with the aim of exploiting colour information for a variety of purposes within COD; for e.g., colour information is used to detect the human torso (section 6.4) as well as the carried object (subsection 6.5.1). Another key contribution of the work presented in this chapter is the redefinition of the viewing direction estimation algorithm by using the concept of machine learning (section 6.3). A further minor contribution of the chapter is the selection of the best matching human like exemplar, which is subtracted from the temporal template to deduce the protruding regions. Since the conventional temporal template has been substituted with the colour based temporal template it was important to reconsider the definition of the energy function that facilitates the segmentation of bags via the use of an energy minimisation approach based on Graph Cuts (section 6.5). The primary aim is to increase the accuracy of the system but at the same time to reduce the false positive detections which were common for the previous system. The detailed experimental results reflect the effectiveness of the suggested improvements in terms of accuracy, performance, and segmentation precision of the shape of the bags (section 6.7). Finally, as the proposed system is not 100% accurate, ideas for further research are proposed in section 6.8.

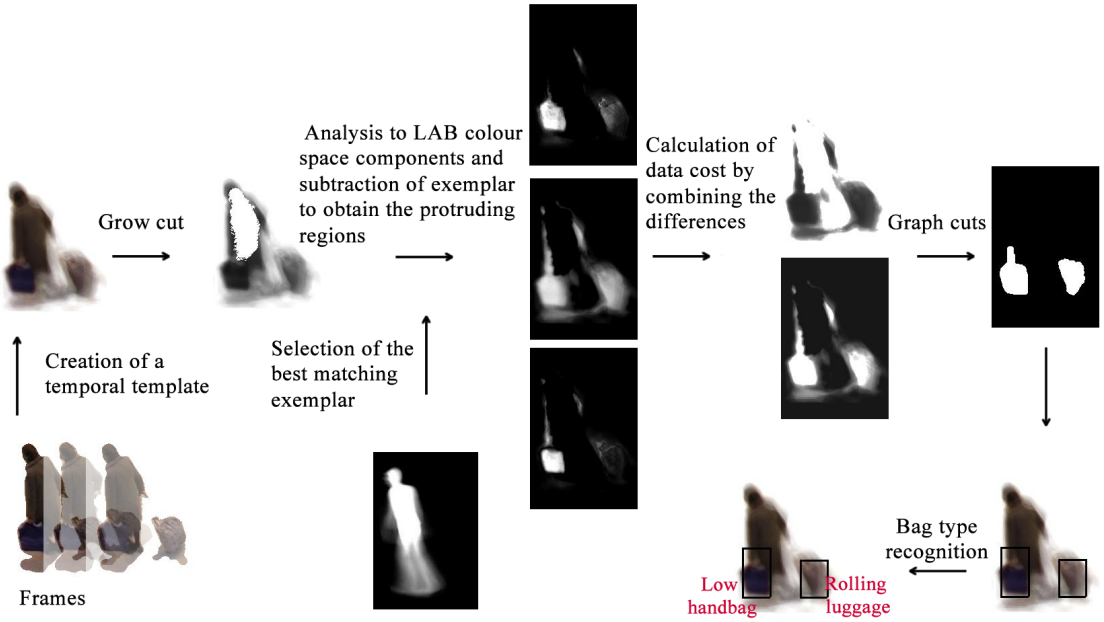
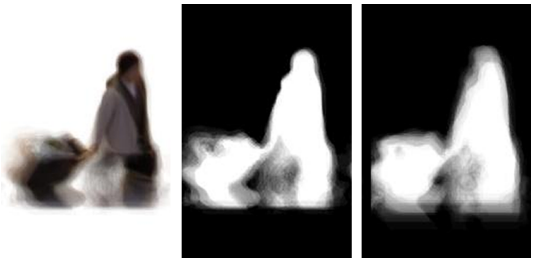


Figure 6.1: A summary of the baggage detection system.

6.1 System Overview

The work of D. Damen and D. Hogg is based on the creation of a temporal template of a moving person and further analysis of its shape to detect the carried object. The concept of temporal template was firstly defined by I. Haritaoglu in [53] as the average of foreground silhouettes. Since it is intended to use colour information, it was required to create a colour temporal template. Therefore, the baggage detector takes as input a sequence of colour foreground images which are aligned using image registration to create a colour temporal template (see Figure 6.1). Then, the temporal template is matched against exemplar temporal templates of 13 different sizes, 8 viewing directions, and 7 rotations until the best match is found. Subsequently the temporal template is decomposed into CIELAB colour space components and the best matching exemplar is subtracted from each of them to reveal the protruding regions which are likely to be carried objects. For further enhancement of the accuracy, D. Damen adopts trained view specific models that map the probable bag location and are used to weight the protruding regions. However, the effectiveness of these priors is doubted and, for the proposed application a more conservative prior of homogeneous nature is chosen to weigh the temporal templates.

It is common sense that the location of the bag and its visibility depends on the position of the viewer in relevance to the position, and moving direction of the pedestrian. This fact brings the need of classifying the temporal template



(a) (b) (c)

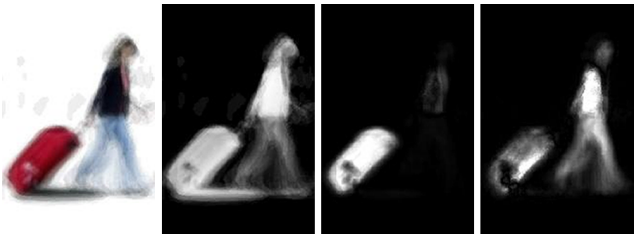
Figure 6.2: (a) is the colour temporal template generated using subpixel image registration, (b) is the corresponding frequency temporal template and (c) is the template generated using ICP.

into 8 categories, according to the direction of motion. One way to estimate the Viewing Direction (VD) is to transfer the motion from the image plane (the two-dimensional image surface) to the ground plane (which is at 90 degrees to the picture plane; commonly the ground that objects move on) via the homography transformation as proposed D. Damen in [33]. The problem here is that the homography transformation demands at least 8 corresponding points between the two planes. When the points are provided manually to the system, it becomes impractical while trying to automatically process videos from different cameras or the same pan-tilt-zoom (PTZ) camera. Certainly, automatic methods for homography calculation such as presented in [87] can be used but they are complicated and computationally expensive. The VD estimation method proposed in chapter 5 is extended here to be used with an SVM classifier.

6.2 Generation of the Temporal Template

The first step is the generation of the colour temporal template. The foreground silhouettes are segmented by a GMM background modelling algorithm as proposed by P. KaewTraKulPong in [68]. Then, a simple connected components tracking algorithm is applied to define the trajectories of the pedestrians. Since all the test videos involve only pedestrians, no human detection algorithm was employed.

The extracted colour foreground allows the utilisation of registration methods based on colour information. The report of R. Szeliski [132] suggests that direct pixel based alignment could be a solution. To reduce the computational load, Fourier transform is applied to the non-registered images to achieve reliable and fast alignment. Such an algorithm is proposed in [50] as subpixel image registration and its MATLAB source code is freely available for research purposes. This algorithm requires a small pixel sample to perform the registration and is accurate and fast enough to align a long sequence of images. The shoulder area has been



(a) (b) (c) (d)

Figure 6.3: (a) is the colour temporal template, (b) is the inverse grayscale temporal template, (c) is the **a*** component and (d) is the **b*** component.

selected as the pixel sample, since it remains stable over time unlike the limbs. The output of the algorithm is a translation vector and an error rate. If the error rate is larger than 29% or if the translation vector suggests displacement greater than 18 pixels then the current image is not registered to the preceding one to avoid large inconsistencies in the temporal template. Initially, all the silhouettes of a trajectory are centred and cropped to an image of fixed size $h \times w$ and then aligned by two consecutive frames at a time, via subpixel registration. As Colour Temporal Template (CTT) is defined the average of all silhouettes in each RGB channel using the formula in Equation 2.1. To differentiate the colour temporal template from the conventional one, which is defined as the average of binary silhouettes [33], henceforth it will be called as Frequency Temporal Template (FTT). Figure 6.2 shows an example comparing subpixel image registration with the ICP image alignment method employed in [33]. To create the corresponding FTT to the CTT, the calculated translation vector was simply applied to the binary silhouettes.

By having a colour temporal template it is important to decide upon which colour space properties best represent reality. It was decided to utilise the CIELAB colour space in [118] because it approximates human vision and therefore has an inherent property of segmentation based on colour (see an example in Figure 6.3). Because of the fact that the **a*** and **b*** derivatives of the L**a*b*** image might have negative pixel values the negative values have been transferred to the positive axis and the pixel values have been linearly adjusted. The purpose of the creation of these images will be analysed subsequently in this chapter.

6.3 Viewing Direction estimation

For the viewing direction estimation, the motion information as shown in subsection 5.2.1 was combined with the pose estimation based on shoulder shape features. The goal is to categorise the temporal template into one of the VD categories in Figure 6.4.

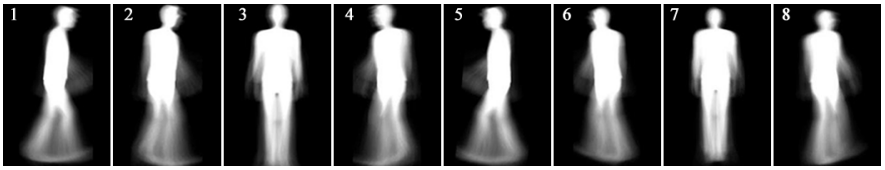


Figure 6.4: Exemplar temporal templates of 8 viewing directions.

The temporal template is thresholded as previously, in chapter 5, with Otsu's method to obtain the binary silhouette and the following features are calculated as in subsection 5.2.1 to feed an SVM classifier.

1. $DistanceRatio$
2. $Head\_ShoulderRatio$
3. $Head\_ShoulderRangeRatio$
4. $ShoulderRangeRatio$
5. $ShoulderVerticalRange\_LengthRatio$
6. $Shoulder\_HeightRatio$
7. $RightHead\_Shoulderdistance$
8. $LeftHead\_Shoulderdistance$
9. $MeanDifference$
10. $Length\_ShoulderDifference$
11. $Angle\_LengthDifference$
12. $Range\_LengthDifference$

An SVM classifier is trained to classify into 3 target classes: class 1 (directions 1 and 5), class 2 (directions 2, 4, 6, and 8), class 3 (directions 3 and 7). After acquiring the basic information about the pose from the classifier the VD can be identified by combining its pose, and the angle of motion vector (unit vector) from a vertical unit vector (0,-1) on the image plane. The rest of the procedure followed is exactly as described in subsection 5.2.1, subsubsection "Final Viewing Direction Estimation by Combining the Body Orientation with the Motion Vector".

6.4 Selection of Best Matching Model and GrowCut Segmentation

To find the exemplar model that best matches the CTT under query, the correlation function is utilised. Initially the temporal template is roughly aligned to the exemplar model by the head location and its size is approximated by the height of the temporal template. To refine the alignment a correlation measure is maximised to confirm the best match.

The correlation function between matrices A and B of size $w \times h$ is defined as follows.

$$C(A,B)) = \frac{\sum_{(x,y)=(1,1)}^{(w,h)} \big(A(x,y) - \bar{A}\big)\big(B(x,y) - \bar{B}\big)}{\sqrt{\left(\sum_{(x,y)=(1,1)}^{(w,h)} (A(x,y) - \bar{A})^2\right)\left(\sum_{(x,y)=(1,1)}^{(w,h)} (B(x,y) - \bar{B})^2\right)}} \tag{6.1}$$

where $\bar{A}$ and $\bar{B}$ are the averages of the matrices A and B. Since the exemplar models EM are frequency templates (with intensities varying from 0 to 1) the CTT has to be adjusted accordingly. Therefore, it is converted to greyscale, inverted, and normalised to the range [0, 1]. The similarity measure function between the inverse temporal template PT and the exemplar model EM to be maximised is defined as

$$S(PT, EM) = 0.8C(PT_{tg}, EM) + 0.2C(PT_g, EM) \tag{6.2}$$

where,

$$PT_g(x,y) = PT(x,y)g(y) \tag{6.3}$$

and

$$PT_{tg}(x,y) = \begin{cases} PT(x,y)g'(y), & if\ EM(x,y) > 0 \\ 0, & if\ EM(x,y) = 0 \end{cases} \tag{6.4}$$

where, $g(y) = [4.40 : -0.01 : 2.01]$ and $g'(y) = [3.20 : -0.005 : 2.005]$. EM_{best} is selected as the best matching transformed exemplar that maximises $S(PT, EM)$.

$$arg\max_{EM} S(PT, EM) = \{EM_{best}\} \tag{6.5}$$

For further enhancement of accuracy the selected exemplar template is dilated by a square structuring element $SE_{ij}, i, j = \{1, 2\}$

$$EM_{best} = \begin{cases} EM_{best} \bigoplus SE, & if\ for\ Diff(x,y)\ Shoulder\_HeightRatio > 0.1 \\ EM_{best}, & otherwhise \end{cases} \tag{6.6}$$

where, $Diff(x,y) = |PT(x,y) - EM_{best}(x,y)|$ and '$\bigoplus$' is the dilation operator. The threshold for dilation implementation is selected after testing through some selected datasets.

