Contextual Bayesian Inference for Visual Object Tracking and Abnormal Behavior Detection

Philippe Bouttefroy

Contextual Bayesian Inference for Visual Object Tracking and Abnormal Behavior Detection

A thesis submitted in fulfilment of the
requirements for the award of the degree

Doctor of Philosophy

from

The University of Wollongong

by

Philippe Loïc Marie Bouttefroy
Masters of Engineering Studies
(Telecommunications and Computer Science),
Diplôme d'ingénieur

School of Electrical, Computer
and Telecommunications Engineering
2010

To my hair †

Abstract

Visual object tracking has been extensively investigated in the last two decades for its attractiveness and profitability. It remains an active area of research because of the lack of a satisfactory holistic tracking system that can deal with intrinsic and extrinsic distortions. Illumination variations, occlusions, noise and errors in object matching and classification are only a fraction of the problems currently encountered in visual object tracking. The work developed in this thesis integrates contextual information in a Bayesian framework for object tracking and abnormal behavior detection; more precisely, it focuses on the intrinsic characteristics of video signals in conjunction with object behavior to improve tracking outcomes.

The representation of probability density functions is essential for modeling stochastic variables. In particular, parametric modeling is convenient since it makes possible the efficient storage of the representation and the simulation of the underlying stochastic process. The Gaussian mixture model is employed in this thesis to represent the pixel color distribution for segregation of foreground from background. The model adapts quickly to fast changes in illumination and resolves the problem of "pixel saturation" experienced by some existing background subtraction algorithms. The technique leads to better accuracy in the extraction of the foreground for higher-level tasks such as motion estimation.

The solution of the Bayesian inference problem for Markov chains and, in particular, the well-known Kalman and particle filters is also investigated. The integration of contextual inference is of paramount importance in the aforementioned estimators;

it results in object-specific tracking solutions with improved robustness. The vehicle tracking problem is explored in detail. The projective transformation, imposed by the environment configuration, is integrated into the Kalman and particle filters, which yields the "projective Kalman filter" and the "projective particle filer". Extensive experimental results are presented, which demonstrate that the projective Kalman and particle filters improve tracking robustness by reducing tracking drift and errors in the estimated trajectory. The constraint on the known nature of the environment is then relaxed to allow general tracking of pedestrians. A mixture of Gaussian Markov random fields is introduced to learn patterns of motion and model contextual information with particle filtering. Such inference results in an increased tracking robustness to occlusions.

The local modeling with the Markov random fields also provides inference on abnormal behavior detection. Since local patterns are unveiled by the Markov random field mixture, detecting abnormal behavior is reduced to the matching of an object feature vector to the underlying local distribution, whereas the global approach, introducing generalization errors, involves complex, cumbersome and inaccurate decisions. Experimental evaluation on synthetic and real data show superior results in abnormal behavior detection for driving under the influence of alcohol and pedestrians crossing highways.

Résumé

Le suivi d'objets visuel a été un domaine de recherche intense durant ces deux dernières décennies pour son attrait scientifique et sa rentabilité. Il reste un sujet de recherche ouvert de par le manque de système de suivi holistique satisfaisant, prenant en compte les distorsions intrinsèques et extrinsèques. Variations d'éclairement, occlusions, bruits et erreurs dans la correspondance et la classification d'objets ne sont qu'une partie des problèmes actuellement rencontrés en suivi d'objets. Le travail développé dans cette thèse intègre l'information contextuelle dans le cadre Bayesien pour le suivi d'objets et la détection de comportements anormaux. Plus précisément, la recherche porte sur les caractéristiques intrinsèques du signal vidéo en conjonction avec le comportement d'objets dans le but d'améliorer les résultats du suivi.

La représentation de fonctions de densité de probabilité est cruciale pour modéliser les variables aléatoires. En particulier, les modèles paramétriques sont pratiques puisqu'ils permettent un stockage compact de la représentation ainsi que la simulation du processus aléatoire sous-jacent. La mixture de Gaussiennes est utilisée dans cette thèse pour représenter la distribution de couleur d'un pixel dans le but de séparer l'avant-plan de l'arrière-plan. Le modèle s'adapte aux changements rapides d'éclairements et résout le problème de "saturation de pixels" rencontré avec certains algorithmes de soustraction d'arrière-plan. Il résulte de cette technique une meilleure précision lors de l'extraction de l'avant-plan pour des tâches de plus haut niveau telles que l'estimation du mouvement.

La solution au problème d'inférence Bayesienne pour les chaines de Markov, et en particulier, les filtres de Kalman et particulaire, est étudiée. L'intégration d'une inférence contextuelle dans ces estimateurs est primordiale pour améliorer le suivi d'objet. Il en découle des solutions propres à un contexte spécifique. Le problème de suivi de véhicules est également exploré en détails dans cette thèse. La transformation projective, imposée par la configuration de l'environnement, est intégrée dans les filtres de Kalman et particulaire, engendrant le "filtre de Kalman projectif" et le "filtre particulaire projectif". Des résultats expérimentaux exhaustifs sont présentés pour démontrer l'amélioration de la robustesse au suivi par les filtres de Kalman et particulaire projectifs. L'amélioration est caractrisée par la réduction de la dérive du suiveur et la réduction de l'erreur dans l'estimée de la trajectoire. La contrainte sur le caractère connu de l'environnement est ensuite supprimée pour permettre le suivi de piétons. Une mixture de champs aléatoires de Markov Gaussiens est introduite dans l'objectif d'apprendre les motifs de mouvements et de modéliser l'information contextuelle pour le filtrage particulaire. Une augmentation de la robustesse du suivi sous occlusion résulte d'une telle inférence.

La modélisation locale avec les champs aléatoires de Markov fournit également une inférence pour la détection de comportements anormaux. Puisque les motifs locaux sont révélés par la mixture de champs aléatoires de Markov, la détection de comportements anormaux est réduite à l'étude de la correspondance entre le vecteur de caractéristiques et la distribution locale sous-jacente. L'approche globale, quant à elle, introduit des erreurs de généralisation et implique des décisions complexes, peu élégantes et imprécises. L'évaluation expérimentale de la méthode proposée sur des données synthétiques et réelles présente des résultats supérieurs pour la détection des comportements anormaux de conducteurs en état d'ébriété et de piétons traversant les autoroutes.

Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution, to the exception of the University Paris 13 (France) with which a cotutelle agreement (Joint Doctorate) has been signed.

Signed

Philippe Loïc Marie Bouttefroy
21^{st} of January, 2010

Acknowledgments

I would like to express my gratitude to all of those who provided me with the resources to complete my thesis. First, I would like to thank my supervisors, Prof. A. Bouzerdoum and Prof. A. Beghdadi as well as my co-supervisor Dr. Phung, for their insights on my research progresses throughout the thesis. More importantly, I acknowledge their open-mindedness towards postgraduate research that enabled me to explore fields of personal interests rather than being trapped in a predefined path leading to the completion of my thesis. I also owe my genuine appreciation to Prof. A. Bouzerdoum who financially supported me during the first year of this journey, in particular. Second, my deepest expression of gratitude goes to my family. Dad and Mom for their relentless efforts to make me grow as a person, for their support in tough moments and also, for their financial help throughout my education; Aymeric, Marjorie, Séverine and Alice for their brotherhood and for saying the right word at the right moment. The love and care of the family were always shinning in the distance to guide and comfort me along the different steps of the thesis. Third, my postgraduate experience wouldn't have been the same without Weerona College. The distractions and the support of the community, in particular Leanne, the SR team and the residents, most definitely helped me to keep my sanity during the thesis. Fourth, I would like to specially thank two people who are very dear to me: Tracey and Rachel. Tracey, thank you for pushing me back to the top of the roller coaster that the thesis writing stage is. Rachel, thank you for the insightful and non-technical conversations we had during our Friday meetings at the North Wollongong pub and the now famous G&T's on the balcony. Fifth, my thoughts go

to Laëtita who has taught me more than I could ever understand and who pushed me to always aim for excellence by her attitude. I would most certainly not be where I am now had our paths not met. Finally, I am indebted to a number of university groups and members, namely the ICT postgraduate students and Roslyn, for their support and joy.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

ABD Abnormal Behavior Detection

ADABOOST Adaptive Boosting

ANN Artificial Neural Network

AVC Advanced Video Coding

BAC Breath Alcohol Content

CCD Charge-Coupled Device

CMOS Complementary Metal Oxide Semiconductor

CONDENSATION Conditional Density Propagation

DCT Discrete Cosine Transform

DUI Driving Under the Influence

DWT Discrete Wavelet Transform

EKF Extended Kalman Filter

EM Expectation-Maximization

EPF Extended Particle Filter

GMM Gaussian Mixture Model

GMPHDF Gaussian Mixture Probability Hypothesis Density Filter

GMRF Gaussian Markov Random Field

GMRFM Gaussian Markov Random Field Mixture

GMRFMPF Gaussian Markov Random Field Mixture Particle Filter

HMM Hidden Markov Model

JPDAF Joint Probability Data Association Filter

LOG Laplacian Of Gaussians

MAP Maximum A Posteriori

MCM Motion Correspondence Matrix

ML Maximum Likelihood

MLP Multi Layer Perceptron

MMSE Minimum Mean Square Error

MPDA Merged Probabilistic Data Association

MPEG Moving Picture Experts Group

MRF Markov Random Field

MSE Mean Square Error

OOP Object-Oriented Programming

PCA Principle Component Analysis

PCNSA Principal Component Null Space Analysis

pdf probability density function

PF Particle Filter

PHD Probability Hypothesis Density

PKF Projective Kalman Filter

PPF Projective Particle Filter

ROC Receiver Operating Characteristic

SIR Sampling Importance Resampling

SIS Sequential Importance Sampling

SOM Self Organizing Map

SSD Sum of Squared Differences

SVD Singular Value Decomposition

SVM Support Vector Machine

UKF Unscented Kalman Filter

UPF Unscented Particle Filter

UT Unscented Transform

WB White Balance

Chapter 1

Preliminaries

1.1 Introduction

Computer vision has become ubiquitous in recent years for its ability to model human perception. Although there have been fundamental and groundbreaking advances in the field, many problems remain unsolved to date, and computer vision is, more than ever before, an active area of research. The thesis presented here proposes an investigation into the object tracking field which gathers the different techniques developed to mimic the natural process of tracking performed by human beings in their daily life. Visual object tracking is based solely on videos and the environment surrounding the object (called scene). Furthermore, analog videos do not suit the tracking purpose since the process is carried out by digital architectures such as computers or embedded systems. Computer vision therefore disposes of the entire range of numerical tools available for processing digital signals.

This chapter first introduces the reader to basic notions of video signal processing. The representation of videos is described from a signal processing perspective to define the acquisition process as well as the degradations undergone by the signal to form the video. Second, the motivation of the research and the assumptions underlying the framework are presented. Finally, the contributions of the thesis to the visual object tracking field are summarized.

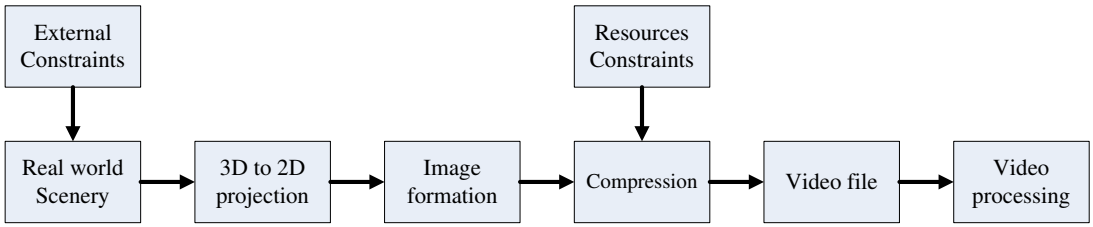


Figure 1.1 Video formation process. The scene and the video undergo a number of transformation altering the information acquired in the video file.

1.2 Representation of Video Signals

Video recordings are used for the purpose of entertainment, learning, training, behavior analysis, remote surgery, virtual reality, surveillance, etc. Although an exhaustive list of usages and applications is impossible to compile, any digital video can be described in an accurate and generic manner from the viewpoint of signal processing: a digital video is a temporal sequence of images which are represented by a matrix of numbers. Analog video is out of the scope of this thesis and digital video will be referred to as video hereafter. Although the thesis is focusing on visual object tracking, it is necessary to introduce the pathway leading to the creation of a video file to understand the object tracking framework and its challenges. Figure 1.1 presents an overview of the video formation process.

1.2.1 Concepts and Notation

A video is a sequence of images, referred to as frames, that generates a coherent animation from the human point of view. Figure 1.2 displays the general structure of a video. The matrix representing a frame in the sequence is of size $W \times H \times N$ where W is the width, or the number of columns in the matrix, H is the height, or number of rows in the matrix, and N is the number of channels. The set of values that characterizes a particular element given by its position in terms of rows and columns is called a *picture element* (or *pixel* for short). The number of channels defines the type of image and, subsequently, the type of video:

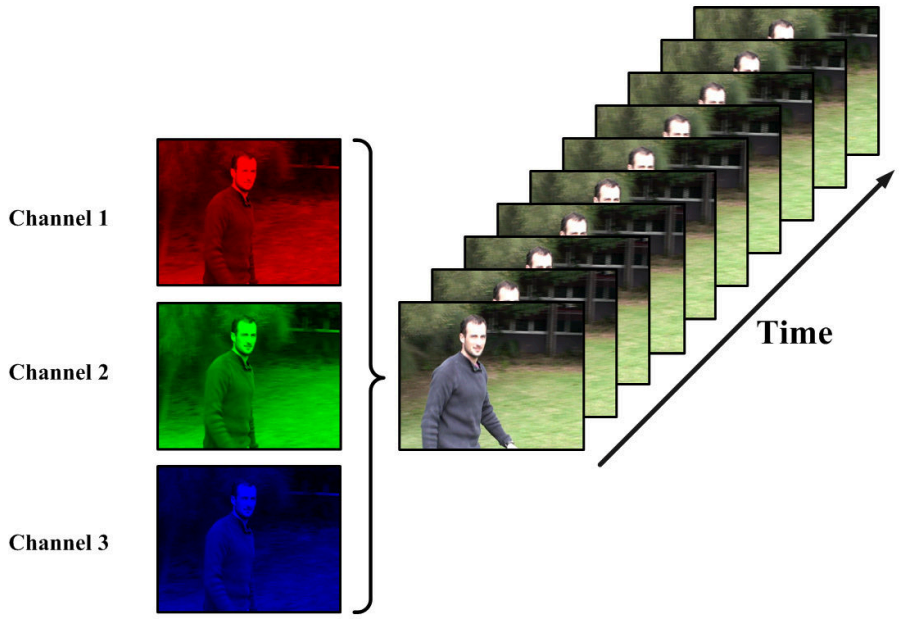
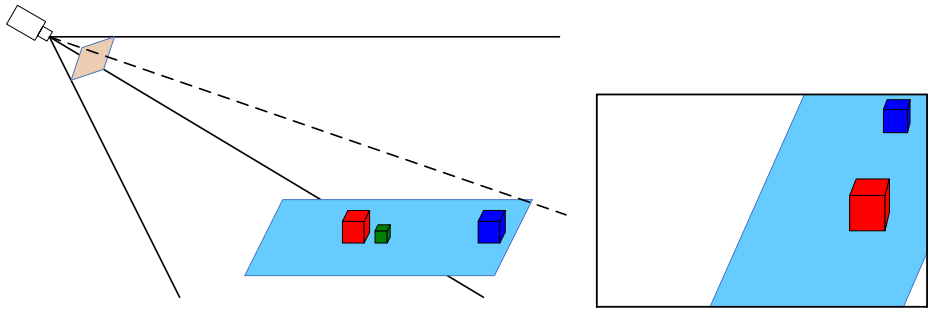


Figure 1.2 Video Structure and representation. A video is decomposed into a sequence of images which are composed of channels. In this example, there are three channels: R, G and B. Each channel is represented by a 2D matrix of numbers.

- $N = 1$ - Grayscale video: This type of video was used in the early days of Television. It is still used in specific applications where the transfer of information is costly (*e.g.* some type of remote videosurveillance).
- $N = 3$ - Color video: This type of video is widely used today because the rendering is close to human perception.
- $N > 3$ - Multispectral video: This type of video embeds information that is not visual such as infrared images. They are found in very specific applications (*e.g.* military day/night vision goggles and medical imaging).

In terms of time frame, a video starts at time T_s and stops at time T_f. However, for convenience in the notation, the time T_s is taken as 0 and T_f is denoted T. For live footage and real-time processing, T is the present time or latest available time. The video represents a sequence of snapshots of the scene captured by the camera at discreet time instants, t. Formally, if the image at time t is denoted $\mathcal{I}_t$, a video



(a) General View (b) Camera plane

Figure 1.3 Scene projection and distortion: display of a projection of the real world (3D) to the camera plane. Projection affects the apparent size of objects: the blue box appears smaller than the red one. Projection also causes occlusion: the green box has disappeared in (b).

$\mathfrak{I}$ is the concatenation of images at different times t such that

$$\mathfrak{I} = \{\mathcal{I}_t : t = 1..T\}\,. \tag{1.1}$$

However, the information regarding an object of interest in a video is not readily accessible due to various degradations during data acquisition.

1.2.2 Video Acquisition

Video acquisition refers to the recording process of a real world scene onto a capturing device. There are three factors to consider during this process: the projection from the 3D world to a 2D camera plane, the influence of the framerate on the apparent trajectory, and the motion of the camera. The type of video capture device is irrelevant from a purely signal processing perspective. The two main technologies available are *Charge-Coupled Device* (CCD) and *Complementary Metal Oxide Semiconductor* (CMOS) . The reader is referred to [40] for a comparative study of the two technologies.

The planar nature of the sensor restricts the capture of real-world scene to its projection on the sensor image plane. It results in the apparent distortion of objects, partial or total occlusion, and scaling problems. These are illustrated in Fig. 1.3. The figure shows that, in this particular setup, the blue box appears smaller than the red box due to the characteristics of the projection imposed by the camera

position. The real world scene undergoes a homographic transform onto the camera plane. Another effect of the transform is the partial or total occlusion of objects. In this scenario, the green box has disappeared from the captured image because the projection support of the green box onto the camera plane is included in the projection support of the red box.

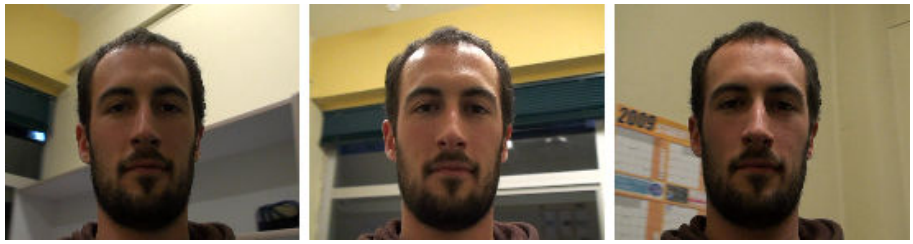
Another factor to account for in the acquisition process is the framerate of the capturing device. Recording devices capture images at discreet times only. The illusion of motion is restored by the brain integrating the images displayed through time. Let Δ_t be the time elapsed between two frames. The framerate is therefore defined as $F_S = 1/\Delta_t$. In terms of signal, this means that any sporadic information occurring between two images is lost. An example of loss of information by inadequate framerate is the stroboscopic effect. In this case, the frequency of the state of an object is equal to the frequency of the image capture: the object seems static when it is in fact dynamic. To a lesser extent, for fast moving objects, a low framerate can create misleading apparent motion of an object (*e.g.* illusion of car wheels spinning backwards in videos).

Finally, the camera observational reference frame plays an important role in the capture of the scene. Without loss of generality, there are two possible types of videos: (i) videos captured with a fixed camera as shown in Fig. 1.4(a) (*e.g.* videosurveillance) and (ii) videos captured with a camera in motion as shown in Fig. 1.4(b) (*e.g.* action shots in movies). A duality regarding the object of interest of a scene is often observed between the two cases: for a fixed camera, the background is static and the foreground is dynamic, whereas for a moving camera, the background is dynamic and the foreground is quasi static.

1.2.3 Information Distortion

Once the signal has arrived to the capturing device or sensor, it is compressed to be efficiently stored on the digital media. At this stage, the signal may be degraded, *i.e.*, some information is lost during compression to produce a data file of acceptable size. Here, we omit the details of video coding because it is out of the scope of this thesis. However, to be able to model the degradations and account for the loss in

(a) Fixed camera

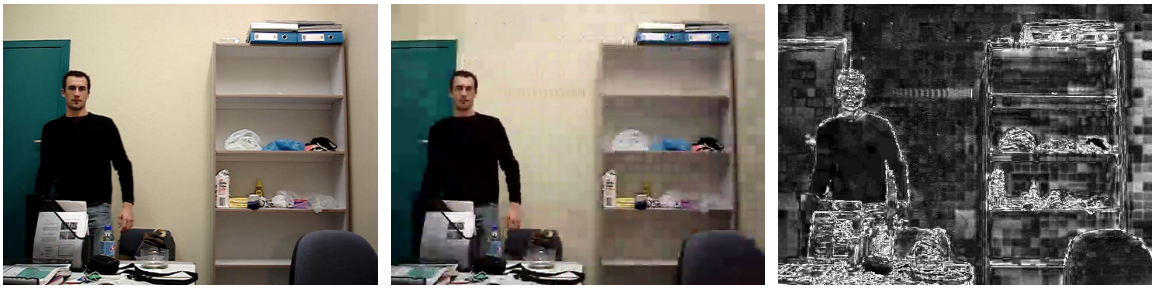


(b) Moving camera

Figure 1.4 Fixed camera versus moving camera. (a) Fixed camera: the background is fixed while the foreground is in motion; (b) Moving camera: the background is in motion while the foreground is fixed.

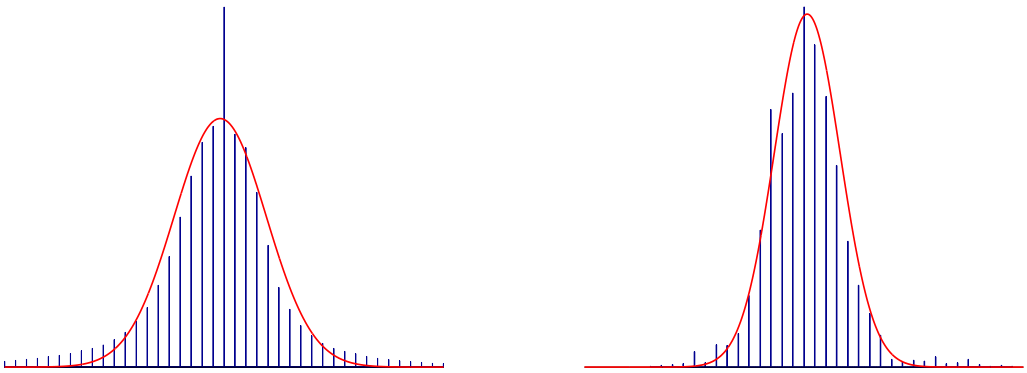
the modeling of the original signal, it is essential to characterize the type of noise introduced in the process. One of the major issues encountered in video compression (*e.g.* MPEG-2) is the so-called blocking artefact inherent to the decomposition of the frame into blocks to perform efficient *Discrete Cosine Transform* (DCT) . Since compression introduces errors in the signal on a block basis, errors can add up at the edges of a block, leading to horizontal or vertical artefacts, typical of a highly compressed video. Here, we propose a very simple yet effective experiment to identify the nature of the noise introduced by compression. A video sequence is captured with low compression ratio (11Mbps) in MPEG-2. Then, it is compressed with a H.264/AVC codec at high compression ratio (128kbps). Figure 1.5 displays a frame of the original and the compressed videos. Figure 1.5(b) presents some artefacts that are not in Fig. 1.5(a). Figure 1.5(c) shows the squared error between the two frames introduced by compression: the error is predominant around the edges.

Now, is it possible to fit a model to the noise distribution? Let us first consider the spatial noise, *i.e.* , the noise that is introduced in a given frame. The histogram rep-



(a) Original frame (b) Compressed frame (c) Mean Square Error

Figure 1.5 Displays of an original video and its compressed version. Some degradations, called artefacts, are present in the compressed video (b). (c) is the error between the original and the compressed video (magnified 5 times for the purpose of display).



(a) Spatial Noise Histogram (b) Temporal Noise Histogram

Figure 1.6 Histogram representations of the spatial and temporal noise generated by compression of a video. (a) The spatial noise is taken over the entire image for frame 210. (b) The temporal noise is for 500 frames and for pixel [35,175].

resentation of the difference between the original frame and the compressed frame is presented in Fig. 1.6(a). It can be inferred from the figure that the noise distribution is near Gaussian. The bin centered on 0 has a higher value because it contains the quantization noise. The same experiment is run to determine the nature of the temporal noise. A histogram representation of the difference between a pixel in the original and in the compressed videos is displayed in Fig. 1.6(b). The difference is taken over 500 frames. The distribution can be considered near Gaussian. The shape of the noise come from compression and, in particular, prediction (Gaussian noise) and quantization (uniform noise).

To sum up, the degraded video $\mathfrak{I}_d$ can be approximated as:

$$\mathfrak{I}_d = \mathfrak{I} + v\,, \tag{1.2}$$

where $v \sim \mathcal{N}(., \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{N}(., \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution (also called Gaussian distribution) with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Because the Gaussian is centered at 0, the mean $\boldsymbol{\mu}$ is equal to 0.

1.2.4 Research Motivation and Assumptions

The visual object tracking field is still an active area of research and optimal solutions are yet to be found. Although simple scenarios are reliably and accurately handled by existing methods, more complex scenarios such as tracking in clutter, high density of similar objects or occlusions remain a challenge. The motivation of this research is to improve the robustness of tracking by integrating contextual, environment-dependent and local information in tracking. Indeed, the techniques proposed to date lack context in the estimation of object tracks due to the use of general, fit-everything models. While they offer the advantage of tracking all type of variables, from vectors representing objects to financial market or weather forecast, they make little use of information characteristic of video processing. Therefore, the investigations conducted in this thesis will be based on the phenomena pertaining to video processing and object characteristics described in this section.

The visual object tracking field starts from the compressed video file and aims to provide robust and efficient tracking of objects. The field of study is extended to the analysis of abnormal behavior. Since visual object tracking encompasses a wide area of techniques, it is necessary to make specific yet weak assumptions on the nature of objects and videos to narrow the scope of the thesis. Videosurveillance, vehicle traffic surveillance and monitoring, in general, fall into the scope of these assumptions, made in line with the observations presented in this section.

Slow object motion Objects in videos have a slow motion compared to the framerate. Misleading apparent motion is discarded with this assumption and all objects in motion are presumed to have a non-null apparent motion. The disambiguation of adjacent object tracking is also ensured.

Gaussian noises Gaussian noise enables the use of tracking techniques based on closed-form derivations providing optimal tracking. The Gaussian context also provides a parametric representation of the noise for statistical analysis;

Fixed camera video Most tracking and pre-processing techniques are based on modeling regions of the scene with pixel per pixel processing. A moving camera does not ensure the mapping of a region to specific pixels. Fixed camera provides the track of an object in a fixed (terrestrial) reference frame;

Small object size The area covered by an object is small compared to the entire image. The thesis aims to track objects with efficient algorithms that can run in real- or near realtime. A large size object involves complex shape modeling that violates this constraint.

1.3 Contributions of the Thesis

The contributions of the thesis are summarized below:

1. *Transversal literature review.* The literature review investigates visual object tracking in a transversal or object-oriented approach. Previous contributions, although very comprehensive, have focused on a top-down organization which does not bring the modularity of the field into light [275]. Chapter 2 identifies and describes modules which can be assembled together for efficient tracking.

2. *Illumination-invariant background subtraction.* A new technique for generating illumination-invariant background with a Gaussian mixture model is presented. The contribution lies in the update of the mixture parameters. While proposed methods use pre-/post-processing, a semi-constrained Gaussian mixture model is implemented in order to detect foreground in environments with fast changes in illumination. The phenomenon of *pixels saturation*, occurring with large and recurrent changes in illumination, is also addressed in Chapter 3.

3. *Projective Kalman filter.* The extended Kalman filter performs the estimation of a feature vector in a Gaussian environment. However, it does not make use

of application-specific information. In Chapter 4, the projective Kalman filter is developed in the framework of vehicle tracking, integrating the projective transformation undergone by the vehicle tracks from the real-world onto the camera plane. It is used in the aforementioned vehicle tracking framework. The projective Kalman filter contributes to tracking drift reduction and provides accurate and robust vehicle tracking.

4. *Projective particle filter.* The particle filter relaxes the constraint on the Gaussian environment imposed by the Kalman filter. However, its accuracy is inversely proportional to the number of particles. The projective particle filter presented in Chapter 5 improves the tracking of vehicles by integrating the projective transformation in the importance density. The contribution of the projective particle filter lies in the reduction of the particle set size to track vehicles.

5. *Contextual Bayesian inference.* Bayesian inference for particle filters is achieved by the importance density. Traditionally, the inference is general to suit a wide range of tracking problems. The introduction of contextual Bayesian inference through the learning of local information with Markov random fields contributes to the state of the art in visual object tracking. Chapter 6 presents the Gaussian Markov random field mixture, which provides contextual Bayesian inference from pedestrian and vehicle tracking. The technique improves the tracking rate and the recovery after occlusion.

6. *Contextual abnormal behavior detection.* Abnormal behavior detection is improved in Chapter 7 with the integration of contextual Bayesian inference. A local approach is proposed which trains the Gaussian Markov random field mixture with a stochastic clustering algorithm. While existing techniques focus on a global approach, leading to complex decisions, the technique developed contributes to abnormal behavior detection by providing simple local decisions. The system outperforms current techniques in terms of abnormal behavior detection accuracy.

1.4 Publications

The research undertaken in this thesis has resulted in the following publications:

- P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Vehicle Tracking by non-Drifting Mean-shift using Projective Kalman Filter," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 61-66, 2008;

- P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Local estimation of displacement density for abnormal behavior detection," in *IEEE Workshop on Machine Learning for Signal Processing*, pp. 386-391, 2008;

- P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Abnormal behavior detection using a multi-modal stochastic learning approach," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 121-126, 2008;

- P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Vehicle Tracking Using Projective Particle Filter," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 7-12, 2009 [**Best student paper award**];

- P. L. M. Bouttefroy, A. Bouzerdoum, A. Beghdadi, and S. L. Phung, "On the analysis of background subtraction techniques using Gaussian mixture models," to appear in *the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010;

- P. L. M. Bouttefroy, A. Bouzerdoum, A. Beghdadi, and S. L. Phung, "Object Tracking using Gaussian Markov Random Field Mixture for Occlusion handling with Particle Filters," submitted to *the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

Chapter 2

Roadmap for the Object Tracking Maze

2.1 Introduction

Object tracking is traditionally presented in a top-down approach starting from object representation. Here, a different architecture is proposed since the object tracking field has recently become extremely complex and ramified. Instead of describing each of the branches individually, we propose to investigate the different modules involved in tracking. The presentation of the literature finds an analogy with Object-Oriented Programming (OOP) : abstraction of the implementation details is omitted to focus on the function of each module. Nevertheless, key references are provided to the reader for further details. This approach offers distinct advantages:

- the broad area of visual object tracking can be presented succinctly with abstraction of complex implementation details;
- the modularity of visual object tracking is enlightened;
- the cumbersome enumeration of different implementation is avoided;
- the transversal approach precludes the redundancy of description imposed by the use of the same technique for a different purpose.

However, background and technical analysis of particular fields of interest require further investigation that will be provided in the relevant chapters.

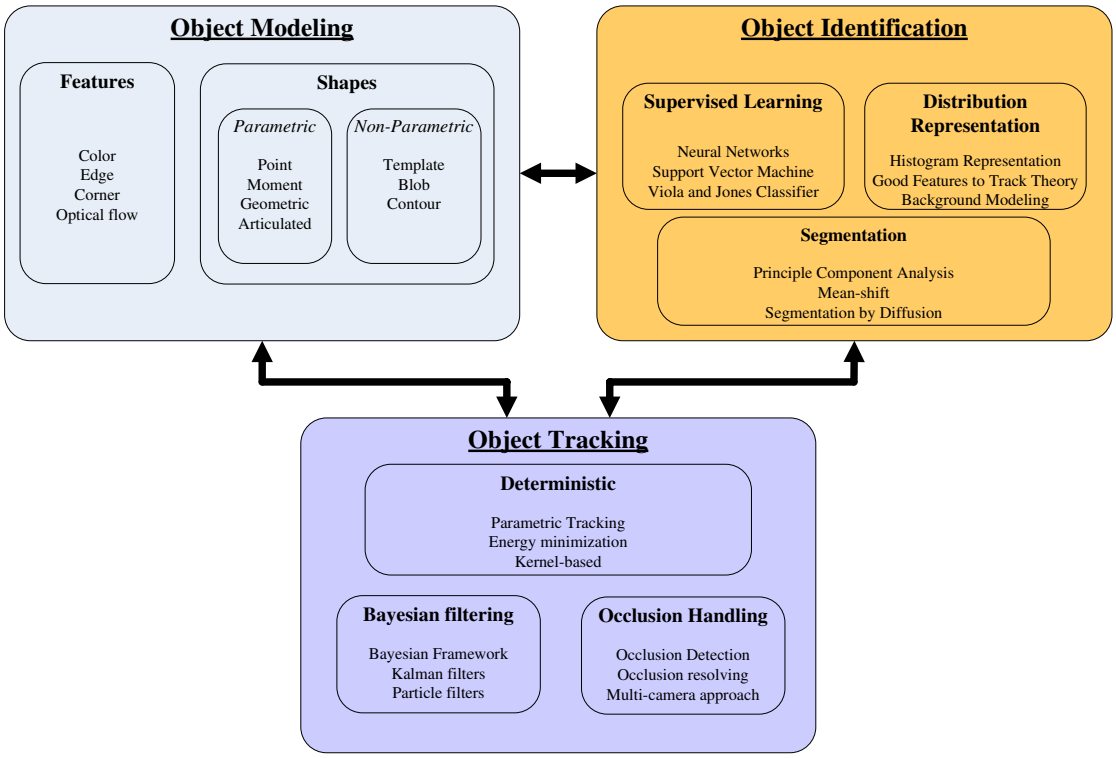


Figure 2.1 Functional diagram of visual object tracking, organized in modules.

The review of the literature is based on the diagram displayed in Fig. 2.1. The visual object tracking field relies on three modules that interact with each other to perform robust object tracking. First, Section 2.2 reviews the different techniques used for object modeling and further details object features and representations. Second, Section 2.3 presents object detection and, in particular, supervised learning, distribution representations and segmentation. Third, Section 2.4 explores object tracking techniques and occlusion handling. The reader is referred to the comprehensive survey on visual object tracking by Yilmaz *et al.* to complement the literature review with a traditional top-down approach of the field [275].

2.2 Object Modeling

Object modeling plays a crucial role in visual tracking because it characterizes an object of interest. Only the feature defined by the object model is used to maintain

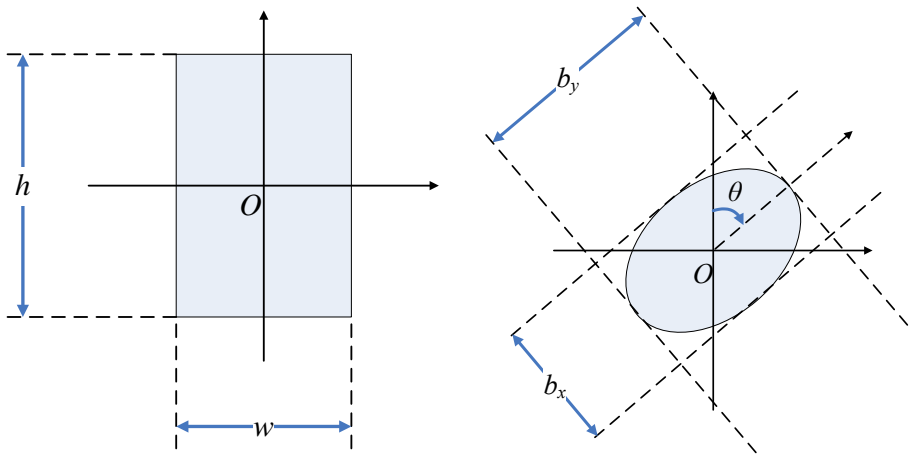
the estimate of the track. Object modeling therefore consists of two attributes: the representation of the object, which delineates its span in the frame, and the features, which characterize it. Consequently, a poor choice of object model inevitably leads to poor tracking. The range of object representations encompasses various types of models and is application dependent. Some applications only require a simple model, while others require accurate and complex object models to achieve tracking. This section presents the various model representations, in particular, parametric and non-parametric shape representations and features used in tracking.

2.2.1 Parametric Representations

The parametric representation is simple because it characterizes the object with basic geometric shapes defined by a limited number of parameters. Various signal processing operations such as transforms, estimations or learning can be directly applied to parameters in order to achieve tracking. Parametric representations are desirable when more accurate information is not available or too time-consuming to obtain, for instance. This subsection reviews the point representation, conventional shape representations such as rectangles, ellipses and their trivial form, the square and the circle, respectively. Finally, articulated shapes are presented.

Point Representation

In visual object tracking, the trivial shape is the point. An object is represented with a pixel location representing either some statistics on the object, such as the centroid, or a particular characteristic of interest. Point representation has been used in a plethora of applications due to its processing simplicity and the ease of point manipulation with complex algorithms. For instance, it has been used for point tracking in radar imagery [185], distributed point tracking [133] or for Monte Carlo techniques where the number of samples prohibits heavy calculations [7, 91, 143]. Point tracking also alleviates the uncertainty regarding the position of the object of interest in the frame since it is based on a single point. It can be complemented with various order moments describing the distribution of the shape, such as the variance of pixels in the object of interest [47, 240, 268]. Points have also been used to generate heuristics on some characteristics of the object. They are also used in



(a) Rectangle (b) Ellipse

Figure 2.2 Example of rectangular and elliptic shapes with their nomenclature. The shapes are defined by a reduced number of parameters: 3 for the rectangle; 4 for the ellipse.

the calculation of optical flow: due to the large number of vectors to estimate, only the point representation can be afforded [41, 140, 244].

Conventional Shapes

Notwithstanding the aforementioned attractive properties, the point representation of an object can lead to simplistic models that do not grasp the entire dynamics of the object. For instance, rotation is not catered for with point representation. More advanced parametric shapes are, therefore, necessary to address these types of problems. Conventional shapes can be of any form as long as their representation is parametric and compact. In practice, almost every tracking system based on conventional shapes is designed around two representations: rectangular and elliptic.

Figure 2.2 displays the rectangular and elliptic representations. Rectangles are entirely defined by their center (O), also called origin, and the height h and width w. Trivially, when $h = w$, the rectangle becomes a square. The assumption further reduces the number of parameters. The rectangle representation is ubiquitous in geometric object tracking such as cars [172, 228] or in low-distortion object tracking such as people [61, 246, 274]. Traditionally, the object representation with a rectangle does not integrate a tilting parameter to enable rotation; the width and the

height of the rectangle are set along the image axis. The ellipse is usually preferred when rotation is required [42]. An ellipse is defined by its center point, (O), the large and small axes, b_x and b_y, and the angle of rotation, θ. The four shape parameters enable the ellipse to fit most object shapes and, in particular, non-geometric objects for which the coarse outline provided by rectangles is not suitable. Indeed, the projection of compact objects onto the camera plane can be assimilated to an ellipsoidal blob. The ellipse offers the advantage of "rounding" the edges compared to the rectangle when the object does not have sharp edges [43]. Finally, the ellipse is the contour of equiprobability for a 2D Gaussian distribution. This property is exploited to generate samples from Gaussian distributions for sequential Monte Carlo methods in a Gaussian environment. For instance, the covariance matrix can be defined proportionally to the axes of the ellipse. Therefore, the probability distribution of the target dispersion is conveniently modeled by the shape representation.

Articulated Shapes

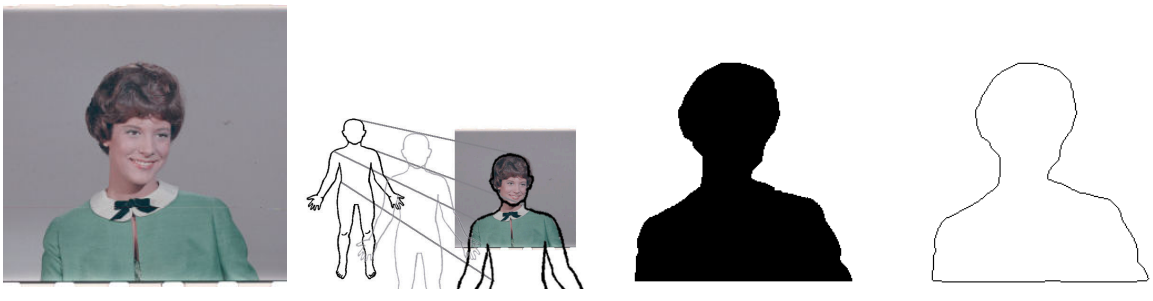
Articulated shapes are employed for tracking if different portions of the object of interest are to be described individually (*e.g.*, legs, arms and head). For instance, Ramanan and Forsyth developed an articulated shape model to describe the body configuration and disambiguate overlapping tracks [202]. Articulated shapes require the definition of interactions between the different parts of the object and the learning of appearance from examples, resulting in a significant computational load for tracking. However, it gives insight into the characterization of the gait, which is essential for certain types of abnormal behavior detection. For instance, the video-surveillance software "W^4" marks the position of the different body limbs to analyze the behavior of people [99]. Articulated shapes are therefore composed of a system of basic conventional shapes such as rectangles, circles and ellipses tied up with spatial and kinematic dependencies. As mentioned before, the main drawback of articulated shapes is the inherent computational load that makes any stochastic tracking algorithm prohibited. The processing time is multiplied by the number of elementary shapes, in addition to the calculations due to the dependency requirements. This type of representation is out of the scope of the thesis since it violates the requirement of near-real time tracking. Articulated shapes also provide little

improvement, if any, to tracking since objects are assumed to be small in the image; delineating the different parts of the object is therefore cumbersome.

Finally and for the sake of completeness, it is important to mention the object skeleton that is presented by some authors (*e.g.* [275]) as a technique for object representation. Skeletons model a set of dependencies between different parts of the object; they do not represent nor delineate the object. We define the skeleton as a set of articulations within an object that describes the dependencies and defines constraints between the representation of the parts. Skeletons are therefore a tool for describing articulated objects.

2.2.2 Non-parametric Representations

One of the major shortcomings of parametric representations is the accuracy of the object spatial delineation. Indeed, the trade-off for limiting the number of parameters describing the shape is the lack of adaptability to awkward, or non-conventional, shapes. Non-parametric representations address this shortcoming with a pixel by pixel delineation at the expense of an exhaustive description of the object. It is worthwhile noting here that we define non-parametric representations as representations that are not purely parametric. In this sense, semi-parametric representations are included in non-parametric representations. Figure 2.3 illustrates the three main types of non-parametric representation described in this subsection: templates, blobs and contours.



(a) Original Image (b) Template (c) Blob (d) Contour

Figure 2.3 Non-parametric representations of a person. (a) The original image with the object of interest. (b)–(d) The three techniques provide a delineation of the object superior to parametric representations.

Templates

A comprehensive description of the use of templates in computer vision can be found in [32]. Templates aim to represent objects with a set of predefined models. In that sense, templates can be categorized as semi-parametric representations. The predefined models are *a priori* non-parametric and can be of arbitrary form, providing single or multiple views of the object of interest. However, the matching of the model is performed by projection, distortion, scaling, etc., which are parametric transforms. One of the main tasks concerning templates is to maintain the set of models to minimize their number and maximize their relevance to the scene. First, if the appearance of the object is assumed to be static, the set of templates can be generated at initialization and updates are not necessary [17]. If the object changes appearance but is limited to a pre-defined range, the set of templates can be learnt off-line [129], thereby limiting its size. Another approach is on-line update and pruning of the set throughout time [145]. Templates are simple non-parametric representations to manipulate due to the restriction in the set of models and the parametrization of transforms used for matching.

Blobs and Silhouettes

When learning or updating is not possible because there is no pattern in the representation, an exhaustive description of the spatial delineation of the object cannot be avoided. Blobs, also called silhouettes, are used for this purpose. A blob is merely defined in the general context as "a small lump, drop, splotch, or daub" [1]. In computer vision, a blob is a dense, non-disjoint, binary mask that represents an object of interest. Blobs are of particular importance for pixel-wise processing. For instance, background subtraction provides blobs identifying the foreground or the moving objects in a scene [75, 279, 283]. Blobs can also result from classification such as skin segmentation [115, 196] or color segmentation [53, 54].

Contours and Splines

Contours provide a convenient non-parametric trade-off between an exhaustive description of the object and storage requirements. Instead of storing the entire silhouette, contours only describe the edges enclosing the object. The gain in storage

is counter-balanced by an increase in processing when retrieving the entire blob. It is also necessary that the contour be closed in order to avoid ambiguity of reconstruction, although some closure [236, 273, 280] and tracking [205] techniques handle small breaks in the continuity of the shape. Despite these requirements, contours are widely used because a tracking framework based on splines has been developed [244, 272]. Subsection 2.4.1 provides more insight into contour tracking techniques. Here, we limit the investigation to spline modeling, a technique for contour parametrization. Splines are a piecewise function of polynomials with smoothness constraints. They were introduced by Schoenberg in 1946 [219]. The description of splines below is based on [245]. A spline s modeling the contour $C = \{k_1, .., k_n\}$ is uniquely described as

$$s(x) = \sum_{k \in C} c(k)\beta(x - k)\,, \tag{2.1}$$

where β is a *B-spline* function and $c(k)$ are estimated coefficients. The objective of contour tracking is the estimation of the parameters $c(k)$ and the spline basis. Applications of active contours for object tracking are varied, from tracking with optical flow [244] or through severe occlusion [87, 272, 276] to Bayesian estimation [206] or Gaussian mixture assisted segmentation [260].