Sometimes the shape of the best matching exemplar is not enough to cover the torso area. Hence, GrowCut algorithm [138] is employed to define the torso shape. In more detail, the Grow cut algorithm takes as input two images: the seeds and the image on which the algorithm is applied. The seeds are the labels for the foreground, background and the unknown regions. The seed image will be

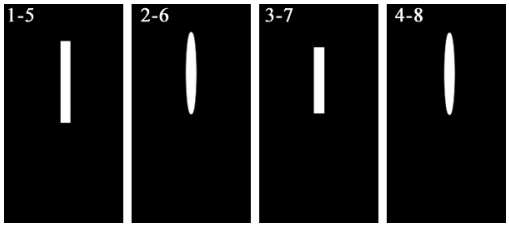


Figure 6.5: Direction specific mask images used as input seeds to GrowCut algorithm.

(a) (b) (c) (d) (e) (f)

Figure 6.6: GrowCut segmentation: (a) is the temporal template, (b) is the selected best matching exemplar, (c) is the transformed (size and position) mask, (d) is the GrowCut segmentation results, (e) is the difference image and (f) is the labelled bags.

called as mask M and it will be labelled as follows:

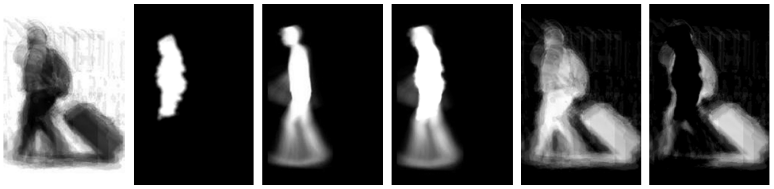
$$M(x,y) = \begin{cases} 1, & if\ foreground \\ -1, & if\ background \\ 0, & if\ uncertain \end{cases} \tag{6.7}$$

The foreground is a predefined pixel selection as shown in Figure 6.5 which is transformed according to transformations applied to the best matching exemplar to match the size and rotation of the temporal template. The background comprises of all the pixel values greater than 200 (selected as threshold after testing) on the CTT, and unknowns are all the rest. The **a*** component of L**a*b*** image representation is selected as the second input argument. Thus the result of segmentation are the labels indicating the foreground region; here the torso (see Figure 6.6).

$$labels = GrowCut(\mathbf{a}^*, M) \tag{6.8}$$

As mentioned above, the torso segmented via GrowCut is used to redefine the exemplar itself. Therefore, if the blurred GrowCut foreground is B (Figure 6.7 (b)) then the final best matched exemplar template is defined as follows (Figure 6.7 (d)):

$$EM_F(x,y) = \min(1, B(x,y) + EM_{best}(x,y)) \tag{6.9}$$



(a) (b) (c) (d) (e) (f)

Figure 6.7: Exemplar redefinition with the segmented torso. (a) is the temporal template, (b) is the blurred GrowCut result, (c) is the best matching exemplar, (d) is the combination of exemplar with the segmented torso, (e) is the inverse grayscale temporal template and (f) the difference image $v(x, y)$.

The difference image $v(x, y)$ in Equation 6.10 reveals the protruding regions that will undergo further analysis for carried objects extraction (Figure 6.7 (f)). The corresponding differences for the **a*** and **b*** components of the L**a*b*** colour space are presented in Equation 6.11 and Equation 6.12.

$$v(x, y) = \max(0, PT(x, y) - EM_F(x, y)) \tag{6.10}$$

$$v_a(x, y) = \max(0, PT(x, y) - \mathbf{a}^*(x, y)) \tag{6.11}$$

$$v_b(x, y) = \max(0, PT(x, y) - \mathbf{b}^*(x, y)) \tag{6.12}$$

6.5 Segmentation by Energy Function Minimisation

The segmentation of bags is handled as a labelling problem by D. Damen in [33] where the segmentation of carried objects is achieved via energy minimisation using the Graph Cuts algorithm [14] (see Appendix C). Therefore the pixels that belong to carried objects should be labelled as foreground and limbs as background. Assuming that the label assigned to a certain pixel $p_i(x, y)$ depends only on the labels assigned to its neighbouring pixels, the difference image $v(x, y)$ can be considered as a first-order MRF. Since a 4-neighbourhood system on a 2D lattice is assumed and the label set is $L = \{0, 1\}$, the Gibbs energy function follows a special case of the Ising model [86]. If P is an image (lattice) $m \times n$ and $N \subset P$ is the neighbourhood system for P defined as $N = \{N_p | \forall p \in P\}$, then according to Bayes' theorem, to achieve segmentation, a maximum a posteriori (MAP) solution should be given (i.e. maximising the posterior probability). This is equivalent to

minimising the following energy function

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \tag{6.13}$$

Where D_p is the data cost function, and $V_{p,q}$ is the smoothness cost or clique potential function that defines the interaction between the neighbouring pixels. Supposed $f = \{f_1, ..., f_{m \times n}\}$ is a set of labelling configurations defined on the lattice P such that $f_p = f(p), \forall p \in P$, then f_p assigns a label $l \in L$ to the pixel $p \in P$, [86],[71]. The data cost function D_p can be an arbitrary probability density function (p.d.f) that measures the cost of assigning the configuration f_p to a pixel. Therefore, we will search for the p.d.f that best describes the given data. In our case the smoothness cost function is defined as

$$V_{p,q}(f_p, f_q) = \begin{cases} s, \ if \ f_p \neq f_q \\ 0, \ if \ f_p = f_q \end{cases} \tag{6.14}$$

and the data cost function is defined as

$$D_p(f_p) = -\ln\left(p_1(v(x,y)|f_p) + p_2(v(x,y)|f_p)\right) \tag{6.15}$$

Hence, the class conditional distributions are:

$$\begin{cases} p_1(v|f_p = 1) = \frac{1}{\kappa}\mathcal{M}(x,y) * \hat{W}(y) \\ \\ p_1(v|f_p = 0) = \kappa \ln \mathcal{N}(x,y) * (1 - \hat{W}(y)) \end{cases} \tag{6.16}$$

$$\begin{cases} p_2(v|f_p = 1) = (\lambda v_a(x,y) + \lambda v_b(x,y)) * \hat{W}^3(y) \\ \\ p_2(v|f_p = 0) = -(\lambda v_a(x,y) + \lambda v_b(x,y)) * \hat{W}^3(y) \end{cases} \tag{6.17}$$

As p.d.f are selected the mixture $\mathcal{M}$ of normal distributions and log-normal ($\ln\mathcal{N}$) distribution, multiplied by constants $1/\kappa$ and κ respectively. The coefficient λ that is used to weigh the differences $v_a(x,y)$ and $v_b(x,y)$ is found to be proportional to κ such that $\lambda = 5/\kappa$. In this manner there will be only one coefficient that will determine the sensitivity of the system. Finally, $\hat{W}$ is a Gaussian weight function and its purpose will be clarified in subsection 6.5.2.

Considering the weight function $\hat{W}$ as a prior it is possible to apply Bayes rule to compute the posterior probability. According to Bayes theorem for data analysis [127] the posterior probability (represents our state of knowledge about the truth of the hypothesis in the light of data) of hypothesis H given data D

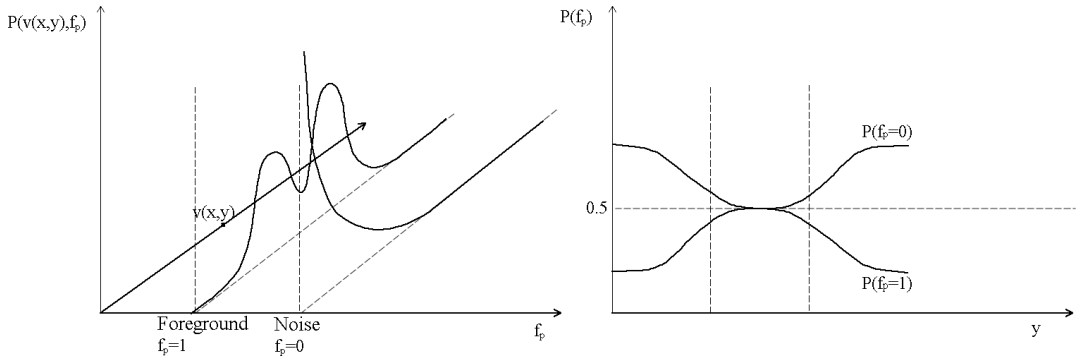


Figure 6.8: Class conditional likelihood distributions (left) and the prior distribution (right).

equals to

$$P(H|D) = \frac{P(D,H)}{P(D)} = \frac{P(H)P(D|H)}{P(D)} \tag{6.18}$$

where $P(H)$ is the prior distribution and $P(D|H)$ is the class conditional likelihood distribution of data given the hypothesis. The normalizing factor $P(D) = \Sigma_H P(H)P(D|H)$ is the sum over all possible values of H [2].

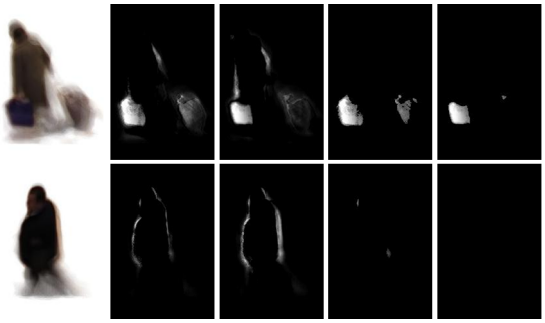
Here, the prior distribution is assumed to be the weight function $\hat{W}$ which gives us a general true understanding at which areas of the template it is more likely to find a bag. The class conditional prior $P(D|H)$ is as defined in Equation 6.16 (see Figure 6.8 left). Therefore the posterior distribution is defined as follows:

$$\begin{aligned} P(f_p = 1|v(x,y)) &= \frac{P(f_p = 1, v(x,y))}{P(X)} \\ &= \frac{P(f_p = 1)P(v(x,y)|f_p = 1)}{P(f_p = 1)P(v(x,y)|f_p = 1) + P(f_p = 0)P(v(x,y)|f_p = 0)} \end{aligned} \tag{6.19}$$

$$\begin{aligned} P(f_p = |v(x,y)) &= \frac{P(f_p = 0, v(x,y))}{P(X)} \\ &= \frac{P(f_p = 0)P(v(x,y)|f_p = 0)}{P(f_p = 1)P(v(x,y)|f_p = 1) + P(f_p = 0)P(v(x,y)|f_p = 0)} \end{aligned} \tag{6.20}$$

where $P(f_p = 1) = \hat{W}$ and $P(f_p = 0) = 1 - \hat{W}$ (see Figure 6.8 right). The found posterior is regarded as prior and is forwarded to Equation 6.16 which becomes

$$p_1(v(x,y)|f_p) = \begin{cases} P(f_p = 1|v(x,y)) & if \quad f_p = 1 \\ \\ P(f_p = 0|v(x,y)) & if \quad f_p = 0 \end{cases} \tag{6.21}$$



(a) (b) (c) (d) (e)

Figure 6.9: Temporal template in (a) and the $v_a(x, y)$ and $v_b(x, y)$ difference images in (b) and (c) respectively. (d) and (e) are the $v_a(x, y)$ and $v_b(x, y)$ after noise reduction.

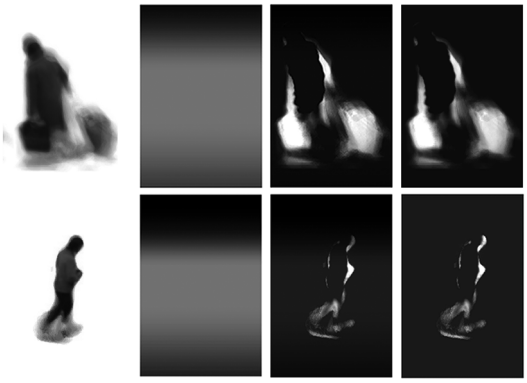
6.5.1 Utilisation of La*b* Colour Space Derivatives a* and b* as Difference Images

Due to the fact that the difference images $v_a(x, y)$ and $v_b(x, y)$ contain noise that occur because of the significant colour information present at the outline of the silhouette, the connected components that are unlikely to represent a bag have been removed by considering their aspect ratio, their size relative to the silhouette and the pixel intensity. Thus, the connected components that have been removed could be characterised as unusually long, small or lacking colour.