2.2.3 Object Features

The term object feature encompasses every data that is employed to characterize and discriminate an object from the rest of the image, including other objects. The ideal feature for object tracking is an invariant of the object, *i.e.*, at least robust to any type of transform, any change of illumination, any degradation. This feature, if existent, has not been found yet. This subsection presents features characterizing the object delineated by the object representation namely, color, edges, corners and optical flow.

Colors Representation

Colors are the most intuitive features to describe an object since they are the most obvious one for the human eye; they have been the primary source of identification and discrimination. However, the perception of color by the human eye differs from

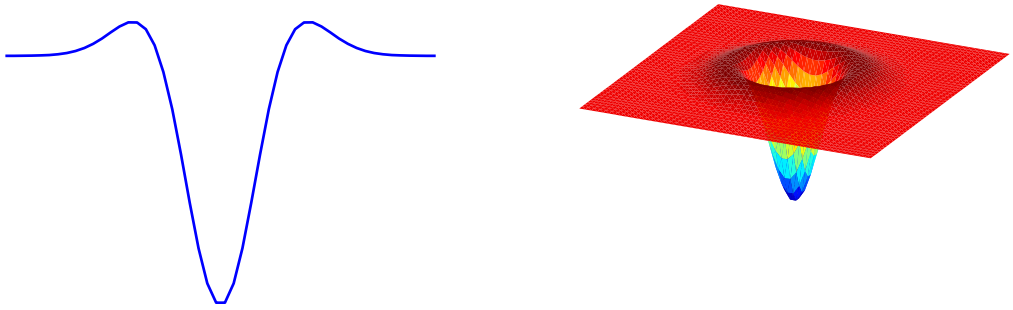
the "perception" by a computer. The integration of color perception models for video coding has been a field of research for many years [19,210]. One of the focuses of attention is the transformation of the RGB channels into a different color space. A practical description of the most common color spaces can be found in [113]. Many transformations have been investigated all sharing the same objective: separate the information into perceptually relevant channels. The Yxx group, encompassing YUV, YIQ, and YCbCr, aims to isolate the luma component in the signal. The Hxx group, including HSI, HLS and HSV, focuses on Hue and Saturation. Other color spaces offer the advantage of representing the color in a perceptually linear space (*e.g.* CIELUV, CIELAB), at the cost of non-linear transformations. While humans naturally adapt to changes in illumination when tracking an object, this is a major challenge in visual object tracking. Color space transforms aim to address this issue by isolating the illumination component and processing the illumination-independent components only. Color tracking is ubiquitous in different areas of tracking. Some examples are kernel-based object tracking [52,56], Bayesian filtering [94,96,150], Skin-color tracking [31] and texture tracking [135].

Edges

Edges, although less intuitive features than color, are widely used because they are insensitive to illumination changes. Numerous filter masks have been designed to detect edges in an image. The reader is directed to the review on edge detection by Ziou and Tabbone in [297]. Edges can be detected with a bank of high-pass filters, in horizontal and vertical directions, expressed as

$$\begin{bmatrix} -1 & a & -1 \\ 0 & 0 & 0 \\ -1 & a & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & -1 \\ a & 0 & a \\ -1 & 0 & -1 \end{bmatrix} \tag{2.2}$$

where a is a positive real number. The Prewitt filters ($a = 1$) and the Sobel filters ($a = 2$) are two examples of these filters. However, they are sensitive to noise. The Laplacian of Gaussians (LOG) has been introduced to increase the robustness to noise with the smoothing property of the Gaussian: the Laplacian operator, which is the second partial derivative in horizontal and vertical directions (mixed derivative being equal to 0), is applied to the Gaussian function [231].



(a) 1D LOG profile (b) 2D LOG profile

Figure 2.4 Profile of the 1D and 2D Laplacian of Gaussians.

Figure 2.4 presents the profile of the filter. Canny also proposed a technique to improve the edge detection by suppressing the non-maximum edges [38]. A hysteresis thresholding is performed on the edges to this effect. Bowyer *et al.* evaluated the performances of 11 edge detectors (incl. Sobel and Canny) on a set of images and their ground truth via ROC curves [29]. They concluded that complex edge detection shows little improvement compared to the Canny edge detector.

To a higher level of abstraction, it is of interest to detect edges that are correlated together to extract meaning in the image. The Hough transform [72] performs a search of linear edges at every edge pixels location (x, y) by fitting a line with the affine equation $y = mx + b$. For computation purposes, the polar representation is adopted. For each edge pixel, the distance ρ and the angle θ of the intersection between the line passing through the pixel and the perpendicular passing through the origin are recorded. The two variables ρ and θ are quantized and an accumulator counts the number of occurrences for each pair. Lines in the image are thus detected by selecting the largest accumulator values for $\{\rho, \theta\}$. Ballard later introduced the Generalized Hough Transform detecting any shape which can be parameterized [12].

Corners and Salient Features

Corners, and salient features in general, are simple yet robust object feature. This subsection focuses on the three main techniques for identifying corners: the Moravec

[177], the Harris and Stephens [100] and the Trajkovic and Hedley [242] corner detectors.

The Moravec algorithm is a basic corner detector that computes the intensity sum of squared differences (SSD) between two sub-images to find the degree of similarity. For each pixel at location (x, y), the SSD is calculated between a sub-image and its shifted version such that:

$$SSD(x,y) = \sum_{(\delta_x,\delta_y)\in\mathcal{D}} \left(I(x,y) - I(x+\delta_x, y+\delta_y)\right)^2 , \tag{2.3}$$

where $\mathcal{D}$ is a domain to be defined. The SSD is small for homogeneous regions and large for heterogeneous regions. In this sense, Moravec corner detector measures the dissimilarity or the *cornerness* of a pixel location. A feature with high dissimilarity is a good feature to track since the tracker will be less distracted by the neighboring pixels.

The Harris and Stephens corner detector is based on the calculations of a weighted SSD. Harris and Stephens proposed to linearize the SSD using the first order Taylor series expansion to allow a matrix formulation of the problem. Equation (2.3) is therefore rewritten as

$$SSD(x,y) = (x,y)A(x,y)^T , \tag{2.4}$$

with

$$A = \sum_{(\delta_x,\delta_y)} w(\delta_x,\delta_y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_yI_x & I_y^2 \end{bmatrix} . \tag{2.5}$$

Finally, the magnitudes of the eigenvalues for the matrix A are analyzed. Large values mean that the pixel is a feature of interest because it expresses an important dissimilarity of the pixel feature with the neighboring ones, hence saliency. The Harris and Stephens algorithm is found in different applications where robust feature tracking is necessary, *e.g.*, optical flow [97, 156].

A different approach, conserving the geometric structure of the neighborhood, has been proposed by Trajkovic and Hedley. They examine the dissimilarity of radially opposed pixels on a circle $\mathcal{C}$ with regards to the pixel of interest as the minimum of the sum of the distances to the feature of interest. This is expressed as:

$$C(x,y) = \min\left((Ip - Ic)^2 + (Ip' - Ic)^2\right) , \tag{2.6}$$

where $p \in \mathcal{C}$ and p' is the pixel diametrally opposed. This technique is very fast compared to others and provides directionality of the corner.

Optical Flow

Finally, on object can be modeled by its optical flow, or loosely speaking, by its internal and external apparent motion. We chose to include optical flow as a feature rather than a detection or tracking technique, since optical flow provides information on the characteristics of an object independently of the calculation method. A review of optical flow techniques is available in [14]. Many techniques have been proposed to estimate the optical flow such as phase correlation [102], energy-based techniques or block-matching (used in video standards, *e.g.*, MPEG2 and H.264). However, differential methods are the most employed techniques due to their accuracy and robustness. Without loss of generality, differential methods estimate the optical flow under constant illumination assumption. If the intensity of a pixel at position (x, y) and time t is denoted $I(x, y, t)$, the constraint on illumination is written as

$$I(x, y, t) = I(x + \delta_x, y + \delta_y, t + \delta_t)\,, \tag{2.7}$$

where δ_x, δ_y and δ_t are variations in x, y and t, respectively. Assuming that the variations are small and developing Eq. (2.7) in Taylor series yields

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0\,. \tag{2.8}$$

This problem is ill-posed since two variables V_x and V_y are to be estimated with one equation. An additional constraint would therefore determine the problem. Horn and Schunck used a global smoothness condition [105], and Lucas and Kanade introduced a constraint on the velocity in the neighborhood of the point of interest to find the solution to Eq. (2.8) [159]. Nagel was the first to include second order derivative constraints on the vector flow [181].

2.2.4 Summary of Object Modeling

An object can be represented by different techniques, from a point to non-parametric representation, depending on: (1) its shape and complexity; (2) the requirement of the application; and (3) the system resources. The framework of the thesis as well

as the assumptions articulated in Chapter 1 limit the shape representation to point or conventional parametric shape representations such as rectangles or ellipses. In particular, the assumption pertaining to the size of objects render articulated shapes representation cumbersome and unreliable. The second attribute of object modeling is the discrimination of the object itself. Features model the object of interest and differentiate it from others in the image. Features encompass colors, edges, corners and optical flows.

2.3 Object Identification

Object identification, also called object detection, is a preliminary step towards tracking; the object of interest needs to be identified in the frame before estimation of its characteristics can be performed. Object identification can either provide the initialization for a tracking algorithm only or be integrated into the tracking algorithm to provide object identification. Detection is based on object modeling and is therefore dependent on the feature selection. We investigate in this section the different techniques employed for object identification, namely, supervised learning, distribution representation and segmentation.

2.3.1 Object Detection using Supervised Learning

Supervised learning techniques aim to learn complex patterns from a set of exemplars for which the class label is given (*e.g.*, face/non-face classes). Learning provides high-level decisions from the available data based on the analysis of low-level, simple elementary features. Several theses, books and journal articles are entirely dedicated to supervised learning techniques [22,215,107,199]. This subsection provides a short introduction to artificial neural networks, support vector machines and adaptive boosting, the main algorithms used for object detection nowadays.

Artificial Neural Networks

Artificial Neural Networks (ANNs) for pattern recognition has started with the invention of the perceptron in 1957 by Rosenblatt [214]. ANNs can be decomposed

into a structure composed of atomic elements, the neuron, and its associated activation function which can be of different forms: step function, piecewise linear, sigmoid, radial basis function (*e.g.*, Gaussian), shunting inhibitory [27], etc. The Multi Layer Perceptron (MLP) is the basic ANN. In object recognition, the input vector is a set of features. The learning phase aims to teach the desired behavior to the ANNs using a supervised learning algorithm. Traditionally, the minimization of the empirical risk is used in the training process. For sample n in the training dataset, let us denote the desired output $d(n)$ of the ANN to a given input $x(n)$. If the actual output is $y(n)$, the empirical risk is expressed as

$$R(y) = \sum_{n=1}^{N} \phi\left(y(n) - d(n)\right) p(x(n)), \tag{2.9}$$

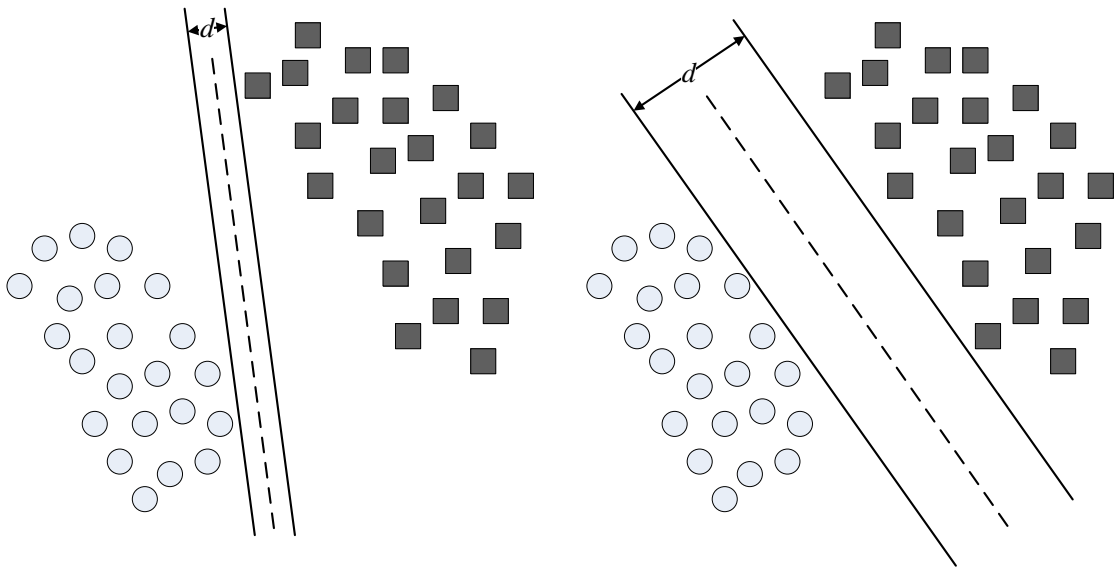
where $\phi(.)$ is a cost function and $p(x(n))$ is the probability density function of $x(n)$. The minimization of the empirical risk $R(n)$ is achieved through the adjustment of the set of weights in the neural network. Empirical risk minimization has, as its objective, the convergence of the output y to the desired output d via minimization of the cost function $R(y)$.

Artificial neural networks are found in a wide variety of applications from object detection, such as faces [160,215] and pedestrians [152], to vehicles [262,284] or skin detection [45, 24]. Also, different types of neural networks exist, depending on the type of connections such as recurrent networks (*e.g.*, Hopfield networks [104]), the choice of activation functions (*e.g.*, Radial Basis Function networks) or dimension of the input (convolutional networks).

Support Vector Machines

Contrary to artificial neural networks, support vector machines (SVMs) do not minimize the cost $R(y)$ but minimize the structural risk. In a 2-class problem, this is equivalent to maximizing the distance between the two hyperplanes lying between the two classes as shown on Fig. 2.5.

Support vector machine provides a subset of samples from each class, called support vectors, that describes the separating hyperplanes. Intuitively, those are the vectors closest to the boundary separating two classes, the other vectors can be discarded.



(a) Non-maximum (b) Maximum

Figure 2.5 Maximization of the distance d between two hyperplanes lying between two classes.

It can be shown that training an SVM is equivalent to solving a linear constrained quadratic problem [190]. The reader is referred to [258] for a comprehensive introduction on SVMs and to [34] for a practical tutorial on SVM implementation. Support vector machines have been successfully applied to object detection with infrared cameras [217, 230], pedestrian [5], eyes [141] and moving object [294].

Viola and Jones Classifier

The Viola and Jones classifier is presented herein. The inherent concepts of integral image and adaptive boosting are also described. The reader is referred to [256] for more details.

The Viola and Jones technique is similar to the summed-area table introduced by Crow for texture mapping [58]. The integral image ii is an image in which each pixel represents the sum of the pixel values that are in the upper top-left rectangle of a feature image f[1]. It can be expressed as:

$$ii(x, y) = \sum_{i=1}^{x} \sum_{j=1}^{y} f(i, j) \,. \tag{2.10}$$

[1]Images and frames are usually indexed starting from the top-left corner

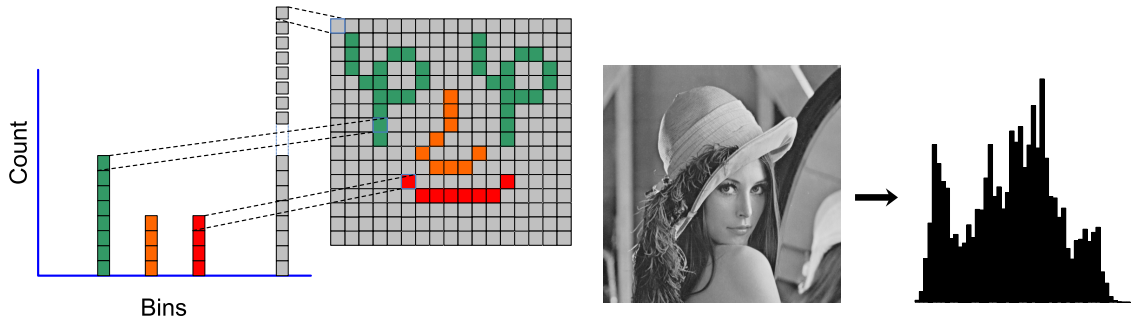
With this technique, the sum of pixels in a rectangular area is computed in constant time through the integral image. Sum of features inside a rectangular area is therefore performed with 3 additions/subtractions once the integral image has been formed. The features of the rectangular area are then fed into weak classifiers selected by adaptive boosting (ADABOOST) [81] to create a strong classifier. ADABOOST is a meta-classifier in this sense since the structure of the weak classifiers is irrelevant. Weak classifiers can be perceptron, dot products, ANNs, SVMs, etc. The idea underlying ADABOOST is to test all the weak classifiers for different features and perform a weighted average of the classifiers providing the lowest classification error that defines the final classification. In their seminal paper [81], Freund and Schapire compared ADABOOST to betting on a pool of horses to maximize gains in a race.

The Viola and Jones classifier has been extensively employed for its ability to detect objects of different natures, from detection of facial expressions [64], hand [164] or pedestrian [122, 257] to detection of vehicles with triangular features [101]. The technique is also implemented for crater detection in geophysics [168]. Beyond the Viola and Jones classifier, ADABOOST has been used with color features for face detection [282] or edge density for pedestrian detection [195].

2.3.2 Distribution Representation for Object Detection

Distribution representation is one of the cornerstones in robust object tracking. A convenient and discriminative representation of an object is the distribution of its features. If an object of interest is known by its feature distribution, implicit detection can be performed by distribution matching in the frame. Two different types of distribution representations exist: parametric and non-parametric. The first one assumes a pre-set functional to model the distribution, *e.g.*, Gaussian mixture models, whilst the second one relaxes this constraint at the expense of computation load. The different techniques pertaining to the modeling, the comparison and the degree of discrimination of distributions are presented hereafter. This includes object detection via histograms, including the Bhattacharyya measure and the "good feature" theory, and object detection by background subtraction.

Histogram representation for Object Detection



(a) Synthetic image (b) Lena

Figure 2.6 Color histogram representation: the original image and the histogram representation. (a) Histogram representation on a synthetic image. Each bar of the histogram represents the proportion of the feature space falling into the bin width. (b) Original and histogram of gray scale Lena.

The histogram is a non-parametric representation of the features, sampling the feature space in m bins. Histograms can model the distribution of object features such as colors, edges, corners, vector flows, and so on. Figure 2.6 displays examples of color histograms. Let us now assume that a prior model of the object feature $\mathbf{q}$, also called the target, is known. A candidate histogram $\mathbf{p}(\mathbf{s})$ can be defined by the representation of the features in a patch centered on $\mathbf{s}$. To detect the object in the image, the minimization of a simple distance measure between the target histogram $\mathbf{q}$ and a candidate histogram $\mathbf{p}(\mathbf{s})$ can be performed. There are many measures that estimate the distance between two histograms [248]. The Bhattacharyya measure, traditionally employed due to its simplicity and good results, is expressed as follows:

$$\rho(\mathbf{s}) = \sum_{u=1}^{m} \sqrt{\mathbf{p}_u(\mathbf{s})\mathbf{q}_u}\,. \tag{2.11}$$

Trivially, the position of the object of interest is at $\mathbf{s}_O = \underset{\mathbf{s}}{\operatorname{argmin}}\, \rho(\mathbf{s})$. Histogram representation is seldom employed alone but usually in conjunction with a tracking algorithms to reduce the search of the object of interest. However, histograms have also been used for object detection (and subsequently, tracking). Bradski developed the camshift algorithm that finds the position of the object $\mathbf{s}_O$ of interest with a 1-D histogram based on the hue component [30]. Comaniciu *et al.* and,

later, Han *et al.* used histograms to segregate the object in an image and perform tracking [56, 94, 96]. Birchfield and Rangarajan proposed to incorporate the mean and covariance of the pixel position into the histogram for more robust tracking [20]. Finally, Shen *et al.* used color histogram and annealing to detect the object [225, 226, 224].

Good Features for Tracking

The problem of selecting the best features to model the appearance of an object is explored here. Indeed, the detection of the object and the selection of the features are concomitant; if the features are not unique and do not characterize the object, the robustness of the tracker is affected. "Good features to track" is the terminology used to define discrimination in visual object tracking. Shi and Tomasi proposed a qualitative analysis of features through *affine motion fields* and *pure translation models* [227]. However, the dissimilarity measures introduced do not set a clear framework to track the so-called good features.

An alternative to these dissimilarity measures is the condition number that defines whether a problem is numerically well conditioned, *i.e.*, a small change in the input data would lead to a small change in the output, or, on the other hand, ill conditioned, *i.e.*, a small change in the input data would lead to a large change in the output. This numerical analysis as been conducted for histogram representations with the Bhattacharyya measure $\rho(\mathbf{s})$. It has been shown that maximizing the Bhattacharyya measure is equivalent to minimizing the Matusita metric $O(\mathbf{s}) = ||\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{s})}||^2$ [92]. Therefore, if a correction $\Delta\mathbf{s}$ is introduced to maximize $\rho(\mathbf{s})$, the Matusita metric should be minimized. Assuming that the feature representation of an object is smoothed with a kernel $\mathbf{K}$, the first order Taylor expansion of $\sqrt{\mathbf{p}(\mathbf{s}+\Delta\mathbf{s})}$ is

$$\sqrt{\mathbf{p}(\mathbf{s}+\Delta\mathbf{s})} = \sqrt{\mathbf{p}(\mathbf{s})} + \frac{1}{2}\mathrm{diag}(\mathbf{p}(\mathbf{s}))^{-\frac{1}{2}}\mathbf{U}^t\mathbf{J}_{\mathbf{K}}(\mathbf{s})\Delta\mathbf{s}\,, \tag{2.12}$$

where $\mathrm{diag}(\mathbf{p}(\mathbf{s}))$ is a square matrix with $\mathbf{p}(\mathbf{s})$ on its diagonal, $\mathbf{U}$ is the catenation

of the histogram intervals and

$$\mathbf{J_K}(\mathbf{s}) = \begin{bmatrix} \nabla_{\mathbf{s}} K(\mathbf{x_1} - \mathbf{s}) \\ \vdots \\ \nabla_{\mathbf{s}} K(\mathbf{x_n} - \mathbf{s}) \end{bmatrix} \quad , \tag{2.13}$$

with $\{\mathbf{x_1}, .., \mathbf{x_n}\}$ the features in the kernel support. Minimizing the Matusita metric after integration of the Taylor expansion requires a full rank on the matrix $\mathbf{J_U} = \text{diag}(\mathbf{p})^{-\frac{1}{2}}\mathbf{U}^t\mathbf{J_K}$. The full derivation can be found in [93]. In terms of features, the condition number offers a numerical, and therefore quantitative, method to describe the "goodness" of the tracked features. The technique has been implemented in [93, 78, 77]. The condition number also appears in [134] for performing the detection of salient points for image registration. Finally, Dewan and Hager proposed a criterion for determining the optimal kernel for tracking [68]. Although achieving good results in the feature selection, all the aforementioned techniques require significant calculation time.

Background Modeling

Background modeling is a technique used in computer vision to extract relevant foreground motion from the video sequence. In the early days of computer vision, Jain and Nagel proposed a frame differencing algorithm subtracting two consecutive images from one another, thus canceling static areas in the scene [114]. Since then, the research effort has focused on improving the modeling of the background. Without loss of generality, the background is defined as the most probable surface(s) in the scene at a given location, whether the probability is based on the average time of presence, clues of the surface to be an irrelevant object with regards to the processing task, and so on. Consequently, distribution estimation techniques are ubiquitous in background modeling. However, they restrict background subtraction to fixed cameras since they are pixel or region-based techniques. For non-parametric models, kernel density estimation methods [176, 167] are traditionally implemented with Bayesian probabilities [223, 266]. It is worthwhile noting that some other techniques are also available such as the eigenbackground proposed by Oliver *et al.* which processes the entire image as a vector and performs Principle Component Analysis (PCA) over time to retain the K first vectors as background

models [189]. Even though the use of non-parametric techniques can model a wider range of distributions, they are prohibitively costly in terms of computation time for most applications. Parametric models are preferred as the processing time can be controlled through the adjustment of the model complexity.

Parametric techniques aim to estimate the pixel distribution over time via the calculation of a limited set of parameters. Wren *et al.* developed a unimodal running Gaussian to model the color distribution [267]. Stauffer and Grimson proposed in their seminal paper a K-Gaussian mixture model to represent the distribution of a pixel over time and a classification criterion differentiating between foreground and background based on prior knowledge on the proportion of background over time [233, 234]. This technique updates the mixture model with first order recursive difference equation for the sake of reduction in computation. Such a method integrates recurrent motion, *e.g.*, branch swaying in the wind, thanks to the multimodal representation. The expectation-maximization (EM) procedure can replace the first order filter to provide a maximum likelihood estimate of the means and variances [138]. However, this technique is costly in terms of memory storage and calculations. Finally, several works have successfully combined the Gaussian mixture model with different techniques to increase the robustness of the foreground detection. For instance, Zhou and Zhang merged the foreground extracted by the mixture of Gaussians algorithm with the Lucas-Kanade optical flow to obtain better segmentation of foreground objects [293]. The multi-scale approach has been used to enhance the discrimination between the background and the foreground [193, 283]. Active contours [260] and skin detection [209] have also been combined with the Gaussian mixture model to provide better delineation of the foreground blob. In this thesis, a new technique to handle fast changes in illumination will be presented in Section 3.4.

2.3.3 Object Segmentation

Segmentation is an efficient technique to detect objects since it delineates different shapes in an image. Segmentation aims to label each feature depending on the object it belongs to; all features in the same object are attributed the same label.

Segmentation is applied to patches or dense and homogeneous areas of an image. Therefore, features such as color and optical flow are suitable for segmentation. In this subsection, we present the most common segmentation techniques: principle component analysis (PCA) , mean-shift, watershed and diffusion methods.

Principle Component Analysis

Principle component analysis is used to generate uncorrelated features (principal components) from correlated features. It is defined as an orthogonal linear transformation that provides a new coordinate system where projection of the feature on the first principal component minimizes the largest variance in the data. The second minimizes the second largest variance, and so on. Formally, PCA performs an eigen decomposition on the covariance matrix of the centered data, that is, if $\mathbf{C}$ is the covariance matrix of the data in the feature space, $\mathbf{C} = \mathbf{V}^T\mathbf{D}\mathbf{V}$. The column vectors in $\mathbf{V}$ represent the eigenvectors, or the principal components, and the values on the diagonal of $\mathbf{D}$, their eigenvalue. The first K eigenvalues are retained with their eigenvectors being the basis vectors.

With the assumption that the noise is small, PCA provides a good modeling of the object since noise is discarded with the last $N - K$ components. Objects are then detected by projecting the features onto the new components. This technique achieves outstanding results when the appearance of the object varies within a given category. The detection of the object is determined as the minimum of the projection of the features on the principal components. PCA is employed in [284] to extract relevant features for detection of vehicles. Pedestrians have been detected in images based on edges and color detection [162]. PCA was also implemented to generate eigenimages or eigenspaces. Bischof *et al.* proposed an algorithm that is insensitive to illumination changes by finding the principal components of images passed through a filter bank [21]. In [119], Jogan and Leonardis proposed to detect a portion of a 360° image by projection on the principal components of the entire image. Ali and Shah built an eigentemplate based on kernel PCA to model different objects for detection in images [2, 3]. Kernel PCA is a projection of the features in a higher dimension space that enables fast PCA.

Mean-shift

Mean-shift is a widely used non-parametric algorithm for object segmentation. As an explorative technique, mean-shift seeks local maxima in a distribution by kernel estimation. Let us assume that a set of N samples $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ is available (*e.g.*, $\mathbf{x}_i$ is a pixel color in an image), it is possible to estimate the distribution at each location $\mathbf{x}$ of the sample space as

$$\hat{f}_K(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} K(\mathbf{x} - \mathbf{x}_i)\,, \tag{2.14}$$

where K is a kernel as defined in [48]. For example, K can follow a Gaussian distribution. Now, if a location $\mathbf{x}$ is randomly selected as starting point in $\mathbf{X}$, it can be shown that by iteratively shifting $\mathbf{x}$ with the mean-shift vector

$$m_K(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in \mathbf{X}} \mathbf{x}_i . K\left(\mathbf{x} - \mathbf{x}_i\right)}{\sum_{\mathbf{x}_i \in \mathbf{X}} K\left(\mathbf{x} - \mathbf{x}_i\right)} - \mathbf{x}\,, \tag{2.15}$$

the location $\mathbf{x}$ will reach the local mode (see [55] for example). If the procedure is iterated for each pixel in $\mathbf{X}$, and the value of the local mode is used to label the starting point pixel, the image is segmented by exploration of the density and objects with similar features will be identified as the same object.

This technique was first introduced by Fukunaga and Hostetler [83] in 1975, and was generalized by Cheng in 1995 [48]. Comaniciu and Meer have greatly contributed to the analysis and understanding of the mean-shift for object segmentation [53,55, 54]. The mode seeking property of mean-shift algorithm has also been thoroughly exploited in object tracking [226,4], object identification such as edge detection [51], color [116, 53] or spatio-temporal segmentation [125] as well as non-photorealistic rendering [79]. Its popularity is due to the wide range of distributions that can be modeled.

Diffusion Methods

Diffusion methods for object segmentation are techniques identifying homogeneous regions in a feature image by diffusion. Starting from a pre-determined set of points with a unique label, the algorithm defines similar regions by processing neighborhood pixels. Once every pixel has been visited, the image is segmented.

The simplest diffusion technique is the region growing algorithm [103]. Starting from selected points in the image, the algorithm associates neighboring pixels to the same class if the cost between the reference pixel and the neighbor is below a given threshold. The cost function can be of different nature such as Euclidian distance or absolute error. If a pixel does not belong to any class, a new class is created. Constraints are added on the regions so that the relevance of the segmentation is increased. For instance, Ying-Tung *et al.* use morphological edge detection to partially delineate the regions [280]. Recursive median filtering is included in the region growing process in [88] and graphs between different segments are used to constrain the growth in [243].

Graphs and dynamic trees have also been successfully employed to segment objects. Graphs model the inter-dependence of different segments in order to classify pixels. In the probabilistic framework, maximum likelihood is used to determine the label of the pixels with a graph. Hidden Markov models and Markov random fields are examples of statistical graph modeling for image segmentation [49, 178]. Dynamic trees are graphs for top-down decisions; a multi-scale approach is traditionally adopted. Specifically, a low resolution image is segmented and used to determine the segmentation in higher level images. Discrete wavelet transform (DWT) provides a convenient framework for multi-scale approach. Successful segmentation has been reported with a probabilistic affection model in [255] and with a texture model in [208].

Watershed algorithms are based on region growing techniques inspired by natural flooding. A description of different watershed techniques can be found in [213]. The process is initialized at different points called markers. Crests and valleys are defined as the highest and lowest points of the intersection of two or more surfaces. Images can be seen as a surface with crests and valleys. The watershed process is often described as filling the surface, starting from valleys, with water until crests are reached yielding basins of homogeneous and plane surfaces. Watershed has produced good results in internal edge suppression for object segmentation [229], or multiscale image segmentation [127].

2.3.4 Summary of Object Identification

Identification is based upon the representation of an object and results in the discrimination of unique features that enable tracking and differentiate an object from the others. Identification revolves around algorithms that seek characteristics of the object. Supervised learning techniques such as artificial neural networks, support vector machines or ADABOOST lead to robust identification from prior training. However, the training requirement renders these techniques prohibitively costly in terms of computation time when applied to multi-category object identification. Distribution representation offers an efficient alternative for object detection. In particular, histograms combined with prior inference on the object efficiently address this problem but can lead to less robust results if the modeled features are not selected adequately. Finally, segmentation performs the grouping of similar regions. Principle component analysis, mean-shift or diffusion methods have yielded to good results in segmenting different object in a frame, and thus providing object identification.

2.4 Object Tracking

Object tracking is the main focus of this thesis. As described in Section 2.1, there is a very strong interaction between object representation, object identification and tracking because tracking is performed on discriminative features of the object defined by the first two tasks. Also, because tracking is the centerpiece of this thesis, this section only focuses on the description of existing tracking algorithms and their characteristics: the formal introduction of the theory underlying the tracking techniques as well as its framework is omitted or limited to a minimum here. However, we will refer to the relevant sections and chapters when appropriate for an in-depth analysis. The aim is to provide a clear and simple overview of the field. The reader is referred to the book on multitarget-multisensor tracking by Bar-Shalom and Li for more insights on tracking theory [13].

Tracking algorithms provide generic estimation tools applicable to a wide range of fields, including financial market estimation [170,264], meteorology and climatology

[173] or quantum mechanics [46]. It is thus necessary to clearly define the framework of object tracking algorithms from a video processing perspective. The object is represented by a feature vector that includes some characteristics to track. The feature vector at time t is denoted $\mathbf{x}_t$. Without loss of generality, if it is assumed that tracking of an object starts at time $t = 1$. The feature track $\mathbf{X}$ at time $t = T$ is defined as

$$\mathbf{X} = \{\mathbf{x}_t | t = 1..T\}\,. \tag{2.16}$$

Some models assume that the feature vector $\mathbf{x}_t$, and subsequently the track $\mathbf{X}$ are not accessible, but only an observation $\mathbf{z}_t$ is. In this case, the observation track can be defined in a similar fashion

$$\mathbf{Z} = \{\mathbf{z}_t | t = 1..T\}\,. \tag{2.17}$$

Finally, we denote a portion of feature track from start time t_s to finish time t_f as $\mathbf{x}_{t_s:t_f} = \{\mathbf{x}_t | t = t_s..t_f\}$, and likewise for the observation track $\mathbf{z}_{t_s:t_f} = \{\mathbf{z}_t | t = t_s..t_f\}$. Note that $\mathbf{X}$ and $\mathbf{Z}$ can be trivially denoted by $\mathbf{X} = \mathbf{x}_{1:T}$ and $\mathbf{Z} = \mathbf{z}_{1:T}$.

Before further investigations into object tracking, it is essential to clarify the difference between tracking and real-time object identification. Indeed, although some techniques described in Section 2.3 can be performed in real-time, they do not provide tracking of the object *per se*. In this section, we present deterministic and probabilistic tracking, the two main approaches in the field. The handling of occlusions which relies upon object representation, identification and tracking is also introduced.

2.4.1 Deterministic Tracking

Deterministic tracking has been widely used in the literature due to its simplicity. The terminology "deterministic" means that the tracking algorithm does not integrate any uncertainty in the modeling of the problem. Nevertheless, this does not mean that problems including noise or other types of uncertainty cannot be tackled by deterministic algorithms; the uncertainty is simply not catered for. Deterministic algorithms are convenient because they require little computation. They traditionally rely on simple parametric tracking for points and contours. However,

more advanced models and in particular kernel-based tracking have also been implemented.

Parametric Tracking

Parametric tracking relies on a set of samples to determine the state of the feature vector at time t from a portion of the feature track. Without loss of generality and because the feature vector depends at most on the entire feature track at time $t-1$, $\mathbf{x}_t$ is written as

$$\mathbf{x}_t = f(\mathbf{x}_{1:t-1}, \Theta)\,, \tag{2.18}$$

where Θ is the vector of parameters. Classically, the problem is reduced to a linear or locally linearized transform to simplify calculations so that the tracking can be formulated in matrix form, *i.e.*, $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{1:t-1}$. Parametric tracking embeds a set of motion constraints to determine the state of the feature vector. Kinematic models are generally used for their relevance to tracking based on position and speed. Blair presents in [23] a parametric tracking based on an Alpha-Beta filter, *i.e.*, quasi constant velocity and acceleration with zero-mean noise. However, parametric techniques were essentially employed in the early to mid 90s because of the great performance they offered for a low computational cost; Yilmaz *et al.* give examples of such tracking [275, Section 5.1]. A first category is defined on rigidity constraints to find the optimum match of the feature vector state [109, 203, 221, 216]. The second category, which is the ground work for most probabilistic tracking algorithms, is based on motion constraints, often directly derived from Newton's laws [222, 253].

Snakes and Contour Tracking

Contour tracking estimates the variation in the contour of an object at time t. The contour is described by a spline consisting of a set of control points (see Subsection 2.2.2) and the tracking is performed recursively on the contour at time $t-1$ through minimization of an energy functional. The functional is composed of an internal energy $E_{internal}$ and an image energy E_{image}. The total energy E is given by

$$E = \oint \left(E_{internal} + E_{image}\right). \tag{2.19}$$

Contour tracking algorithms differ in their representations of the energies and the minimization techniques. For instance, E_{image} is traditionally based on the gradient image that represents the edge energy [174, 191, 207, 206]. The internal energy $E_{internal}$ models the constraints on the active contour. Such constraints can be smoothness [179] or contour speed [276]. Contour tracking has been employed in numerous fields of application, including tracking [184, 211, 276], segmentation [238, 280, 260, 273] or higher level tasks such as shape classification [236].

Kernel-based Tracking - Mean-shift

Kernel-based tracking has been the focus of attention in recent years due to the convenient framework it provides for object tracking. Here, only an overview of the widespread mean-shift technique for object tracking is presented. Kernel-based techniques rely on a smoothing operator, a kernel, to locally estimate a distribution. The aim is to climb a gradient of feature probability distribution to reach the maximum probability of an object feature representation. Traditionally, mean-shift relies on color representation, and histograms in particular, to track the object. Han and Comaniciu have been major contributors in this field with numerous publications [56, 94, 96]. However, because mean-shift is a local gradient ascent algorithm, the convergence to a global maximum is not guaranteed and the technique is still an active field of research. One of the major problems with mean-shift is the adjustment of the kernel bandwidth. Multi-scale approaches [10] and direct kernel bandwidth tuning have been proposed in recent years [52]. Multiple kernel tracking has also been proposed to tackle the problem [93, 192]. Finally, Bouttefroy *et al.* proposed to estimate the kernel bandwidth and initialization through the Kalman filter for the purpose of vehicle tracking [26]. This work is presented in Section 4.4 of this thesis.

2.4.2 Probabilistic Tracking

Probabilistic tracking has emerged from the need to account for uncertainty in tracking. There are several sources of uncertainties in a video. First, the signal is degraded with noise. Second, the information on the object of interest can be inaccessible due to occlusion, clutter or simply because the information is hidden. Finally, it might be required to estimate the state of the feature vector with a

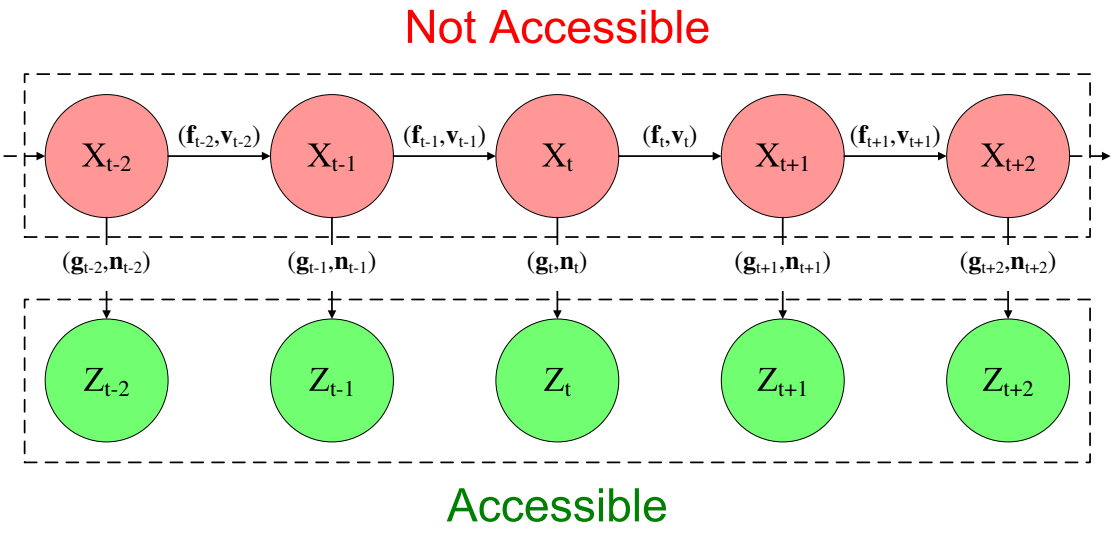


Figure 2.7 Representation of the hidden Markov chain model. The hidden state (in red) is not accessible. The observation (in green) is accessible.

precision greater than what is available. For instance, probabilistic estimation is used in super-resolution or tele-medicine to achieve sub-pixel accuracy in tracking. This subsection provides the reader with an overview of probabilistic tracking. The hidden Markov model and the recursive Bayesian approach as well as the Kalman filter and the particle filter are developed below.

Hidden Markov Model and Recursive Bayesian Approach

The hidden Markov model (HMM) is employed in visual object tracking for its ability to handle degradations introduced during the acquisition process, which was described in Section 1.2. The hidden Markov model is composed of two layers: a hidden layer, representing the Markov chain on the state, and an observation layer, providing inference on the state of the hidden Markov chain. Figure 2.7 displays a schematic view of the system. The diagram can be mathematically expressed as follows:

$$\mathbf{x}_t = \mathbf{f}_{t-1}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})\,, \tag{2.20}$$

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t)\,, \tag{2.21}$$

where $\mathbf{f}_{t-1}$ and $\mathbf{h}_t$ are vector functions; they are assumed to be known, possibly nonlinear and time dependent. The functions depend on the states $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ and the process and observation noises, $\mathbf{v}_{t-1}$ and $\mathbf{n}_t$, respectively. The hidden Markov model sets up the framework for recursive Bayesian filtering. The Bayesian approach is based on Eqs. (2.20) and (2.21); it aims to provide some degree of belief for the state $\mathbf{x}_t$ from the set of observations $\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_t\}$ available at time t. In other words, the Bayesian recursion estimates the *posterior* density $p(\mathbf{x}_t|\mathbf{Z}_t)$ to estimate the state of an object using Bayes rule. Let us assume that the posterior probability density function $p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})$ at time $t-1$ is known. The Bayesian recursion is performed in two steps: prediction and update.

Prediction step Considering the observation $\mathbf{z}_t$ is not available, the predicted pdf $p(\mathbf{x}_t|\mathbf{Z}_{t-1})$ is derived via the Chapman-Kolmogorov equation that enables the marginalization of a variable in the joint pdf $p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{Z}_{t-1})$. Assuming the Markov property on the process yields

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) d\mathbf{x}_{t-1} \,. \tag{2.22}$$

The random variable $\mathbf{x}_{t-1}$ is therefore marginalized. Equation (2.22) describes the predicted density $p(\mathbf{x}_t|\mathbf{Z}_{t-1})$ in terms of the posterior pdf at time $t-1$ and the *prior* density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ defined by the process equation (2.20).

Update step When the observation $\mathbf{z}_t$ becomes available, the predicted pdf is updated via Bayes theorem, to obtain the posterior pdf at time t

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{Z}_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_{t-1})} \,, \tag{2.23}$$

with $p(\mathbf{z}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{Z}_{t-1}) d\mathbf{x}_t$ being a normalizing constant (independent of the marginalized variable $\mathbf{x}_t$). The posterior density therefore depends on the prior density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the *likelihood* function $p(\mathbf{z}_t|\mathbf{x}_t)$ using the observation equation (2.21). The posterior density at each time is derived from the pair of recursive equations presented above. To perform tracking, *i.e.*, update the estimate of the posterior, only the initial density $p(\mathbf{x}_0)$, the prior density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ are required.

In the Bayesian framework, the system collects clues to corroborate or reject a prior hypothesis. This accumulation is called the Bayesian inference. The Bayesian probabilistic approach has been implemented for data analysis and classification [39,137,278], object segmentation [49,60,146], surface reconstruction [69], behavior recognition [47,189,36,271,288], etc.

Kalman Filters

The Kalman filter provides the optimal solution for tracking in a linear and Gaussian environment [130]. This result is admitted here and will be shown in Section 4.3. The Gaussian context allows the recursive estimation of the state from the observation in closed form. The distribution of the state in the feature space is Gaussian; therefore, it is only necessary to keep track of the mean vector and covariance matrix of the state to characterize the entire distribution. Consequently, the Kalman filter performs estimation for a low computational cost. The Kalman filter can be applied to any object representation and tracking technique, from kinematic models [183] to entropy based methods [67,298] or elastic matching (B-splines) [272]. The Kalman filter is also used in 3D object or track modeling [144,197,232].

One of the main limitations of the Kalman filter is the inability to handle non-linear models. The extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [126] address this issue by approximation of the non-linearities. The first one uses a local linear estimator based on the Jacobian of the non-linear functions. The Jacobian provides a first-order approximation of the non-linearities. To include the second order term, the unscented transform is employed leading to the UKF. The unscented transform captures the non-linearities through the transformation of sigma points [8]. Chosen adequately, sigma points give a local numerical approximation of the non-linear functions that are assumed for tracking. Furthermore, different variants of these algorithms have also been proposed such as the Gaussian mixture probability hypothesis density filter (GMPHDF) that offers multi-modality via a mixture of Kalman filters [259].

Particle Filters

Particle filters offer the advantage of relaxing the Gaussian and linearity constraints imposed upon Kalman filters. The range of problems tackled is therefore increased. On the downside, particle filters only provide a suboptimal solution which statistically converges to the optimal solution. The asymptotic convergence is ensured by Monte Carlo methods and follows the central limit theorem. An introduction to Monte Carlo methods can be found in [84]. As for Kalman filters, the framework and derivations related to particle filters are omitted here and will be presented in Section 5.2. One of the drawbacks of particle filters is the computational complexity for high dimensional state vectors. For this reason, particle filters have only been introduced in the object tracking field with the increase of computation power. It was first applied to splines estimation with the Conditional Density Propagation (CONDENSATION) algorithm proposed by Isard and Blake in 1998 (see [110] and [112]). The same year, Doucet proposed a technical report setting the framework of particle filtering for visual object tracking in [70]. It is also worthwhile directing the reader to a tutorial on particle filters by Arulampalam *et al.* that provides an introduction to Bayesian filtering [9].

Within the last decade, the interest in particle filters has been growing exponentially. Early contributions were based on the Kalman filter models; for instance, Van Der Merwe *et al.* discussed an extended particle filter (EPF) and proposed an unscented particle filter (UPF) , using the unscented transform to capture second order nonlinearities [247]. Later, a Gaussian sum particle filter was introduced to reduce the computational complexity [139]. As far as applications are concerned, particle filters are ubiquitous from head tracking via active contours [286, 80] or edge and color histogram tracking [274, 150] to sonar [246] and phase [296] tracking. Audio-visual fusion has been proposed in the particle filter framework [182] and particle filters have also been used for object discrimination [61, 265]. There has also been a plethora of theoretic improvements to the original algorithm such as the kernel particle filter [42,43], the iterated extended Kalman particle filter [153], the adaptive sample size particle filter [142, 143] and the augmented particle filter [225].

2.4.3 Occlusion Handling

The ability of tracking algorithms to handle occlusion is crucial to provide a good estimate of the object state. Occlusion handling aims to reduce the effects of the lack of information on an object under occlusion. This subsection presents the definition and the detection of occlusion before investigating the two main techniques employed to resolve the problem, namely, the integration of prior inference and the use of multi-camera tracking.

Definition and Detection of Occlusion

Occlusion is defined as the lack of visual clues on part of or on the totality of an object. In the framework of tracking, the alteration of the observations is the result of occlusion and it can be mathematically expressed, following from Eq. (2.17), as:

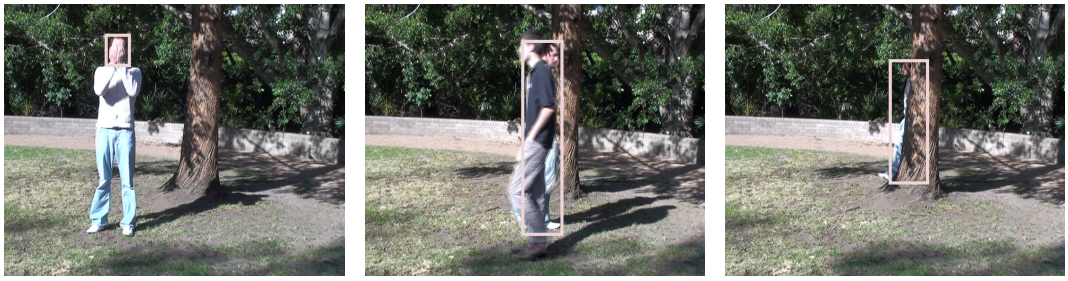
$$\hat{\mathbf{Z}} = \{\mathbf{z}_t | t \in T_N, \hat{\mathbf{z}}_t | t \in T_O\}\,. \tag{2.24}$$

where $\mathbf{z}_t$ is the observation with no occlusion, that is for "normal" time step T_N, and $\hat{\mathbf{z}}_t$ is the observation under partial or total occlusion, for "occlusion" time step T_O. Note that $T_N \cap T_O = \emptyset$. There exist three different cases of occlusion:

Self occlusion (Fig. 2.8(a)) The object of interest is articulated and the constraints on motion do not prevent the overlap when the object is projected on the camera plane. Self occlusion will not be dealt with since articulated objects are out of the scope of this thesis.

Inter-object occlusion (Fig. 2.8(b)) The object of interest is occluded by another object in the frame. Inter-object occlusion can occur at any time since the environment in which the object evolves is not controlled. Inter-object occlusion can be of any duration.

Occlusion from a background object (Fig. 2.8(c)) The object of interest is occluded by the background. Typically, the object passes behind a tree, a house, etc. The background is usually static and therefore enables the learning of inference on occlusion. However, occlusion is usually total and the observation $\hat{\mathbf{z}}_t$ does not exist, *i.e.* $\hat{\mathbf{z}}_t = \emptyset$.



(a) Self Occlusion (b) Inter-Object Occlusion (c) Background Occlusion

Figure 2.8 The three different types of occlusion. (a) The face is occluded by hands; (b) the person is occluded by another person; (c) the tree occludes the person.

The analysis of the altered observation $\hat{\mathbf{Z}}$ leads to the detection of occlusion. For instance, incoherent observations are a clue to occlusion. More precisely, if the probability of an observation drops rapidly, the object can face partial or total occlusion. Simple analysis of the observation such as probability of occurrence thresholding provides a criterion to potential occlusion. The degree of occlusion can also be inferred from observations. Occlusion detection is crucial since it provides an indicator of the tracking confidence.

Occlusion Resolving

Occlusion resolving is performed to estimate the state of the object when observations are altered or lacking. To date, two different approaches have been proposed to resolve occlusion: estimating the state from prior inference and using multiple cameras to alleviate the occlusion. A third alternative, although not resolving the occlusion but allowing recovery of the track, is presented as data association.

Prior inference traditionally substitutes observation in the case of occlusion. Indeed, when there is a shortage of observations, the prior behavior of the object can provide clues on the current feature state. However, the closer the estimation is from the actual behavior, the better the recovery of tracking is after occlusion. The techniques developed to handle occlusion differ, depending on the nature of the tracking. Kinematic models have been used to handle self-occlusion in 3D vehicle tracking [118] or inter-object occlusion [241, 33, 290] while prior shape modeling has been employed for self-occlusion [235] and inter-object/background occlusion [276, 281].

For difficult occlusions, and in particular when the prior information is not suitable to estimate the state of the object under occlusion, it is necessary that a substitute for the observation is obtained. A solution is provided by the use of multiple views or multiple camera tracking. Multiple views of the scene can originate from stereo-vision where images from the scene are captured from slightly different angles [124, 175, 18, 155]. Multiple view systems synthesize the state of an object from images of different cameras with overlapping or non-overlapping fields of view [254, 201, 223, 295, 44].

Finally, data association is necessary to identify tracks when multiple objects are under occlusion. Even though this technique does not resolve the occlusion problem, it improves recovery of the tracks after occlusion. The different techniques available in the literature are algorithm dependent. The joint probability data association filter (JPDAF) is a generalization of the Kalman filter to multi-target tracking where the final probability of a state is the weighted sum of the posterior probability over each observation [205]. The same framework can be applied to particle filters by computing the statistical distance between different tracks. It results in the merged probabilistic data association (MPDA) introduced in [131]. In contrast with the previous techniques, the probability hypothesis density (PHD) handles data association directly in the update of the posterior density [259]. Finally, Chang *et al.* proposed a deterministic motion correspondence matrix (MCM) where the maximum a posteriori fitness of an observation to the state of the object associates a track to an observation; correspondence is therefore carried out [43].

2.4.4 Summary of Object Tracking

Object tracking brings together object representation and object modeling to provide an estimate of the object state. Tracking is therefore dependent upon the description of the object and is subject to uncertainties. Deterministic tracking is a powerful and efficient method to estimate the state of an object. General parametric tracking, based on kinematic models, contour tracking and kernel-based tracking provide efficient and fast solutions when noise is negligible. To handle uncertainties, probabilistic tracking has been developed in visual object tracking. The Bayesian

filtering framework allows tracking through noise. In particular, Kalman and particle filters have been extensively employed. Finally, the descriptions and solutions for occlusion handling have been presented in this section. Occlusion can be divided into three categories: self-occlusion, inter-occlusion and occlusion from the background. Occlusion can be resolved with prior knowledge or multi-view tracking. Data association is necessary in case of occlusion.

We will focus on probabilistic modeling and occlusion handling in this thesis: deterministic tracking is not suitable since it does not fit into the framework of this study and in particular does not cater for the uncertainties described in Chapter 1.

Chapter 3

Semi-Constrained Gaussian Mixture Model for Background Subtraction

3.1 Introduction

Noise, pixel value evolution through time, object representation or behavior can be modeled with probability distributions. A particularity of visual object tracking is that the information received stems from unknown or unaccessible probability density functions (pdfs) and only observations of samples are readily accessible. For instance, the value of a pixel through time can be seen as a stream of incoming samples of an underlying density characterizing the presence of different surfaces (objects) in a particular area of the image. Therefore, there is an interest in knowing the density for subsequent tasks, such as noise reduction or segmentation. On a global scale, learning the pdf of motion patterns can lead to the detection of abnormal behavior.

The objective of density representation is to model the probability density function of a random variable over a support $\mathcal{D}$. The pdf is defined as a real function $f_{\mathbf{X}}(\mathbf{x}), f : \mathcal{D} \mapsto \Re^{+}$. In the discrete case, the probability density function is a probability mass function. To allow a common framework for continuous and discrete case, the probability mass function is defined as $f_{\mathbf{X}}(\mathbf{x}) = Pr[\mathbf{X} = \mathbf{x}]$. We are interested in recovering an estimate p of the probability density function with a set of N

samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, of dimension $n \times 1$, drawn from the underlying pdf f.

The representation of probability density functions can be divided into two main categories: non-parametric representation such as histograms and Parzen windows or other kernel-based representations, and parametric representation such as the Poisson, Gaussian or Gaussian mixture models. This chapter develops the Gaussian mixture model (GMM) for background subtraction and presents a new Gaussian mixture algorithm handling fast changes in background distribution. Section 3.2 introduces the general formulation along with an online technique for estimating the optimal values of the set of parameters. Section 3.3 describes the implementation of the GMM for background subtraction and presents the current shortcomings of the technique in environments with fast changes in illumination. Section 3.3.3 focuses on the analysis of the GMM parameters and Section 3.4 proposes a new, semi-constrained, Gaussian mixture model handling illumination changes. Finally, Section 3.5 presents the experimental setup and provide some results for various scenarios with illumination changes.

3.2 Density Representation with Gaussian Mixture Model

The study of parametric representations for visual object tracking is crucial in that they provide an accurate estimate with a priori knowledge on the shape of the density of interest. In contrast with non-parametric techniques, parametric representations are based upon the assumption that the density f follows a pre-defined functional that can be entirely characterized by a vector of parameters that forms $\Theta = \{\theta_1, ..., \theta_k\}$. A parametric estimate is represented as a function $p(\mathbf{x}|\mathbf{\Theta})$ dependent on the set of parameters Θ.

The maximum-likelihood estimator provides the optimal value of the set of parameters $\mathbf{\Theta}$. If a closed form expression cannot be derived from the density estimate $p_{\mathbf{\Theta}}(\mathbf{x})$, the Expectation-Maximization algorithm is used to recursively approximate the optimal set of parameters. There are numerous books on the topics and the reader is referred to [71] and [169] for a comprehensive introduction to these tech-

niques. The most common functional is the Gaussian density; this chapter focuses on the Gaussian mixture model, for its ubiquity in computer vision and its attractive characteristics such as fast update and compactness of representation through the vector of parameters $\mathbf{\Theta}$. Assuming that the pdf f is completely defined by a mixture of K Gaussians, the estimate $p(\mathbf{x}|\mathbf{\Theta})$ becomes

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{k=1}^{K} P(k)p(\mathbf{x}|k,\theta_k)\,, \tag{3.1}$$

with $\mathbf{\Theta} = \{\theta_1, ...\theta_K\}$. The probability $P(k)$ is called *mixing parameter* or *weight*, the density $p(\mathbf{x}|k,\mathbf{\Theta}_k)$ is a *component*. The component density is given by

$$p(\mathbf{x}|k,\theta_k) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k,\mathbf{\Sigma}_k) = \frac{|\mathbf{\Sigma}_k|^{-1/2}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right), \tag{3.2}$$

where $\boldsymbol{\mu}_k$ is the mean vector, $\mathbf{\Sigma}_k$ is the covariance matrix, ^{T} is the transpose operator, $|\mathbf{\Sigma}_k|$ is the determinant of the covariance matrix and n is the dimension of the column vector $\mathbf{x}$. The optimal set of parameters for the Gaussian mixture model is given by the ML estimator derived from the joint probability $p(\mathbf{x}_1, ..\mathbf{x}_n|\mathbf{\Theta})$. Considering that the samples $\mathbf{x}_i$ are *independent and identically distributed*, the maximum likelihood estimator yields [71]

$$P(k) = \frac{1}{N}\sum_{i=1}^{N} P(k|\mathbf{x}_i,\mathbf{\Theta})\,, \tag{3.3}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} P(k)\mathbf{x}_i}{\sum_{i=1}^{N} P(k)}\,, \tag{3.4}$$

and

$$\mathbf{\Sigma}_k = \frac{\sum_{i=1}^{N} P(k)(\mathbf{x}_i-\boldsymbol{\mu}_k)(\mathbf{x}_i-\boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} P(k)}\,, \tag{3.5}$$

where $P(k|\mathbf{x}_i,\mathbf{\Theta})$ is the posterior probability of being in the presence of the kth component knowing the estimate of the set of parameters and the sample $\mathbf{x}_i$. However, Eqs. (3.3), (3.4) and (3.5) are seldom used if the size of the sample set $\mathbf{X}$ is large due to the memory requirements for storing the entire history information. Instead, recursive online approximations are used, reducing the storage to the previous value

of the parameters:

$$P(k,t) = (1-\alpha)\, P(k,t-1) + \alpha\, P(k|\mathbf{x}_i, \boldsymbol{\Theta})\,, \tag{3.6}$$

$$\boldsymbol{\mu}_k(t) = (1-\beta)\, \boldsymbol{\mu}_k(t-1) + \beta\, \mathbf{x}_i\,, \tag{3.7}$$

and

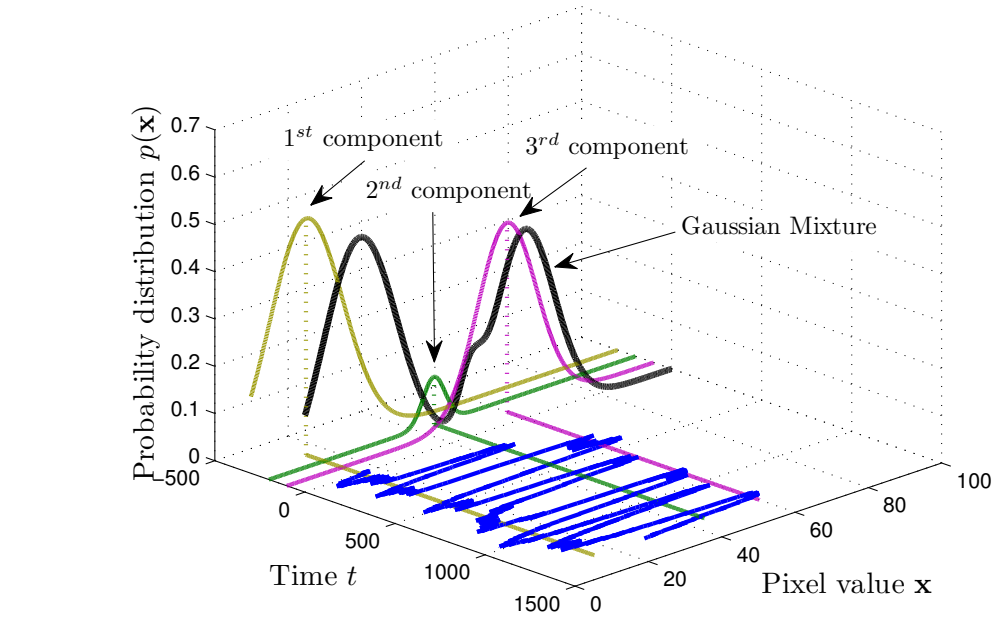
$$\boldsymbol{\Sigma}_k(t) = (1-\beta)\, \boldsymbol{\Sigma}_k(t-1) + \beta\, (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\,. \tag{3.8}$$

The online form is adopted hereafter in the derivation of the Gaussian mixture model for background subtraction.

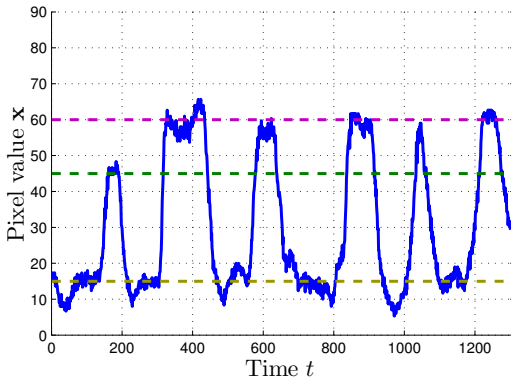
3.3 Background Modeling using the Gaussian Mixture Model

Background modeling by Gaussian mixtures is a pixel based process. In a video sequence, a given pixel is presented with surfaces corresponding to different states, *e.g.* different objects, changes in illumination conditions, etc. Each pixel in the video sequence is assumed to follow a random process with underlying density f. This section presents the Gaussian mixture model for background subtraction and the classification background/foreground as proposed by Stauffer and Grimson [233]. Let us denote by $\mathbf{x}$ the switching random variable taking the value of a pixel throughout the sequence; the presence of different surfaces causes the switching of $\mathbf{x}$ to different states. The surface from which the sample $\mathbf{x}_i$ is drawn is labeled with an index $k \in [1..K]$ where K is the total number of surfaces. Assuming the noise around each mode is Gaussian, the probability density of the random variable $\mathbf{x}$ is fully recovered with a mixture model composed of K Gaussians. For a given pixel, the pdf of the value $\mathbf{x}$ is modeled by the sum of a set of independent Gaussian densities.

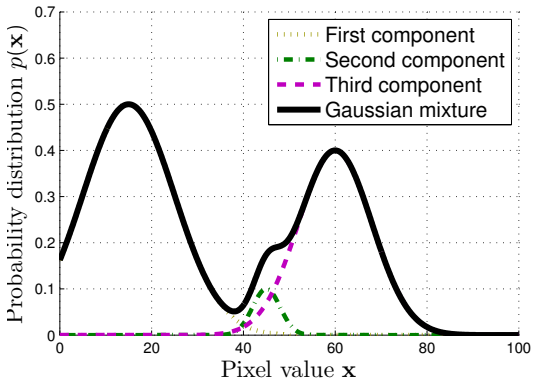
Figure 3.1 shows the evolution of a pixel intensity density over time along with the probability density estimate. The probability density function of a Gaussian mixture comprising K Gaussian component densities is given by Eq. (3.1). Therefore, the mixture can effectively be modeled with a set of weights $P(k)$ and a set of parameters $\boldsymbol{\Theta} = \theta_k$ with $\theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. The aim in background modeling is to estimate the set of parameters $\boldsymbol{\Theta}$ over time to obtain an estimate of the pixel value



(a) Global view



(b) Pixel value over time



(c) Mixture model

Figure 3.1 Probability density function of a pixel intensity in a video sequence over time and associated Gaussian mixture model. (a) Global view of the pixel value over time and the inherent probability density of the pixel intensity. (b) View of the pixel intensity over time. The value switches between the different modes of the pdf . (c) Gaussian components and Gaussian mixture model.

density. The Gaussian mixture model offers an adequate framework for such an estimation as it makes possible the coexistence of several hypothesis for the background and the foreground: bimodal (*e.g.* blinking traffic lights) or, more generally, multimodal densities, where the weight of each component accounts for its probability of occurrence.

The Gaussian mixture model is updated with Eqs. (3.6), (3.7) and (3.8) where the posterior probability $P(k|\mathbf{x}, \boldsymbol{\Theta})$ of a pixel to be drawn from the kth component can be rewritten from Eq. (3.3) as

$$P(k|\mathbf{x}, \boldsymbol{\Theta}) = \frac{P(k)p(\mathbf{x}|k, \boldsymbol{\Theta})}{p(\mathbf{x}|\boldsymbol{\Theta})} = \frac{P(k)\mathcal{N}(\mathbf{x}; \boldsymbol{\theta}_k)}{\sum_{k=1}^{K} P(k)\mathcal{N}(\mathbf{x}; \boldsymbol{\theta}_k)} \,. \tag{3.9}$$

Also, since background subtraction by Gaussian mixture model is an on-line process, it is necessary to cater for new surfaces. To enable a fast integration of a previously unseen surface, the posterior needs to be truncated when the probability falls below a set threshold τ. The Mahalanobis distance between the switching random variable $\mathbf{x}$, representing the pixel value, and each component of the Gaussian mixture is compared to τ to determine whether or not the pixel is a member of the kth component:

$$(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \leq \tau \,. \tag{3.10}$$

The threshold value τ sets the boundary between a *match* and a *non-match* of the pixel value with each mixture component. If the Mahalanobis distance is smaller than the threshold, there is a match and the component of the mixture model is updated with the value of the pixel. If the pixel value does not match with any of the Gaussians, a new Gaussian component is created. Because the number of Gaussians in the model is fixed to K, the new component replaces the Gaussian with the lowest probability of occurrence, $P(k)$, since it contributes the least to the density estimate as it is the less likely to match an incoming pixel. The suppression of the least probable component in the density leads to minimal estimation error. Furthermore, to lower the computation cost of the algorithm, it is commonly assumed that the pixel values are isotropically distributed. This assumption results in a diagonal covariance matrix $\boldsymbol{\Sigma}_k = \sigma_k^2 I$, where I is the identity matrix. Even though this assumption reduces the degrees of freedom of the Gaussian, and hence the capability

of adapting to the true density, it significantly lowers the computation complexity of the algorithm, avoiding a costly computation of the inverse of a full matrix $\mathbf{\Sigma}_k^{-1}$. The set of parameters $\mathbf{\Theta}$ and the probabilities $P(k)$ are updated according to the Gaussian mixture algorithm described in Algorithm 3.1— for the sake of clarity, the mixing parameters, $P(k)$, and the posterior probabilities, $P(k|\mathbf{x}, \mathbf{\Theta})$, are denoted by w_k and q_k, respectively.

Algorithm 3.1 Generic Gaussian Mixture Algorithm

Require: $0 < \alpha < 1$ and $0 < \beta < 1$

Initialization

$$w_k = \alpha\,, \quad \boldsymbol{\mu}_k = \mathbf{x}_0\,, \quad \sigma_k^2 = \sigma_0^2\,, \tag{3.11}$$

where $\mathbf{x}_0$ is the vector of pixel values at time $t = 0$, and $\sigma_0^2 > 0$.

while incoming image i **do**

for each pixel $\mathbf{s}$ in the image **do**

for each Gaussian component k **do**

Compute the posterior probability q_k as follows:

$$q_k = \begin{cases} w_k\,\mathcal{N}(\mathbf{x}_i^{\mathbf{s}}; \boldsymbol{\theta}_k) & \text{if} (\mathbf{x}_i^{\mathbf{s}} - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x}_i^{\mathbf{s}} - \boldsymbol{\mu}_k) \geq \tau\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.12}$$

if $\sum_{k=1}^{K} q_k \neq 0$ **then**

$$q_k = \frac{q_k}{\sum_{k=1}^{K} q_k}\,, \tag{3.13}$$

$$w_k(t) = (1 - \alpha)\, w_k(t-1) + \alpha\, q_k\,, \tag{3.14}$$

$$\boldsymbol{\mu}_k(t) = (1 - \beta)\boldsymbol{\mu}_k(t-1) + \beta \mathbf{x}_i^{\mathbf{s}}\,, \tag{3.15}$$

$$\sigma_k^2(t) = (1 - \beta)\, \sigma_k^2(t-1) + \beta\, (\mathbf{x}_i^{\mathbf{s}} - \boldsymbol{\mu}_k(t))^T (\mathbf{x}_i^{\mathbf{s}} - \boldsymbol{\mu}_k(t))\,, \tag{3.16}$$

Note: There are different techniques for the selection of β (details in Subsection 3.3.3, Eqs. (3.23) and (3.24)).

else

$$j = \underset{k}{\operatorname{argmin}}(w_k)\,, \tag{3.17}$$

$$w_k(t) = (1 - \alpha) w_k(t-1) \text{ for } k \neq j\,, \tag{3.18}$$

$$w_j = \alpha\,, \quad \boldsymbol{\mu}_j = \mathbf{x}_t\,, \quad \sigma_j^2 = \sigma_0^2\,. \tag{3.19}$$

end if

end for

end for

end while

3.3.1 Background/Foreground Classification

The aim of the background/foreground classification is to separate the subset of Gaussians modeling the background from the subset representing the foreground. The classification is necessary since the Gaussian mixture models all surfaces seen by the pixels. Indeed, the pseudo-code described in Algorithm 3.1 estimates a density but does not provide information about the classification of the Gaussians; a Gaussian can represent the probability of occurrence of either the background or the foreground. Stauffer and Grimson [233] proposed an efficient method to perform such classification. The K Gaussians of the model are sorted by decreasing weight-to-standard-deviation ratio, w_k/σ_k. Intuitively, Gaussians with the highest probability of occurrence, w_k, and lowest variability in the density, measured by σ_k, indicating a greater stability, are the most likely to model the background. However, because the number of components of the background is not known, it is assumed that the background is present with a ratio λ. After sorting the weight-to-standard-deviation ratios, the background (B) is defined as

$$B = \text{argmin}_{K_B} \left(\sum_{k=1}^{K_B \leq K} w_k > \lambda \right). \tag{3.20}$$

For a small value of λ, the background is most likely unimodal, whilst a larger value of λ leads to multimodal background. In [148], Lee has also proposed a method to model the background by training a sigmoid function on a set of ratios w_k/σ_k such that

$$P(B|\boldsymbol{\theta}_k) = \frac{1}{1 + e^{-a.w_k/\sigma_k + b}}. \tag{3.21}$$

where a and b are trained parameters. The sigmoid thus offers a soft boundary between foreground and background. A pixel is deemed to belong to the foreground if

$$P(B) = \frac{\sum_{k=1}^{K} \boldsymbol{\mu}_k . P(B|\boldsymbol{\theta}_k) w_k}{\sum_{k=1}^{K} P(B|\boldsymbol{\theta}_k) w_k} < \lambda. \tag{3.22}$$

where λ is empirically found to be equal to 0.5. However, Lee narrows the scope of the algorithm by training the sigmoid on a set of data representative of the background. This assumption forces the training of the system on the dataset before background extraction and limits the scope of the foreground extraction to off-line

use. The approach provides good results on the trained dataset. However, the background detection described in Eq. (3.20) is used to provide a common framework and therefore a fair comparison between Lee's and Stauffer and Grimson's algorithms.

3.3.2 State of the Art and Current Shortcomings

Since the original Gaussian mixture model proposed by Stauffer and Grimson [233], there has been little change to the update of the Gaussian mixture model itself; except the technique proposed by Lee to increase the learning rate of the recursive filters in order to accelerate the convergence of the parameters to their steady-state value [148]. However, several works have successfully combined the mixture model with different techniques to increase the robustness of the foreground detection. For instance, Zhou and Zhang merged the foreground extracted by the mixture of Gaussians algorithm with the Lucas-Kanade vector flow to obtain better segmentation of foreground objects [293]. The multi-scale approach has been used to enhance the discrimination between the background and the foreground [193, 283]. Active contours [260] and skin detection [209] have also proven better delineation of the foreground blob when combined with the Gaussian mixture model.

Most of the recent works based on the mixture of Gaussians introduced by Stauffer and Grimson, focus on shadow removal. There is no specific algorithm dedicated to change in illumination to date. The main difference between shadow and change in illumination is that shadow of moving objects generally occurs on background pixels whereas change in illumination encompasses both background and foreground. Shadow suppression usually relies on transformation of the color-space [270, 239] or analysis of the intensity in the RGB color-space through a 3D cone model [285]. Wu *et al.* proposed to remove shadow with graph cut and DFT [269]. Martel-Brisson and Zaccarin introduced the Gaussian Mixture Shadow Model that differentiates pixel density from cast shadows [165, 166]. Finally, Liu *et al.* proposed to extract reflectance from irradiance by homomorphic filtering [158]. Since the shadow is carried by the irradiance, the mixture model estimates shadow-free foreground through reflectance. These techniques rely on pre- or post-processing of the video sequence or on the classification of the background pixels but fail to provide an insight into

the pixel density modeling by the Gaussian mixture model.

3.3.3 Analysis of Background Substraction with GMM

In this subsection, we conduct an investigation into the effects of the Gaussian mixture parameters, in particular the influence of the variance, on the performance of background subtraction. Our objective is to suppress the effects of illumination changes without using a pre- or post-processing stage that would slow down the segmentation dramatically. The update of the covariance matrix in Eq. (3.16) is subject to the inherent trade-off between speed of adaptation and accuracy of the estimate caused by the on-line update of the learning rate. This section provides a study of the effect of parameter update along with the description of the inherent *saturated pixels* phenomenon occurring in environments with fast changes in illumination.

The common learning rate between the mean and variance updates in Eqs. (3.15) and (3.16) leads to a trade-off between the error in the estimate and the time of adaptation. In their seminal paper [233], Stauffer and Grimson use a single learning rate β defined as

$$\beta = \alpha \, \mathcal{N}(\mathbf{x}|k; \boldsymbol{\theta}_k) \, . \tag{3.23}$$

Even though the algorithm is robust in controlled environments, and in particular when background changes are slow, the algorithm fails to maintain an accurate pdf when the scene undergoes severe illumination changes. This limitation has been acknowledged by the authors [233]:

> "The tracker was relatively robust to all but relatively fast lighting changes."

Lee proposed a modified approach to address the aforementioned shortcoming with the implementation of a variable learning rate β [148]. The rate β is increased in the initial learning phase of the algorithm, hence providing a quicker adaptation when a new surface appears and, in particular, in the first few frames after the initialization of a new Gaussian. The modified learning rate proposed by Lee is as follows:

$$\beta_k = \left(\frac{1 - \alpha}{c_k} + \alpha \right) q_k \, . \tag{3.24}$$

where c_k is a counter incremented with the posterior probability, *i.e.*, $c_k \leftarrow c_k + q_k$. After the initial learning phase, the adaptive rate tends toward the value defined by Stauffer and Grimson.

Lee's formulation of the GMM parameter update raises an issue regarding the update of the variance. Indeed, whilst increasing the learning rate in the transient phase is adequate for the update of the mean, it becomes problematic with the variance as it is updated with a quadratic quantity, $(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{x} - \boldsymbol{\mu})$, which tends to increase rapidly. A large value of β thus leads to a quick degeneracy of the Gaussian when the variance becomes too large. The degeneracy means that any incoming pixel matches a Gaussian component. A temporary overestimation of a Gaussian component variance jeopardizes the stability of the variance estimate: the large value of the variance increases the spread of the Gaussian component and reduces the Mahalanobis distance that defines a match between a pixel and the Gaussian density. In turn, the matching pixel increases the variance estimate. The Gaussian expands until it covers the entire range of possible values for the given pixel. Every value will then match a unique Gaussian, and the pixel location reaches a saturated state; that is, the pixel will always be assigned to the same Gaussian component regardless of whether it belongs to the background or foreground. For lack of better term, we denote this phenomenon a *saturated pixel*. The pixel can either become a false foreground or a false background pixel, depending on the weight of the Gaussian. The phenomenon is illustrated in Fig. 3.2. The sequence of images from the video *HighwayII* displays the foreground extraction of a vehicle blob throughout time. The zone marked by the square is saturated, *i.e.*, most of the pixels inside the square have a degenerated variance, resulting in an absence of object detection since the underlying Gaussian is classified as background. The vehicle is not detected in the saturated zone because the value of the variance with Lee's approach is overestimated. The system recovers the detection of the moving object only after it leaves the saturated zone.

Fig. 3.3 represents the detection mask over time: Fig. 3.3(a) presents the initial position of the object in frame 140 and Fig. 3.3(b), the final position in frame 149. The object is segmented in both frames. Fig. 3.3(c) is the marginal mask over time,

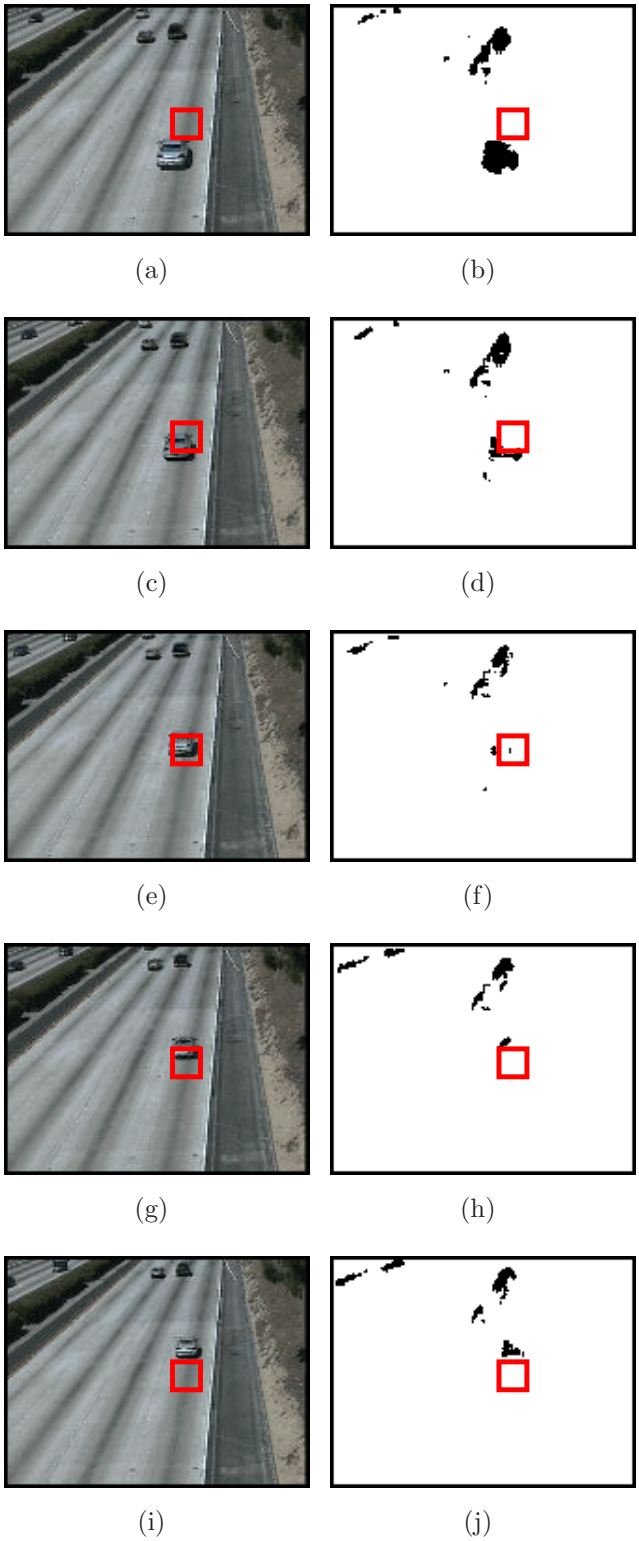
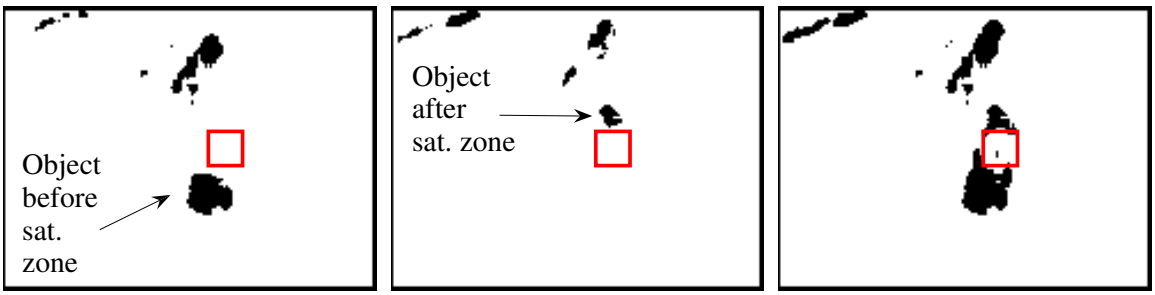
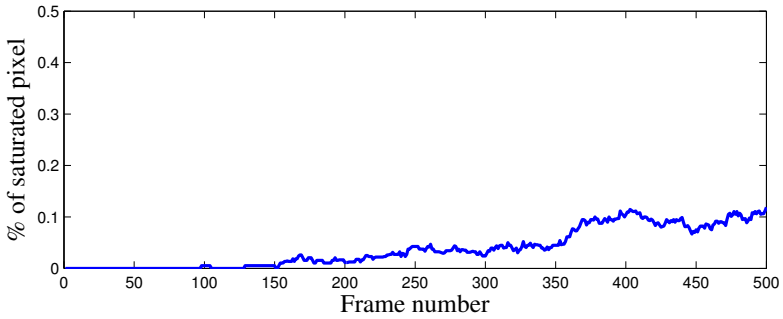


Figure 3.2 Original (a, c, e, g, i) and foreground segmentation (b, d, f, h, j) of an object passing through a saturated zone in the *HighwayII* sequence (frames 140, 142, 144, 146 and 148). The red rectangle delineates a particular saturated region.

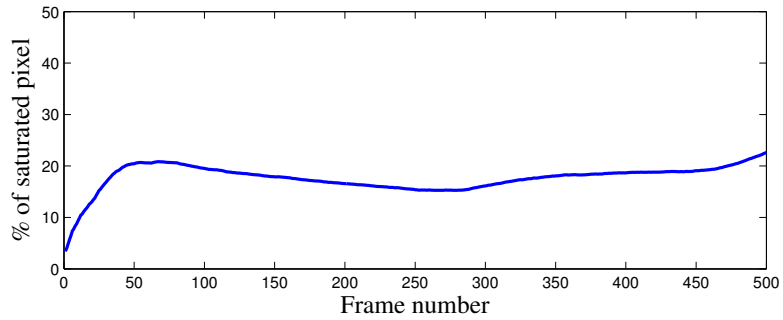


(a) (b) (c)

Figure 3.3 The pixel saturation phenomenon of Lee's method. 3.3(a) and 3.3(b) show the position of the object before entering the saturated zone (frame 140) and after leaving the saturated zone (frame 149) in the video sequence *HighwayII*, respectively. 3.3(c) displays the sum of foreground mask for frames 140 to 149 and the saturated zone delineated by the red square.



(a) Stauffer's Method



(b) Lee's method

Figure 3.4 Percentage of saturated pixels in a video sequence, *i.e.*, pixels classified as foreground or background regardless of the surface in presence. The trade-off between speed of adaptation and saturated pixel imposes: (a) a slow adaptation for a low percentage of saturated pixels (Stauffer and Grimson) or (b) a high percentage of saturated pixels for fast adaptation (Lee).

i.e., the summation of the masks for time 140 to 149 during which the object passes through the saturated zone. The zone delineated by the square does not present any detection throughout the sequence.

Figure 3.4 presents the percentage of saturated pixels in the video sequence; the fast alternation of different surfaces leads to large variation of the variance resulting in pixel saturation for large learning rate β. The increase of the learning rate for the first few frames with Lee's method yields a large variance that saturates the pixels. After 100 frames, Lee's method presents a percentage of saturation close to 20% (Fig. 3.4(b)) while Stauffer and Grimson's method displays 0.5% saturation (Fig. 3.4(a)). The saturation is particularly strong in case of large variance due to the frequent change of surfaces from the flow of vehicles. To the best of our knowledge, this phenomenon has never been investigated and was considered as an adaptation of the background to a changing density.

Furthermore, the analysis of the time to background adaptation developed hereafter shows that a new surface is integrated in the background only after a minimum time has elapsed, rejecting the hypothesis of fast background adaptation. Let us first assume that a foreground pixel generates a new Gaussian with $w_k = \alpha$. This is a reasonable assumption since the occurrence of a foreground pixel is not predictable and does not obey any periodic pattern; therefore, it is not modeled by the density estimate.

Consider the solution of the difference equation in (3.14) for a constant posterior probability, q_k. Without loss of generality, we can omit the subscript k. Since $|1 - \alpha| < 1$, the solution of (3.14) for $t > 0$, is given by

$$w(t) = (w(0) - q)\,(1 - \alpha)^t + q\,, \tag{3.25}$$

where $w(0)$ is the initial value of $w(t)$. The time t_{min} required for $w(t)$ to reach or exceed a particular value w_{min}, is given by the following inequality:

$$w(t) = (w(0) - q)\,(1 - \alpha)^{t_{min}} + q \geq w_{min}\,. \tag{3.26}$$

Let's consider a mixture of K Gaussians with K_B Gaussians belonging to the background. Let's assume their respective weights are ordered, *i.e.*, $w_1 \geq w_2 \geq ... \geq$

$w_{K_B} \geq ... \geq w_K$. The weight $w_{K_B} = w_{min}$ is then the minimum weight of the Gaussian components belonging to the background. We have, by definition (Eq. (3.20)):

$$\sum_{i=1}^{K_B-1} w_i + w_{K_B} < \lambda. \tag{3.27}$$

Since all the weights sum up to 1,

$$\sum_{i=K_B+1}^{K} w_i \geq 1 - \lambda. \tag{3.28}$$

Minimizing the weight w_{K_B} imposes that $w_{K_B} = w_i, \forall i \geq K_B$. Therefore,

$$\sum_{i=K_B+1}^{K} w_i = (K - K_B)w_{K_B} \geq 1 - \lambda. \tag{3.29}$$

Trivially,

$$w_{min} = w_{K_B} \geq \frac{1-\lambda}{K - K_B}. \tag{3.30}$$

Solving the inequality in Eq. (3.26) yields

$$t_{min} \geq \frac{1}{\ln(1-\alpha)} \ln \left[\frac{w_{min} - q}{w(0) - q} \right] \tag{3.31}$$

Using the above equation, we can find the elapsed time required for a Gaussian component to be included in the background. Let us denote w_{min} the minimum weight of a Gaussian to be part of the background model. From Eq. (3.30), the minimum weight is $w_{min} = (1-\lambda)/(K - K_B)$, and, assuming that $w(0) = \alpha$, $q = 1$ and $\alpha \ll 1$, the time required is given by

$$t_{min} \geq \frac{1}{\ln(1-\alpha)} \ln \left[\frac{(1-K) - \lambda + K_B}{(K - K_B)(\alpha - 1)} \right] \approx \frac{1}{\alpha} \ln \left[\frac{K - K_B}{K + \lambda - (K_B + 1)} \right]. \tag{3.32}$$

Figure 3.5 shows the plot of t_{min} versus α for different values of K, $K_B = 2$ (bi-modal background) and $\lambda = 0.7$.

It is important to note that the curves represented in Fig. 3.5 set the lower bound of the time adaptation and are reached when $q_k = 1$ at all time (for a constant λ equal to 0.7). In terms of pixel value, the probability $q_k = 1$ implies the mean of the Gaussian matches exactly the pixel value at all time (which is highly implausible). In that case, for $K = 3$, the background will adapt only after 72 frames. For $K = 5$,

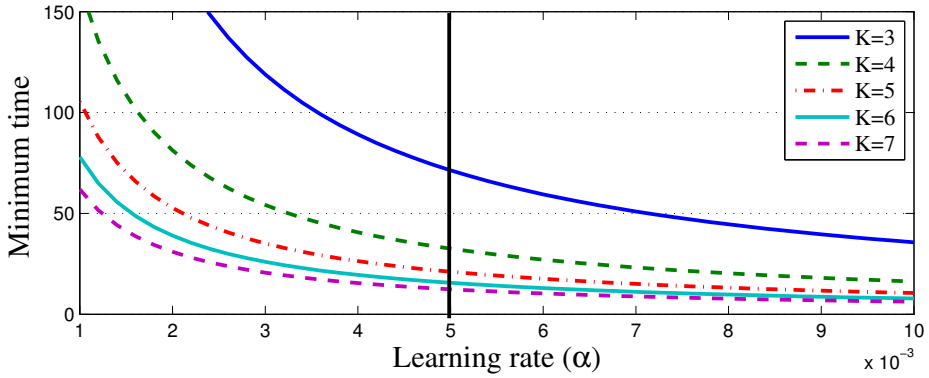


Figure 3.5 Minimum background adaptation time for a new mixture component versus the learning rate α for different values of K. The minimum time is a decreasing function of α and K.

this value drops to 33 frames. In a video sequence, it is very unlikely that an object has exactly the same value during such a number of frames, unless it is effectively static. For instance, the vehicle in Fig. 3.2 crosses the saturated zone in less than 10 frames. It follows from Eq. (3.32) that the saturated zones are not a consequence of the background adaptation but of the variance degeneracy.

To conclude, the method proposed by Stauffer and Grimson handles background subtraction with a great efficiency provided that there is no rapid changes in the background. Lee proposed an accelerated adaptation of the parameters for the early updates of a new Gaussian. However, after the transient phase, Lee's algorithm is unable to efficiently update the parameters in case of abrupt changes because the variable learning rate β_k converges to the rate β of Stauffer and Grimson's method. Furthermore, the variable learning rate also leads to variance degeneracy if the learning rate α is not set to a very low value, which defeats the purpose of Lee's method, whose aim is to accelerate learning.

3.4 Semi-Constrained Gaussian Mixture Model

In the following, we propose a new algorithm capable of handling changes in the pixel density via a fast adaptation of the mixture model. The proposed method

relies on a variable learning rate for the mean, β, and a fixed learning rate or the variance, γ. A constraint on the range of the variance is also imposed.

3.4.1 Mean Variable Learning Rate

As shown in Section 3.3.3, learning rates play a critical role in the quality of the background model. They determine how fast the Gaussian parameters can adapt to changes in the background. Consequently, the sensitivity of the algorithm as well as the speed of adaptation the background model are affected. It is therefore of paramount importance to decouple the different learning rates. While Stauffer and Grimson proposed a fixed learning rate for the mean and Lee extended the algorithm to an adaptive learning rate for the learning phase only, we propose to define an adaptive learning rate β_k for each Gaussian component, updating the parameter $\boldsymbol{\mu}_k$ that includes clues of the relative probability of a pixel belonging to the kth Gaussian as

$$\beta_k \leftarrow \min\left(\max\left(\beta_k + q_k - \frac{1}{K}\sum_{j=1}^{K} q_j, 0\right), 1\right) . \tag{3.33}$$

As a result, a Gaussian already trained and with a high posterior probability q_k will have a faster rate of update than a Gaussian in the learning stage. The rate β_k is increased if the hypothesis represented by the kth Gaussian is above the expectation of the posterior probability over the K hypotheses, and decreased otherwise. The learning encourages the less probable modes to update slower whilst modes with higher probability are updated faster. This strategy is contrary, in essence, of Lee's method. As shown earlier, a fast update rate in the learning stage jeopardizes the stability of the filter; a slow learning rate does not. If the entire mixture model is considered, the need for a fast update of the mean is unnecessary. Indeed, incoming pixels with large Mahalanobis distances from the mean, requiring a fast learning rate to improve convergence, must be modeled with a new Gaussian component. To sum up, our point of view diverges from Lee's because we believe that large corrections to the means should be carried out in the matching process and not in the update of the Gaussian mixture. It is only after several occurrences of matching pixel values that the component can be considered as modeling a relevant surface. Then, the update can be accelerated if the posterior probability $P(k|\mathbf{x}, \boldsymbol{\Theta})$ is high. It

is important to note that the value of β_k is bounded in order to retain the stability of the filter. However, the bounds are not reached in practice due to the small variations of the posterior q_k. The value of β is initialized with a small value α. However, the initialization has little impact on the segmentation results as long as it remains small.