More specifically, the separation between the objects and non-object areas is executed in two phases: in the first phase the thresholds are applied directly to the binary $v_a(x, y)$ and $v_b(x, y)$ images and in the second phase the thresholds are applied to the binary $v_a(x, y)$ and $v_b(x, y)$ images after morphological erosion. The values that are thresholded for each object, in both phases, are: the ratio of the major and minor axes lengths, the area, the mean value of pixel intensities and an additional aspect ratio feature for the second phase. Subsequently, the results of both phases are combined into one image and its equivalent greyscale image is obtained from the original $v_a(x, y)$ and $v_b(x, y)$ images. The results of processing are shown in Figure 6.9.

6.5.2 Definition of the Weight Function $\hat{W}$

As it is expected some of the areas with high probability do not always belong to a carried object, especially the ones that are located around the head or well under the feet. The face colour contains high intensity values, due to which the potential objects extracted from the **a*** and **b*** images belong to the head region on a number of occasions. Therefore there is a need for a function that weighs the difference images $v(x, y)$, $v_a(x, y)$, and $v_b(x, y)$ to reduce the probability in



(a) (b) (c) (d)

Figure 6.10: The effect of weight function application: The temporal templates in (a) are wighed by their gradient like weights in (b). (c) and (d) are the difference images multiplied by the weight vector $\hat{W}$ and not respectively.

the unlikely areas. The weight function is constructed as a vector that will be multiplied with each column of the difference image. A linear function does not assist the intended aim since the transition across the curve is very steep. On the other hand the Gaussian function offers a smooth and slow transition and is evaluated as the most suitable due to its bell shape (Equation 6.22).

$$f(x) = a \exp\left(-\frac{(x-b)^2}{2c^2}\right) \tag{6.22}$$

Thus, for a given vector $x = [1, 2, ..., h]$, where h is the height of the temporal template image, a vector W of size $h \times 1$ is constructed. W contains all the weights in the following way: Let us assume two different vectors that obey two different Gaussian functions,

$$\begin{cases} (W_{1i})_{i=1}^{h} = [(d_i)_{i=1}^{\frac{h}{2}} \quad (c_i)_{i=\frac{h}{2}+1}^{h}] \\ (W_{2i})_{i=1}^{h} = [(c_i)_{i=1}^{\frac{h}{2}} \quad (d_i)_{i=\frac{h}{2}+1}^{h}] \end{cases} \tag{6.23}$$

where, $(c_i)_{i=1}^{\frac{h}{2}} = (c_i)_{i=\frac{h}{2}+1}^{h} = 0.5$ and

$$(d_i)_{i=1}^{\frac{h}{2}} = 0.5e^t, \qquad \text{where } t = -\frac{((x_i)_{i=1}^{\frac{h}{2}} - h/2)^2}{2(100(S_a - 0.2))^2}$$

$$(d_i)_{i=\frac{h}{2}+1}^{h} = 0.5e^t, \qquad \text{where } t = -\frac{((x_i)_{i=\frac{h}{2}+1}^{h} - h/2)^2}{2(100(S_a + 0.2))^2}$$

Because the size of the temporal templates differs it would be unfair to use the same weight vector for different sizes. Hence, the weight vector is adapted according to the size $S_a = \{0.2, 0.25, 0.3, 0.35, ..., 0.8\}$ of the best matched exemplar so that the higher probabilities are concentrated around the silhouette's torso. The

final weight vector $\hat{W}$ is

$$\begin{cases} (\hat{W}_i)_{i=1}^{\frac{h}{2}+10} = (W_{1k})_{k=i+50S_a} \\ \\ (\hat{W}_i)_{i=\frac{h}{2}+11}^{h} = (W_{2k})_{k=i-50S_a} \end{cases} \tag{6.24}$$

To clarify the concept Figure 6.10 demonstrates how exactly $\hat{W}$ looks like and the results of its multiplication with difference images.

6.5.3 Definition of the Probability Function $\mathcal{M}(x, y)$

As it was proposed by D. Damen the pixel values under a threshold constitute to noise and all the rest are likely to belong to a carried object. This threshold is decided using the following equation $thresh = \min(0.38,\ \max(v(x, y))/1.72)$. The numeric values that appear in this function are considered to be the safest ones after testing through a training sample. As an example, the distribution of pixels greater than the threshold is presented in Figure 6.11. Since the pixel values on the greyscale temporal template vary with the colour of the clothing and the carried object, the conditional distribution cannot be approximated by a single distribution and cannot be the same for all the cases. Thus, a non-parametric density estimation technique is used to obtain an intuition of how the pixel values are distributed and to determine the parameters of the Gaussian distribution or the mixture of Gaussian distributions that would form the likelihood function.

The averaged shift histogram is a common non parametric density estimation technique which defines the probability of a pixel p having a value x_t out of a sample of values $x_1, x_2, ...x_n$. Given the kernel function $K(t)$ the probability is defined in [122] as follows:

$$Pr(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{6.25}$$

where n is the sample size and h is the smoothing factor, also known as bandwidth. Built in MATLAB routine "ksdensity" was used to perform density estimation, having the normal distribution as kernel function $K(t)$ and h computed as suggested by Silverman in [126]. Hence,

$$h = \hat{\sigma} C_\nu(k) n^{-1/(2\nu+1)} \tag{6.26}$$

where $\hat{\sigma}$ is the sample's standard deviation, ν is the order of the kernel and $C_\nu(k)$ is the constant for the chosen distribution, which for a Gaussian kernel of size $\nu = 2$ is equal to 1.06. Therefore, Equation 6.26 can be simply written as $h = 1.06\hat{\sigma}n^{-1/5}$.

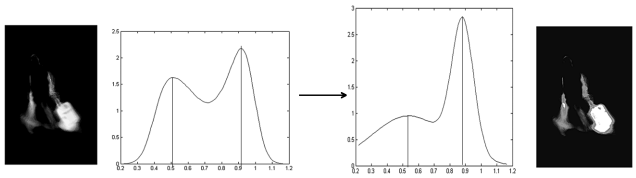


Figure 6.11: An example showing non parametric density curve with two peaks/two means (left). In the proposed technique the standard deviations are adjusted proportionally to the means and the resulting curve is shown in the right. The corresponding templates are on either side of the curves.

The above density estimation was applied to the difference image $v(x, y)$. The main purpose of this operation is to smooth out the intensity histogram to obtain the mean values of the mixture of Gaussian distributions. For example given the curve in Figure 6.11 we separate our sample into two groups as indicated by local minima and each subsample has its mean at local maxima. From the experiments it was found that in the case of a true detection, the standard deviations of the distributions are proportionate to the computed mean values. Hence, s is found to be

$$s = \frac{(\bar{x} - thresh)(0.01 - b)}{1 - thresh} + b \tag{6.27}$$

where $b = thresh - 0.09$, s belongs to the interval $(0.01, b)$ and $\bar{x}$ to $(thresh, 1)$. The above function establishes a correspondence between the mean and the standard deviation. The likelihood mixture $\mathcal{M}$ of Gaussians is computed in such a way that the sum of their coefficients k_i approximates 1. Therefore, if the number of found distributions is G_n , then

$$\mathcal{M} = \sum_{i=1}^{G_n} k_i \mathcal{N}(\bar{x}_i, s_i) \tag{6.28}$$

The coefficient values k_i are decided to be analogical to the highest probability for each mean value; this is analogical to the local maxima of the non-parametric density curve. Thus,

$$k_i = \frac{max_i}{\sum_{j=1}^{G_n} max_j}, \quad i = \{1...G_n\} \tag{6.29}$$

where, max_i are the local maxima.

6.5.4 Definition of the Probability Function $\ln \mathcal{N}(x, y)$

By observing the first histogram in Figure 6.12, which represents the distribution of the pixel values smaller than the defined threshold $thresh$, it is concluded that

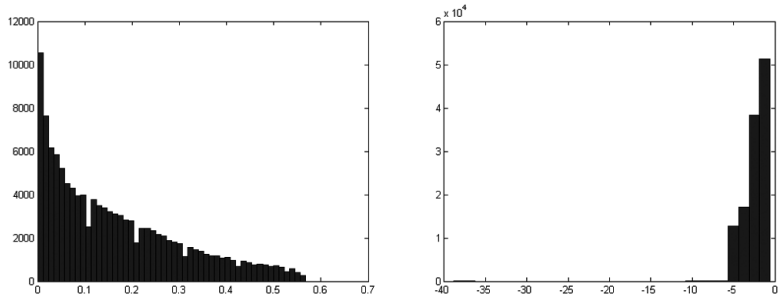


Figure 6.12: The noise distribution for pixel values < 5.8 (left) and the distribution of their log values (right).

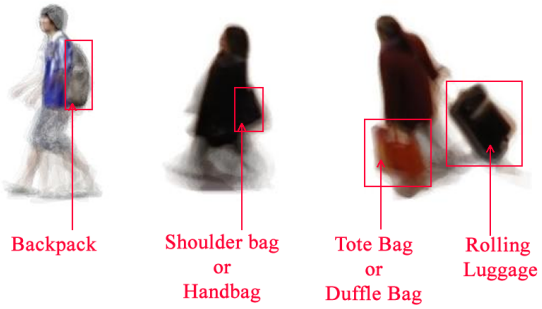


Figure 6.13: Bags type example in bag type recognition module.

they follow a log-normal distribution. To find the parameters of the log-normal distribution it is required to plot the log values of the pixels [115]. Since the second histogram in Figure 6.12 resembles a normal distribution, the mean value for the log-normal distribution is approximately $\mu = -0.7$ and the standard deviation is $\sigma = 2.2$ as obtained from the normal distribution histogram in Figure 6.12 (right).

6.6 Carried Bag Type Recognition

Bag type recognition algorithm is developed in the previous chapter is tested across the newly acquired bags. The algorithm classifies the detected baggage into 5 categories subject to the position of the bag relative to the human body. The 5 categories include *backpack, hand bag or shoulder bag, tote bag or duffel bag, rolling luggage*, and *other* to represent anything else which does not belong to the above (see Figure 6.13). The result of this part is highly dependent on the baggage detection output as the position of the detected bag will indicate its type. Any inaccuracies of the baggage detector's output will have an impact on the bag type recognition results. For this reason all algorithmic improvements previously made to the system, make the bag position detection more accurate and render the system more suitable for the bag type recognition module.

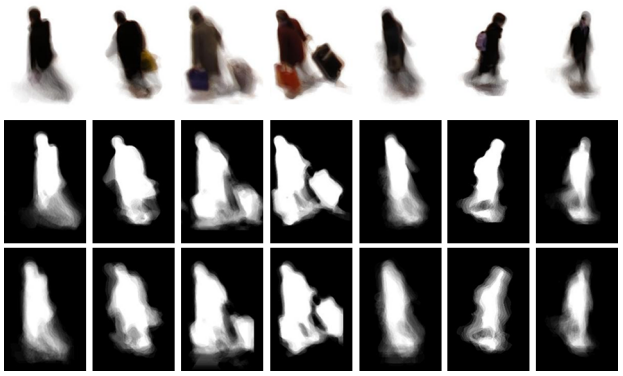


Figure 6.14: Comparison of image alignment techniques: (a) illustrates the CTT created by using subpixel image registration and (b) shows the corresponding FTT. (c) shows the FTT generated by the ICP algorithm.

6.7 Experimental Results and Analysis

In order to critically evaluate the performance of the improved system a significant number of detailed experiments were conducted. The data used for the experiments were obtained from the PETS 2006 dataset (3^{rd} camera view) and further videos recorded in-house using hand-held and CCTV cameras to cover specific test scenarios that were not present in the PETS dataset. Since the system is not capable of processing occluded silhouettes, these have been removed manually. The experiments have been conducted on a computer with a 2.53 GHz processor and 4.00 GB RAM memory. Each person's trajectory has been split in such a way that it records 2 seconds of movement (approximately 2 walking cycles) as suggested by D. Damen. Therefore, hereafter we will refer to each part of a trajectory as a separate individual and consequently the sample size will increase accordingly. The final sample size will be 239 (or 179 not split) individuals for the PETS dataset, 95 (or 39 not split) for the in-house videos obtained by the hand-held camera, and 94 (or 45 not split) for videos obtained via CCTV cameras. The implementation was carried out in MATLAB and constitutes a major revision of the initial implementation by D. Damen and Hogg. The following subsections present the experimental results for each step of the procedures described in this chapter.