3.4.2 Standard Deviation Learning Rate

In previous implementations of the mixture of Gaussians for background subtraction, the variance is updated at the same rate as the mean. Here, we propose to decouple the learning rates for the mean and variance and to limit the variance range. Indeed, the update of the variance should be restrained to a maximum speed of adaptation in order to limit the effect of transient states which cause the variance degeneracy as described in Section 3.3.3. A semi-parametric variance is thus to be designed, enabling a quasi-linear adaptation in case of small adjustments and a flattened response for large adjustments. The sigmoid function is used to this aim:

$$f_{a,b}(\mathbf{x}, \boldsymbol{\mu}_k) = a + \frac{b-a}{1 + e^{-s\varepsilon(\mathbf{x}, \boldsymbol{\mu}_k)}}\,, \tag{3.34}$$

where $\varepsilon(\mathbf{x}, \boldsymbol{\mu}_k)$ is defined as $\varepsilon(\mathbf{x}, \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_k)^T(\mathbf{x} - \boldsymbol{\mu}_k)$ and the update of the variance, Eq. (3.16), is modified as:

$$\sigma_k^2(t) = (1-\gamma)\,\sigma_k^2(t-1) + \gamma\, f_{a,b}(\mathbf{x}, \boldsymbol{\mu}_k(t-1))\,. \tag{3.35}$$

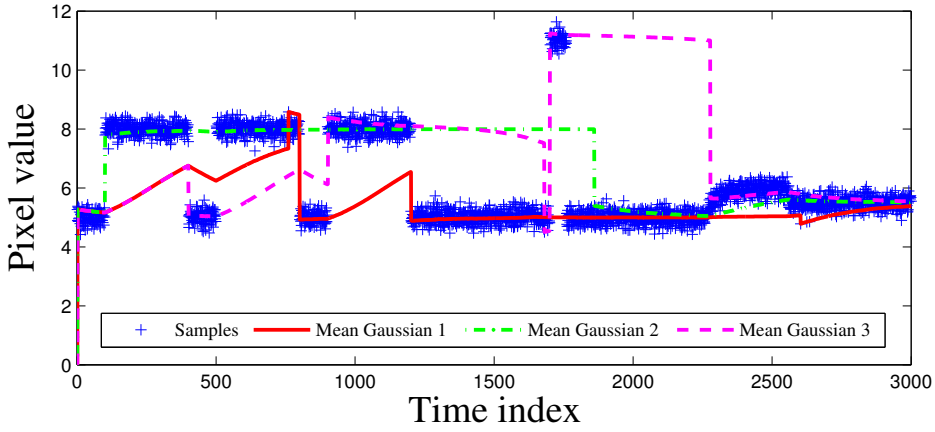
By definition, the function $\varepsilon(\mathbf{x}, \boldsymbol{\mu}_k)$ is $\Re^2 \rightarrow \Re^+$ and imposes a restriction on the function $f_{a,b}(\mathbf{x}, \boldsymbol{\mu}_k)$. Consequently, the function $f_{a,b}(\mathbf{x}, \boldsymbol{\mu})$ bounds the variance value to the domain $\mathcal{D} \in [\frac{a+b}{2}, b]$. The upper bound of the variance is justified by the nature of the mixture of Gaussian. When the value becomes too large, the Gaussian spreads over most of the pixel value range and the mixture of Gaussian becomes ineffective as all the pixels will be "phagocytosed" and merged in a unimodal density. This leads to saturated pixels. On the other extreme, a variance converging toward 0 would represent a very stable surface. The probability of a matching pixel to be part of the surface is thus very high. In this case, the incoming surface is systematically considered as a non-match when the Mahalanobis distance between the Gaussian and the value of the pixel is greater than τ (see Eq. (3.10)). This distance will

increase to ∞ when σ_k^2 tends to 0. The noise present in videos prohibits the use of such a restrictive condition. A too small variance would thus lead to the *starvation* of the Gaussian, and a decay of the weight w_k until it is replaced by a new hypothesis, even though the low variance shows a highly probable mode. To compensate for the large variations in the variance value, Lee decreased the learning rate α, hence giving less importance to incoming values. Unfortunately, this is to the detriment of the adaptation speed of the pixel density estimate. The semi-parametric definition of the variance enables fast linear update for small values and a quasi-constant rate for large values.

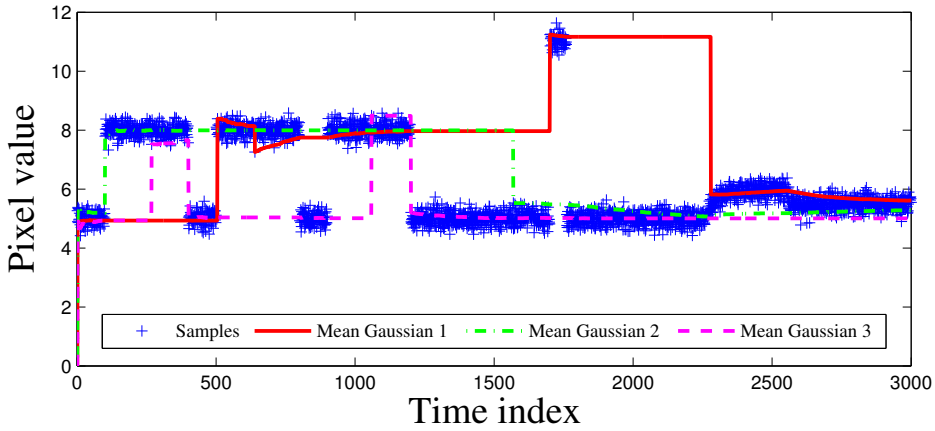
3.4.3 Performance Analysis on Synthetic Data

The system has been tested on different sets of synthetic data. A pool of synthetic data has been drawn from a normal density $\mathcal{N}(x, \mu = m(t), \sigma)$. The mean $m(t)$ is a switching process generated by different functions modeling the behavior of a changing background. It is a concatenation of bi-modal signals with different periods, modeling trees' moving in the wind or other abrupt changes in surface, and smoother signals such as a first order filter response, representing a change of illumination for instance. The three algorithms, Stauffer and Grimson, Lee and the proposed algorithm, are tested on a controlled environment to evaluate their intrinsic performance; Section 3.5 provides an evaluation in various uncontrolled environments from videos sequences.

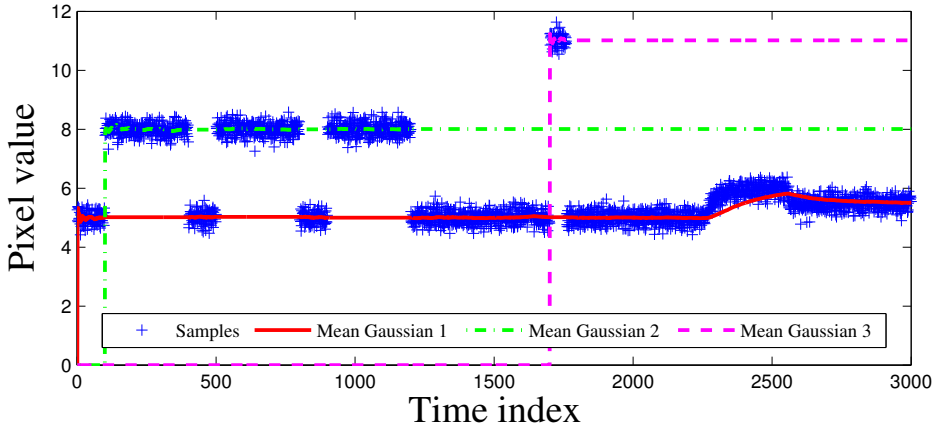
The algorithms have been tested over 20 sequences. The learning ability of the three algorithms is evaluated on the sequences described above. The Gaussian mixture for each algorithm is composed of $K = 3$ components, initialized with the same parameter values, that is, a standard deviation $\sigma_0^2 = 5$, a mean learning rate $\beta = 0.005$, a variance learning rate $\gamma = 0.6$ and a match threshold $\tau = 0.7$. Here, the standard deviation σ of the synthetic data density is equal to 0.2. Stauffer and Grimson's, Lee's and the proposed algorithm are evaluated on the synthetic sequences. The plots in Fig. 3.6 represent the adaptation of the background model over time for one of the sequences. It can be inferred that, although the density is always adequately modeled throughout the sequence with the three methods, the



(a) Stauffer and Grimson



(b) Lee



(c) Proposed

Figure 3.6 Performance of Lee, Stauffer and Grimson and the proposed approach on synthetic data. The cloud of crosses represents the samples from a normal density $\mathcal{N}(x, s, 0.2)$ through time. The Gaussian mixture model is composed of $K = 3$ components. The means μ_k are represented by the lines.

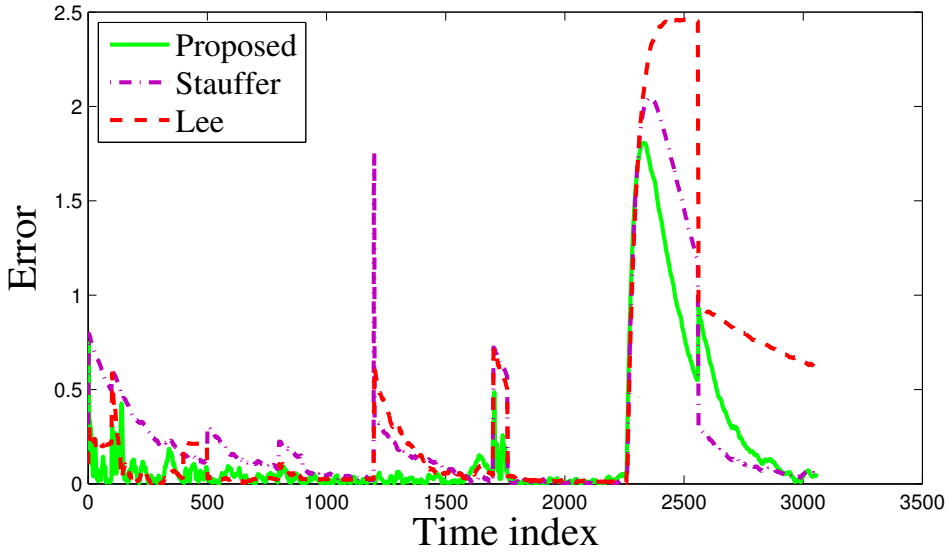


Figure 3.7 MSE between the true mean and the estimated mean of the sequence with the Gaussian mixture model for Stauffer and Grimson's, Lee's and the proposed algorithm.

proposed approach adapts faster to changes and the allocation of the Gaussians is optimized over time. Indeed, the proposed approach introduces a new Gaussian only when a new surface appears, while in the two other methods the algorithm keeps adjusting to the density. The mixture components of Stauffer and Grimson's algorithm slowly adapt to the current mode, resulting in a slow convergence. The degeneracy of the variance of Lee's technique leads to the Mahalanobis distance reduction, making the mixture component switch mode because Eq. (3.10) does not holds. The switching is the result of a lack of constraint on the standard deviation and a fast adaptation of the mean. The three algorithms have also been compared in terms of speed of adaptation and accuracy of background model. Figure 3.7 displays the mean square error (MSE) defined as the Euclidian distance between the true mean and the mean of the Gaussian modeling the data density. Firstly, it should be noted that the variable learning rate introduced in [148] improves the convergence time of Lee's model in the initial learning phase compared to Stauffer and Grimson's model. However, after the initial learning phase, Lee's method shows the same adaptation rate as Stauffer and Grimson's method. Secondly, the proposed algorithm performs better throughout the entire sequence. It is also more robust

to the variability in the density due to more efficient estimation of the Gaussian parameters for each mode (Fig. 3.6(c)). It is worthwhile noting that Stauffer and Grimson's method displays a smaller error from frame 2560 onwards due to the algorithm limited speed of adaptation. Indeed, it can be observed from Fig. 3.6(a) that the slow adaptation of the algorithm to the gradual change happens to make the mean of a Gaussian perfectly match the new mean of the sequence at frame 2560. The lack of adaptation of the algorithm makes the true mean of the switching process converge to one of the Gaussian means. Nevertheless, the proposed method still presents a steeper slope than the two others due to the variable learning rate: at frame 3000, the proposed algorithm shows an error comparable to Stauffer and Grimson's while Lee's method is still recovering.

3.5 Experiment Results

The proposed system has been tested on video sequences to evaluate the quality of foreground segmentation. A comparative analysis with Stauffer and Grimson's and Lee's methods is conducted. First, a short description of the dataset, highlighting the characteristics of each video sequence, is provided. Second, background/foreground segmentation is performed on controlled changes in illumination for objective qualitative and quantitative comparison between the algorithms. Third, the algorithms are run on uncontrolled environment, *i.e.* , natural changes in illumination for qualitative analysis and validation of the proposed technique.

3.5.1 Experimental Setup

The three algorithms have been tested on indoor and outdoor data. The data is divided into four subsets: outdoor publicly available sequences, people walking surveillance, vehicle traffic surveillance and a collection of indoor video sequences. The entire dataset represents several hours of footage with different camera settings and illumination conditions.

Publicly available sequences These sequences are available on the Internet[1] and

[1] *e.g.* http://www.openvisor.org

Table 3.1 GMM Parameter Initializing Values

K	σ_0	w_0	τ	α	λ	γ	a	b	s
5	5	0.05	2.5	0.05	0.7	0.7	-8	20	0.005

have been used as benchmarks in various research projects. The dataset is composed of five videos: *HighwayII*, *Campus*, *Laboratory*, *Office_1* and *Campus*. *HighwayII* and *Laboratory* video sequences were already used for foreground segmentation purposes in [283] and are therefore processed for comparison purposes. *Office_1*, exhibits a large portion of the background covered by the foreground throughout the video, making the learning of the background more challenging.

People walking sequences The dataset is composed of ten videos and represents pedestrians walking in open environments. The *People_Walking_x* subset has been chosen in the experiments for the artefacts introduced by the compression and the automatic cuts operated by the video surveillance system, when no motion is detected in the scene.

Vehicle traffic surveillance sequences The dataset is composed of fifteen videos of vehicles on a highway. The *Traffic_Monitoring_x* subset includes video-surveillance sequences selected for the sudden changes in weather conditions and changes in illumination due to the activation of the white balance (WB) setting on the camera. This dataset will be further described in Subsection 4.6.1.

Indoor video sequences The dataset includes four video sequences of meeting room in indoor environment. The *Long_Room_x* subset is composed of indoor scenes with ceiling lighting variations, resulting in severe changes in illumination of the background and the moving objects.

The video sequences are segmented with Stauffer and Grimson's, Lee's and the proposed algorithm. It should be noted that for comparison purposes all constants are set to the same value in all algorithms. Table 3.1 summarizes the parameter values used in the experiments.

3.5.2 Controlled Environment

The three systems have been tested on the videos described above for foreground extraction. The illumination is artificially modeled in the set of video sequences to control the performances of each system with regard to the changes. First, the original video sequences are segmented to provide a pseudo ground truth of the foreground. Then, the videos are modified by embedding changes in illumination. A framework to analyze the changes in illumination is thus set up. Because the lighting variations are controlled, it is possible to qualitatively and quantitatively analyze the influence of the changes in illumination on the performance of the algorithms. The process also ensures that poor segmentation resulting from difficult extractions of the motion is detected (in the original video sequence). The changes in illumination represent intermittent partial occlusion of the source of light such as a series of clouds covering the sun or people passing before the lamp in an indoor environment. Such disturbances are modeled as a fast but smooth change due to the gradual transition from penumbra and umbra as described in [158]. Consequently, a bi-modal density cannot effectively describe the change. We model the illumination variations over the entire frame $\mathcal{I}$ as an additive sinusoidal component:

$$\mathcal{I}(t) \leftarrow \begin{cases} 255 & \text{if } \mathcal{I}(t) + 20\cos(\frac{2\pi}{100}t) \geq 255, \\ 0 & \text{if } \mathcal{I}(t) + 20\cos(\frac{2\pi}{100}t) \leq 0, \\ \mathcal{I}(t) + 20\cos(\frac{2\pi}{100}t) & \text{Otherwise}. \end{cases} \tag{3.36}$$

The results presented in this subsection are the raw foreground segmentations of the video sequences with no additional post-processing but a median filtering performed by a kernel of size 3×3. Figure 3.8 compares the performance of the three algorithms on the original and the modified *HighwayII* video. *HighwayII* has been selected for the excellent segmentation on the original sequence. The foreground extraction is evaluated by summing the number of pixels classified as foreground throughout the video sequence. The foreground extraction from the original sequence serves as the reference segmentation, *i.e.*, pseudo ground truth. The segmentation results were consistent with all three methods on the original video. By contrast, only the proposed method was able to accurately segment the modified video (Fig. 3.8(c)). Indeed, Stauffer and Grimson's algorithm is incapable of updating the background

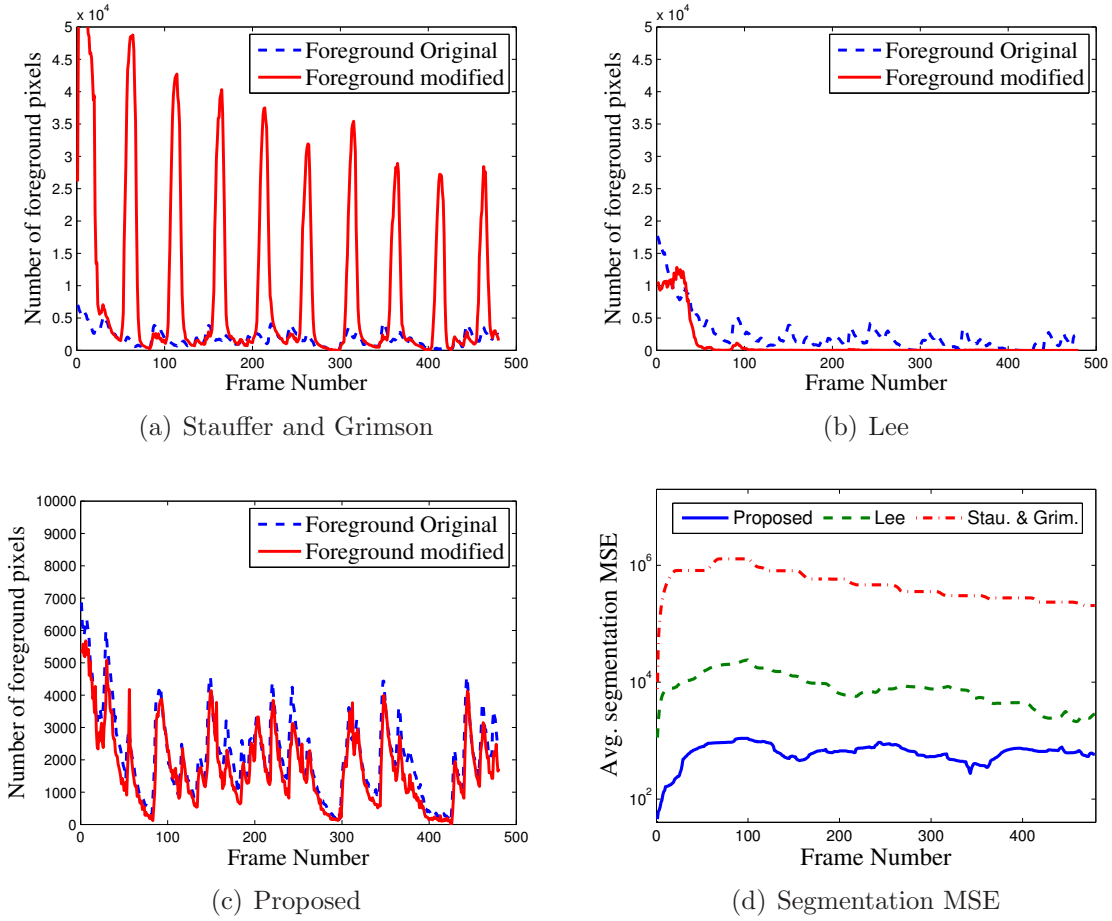
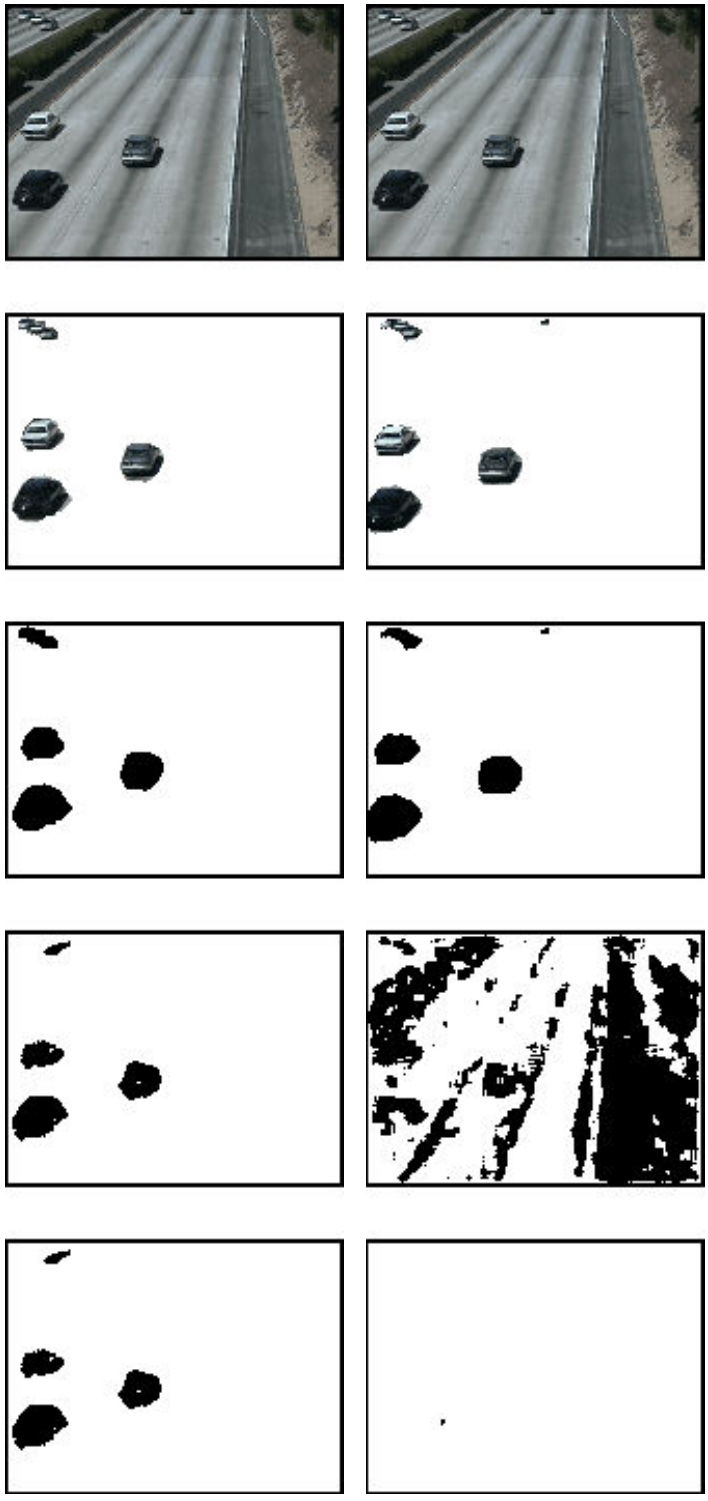


Figure 3.8 Plots of the number of foreground pixels for the original (*dashed line*) and the modified (*plain line*) *HighwayII* video sequence.

model fast enough and portions of sines, deemed to represent the changes in illumination, are falsely detected as foreground (Fig. 3.8(a)). Conversely, the Gaussian variances in Lee's algorithm quickly degenerate, resulting in partial saturation of the image and partial foreground detection (Fig. 3.8(b)). Figure 3.8(d) displays the average MSE in the pixel count between the original and the modified video sequence.

Figure 3.9 displays the results of segmentation for frame 328 of the *HighwayII* video sequence and shows the limitation of the two other methods in adapting to fast changes. Figure 3.9(a) is the foreground segmentation of the original image and Figure 3.9(b) is the segmentation of the video sequence with changes in illumination. Figure 3.10 presents segmentation results of the video sequence *People_Walking_1*.



(a) Original (b) Modified

Figure 3.9 Foreground segmentation of the *HighwayII* video sequence. For each column and from top to bottom: original image; foreground extraction with the proposed method; foreground mask with the proposed method; foreground mask with Stauffer and Grimson's method; foreground mask with Lee's method.

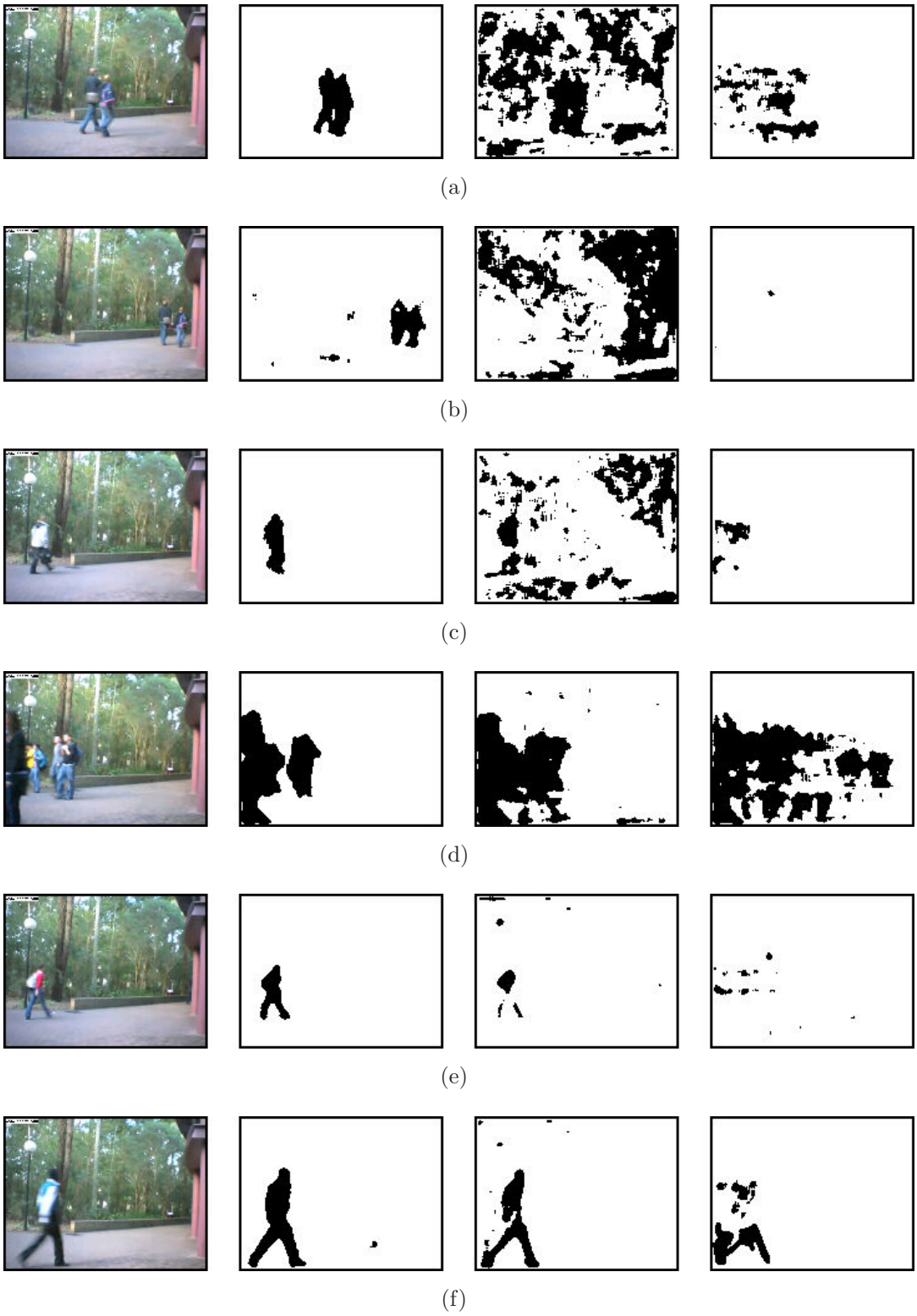


Figure 3.10 Foreground segmentation of the *People_Walking_1* video sequence. For each row and from left to right: original image; foreground mask with the proposed method; foreground mask with Stauffer and Grimson's method; foreground mask with Lee's method.

Figures 3.10(a) and 3.10(b) show that Stauffer and Grimson's method is not adequate for fast changes in illumination; after 28 frames, the algorithm has still not recovered from the change in illumination. It can also be inferred from Figs. 3.10(a), 3.10(b) and 3.10(c) that Lee's algorithm fails to detect motion when there is a high rate of surface changes due to the constant flow of people. The video-surveillance sequence exhibits recurrent non-periodic patterns of walking tracks, presenting a large diversity of surfaces to the pixel. Consequently, the variance will increase rapidly resulting in saturated zones. Lee's method is incapable of recovering throughout the video sequence. Figures 3.10(d), 3.10(e) and 3.10(f) display the segmentation of the foreground after recovery of the illumination change by Stauffer and Grimson's algorithm: the segmentation is altered compared to the proposed approach.

The three algorithms have also been evaluated on indoor scenes. Figure 3.11 shows that the proposed algorithm provides the best segmentation results. In Fig. 3.11(a), the segmentation of the person presents the same alteration as in Fig. 3.10(e) for the reason stated before. Figure 3.11(b) emphasizes the ability of the proposed algorithm to adapt to a new density. During the *Laboratory* sequence, a closet door is open, presenting a new surface to the related pixels. The change is quickly integrated by the proposed algorithm, but not by the other two algorithms. It is also worthwhile noting that Lee's algorithm has better performance on video sequences that show low foreground/background ratio, especially in the initialization phase. For instance, *Laboratory* sequence does not present any foreground in the first few seconds of the video whilst *People_Walking_1* sequence does, leading to better segmentation in the first case (see, *e.g.*, Fig. 3.11(a)) than in the second one (*e.g.* Fig. 3.10(a)). Finally, Fig. 3.11(c) displays a person walking slowly in the *Office_1* video sequence. The proposed method provides a complete capture of the motion. The person is not included in the background despite the homogeneity of the color of the clothes.

To conclude, it has been observed that the algorithm proposed by Lee provides robust segmentation when a complete representation of the background is available in the initialization phase. However, if there are recurrent changes in the video (*e.g.* path or illumination), it will lead to saturated zones and, consequently, poor segmentation of the foreground. Stauffer and Grimson's algorithm is unable to

(a) (b) (c)

Figure 3.11 Foreground segmentation for office scenes. (a) and (b) are from *Laboratory*; (c) is from *Office_1*. For each column and from top to bottom: original image; foreground extraction with the proposed method; foreground mask with the proposed method; foreground mask with Stauffer and Grimson's method; foreground mask with Lee's method.

model the density with lighting changes due to the non-adaptive learning rate. The proposed algorithm is the only one that can consistently handle fast changes in illumination.

3.5.3 Natural Changes in Illumination

The proposed technique is tested on video sequences showing changes in illumination. In this case, the lighting conditions are not controlled and the underlying true surface density is not known. This makes the analysis of the algorithm difficult because it is impossible to guarantee that the falsely extracted foreground is actually due to the changes in illumination; some other phenomena, spatio-temporally aligned with the changes in illumination, could be the cause of poor segmentation. Nevertheless, we will consider this situation improbable in the sequel and focus on two different types of background subtraction: indoor and outdoor scenes. The parameters for the algorithms remain the same as in Table 3.1, except that the learning rate α is lowered to 0.005 to decrease the number of the saturated pixel with Lee's algorithm.[2]

Outdoor Scenes

The outdoor scenes analyzed here are extracted from the vehicle traffic surveillance dataset. The camera was fixed above the highway. Apart from challenges due to the low quality and low resolution of the video sequences, there are a number of observed changes in illumination reducing the quality of foreground segmentation. Figure 3.12 displays the results of segmentation. Figures 3.12(a) to 3.12(c) show different frames of the video sequence *Traffic_Monitoring_11* where the White Balance (WB) of the camera yields a global change in illumination. Stauffer and Grimson's as well as Lee's technique result in poor segmentation during and after the WB change while the proposed technique is insensitive to such changes. Figures 3.12(d) to 3.12(f) show changes in illumination due to the weather in video *Traffic_Monitoring_12*: half of the scene is shaded by a cloud while the other half is in the sunlight. The horizontal edge between shade and light (not to be confused with the lines of the road which are also detected due to the small jitter of the camera support) is moving downward

[2]Note that the value of $\alpha = 0.005$ is the learning rate adopted by Lee in [148].

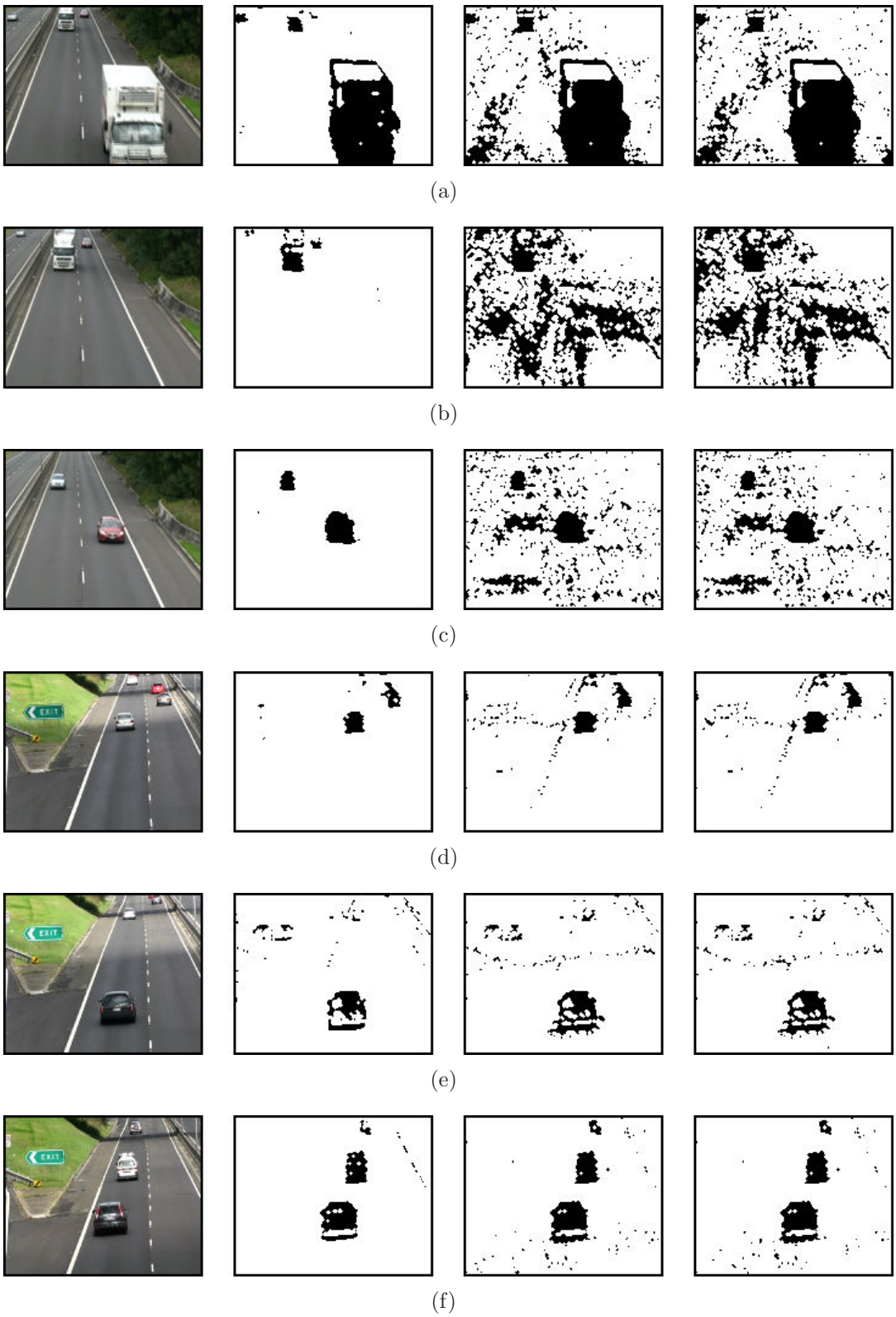


Figure 3.12 Foreground segmentation in outdoor environment. For each row and from left to right: original image; foreground mask with the proposed method; foreground mask with Stauffer and Grimson's method; foreground mask with Lee's method. (a) to (c) represent false foreground detection imputable to the lighting changes of automatic White Balance setting of the camera; (d) to (f) exhibits false foreground detection for a moving sahdow/sunlight edge.

which leads to false foreground detection in the transition area with Stauffer and Grimson's and Lee's method.

Indoor Scenes

Indoor scenes are usually considered more challenging than outdoor scenes because noise is higher and illumination is weaker. Also, moving objects have more impact on the segmentation result due to the cluttered and confined nature of the environment; direct lighting and projectors create shadows impoverishing foreground segmentation. Figure 3.13 presents some results for videos *Long_Room_3* and *Long_Room_4*. The videos undergo severe changes in illumination in a dark and noisy environment; the toughest setting for foreground extraction. The segmentation of the objects is incomplete with the three methods because the level of lighting is very low; dark pixels are misclassified as background. However, the proposed method handles the fast changes in illumination better than the two other techniques as it is able to adapt quickly to changing density without generating saturated pixels. The trade-off imposed by a common learning rate for Stauffer and Grimson's and Lee's method precludes their use in such difficult environments.

3.6 Summary of the Gaussian Mixture Model for Background Modeling

This chapter was dedicated to the parametric representation of densities with Gaussian mixture model, where the pdf estimate is characterized by a set of parameters $\boldsymbol{\Theta}$. We investigated the use of the parametric representation to model background in videos. A new algorithm for foreground extraction handling fast changes in background density was presented.

An investigation was conducted on the limitations of a shared learning rate for the parameters (mean and variance) of the Gaussians in the mixture. It was shown that a trade-off was imposed on the speed of adaptation and the accuracy of the parameter estimates by a common update rate. A fast update of the parameters creates *saturated pixels* while a slow update fails to adapt to fast changes in the

Figure 3.13 Foreground segmentation in indoor environment. For each row and from left to right: original image; foreground mask with the proposed method; foreground mask with Stauffer and Grimson's method; foreground mask with Lee's method. The changes in illumination provoke background surfaces to be detected as foreground with Stauffer and Grimson's and Lee's algorithm.

pixel density. To address this issue, the parameters of the mixture are updated with independent learning rates, improving the convergence of estimated parameters to their true value. The proposed algorithm handles fast changes in pixel density for controlled and uncontrolled environments better than two existing algorithms. In particular, Stauffer and Grimson's and Lee's algorithms are unable to accurately segment the background from the foreground with lighting changes (*e.g.* white balance adjustments, sunlight/shadow edges or global variations in illumination).

The proposed method provides better motion segmentation consistently on indoor and outdoor sequences. This improvement is fundamental because subsequent tasks will be carried out more accurately, since foreground detection is a low-level process. In particular, vehicle tracking can benefit from the segmentation to achieve better performance in challenging environments. The proposed foreground extraction method will be integrated in the vehicle tracking system presented in Chapter 4.

Chapter 4

Projective Kalman Filter for Vehicle Tracking

4.1 Introduction

Vehicle tracking has been a focus of attention in recent years due to increasing demand in visual surveillance and security on highways. The increase in computing power as well as the low cost of video processing embedded systems have made real-time vehicle tracking in video sequences an accessible technology. The area of Intelligent Transportation Systems covers a wide range of automated tasks for which robust vehicle tracking is crucial. Vehicle tracking is an elementary task at the bottom-end of the system. Accurate trajectory extraction provides essential statistics for traffic control, such as speed monitoring, vehicle count and average vehicle flow. The current infrastructure for the acquisition of such statistics is prohibitively costly to implement. For example, the installation of inductive loop sensors generates traffic perturbations that cannot always be afforded in high traffic areas. Also, robust video tracking opens new prospects such as vehicle identification and customized statistics that are not available with current technologies, *e.g.*, suspect vehicle tracking or differentiated vehicle speed limits. At the top-end of the system are high level-tasks such as event detection (*e.g.*, accident and animal crossing) or traffic regulation (*e.g.*, dynamic adaptation and lane allocation). Robust vehicle tracking is therefore necessary to ensure effective performance of high-level tasks.

In the framework of hidden Markov chains, recursive Bayesian filtering has been extensively implemented for vehicle tracking; in particular Kalman filters [16, 65, 128, 149] and particle filters [135, 171]. Kalman filters have been a particular focus of attention because of their implementation simplicity and relatively low computation cost. Some authors have modeled the state vector with data such as kinematic parameters [16, 89, 136, 172, 289] or scale [128], directly available from foreground blobs. Other authors proposed to further process the image and extract corners [200] or contours [136, 154] that are then fed into the Kalman filter. Tracking can also be achieved without an explicit recursive kinematic model. For instance, Choi *et al.* [50] used a quad-tree scale invariant segmentation and a template matching technique to achieve tracking of vehicles.

This chapter presents a new tracking algorithm based on background subtraction, mean-shift and the Kalman filter to improve the quality and robustness of vehicle tracking on highways. The main contribution is the implementation of the projective Kalman filter (PKF) integrating inference on the characteristics of the traffic surveillance system. The linear fractional transformation that maps the real trajectory of the vehicle to the apparent trajectory on the camera plane is developed in Section 4.2. The Kalman filter and its extensions are then introduced in Section 4.3. The framework of the projective Kalman filter integrating the fractional linear transformation into the Kalman filter is set up in Section 4.4. The vehicle tracking system is presented in Section 4.5 and the tracking results are presented in Section 4.6.

4.2 Constraining the Tracking with the Environment

The task of vehicle tracking can be approached as a specific application of object tracking in a constrained environment. Indeed, vehicles do not evolve freely in their environment but follow particular trajectories. This section presents the motivations, that is, the constraints imposed upon the vehicle trajectories in the image, and introduces the linear fractional transformation, also called projective transfor-

mation, mapping the scene onto the camera plane.

4.2.1 Motivations

Vehicle tracking from traffic monitoring presents particular characteristics due to the nature of the video sequences and the vehicle trajectories compared to other object tracking tasks:

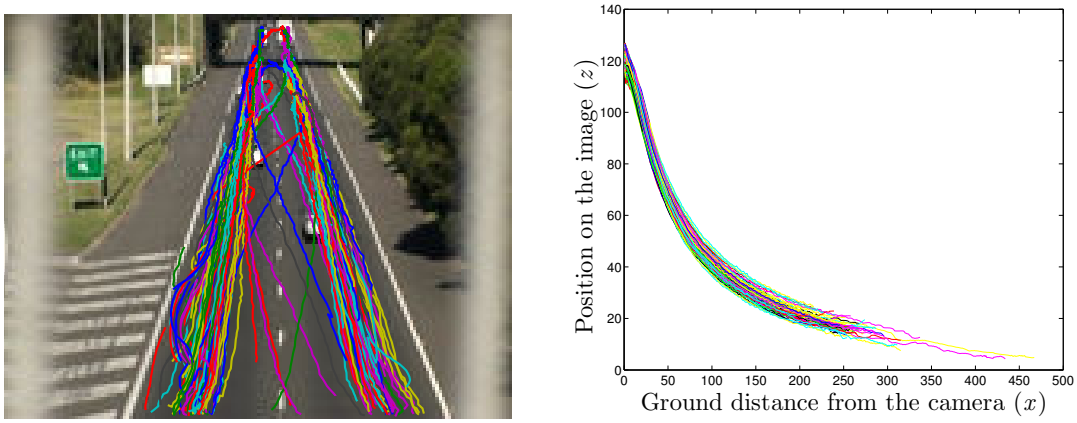
Low definition and highly compressed videos. Traffic monitoring video sequences are often of poor quality because of the inadequate infrastructure of the acquisition and transport system. Therefore, the result is a restricted bandwidth only allowing low bit flows and the presence of artefacts generated by compression;

Very low frame rate. The very low frame rate is also due to the infrastructure of the network. It makes the information about the position of the vehicle sparse due to the restricted bandwidth. A fine estimation of the position is thus necessary to ensure robust tracking;

Slowly-varying vehicle speed. A common assumption in vehicle tracking is the uniformity of the vehicle speed. The narrow angle of view of the scene and the short period of time a vehicle is in the field of view justify this assumption, especially when tracking vehicles on a highway;

Constrained real-world vehicle trajectory. Normal driving rules impose a particular trajectory on the vehicle. Indeed, the curvature of the road and the different lanes constrain the position of the vehicle. Figure 4.1 also points out the pre-defined pattern of vehicles trajectories resulting from projective constraints that can be used for vehicle tracking; and

Projection of vehicle trajectory on the camera plane. The trajectory of the vehicles on the camera plane undergoes severe distortion due to the low elevation of the traffic surveillance camera. The curve described by the position of the vehicle is asymptotic and converges to the vanishing point.



(a) (b)

Figure 4.1 Examples of vehicle trajectories from a traffic monitoring video sequence. Most vehicles follow a pre-determined path: (a) Vehicle trajectories in the image; (b) Vehicle positions in the image *w.r.t.* the distance from the monitoring camera.

We propose to exploit the aforementioned characteristics in this chapter in order to improve the robustness and accuracy of vehicle tracking.

4.2.2 Linear Fractional Transformation

An important characteristic of traffic video sequences is the severe distortion in the vehicle trajectory caused by the low elevation of the surveillance camera. The linear fractional transformation, also called homographic transformation, has been implemented to compensate for the distortion of the projection on the camera plane [186,194,287]. In [149], a calibration of the system is performed in order to linearize the trajectory of the vehicle. Recently, Kanhere and Birchfield have considered a homographic transformation to recover the 3 dimensions of the real world [133] from the 2 dimensions projection on the camera plane through the so-called *Plumb-Line Projection*. The height of the vehicle center is thus recovered and the ground distance of the object can be evaluated. This method results in a better estimation of the vehicle position. The linear fractional transformation has also been extensively used in feature matching [90, 129], image registration [37, 76, 163], and 3-D scene modeling [85,86,188,198,218]. The fractional transformation is used to compensate the homographic projection of the position of the vehicle on the road (d-axis) onto the camera plane (d_p-axis) as shown on Fig. 4.2. In this subsection, we show that

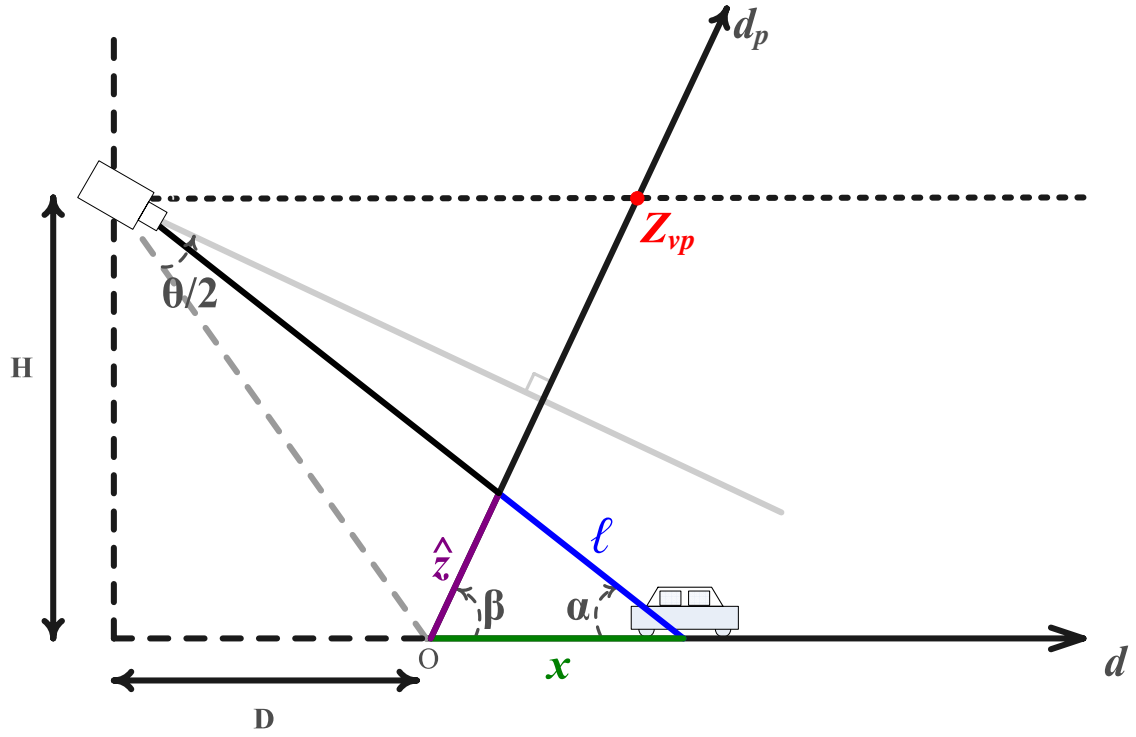


Figure 4.2 Projection of the vehicle on a plane parallel to the image plane of the camera. The graph shows a cross section of the scene along the direction d (tangential to the road).

the trajectory of a vehicle follows a homographic transformation of the form:

$$z = \frac{\lambda_1 x + \lambda_2}{\lambda_3 x + \lambda_4}\,, \tag{4.1}$$

where λ_i's are constant coefficients. The distortion of the vehicle trajectory on the camera plane happens along the d-axis. The homography projects the physical trajectory onto the camera plane as shown in Fig. 4.2. For practical implementation, it is useful to express the projection in terms of video footage parameters that are easily accessible. The projection of trajectories along the tangential direction d onto the d_p axis is determined by the following parameters:

- Angle of view (θ),
- Height of the camera (H), and
- Ground distance (D) between the camera and the first location captured by the camera.

It can be deduced from Fig. 4.2 that

$$\hat{z}^2 = x^2 + \ell^2 - 2x\ell\cos\alpha\,, \tag{4.2}$$

and

$$\ell^2 = \hat{z}^2 + x^2 - 2x\hat{z}\cos\beta\,, \tag{4.3}$$

where $\cos\alpha = D + x/\sqrt{(H^2 + (D+x)^2)}$ and $\beta = \arctan(D/H) + \theta/2$. After substituting Eq. (4.3) in Eq. (4.2) and squaring, we obtain

$$(x\cos\alpha)^2\left(\hat{z}^2 + x^2 - 2x\hat{z}\cos\beta\right) = \left(x^2 - x\hat{z}\cos\beta\right)^2. \tag{4.4}$$

Grouping the terms in $\hat{z}$ to get a quadratic form leads to:

$$\begin{aligned}\hat{z}^2x^2(\cos^2\alpha - \cos^2\beta) + 2\hat{z}x^3\cos\beta(1-\cos^2\alpha)\\ + x^4(\cos^2\alpha - 1) = 0\,.\end{aligned} \tag{4.5}$$

After discarding the non-physically acceptable solution, one gets

$$\hat{z}(x) = \frac{xH}{(D+x)\sin\beta + H\cos\beta}\,. \tag{4.6}$$

Furthermore, because $D \gg H$ and θ is small in practice, the angle β is approximately equal to $\pi/2$ and, consequently, Eq. (4.6) simplifies to $\hat{z} = xH/(D+x)$. Note that this result can be verified using the triangle proportionality theorem. Finally, we scale $\hat{z}$ with the position of the vanishing point Z_{vp} in the image to find the position of the vehicle in terms of pixel location[1], and define the projection function h_x as

$$z = h_x(x) = \hat{z}(x) \times \frac{Z_{vp}}{\lim\limits_{x\to\infty}\hat{z}(x)} = \hat{z}(x) \times \frac{Z_{vp}}{H}\,. \tag{4.7}$$

The projected speed and the observed size of the vehicle in the camera plane are also important variables for the problem of tracking and are thus necessary to derive. These measures are integrated in the projective Kalman filter (see Subsection 4.4). They can be directly extrapolated from the position of the object in the camera plane. The observed speed of the vehicle $\dot{z}$ is defined as:

$$\dot{z} = z_t - z_{t-1} = \frac{D\dot{x}}{(x+D)(x-\dot{x}+D)}\,. \tag{4.8}$$

[1]The position of the vanishing point can be approximated either manually or automatically [220]. Here, we manually estimate the vanishing point.

where t refers to time. When the real size of the vehicle s is known, its observed size b can also be derived from the position z as follows:

$$\begin{aligned} b &= h_x(x + \frac{s}{2}) - h_x(x - s/2) \\ &= \frac{sD}{(x+D)^2 - (s/2)^2} \,. \end{aligned} \tag{4.9}$$

The variables z, $\dot{z}$ and b are introduced in the Kalman filter to track vehicles using the projective Kalman filter, a Kalman filter integrating the projective transformation.

4.3 The Kalman Filter

The Kalman filter is presented in this section to introduce the projective Kalman filter developed in Section 4.4. The Kalman filter provides the optimal solution to the Bayesian problem stated in Subsection 2.4.2 in Gaussian and linear environment. The Gaussian framework refers to:

- Gaussian posterior density at time $t-1$, including the initial density $p(\mathbf{x}_0)$, such that:

  $$p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}) \,, \tag{4.10}$$

  where $\hat{\mathbf{x}}_{t-1|t-1}$ refers to the estimate of the state at time $t-1$ given the observation at time $t-1$;

- additive Gaussian process noise at time $t-1$:

  $$\mathcal{N}(\mathbf{v}_{t-1}; 0, \mathbf{Q}_{t-1}) \,; \tag{4.11}$$

- additive Gaussian observation noise at time t:

  $$\mathcal{N}(\mathbf{n}_t; 0, \mathbf{R}_t) \,; \tag{4.12}$$

where

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{4.13}$$

with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. This section presents the derivation of the Kalman filter, its analytical extension, the extended Kalman filter, and its numerical extension, the unscented Kalman filter.

4.3.1 Closed-form Solution to the Bayesian Problem

The linear and Gaussian environment of the Kalman filter is used to derive a closed-form solution of the Bayesian problem. The linearity constraint infers that the process and observation functions be linear, possibly time-variant, in $\mathbf{x}$. This yields the following matrix representation for the system

$$\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{v}_{t-1}\,, \tag{4.14}$$

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{n}_t\,, \tag{4.15}$$

where $\mathbf{F}_{t-1}$ and $\mathbf{H}_t$ are the matrix representations of the process and observation functions. The Gaussian framework is maintained with Eqs. (4.14) and (4.15) since a Gaussian posterior pdf at time $t-1$ ensures a Gaussian posterior pdf at time t through linearity. The Gaussian environment allows a Gaussian representation of the predicted density from the state space equations. Considering Eqs. (4.10) and (4.14), the predicted density is given by

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \tag{4.16}$$

where the predicted state $\hat{\mathbf{x}}_{t|t-1}$ and covariance matrix $\mathbf{P}_{t|t-1}$ are

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1|t-1}\,, \tag{4.17}$$

$$\mathbf{P}_{t|t-1} = \mathbf{Q}_{t-1} + \mathbf{F}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{F}_{t-1}^T\,. \tag{4.18}$$

The subscript $t|t-1$ denotes the prediction of the state $\hat{\mathbf{x}}$ at t given the observation at time $t-1$. The posterior pdf is then

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})\,, \tag{4.19}$$

with

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1})\,, \tag{4.20}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T\,, \tag{4.21}$$

$$\text{where} \qquad \mathbf{S}_t = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t\,. \tag{4.22}$$

The term $\mathbf{S}_t$ is sometimes called the covariance matrix of the innovation process $e_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$. The optimal Kalman gain is given by

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}\,. \tag{4.23}$$

The Kalman filter recursively updates the mean $\hat{\mathbf{x}}_{t|t}$ and the covariance $\mathbf{P}_{t|t}$ which entirely characterize the Gaussian posterior pdf $p(\mathbf{x}_t|\mathbf{Z}_t)$.

Although the Kalman filter is the optimal solution for the Bayesian filtering problem, it is seldom used in this form in visual object tracking. Indeed, the restrictive constraints on the system functions prevent its use in non-linear problems. The extended and the unscented Kalman filters have been developed to relax the linearity constraint and widen the scope of the Kalman filter. The extended Kalman filter provides a sub-optimal solution to the Bayesian problem by analytic approximation of the system functions, while the unscented Kalman filter offers a numerical solution. The Gaussian framework is conserved but the system equations can now be generalized as

$$\mathbf{x}_t = \mathbf{f}_{t-1}(\mathbf{x}_{t-1}) + \mathbf{v}_{t-1}\,, \tag{4.24}$$

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{n}_t\,. \tag{4.25}$$

4.3.2 The Extended Kalman Filter

The extended Kalman filter linearizes the system functions using the Jacobian matrix. The Jacobian of a function $\mathbf{g}$ evaluated at $\mathbf{x}$ is denoted $\nabla_{\mathbf{x}}\mathbf{g}$. To recover the derivation of the Kalman filter, the process function is locally linearized at $\mathbf{x}_{t-1}$ by the approximation $\hat{\mathbf{F}}_{t-1} = \nabla_{\hat{\mathbf{x}}_{t-1|t-1}}\mathbf{f}_{t-1}(\hat{\mathbf{x}}_{t-1|t-1})$ replacing $\mathbf{F}_{t-1}$ in the Kalman filter framework. The observation function is linearized as $\hat{\mathbf{H}}_t = \nabla_{\hat{\mathbf{x}}_{t|t-1}}\mathbf{h}_t(\hat{\mathbf{x}}_{t|t-1})$. The derivation of the EKF is identical to that of the Kalman filter, after substitution of the system function approximations, except Eqs. (4.17) and (4.20) where the terms $\mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$ need not be approximated and are therefore evaluated as $\mathbf{f}_{t-1}(\hat{\mathbf{x}}_{t-1|t-1})$ and $\mathbf{h}_t(\hat{\mathbf{x}}_{t|t-1})$, respectively.

The extended Kalman filter provides an estimate of the posterior pdf to the first order of non-linearities through the estimation of the Jacobian. The EKF is the most

employed filter in object tracking since it handles non-linearities and has a relatively low computation cost. The analytic solution makes possible the computation of the Jacobians before the process starts if tracking is restricted to time-invariant system functions. This is the case in most visual object tracking applications.

4.3.3 The Unscented Kalman Filter

The Unscented Kalman Filter provides a solution to relax the linearity constraint imposed on the Kalman filter by numerical approximation. The UKF captures the first and second order non-linearities of the system equations and, therefore, offers better performance than the EKF for highly non-linear problems. The filter is named after the unscented transform (UT) that it relies upon [8].

The unscented transform is a linearization of the system function that carries the statistics of the density undergoing the transform via *sigma points*. In the Gaussian framework, an appropriate choice of sigma points capture the mean and the covariance of the density. Sigma points are samples of the density at particular values. Let us assume a generic random variable $\mathbf{x}$ of dimension $n_{\mathbf{x}}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The UT produces a set of $2n_{\mathbf{x}}+1$ sigma points $\{\mathcal{X}_0, ..., \mathcal{X}_{2n_{\mathbf{x}}}\}$, selected around the mean, and an associated set of weights $\{\mathcal{W}_0, ..., \mathcal{W}_{2n_{\mathbf{x}}}\}$ as follows

$$\mathcal{X}_0 = \boldsymbol{\mu} \qquad \mathcal{W}_0 = \frac{\kappa}{(n_{\mathbf{x}}+\kappa)} \quad i = 0 \tag{4.26}$$

$$\mathcal{X}_i = \boldsymbol{\mu} + \left(\sqrt{(n_{\mathbf{x}}+\kappa)\boldsymbol{\Sigma}}\right)_i \qquad \mathcal{W}_i = \frac{\kappa}{2(n_{\mathbf{x}}+\kappa)} \quad i = 1, ..., n_{\mathbf{x}} \tag{4.27}$$

$$\mathcal{X}_i = \boldsymbol{\mu} - \left(\sqrt{(n_{\mathbf{x}}+\kappa)\boldsymbol{\Sigma}}\right)_i \qquad \mathcal{W}_i = \frac{\kappa}{2(n_{\mathbf{x}}+\kappa)} \quad i = n_{\mathbf{x}}+1, ..., 2n_{\mathbf{x}} \tag{4.28}$$

where κ is a scaling parameter and $\left(\sqrt{(n_{\mathbf{x}}+\kappa)\boldsymbol{\Sigma}}\right)_i$ represents the ith row of the covariance matrix square root $\mathbf{J}$ such that $\mathbf{J}^T\mathbf{J} = (n_{\mathbf{x}}+\kappa)\boldsymbol{\Sigma}$. In the unscented Kalman filter, the sigma points $\mathcal{X}^i_{t-1}$ are spread around the estimate of the mean $\hat{\mathbf{x}}_{t-1|t-1}$ at time $t-1$. This yields the predicted density given in Eq. (4.16) with

$$\mathcal{X}^i_{t|t-1} = \mathbf{f}_{t-1}(\mathcal{X}^i_{t-1}), \tag{4.29}$$

$$\hat{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2n_{\mathbf{x}}} \mathcal{W}^i_{t-1}\mathcal{X}^i_{t|t-1}, \tag{4.30}$$

$$\mathbf{P}_{t|t-1} = \mathbf{Q}_{t-1} + \sum_{i=0}^{2n_{\mathbf{x}}} \mathcal{W}_{t-1}^{i} \left(\mathcal{X}_{t|t-1}^{i} - \hat{\mathbf{x}}_{t|t-1}\right) \left(\mathcal{X}_{t|t-1}^{i} - \hat{\mathbf{x}}_{t|t-1}\right)^{T} , \tag{4.31}$$

$$\hat{\mathbf{z}}_{t-1|t} = \sum_{i=0}^{2n_{\mathbf{x}}} \mathcal{W}_{t-1}^{i} \mathbf{h}_{t}(\mathcal{X}_{t|t-1}^{i}) . \tag{4.32}$$

The update step leads to the posterior density in (4.19) with

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_{t}(\mathbf{z}_{t} - \hat{\mathbf{z}}_{t|t-1}) , \tag{4.33}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_{t}\mathbf{S}_{t}\mathbf{K}_{t}^{T} . \tag{4.34}$$

The innovation covariance matrix and the Kalman gain are given by

$$\mathbf{S}_{t} = \sum_{i=0}^{2n_{\mathbf{x}}} \mathcal{W}_{t-1}^{i} \left(\mathbf{h}_{t}(\mathcal{X}_{t|t-1}^{i}) - \hat{\mathbf{z}}_{t|t-1}\right) \left(\mathbf{h}_{t}(\mathcal{X}_{t|t-1}^{i}) - \hat{\mathbf{z}}_{t|t-1}\right)^{T} + \mathbf{R}_{t} , \tag{4.35}$$

$$\mathbf{K}_{t} = \left(\sum_{i=0}^{2n_{\mathbf{x}}} \mathcal{W}_{t-1}^{i} \left(\mathcal{X}_{t|t-1}^{i} - \hat{\mathbf{x}}_{t|t-1}\right) \left(\mathbf{h}_{t}(\mathcal{X}_{t|t-1}^{i}) - \hat{\mathbf{z}}_{t|t-1}\right)^{T} \right) \mathbf{S}_{t}^{-1} . \tag{4.36}$$

The Kalman filter and its extensions are used in the following section to develop the projective Kalman filter.

4.4 Projective Kalman Filter

The projective Kalman filter is designed to cater for the non-linear nature of the homographic transformation. Indeed, a slight change in the observation is the result of a large change in the state for distant objects. The traditional approach for tackling this problem is to perform a homographic transformation followed by Kalman filtering. However, this technique fails to maintain an accurate estimate of the state vector because the error due to the physical trajectory and the error due to the projection on the plane are not differentiated. We propose to integrate the homographic transformation in the extended Kalman filter to compensate the distortion due to projection on the camera plane. The projective Kalman filter provides a better estimate because these two errors are modeled by two separate Gaussian processes, $\mathbf{v}_{t-1}$ and $\mathbf{n}_{t}$, respectively. The position and speed of the vehicle along the direction of the road are estimated. The projection severely distorts the observations and is highly non-linear. The projection on the normal direction, also

non-linear, does not require such a fine estimation because the distortion is not as drastic. A non-linear model in the normal direction would bring little improvement to the estimation but would increase the computation complexity of the algorithm. The state vector is defined as

$$\mathbf{x} = \begin{pmatrix} x \\ \dot{x} \\ s \end{pmatrix} , \tag{4.37}$$

where x and $\dot{x}$ are the position and speed of the vehicle following the tangential direction and s is the size of the vehicle. The observation vector $\mathbf{z}$ is composed of the apparent position, speed and size on the camera plane, *i.e.*,

$$\mathbf{z} = \begin{pmatrix} z \\ \dot{z} \\ b \end{pmatrix} . \tag{4.38}$$

The process equation, Eq. (4.24), models the physical process applying to the vehicle (Newton's laws), and the observation equation, Eq. (4.25), models the observed trajectories projected on the image plane via the projective transformation, hence the name projective Kalman filter. Therefore, assuming that the vehicle speed varies slowly, the system equation $\mathbf{f}$ is written as:

$$\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) = \begin{bmatrix} x_t \\ \dot{x}_t \\ s_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \dot{x}_{t-1}\Delta_t \\ \dot{x}_{t-1} \\ s_{t-1} \end{bmatrix} + \mathbf{v}_{t-1}. \tag{4.39}$$

The observation function $\mathbf{h}$ is the homographic transformation derived in Subsection 4.2.2 applied to the position, the speed and the size of the vehicle:

$$\mathbf{h}(\mathbf{x}_t, \mathbf{n}_t) = \begin{bmatrix} z_t \\ \dot{z}_t \\ b_t \end{bmatrix} = \begin{bmatrix} x_t Z_{vp}/(x_t + D) \\ \frac{D\dot{x}_t}{(x_t+D)(x_t-\dot{x}_t+D)} \\ \frac{s_t D}{(x_t+D)^2-(s_t/2)^2} \end{bmatrix} + \mathbf{n}_t \, . \tag{4.40}$$

Note that the vector-valued function $\mathbf{h}$ depends on H and θ implicitly through the vanishing point Z_{vp}.

4.4.1 State and Observation Updates

The projective Kalman filter must be able to tackle non-linear problems due to the nature of the observation function. The extended Kalman filter is preferred over the Unscented Kalman filter (see discussion in Subsection 4.4.3). Let $\hat{\mathbf{F}}$ and $\hat{\mathbf{H}}$ be the respective Jacobian matrices of the process and observation functions $\mathbf{f}$ and $\mathbf{h}$. The traditional EKF recursively estimates the state vector in two steps: prediction and update. However, because the apparent position is not directly accessible in the frame, the mean-shift procedure performs a search of the maximum likelihood of the apparent position between the prediction and update steps. The mean-shift procedure implemented here is described in Subsection 4.4.2. The projective Kalman filter is thus divided into three steps.

Prediction The state vector $\hat{\mathbf{x}}$ and its covariance matrix $\hat{\mathbf{P}}$ are estimated from the posterior density at time $t-1$:

$$\hat{\mathbf{x}} = \mathbf{f}(\mathbf{x}_{t-1}, 0)\,, \tag{4.41}$$

$$\hat{\mathbf{P}} = \mathbf{Q_{t-1}} + \hat{\mathbf{F}}_{t-1}\mathbf{P}_{t-1}\hat{\mathbf{F}}_{t-1}^{T}\,. \tag{4.42}$$

Apparent position estimation Mean-shift is applied to reach the mode of the apparent position z in the frame. The center $\hat{c}_x$ and the bandwidth b are initialized with the predicted observations $\hat{z}$ and $\hat{b}$ from $\hat{\mathbf{z}} = \mathbf{h}(\hat{\mathbf{x}}, 0)$. The observation at time t is then available as $\mathbf{z}_t = [c_x\ \dot{z}\ \hat{b}]^T$.

Update When a new observation $\mathbf{z}_t$ becomes available, *i.e.*, when the mean-shift tracker has converged to the center of the blob, the state vector is updated as follows:

$$\mathbf{x}_t = \hat{\mathbf{x}} + \mathbf{K}_t\big[\mathbf{z}_t - \hat{\mathbf{z}}\big]\,, \tag{4.43}$$

$$\mathbf{P}_t = \hat{\mathbf{P}} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{T}\,. \tag{4.44}$$

where

$$\mathbf{S}_t = \hat{\mathbf{H}}_t\hat{\mathbf{P}}\hat{\mathbf{H}}_t^{T} + \mathbf{R}_t \quad \text{and} \quad \mathbf{K}_t = \hat{\mathbf{P}}\hat{\mathbf{H}}_t^{T}\mathbf{S}_t^{-1}\,.$$

The Jacobians $\hat{\mathbf{F}}_{t-1}$ and $\hat{\mathbf{H}}_t$ are evaluated at $\mathbf{x}_{t-1}$ and $\hat{\mathbf{x}}$, respectively, with process and observation noise equal to 0.

(a) (b)

Figure 4.3 Background subtraction on a low definition image (128 × 160): (a) original image; (b) motion mask comprising moving objects.

4.4.2 The Mean-shift Procedure

Mean-shift is applied to the motion mask, representing the foreground detection, in order to determine the position of a blob center. The binary motion mask is obtained with the background subtraction techniques proposed in Section 3.3. Figure 4.3 displays an image and its corresponding motion mask extracted using background subtraction. Let us denote the approximate position of the blob center $\hat{\mathbf{c}} = [c_x \; c_y]^T$, the set of N motion pixel locations $\mathbf{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$ and K a Gaussian isotropic kernel with bandwidth b as defined in [48]. The new position of the blob center $\mathbf{c}$ is defined as

$$\mathbf{c} = \frac{\sum_{n=1}^{N} K\left(\|(\hat{\mathbf{c}} - \mathbf{m}_n)/b\|^2\right) \mathbf{m}_n}{\sum_{n=1}^{N} K\left(\|(\hat{\mathbf{c}} - \mathbf{m}_n)/b\|^2\right)} \,. \tag{4.45}$$

The mean-shift vector defining the shift in the center estimation is

$$\overrightarrow{p_b}(\hat{\mathbf{c}}) = \mathbf{c} - \hat{\mathbf{c}} \,. \tag{4.46}$$

The mean-shift vector $\overrightarrow{p_b}(\hat{\mathbf{c}})$ points toward the blob center. Equation (4.45) is iterated until $||\overrightarrow{p_b}(\hat{\mathbf{c}})|| < \gamma$ with $\hat{\mathbf{c}} \leftarrow \mathbf{c}$.

The convergence to the true blob center is ensured under two conditions:

1. *the estimated center* $\hat{\mathbf{c}}$ *is initialized in the basin of attraction of the blob.* The basin of attraction of a blob is defined as the set of locations for which the mean-shift converges to the blob center. In particular, the area delineated by the blob is included in the basin of attraction. Failing to initialize the mean-shift in the basin of attraction causes the divergence of the mean-shift tracker and the loss of the object track.

2. *the bandwidth b of the kernel matches the size of the blob.* The match between the bandwidth of the kernel and the size of the blob is also essential to ensure convergence. Indeed, a too large bandwidth would cause divergence in the presence of neighboring blobs; on the other hand, a too small bandwidth would lead to uncertainty in the blob location.

The estimated center and bandwidth are provided by the prediction step of the projective Kalman filter. After convergence, the estimated center is fed into the update step.

4.4.3 Extended versus Unscented Kalman Filter

Let us consider the system described by Eqs. (4.39) and (4.40). Because Eq. (4.39) is linear, the first order estimation of the process equation is exact and the use of the unscented transformation is unnecessary. On the other hand, higher order non-linearities are present in the observation equation. Let us consider the vector-valued function described by Eq. (4.40) which is continuously differentiable. The expansion of $\mathbf{h}(\mathbf{x}_t, 0)$ in Taylor series for the vector point $\mathbf{p}$ leads to the following approximations for the extended $\mathcal{T}_{EKF}$, and the Unscented, $\mathcal{T}_{UKF}$, Kalman filters, respectively:

$$\mathcal{T}_{EKF}(\mathbf{h}) = \mathbf{h}(\mathbf{p}) + \mathcal{J}_{\mathbf{h}}(\mathbf{p})(\mathbf{x}_t - \mathbf{p}) + o(\mathbf{x}_t - \mathbf{p})\,, \tag{4.47}$$

and

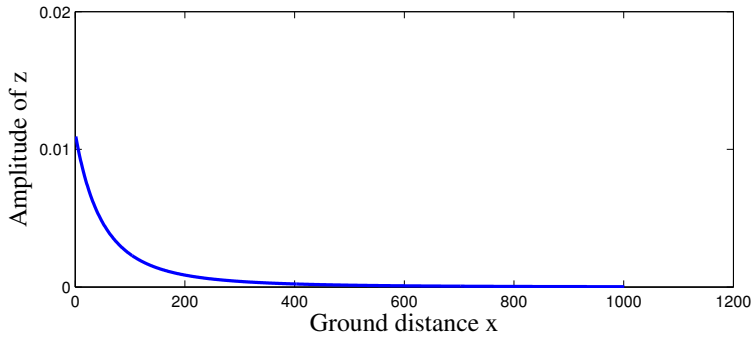
$$\begin{aligned} \mathcal{T}_{UKF}(\mathbf{h}) &= \mathbf{h}(\mathbf{p}) + \mathcal{J}_{\mathbf{h}}(\mathbf{p})(\mathbf{x}_t - \mathbf{p}) \\ &+ \frac{1}{2}(\mathbf{x}_t - \mathbf{p})^T \mathcal{H}_{\mathbf{h}}(\mathbf{p})(\mathbf{x}_t - \mathbf{p}) + o\left((\mathbf{x}_t - \mathbf{p})^2\right)\,, \end{aligned} \tag{4.48}$$

where $\mathcal{J}_{\mathbf{h}}$ and $\mathcal{H}_{\mathbf{h}}$ denote the Jacobian and the Hessian matrices of $\mathbf{h}$, respectively. The Unscented Kalman filter takes into account the second order nonlinearity; the difference in estimation between the EKF and the UKF lies in the Hessian matrices. Consequently, to evaluate the algorithms performance, we can evaluate the Hessian matrices in $\mathbf{p}$. The vector-valued function $\mathbf{h}$ is of size $[3 \times 1]$. Consequently, the Hessian is a tensor of order 3 and size $[3 \times 3 \times 3]$.

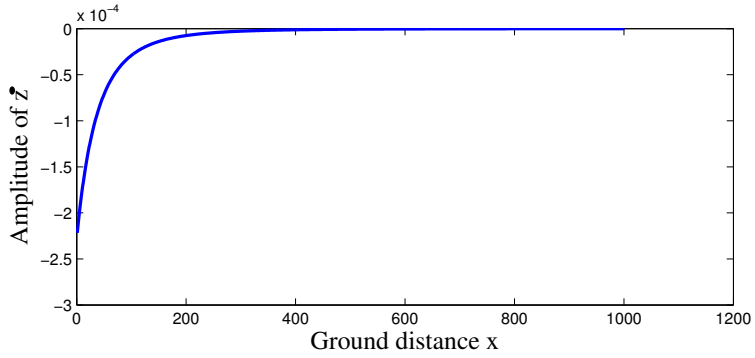
The Hessian tensor is derived with regards to the three real-world variables, namely x_t, $\dot{x}_t$ and s_t, and forms a tensor of 27 partial derivatives. Assuming the vanishing point Z_{vp} is fixed for a given video sequence, the Hessian tensor depends only on the parameter D. Figure 4.4 displays the theoretical improvement of the UKF over the EKF for the second derivatives with regards to the ground distance for a value of D equal to 43m (low value of the dataset, see Table 4.1). The 27 second partial derivatives representing the second order nonlinearity captured by the UKF account for subpixel accuracy ($< 10^{-2}$) of the position, speed and size of the object in the framework of our experiments. The second order nonlinearity becomes significant (> 0.5) for values of D below 7, which is not suitable for vehicle tracking. As a result, the UKF does not improve the quality of vehicle tracking compared to the EKF. Figure 4.5 presents the square error on the position estimation for both the EKF and the UKF for synthetic data generated with parameters derived from the video sequences Video_013 (see Table 4.1, Section 4.6). The mean square error is 0.3835 for the UKF and 0.3862 for the EKF. These results were confirmed on the vehicle tracking sequence: the UKF does not improve the performance of the tracking algorithm compared to the EKF which already achieves subpixel accuracy. To conclude, the EKF is preferred for tracking vehicles because it does not require the estimation of sigma points and the Jacobian can be pre-computed; the computation complexity of the EKF is lower than that of the UKF.

4.5 Vehicle Tracking System

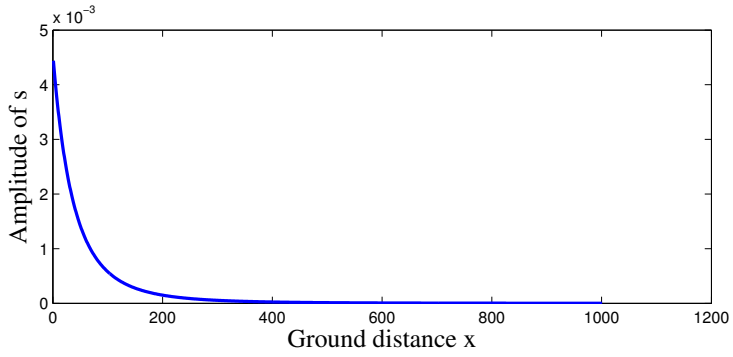
This section develops the vehicle tracking system based on the projective Kalman filter. The system is based on a sequential approach that processes incoming frames to update the trajectory of the vehicles in the scene. First, the foreground image is



(a) Second derivative *w.r.t.* z

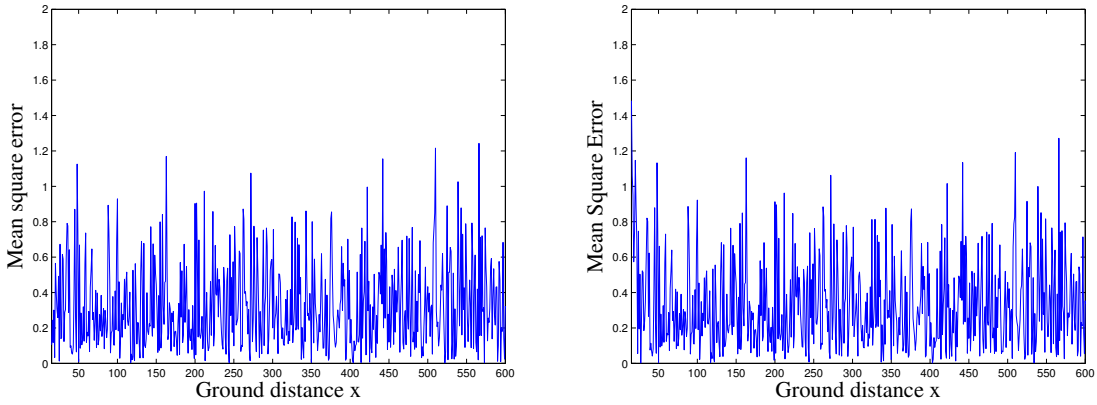


(b) Second derivative *w.r.t.* $\dot{z}$



(c) Second derivative *w.r.t.* s

Figure 4.4 Contribution of the Hessian matrix $\mathcal{H}_{\mathbf{h}}$ to the Unscented Kalman filter for value of x from 0 to 1000 meters for the parameters of video Video_011.



(a) Extended Kalman filter (b) Unscented Kalman filter

Figure 4.5 Mean square error of the pixel position of the object on the camera plane *w.r.t.* the ground distance x. Average computed on synthetic data over 500 trials.

extracted from the incoming video stream with a background subtraction algorithm using the Gaussian mixture model. The set of foreground pixel location $\mathbf{M}$ is fed into a blob labeling procedure to detect new vehicles in the frame in the detection zone. The features of all vehicles tracked (existing and new) are stored in a structure for further processing. The structure is called "objects" hereafter. The estimation of the vehicle state vector is performed conjointly by the projective Kalman filter and the mean-shift algorithm. Finally, a pruning step reduces the number of objects detected and merges adjacent blobs. An overview of the system is presented in Fig. 4.6 and the sequential pseudo-code is described in Algorithm 4.1.

Algorithm 4.1 Generic Projective Kalman Filter Algorithm

objects = $Initialization()$ (see Subsection 4.5.1)
while $incoming\_frame$ **do**
  $\mathbf{M} = background\_subtraction(incoming\_frame)$
  objects = $tracker\_initialisation(\mathbf{M}, detection\_zone)$
  for $i = 1$ to $number\_of\_objects$ **do**
    $(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = KF\_prediction(\text{objects}(i).\mathbf{x})$
    $\mathbf{z} = mean\_shift(\hat{\mathbf{z}})$
    objects(i).$\mathbf{x} = KF\_update(\mathbf{z}, \hat{\mathbf{z}})$
    $tracker\_pruning(\text{objects})$
    $display\_results$
  end for
end while

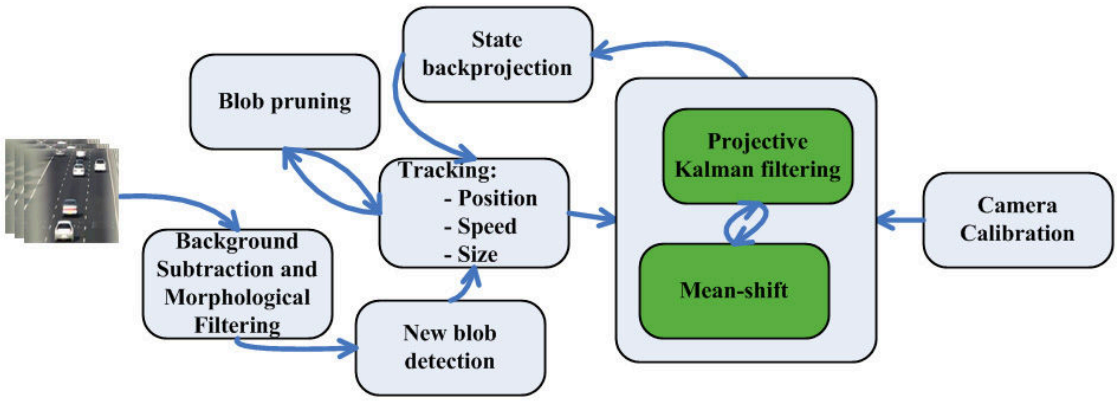


Figure 4.6 Overview of the vehicle tracking algorithm with projective Kalman filter.

4.5.1 Tracker Initialization and Pruning

A tracker is the feature vector of each object. By definition, the tracker includes at least the position of the object in the video frame. Tracker initialization and pruning are essential steps of the algorithm: the first one enables tracking of the objects and the second one removes redundant tracking of objects, which impairs the efficiency of the algorithm. A connected component procedure initializes the object position in the frame. Connected components usually finds labels in two passes. The first pass labels the components and the second one eliminates redundancy in labeling within connected components. The literature on the topic is abundant (see *e.g.* [11, 82, 98, 237]). For vehicle tracking on highways, the entrance zone of objects is known for a given sequence. The vehicle detection can thus be performed on a small area of the frame, reducing the computation load. The detection of new blobs is performed on each frame. However, this procedure does not ensure a unique tracker per object; a simple but efficient pruning operation is performed on the set of trackers. This procedure merges adjacent trackers, suppresses tracks of small sized objects and lost trackers, *i.e.* ,trackers that are not in the blob vicinity.

4.5.2 PKF Initialization and Vehicle Detection

The initialization of the variables is essential since the projective Kalman filter estimates the value of the state recursively. The vehicle is detected with the connected

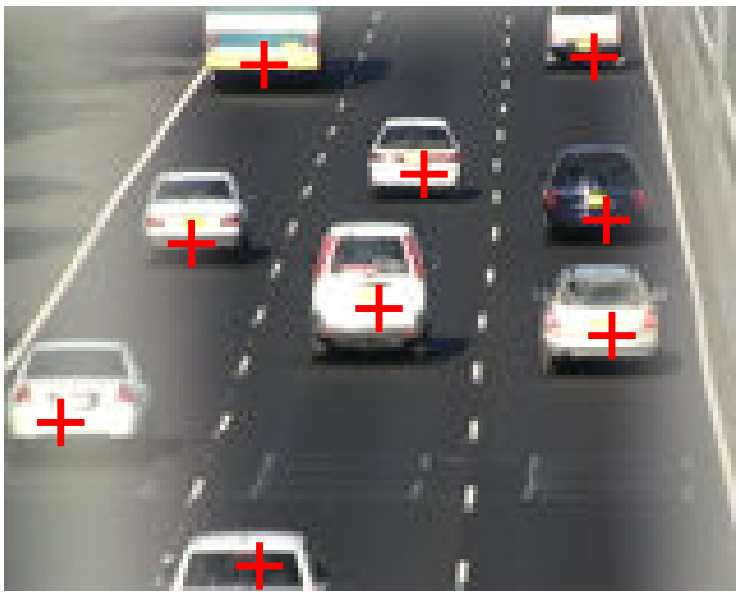


Figure 4.7 Example of tracking in dense vehicle flow. The incoming vehicles are well delineated due to their large size. Each vehicle is thus uniquely labeled by the tracking algorithm.

component procedure described in Subsection 4.5.1. We assume here that the vehicle blobs in the detection zone are well delineated. This condition is met in most practical cases since the gap between vehicles is large in the detection zone. In the experiments, the rare cases where two vehicles are merged in the same blob occur when the traffic is very dense and there is a continuous flow of vehicles. Most of the time, the dense flow of vehicles is correctly segmented.

Figure 4.7 shows a case of successful tracking of a dense flow of vehicles. The center $\mathbf{c}$ of each blob in the detection zone is computed as the mean location of the set of pixels with identical label. The initial state vector value $\mathbf{x}_0$ is set as

$$\mathbf{x}_0 = \begin{pmatrix} h_x^{-1}(c_x) \\ \dot{x}_0 \\ s_0 \end{pmatrix}, \tag{4.49}$$

where c_x is the position of the object on d_p-axis and h_x^{-1} is the inverse function of h_x (see Eq. (4.7)). The values $\dot{x}_0$ and s_0 are set to the speed and the size of vehicles, respectively. We found that $\dot{x}_0 = 25\text{m/s}$ and $s_0 = 5\text{m}$ provide good results for the tested sequences. The initial state covariance matrix $\mathbf{P}_0$ is set to 0 because the state

is assumed known with certainty at time $t = 0$. The process noise and measurement covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$ respectively, are initialized as follows:

$$\mathbf{Q} = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.50}$$

4.6 Performance Analysis on Vehicle Tracking

The performance of the proposed technique is tested on vehicle tracking on highways. The results aim to evaluate the projective Kalman filter in different scenarios on the dataset of traffic surveillance. First, we compare the performances of the projective Kalman filter with the extended Kalman filter for a frame rate of 30 fps (frames/s). Second, we compare the two Kalman filters for different frame rates, from 30 fps down to 3 fps. The second scenario provides an accurate evaluation of the algorithm performances for traffic monitoring where the frame rate is usually low. Finally, the number of mean-shift iterations necessary for both algorithms is discussed. It provides a qualitative measure of the Kalman filters estimation accuracy.

4.6.1 Experimental Setup and Data

The algorithm is tested on 15 traffic monitoring video sequences. The number of vehicles, the duration of the video sequences as well as the parameters of the homographic transformation are summarized in Table 4.1. Around 2600 vehicles are recorded on the set of video sequences. The videos range from clear weather to cloudy with weak illumination conditions. The camera was positioned above highways at a height of 5.5m to 8m. The video sequences are low-definition (128×160) to comply with the characteristics of traffic monitoring sequences. The threshold γ on the norm of the mean-shift vector is arbitrarily set to 0.2 to achieve subpixel accuracy. A lower value for γ does not improve the tracking accuracy in our experiments. The different parameters used for the experiments are summarized in Table 4.2.

The extended Kalman filter has been implemented in several traffic monitoring and

Table 4.1
Video sequences used for the evaluation of the algorithm performance along with the duration, the number of vehicles and the setting parameters, namely the height (H), the angle of view (θ) and the distance to field of view (D).

Video Sequence	Duration	No. of Vehicles	Camera Height (H)	Angle of view (θ)	Distance to FOV (D)
Video_001	199s	74	6m	8.5 ± 0.10 deg	48m
Video_002	360s	115	5.5m	15.7 ± 0.12 deg	75m
Video_003	480s	252	5.5m	15.7 ± 0.12 deg	75m
Video_004	367s	132	6m	19.2 ± 0.12 deg	29m
Video_005	140s	33	5.5m	12.5 ± 0.15 deg	80m
Video_006	312s	83	5.5m	19.2 ± 0.2 deg	57m
Video_007	302s	84	5.5m	19.2 ± 0.2 deg	57m
Video_008	310s	89	5.5m	19.2 ± 0.2 deg	57m
Video_009	80s	42	5.5m	19.2 ± 0.2 deg	57m
Video_010	495s	503	7.5m	6.9 ± 0.15 deg	135m
Video_011	297s	286	7.5m	6.9 ± 0.15 deg	80m
Video_012	358s	183	8m	21.3 ± 0.2 deg	43m
Video_013	377s	188	8m	21.3 ± 0.2 deg	43m
Video_014	278s	264	6m	18.5 ± 0.18 deg	64m
Video_015	269s	267	6m	18.5 ± 0.18 deg	64m

Table 4.2 Vehicle Tracking System and PKF Parameter Initializing Values

Parameters	γ	s_0	$\dot{x}_0$
Value	0.2	5m	25m/s

analysis systems, see *e.g.* [16] and [200]. The extended Kalman filter implements the same process function as Eq. (4.39). However, the observation function is modeled with the identity matrix whereas the proposed projective Kalman filter uses the observation function described in Eq. (4.40). The main problem encountered in vehicle tracking is the phenomenon of tracker drift. We propose here to estimate the robustness of the tracking by introducing a drift measure and to estimate the percentage of vehicles tracked without severe drift, *i.e.*, for which the track is not lost. Since the vehicles are converging to the vanishing point, the trajectory of the vehicle along the tangential axis is monotonically decreasing. As a consequence, we propose to measure the number of steps where the vehicle position decreases (p_d) and the number of steps where the vehicle position increases or is constant (p_i), which is characteristic of drift of a tracker. The rate of vehicles tracked without severe drift is then calculated as

$$\text{Correct Tracking Rate} = \frac{p_d}{p_d + p_i}\,. \tag{4.51}$$

4.6.2 Comparison of the PKF and the EKF

The average over the entire dataset shows a percentage of correct tracking of 84.5% for the extended Kalman filter and 98.3% for the projective Kalman filter. The proposed tracker shows more robust tracking, especially when vehicles are in the long distance. Visually, it translates as a migration of a tracker from one vehicle to another one in the neighborhood. Fig. 4.8 is an example of a tracker that drifts. With the EKF, the tracker on vehicle 1 slowly drifts away onto vehicle 2 because it is initialized on the edge of the two basins of attraction. After 25 frames, the tracker has changed basin of attraction and tracks vehicle 2. The drifting of the tracker is due to the failure of the extended Kalman filter to estimate accurately the distribution of the state vector in the particular non-linear environment. The homographic transformation integrated in the projective Kalman filter enables the proposed al-

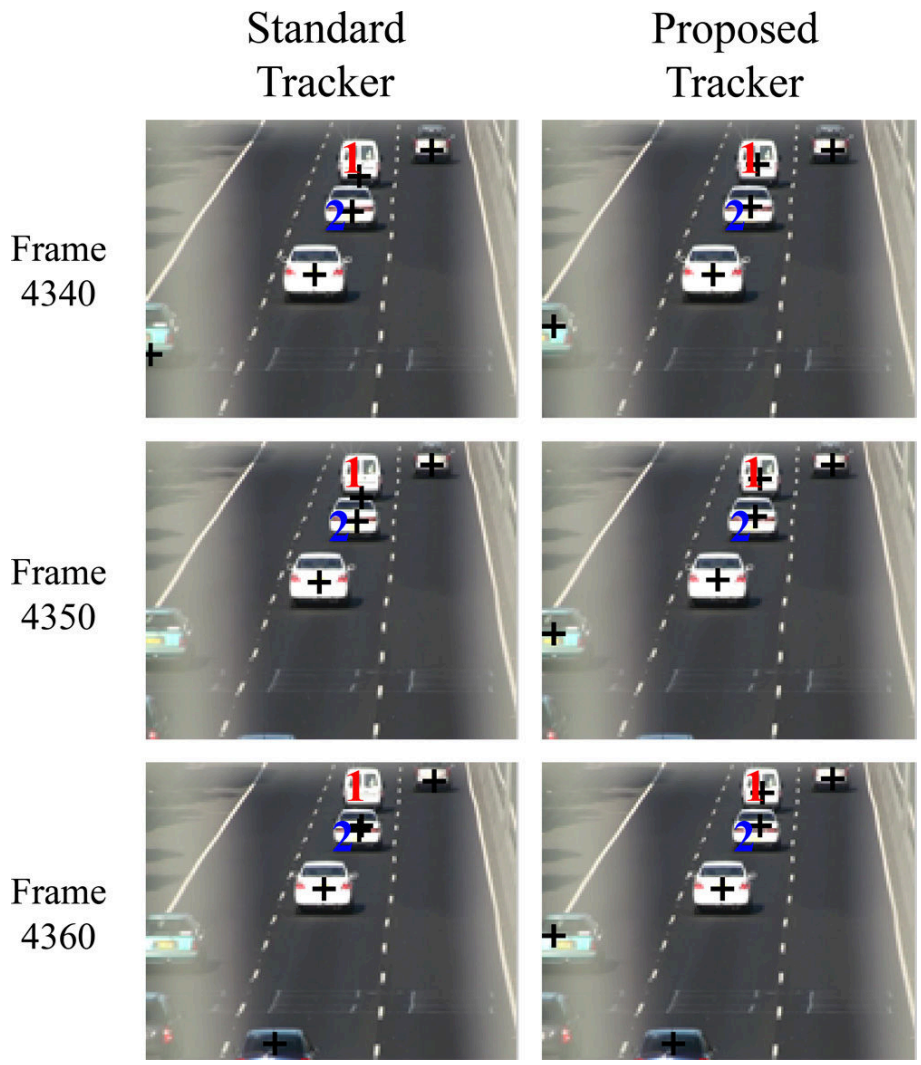


Figure 4.8 Sequence showing the drift of a tracker on vehicle 1: the positions of each tracked object is indicated by a dark cross. The tracker initialized on vehicle 1 drifts on vehicle 2 throughout the sequence.

gorithm to successfully track the vehicle throughout the sequence. Fig. 4.9 shows a successful tracking with the projective Kalman filter where the extended Kalman filter fails.