6.7.1 Temporal Template Generation and La*b* Colour Space Exploitation

As it was explained in the relevant section, the temporal template is generated by averaging the extracted foreground silhouettes. To achieve this, all the images through the frames have to be aligned. Here, it is proven that the proposed image registration method offers a considerably better outcome than the ICP alignment.

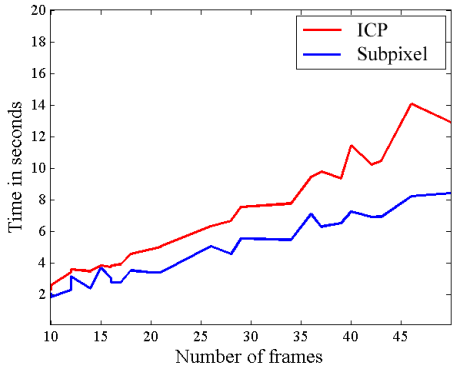
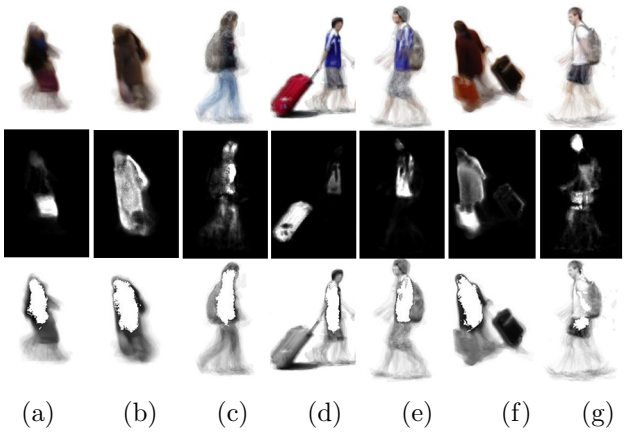


Figure 6.15: Processing time comparison of image alignment techniques.

The figures that follow, compare the ICP and Subpixel image registration methods (see Figure 6.14 and Figure 6.15). Undoubtedly, the subpixel registration has outperformed the ICP in terms of accuracy and time.

Another division of the proposed system that needs to be justified is the GrowCut exploitation and L**a*b*** colour space selection. The main reason for GrowCut application is the definition of human torso. We take advantage of the fact that the colour of clothes is uniform over the entire torso region and the bag colour differs from that of the clothes. In a sense, the high contrast attribute of **a*** and **b*** components of the L**a*b*** colour space is utilised. Figure 6.16 illustrates the results of applying GrowCut segmentation on the **a*** component. Apparently, the segmentation is not successful on white colour clothes due to the lack of colour information (Figure 6.16 (g)).

From the viewpoint of definition of bags the L**a*b*** colour space offers significant improvement to the system. Whereby, not only it enables the detection of



(a) (b) (c) (d) (e) (f) (g)

Figure 6.16: Human torso definition: The first row depicts the CTT, while the second row their **a*** component of the L**a*b*** colour space. The result of GrowCut algorithm as a mask over the temporal template is shown in the last row. The algorithm fails while expanding over colourless areas like in (g).

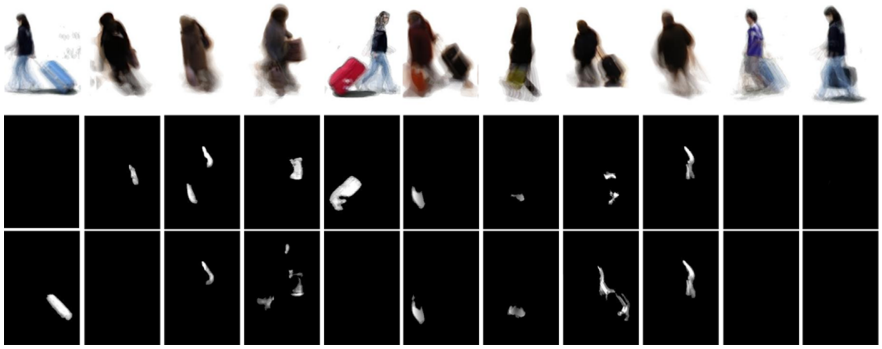


Figure 6.17: Segmentation of the carried objects by means of **a*** and **b*** derivatives of the CIELAB colour space.

otherwise undetected objects, but it also contributes to the accurate acquisition of bags shape by reducing the noise caused by shadow. Figure 6.17 shows some of the successfully and unsuccessfully isolated bags. By observing the presented examples it is possible to understand which colours are encoded by each of the **a*** and **b*** components and the reason that some bags are not segmented; i.e. mainly the weak colour representation of the bag including considerable amount of black and white colour.

6.7.2 Direction Estimation Evaluation

An SVM classifier has been trained to obtain the pose of the pedestrians. The LIBSVM library was chosen as a tool to implement the classification [21]. As a matter of fact, the classifier can recognise between three different poses. The motion vector of the pedestrian is used to refine the results and classify the object to one of the 8 categories. Since the performance of the classifier cannot be evaluated objectively over the whole dataset, 8 random images have been selected from each direction group to test the classifier; thus the testing set consists of 8*5=40 samples.

Initially, 12 images from all available datasets for each of the 5 groups shown in

(a) (b) (c) (d) (e)

Figure 6.18: Each of the shown templates is an example for each of the 5 categories with 3 different labels. Therefore for each image there are (a) direction=1, label=1, (b) direction=2, label=2, (c) direction=3, label=3, (d) direction=6, label=2, (e) direction=5, label=1.

Table 6.1: Direction estimation comparison of the proposed method with the D. Damen's method.

	Accuracy		
	PETS 2006	In-house Videos	CCTV
Proposed Method	0.90	0.89	0.81
D. Damen's Method	0.83		

Figure 6.18 were selected and labeled with 3 different labels, depending on their posture. Hence the training set consists of 12*5=60 real life samples. As the set of training images is not large enough for training and not all the features are rotation and scale invariant, the 60 temporal templates have been flipped vertically and rescaled to a range of $\{0.5, 0.6, \ldots 1\}$; in total obtaining 60*6*2=720 samples. A further set of 90 exemplar templates of selected sizes and rotations have been added as a supplement to the training set.

The classifier has been optimised by performing 6-fold cross-validation and 'grid-search' as suggested by C. Hsu et al. in [61] to select the kernel function and the training parameters. The set was separated in a way so that the rescaled versions of each image were at the same group; i.e. testing or training. The set of 90 exemplars was included in each training fold, since it always gives accuracy of 100%. Among linear, polynomial and Radial Basis kernel functions, the latter one was selected since it gives the highest average accuracy of 85.6% for the training parameters $C = 0.46$ and $\gamma = 0.004$. The final classification model was tested over the sample of 40 silhouettes and the accuracy achieved was 80%.

To compare the direction estimation method used by D.Damen with the proposed one, the 3rd camera view videos have been selected from the PETS dataset. The rest of the datasets in Table 6.1 simply demonstrate the capability of the system to identify other viewing directions as well (CCTV videos: mostly directions 3 and 7, videos captured using hand-held cameras: mostly directions 1 and 5,). Although the accuracy values are high, they do not approximate to 100%. This is due to the deformation of shoulder shape, either as a result of poor temporal template generation or objects carried on the shoulders.

A significant proportion of this research was devoted for direction estimation for the reason that it largely affects the final results. For example, if for actual viewing direction 1, the detected viewing direction is 2, then possibly a carried backpack will not be detected or vice versa, i.e., the shoulders might be detected as a backpack.

Table 6.2: Bag type recognition results for the PETS 2006 dataset and captured videos.

	True	False	Total	Accuracy
PETS 2006	98	45	143	68.53%
Hand-held camera	43	20	63	68.25%
CCTV	30	5	35	85.71%
Total	171	70	241	70.95%

Table 6.3: Confusion matrix for bag types recognised for all datasets.

		Predicted class					
		backpack	rolling luggage	tote bag	hand bag	other	total
Actual class	backpack	17	0	0	14	0	31
	rolling luggage	0	45	9	0	2	56
	tote bag	2	3	67	14	3	81
	hand bag	5	0	2	39	1	47
	other	0	0	0	8	1	9

6.7.3 Bag Type Recognition

For BTR only the true positives recognised by the improved system are taken into account. It is important to note that the recognition is based only on the location of the bag and no other features such as size and shape are taken into account. Another perspective that has not been explored yet is the usage of gradient information for bag shape recognition, which could likely distinguish it from body parts. As summarised in Table 6.2 the overall accuracy for bag type recognition is 70.95%.

The confusion matrix in Table 6.3 illustrates the ability of the system to distinguish between the four different classes, but also reveals the weakness to distinguish between backpacks and hand bags. It is good to mention that the category 'other' does not overflow.

6.7.4 Overall Performance Evaluation

After applying all of the proposed improvements it is important to examine their collective contribution to the overall performance of the system. Due to the fact that the system cannot deal with solely occluded objects, they were not annotated as ground truth. Since the detection is enhanced by various factors it is vital

Table 6.4: Baggage detection results for the three datasets during the different stages of evolution of the system and comparison with the D. Damen's energy function.

		Accuracy	Precision	Recall	Specificity
Frequency template					
A. Energy Function as defined by D. Damen. $V_{p,q}(f_p f_q) = 3$	PETS	0.57	0.64	0.59	0.55
	Hand-held	0.49	0.72	0.56	0.15
	CCTV	0.63	0.88	0.45	0.90
	Total	**0.56**	**0.69**	**0.56**	**0.57**
B. Energy Function as defined by D. Damen and substituting the trained bags model with gradient weight and **a* b*** images. $V_{p,q}(f_p f_q) = 3$	PETS	0.61	0.69	0.64	0.57
	Hand-held	0.56	0.90	0.56	0.57
	CCTV	0.60	0.79	0.47	0.81
	Total	**0.60**	**0.75**	**0.58**	**0.63**
Colour temporal template					
C. Energy Function as proposed in this chapter. $V_{p,q}(f_p f_q) = 3$	PETS	0.61	0.65	0.69	0.50
	Hand-held	0.66	0.93	0.66	0.64
	CCTV	0.62	0.82	0.50	0.83
	Total	**0.63**	**0.74**	**0.65**	**0.59**
D. Energy Function as proposed in this chapter without the **a* b*** enhancement. $V_{p,q}(f_p f_q) = 3$	PETS	0.56	0.66	0.53	0.59
	Hand-held	0.65	0.98	0.63	0.90
	CCTV	0.54	0.85	0.34	0.89
	Total	**0.58**	**0.77**	**0.53**	**0.68**
E. Energy Function as proposed in this chapter with D. Damen's best model match selection (i.e. template-exemplar alignment). $V_{p,q}(f_p f_q) = 3$	PETS	0.58	0.60	0.75	0.37
	Hand-held	0.62	0.86	0.66	0.41
	CCTV	0.66	0.82	0.56	0.81
	Total	**0.60**	**0.68**	**0.69**	**0.46**
F. Bayes energy function With $\kappa = 1$ and $V_{p,q}(f_p f_q) = 3$	PETS	0.60	0.65	0.66	0.52
	Hand-held	0.67	0.92	0.68	0.60
	CCTV	0.61	0.82	0.47	0.83
	Total	**0.62**	**0.74**	**0.63**	**0.60**

Table 6.5: Comparison of the primary D. Damen's system with the proposed one over the PETS dataset.

	Accuracy	Precision	Recall/Sensitivity	Specificity
Proposed method	0.61	0.65	0.69	0.50
D. Damen's method	0.50	0.54	0.55	0.44

to check the degree of enhancement that they offer at each stage. Numerous experiments were conducted on all the datasets changing the different components of the system.

It is obvious that the occurrence of coefficients at different stages is quite frequent and their selection was not a trivial task. However they are designed in such a way so that the one is derived from another. Finally there is only one coefficient left that if changed would determine the sensitivity of the system. The performance of the system has been thoroughly examined for the different values of this coefficient over the PETS dataset. The coefficient values that gave the best results were tested on the other two datasets for the selection of the final coefficient value. Therefore the selected coefficient in Equation 6.16 is $\kappa = 0.75$.

To begin with, the significance of L**a*b*** colour information to the detection of bags should be examined. It is also important to incorporate the energy function as defined by D. Damen into the proposed system and see how exactly the two different energy functions influence the final results when applied on the same data. Since the effectiveness of the trained models is questionable, they are replaced with the gradient weight $\hat{W}$ and the L**a*b*** images in D. Damen's energy function. Afterwards, the energy function that produces the best results is compared with the original method proposed by D. Damen.

Table 6.4 shows all the results for the 3 datasets and 6 different energy functions. The total values in Table 6.4 are calculated by combining the results from all the datasets, not by averaging the percentages obtained. The cases selected for examination are: A) the energy function as defined by D. Damen in [33] applied on the FTT and combined with the trained bags models; B) the same D. Damen's energy function but with gradient weight $\hat{W}$ and **a*** **b*** enhancement instead of the trained bags models; C) the energy function as defined in this chapter applied on the CTT; D) the same energy function but without the **a*** **b*** information; E) the proposed energy function but with D. Damen's best model match search; and finally F) the Bayes energy function. By examining the table it can be inferred that the **a*** **b*** images offer a substantial enhancement to the proposed system and to D. Damen's energy function as well. By employing **a*** **b*** images in the case C the precision drops 3% while the recall increases 12% and as a consequence

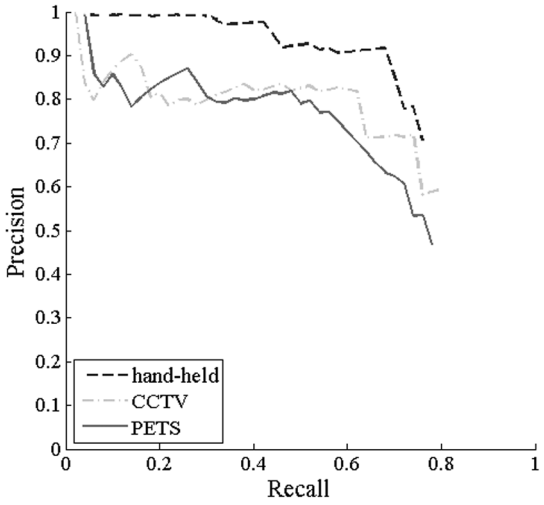


Figure 6.19: Precision-recall curves for the final and improved system in case C of Table 6.4 for the 3 different datasets.

the accuracy level increases from to 58% to 63%.

The low precision and specificity values in the case A of Table 6.4 show that the D. Damen's energy function suffers from a significant number of false positives. The aim is to retain the recall values but at the same time to increase the precision as and the definition of the shape of the segmented bags. Concerning the alignment of the temporal template with the best matching exemplar the cases C and E should be considered. It is obvious that D. Damen's method does not serve the requirements of the proposed system and therefore it has been modified.

In general the specificity numbers are low which shows the weakness of the system in recognizing the negative cases. However the accuracy and precision levels have been increased. The reader is advised to observe the table to make further conclusions.

Case C, which is proposed in this chapter, provides the best results; therefore, it is the one to be compared with the D. Damen's original system. The results are presented in Table 6.5.

To test the accuracy of the system, the construction of Precision-Recall (PR) curve was employed as described in [114] and [4]. The ground truth box for the position of the bags on the temporal templates was obtained manually. A Detection is considered as successful only if the overlap between the ground truth bounding box and the detected one is higher than 20%. In case the overlap is between 0% and 20% then, if the bounding box obtained is inside that of the ground truth then it is labeled as false negative, else if the ground truth box is inside the bounding box obtained then it is labeled as a false positive. Any other case would suggest that the bounding box obtained and the ground truth box is

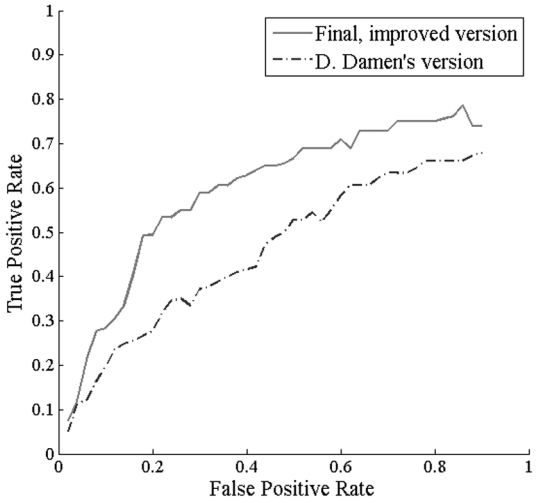


Figure 6.20: Receiver operating characteristic curves for the final and improved system in case C of Table 6.4 for and the primary one as reported by D. Damen.

not related to each other and therefore the detection is labeled as a false positive and a false negative, respectively.

The Precision-Recall (PR) curves in Figure 6.19 are derived from each of the three available datasets while the ones in Figure 6.21 are only derived from the PETS dataset. They are the result of linear interpolation of recall points corresponding to maximum precision. The receiver operating characteristic (ROC) curves in Figure 6.20 are constructed as complementary to the PR curves. In Figure 6.19 the results are very encouraging for the in-house videos captured via the hand-held cameras; this is because the camera angle and distance is optimised for the purpose of the application. The curves in Figure 6.20 and Figure 6.21 compare the proposed method with that of D. Damen's, for the PETS dataset. The ROC curves show that the false positive rate has been eliminated and the true positive rate has been increased.

The last metric introduced is the Average Precision (AP), which summarises the shape of the precision/recall curve, and is defined as the mean precision at a set of eleven equally spaced recall levels [0, 0.1, ..., 1] [117]:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{interp}(r) \tag{6.30}$$

The precision at each recall level is interpolated by taking the maximum precision measured for all recalls greater than r

$$p_{interp}(r) = \max_{\bar{r}:\bar{r} \geq r} p(\bar{r}) \tag{6.31}$$

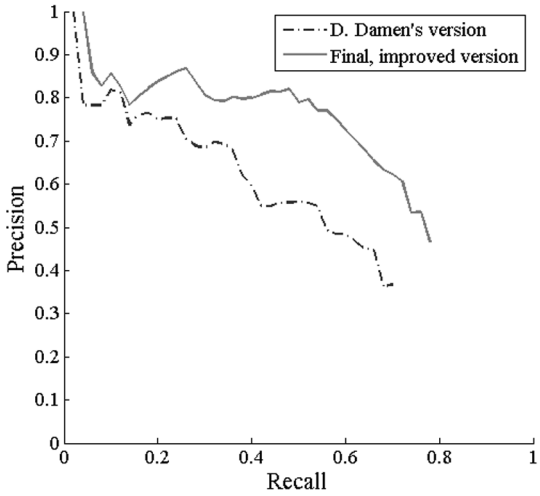


Figure 6.21: Precision-recall curves for the final and improved system in case C of Table 6.4 and the primary one as reported by D. Damen.

where $p(\bar{r})$ is the measured precision at recall $\bar{r}$. The calculated AP for the improved version is 0.64 while for the D. Damen's system it is 0.49. These results show the significance of the improvement of the system.

6.8 Summary and Discussion

In this chapter have been presented a number of innovative methods that incrementally improve the overall performance of the baggage detection system originally proposed by D. Damen.

The first goal was to develop a robust direction estimation algorithm. For this reason the pedestrian's shoulder shape features and their displacement over the image plane have been exploited. However, not all available motion information has been fully utilised. For example the duration of motion on the scene and the silhouette size variation during motion can indicate the direction of motion. Hence, there where objects' size remains approximately the same as they move, tend to cross the scene horizontally, while the ones that are getting smaller in size tend to walk vertically. These are perspective related properties that could be employed to enhance the proposed algorithm.

The next goal was to identify the clothes and torso of the person. Therefore, the GrowCut algorithm used can be combined with colour clustering techniques to improve the accuracy of torso detection. In such case the torso of the best matched template should be substituted by the one obtained. Another important aspect was the segmentation of bags using the **a*** and **b*** components of the L**a*b*** colour space. Since the L**a*b*** colour space values change with illumination, good

illumination conditions are in favour of the proposed method. To this end, the system has been tested under 3 different illumination conditions, where 2 of them were outdoors. The achieved results were encouraging.

As a summary the system has been improved not only in terms of accuracy and precision but also in accurate bag shape segmentation. However, the method still suffers from some inaccurate alignments of the exemplar with the temporal template and some inaccurate direction estimations. Other major challenges that need further attention are loose clothes and partially occluded or small objects.

Chapter 7

Conclusions and Future Perspectives

The research in this thesis focuses on moving object segmentation via background subtraction and carried object detection and identification in video sequences. Other contributions refer to the estimation of human body orientation and design of an edge continuity enhancing filter, as part of developing the above algorithms. The steps taken to achieve the above contributions are those that differentiate the current work from other methods. These steps, the relevant original contributions made and the conclusions made are outlined in section 7.1. Section 7.2 discusses the limitations of the proposed algorithms that arise mainly from practical and operational complexities of the nature of the open research problems addressed in this thesis. Finally the chapter concludes with suggestions for future research (section 7.3).

7.1 Conclusion

The novel moving object segmentation algorithm based on background subtraction presented in chapter 4 of this thesis made use of multi-directional gradient and complementary phase congruency features to model the background. These features ensured the detection of crude foreground object contours that are subjected to gradual illumination changes and moderate shadow conditions. Due to the fact that moderate shadows lack sufficiently well-defined contours and may extend their influence to areas beyond the boundary of an object, it was decided that targeting such contours rather than regions is more sensible. Therefore a filter that would facilitate foreground contour continuity was designed as an intermediate between the traditional Gaussian and the truncated *sinc* filters. If reflected onto the edge map of an image, the crude foreground contours were replaced by the

edges of foreground objects. This operation enabled further contour refinement by eliminating edges that were present due to noise and shadows.

It was shown that while the elimination of noise edges can be simply achieved by examining the colour similarity across each side of an edge segment, the reduction of shadows required more sophisticated measures. To that end, four measures of region similarity along an edge segment were defined to discriminate between shadow and non-shadow edges. In addition to that, a subsequent contour completion step, which can be employed as a post-processing step in any foreground segmentation scheme, was incorporated into the system. This ensured closed foreground contours and completed object shapes without discontinuities and breaks. The conducted experiments proved the capability of the proposed methods to eliminate ghosting effects that occur during background initialisation and at the moment that a previously stationary background objects begin to move. (e.g. cars stopped at traffic lights beginning to move). Detailed experimental analysis demonstrated the ability of the proposed foreground object segmentation approach to be in par with the state-of-the-art algorithms and in specific cases, to supersede their performance.

A novel viewing direction estimation was proposed in chapter 5. The viewing direction estimation algorithm made use of features derived from the shape of the shoulders of a human being. The shoulder area exhibits better stability in shape during a walking cycle, as compared to the upper and lower limbs. These features described how the head and shoulder proportions change under different viewpoints. The experimental results revealed that the proposed method performs better in comparison to those methods that adopt the transfer of motion from the image plane to ground plane in order to recover the viewing direction.

The COD system based on colour information proposed in chapter 6 made use a CTT, which contains colour, textural and frequency information. It was shown that to consider COs as abnormal silhouette protrusions it is vital to separate the protruding regions that belong to body parts or clothing. To that end a torso estimation method was applied as a region growing procedure in one of the colour channels. This enhanced the torso shape of the best matched exemplar model and reduced the occurrence of false positive detections. In addition it was noted that the colour temporal template facilitated accurate segmentation of those carried objects that had intense colour saturation. To reduce the likelihood of head and feet being detected as carried objects, weighting was imposed on the human temporal template. Experiments conducted to analyse the performance of the proposed COD approach proved that it outperformed the current state-of-the-art algorithm in terms of its robustness to practical challenges.

Carried object recognition is a challenging task due to the wide range of ob-

jects that could be carried by general public. Assuming that the interest of the proposed surveillance application is common public spaces such as airports and train stations, the objects that are most likely to be carried are different luggage types. Therefore, a bag type classification algorithm was proposed in chapter 5 by examining their locations in relevance to the human body that carries them. Detailed experimental results conducted on five commonly used baggage types revealed an overall classification accuracy of 70.95%.

It was observed that the foreground segmentation algorithm employed for silhouette extraction for COD affects the accuracy of the final outcome. For instance, the cast shadows extend the shape of the silhouettes and the broken or incomplete foreground shapes reduce the likelihood of presence of carried objects. This was the incentive for the development of the proposed foreground segmentation algorithm that eliminates shadows and guarantees complete silhouette contours. The detailed performance evaluation of the proposed novel CFC segmentation algorithm within the framework of the proposed COD system requires further attention and is proposed as a future developmental task.

7.2 Limitations

The proposed algorithms are constrained by several limitations if specific scenarios are to be served. This section critically reviews the underlying reasons for the limitations and the potential impact they may have in practical applications.