4.6.3 Effects of the Frame Rate on Tracking

In this subsection, the two algorithms are evaluated for different frame rates. Aside from their low-definition, traffic monitoring video sequences present a very low frame rate due to the difficulty of transmitting the video stream to the traffic agency. We

Figure 4.9 Comparison of the standard tracking algorithm (*left*) and the proposed algorithm (*right*). The proposed algorithm presents better ability to track long distance vehicles.

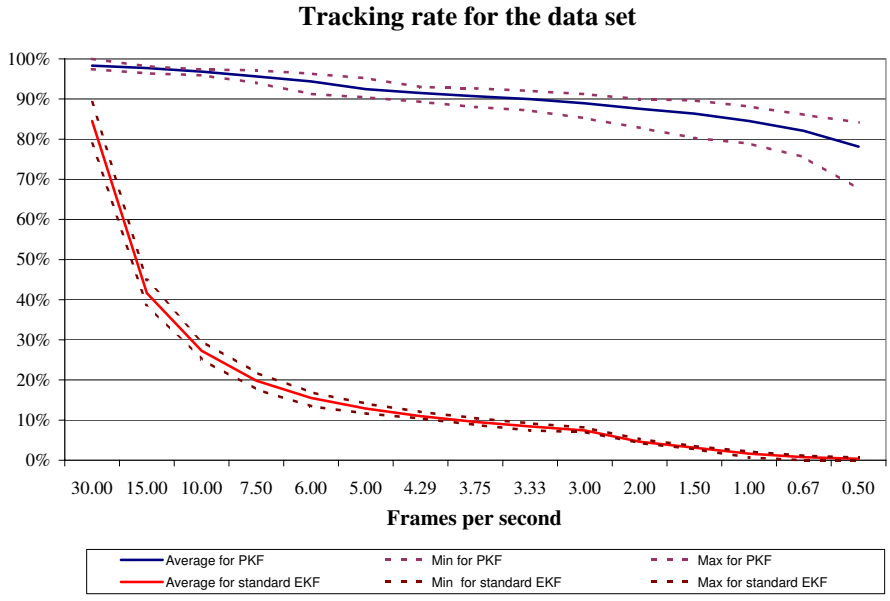


Figure 4.10 Effects of the frame rate on the tracking performances. Average tracking rate over the dataset for projective and extended Kalman filters are displayed with the minimum and maximum rate for the dataset.

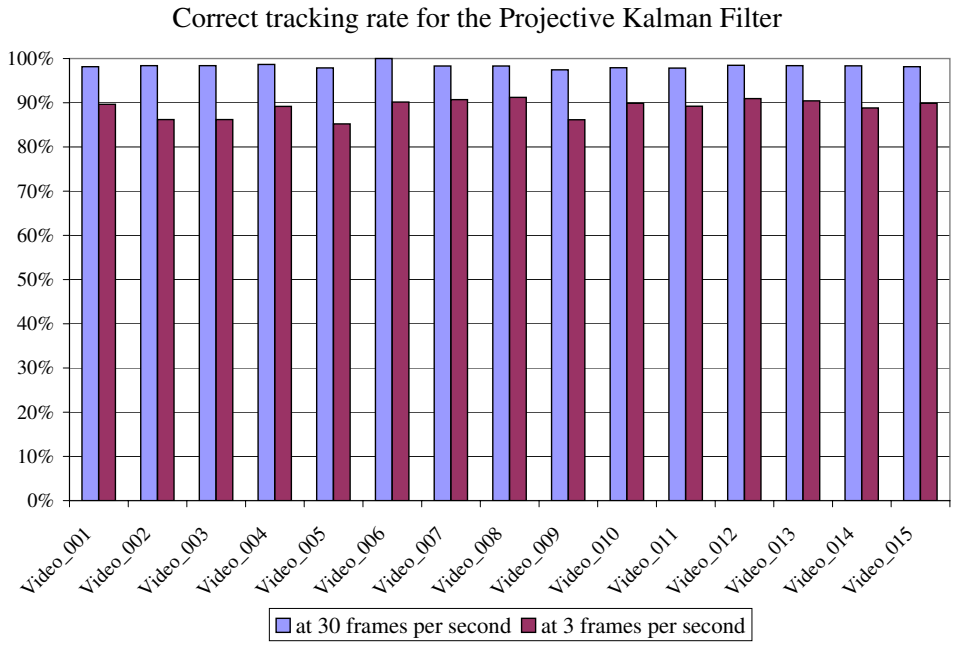
propose here to evaluate the performances of the extended Kalman filter and the projective Kalman filter on video sequences with decreasing frame rates, from 30fps to 0.5fps. Note that even though rates below 3fps are unusual, they are presented here for the sake of completeness. The tracking robustness is evaluated in terms of tracking rate as defined in Eq. (4.51). Figure 4.10 displays the average rate of tracking over the entire dataset with the maximum and the minimum tracking

rate for the extended and the projective Kalman filters. The extended Kalman filter shows a quick rate of decay with the frame rate. The tracking rate for the projective Kalman filter is also decreasing with the frame rate; however, the tracking rate of the projective Kalman filter is less sensitive to the frame rate compared to that of the extended Kalman filter. For example, at 3 frames per second, the PKF presents a tracking rate of 89% whilst the EKF tracking rate is 7.4%. Indeed, when the number of frames per second decreases, the displacement of the vehicle and, more importantly, the uncertainty of the vehicle location in the frame increases. As a consequence, the standard method is unable to track the vehicles because the algorithm fails to initialize the mean-shift in the basin of attraction.

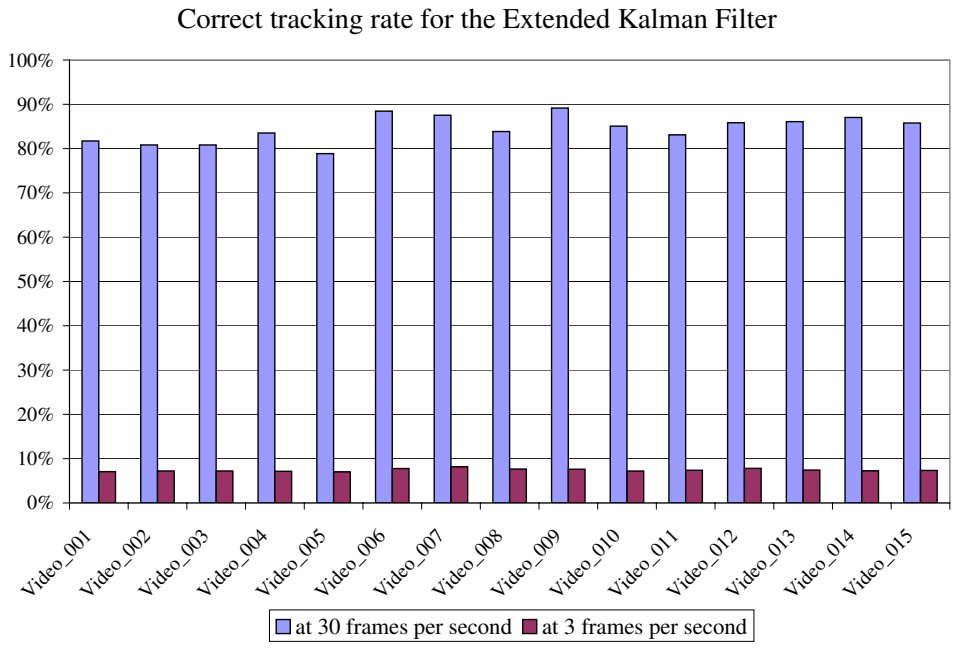
Figure 4.11 displays the tracking rate over the set of video sequences for a frame rate of 30 and 3fps. Some examples of vehicle tracking are presented in Fig. 4.12. Tracking with the extended Kalman filter fails for distant objects because the basin of attraction is small and the extended Kalman filter does not provide a fine estimation of the position for the initialization of the tracker. The projective Kalman filter, on the other hand, provides an accurate estimation of the vehicle position in the image via a fine adjustment of the vehicle speed when the frame rate decreases and the information becomes sparse. Therefore, the proposed approach is less sensitive to frame rate.

4.6.4 Mean-shift Convergence Speed at Low Frame Rates

The speed of convergence of the mean-shift is a key factor in the proposed algorithm. As the mean-shift is a gradient ascent procedure, the speed of convergence represents the proximity of the feature vector to the mode of the distribution in the feature space. The distance is the error between the kernel density estimation of the mode by mean-shift and the prediction of the state value in Eq. (4.41) by the projective Kalman filter. Figure 4.13 displays the number of iterations of the mean-shift for the first 3000 runs of the procedure. A run represents the convergence of one vehicle in one frame. The results displayed are also smoothed with a sliding window of size 100 for clarity of presentation. The average rate calculated over 33,000 runs for the video sequence Video_012 is 4.19 for the projective Kalman filter and 7.00 for the extended Kalman filter, which represents a gain of 67%.



(a)



(b)

Figure 4.11 Comparison of the tracking rate for the projective Kalman filter and the extended Kalman filter at 30 and 3 frames per second. The results are displayed for the 15 videos of the dataset. (a) Tracking rate for the projective Kalman filter at 30 and 3 frames per second; (b) Tracking rate for the extended Kalman filter at 30 and 3 frames per second.

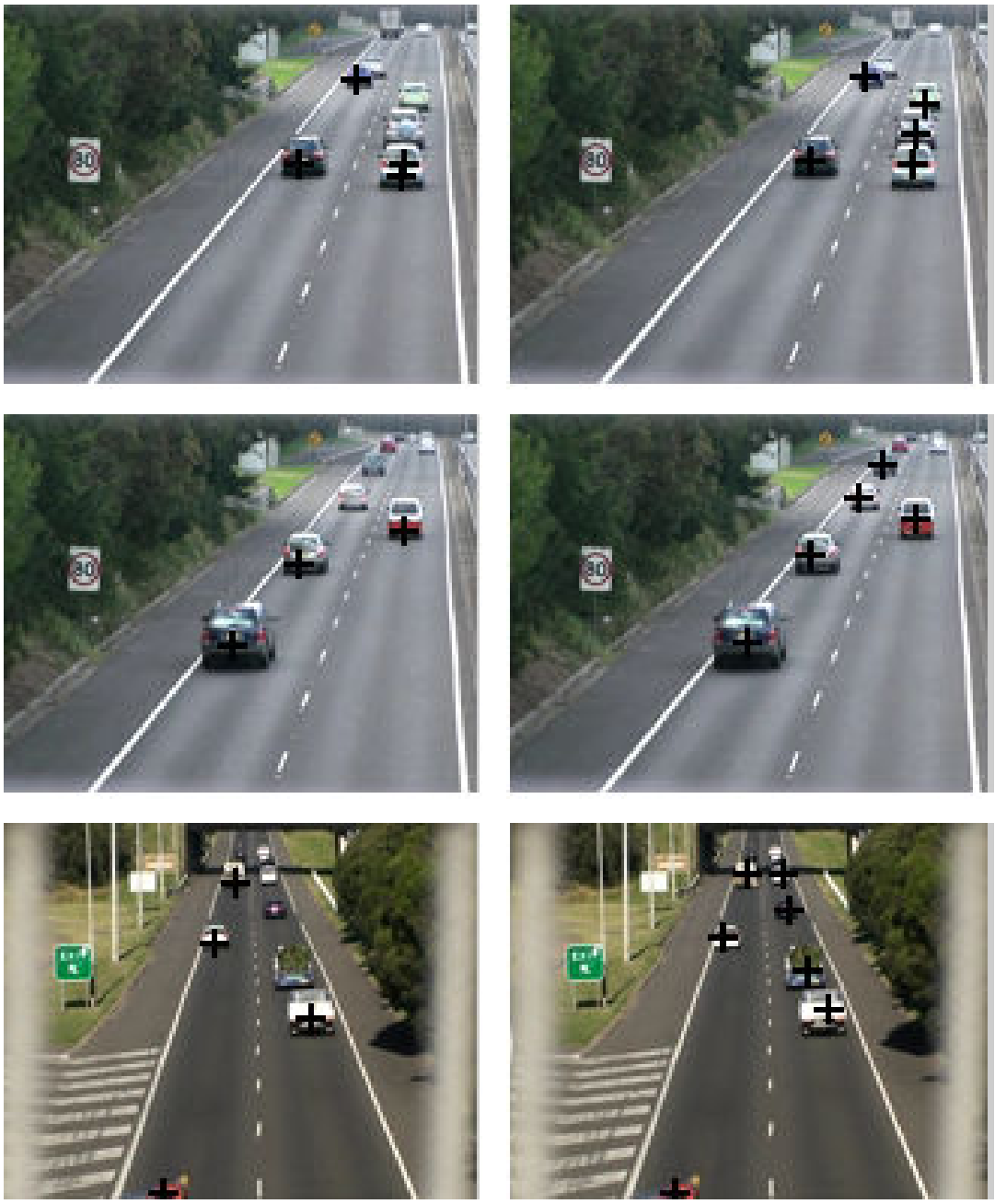


Figure 4.12 Tracking robustness in low frame rate (3fps) for the standard (*left*) and the proposed method (*right*). With the standard method, the tracker drifts quickly and is unable to track the vehicle.

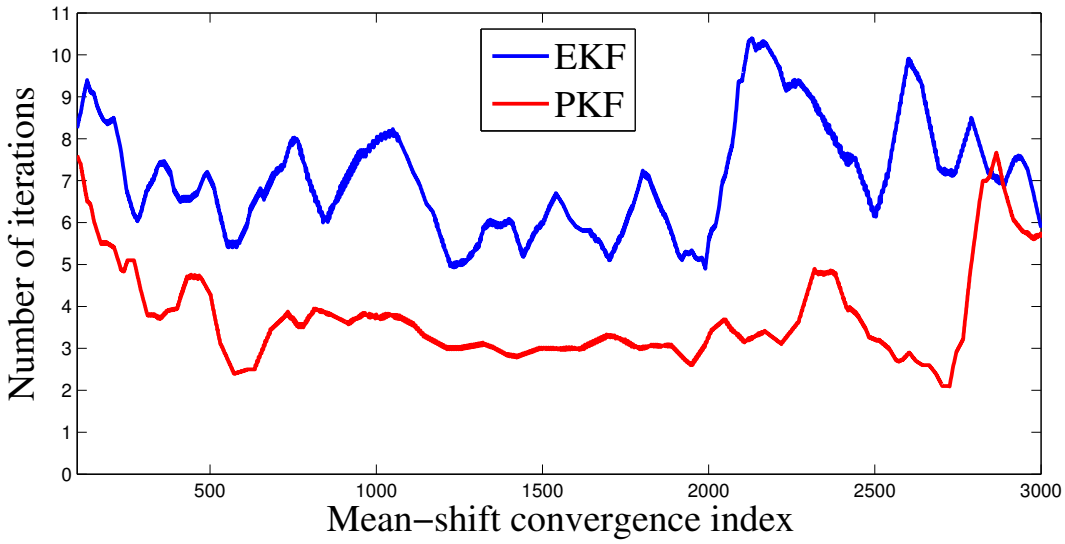


Figure 4.13 Average number of mean-shift iterations for the Projective Kalman filter and the extended Kalman filter. The data is smoothed with a sliding window of size 100 for clarity.

4.7 Summary of the Projective Kalman Filter

This chapter proposed a tracking algorithm based on a tracker/estimator pair. The mean-shift coupled with the projective Kalman filter achieves robust tracking due to the integration of the homographic projection of the real-world vehicle trajectories on the camera plane. In particular, the observation function of the projective Kalman filter models the trajectory of vehicles with respect to their ground distance to the camera. It results in a fine estimation of the vehicle position both in the real-world and on the camera plane, providing a tracking with reduced drift. The combination of the mean-shift and the PKF also leads to more accurate observations, which reduces the error in the distribution of the state estimate. The results showed that both the extended and the projective Kalman filter algorithms achieve robust tracking at a rate of 30fps even though the projective Kalman filter performs better. At very low frame rates (*e.g.*, 3 fps), the extended Kalman filter provides very poor results whereas the proposed algorithm still tracks vehicles with 89% accuracy. The robustness of the extended Kalman filter drops quickly with the frame rate compared to the projective Kalman filter. Finally, we showed that the

number of iterations required for the convergence of mean-shift is lower with the proposed method, thereby reducing the computation load.

Chapter 5

Projective Particle Filter for Vehicle Tracking

5.1 Introduction

This chapter investigates the integration of the projective transformation, developed in Chapter 4, with particle filters. The particle filter is a suboptimal solution, based on Monte Carlo simulations, to the Bayesian problem since it approximates the density of interest instead of providing an exact representation. However, it relaxes the Gaussian and linearity constraints, and therefore copes with a wider range of pdfs for tracking. The particle filter has the property of achieving an accuracy in the state estimate proportional to the square root of the number of particles. The number of particles and their distribution in the feature space become critical elements in the development of tracking algorithms with particle filtering.

The vehicle tracking algorithm has already been introduced in Chapter 4. This chapter therefore focuses on the particle filter algorithm and the integration of the homographic transformation. We propose to refine the importance density, from which samples are drawn, with the projective transformation to increase tracking accuracy for a given number of samples. More specifically, the projective particle filer aims to reduce the size of the particle set for a given mean square error, or, maintain the latter while reducing the former. Section 5.2 develops the sequential

Monte Carlo framework and particle filtering approach. It also introduces to the notion of sample degeneracy and resampling. Section 5.3 describes the projective particle filter (PPF) implementation for vehicle tracking. Section 5.4 presents the performance analysis of the projective particle filter.

5.2 Sequential Monte Carlo and Particle Filtering

Monte Carlo methods encompass a range of techniques based on stochastic simulations. They estimate complex and often analytically intractable problems. Consequently, the study of Monte carlo methods has developed with the increase of computer power. Monte carlo simulations are based on numerical approximation of a system of interest by sampling. The process is sequential and is composed of three steps: particle generation, particle diffusion and statistical interpretation.

Particle generation. Monte Carlo simulations rely on the sampling of probability density. The initial density of the *samples* is designed either arbitrarily or based on prior inference. The particles carry the statistics of the density via sampling as the unscented transform did for the UKF in the Gaussian framework (see Section 4.3.3). However, a larger number of particles is required since the density is unknown.

Particle diffusion through the system. The particles are fed into the system and the output is a set of samples individually transformed by the studied process. Although an analytical solution of the system output for the initial density is not available, the set of samples represents the transformation of the input density by the system. For instance, output particles will agglomerate around the modes of the output density.

Statistics generation and interpretation. The set of output samples provides information on the output density although the latter one is not readily available. Different statistics can be drawn to characterize the system such as the expected value or other higher order moments. If the output density is to be reconstructed, traditional techniques for kernel density estimation can be

employed. For instance, the regularized particle filter uses kernel estimation to reconstruct the output density [180, 42].

Monte Carlo methods rely on the theory of large numbers for statistical estimation of systems or functions. The main result of interest in this section is called the Monte Carlo integration, which is described hereafter. We follow to some extent the derivation proposed by Ristic *et al.* in [212] for the derivation of the theory. Let us consider a function $\mathbf{g}$, integrable on its domain $\mathcal{D}$:

$$I = \int_{\mathcal{D}} \mathbf{g}(\mathbf{x}) d\mathbf{x} \,. \tag{5.1}$$

Assume now that the function $\mathbf{g}(\mathbf{x})$ can be factorized as

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})\pi(\mathbf{x}) \,. \tag{5.2}$$

where $\pi(\mathbf{x})$ is a probability density function. Given N_S samples $\mathbf{x}^i$ drawn from the density $\pi(\mathbf{x})$, with N_S large, the integral in Eq. (5.1) can be approximated with a sum I_{N_S}

$$I_{N_S} = \frac{1}{N_S} \sum_{i=1}^{N_S} \mathbf{f}(\mathbf{x}^i) \,. \tag{5.3}$$

Monte Carlo integration states that there is asymptotic statistical convergence or, in other words almost sure convergence, between the integral I and the sum I_{N_S}, *i.e.*,

$$\lim_{N_S \to \infty} \frac{1}{N_S} \sum_{i=1}^{N_S} \mathbf{f}(\mathbf{x}^i) = \int_{\mathcal{D}} \mathbf{g}(\mathbf{x}) d\mathbf{x} \,. \tag{5.4}$$

This result holds if the variance of $\mathbf{f}(\mathbf{x})$, $\sigma^2_{\mathbf{f}(\mathbf{x})} = \int_{\mathcal{D}} (\mathbf{f}(\mathbf{x}) - I)^2 \pi(\mathbf{x}) d\mathbf{x}$, is finite. In this case, the central limit theorem also ensures that the estimation error from the Monte Carlo simulation converges with a speed $\mathcal{O}(\sqrt{N_S})$ to the normal density

$$\lim_{N_S \to \infty} \sqrt{N_S}(I_{N_S} - I) \sim \mathcal{N}(0, \sigma^2) \,. \tag{5.5}$$

Monte Carlo simulations are an elegant solution to circumvent the direct and sometimes impractical calculation of an integral in a system by drawing a number of samples that carry the statistics of the density underlying the process. Markov chains associated with the Bayesian problem described in Subsection 2.4.2 take advantage of the asymptotic property. An excellent introduction to the use of Markov chains with Monte Carlo simulations can be found in [84].

5.2.1 A Sub-optimal Bayesian Solution: The Particle Filter

The particle filter (PF) is a technique for approximating the recursive Bayesian solution while relaxing the Gaussian and linear constraints on the system. Recalling the fundament of the recursive Bayesian solution lies in the pair of prediction and update equations (2.20) and (2.21), the particle filter aims to estimate the posterior density $p(\mathbf{x}_t|\mathbf{Z}_t)$ through Monte Carlo simulations. The main hindrance to relaxing the Gaussian constraint in the Bayesian solution is the Chapman-Kolmogorov integral in the prediction step, Eq. (2.22). The comparison with Eq. (5.2) leads to $\mathbf{f}(\mathbf{x}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $\pi(\mathbf{x}) = p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})$. For the moment, we keep the notations $\mathbf{f}(\mathbf{x})$ and $\pi(\mathbf{x})$ for the sake of clarity.

Importance Sampling

Bearing in mind that $\pi(\mathbf{x})$ is unknown since it is the posterior density (to be estimated), a proposal density $q(\mathbf{x})$, referred to as the *importance* density in the rest of the thesis, is used to draw the set of samples. The importance density shall be as close as possible to the posterior and in particular have the same support, *i.e.*, $\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0, \forall \mathbf{x} \in \mathcal{D}$. This leads to the reformulation of the Monte Carlo integration as

$$I = \int_{\mathcal{D}} \mathbf{f}(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \; = \int_{\mathcal{D}} \mathbf{f}(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}\,, \tag{5.6}$$

yielding

$$I_{N_S} = \frac{1}{N_S}\sum_{i=1}^{N_S} \mathbf{f}(\mathbf{x}^i)\frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)} = \frac{1}{N_S}\sum_{i=1}^{N_S} \mathbf{f}(\mathbf{x}^i)\tilde{w}(\mathbf{x}^i)\,. \tag{5.7}$$

where $\tilde{w}(\mathbf{x}^i)$ are weights given by

$$\tilde{w}(\mathbf{x}^i) = \pi(\mathbf{x}^i)/q(\mathbf{x}^i)\,. \tag{5.8}$$

The weights readjust the error introduced by the sampling from the importance density. Also, the weights need to be normalized and the Monte Carlo estimate becomes

$$I_{N_S} = \sum_{i=1}^{N_S} \mathbf{f}(\mathbf{x}^i)w(\mathbf{x}^j)\,, \tag{5.9}$$

where

$$w(\mathbf{x}^i) = \frac{\tilde{w}(\mathbf{x}^i)}{\sum_{j=1}^{N_S} \tilde{w}(\mathbf{x}^i)} \,. \tag{5.10}$$

Sequential Importance Sampling

Sequential importance sampling (SIS) maintains the pdf of interest in the form of a set of samples and associated weights, together called *particles*, to recursively approximate the posterior density of a state. This provides a solution to the Bayesian problem that asymptotically converges to the optimal estimator. In practice, the solution is sub-optimal since it is impossible to have an infinite set of samples. To derive the solution, let us first consider that a set of particles, composed of a set of samples $\{\mathbf{x}^i_{t-1}, i = 1..N_S\}$ and a set of associated weights $\{w^i_{t-1}, i = 1..N_S\}$, is available and approximates the posterior density at time $t-1$ such that

$$p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) \approx \sum_{i=1}^{N_S} w^i_{t-1} \delta(\mathbf{x}_{t-1} - \mathbf{x}^i_{t-1}) \,, \tag{5.11}$$

where $\delta(.)$ is the Dirac delta function. The notation in Eq. (5.11) is a discretization of the posterior pdf at time $t-1$. Here, we differentiate our reasoning from Ristic *et al.* [212] to show the recursive update of the set of particles. Considering the update step in the Bayesian problem, and integrating the predicted and likelihood densities with the Monte Carlo estimation, we have from Eq. (2.23)

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Z}_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_t)} \,, \tag{5.12}$$

$$= \frac{p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})d\mathbf{x}_{t-1}}{p(\mathbf{z}_t|\mathbf{Z}_t)} \,, \tag{5.13}$$

$$\approx \frac{p(\mathbf{z}_t|\mathbf{x}_t) \sum_{i=1}^{N_S} w^i_{t-1} p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1}) \delta(\mathbf{x}_{t-1} - \mathbf{x}^i_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_t)} \,, \tag{5.14}$$

$$\approx \frac{\sum_{i=1}^{N_S} w^i_{t-1} p(\mathbf{z}^i_t|\mathbf{x}^i_t) p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1}) \delta(\mathbf{x}_{t-1} - \mathbf{x}^i_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_t)} \,. \tag{5.15}$$

If the importance density is chosen as

$$q(\mathbf{x}_t|\mathbf{Z}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{Z}_t) q(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) \,, \tag{5.16}$$

the update of the samples from time $t-1$ to t is given by the importance transition density $q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{Z}_t)$. Updating the set of samples in Eq. (5.15) yields

$$p(\mathbf{x}_t|\mathbf{Z}_t) \approx \frac{\sum_{i=1}^{N_S} w_{t-1}^i p(\mathbf{z}_t^i|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i) \delta(\mathbf{x}_t - \mathbf{x}_t^i)}{p(\mathbf{z}_t|\mathbf{Z}_t) q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{Z}_t)} \,. \tag{5.17}$$

$$\approx \sum_{i=1}^{N_S} w_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i) \,. \tag{5.18}$$

Equation (5.18) is identical to (5.11) for time t. The new set of weights that appears in Eq. (5.18) is as follow

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{z}_t^i|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{p(\mathbf{z}_t|\mathbf{Z}_t) q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{Z}_t)} \,, \tag{5.19}$$

$$\propto w_{t-1}^i \frac{p(\mathbf{z}_t^i|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{Z}_t)} \,. \tag{5.20}$$

Another derivation of the Bayesian problem with Monte Carlo simulations can be found in Chapter 3 of [212] and in [9], where the authors work with the joint posterior pdf $p(\mathbf{X}_t|\mathbf{Z}_t)$ instead of the posterior pdf $p(\mathbf{x}_t|\mathbf{Z}_t)$.

We showed in this subsection that the Bayesian problem can be recursively approximated with a sub-optimal solution that converges to the optimal solution when $N_S \to \infty$. The recursive update lies in Eqs. (5.16) and (5.20).

5.2.2 Samples Degeneracy and Resampling

The choice of the importance density is crucial to obtaining a good estimate of the posterior pdf $p(\mathbf{x}_t|\mathbf{Z}_t)$. The optimal choice for the importance density is the posterior itself. However, because this density is not available, it has been shown that the set of particles and associated weights $\{\mathbf{x}_k^i, w_k^i\}$ will eventually degenerate, *i.e.*, most of the weights will be carried by a small number of samples and a large number of samples will have negligible weight [137]. This phenomenon is also known as *sampling impoverishment*. Resampling is necessary to circumvent the degeneracy problem. Intuitively, the set of particles does not represent the density under estimation when the distribution of the weights is not homogeneous anymore, *i.e.*, when the variance becomes large. An evaluation of the effective sample size has

been reported in [9] as

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_S} (w_t^i)^2} . \tag{5.21}$$

The effective sample size is a value between 1 and N_S that represents the degree of suitability of the particle set. Traditionally, a threshold N_{th} is applied on $\hat{N}_{eff}$ to make the resampling decision. Resampling can be performed with the cumulative function of the weights and a random variable uniformly distributed in the interval $\mathcal{U}[0, N_S^{-1}]$. Algorithm 5.1 shows a sequential implementation of the technique. This technique is called systematic resampling. It is generally adopted because it is the fastest algorithm to resample the set of particles. Other methods have been implemented and are reviewed in [247].

5.2.3 Particle Filter Summary

The particle filter relies on a set of particles $[\{\mathbf{x}^i, w^i\}_{i=1}^{N_S}]$ to estimate the posterior density at time t from the posterior density at time $t-1$ through an unknown system based on the hidden Markov model. The particle filter, based on Monte Carlo simulations, relaxes the assumptions made on the nature of the noise or the system equations for the Kalman filter. However, the particle filter is a sub-optimal solution because the number of particles is finite. There are numerous variations of the particle filter in the literature and the reader is referred to Subsection 2.4.2 for examples. Most of these techniques aim to refine the importance density to obtain a better approximation of the underlying density. For instance, extended

Algorithm 5.1 Resampling Algorithm

```
l = 0
ε = U[0, N_S^{-1}]
for i = 1 to N_S do
  σ_i = cumsum(w_k^i)
  while ε + l/N_S < σ_i do
    x_k^l = x_k^i
    w_k^l = 1/N_S
    l = l + 1
  end while
end for
```

and unscented particle filters use the EKF and UKF to estimate the state of the samples. The regularized particle filter smoothes the posterior density at time $t-1$ from which the samples are drawn.

5.3 Projective Particle Filter

The projective particle filter is based on the linear fractional transformation introduced in Subsection 4.2.2. The difference between projective Kalman and particle filters resides in the tracking algorithm and the state vector. The first one is based on vehicle blobs obtained with background subtraction and the second one on kernel-based color tracking. Therefore, only the tracking implementation of the system is described here. The proposed particle filter is named projective particle filter because the vehicle position is projected on the camera plane. The projection is used as an inference to diffuse the particles in the feature space. One of the particularities of the PPF is to differentiate between the importance density and the prior pdf whilst the sampling importance resampling (SIR) filter, also called standard particle filter, does not. Therefore, we need to define the importance density $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$ from the fractional transformation as well as the prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ in order to update the weights in Eq. (5.20).

5.3.1 Importance Density and Prior

The projective particle filter integrates the linear fractional transformation into the importance density $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$. The state vector is modeled with the position, the speed and the size of the vehicle in the image:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ b \end{bmatrix}, \tag{5.22}$$

where x and y are the Cartesian coordinates of the vehicle, $\dot{x}$ and $\dot{y}$ are the respective velocity components along the x- and y-axes, respectively, and b is the size of the

vehicle in the image. From Subsection 4.2.2, the apparent speed $\dot{x}$ and size of the vehicle b can be derived in terms of apparent position of the vehicle for the projective particle filter. Considering the real speed is constant, Eqs. (4.8) and (4.9) yield

$$\dot{x} = f_{\dot{x}}(z) = \frac{(H-z)^2 v}{H(D-v)+zv}\,, \tag{5.23}$$

and

$$b = f_b(z) = \frac{sD}{(\frac{HD}{H-z})^2 - (\frac{s}{2})^2}\,. \tag{5.24}$$

It is worthwhile noting that for the projective particle filter the state is the apparent trajectory of the vehicle, while for the projective Kalman filter the state is the real trajectory. Object tracking is traditionally performed using a standard kinematic model (derived from Newton's Laws of motion), taking into account the position, the speed and the size of the object[1]. For the projective particle filter, the kinematic model is refined with the estimation of the speed and the object size via linear fractional transformation. Let us define the vector-valued process function $\mathbf{f}$ as

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}) = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \\ b_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + f_{\dot{x}}(x_{t-1}) \\ y_{t-1} + \dot{y}_{t-1} \\ f_{\dot{x}}(x_{t-1}) \\ \dot{y}_{t-1} \\ f_b(x_{t-1}) \end{bmatrix}. \tag{5.25}$$

It is important to note that, as for the projective Kalman filter, the distortion is severe along the x-axis and the function $f_{\dot{x}}$ provides a better estimate than a simple kinematic model taking into account the speed of the vehicle. On the other hand, the distortion along the y-axis is much weaker and the compensation is not necessary. The novelty of the PPF resides in the estimation of the vehicle position along the x-axis and its size through $f_{\dot{x}}$ and $f_b(x)$, respectively. It is worthwhile noting that the standard kinematic model of the vehicle is recovered when $f_{\dot{x}}(x_{t-1}) = \dot{x}_{t-1}$ and $f_b(x) = b_{t-1}$. Let $\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1})$ denote the standard kinematic model assuming zero acceleration. The vector-valued function $\mathbf{g}(\mathbf{x}_{t-1}) = \{\mathbf{f}(\mathbf{x}_{t-1}) | f_{\dot{x}}(x_{t-1}) = \dot{x}_{t-1}, f_b(x) = b_{t-1}\}$ denotes the standard kinematic

[1]The size of the object is maintained for the purpose of likelihood estimation.

model

$$\mathbf{g}(\mathbf{x}_{t-1}) = \begin{bmatrix} x_{t-1} + \dot{x}_{t-1}\,\Delta_t \\ y_{t-1} + \dot{y}_{t-1}\,\Delta_t \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \\ b_{t-1} \end{bmatrix} . \tag{5.26}$$

Consequently, the samples are drawn from the importance density $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = \mathcal{N}(\mathbf{x}_t, \mathbf{f}(\mathbf{x}_{t-1}), \Sigma_q)$, and the standard kinematic model is used in the prior distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t, \mathbf{g}(\mathbf{x}_{t-1}), \Sigma_p)$, where $\mathcal{N}(., \boldsymbol{\mu}, \Sigma)$ denotes the normal distribution of covariance matrix Σ centered on $\boldsymbol{\mu}$. The distributions are considered Gaussian and isotropic to evenly spread the samples around the estimated state vector at time t.

5.3.2 Likelihood Estimation

The estimation of the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ is based on the distance between color histograms as in Comaniciu *et al.* [56]. Let us define an m_n-bin histogram $H = \{H[u]\}_{u=1..m_n}$, representing the distribution of J color pixel values $\mathbf{c}$, as follows:

$$H[u] = \frac{1}{J}\sum_{i=1}^{J} \delta[\kappa(\mathbf{c}^i) - u]\,, \tag{5.27}$$

where u is the set of bins, regularly spaced on the interval $[1, m_n]$, κ is a linear binning function providing the bin index of pixel value $\mathbf{c}^i$, and $\delta(.)$ is the Kronecker delta function. The pixels $\mathbf{c}^i$ are selected from a circle of radius b centered on (x, y), coordinates of the center of vehicle in the frame. Indeed, after projection on the camera plane, the circle is the standard shape that delineates the vehicle best. Let us denote the target and the candidate histograms by H_t and $H_{\mathbf{x}}$, respectively. The Bhattacharyya distance between two histograms is defined as

$$\Delta(\mathbf{x}) = \left(1 - \sum_{u=1}^{m_n} \sqrt{H_t[u]H_{\mathbf{x}}[u]}\right) . \tag{5.28}$$

Finally, the likelihood $p(\mathbf{z}_t|\mathbf{x}_t^i)$ is calculated as $p(\mathbf{z}_t|\mathbf{x}_t^i) \propto \exp\left(-\Delta(\mathbf{x}_t^i)\right)$.

5.3.3 System Implementation

The implementation of the projective particle filter algorithm is summarized in Algorithm 6.1. Because most approaches to tracking take the prior distribution as importance density, the samples $\mathbf{x}_t^i$ are directly drawn from the standard kinematic model. In this subsection we differentiate between the prior and the importance density to obtain a better distribution of the samples. The initial state $\mathbf{x}_0$ is chosen as $\mathbf{x}_0 = [x_0, y_0, 10, 0, 20]^T$ where x_0 and y_0 are the initial coordinates of the object. The value $\mathbf{x}_0$ is thus used to draw the set of samples $\mathbf{x}_0^i \sim q(\mathbf{x}_0|\mathbf{z}_0) = \mathcal{N}(\mathbf{x}_0^i, \mathbf{f}(\mathbf{x}_0), \Sigma_q)$. The prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the importance density $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$ are both modeled with Gaussians. The covariance matrices Σ_p and Σ_q are initialized as follows:

$$\Sigma_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix} \quad \Sigma_q = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix} \tag{5.29}$$

Algorithm 5.2 Projective Particle Filter Algorithm

Require: $\mathbf{x}_0^i \sim q(\mathbf{x}_0|\mathbf{z}_0)$ and $w_0^i = 1/N_S$
  for $i = 1$ to N_S **do**
    Compute $\mathbf{f}(\mathbf{x}_{t-1}^i)$ from Eq. (5.25)
    Draw $\mathbf{x}_t^i \sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t) = \mathcal{N}(\mathbf{x}_t^i, \mathbf{f}(\mathbf{x}_{t-1}^i), \Sigma_q)$
    Compute ratio $\gamma_t = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\mu}_\gamma, \Sigma_\gamma)$
    Update weights $w_t^i = w_{t-1}^i \times \gamma_t p(\mathbf{z}_t|\mathbf{x}_t)$
  end for
  Normalize w_t^i
  if $N_{eff} < N$ **then**
    $l = 0$
    for $i = 1$ to N_S **do**
      $\sigma_i = \text{cumsum}(w_t^i)$
      while $\frac{l}{N_S} < \sigma_i$ **do**
        $x_t^l = x_t^i$
        $w_t^l = 1/N_S$
        $l = l + 1$
      end while
    end for
  end if

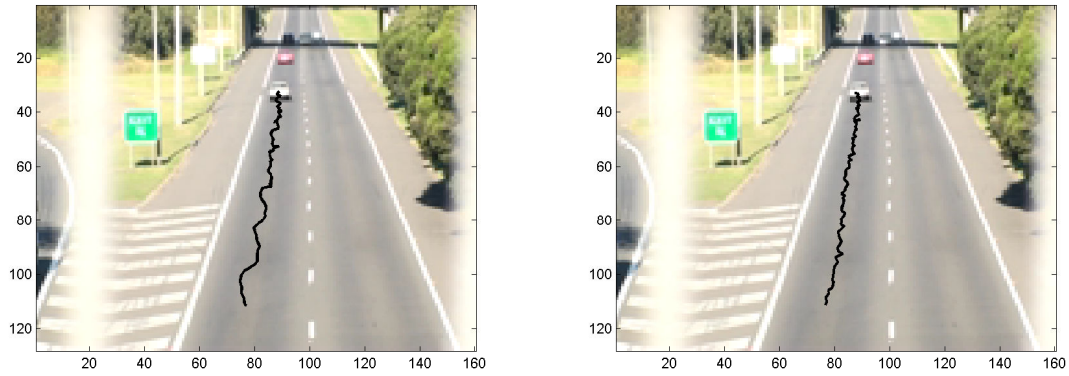
and the mean vectors are initialized as follows: $\boldsymbol{\mu}_p = \mathbf{g}(\mathbf{x}_0)$ and $\boldsymbol{\mu}_q = \mathbf{f}(\mathbf{x}_0)$. As a result, the variable γ_t is itself drawn from a Gaussian process $\mathcal{N}(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\mu}_\gamma, \Sigma_\gamma)$ with covariance matrix $\Sigma_\gamma = (\Sigma_p^{-1} - \Sigma_q^{-1})^{-1}$ and $\boldsymbol{\mu}_\gamma = \Sigma\,(\Sigma_p^{-1}\boldsymbol{\mu}_p - \Sigma_q^{-1}\boldsymbol{\mu}_q)$ and $\Sigma_p \neq \Sigma_q$.

A resampling scheme is necessary to avoid the degeneracy of the particle set. Systematic resampling, as introduced in Subsection 5.2.2, is performed when the variance of the weight set is too large, *i.e.* when the number of the effective sample size $\hat{N}_{eff}$ falls below a given threshold N_{th}, arbitrarily set to $0.6N_S$ in the implementation.

5.4 Experiments and Results

In addition to the drift measure introduced in Eq. (4.51), an important measure in vehicle tracking with particle filters is the variance of the trajectory since it directly depends on the particle set size. Indeed, high-level tasks, such as abnormal behavior detection or driving under the influence of alcohol (DUI) detection, require an accurate tracking of the vehicle and, in particular, a low mean square error for the position. The standard and the projective particle filters are evaluated in this section on the traffic surveillance data introduced in Subsection 4.6.1. The video sequences are footage of vehicles traveling on a highway. Although the roads are straight in the dataset, the algorithm can be applied to curved roads with approximation of the parameters on short distances because the projection tends to linearize the curves on the camera plane. The parameters θ, H and D defining the linear fractional transformation are recalled in Table 5.1. The reduced dataset used here is composed of 205 vehicles assumed to have a constant speed of $v = 25\text{m.s}^{-1}$. Note that the constraint on the speed can be relaxed as long as the variations are slow.

Figure 5.1 displays a track estimated with the projective (Fig. 5.1(b)) and the standard particle filter (Fig. 5.1(a)). Qualitatively, it is clear that the projective particle filter shows a smaller variability in the track estimation. We run two experiments to evaluate the variance for the standard and the projective particle filters: one with automatic variance estimation and the other one with ground truth labeling. A third experiment is conducted to evaluate the suitability of the importance pdf . Finally,



(a) Standard (b) Projective

Figure 5.1 Vehicle track for (a) the standard and (b) the projective particle filter. The projective particle filter exhibits a lower variance in the position estimation.

Table 5.1 Linear Fractional Transformation Parameters

Video Sequence	H	θ	D
Video_5	5.5 m	12.5 ± 0.15 deg	80 m
Video_6	5.5 m	19.2 ± 0.2 deg	57 m
Video_8	5.5 m	19.2 ± 0.2 deg	57 m

a summary of the drift rate characterizing the suitability of the different algorithms to accurately track vehicles is presented along with a discussion on projective and extended Kalman filters, and projective and standard particle filters.

5.4.1 Mean Square Error Performance

In the first experiment, the performance of each tracker is evaluated in terms of MSE using the reduced dataset. In order to avoid the tedious task of manually extracting the ground-truth of every track, a synthetic track is generated automatically based on the parameters of the real world projection of the vehicle trajectory on the camera plane. Figure 5.2 shows that the calculated and the manually extracted tracks match very well. The initialization is performed as for the projective Kalman filter (Section 4.5). However, because the initial position of the vehicle when the tracking starts may differ from one track to another, it is necessary to align the calculated and the manually extracted tracks in order to cancel the bias in the

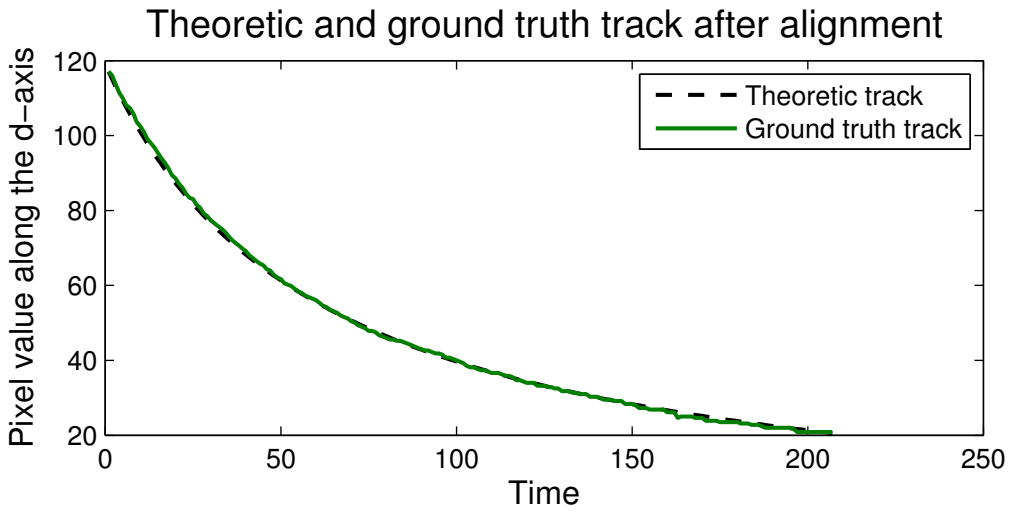


Figure 5.2 Alignment of calculated and extracted trajectories along the d-axis. The difference between the two tracks represents error in the estimation of the trajectory.