First, the methodology followed in the proposed CFC algorithm intrinsically does not favour dynamic backgrounds that include objects such as, tree leaves, fountains, or in general a rather cluttered background. The edges that belong to tree leaves are unlikely to be removed due to the high curvature of their shapes and the absence of a specific pattern, i.e. presence of random shapes. Moreover,

Figure 7.1: An example output of CFC algorithm where parts of the background are included into the foreground.

for example, if the background of an edge of a leaf is the sky, it is not removed as according to the specified rules adopted in segmentation, dissimilar colours across the edge segment imply that it belongs to foreground. Hence, in the case of tree leaves the edges that are not eliminated are extended into a closed shape, which includes all tree leaves along with some patches of the sky. This increases the false positive rate detection if compared with other foreground segmentation methods, which incorporate in the foreground only the waving leaves and not the sky patches. Another example is the integration of the background area between the limbs of a human object to the foreground, when strong shadow edges are not removed (see Figure 7.1). Given the fact that the region between the limbs is the same as the background, it can be removed if post processing at region level is applied. Nonetheless, the proposed CFC algorithm performs specifically well for traffic monitoring, where the shadows are frequent, the recovery of the whole car is challenging, and the car shapes are convex in contrast to humans.

The viewing direction estimation module is affected by anything that deforms the natural shape of the segmented silhouette. This could be a load carried high on the shoulders or over the head, or the shadows beneath the legs, or even falsely segmented additional foreground. As most of the features are related to the shape of the shoulder and one feature to the height of the silhouette, in case of the above mentioned shape deformations, it is expected that there will be classification errors. Therefore, a robust foreground segmentation method is very important in every aspect of COD process. A positive fact is that the proposed viewing direction estimation method is stable under in-plane rotations.

As it was mentioned above the performance of the COD highly depends on the accuracy of foreground object segmentation. Therefore this is the most important and fundamental constraint. Assuming that the foreground segmentation is absolutely correct, the next problem would be the viewing direction estimation of distorted body shapes. Another challenge is the partially occluded carried objects, which do not protrude enough to be detected. For instance, it is difficult to detect the bright pink bag in Figure 7.1 as it does not protrude significantly. However, since the proposed enhancement based on colour object segmentation favours the objects with intense colours, the pink bag is partially segmented. The rest of the bag could be recovered if region growing or simple colour clustering is applied. It should be noted that, the detection of white and black carried objects will solely depend on their grey level intensity. This means that the likelihood of detecting white objects is very low, and could be enhanced by a frequency temporal template. Finally detailed experiments have also revealed that the lose clothes or people of bigger than average size are more likely to produce false positive detections.

The proposed bag type recognition approach is limited only to carried but not still luggage classification. The fact that the bag type recognition depends only on their location, it is difficult to discriminate the bag types that reside at the same or nearby locations. For example the most bulged part of a backpack resides at a location similar to the typical location of a ladies shoulder bag. Thus under some viewing directions their locations appear to be similar.

7.3 Future Work

To address the above mentioned limitations the following directions for future research are proposed.

The proposed CFC algorithm is able to detect all available moving contours and successfully complete them. The only stage that requires attention and if enhanced could positively affect the final results is the edge segment classification. More research should be undertaken to deploy texture features invariant to strong shadows. To handle various types of dynamic background the edge segment classification principles could be altered.

To address issues such as camera jitter and various forms of dynamic background, a texture based background modelling approach needs to be designed. Texture features such as Colour Co-occurrence Matrix (CCM), widely used in texture recognition [106], can be employed. The CCM accommodates the inter channel relationship of pixel colours within a specified region into a histogram of occurrences. This means that the CCM will remain approximately the same if the contents within the region are rearranged. The effectiveness of the CCM in texture recognition over the single channel co-occurrence matrix has been proved in [106, 38]. An aspect that might impair the performance of a foreground segmentation method base on CCM is the high dimension of CCM which increases in proportion with the quantisation levels. It is understood that there is a trade-off between accuracy and computational cost that should be balanced. The first and most important perspective would be balancing the computational cost with the colour quantisation levels by reducing the dimensionality of the CCM. Another perspective is the examination of performance of the Haralick features [52].

The experiments conducted reveal that there is space for the improvement of the proposed COD algorithm. The likelihood functions can be continuously updated by new data. One helpful cue could be the distance of the carried object from the body main axis in combination with the pixel intensities that make it up. If the central point under the feet of the temporal template is taken as a reference, then the distance of each pixel from that point can be calculated and positioned into a 3D map as in Figure 7.2. The x axis represents the distance,

the y axis represents the pixel intensity and the colour encodes the frequency of occurrence. It is noticed that the pixels that belong to carried objects form clusters at some distance from zero. Figure 7.2 shows two examples of carried object and two examples of unencumbered temporal templates. This information can be effectively used not only for COD but also for bag type classification based on distance. The bag type classification can be extended by exploiting shape information.

The most challenging task in object detection and tracking is occlusion handling. In real world scenarios people usually move in groups, resulting in occluded view of moving and carried objects. Future research can concentrate on patch based object detection and recognition. This means that the method to be developed should make use of features that do not characterise the object as a whole but other smaller characteristic parts of the object. Another perspective can rely on combining multiple views of the same object.

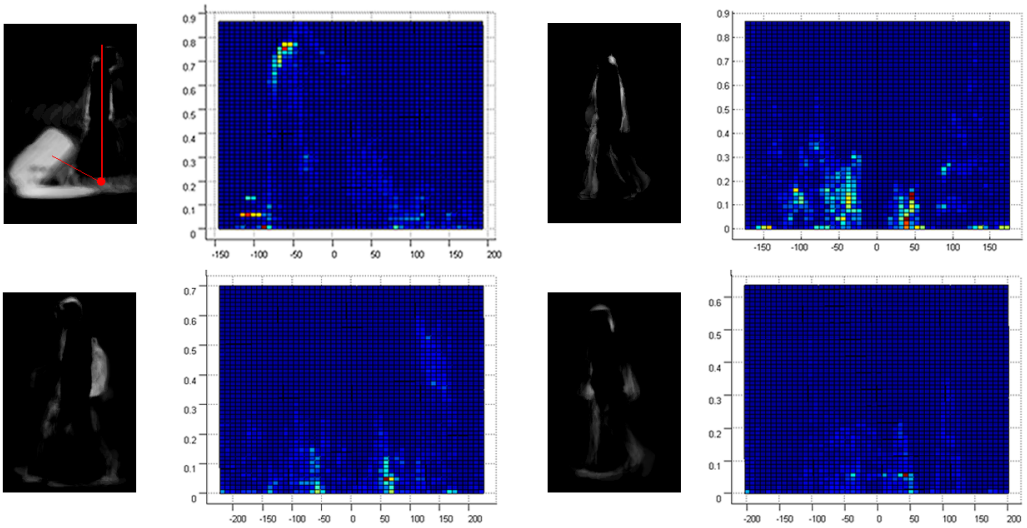


Figure 7.2: From left to right: In the first and second columns are shown the protruding carried objects with their distance-intensity maps, and in the third and fourth column are shown unencumbered temporal templates with the corresponding distance-intensity maps.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–882–II–888 Vol.2, 2004.

[2] Hal S. Stern Andrew Gelman, John B. Carlin and Donald B. Rubin. *Bayesian Data Analysis*. CRC Press, 2004.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 623–630. IEEE, 2010.

[4] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[5] Kirstie Ball, David Lyon, David Murakami Wood, Clive Norris, and Charles Raab. A report on the surveillance society. Technical report, 2006.

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

[7] C. BenAbdelkader and L. Davis. Detection of people carrying objects : a motion-based recognition approach. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 378–383, 2002.

[8] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[9] T. Bouwmans. Subspace learning for background modeling: A survey. *Recent Patent On Computer Science*, 2(3):223–234, 2009.

[10] T. Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey". *Recent Patents on Computer Science*, 4(3):147–176, 2011.

[11] T. Bouwmans and F. El Baf. Modeling of dynamic backgrounds by type-2 fuzzy gaussian mixture models. *MASAUM J. Basic Appl. Sci.*

[12] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, 1(3):219–237, November 2008.

[13] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, Sept 2004.

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[15] G.R. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 238–244, 2000.

[16] A. Branca, M. Leo, G. Attolico, and A. Distante. Detection of objects carried by people. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages III–317–III–320 vol.3, 2002.

[17] A. Branca, M. Leo, G. Attolico, and A. Distante. People detection in dynamic images. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2428–2432, 2002.

[18] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*, chapter 1, pages 1–14. SIAM, 1995.

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

[20] Antoni B. Chan, Vijay Mahadevan, and Nuno Vasconcelos. Generalized stauffer–grimson background subtraction for dynamic scenes. *Mach. Vision Appl.*, 22(5):751–766, September 2011.

[21] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[22] Lun-Yu Chang and W.H. Hsu. Foreground segmentation for static video via multi-core and multi-modal graph cut. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1362–1365, June 2009.

[23] R. Chayanurak, N. Cooharojananone, S. Satoh, and R. Lipikorn. Carried object detection using star skeleton with adaptive centroid and time series graph. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 736–739, 2010.

[24] Cheng Chen, A. Heili, and J. Odobez. Combined estimation of location and body pose in surveillance video. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 5–10, 2011.

[25] Fan-Chieh Cheng, Shih-Chia Huang, and Shanq-Jang Ruan. Advanced motion detection for intelligent video surveillance systems. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 983–984, New York, NY, USA, 2010. ACM.

[26] Hamilton Y. Chong, Steven J. Gortler, and Todd Zickler. A perception-based color space for illumination-invariant image processing. *ACM Trans. Graph.*, 27(3):61:1–61:7, August 2008.

[27] Chi-Hung Chuang, Jun-Wei Hsieh, Luo-Wei Tsai, Sin-Yu Chen, and Kuo-Chin Fan. Carried object detection using ratio histogram and its application to suspicious event analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(6):911–916, 2009.

[28] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, May 2003.

[29] Marco Cristani and Vittorio Murino. Background subtraction with adaptive spatio-temporal neighborhood analysis. In *VISAPP 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications, Funchal, Madeira, Portugal, January 22-25, 2008 - Volume 2*, pages 484–489. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008.

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

[31] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):781–796, Aug 2000.

[32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[33] D. Damen and David Hogg. Detecting carried objects from sequences of walking pedestrians. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(6):1056–1067, 2012.

[34] Dima Damen and David Hogg. Detecting carried objects in short video sequences. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision ECCV 2008*, volume 5304 of *Lecture Notes in Computer Science*, pages 154–167. Springer Berlin Heidelberg, 2008.

[35] J.W. Davis and A.F. Bobick. The representation and recognition of human movement using temporal templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 928–934, Jun 1997.

[36] Brian Decann and Arun Ross. Gait curves for human recognition, backpack detection and silhouette correction in a nighttime environment.

[37] Radu Dondera, Vlad Morariu, and Larry Davis. Learning to detect carried objects with minimal supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.

[38] A. Drimbarean and P. F. Whelan. Experiments in colour texture analysis. *Pattern Recogn. Lett.*, 22(10):1161–1167, August 2001.

[39] A. Elgammal. Nonlinear generative models for dynamic shape and dynamic appearance. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, pages 182–182, June 2004.

[40] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *FRAME-RATE WORKSHOP, IEEE*, pages 751–767, 2000.

[41] Yunus A. engel. *Heat Transfer: A Practical Approach 2nd ed.* McGraw-Hill, 2003.

[42] M. Enzweiler and D.M. Gavrila. Integrated pedestrian classification and orientation estimation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 982–989, 2010.

[43] R.H. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 71 –76, 30 2011-sept. 2 2011.

[44] Adolph Fick. On liquid diffusion. *Journal of Membrane Science*, 100(1):33 – 38, 1995. The early history of membrane science selected papers celebrating vol. 100.

[45] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[46] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, pages 175–181, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[47] N.M. Ghanem and L.S. Davis. Human appearance change detection. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 536–541, 2007.

[48] Debora Gil and Petia Radeva. Extending anisotropic operators to recover smooth shapes. *Comput. Vis. Image Underst.*, 99(1):110–125, July 2005.

[49] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8, 2012.

[50] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. Efficient subpixel image registration algorithms. *Opt. Lett.*, 33(2):156–158, Jan 2008.

[51] A.A. Handzel and T. Flash. Affine invariant edge completion with affine geodesics. In *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pages 97–103, 2001.

[52] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610 –621, nov. 1973.

[53] I. Haritaoglu, R. Cutler, D. Harwood, and L.S. Davis. Backpack: detection of people carrying objects using silhouettes. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 102–107 vol.1, 1999.

[54] I. Haritaoglu, D. Harwood, and L.S. Davis. Hydra: multiple people detection and tracking using silhouettes. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 280–285, 1999.

[55] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809–830, 2000.

[56] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4: Who? when? where? what? a real time system for detecting and tracking people, 1998.

[57] Richard J. Harris. *A primer of multivariate statistics.* Lawrence Erlbaum Associates, 2001.

[58] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657 –662, april 2006.

[59] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38 –43, june 2012.

[60] Thanarat Horprasert, David Harwood, and Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. pages 1–19, 1999.

[61] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification, 2000.

[62] Mao-Hsiung Hung, Jeng-Shyang Pan, and Chaur-Heh Hsieh. Speed up temporal median filter for background subtraction. In *Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on*, pages 297–300, Sept 2010.

[63] Omar Javed, Khurram Shafique, and Mubarak Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Proceedings of the Workshop on Motion and Video Computing*, MOTION '02, pages 22–, Washington, DC, USA, 2002. IEEE Computer Society.

[64] Omar Javed and Mubarak Shah. 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, 2002. Springer-Verlag.

[65] Sheng Jiang and Yingying Zhao. Background extraction algorithm base on partition weighed histogram. In *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, pages 433–437, Sept 2012.

[66] Nancy L. Nihan Mark E. Hallenbeck Jianyang Zheng, Yinhai Wang. Extracting roadway background image: Mode-based approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1944(1):82–88, 2006.

[67] John H Lienhard IV John H Lienhard, V. *A Heat Transfer Textbook: Fourth Edition*. 2011.

[68] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In Paolo Remagnino, GraemeA. Jones, Nikos Paragios, and CarloS. Regazzoni, editors, *Video-Based Surveillance Systems*, pages 135–144. Springer US, 2002.

[69] Wen-xiong Kang, Wen-zhuo Lai, and Xiang-bao Meng. An adaptive background reconstruction algorithm based on inertial filtering. *Optoelectronics Letters*, 5(6):468–471, 2009.

[70] Benjamin B. Kimia, Ilana Frankel, and Ana-Maria Popescu. Euler spiral for shape completion. *Int. J. Comput. Vision*, 54(1-3):157–180, August 2003.

[71] Ross Kindermann, 1925-(joint author.) Snell, J. Laurie (James Laurie), and American Mathematical Society. *Markov random fields and their applications / Ross Kindermann, J. Laurie Snell*. Providence, R.I. : American Mathematical Society, 1980. References p.133-142.

[72] Granino Arthur Korn and Theresa M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Publications, 2000.

[73] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[74] Peter Kovesi. Image features from phase congruency. *Videre*, 1(3), 1999.

[75] Peter Kovesi. Phase congruency: A low-level image invariant. *Psychological Research*, 64(2):136–148, 2000.

[76] Peter Kovesi. Phase congruency detects corners and edges. In *in The Australian Pattern Recognition Society Conference: DICTA 2003*, pages 309–318, 2003.

[77] Fredrik Kristensen, Peter Nilsson, and Viktor Öwall. Background segmentation beyond rgb. In *Proceedings of the 7th Asian conference on Computer Vision - Volume Part II*, ACCV'06, pages 602–612, Berlin, Heidelberg, 2006. Springer-Verlag.

[78] Chung-Ming Kuo, Wei-Han Chang, Sheng-Bin Wang, and Chih-Shan Liu. An efficient histogram-based method for background modeling. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, pages 480–483, Dec 2009.

[79] Anh-Nga Lai, Hyosun Yoon, and Gueesang Lee. Robust background extraction scheme using histogram-wise for real-time tracking in urban traffic video. In *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, pages 845–850, July 2008.

[80] Jean-Franois Lalonde, AlexeiA. Efros, and SrinivasaG. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 322–335. Springer Berlin Heidelberg, 2010.

[81] Chan-Su Lee and Ahmed Elgammal. Carrying object detection using pose preserving dynamic shape models. In *Proceedings of the 4th international conference on Articulated Motion and Deformable Objects*, AMDO'06, pages 315–325, Berlin, Heidelberg, 2006. Springer-Verlag.

[82] Chan-Su Lee and Ahmed Elgammal. Dynamic shape outlier detection for human locomotion. *Computer Vision and Image Understanding*, 113(3):332 – 344, 2009. Special Issue on Video Analysis.

[83] L. Lee, G. Dalley, and K. Tieu. Learning pedestrian models for silhouette refinement. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 663–670 vol.1, Oct 2003.

[84] L. Lee and W. E L Grimson. Gait analysis for recognition and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 148–155, May 2002.

[85] Bo Li, Zhen Tang, Baozong Yuan, and Zhenjiang Miao. Segmentation of moving foreground objects using codebook and local binary patterns. In *Image and Signal Processing, 2008. CISP '08. Congress on*, volume 4, pages 239 –243, may 2008.

[86] Stan Z. Li. Ch. 13. modeling image analysis problems using markov random fields. In D.N. Shanbhag and C.R. Rao, editors, *Stochastic Processes: Modelling and Simulation*, volume 21 of *Handbook of Statistics*, pages 473 – 513. Elsevier, 2003.

[87] Fengjun Lv, Tao Zhao, and R. Nevatia. Camera calibration from video of a walking human. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1513–1518, 2006.

[88] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, 17(7):1168 –1177, july 2008.

[89] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26, June 2012.

[90] Shyjan Mahamud, Lance R. Williams, Karvel K. Thornber, and Kanglin Xu. Segmentation of multiple salient closed contours from real images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:2003, 2003.

[91] R. Manduchi and P. Perona. Pyramidal implementation of deformable kernels. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 378–381 vol.1, Oct 1995.

[92] R. Manduchi, P. Perona, and D. Shy. Efficient deformable filter banks. *Signal Processing, IEEE Transactions on*, 46(4):1168–1173, Apr 1998.

[93] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms

and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[94] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, May 2004.

[95] N.J.B. McFarlane and C.P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, 1995.

[96] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302 – II–309 Vol.2, june-2 july 2004.

[97] A. Nabout, Bing Su, and H.A. Nour Eldin. A novel closed contour extractor, principle and algorithm. In *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, volume 1, pages 445–448 vol.1, Apr 1995.

[98] V.S. Nalwa. Edge-detector resolution improvement by image interpolation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(3):446–451, May 1987.

[99] H. Nanda, C. Benabdelkedar, and Larry S. Davis. Modelling pedestrian shapes for outlier detection: a neural net based approach. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 428 – 433, 2003/06/ 2003.

[100] M. Narayanan and B. Kimia. To complete or not to complete: Gap completion in real images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 47–54, June 2012.

[101] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 9 –14, june 2012.

[102] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, 1979.

[103] Jonathan Owens, Andrew Hunter, and Eric Fletcher. A fast model-free morphology-based object tracking algorithm. In Paul L. Rosin and A. David Marshall, editors, *BMVC*. British Machine Vision Association, 2002.

[104] O. Ozturk, Toshihiko Yamasaki, and Kiyoharu Aizawa. Estimating human body and head orientation change to detect visual attention direction. In Reinhard Koch and Fay Huang, editors, *Computer Vision, ACCV 2010 Workshops*, volume 6468 of *Lecture Notes in Computer Science*, pages 410–419. Springer Berlin Heidelberg, 2011.

[105] Chia-Jung Pai, Hsiao-Rong Tyan, Yu-Ming Liang, Hong-Yuan Mark Liao, and Sei-Wang Chen. Pedestrian detection and tracking at crossroads. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–101–4 vol.3, Sept 2003.

[106] Christoph Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965 – 976, 2004.

[107] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1034 – I–1040 vol.1, 2001.

[108] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[109] Pietro Perona. Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:488–499, 1991.

[110] Jean Philibert. One and a half century of diffusion: Fick, einstein, before and beyond. *The Open-Access Journal for the Basic Principles of Diffusion Theory, Experiment and Application*, 4:1–19, 2006.

[111] Yue Qi, Guo-Chang Huang, and Yi-Ding Wang. Carrying object detection and tracking based on body main axis. In *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, volume 3, pages 1237–1240, 2007.

[112] Yang Ran, Qinfen Zheng, Rama Chellappa, and Thomas M. Strat. Applications of a simple characterization of human gait in surveillance. *Trans. Sys. Man Cyber. Part B*, 40(4):1009–1020, August 2010.

[113] Vikas Reddy, Conrad Sanderson, Andres Sanin, and BrianC. Lovell. Mrf-based background initialisation for improved foreground detection in cluttered surveillance videos. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision ACCV 2010*, volume 6494 of *Lecture Notes in Computer Science*, pages 547–559. Springer Berlin Heidelberg, 2011.

[114] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[115] J.L. Romeu. Empirical assessment of normal and lognormal distribution assumptions. *Selected Topics in Assurance Related Technologies*, 9(6), 2002-6.

[116] Lukas Rybok, M. Voit, H.K. Ekenel, and R. Stiefelhagen. Multi-view based estimation of human upper-body orientation. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1558–1561, 2010.

[117] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[118] Janos Schanda. *Colorimetry: Understanding the CIE System*. John Wiley & sons, Inc., Inc., Hoboken, New Jersey, USA, 2007.

[119] Harry Moritz Schey. *Div, grad, curl and all that: an informal text on vector calculus. 4th ed.* W W Norton & Company Incorporated, 2005, 2005.

[120] A. Schick, M. Bauml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 27 –31, june 2012.

[121] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *Proceedings of the 7th Asian conference on Computer Vision - Volume Part II*, ACCV'06, pages 581–590, Berlin, Heidelberg, 2006. Springer-Verlag.

[122] DavidW. Scott. Multivariate density estimation and visualization. In James E. Gentle, Wolfgang Karl Hardle, and Yuichi Mori, editors, *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, pages 549–569. Springer Berlin Heidelberg, 2012.

[123] T. Senst, R.H. Evangelio, and T. Sikora. Detecting people carrying objects based on an optical flow motion model. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 301–306, Jan 2011.

[124] Tobias Senst, Ruben Heras Evangelio, Volker Eiselein, Michael Ptzold, and Thomas Sikora. Towards detecting people carrying objects - a periodicity dependency pattern approach. In *VISAPP (2)'10*, pages 524–529, 2010.

[125] Douglas Shy and Pietro Perona. X-y separable pyramid steerable scalable kernels. In *CVPR*, pages 237–244, 1994.

[126] B. W. Silverman. *Density estimation for statistics and data analysis*. Published in Monographs on Statistics and Applied Probability. London: Chapman and Hall, 1986., London, 1998.

[127] Devinderjit Sivia and John Skilling. *Data Analysis A Bayesian Tutorial*. Oxford University Press, 2006.

[128] Kai-Tai Song and Jen-Chao Tai. Real-time background estimation of traffic imagery using group-based histogram. *J. Inf. Sci. Eng.*, 24(2):411–423, 2008.

[129] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747 –757, aug 2000.

[130] Chris Stauffer and W. E L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[131] Yunda Sun, Bo Li, Baozong Yuan, Zhenjiang Miao, and Chengkai Wan. Better foreground segmentation for static cameras via new energy form and dynamic graph-cut. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 49 –52, 0-0 2006.

[132] Richard Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, January 2006.

[133] Dacheng Tao, Xuelong Li, S.J. Maybank, and Xindong Wu. Human carrying status in visual surveillance. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1670–1677, 2006.

[134] Aryana Tavanai, Muralikrishna Sridhar, Feng Gu, AnthonyG. Cohn, and DavidC. Hogg. Carried object detection and tracking using geometric shape models and spatio-temporal consistency. In Mei Chen, Bastian Leibe, and Bernd Neumann, editors, *Computer Vision Systems*, volume 7963 of *Lecture Notes in Computer Science*, pages 223–233. Springer Berlin Heidelberg, 2013.

[135] Patrick Cheng-San Teo. *Theory and Applications of Steerable Functions*. PhD thesis, Stanford University. Computer Science Dept, 1998.

[136] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255 –261 vol.1, 1999.

[137] M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37, June 2012.

[138] Vladimir Vezhnevets and Vadim Konouchine. "growcut" - interactive multi-label nd image segmentation by cellular automata. In *international conference on Computer Graphics and Vision*.

[139] Hanzi Wang and D. Suter. A re-evaluation of mixture of gaussian background modeling [video signal processing applications]. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages ii/1017 – ii/1020 Vol. 2, march 2005.

[140] Hanzi Wang and David Suter. Tracking and segmenting people with occlusions by a sample consensus based method. In *In IEEE International Conference on Image Processing (ICIP*, 2005.

[141] Liang Wang, Tieniu Tan, Weiming Hu, and Huazhong Ning. Automatic gait recognition based on statistical shape analysis. *Image Processing, IEEE Transactions on*, 12(9):1120–1131, Sept 2003.

[142] Zhiyu Wang, Hui Xu, Lifeng Sun, and Shiqiang Yang. Background subtraction in dynamic scenes with adaptive spatial fusing. In *Multimedia Signal Processing, 2009. MMSP '09. IEEE International Workshop on*, pages 1 –6, oct. 2009.

[143] Joachim Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag (Stuttgart), 1998.

[144] Joachim Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2-3):111–127, 1999.