Table 5.2 MSE for the Standard and the Projective Particle Filters

Video Sequence	Video_5	Video_6	Video_8
Avg. MSE Std PF	2.26	0.99	1.07
Avg. MSE Proj. PF	1.89	0.83	1.02

estimation of the MSE. The average MSE for each video sequence is summarized in Table 5.2. It can be inferred from the table that the projective particle filter performs better on the entire dataset than the standard particle filter.

In the second experiment, we evaluate the performance of the two tracking algorithms *w.r.t.* the number of particles. Here, the ground truth is manually labeled in the video sequence. We arbitrarily decide to ground truth the first 5 trajectories of the first video to ensure the impartiality of the evaluation. Figure 5.3 displays the average MSE over 10 runs for the first trajectory and for different values of N_S. In theory, the MSE decreases at a rate $O(\sqrt{N_S})$ which is not the case here. Figure 5.3 shows a decrease in the MSE towards a constant rate. This is imputable to the change in the car model (color) through time that imposes a lower bound on the MSE. Figure 5.4 presents the average MSE for 10 runs on the 5 manually extracted tracks for $N_S = 20$ and $N_S = 100$. It is clear that the projective particle filter outperforms the standard particle filter in terms of MSE. The superior accuracy

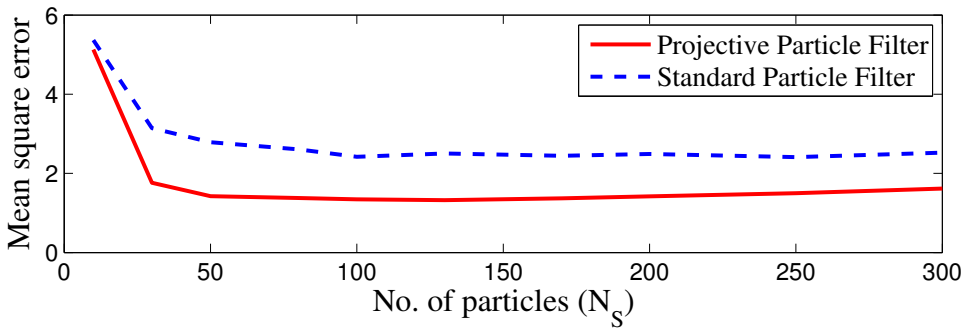


Figure 5.3 Position mean square error versus number of particles for the standard and the projective particle filter.

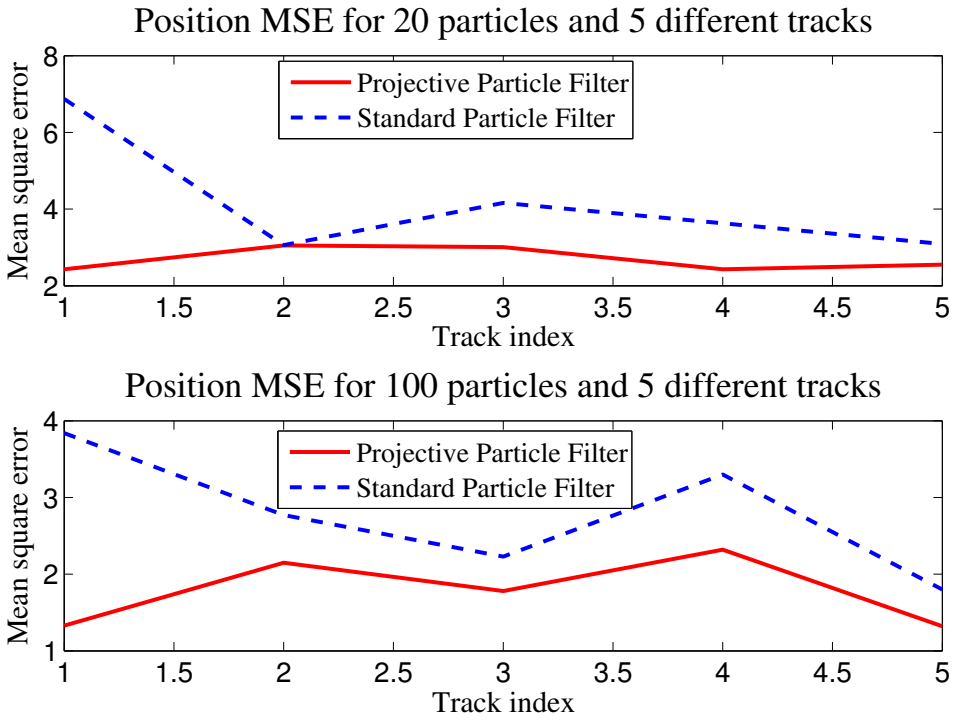


Figure 5.4 Position mean square error for 5 ground truth labeled vehicles using the standard and the projective particle filter. *Top*: with 20 particles; *bottom*: with 100 particles.

of the PPF is due to the finer estimation of the sample distribution by the importance density and the consequent adjustment of the weights since all parameters are identical in the comparison.

5.4.2 Importance Sampling Evaluation

We propose to compare the standard and the projective particle filters without the resampling step. This evaluation determines the suitability of the importance density to the problem. Indeed, the closer the importance density is to the posterior density, the less resampling is needed. However, because the importance and the posterior density are different, a larger number of particles is required for the experiment to avoid losing track after a few iterations. We choose $N_S = 300$ for the evaluation. Figure 5.5 shows the position MSE for the standard and the projective particle filters for the 80 successfully tracked trajectories in Video_8; the average MSEs are 1.10 and 0.58, respectively. For the problem of vehicle tracking, the importance density q used in the projective particle filter is therefore more suitable to draw samples from compared to the prior density used in the standard particle filter. Less resampling is required as a consequence of the adequate choice of importance density. It is also worth noting that the lower MSE in this experiment compared to the one exhibited in Table 5.2 for Video_8 is due to the higher number of particles.

5.4.3 Tracking Performance and Discussion

The drift tracking rate, defined in Eq. (4.51), is evaluated for the projective and standard particle filters developed in this chapter. Figure 5.6 displays the results for the entire (15 videos) traffic surveillance dataset. It shows that the projective particle filter yields better tracking rate than the standard particle filter across the entire dataset. Therefore, it can be inferred that the integration of the homographic transformation improves the tracking rate. Furthermore, it can be observed that the PPF does not perform as well as the PKF in tracking vehicles (see Fig. 4.11 for comparison). Two hypotheses are brought forward to explain this result:

- the vehicle tracking environment is, in reality, Gaussian or quasi-Gaussian. In this case, the Kalman filter provides the optimal solution while the particle

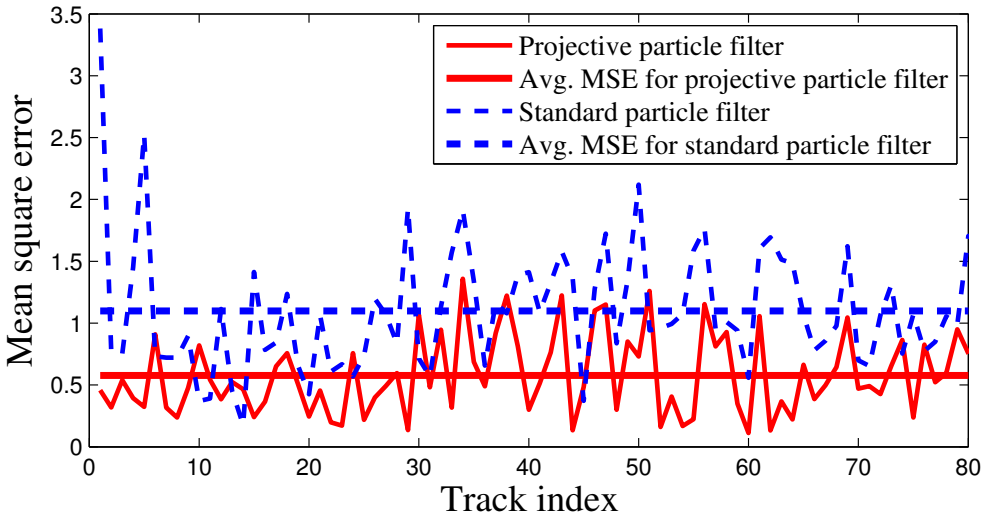


Figure 5.5 Position mean square error for the standard and the projective particle filter without resampling step.

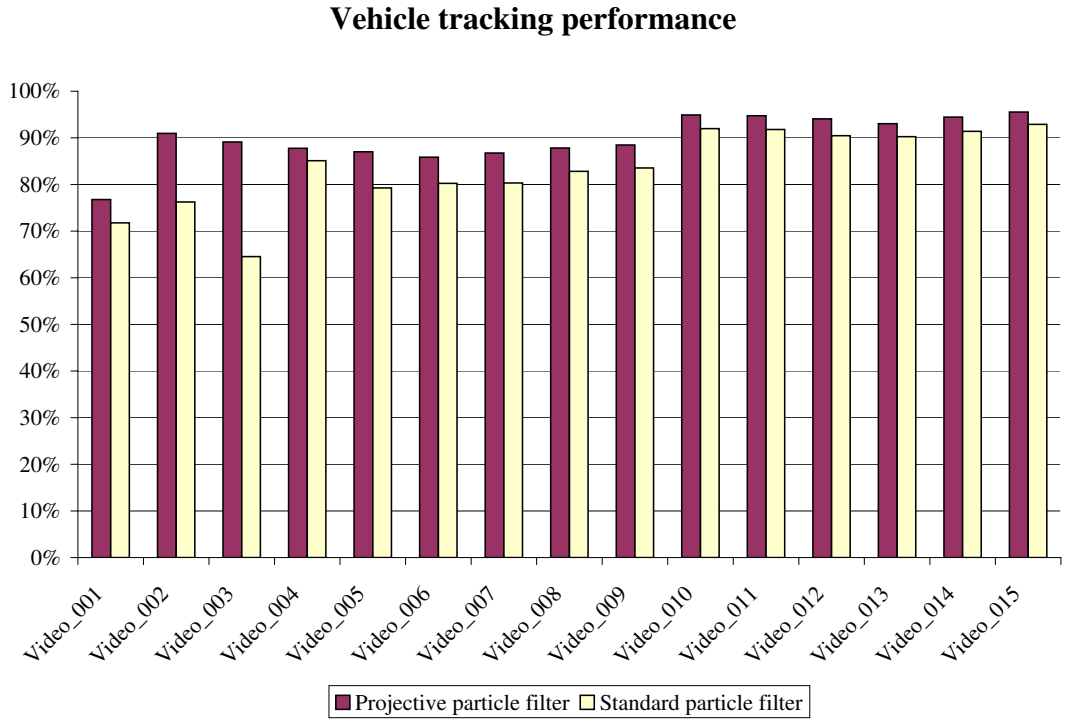


Figure 5.6 Drift tracking rate for the projective and standard particle filters on the traffic surveillance dataset.

filter is only suboptimal. It is then expected that the projective Kalman filter yields better results than the projective particle filter.

- the two tracking algorithms are based on different techniques. The PKF relies on background subtraction providing reliable segmentation of the object. The PPF relies on a histogram-based color tracker without target update. When the color distribution changes, the tracking algorithms can be distracted by the background. An algorithm to update the target appearance, although computationally expensive, can be implemented to address this issue (see *e.g.* [95]).

The comparison between the projective Kalman and particle filters is therefore difficult and somewhat unfair because the two algorithms are based on different techniques, each with their own advantages and disadvantages. However, the PKF is selected for the trajectory extraction that will be used in Chapter7 because it provides a better output tracking rate.

5.5 Summary of the Projective Particle Filter

A new particle filter integrating the linear fractional transformation in the importance density is proposed. This projection maps the real world position of a vehicle onto the camera plane providing a better distribution of the samples in the feature space. However, because the prior is not used to sample, the weights of the designed Projective Particle Filter have to be readjusted. The standard and the projective particle filters have been evaluated on traffic surveillance videos. It has been shown that the MSE on the trajectory of the vehicles is reduced with the projective particle filter. Furthermore, the proposed technique outperforms the standard particle filter in terms of MSE regardless of the number of particles. It has also been shown that the degeneracy of the sample set is reduced when the importance density is based on the linear fractional transformation. However, the projective particle filter is outperformed by the projective Kalman filter in terms of drift tracking rate. The projective Kalman filter is therefore selected for the extraction of trajectories in Chapter 7.

Chapter 6

Tracking Through Occlusion with Markov Random Fields

6.1 Introduction

This chapter is dedicated to the study of Markov random fields (MRFs) to produce Bayesian inference for object tracking. Chapters 4 and 5 developed the enhancement of tracking accuracy and robustness in the framework of traffic surveillance, *i.e.* , in a constrained environment. However, in the general case, the environment constraints are not readily available and hence must be learnt. Therefore, we propose to widen the framework of tracking to any model presenting explicit or implicit patterns in trajectories through Markov random fields. The sequel focuses on the design of Markov random fields for improving the robustness of tracking by optimizing the distribution of samples for particle filtering. A local importance density is therefore learnt in order to generate inference for the particle filter. This algorithm is used for general purpose tracking. The implementation of the particle filter is therefore necessary to convey the multi-modality diffusion of the posterior density—the Kalman filter is not adequate in this framework.

Markov random fields have been used for their ability to model the probability distribution of a random variable at a location given its neighborhood distribution. Therefore, the learning is dependent on adjacent locations. Applied to tracking,

Markov random fields provide a convenient framework for modeling smooth patterns such as trajectory paths. In this chapter, we show that the trajectory of the feature vector in the feature space can be learnt from the local pattern of previous objects and be used as inference for the particle filter through the importance density. The main contribution is the design of a local model that can be used to increase the robustness of tracking in case of occlusion. The work presented in the sequel also sets the framework for abnormal behavior detection that will be investigated in Chapter 7. Section 6.2 introduces the notion of context integration in visual object tracking and the suitability of Markov random field in this task. Section 6.3 develops the proposed mixture of Markov random fields and its update with sparse realizations and simulated annealing. Section 6.4 presents the performance of the tracking system based on Markov random fields and in particular, the performance in terms of mean square error and tracking through occlusion before concluding in Section 6.5.

6.2 Integration of Contextual Information

Contextual information is introduced in tracking to improve the accuracy of the particle filter. In particular, it provides robust recovery of tracking after occlusion. The context provides Bayesian inference through Markov random fields. This section introduces the handling of occlusion, shows the importance of contextual information and presents the Markov random fields.

6.2.1 Occlusion Handling

Traditional Bayesian filtering does not provide a framework to occlusion handling, which is of particular importance to ensure the robustness of object tracking. Occlusion is defined as the total or partial lack of visual clues over an arbitrary period of time on the object of interest. Because object tracking techniques rely upon visual information in computer vision, an occlusion leads to uncertainty, and, in the worst case, to the track loss. To handle occlusion, prior information is to be integrated in the tracking system. Currently, four main approaches to occlusion handling have been proposed. First, prior knowledge on the shape of the object has been used to

achieve successful tracking through occlusion [276, 57, 15, 87]. For instance, physical constraints on the shape improve the fitting of the contour to the object of interest. General Hough transform has also been employed to model the shape for template matching [187]. Second, occlusion reasoning is applied to the object of interest. A set of independent features are employed to solve occlusion with information fusion [281]. Also, trees and semantics are used to describe the nature of the occlusion with high-level track descriptors (*e.g.* split, merge, disappear) [108]. Third, multi-cameras techniques handle occlusion by merging information from sources with different angles of view of the scene [124, 254]. Fourth, dynamic linear and nonlinear models have been used to estimate the state vector of the object during occlusion [111]. Kinematic models are integrated in the tracker to estimate the position of the object. The main advantage of this technique is that the intrinsic dynamic of the shape is irrelevant to the recovery of the track; it can handle better the occlusion of objects with large variation in shape dynamics. In all aforementioned techniques, visual inference is required to ensure the recovery of the track is not due to chance; total occlusion is therefore precluded. The technique developed in this chapter belongs to the latter category in that non-linear dynamic models are catered for. However, our approach is based on local object behavior rather than general kinematic models. We propose to use inference from a Markov random field to estimate the dynamics of the object of interest under total occlusion. The object is tracked in the traditional framework of particle filtering but total occlusion is handled due to adequate modeling of the importance density. The contribution of this chapter is the development of a parametric importance density model relying on contextual information from previous behaviors through a Markov random field.

6.2.2 Importance of Contextual Information

Here, we present a simple scenario to illustrate the importance of contextual information. Let us consider a person A living in her/his house and a person B who has never been to the house before. Person A wants to switch off the light in the living room. With visual clues and knowledge of the house, Person A would go directly and switch off the light. Person B, with only visual clues, would also find the switch. Now, let us consider the same scenario but A and B have to switch

the light on in a dark room. Even without visual clues, A would still be able to reach the switch thanks to the knowledge of the house acquired over time. Person B, without visual clues and contextual knowledge, will rely on chance in the search for the switch. However, if the same experiment is run in person B's house, person B would certainly find the switch and person A would likely fail. Two key aspects to tracking can be inferred from this scenario:

1. Knowledge of the past is essential when visual clues are lacking (*e.g.*, in the presence of occlusion);
2. Knowledge acquired through time provides local inference only.

These observations are corroborated by the results presented in [25] showing that a local mixture model can better characterize the behavior of objects than its global counterpart and that the integration of the neighborhood accelerates the learning of the behavior. Figure 6.1 presents the various distributions of vehicle displacements in a scene, depending on the context (*e.g.* straight line, T-junction, crossroads). For each site of interest, the local pdf of the displacement is modeled by a mixture of Gaussians. The feature vector is composed of the horizontal (dx) and the vertical (dy) displacements. It is clear that the probability density functions vary largely from one site to another. An accurate global representation of vehicle displacement is unrealistic since the size of the feature vector is augmented with the position, leading to a complex and cumbersome model.

6.2.3 Markov Random Fields

Markov random fields are an extension of Markov chains. The Markov chain is a sequence of one-dimensional dependencies of random processes inheriting of the Markov property. Markov random fields are of higher order and, therefore are a mesh of dependencies instead of a chain. For this reason, the causality of the Markov property is not transferable to MRFs and the dependency of a given random variable must be redefined as a noncausal property. Let us first introduce the framework of Markov random fields and, in particular, the notions of undirected graph, neighborhood and clique.

Figure 6.1 Representation of vehicle motion on a road network by local mixture of Gaussians. The diversity of distributions prohibits the use of a global estimate. Instead, local contextual information can be used to describe vehicle behavior.

Undirected Graph

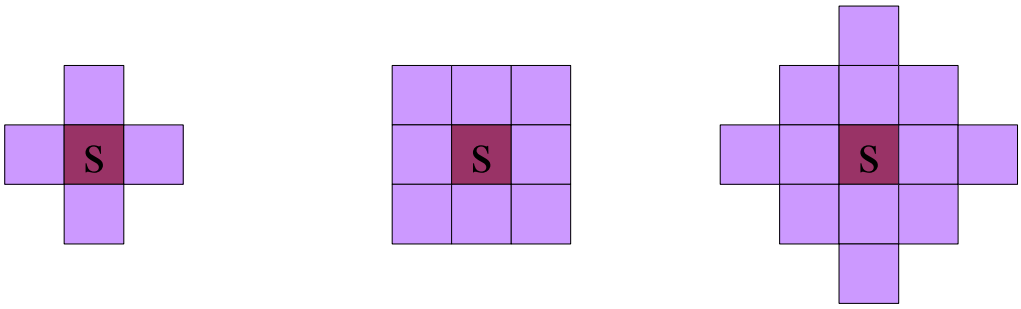
An undirected graph Ω is a collection of N vertices, also called sites, $S = \{s_1, .., s_N\}$ and edges $E = (s_i, s_j)_{\{i,j\} \in [1..N] \times [1..N]}$. The graph is said *undirected* if and only if $(s_j, s_i) = (s_i, s_j)$. In visual object tracking, a graph can describe the inter-dependencies among pixels in an image. For example, pixels of the same object can present color similarities represented as dependencies. The undirected graphs of interest in this thesis are arranged in a 2-D lattice forming a mesh over the image.

Neighborhood

The neighborhood η_{s_i} of a vertex s_i is defined as the set of vertices s_j for which there exists an edge (s_i, s_j) from site s_i to site s_j. It is a subset of Ω describing the spatial contiguity of site s_i. A site cannot be a neighbor of itself, that is, there cannot be an edge (s_i, s_i). Furthermore, the neighborhood of s_i must satisfy: $s_i \in \eta_{s_j} \Leftrightarrow s_j \in \eta_{s_i}$. Consequently, the neighborhood of a site s_i in an undirected graph is symmetric around the site since $(s_i, s_j) = (s_j, s_i)$. Figure 6.2 displays examples of neighborhoods.

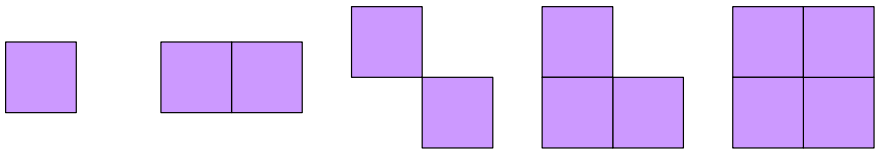
Clique

A clique $\mathcal{C}$ is a subgraph of Ω in which every node is connected to every other node,



(a) 4-Neighborhood (b) 8-Neighborhood (c) 12-Neighborhood

Figure 6.2 Examples of neighborhoods in a graph.



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

Figure 6.3 Examples of cliques for the 8-neighborhood. The entire set is composed of 8 cliques; the missing ones are rotations of the displayed cliques.

i.e. $\exists(s_i, s_j), \forall\{s_i, s_j\} \in \mathcal{C}$. A clique can be composed of a single node or a subset of the graph. Figure 6.3 displays examples of cliques. The pairwise cliques are of particular importance in our study since they will be used in the Markov random fields described hereafter.

Based on Fig. 6.1, we propose to model the distribution of the local feature vector of an object with a parametric model that will later be used as the importance density of the particle filter. The distribution is maintained to represent the local importance pdf via a Gaussian Markov random field mixture (GMRFM) . Markov random fields dispose of two desirable proprieties for the modeling of the importance density: local estimation and integration of the neighboring information. Random fields are sets of random variables X_s over an undirected graph representing dependencies between sites. A random field R is defined over a set of sites Ω as $R = \{X_s, \forall s \in \Omega\}$. For the purpose of our study, the sites are arranged in a 2-D lattice, representing the pixel locations in the image or in a downsampled version of the image. Also, for the

sake of conciseness, we use r to denote the realization of the field $R = r$.

Definition 6.1 *R is a Markov random field on (Ω, η) if the probability of the realization r depends only on the neighborhood η_s, i.e. $\forall s \in \Omega$:*

$$Pr(r_s|r_{\Omega-\{s\}}) = Pr(r_s|r_{\eta_s})\,. \tag{6.1}$$

The Hammersley-Clifford theorem defines the equivalence between a Markov random field and a Gibbs random field. The probability density function in a Markov random field is of the form

$$p(r) = \frac{1}{Z}\exp\left(-\frac{1}{T}U(r)\right), \tag{6.2}$$

where

$$Z = \int \exp\left(-\frac{1}{T}U(r)\right)dr\,. \tag{6.3}$$

The temperature T is used during the learning phase for simulated annealing and $U(r)$ is the energy function. The normalizing constant Z is intractable in practice. However, since $p(r)$ will model the importance density, it needs to be known up to a proportionality constant only. The energy function $U(r)$ can take a large variety of forms and be partially or fully dependent on the clique $\mathcal{C}$. As mentioned before, we restrict our study to pairwise cliques since the aim is to model the interaction between two sites only. Without loss of generality, the energy function is decomposed into a clique-wise potential $V_c(r)$ and a site-wise potential $V_{\eta_s}(r)$ such that

$$U(r) = \sum_{c\in\mathcal{C}} V_c(r) + \sum_{s\in\Omega} V_{\eta_s}(r)\,. \tag{6.4}$$

The function $V_{\eta_s}(r)$ carries spatial dependencies and $V_c(r)$ models the local dependencies among the sites of a clique. The clique potential $V_c(r)$ is a function of the clique c and the realization r.

6.3 Gaussian Markov Random Field Mixture

Because particle filters require samples to be easily drawn from the importance density and to reduce storage requirements, a compact and efficient representation of the density $p(r)$ is to be built. A practical representation of probability density

functions is the Gaussian function where a few parameters characterize the distribution estimate. Let $\boldsymbol{\theta}_c$ represent the set of parameters $\boldsymbol{\theta}_c = \{\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$. A unimodal pdf $p(r)$ can be approximated by a Gaussian pdf $p(r|\boldsymbol{\theta}_c)$. Consequently, the clique potential $V_c(r|\boldsymbol{\theta}_c)$ is defined as the Mahalanobis distance

$$V_c(r|\boldsymbol{\theta}_c) = \frac{1}{2}(r - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(r - \boldsymbol{\mu}_c)\,. \tag{6.5}$$

Furthermore, the spatial potential $V_{\eta_s}(r|n)$ models the dependencies of the site s on the neighborhood site $n \in \eta_s$. A penalty proportional to the square of the Euclidian distance between s and n yields

$$V_{\eta_s}(r|n) = \frac{(s-n)^2}{2\sigma^2}\,, \tag{6.6}$$

where σ is a scaling parameter. The spatial penalty gives more importance to sites n that are close to s. The two inherent probabilities, namely the clique and spatial probabilities, are defined as

$$P_c(r|\boldsymbol{\theta}_c) = \frac{1}{\lambda_c}\exp\left(-\frac{V_c(r|\boldsymbol{\theta}_c)}{T}\right), \tag{6.7}$$

and

$$P_{\eta_s}(r|n) = \frac{1}{\lambda_n}\exp\left(-\frac{V_{\eta_s}(r|n)}{T}\right), \tag{6.8}$$

where λ_c and λ_n are normalizing constants. The aforementioned assumptions yield a Gaussian distribution to the MRF, leading to the so-called Gaussian Markov random field (GMRF) . The density $p(r|\boldsymbol{\theta}_c) \propto \exp\left(-U(r|\boldsymbol{\theta})/T\right)$ is a Gaussian distribution with spatial penalty on the neighborhood η_s. However, a single Gaussian distribution narrows the scope of particle filters because it only provides a unimodal estimate of the importance density. To address this shortcoming and maintain a parametric representation of the importance density, we introduce the Gaussian Markov random field mixture. Let $\boldsymbol{\Theta}$ be the set of parameters of K GMRFs such that $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$. The pdf of the Gaussian Markov random field mixture $p(r|\boldsymbol{\Theta})$ is then defined as

$$p(r|\boldsymbol{\Theta}) = \sum_{k=1}^{K} P(k)p(r|\boldsymbol{\theta}_{c,k})\,. \tag{6.9}$$

The distribution $p(r|\boldsymbol{\Theta})$ can be seen as a local spatio-temporal mixture of Gaussians modeling the pdf of the random field R.

6.3.1 Learning and Posterior Diffusion for Sparse Random Fields

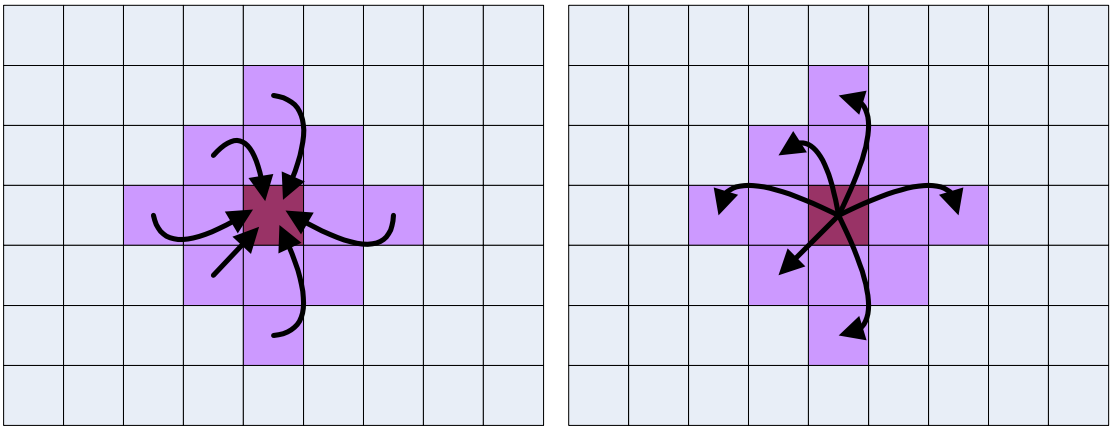
For the Gaussian MRF mixture to provide an accurate modeling of the importance sampling, a learning phase is necessary where the set of parameters $\mathbf{\Theta}$ is adjusted so that the mean square error $E[(R_{\mathbf{\Theta}} - R)^2]$ between the estimated random field $R_{\mathbf{\Theta}}$ and the true random field R is minimized. Furthermore, the problem modeled here presents the particularity of dealing with sparse realization of the random field which requires a different approach to the field update. At a given time t only a few sites l_t, where objects are located, will provide new information. The realizations are therefore composed of sporadic occurrences of random variables $X_s = \mathbf{x}_s$ localized in a limited number of sites $l_t \in \Omega_{l_t}$, where $\Omega_{l_t} \subseteq \Omega$, and $\#(\Omega_{l_t}) \ll \#(\Omega)$[1]. Consequently, the realization is reduced to $r_{l_t} = \{\mathbf{x}_1, .., \mathbf{x}_{l_t}\}$. The sparsity of the random field allows the fast update of the estimated pdf $p(r|\mathbf{\Theta})$ for each realization $\{\mathbf{x}_j : j \in l_t\}$.

Markov random fields are traditionally updated by *integration* of neighboring information at site s. We propose in this chapter to updated the MRF by *diffusion* of information at site s onto the neighborhood η_s. Recalling that, for MRFs, $s \in \eta_n \Leftrightarrow n \in \eta_s$ and that V_{η_s} and V_c are symmetric ensures the equivalence of the two methods in terms of convergence to the true random field equilibrium. Figure 6.4 is an illustration of the two different approaches. The two methods are also equivalent in terms of computation for fully populated realizations. However, when events are sparse, diffusion avoids exhaustive and inefficient update of the random field. Considering that each realization $\mathbf{x}_j$ is independent, the MRF can be updated sequentially with the l_t realizations. This results in the following equivalence for the update of the MRF:

$$p(r|\mathbf{\Theta}) \Leftrightarrow \{p(\mathbf{x}_j|\mathbf{\Theta}) : j \in \Omega_{l_t}\}. \tag{6.10}$$

It is worthwhile mentioning the case where the neighborhood of two or more realizations $\mathbf{x}_j$ are not disjoint. In such case, the estimate is dependent on the order of update. Although this does affect the transient state of R, it does not affect the asymptotic convergence. A technique to circumvent the issue regarding the order of

[1] # denotes the cardinality of a set.



(a) Integration (b) Diffusion

Figure 6.4 MRFs update with integration and with diffusion. With integration, the site s is updated with the neighborhood η_s. With diffusion, the neighborhood is updated with the local information at site s. Integration and diffusion are equivalent in terms of convergence.

update is to work with non-pairwise cliques (*e.g.* cliques with three sites) where the two events can be dealt with simultaneously. However, the non-disjoint case seldom occurs and we decided to update the random field arbitrarily in that situation noting that the convergence of the random field remains unchanged.

Similarly to Gaussian mixtures, we aim to estimate the maximum likelihood for the set of parameters $\mathbf{\Theta}$. However, since the value of the state $\mathbf{x}_{j,t}$ is not accessible directly, the maximum a posteriori (MAP) criterion is used instead. The MAP criterion is a regularization of the maximum likelihood with prior inference. It determines the optimal value k^* for the parameter index as $k^* = \text{argmax}_k[p(\mathbf{x}_{j,t}|\mathbf{\Theta}_t)]$. The aim is to build an online importance density $p(\mathbf{x}_{j,t}|\mathbf{\Theta}_t)$ from the density $p(\mathbf{x}_{j,t}|\mathbf{z}_{j,t})$ at each time step t and the set of parameters $\mathbf{\Theta}_{t-1}$ at time step $t-1$. Recalling that the random variable X_j is not accessible, the realization $\mathbf{x}_{j,t}$ is conditionally dependent on the observation $\mathbf{z}_{j,t}$. The optimal realization $\bar{\mathbf{x}}_j$ of the random variable, in the MSE sense, is given by the minimum mean square error (MMSE) $\bar{\mathbf{x}}_{j,t} = E[\mathbf{x}_j|\mathbf{z}_j]$ which leads to the MAP $k^* = \text{argmax}_k[\bar{\mathbf{x}}_j|\boldsymbol{\theta}_k], \forall k \in [1..K]$. Taking the logarithm and noting that k^* is independent of the spatial potential V_{η_s} yields

$$k^* = \underset{k}{\text{argmax}} \sum_{c \in \mathcal{C}} V_c(\bar{\mathbf{x}}_{j,t}|\boldsymbol{\theta}_k)\,. \tag{6.11}$$

Equation (6.11) defines the optimal index k^* that minimizes the KL-divergence between the pdf $p(\mathbf{x}_{j,t}|\mathbf{\Theta}_t)$ and the posterior density $p(\mathbf{x}_{j,t}|\mathbf{z}_{j,t})$. The expectation-maximization (EM) algorithm has been extensively employed for this optimization problem [66]. However, EM requires the storage of the full history $\mathbf{x}_{j,0:t}$ which is prohibitively costly. We thus opt for an online learning of the parameters $\mathbf{\Theta}$ as for the Gaussian mixture model (see Section 3.2). Also, the diffusion process enables us to restrict the clique set $\mathcal{C}$ to pairwise cliques composed of the site s and the site $n \in \eta_s$ where the pdf is evaluated. Consequently, Eq. (6.11) can be further simplified as

$$k_n^* = \underset{k_n}{\operatorname{argmax}} \left((\bar{\mathbf{x}}_j - \boldsymbol{\mu}_n)^T \mathbf{\Sigma}_n^{-1} (\bar{\mathbf{x}}_j - \boldsymbol{\mu}_n) \right), \forall n \in \eta_s \,. \tag{6.12}$$

6.3.2 Simulated Annealing

Derived from the Gibbs random field, the Markov random field enables simulated annealing in order to increase the convergence rate to the true field R. The energy function $U(r)$ is scaled by the temperature T. A cooling process is applied in order to improve the speed of convergence of the GMRFM to its true value. The temperature is updated with the number of visits v_j at site j according to a logarithmic cooling schedule

$$T_j = \frac{\lambda_T}{log(1 + v_j)} \,, \tag{6.13}$$

where λ_T is an arbitrary constant. Since the update is processed by diffusion, the visit count must integrate the spatial dependency probability $P_{\eta_s}(s|n)$, such that $v_s \leftarrow v_s + P_{\eta_s}(s|n)$ for each visit. The cooling schedule allows a fast estimation of the local pdf for the first visits and a fine estimation based on local context afterwards, hence increasing the convergence rate to the true local pdf . This is crucial due to the restricted and incomplete dataset available for behavior modeling.

6.3.3 MRF Parameters Update

As for the Gaussian mixture model in Section 3.2, the update of the parameters follows an online technique instead of the traditional maximum likelihood to avoid the costly storage of field realizations. The parameters $\boldsymbol{\mu}_{k^*,n}$ and $\Sigma_{k^*,n}$, as well as the mixing parameter $\alpha_{k^*,n} = P(k^*)$ are sequentially updated with a first order difference

equation for each realization $\bar{\mathbf{x}}_{j,t}$. The parameters are updated with the clique and spatial probabilities. Considering that the two probabilities are independent, the learning rate is thus defined as

$$\beta_n = \lambda \prod_{i=\{s,c\}} P_i = \lambda P_c(\bar{\mathbf{x}}_i|\boldsymbol{\theta}_{k^*,n})\, P_{\eta_s}(s|n)\,. \tag{6.14}$$

where λ is an arbitrary constant representing the update rate. The parameters $\boldsymbol{\mu}_{k^*,n}$ and $\boldsymbol{\Sigma}_{k^*,n}$ and $\alpha_{k^*,n}$ are then updated with first order difference equations

$$\alpha_{k^*,n,t} = (1-\beta_n)\alpha_{k^*,n,t-1} + \beta_n\,, \tag{6.15}$$

$$\boldsymbol{\mu}_{k^*,n,t} = (1-\beta_n)\boldsymbol{\mu}_{k^*,n,t-1} + \beta_n \bar{\mathbf{x}}_{j,t}\,, \tag{6.16}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k^*,n,t} = {} & (1-\beta_n)\boldsymbol{\Sigma}_{k^*,n,t-1} \\ & + \beta_n(\bar{\mathbf{x}}_{j,t} - \boldsymbol{\mu}_{k^*,n,t})^T(\bar{\mathbf{x}}_{j,t} - \boldsymbol{\mu}_{k^*,n,t})\,.\end{aligned} \tag{6.17}$$

6.4 Performance Analysis and Discussion

The Gaussian Markov random field mixture is tested on data to compare the performance of the modeled importance density for particle filtering against traditional, kinematic inference. First, the implementation of the tracking system is developed. Second, the experimental procedure is described along with the set of data. Then, the two algorithms are compared in terms of MSE and occlusion handling. Finally, the limitations of the system are discussed.

6.4.1 Object Tracking System Implementation

The proposed algorithm uses a GMRFM to model the local importance density from which samples are drawn. For each object tracked j, we assume that the set of samples $\mathbf{x}^i_{j,t-1}$ and the set of weights $w^i_{j,t-1}$ estimating the distribution of the random variable X_j at time $t-1$ are known.[2] To maintain the recursive estimation of the weights in Eq. (5.20), the likelihood $p(\mathbf{z}_{j,t}|\mathbf{x}^i_t)$, the prior $p(\mathbf{x}_{j,t}|\mathbf{x}_{j,t-1})$ and the importance density $q(\mathbf{x}^i_{j,t}|\mathbf{x}^i_{j,0:t-1}, \mathbf{z}_{j,1:t})$ must be defined. We consider the prior

[2]Note that the subscript $_j$ denotes the jth random variable while the superscript ^{i} denotes the ith sample from Monte Carlo method.

as the intrinsic evolution of the object, regardless of the contextual information, *i.e.*, represented with kinematic model $p(\mathbf{x}_{j,t}|\mathbf{x}_{j,t-1}) = \mathcal{N}(\mathbf{x}_{j,t}, A\mathbf{x}_{j,t-1}, B^2)$. The importance density is modeled with the GMRFM presented in Section 6.3 such that $q(\mathbf{x}^i_{j,t}|\mathbf{x}^i_{j,0:t-1}, \mathbf{z}_{j,1:t}) = p(\mathbf{x}^i_{j,t}|\mathbf{\Theta}_t)$, the update of $\mathbf{\Theta}_t$ being conditionally dependent on $\mathbf{x}^i_{j,0:t-1}$ and $\mathbf{z}_{j,1:t}$. The importance density is therefore local, integrating contextual information through the neighborhood and the history of $\mathbf{x}$ and $\mathbf{z}$.

The computation of the likelihood $p(\mathbf{z}_{j,t}|\mathbf{x}^i_{j,t})$ follows the procedure described in Subsection 5.3.2. However, the color pixels are taken from an elliptic zone defined by the horizontal x-axis (b_x) and the vertical y-axis (b_y) of the image and the inclination (ϕ) of the ellipse. The resampling is also performed as in Subsection 5.2.2. Algorithm 6.1 presents a sequential pseudo-code of the proposed algorithm.

Algorithm 6.1 GMRFM Particle Filter Algorithm

Require: $\mathbf{x}^i_{0,j} \sim q(\mathbf{x}_{0,j}|\mathbf{z}_{0,j})$ and $w^i_{0,j} = 1/N_S$
  for $j = 1$ to l **do**
    for $i = 1$ to N_S **do**
      $\mathbf{x}^i_{j,t} \sim q(\mathbf{x}^i_{j,t}|\mathbf{x}^i_{0:t-1,j}, \mathbf{z}_{1:t,j}) = p(\mathbf{x}^i_{j,t}|\mathbf{\Theta})$

      $w^i_{j,t} = w^i_{j,t-1}\, \gamma^i_{j,k} p(\mathbf{z}^i_{j,t}|\mathbf{x}^i_{j,t})$
    end for
    Compute $\bar{\mathbf{x}}_{t,j} = E[\mathbf{x}_{t,j}|\mathbf{z}_{t,j}]$
    Normalize $w^i_{j,t}$
    Find MAP $k^*_{n,j}$ with Eq. (6.12)
    Compute learning rate β_n from Eq. (6.14)
    Update $p(r|\mathbf{\Theta})$ via parameter $\mathbf{\Theta}$ with Eqs. (6.14)-(6.17)
    Resample $\{\mathbf{x}^i_{j,t}, w^i_{j,t}\}$ if necessary [9]
  end for

6.4.2 Experimental Procedure

The Gaussian Markov random field mixture is tested on video sequences from various semi-constrained environments which are typical of most video-surveillance scenarios (*e.g.*, airports or shopping centers). A semi-constrained environment is defined as any place where the trajectory of the object follows a well-defined path, whether explicitly defined or not. The algorithm was tested on two different datasets, characteristic of object tracking.

Table 6.1 GMRFM Particle Filter Parameter Initializing Values

Symbol	σ	λ	N_s	λ_T	$\boldsymbol{\mu}_0$	$\boldsymbol{\Sigma}_0$
Value	0.5	0.8	200	10	$\mathbf{0}$	I

Vehicle Tracking Dataset The data presented in Subsection 4.6.1 is used in the experiments. The tracking is challenging because total occlusions occur in the data due to the severe distortion of the vehicle projection on the camera plane.

People walking in a courtyard The data represents 8 hours of video surveillance footage of over 170 instances of people walking, running, cycling or wandering around in a courtyard. The difficulty with this dataset lies in the range of different behaviors and paths.

Unless stated otherwise, the algorithm is initialized with the parameter values summarized in Table 6.1 where I is the identity matrix and $\mathbf{0}$ represents the zero-column vector. The state vectors $\mathbf{x}_{0,j}$ are manually initialized. The state vector $\mathbf{x}$ is composed of the position (x, y), speed $(\dot{x}, \dot{y})$ and ellipse parameters (b_x, b_y, ϕ) of the object

$$\mathbf{x} = [x, y, \dot{x}, \dot{y}, b_x, b_y, \phi]^T . \tag{6.18}$$

Evaluation of tracking quality is a challenge in its own right because the hidden state is by definition not accessible. Usually, visual object tracking is evaluated on the expectation of the posterior sample set $\bar{\mathbf{x}} = E[\mathbf{x}|\mathbf{z}]$, either subjectively by visual inspection or objectively, through a measure comparing the estimated track with a reference track. In all cases, the evaluation is performed on observations. In this work, we propose to include the likelihood measure in the validation process to discard false correct tracking due to the distribution of the samples; thereby ensuring that the tracker is *locked* on the object. Correct tracking is achieved when the MSE is below a given threshold (fixed to 5 in our experiments) and when the likelihood is significant. We consider significant a jump in the value, from a residual value to a value sustained through time. Likelihood is represented by a change of color on the track: high likelihood corresponds to a white track, low likelihood is black. The Gaussian Markov random field mixture particle filter algorithm (called GMRFMPF for short) is compared with the CONDENSATION algorithm. They are

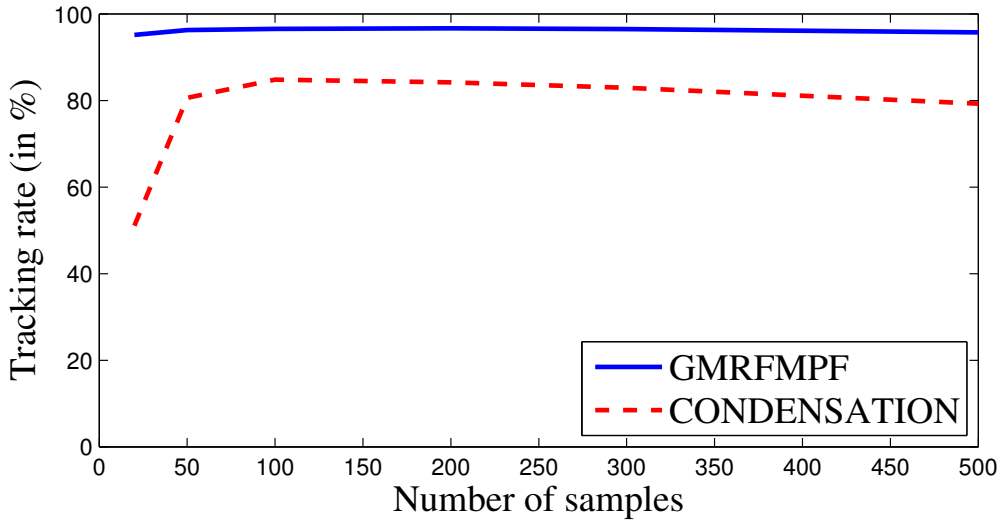


Figure 6.5 Tracking rate (in %) versus the number of samples (N) for the GMRFMPF and CONDENSATION algorithms.

differentiated by the choice of importance density: GMRFMPF uses the GMRFM while CONDENSATION uses the prior.