[145] C. Weinrich, C. Vollmer, and H.-M. Gross. Estimation of human upper body orientation for mobile robotics using an svm decision tree on monocular images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2147–2152, 2012.

[146] George Wolberg. Sampling, reconstruction, and antialiasing. In *The Computer Science and Engineering Handbook*, pages 1270–1299, 1997.

[147] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, Jul 1997.

[148] Andrew Schofield Xiaoyue Jiang and Jeremy Wyatt. Shadow detection based on colour segmentation and estimated illumination. In *Proceedings of the British Machine Vision Conference*, pages 87.1–87.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.87.

[149] Hong Yang, Yihua Tan, Jinwen Tian, and Man Liu. Accurate dynamic scene model for moving object detection. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI –157 –VI –160, 16 2007-oct. 19 2007.

[150] Sheng-Yan Yang and Chiou-Ting Hsu. Background modeling from gmm likelihood combined with spatial and color coherency. In *Image Processing, 2006 IEEE International Conference on*, pages 2801 –2804, oct. 2006.

[151] Jian Yao and J. Odobez. Multi-layer background subtraction based on color and texture. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[152] R. Yumiba, M. Miyoshi, and H. Fujiyoshi. Moving object detection with background model based on spatio-temporal texture. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 352 –359, jan. 2011.

[153] Hongxun Zhang and De Xu. Robust estimation for background subtraction. In *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on*, volume 1, pages 660–664, Aug 2006.

[154] Jiaming Zhang and Chi Hau Chen. Moving objects detection and segmentation in dynamic video backgrounds. In *Technologies for Homeland Security, 2007 IEEE Conference on*, pages 64 –69, may 2007.

[155] Shengping Zhang, Hongxun Yao, and Shaohui Liu. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1556 –1559, oct. 2008.

[156] Shengping Zhang, Hongxun Yao, and Shaohui Liu. Dynamic background subtraction based on local dependency histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1397–1419, 2009.

[157] Shengping Zhang, Hongxun Yao, Shaohui Liu, Xilin Chen, and Wen Gao. A covariance-based method for dynamic background subtraction. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, dec. 2008.

[158] He Zhiwei, Liu Jilin, and Li Peihong. New method of background update for video-based vehicle detection. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 580–584, Oct 2004.

[159] Yue Zhou, Wei Xu, Hai Tao, and Yihong Gong. Background segmentation using spatial-temporal multi-resolution mrf. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 2, pages 8–13, Jan 2005.

[160] Jiejie Zhu, K.G.G. Samuel, S.Z. Masood, and M.F. Tappen. Learning to recognize shadows in monochromatic natural images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 223–230, June 2010.

[161] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, May 2006.

Appendix A

Complementary Experimental Results for the Edge Enhancing Filter

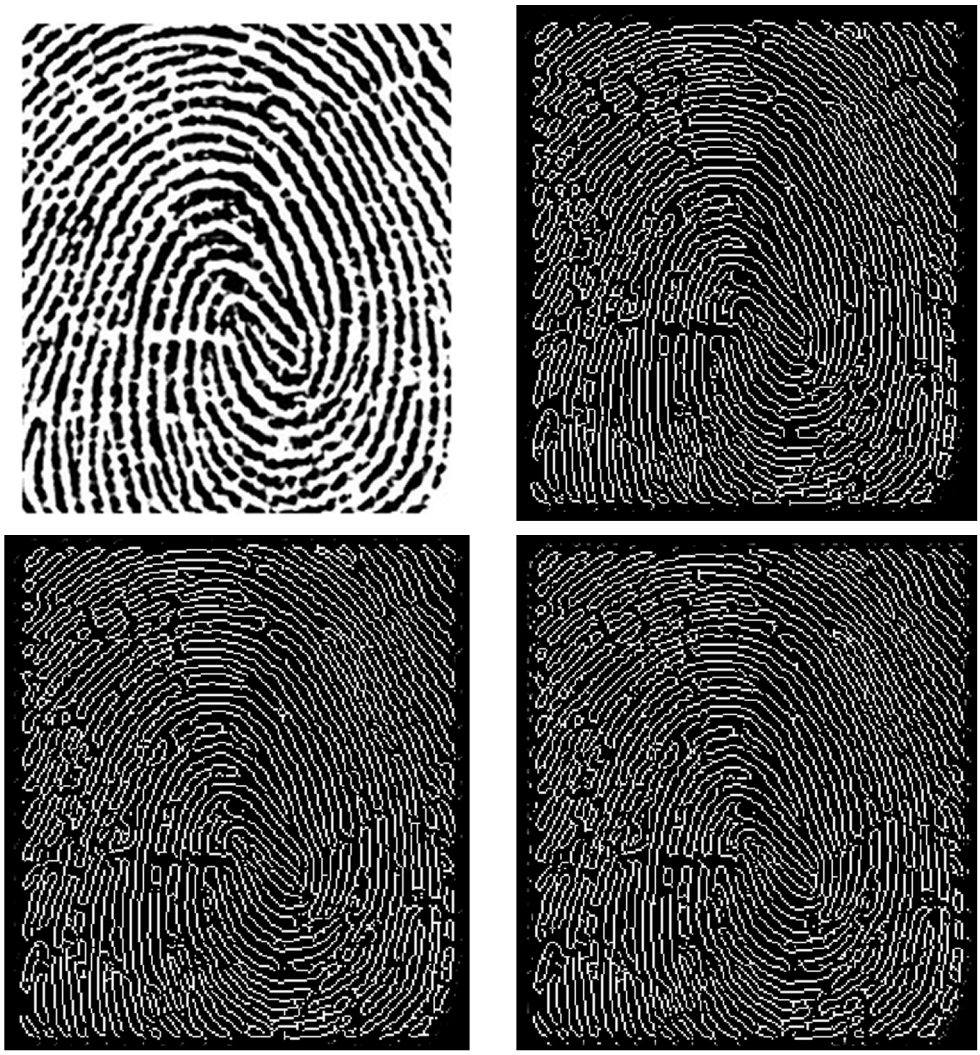


Figure A.1: Fingerprint image filtered with the derivative Gaussian filter of $\sigma = 1.5$, EEF filter and the $sinc(x)$ filter and undergone non-maximum suppression with threshold 0.5. The size of all filters was 13×13 sampled in the interval $[-2\pi, 2\pi]$.

Figure A.2: flower image filtered with the Gaussian filter of $\sigma = 1.5$, EEF filter and the $sinc(x)$ filter. The size of all filters was 13×13 sampled in the interval $[-2\pi, 2\pi]$. The corresponding edges are obtained after non-maximum suppression with threshold 1.3 and hysteresis thresholding with thresholds $thresh_{high} = 0.5$ and $thresh_{low} = 0.01$.

Appendix B

Pseudocode for Foreground Contour Post Processing

```
ProcessForeground(colour_image, foreground, angle, M)
    // M is the maximum moment of phase congruency covariance
    // angle is the direction of gradient
    // foreground is the crude foreground contour
 1  h s v = RGBtoHSV(colour_image)
 2  thresh = 0.3
 3  sigma = 1
 4  edgeS = CannyEdge(s, thresh, sigma)
 5  ind = 1

 6  for thresh = {0.05, 0.08} and sigma = {2, 1}
 7        edgeV(ind) = CannyEdge(v, thresh, sigma)
 8        ind = ind + 1

 9  threshold = 1.3
10  edgeM = NonMaximumSuppression(M, angle, threshold)
11  total_edge = edgeS + edgeV(1) + edgeV(2) + edgeV(3) + edgeM

12  stract_elem1 = square(2)
13  morphologyEdge = Close(Dilate(total_edge, stract_elem1))
14  morphologyForeground = Close(foreground, stract_elem1)

15  edgeInForgr = morphologyEdge * morphologyForeground
16  Binary_Mask = Close(edgeInForgr, stract_elem2)

17  Foreground_edge(1) = edgeS * Binary_Mask
18  Foreground_edge({2, 3, 4}) = edgeV({1, 2, 3}) * Binary_Mask
19  Foreground_edge(5) = edgeM * Binary_Mask

20  for ind = {1, 2, ..., 5}
21        Foreground_edge(ind) = NoiseLineRemove(Foreground_edge(ind),
          angle, colour_image, threshold)
22        Foreground_edge(ind) = ShadowLineRemove(Foreground_edge(ind),
          angle, colour_image, threshold)
23        Total_edge = Total_edge + Foreground_edge(ind)

24  final_foreground_edge =  EdgeExtensionByDiffusion (Total_edge)
```

Figure B.1: Pseudocode for foreground contour post-processing of chapter 4

Appendix C

Graph Cuts for Energy Function Minimisation

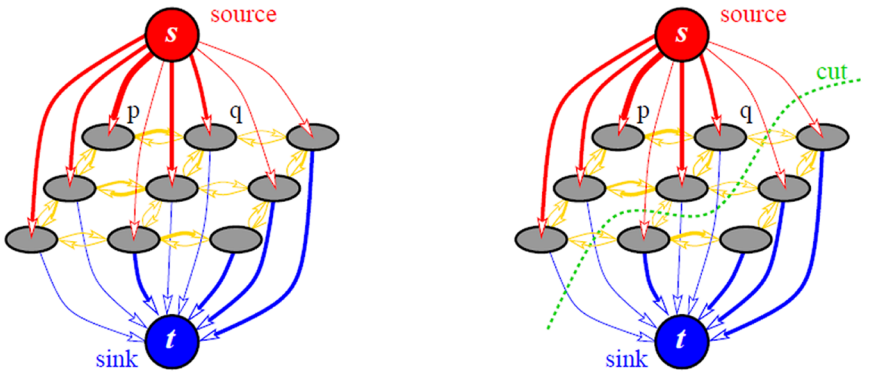
The description of the Graph Cuts principle given below is derived from the work of Y. Boykov and V. Kolmogorov in [13] and Y. Boykov in [14]. The Graph Cuts algorithm proposed in [13, 14] was designed to solve efficiently energy minimisation problems like in Equation 2.8 through finding the minimum cut/maximum flow in graphs.

The energy functions that can be minimised in vision refer to labelling problems such as those described in subsection 2.2.3 by Equation 2.8 and in section 6.5 by Equation 6.13. An image can be considered as an MRF which is an undirected graph with image pixels $X = \{x_1...x_n\}$ considered as nodes and a neighbourhood system $\{N_1...N_n\} = N \subset X$ with N_i the set of neighbours that surround each node x_i. Each pixel $x \in X$ is to be assigned a label from a set of labels $L = \{l_1...l_k\}$. The aim is to find a labelling configuration from the set of configurations $f = \{f_1, ..., f_n\}$ such that f_i is function that assigns a label $l \in L$ to the pixel $x \in X$. This can be achieved by finding the labelling configuration f_i that minimises the following energy function

$$E(f) = \sum_{x_i \in X} D_i(f_i) + \sum_{x_i, x_j \in N} V_{i,j}(f_i, f_j) \tag{C.1}$$

where D_i is an arbitrary data cost function that measures the cost of assigning a label f_i to a pixel x_i, and $V_{i,j}$ is the smoothness cost that defines the interaction between the neighbouring pixels x_i and x_j.

To solve this energy minimisation problem let us consider a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ as in Figure C.1-(a), where $\mathcal{V}$ are the vertices of the graph (in case of image the vertices are pixels) and $\mathcal{E}$ are the directed edges that connect the vertices. The graph in Figure C.1-(a) represents a labelling problem with two labels, which



(a) (b)

Figure C.1: An Example of directed graph. (a) is the graph G while (b) shows the cut over the graph G. The thickness of the edges correspond to the data cost [13].

correspond to the two terminal nodes usually called as sink t and source s. The weight or thickness of the edges connecting the vertices (pixels) to the terminal nodes are proportional to the data cost D_i, while the thickness of edges connecting the neighbouring nodes between them is proportional to the smoothness cost $V_{i,j}$.

An s/t cut separates the graph nodes into two disjoint groups $\mathcal{S}$ and $\mathcal{T}$ where $\mathcal{S}$ contains the source s while $\mathcal{T}$ the sink t. The cost of the cut $\mathcal{C} = \{\mathcal{S}, \mathcal{T}\}$ is equal to the sum of weights of the edges that connect the nodes of the set $\mathcal{S}$ with the nodes of the set $\mathcal{T}$. The minimum cut problem on graphs is to find a partition that ensures the minimum cost $\mathcal{C}$. An example of minimum cut is shown in Figure C.1-(b). The minimum cut on graphs is directly related to the minimisation of energy function in Equation C.1 to solve labelling problems. Since a minimum cut separates the graph into two groups it can be seen as it assigns two different labels to the pixels (nodes). As the weights of the edges are derived from the smoothness and the data cost functions of the energy function, the minimum cost cut corresponds to the minimum energy.