6.4.3 Mean Square Error Analysis

The evaluation of the mean square error is crucial to determining the quality of object tracking. The ground truth of the trajectory of 50 vehicles has been performed manually to compare with the track extracted automatically. Tracks have been extracted with the GMRFMPF and CONDENSATION algorithms. The GMRFM has been trained with the rest of the dataset as described in Subsection 6.3.3. To reduce bias in the tracking rate due to the stochastic nature of the particle filter, each track is fed into the algorithms 10 times. The tracking rate versus the number of samples N is displayed in Fig. 6.5 for the two algorithms. The proposed algorithm shows a higher tracking rate over the entire range of sample number. Furthermore, the GMRFMPF show a lower MSE on the correctly extracted tracks, characteristic of a more stable tracking. The MSE for both algorithms is summarized in Table 6.2.

Table 6.2 Comparison of the MSE for GMRFMPF and CONDENSATION

Particles Num	20	50	100	200	300	500
GMRFMPF	1.65	1.56	1.53	1.52	1.53	1.52
CONDENSATION	1.73	1.86	1.74	1.71	1.79	1.76

6.4.4 Performance with Total Spatio-temporal Occlusion

Total occlusion is challenging to resolve because visual clues are absent and the likelihood, based on observations, is unreliable. However, when samples from the importance density are efficiently spread according to contextual information, the posterior probability distribution is better estimated. Figure 6.6 (p. 147) shows the ability of GMRFMPF and CONDENSATION in handling a total occlusion of 90 frames. Figures 6.6(b) and 6.6(c) show the spread of samples through occlusion. Samples efficiently span the area of high probability of object location with the GMRFMPF and the trajectory is eventually recovered (Fig. 6.6(d)), while it is lost with CONDENSATION (Fig. 6.6(e)). The occlusion underwent 200 iterations for each sample numbers to reduce the variability due to the particle filter algorithm. The results are summarized in Table 6.3. It can be observed that the GMRFMPF consistently outperforms the CONDENSATION algorithm. The first algorithm samples particles more efficiently than the second one since the recovery rate of the object after occlusion is superior with a particle set reduced by 25 times. Figure 6.7 (p. 148) displays some results on people tracking for different occlusion scenarios: the recovery of the object is increased with the GMRFMPF and successful tracking presents a lower MSE, corroborating results in Table 6.2.

Table 6.3 Recovery Rate Under Occlusion

Particles Num	20	50	100	200	300	500
GMRFMPF	8.5%	20.5%	39%	63%	65.5%	65.5%
CONDENSATION	1%	1%	3.5%	4%	5%	8%

The case of inter-object occlusion, when an object occludes another one, is presented here for completeness. This scenario is particularly challenging because objects can share color attributes that are similar. In the case of vehicle tracking, windows and windscreens as well as plate numbers can lead to drift in the tracker from one vehicle to another. Two examples of total occlusion, along with the likelihood, are presented

in Figs. 6.8 and 6.9 (p. 149 and p. 150). In Fig. 6.8(a), the likelihood decreases rapidly because the appearance of the vehicle changes and occlusion occurs in region (1). Between region (1) and region (2), there is total occlusion and the likelihood is quasi-null. The vehicle reappears in region (3) and the track is recovered. The same can be observed from Fig. 6.9(a), except that the tracker is distracted by the occluding vehicles because the likelihood remains non-null, although smaller. After recovery in region (2), the tracker is distracted by surrounding vehicles and shadows in region (3). However, the particle sampling via the GMRFM proves to be efficient and enables the recovery of tracks after over 100 frames of occlusion.

6.4.5 When Will the Algorithm Fail?

One limitation of the proposed algorithm is the lack of implicit path in the scenery. In this context, the GMRF will show little improvement compared to CONDENSATION because the random field R will not converge to a steady-state. Nevertheless, the GMRF will not perform worse than the CONDENSATION algorithm without a kinematic model because the GMRF is initialized to provide an importance sampling $q(\mathbf{x}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{z}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}, 0, B^2)$. This situation occurs in open environments, where object trajectories are not constrained. Tracking of people on an esplanade or on a football pitch are practical examples. However, the number of scenarios displaying a true open environment is very limited. For instance, it can be argued that players on a soccer field follow some predefined paths due to team strategies or, that pedestrians on an esplanade follow a path from one point of interest to another, hence creating a semi-constrained environment.

The other limitation to the GMRF is the number of paths for a given site. If this number exceeds the number of Gaussians modeling the local distribution, the GMRF will provide a sub-optimal solutions because the importance density will not be represented accurately. For instance, two modes can be modeled by one Gaussian distribution. To overcome this issue, a GMRF with a large number of Gaussians or a Dirichlet process could be designed. However, these solutions are computationally intensive and prohibits any near-realtime tracking.

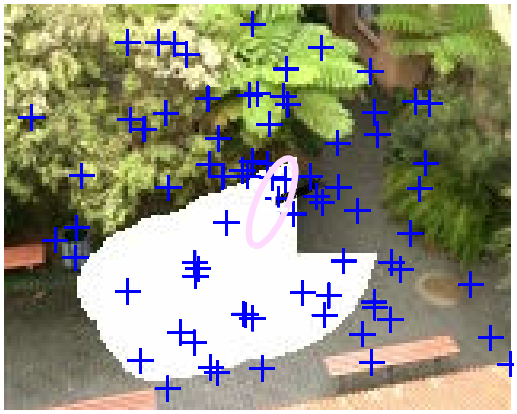
6.5 Summary of Tracking Through Occlusion

This chapter has investigated the integration of contextual information for visual object tracking. The trajectory of an object is highly correlated with the environment in which it evolves. For instance, a pedestrian will follow paths, whether explicit or implicit. We proposed to model the local context through the learning of patterns via Markov random fields. The energy function is the sum of the clique and the spatial potentials modeling the local behavior and the spatial dependencies between sites, respectively. A mixture of Gaussian MRFs has been adopted to cater for multi-modality in the pdf of the feature vector. Furthermore, since the realizations of the field are sparse (limited to a few objects), the Markov random field is locally updated and an online estimation of the parameters is performed.

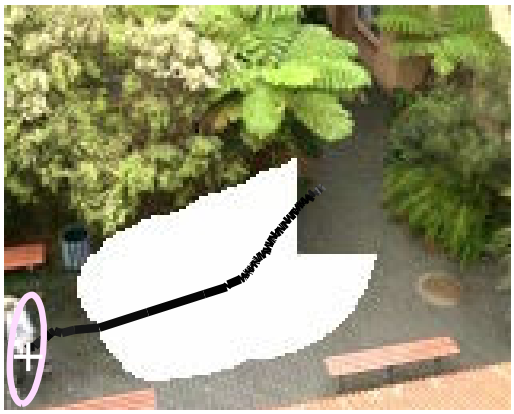
With the proposed technique, the learning of local patterns is ensured and provides inference for the particle filter; the importance density is locally modeled and yields a better distribution of the particles in the feature space. The results show that, after learning the local distribution of feature vectors, the tracking of objects is significantly improved, in particular through occlusion. More specifically, the MSE of the particle filter is reduced for a given number of particles and the recovery of tracks after large spatio-temporal occlusion is increased.

(a) Initialization

(b) Samples distribution with GMRFMPF



(c) Samples distribution with CONDENSATION



(d) Tracking with GMRFMPF

(e) Tracking with CONDENSATION

Figure 6.6 Tracking with GMRFMPF and CONDENSATION through occlusion. With the same initialization of the tracker (a), the importance density modeled with the GMRFM provides a better span of samples (b) than with CONDENSATION (c), resulting in a recovery of the track (d) after 90 frames. CONDENSATION does not recover from the occlusion (e).

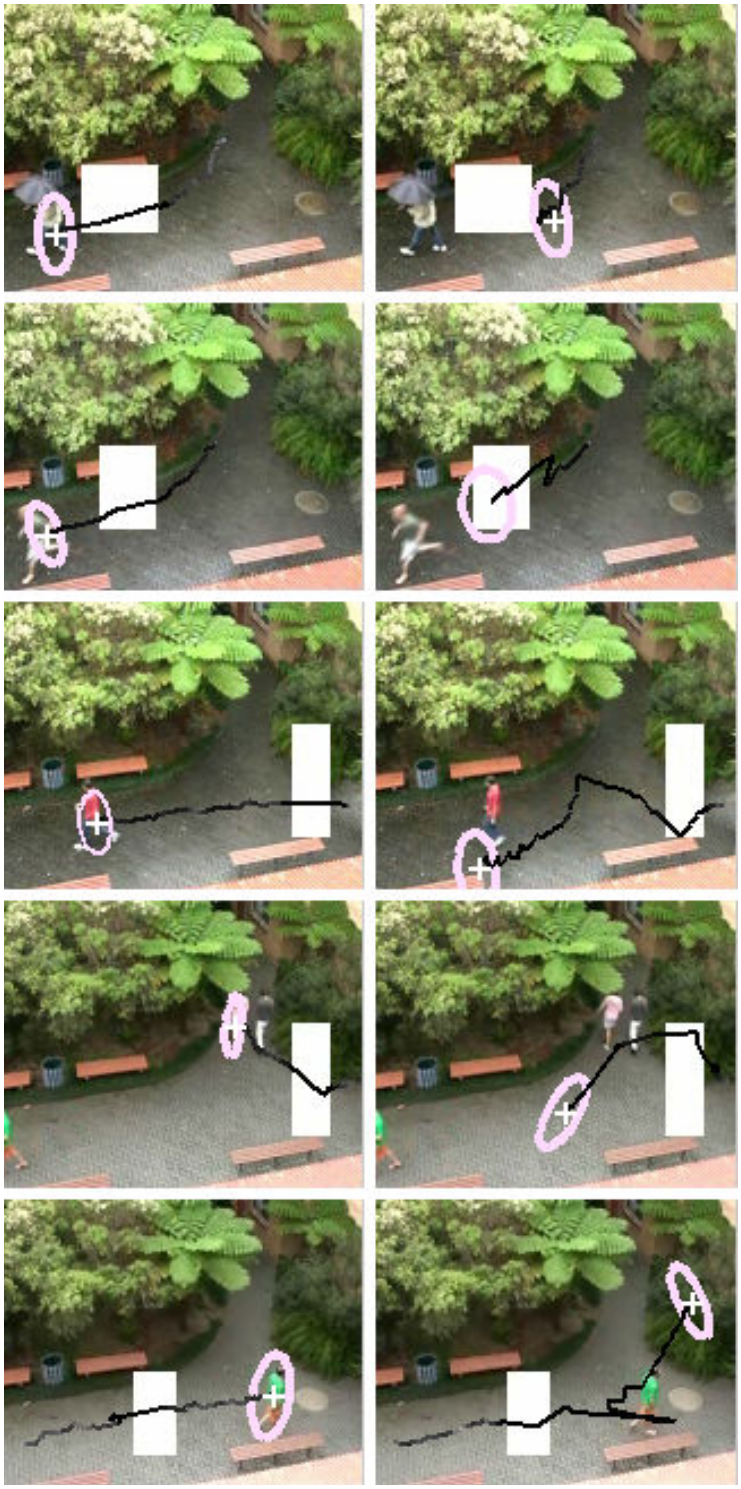
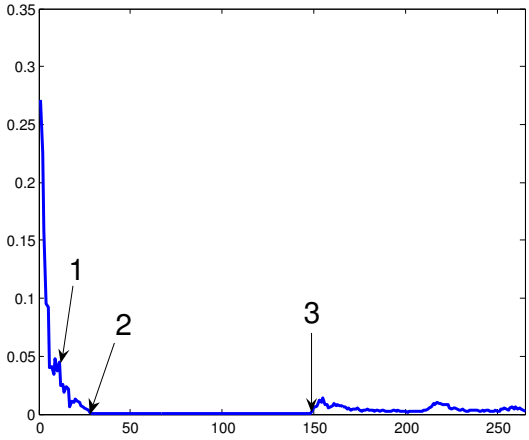
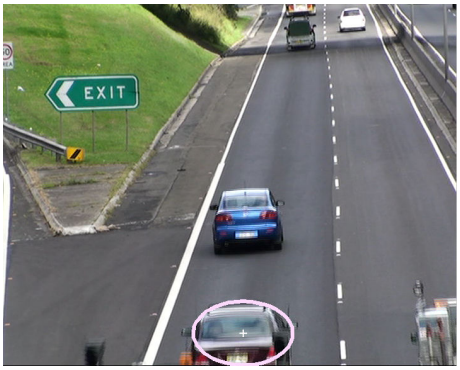


Figure 6.7 Examples of pedestrian tracking through occlusion with GMRFMPF and CONDENSATION. For each scenario (row), the GMRFMPF (left column) provides accurate tracking throughout the sequence while CONDENSATION (right column) fails to recover tracking or provides poor quality tracks for further processing (*e.g.* abnormal behavior detection).



(a) Likelihood



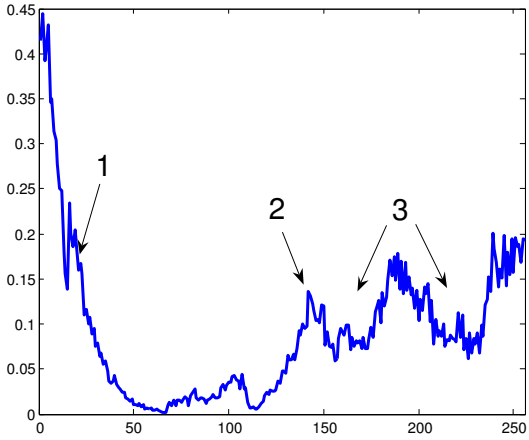
(b) Frame 1

(c) Frame 85

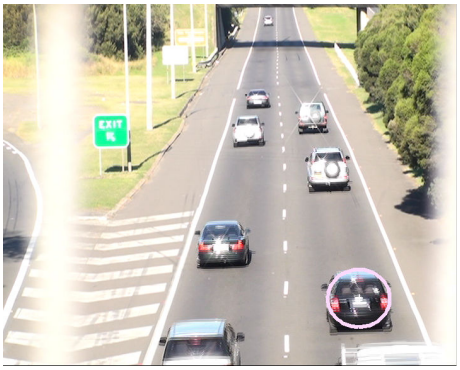
(d) Frame 200

(e) Frame 265

Figure 6.8 Vehicle tracking through large spatio-temporal occlusion. (a) The likelihood shows the transitional occlusion (from partial to total) of the object (1→2), the 100 frames of total occlusion (2→3) and the recovery of the track (3). (b) Tracking initialization. (c) Tracking through occlusion. (d) and (e) Tracking recovery.



(a) Likelihood



(b) Frame 1

(c) Frame 40

(d) Frame 211

(e) Frame 256

Figure 6.9 Vehicle tracking through large spatio-temporal occlusion. (a) The likelihood shows the total occlusion of the object (1→2), the recovery of the track (2) and distraction by visually similar objects and shadows (3). (b) Tracking initialization. (c) Tracking through partial occlusion. (d) and (e) Tracking recovery.

Chapter 7

Abnormal Behavior Detection with Markov Random Fields

7.1 Introduction

Abnormal behavior detection (ABD) via video sequence analysis has been an active topic of research over the last decade in computer vision, video surveillance and security because of the need for automation of behavior supervision. The necessity to increase security around or inside buildings has become a major priority for governments and private businesses. Airports, train stations, supermarkets, hotels or even road traffic surveillance companies are increasing their demand for ABD solutions, either to secure their infrastructure or to ensure the safety of their personnel and customers. In the long term, the outcomes of ABD developments are of the utmost importance, leading to the automatic detection of abnormal events and the notification of the relevant authority. ABD will eventually replace the passive video surveillance performed by a human operator nowadays. However, detecting abnormal behavior remains a challenging task because it is a high-level process.

In this chapter, we develop a system to detect abnormal behavior from vehicle tracks based on the Markov random fields presented in Chapter 6. The track of vehicles from the traffic surveillance dataset, extracted with the projective Kalman filter, are fed into the system to generate a map of displacements modeled by the Gaussian Markov random field mixture. The learning is performed by a stochastic clustering

algorithm to ensure a good estimate of the displacement density modes. The aim of the system is to detect abnormal behavior on highways in the form of people walking, running or cycling on the road. Section 7.2 presents some of the challenges encountered during the detection, starting with the definition of abnormal behavior in object tracking. Section 7.3 briefly reviews the work on abnormal behavior detection in object tracking. Section 7.4 introduces the proposed technique for behavior modeling. In particular, the stochastic clustering algorithm is developed in Subsection 7.4.3. Section 7.5 focuses on the analysis of the parameters for the proposed system, compares the proposed contextual approach with its global counterpart and with the Kohonen self-organizing map, another contextual approach. Section 7.6 uses the tracks from the projective Kalman filter to evaluate the performance of the system on abnormal behavior detection before concluding in Section 7.7.

7.2 Abnormal Behavior Modeling

Detecting abnormal behavior involves making high-level decisions from low-level information. There are three main challenges to ABD. First, behavior depends on both endogenous and exogenous variables describing the object. Only exogenous variables are available. Second, the definition of abnormality is subjective and depends on non-measurable factors such as culture. Third, technical constraints reduce the accuracy of low-level processes, making final, high-level decision on abnormal behavior detection a challenge.

Endogenous and Exogenous Variables

Accurate modeling of the behavior is a crucial step for ABD. The behavior of a person is defined by endogenous and exogenous variables. Endogenous variables display the behavior of the object to internal stimuli. Among others, feelings or cultural differences have a direct impact on the behavior of a person. For instance, the walking side on a path is directly influenced by the country of origin (*e.g.*, Commonwealth: left; Europe, except England: right). Although endogenous variables are undoubtedly accountable for an important part of the behavior description, they are not

directly accessible in video sequences and, therefore, cannot be used to model the behavior. Exogenous variables are more accessible because they are external to the person and thus observable. Exogenous variables range from the environment settings to the shape or the trajectory of the object. While this is clearly a limitation to the ability of behavior modeling, only exogenous variables are used to model the behavior due to the lack of information available on endogenous variables.

Subjectivity of Normal Behavior

The character of abnormality for a behavior is subjective to appreciation; a specific action can be considered as normal in some situations and abnormal in others. For example, running in a library is interpreted differently from running in a stadium. This leads to the following questions regarding the definition of abnormal behavior: how are the same type of actions interpreted in different settings? is it necessary to understand the context in which the action takes place? If yes, what is the minimum set of variables that should be taken into account? Although these questions are fundamental to ABD, it is difficult to get answers based on concrete and tangible criteria. Zhong *et al.* considered *unusual* events as "rare, difficult to describe, hard to predict and [can be] subtle" [291]. Accordingly, they defined two criteria for an unusual event. First, it must be *hard to describe* because it is unforeseen. Second, it must be *easy to verify* because it does not follow the same behavior as usual events. The definition of an unusual event can be stretched to abnormal behavior in the sense that it is hard to describe. However, the hypothesis of easy verification is too restrictive because it does not take into account the potential incomplete representation of the behavior in case of small training sets. The second condition, justified in the framework of usual/unusual event detection, cannot be applied to abnormal behavior detection. Here, an *ab-normal* behavior is defined as a behavior that diverges from normality. In terms of classification, the abnormal behavior is an outlier in the sample set. This generic definition offers the advantage of discarding any subjectivity in the discrimination of normal/abnormal behavior because it does not depend on the nature of the features modeling the object or the completeness of the data set.

Technical Constraints

Abnormal behavior is a high level task dependent on prior processing of the data. Errors generated during the capture of the sequence (*e.g.* , camera jitter, compression, camera settings, etc.) or during lower level processing (*e.g.* , background subtraction, trajectory extraction, and rounding errors) add up to provide noisy measurements on the trajectory. Also, detecting abnormal behavior requires a training set of normal behaviors, a test set of normal behaviors and a test set of abnormal behaviors. While the first 2 are easily accessible, the third one is rare and usually smaller than the first two which results in an imbalanced dataset. The sparsity of data for the abnormal behavior datasets is the primary reason for training systems on normal behavior and considering as abnormal all behaviors rejected by the system.

7.3 Related Work

Abnormal behavior detection is based on low-level tasks and an optimal solution is yet to be found. The plethora of techniques available to perform the low-level tasks does not allow a common framework for ABD. It is therefore crucial to define the major steps of the system in order to perform abnormal behavior detection. There are numerous studies on abnormal behavior detection in the literature; only an overview is presented in this section. The reader is referred to the survey on visual surveillance and behaviors proposed by Hu *et al.* for a comprehensive review on abnormal behavior detection [106]. This section presents a review of existing techniques to address the four main steps in ABD: object descriptor extraction, complexity reduction, activity modeling and behavior classification.

7.3.1 Object Descriptor Extraction

Abnormal behavior detection is based on the distribution analysis of objects descriptors. The descriptors are features extracted from the video sequences to uniquely identify an object and characterize its behavior. Most descriptors utilized for ABD are drawn from visual object tracking. They include the kinematic or trajectory information such as position and speed [36, 62, 117, 121, 161]. They are of primary importance because they define the object track which characterizes the behav-

ior at a low computational cost, hence their widespread use. Global positioning system coordinates have also been used for the same purpose [132]. Higher-level features have been employed to characterize an object. For example, Xiang and Gong not only included kinematic information but also took into account the size and first order moments of the object blob in the feature vector to provide better discrimination [271]. Templates and silhouettes are similarly processed to draw statistics by calculating the distance to the blob center or by projection on orthogonal axes [99, 268]. Histograms are also widely used to gather information on the distribution of features, and in particular, color and edge histograms [47, 147, 263]. Finally, transforms provide a convenient tool to describe the feature vector distribution. Time-frequency transforms such as discrete wavelet transform [292] or Fourier transform [62, 268], $\mathcal{R}$-transform [261] and projections [99, 240] result in a denser representation of the descriptor distribution, thus simplifying the representation.

7.3.2 Activity Modeling

There are three types of activity modeling in the literature: stochastic modeling, graph modeling and holistic modeling. Stochastic modeling measures the probability that the object moves from one state to another in a feature space. Hidden Markov models are very convenient for modeling this transition [63,261,288]. Some variants are also used to change models throughout time (*e.g.*, switching semi-Markov models [73]). Sequential Monte Carlo methods also provide accurate techniques to model the activity [59]. In particular, Vaswani *et al.* proposed to estimate the transition via particle filtering [251,252]. Bayesian networks were introduced to model activity because they have the advantage of both being structured as a graph and having probabilistic transition between nodes [36, 47, 271]. A deterministic graph was proposed by Joo and Chellappa, called attribute grammar, to categorize each type of action [123]. Zhong *et al.* introduced crowd energy, in a holistic method [292]. The energy is calculated through the Lucas-Kanade vector flow. A weighted average of the squared flow field represents the energy of the scene. Therefore, the energy in the image is analyzed without the explicit tracking of the object and abnormal behavior is detected via abnormal energy patterns. Cui *et al.* proposed to model pixel-wise activity via pixel change frequency and pixel change retainment [59].

7.3.3 Complexity Reduction

Complexity reduction is necessary when the feature vector consists of a large number of descriptors to avoid the curse of dimensionality. Without complexity reduction, the feature space dimension is high and the distribution of the feature vector is sparse. In the literature, base change, clustering, support vector machine and neural networks are utilized for this purpose. Space transformations aim to find a space where the feature distribution is better delineated. Xiang and Gong used the eigendecomposition, and Vaswani and Chellappa [249], and Calderara *et al.* [35] implemented singular value decomposition (SVD) to reduce the complexity of the feature vector distribution [271]. Principal component analysis also offers an alternative by setting an orthogonal base onto which the feature vector can be projected [268]. The principal components retain most of the signal energy, *i.e.*, most of the information; the remaining components can be discarded, hence reducing the dimensionality of the feature vector. Principal component null space analysis (PCNSA) provides the approximate null space of the feature vectors for classification [250]. Clustering reduces the complexity of the distribution because it attributes a class to each feature vector. A variety of clustering algorithms have been proposed, ranging from deterministic k-means [204] and graph clustering [291] to probabilistic clustering such as dynamic hierarchical clustering [117] or spectral clustering [6]. Finally, SVMs [47, 277] and neural networks [120] have been implemented and, in particular, self-organizing maps have been used to assign a class to data in the same fashion as the k-means clustering algorithm [62, 151].

7.3.4 Behavior Classification

Behavior classification aims to determine whether a behavior is normal or abnormal. The classification is often narrowed to a binary decision with possibly a confidence interval on the decision. In some cases, complexity reduction and classification are done in a single step. Typically, SVMs and neural networks provide classification along with complexity reduction. Similarity measures, thresholding, maximum and Bayesian probabilities are utilized in abnormality decision. Yin *et al.* determine abnormal behavior via log-likelihood [277] while Vaswani *et al.* relied on the al-

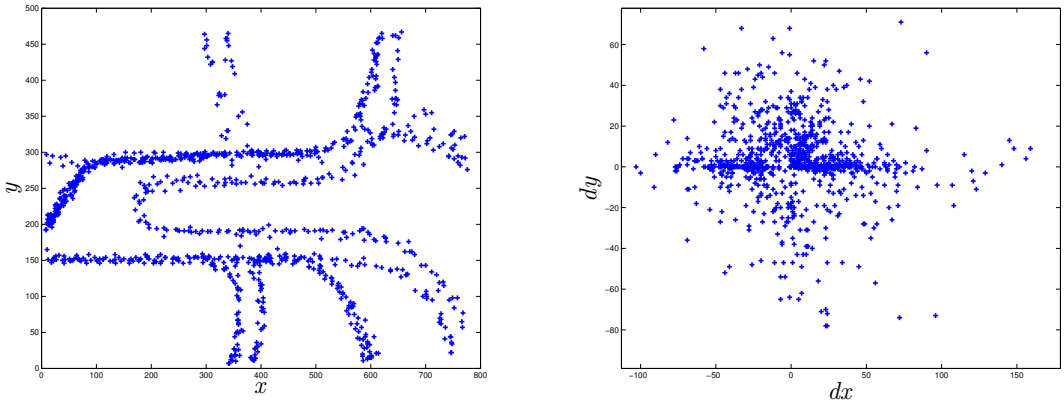
ternative expected log-likelihood [249, 252], Kullback-Leibler [251] or Mahalanobis distance [250]. All these measures determine the distance between a feature vector and the reference descriptors. Bayesian inference also provides information to evaluate the probability of abnormality [47, 73]. Eventually, thresholding is applied to obtain a binary decision [6, 74, 271]. Zhong *et al.* implemented a time-varying threshold to cope with jumps in the crowd energy [292]. Finally, when the abnormal behavior detection system produces a vector output, techniques such as maximum a posteriori and maximum-likelihood can be applied [62, 151].

7.4 Modeling Behavior with MRFs

The technique presented in this section focuses on trajectory-based ABD, that is, abnormal behavior detection concerned with modeling the density of feature vectors consisting of the object coordinates and behavioral features such as vector flow, object size, color pixels, etc. Let us consider the feature vector as a random variable X with realization $\mathbf{x}$ and further refine the analysis by differentiating between the spatial component $\mathcal{S}$ with realization $\mathbf{s}$ and the behavioral component ϕ with realization $\boldsymbol{\varphi}$. Therefore, the feature vector can be rewritten as $X = \{\mathcal{S}, \phi\}$. Markov random fields provide a convenient framework for the representation of the feature vector. As seen in Subsection 6.2.3, the non-causal dependency amongst sites allows a fast and accurate learning of patterns by integration of neighboring information. It is therefore possible to model the behavior of objects through Markov random fields in order to recognize abnormal events.

7.4.1 Feature Vector Dimensionality Reduction

Let us introduce the general framework of abnormal behavior detection before discussing the motivations for dimensionality reduction. It was illustrated in Subsection 6.2.2 that the spatial configuration of a scene was predominant in the behavior of an object. Suppose that the problem of ABD can be described as a Markov random field; one can determine the density $p(r)$ of the feature vector with conditional dependencies on the neighborhood. Furthermore, the analysis of the marginal density components provides an insight into the effect of the spatial configuration



(a) Position marginal (b) Displacement marginal

Figure 7.1 Example of marginal densities for the feature vector $[x, y, dx, dy]$. (a) Marginal density of the spatial component $\mathbf{s}$ representing the position of objects in the scene; (b) Marginal density of the behavioral component $\boldsymbol{\varphi}$, representing the density of the displacement of objects in the scene (vector flow).

on the density of the feature vector. Let us introduce the definition of the spatial marginal density $p_{\mathcal{S}}(\mathbf{s})$ and the behavioral marginal density $p_{\phi}(\boldsymbol{\varphi})$:

$$p_{\mathcal{S}}(\mathbf{s}) = \int_{\mathcal{D}_{\boldsymbol{\varphi}}} p(r)\, d\boldsymbol{\varphi} = \int_{\mathcal{D}_{\boldsymbol{\varphi}}} p(\mathbf{s}, \boldsymbol{\varphi})\, d\boldsymbol{\varphi}\,; \quad \text{and} \tag{7.1}$$

$$p_{\phi}(\boldsymbol{\varphi}) = \int_{\mathcal{D}_{\mathbf{s}}} p(r)\, d\mathbf{s} = \int_{\mathcal{D}_{\mathbf{s}}} p(\mathbf{s}, \boldsymbol{\varphi})\, d\mathbf{s}\,. \tag{7.2}$$

where $\mathcal{D}_{\mathbf{s}}$ and $\mathcal{D}_{\boldsymbol{\varphi}}$ are the respective definition domains of the components $\mathcal{S}$ and ϕ. The marginal densities provide a representation of the spatial and behavioral component spans over their respective subspaces, namely $\mathcal{D}_{\mathbf{s}}$ and $\mathcal{D}_{\boldsymbol{\varphi}}$.

The sparsity of the spatial component due to the constraints of the environment on the objects motivates the distinction between behavioral feature and spatial features. The behavioral component is *a priori* dense. Figure 7.1 shows an example of marginal densities in a vehicle traffic sequence. The set of object position is sparse and clearly follows specific patterns (called routes) while its behavior, represented by the vector flow, is dense. Consequently, the spatial marginal density $p(\mathbf{s})$ is difficult to approximate and the error in the estimation is large. In contrast, the behavioral marginal density $p(\boldsymbol{\varphi})$ is usually dense and can be estimated accurately. Although the spatial component of the feature vector accounts for most of the estimation error

on the density, it plays an important role in the analysis of the behavior because it provides information on the local environment and, thus, has to be considered for abnormal behavior detection. Markov random fields cater for the distinct roles of the two components of the feature vector since the energy function can be expressed as a combination of the behavioral component and the spatial component with different degrees of dependency. The spatial and behavioral components are handled simultaneously by Markov random fields, reducing the complexity of processing. A local characterization of the behavior yielding context-based abnormal behavior detection is therefore possible.

7.4.2 Integration of Contextual Information in the MRF

In trajectory-based tracking, the state of a feature vector $\mathbf{x}_t$ at time t can be considered smooth and be recursively updated from the state at $\mathbf{x}_{t-1}$. As noticed by Johnson and Hogg [121], the feature vector undergoes a small variation from time $t-1$ to time t:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + f_{\mathbf{s}}(\boldsymbol{\varphi}) \,, \tag{7.3}$$

where $f_{\mathbf{s}}(\boldsymbol{\varphi})$ is a local function of the behavioral component. Such a function is difficult to estimate directly because there is no knowledge of behaviors in the scene. Instead, we propose to approximate the recursive relationship in Eq. (7.3) with a Markov random field to capture the probability density in both the spatial and the behavioral domains. The MRF therefore models the necessary knowledge pertaining to the local function $f_{\mathbf{s}}(\boldsymbol{\varphi})$. The density $p(r)$ is modeled with a parametric estimate $p(r|\boldsymbol{\Theta})$, recursively updated from the knowledge accumulated over time. The probability density of the field is therefore represented by a mixture model comprising K components $p(r|\boldsymbol{\theta}_{c,k})$ such that

$$p(r|\boldsymbol{\Theta}) = \sum_{k=1}^{K} P(k) p(r|\boldsymbol{\theta}_{c,k}) \,. \tag{7.4}$$

The mixture model is identical to the one proposed in Eq. (6.9). Following the reasoning in Subsection 6.3.1 and because the purpose of MRF is to estimate the behavior of objects from trajectories, that is, the collection of states $\mathbf{X} = \{\mathbf{x}_0, ..\mathbf{x}_t\}$, the update of the field for sparse realization is adopted. However, the clique potential, being critical in the learning and the detection of abnormal behavior, will

be further investigated in Section 7.5. The two clique potentials of interest are the Euclidian distance and the Mahalanobis distance:

$$V_c(r|\boldsymbol{\mu}_c) = \frac{1}{2}(r - \boldsymbol{\mu}_c)^T(r - \boldsymbol{\mu}_c)\,, \tag{7.5}$$

$$V_c(r|\boldsymbol{\theta}_c) = \frac{1}{2}(r - \boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(r - \boldsymbol{\mu}_c)\,. \tag{7.6}$$

The temperature T in Eq. (6.2) is replaced by stochastic learning taking place outside of the density estimate $p(r|\boldsymbol{\Theta})$. Consequently, an additive shaking process substitutes the relaxation method provided by simulated annealing. The Markov random field represents the probability density of the behavioral component for a particular spatial component. Because objects in the same neighborhood tend to have the same behavior, the feature vectors at neighboring locations are highly correlated. Based on this hypothesis, we propose an algorithm that integrates information from a local neighborhood in order to update the mixture model.

7.4.3 Stochastic Clustering Algorithm

The update of the parameters for the Gaussian Markov random field is performed according to the stochastic clustering algorithm introduced by Bouzerdoum [28]. The density $p(r|\boldsymbol{\Theta})$ is temporally adjusted via the update of the set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, .., \boldsymbol{\theta}_K\}$ with $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. The parameter $\boldsymbol{\mu}_k$ represents the center (mean) of a cluster while $\boldsymbol{\Sigma}_k$ is the covariance matrix of the available samples belonging to the cluster. A stochastic procedure is adopted to allocate the clusters in the feature space for neighboring sites n. This algorithm is similar in nature to a deterministic algorithm. First, the affinity of an incoming feature vector to each cluster is computed. Second, the winning cluster is determined by maximum-likelihood. Third, the winning cluster is updated with the new feature vector. The stochastic procedure differs in that a stochastic value, drawn from a normal distribution, is added to the affinity and acts as a shaking process. The competitive learning process is defined as follows. Consider the cluster center affinity $y_{k,n}$ to the incoming feature vector $\mathbf{x}_i$:

$$y_{k,n} = P_c(\mathbf{x}_i|\boldsymbol{\theta}_{k,n}) + r_{k,n}, \tag{7.7}$$

where $r_{k,n}$ is a centered normal random variable with standard deviation $s_{k,n}$ and $P_c(\mathbf{x}_i|\boldsymbol{\theta}_{k,n}) = \lambda_c^{-1} \exp\left(-V_c(\mathbf{x}_i|\boldsymbol{\theta}_{k,n})\right)$ is the clique probability. The winning cluster ℓ^* is determined by competitive learning as:

$$\ell^* = \underset{k}{\operatorname{argmax}}(y_{k,n})\,. \tag{7.8}$$

Assuming the spatial dependency of the neighboring sites follows a normal distribution, the spatial probability is defined as $P_{\eta_\mathbf{s}}(\mathbf{x}_i|n) = \lambda_{\eta_\mathbf{s}}^{-1} \exp\left(-V_{\eta_\mathbf{s}}(\mathbf{x}_i|n)\right)$. The update rate α of the field parameters is defined as

$$\alpha_n = \lambda P_c(\mathbf{x}_i|\boldsymbol{\theta}_{\ell^*,n})\, P_{\eta_\mathbf{s}}(\mathbf{x}_i|n)\,. \tag{7.9}$$

The mixing component of the winning cluster is incremented with the spatial probability

$$P_n(k) \leftarrow P_n(k) + P_{\eta_\mathbf{s}}(\mathbf{x}_i|n)\,, \tag{7.10}$$

and the mixing components $P(k)$ are renormalized. The mean $\boldsymbol{\mu}_{\ell^*}$ of the winning cluster is updated by a first-order difference equation with learning rate α

$$\boldsymbol{\mu}_{\ell^*,n} \leftarrow (1-\alpha_n)\,\boldsymbol{\mu}_{\ell^*,n} + \alpha_n\,\mathbf{x}_i\,, \tag{7.11}$$

and the covariance matrix $\boldsymbol{\Sigma}_{\ell^*,n}$ corresponding to the cluster center $\boldsymbol{\mu}_{\ell^*,n}$ is updated as follows:

$$\boldsymbol{\Sigma}_{\ell^*,n} \leftarrow (1-\alpha_n)\,\boldsymbol{\Sigma}_{\ell^*,n} + \alpha_n\,(\mathbf{x}_i - \boldsymbol{\mu}_{\ell^*,n})^T(\mathbf{x}_i - \boldsymbol{\mu}_{\ell^*,n})\,. \tag{7.12}$$

As the cluster learns, the standard deviation of the random variable $r_{k,n}$ in Eq. (7.7) is reduced to allow convergence. The cooling schedule is performed by a counter $c_{\ell^*,n}$ incremented with the spatial probability $P_{\eta_\mathbf{s}}(\mathbf{x}_i|n)$

$$c_{\ell^*,n} \leftarrow c_{\ell^*,n} + P_{\eta_\mathbf{s}}(\mathbf{x}_i|n)\,, \tag{7.13}$$

and the standard deviation of the shaking process is updated as follows:

$$s_{\ell,n} = s_0/c_{\ell^*,n}\,. \tag{7.14}$$

The "shaking" process introduced in the clustering algorithm improves the convergence of the cluster centers to the modes of the density [28]. Indeed, because of the on-line nature of the learning algorithm, the initialization of the center value is critical. For example, if the center is initialized on an outlier, a standard learning

algorithm may not converge to a relevant mode of the density. It should also be noted that if the number of outliers increases, the stochastic algorithm performs better in its ability to find the cluster center [28]. In such a case, the standard deviation rate of decrease can be lowered to allow more shaking in the learning phase. This provides a better convergence of the parameters $\boldsymbol{\mu}_{\ell^*,n}$ to the modes; on the other hand, the convergence will be slower. With the randomness introduced by the stochastic clustering algorithm, the cluster center is shifted around until the number of samples is large enough for an accurate estimation of the mode position.

7.5 Analysis of the Stochastic Learning Algorithm on Synthetic Data

This section is dedicated to the evaluation of the stochastic learning algorithm for different parameters and clique functions. The stochastic learning algorithm is tested on synthetic data to detect abnormal behavior. The scope of abnormal behavior is narrowed in this section to the detection of drivers under the influence of alcohol on highways. After tuning of the parameters, a first experiment is conducted to compare the performance of the local behavior modeling with MRFs and the global behavior modeling with a Gaussian mixture model. A second experiment aims to evaluate the performance of the stochastic learning algorithm versus the Kohonen self organizing map.

7.5.1 Experimental Setup

The system is tested on synthetic data modeling the behavior of driving under the influence of alcohol as abnormal behavior. Synthetic data is used due to the difficulty of obtaining real data. It has been shown that consumption of alcohol to a rate of 0.05% of breath alcohol content (BAC), the standard limit in most European countries, increases the variance in trajectory by 3.2 on average [41, 157]. For the experiments, different scenarios of car flows are generated representing typical car trajectories; *e.g.*, roundabouts, intersections crossing, etc. An example of sequence used for the simulation is displayed in Fig. 7.2. The set of data is divided into 3 subsets of 11,900 samples (feature vectors), each representing 50 tracks of 238 steps.

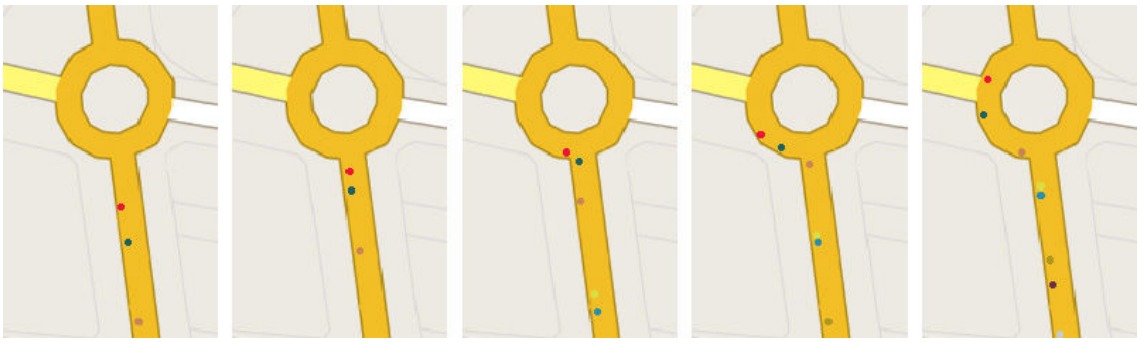


Figure 7.2 Example of generated vehicle tracks used in the experiments. The synthetic sequences are generated on road maps in order to provide realistic scenarios; *e.g.*, intersections, roundabouts, turns, etc. The dots represent cars on the road. From *left* to *right*: frames 10, 14, 18, 22 and 28.

Two sets of trajectories with variance equal to 2 are used to train and test the system on normal behavior. The third set with variance equal to 6.4 is used to test abnormal behavior. The algorithm described in Section 7.4 is tested with a feature vector $\mathbf{x}$ composed of a spatial component $\mathbf{s}$ (position) and a behavioral component $\boldsymbol{\varphi}$ (vector flow)

$$\mathbf{x} = \begin{pmatrix} \mathbf{s} \\ \boldsymbol{\varphi} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ x_t - x_{t-1} \\ y_t - y_{t-1} \end{pmatrix} , \tag{7.15}$$

where x and y are the cartesian coordinates of the object position.

The criterion of abnormality is of primary importance in the evaluation of behavior. There are two different approaches for estimating abnormal behavior from the stochastic learning algorithm. The first one is to consider that the Markov random field models the density of normal behavior, *i.e.* each component of the mixture model contributes to the modeling of the density at site $\mathbf{s}$. In such a case, the behavior is classified as

$$\begin{cases} p(r|\boldsymbol{\Theta}) > T & \rightarrow \text{“}normal\text{”} , \\ p(r|\boldsymbol{\Theta}) \leq T & \rightarrow \text{“}abnormal\text{”} , \end{cases} \tag{7.16}$$

where T is a constant threshold. The estimation of the probability density $p(r|\boldsymbol{\Theta})$ determines whether a behavior is abnormal or not. The higher the probability is,

the higher are the chances for an object to have a normal behavior. Indeed, the probability is high when the feature vector fits well the model, that is, when the object behavior is predictable. On the contrary, low probability means the behavior is unpredictable, hence considered as abnormal.

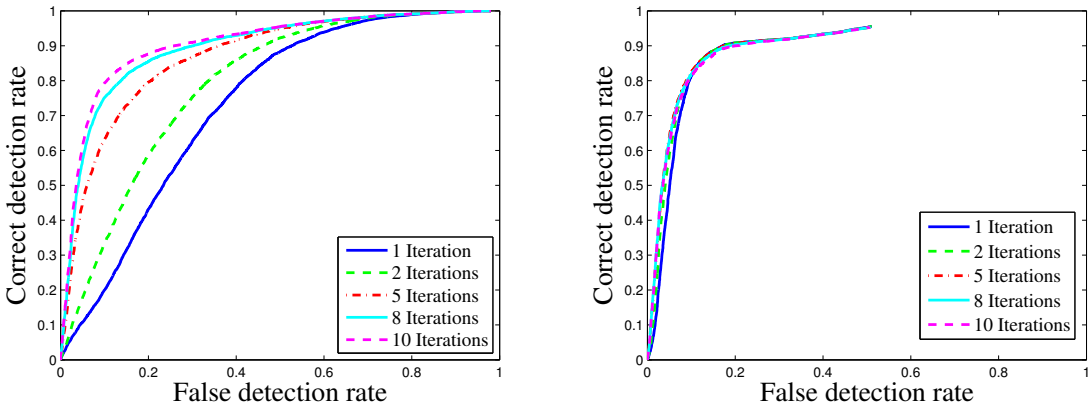
A second approach consists in thresholding the clique probability, *i.e.* the distance of the feature vector $\mathbf{x}$ to the winning cluster ℓ^* as follows

$$\begin{cases} V_c(\mathbf{x}|.) \leq T & \rightarrow \text{“}normal\text{”} , \\ V_c(\mathbf{x}|.) > T & \rightarrow \text{“}abnormal\text{”} , \end{cases} \tag{7.17}$$

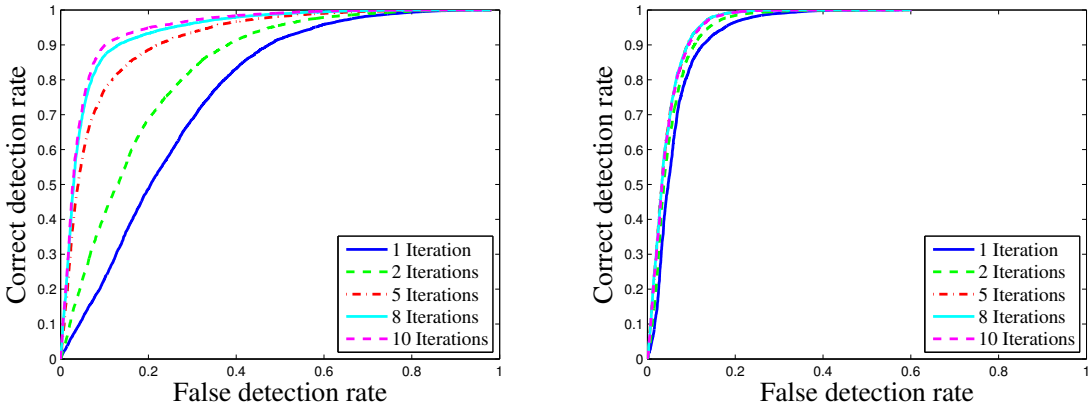
where $V_c(\mathbf{x}|.)$ represents either the Euclidian or the Mahalanobis distance. The components of the mixture model are considered independently here. Indeed, the normal or abnormal character of a behavior is estimated based on a unimodal hypothesis. While the first approach considers the fit of the feature vector to the entire density, this approach rather estimates the fit to the closest cluster in terms of clique potential $V_c(\mathbf{x}|.)$. The definition of a normal and abnormal zone for each component of the model motivates such an approach. Here, each mode of the density models a possible behavioral transition, and then only, the abnormality test is carried out. The two approaches are fundamentally different in the essence of abnormal behavior detection.

7.5.2 Distance Measure Selection

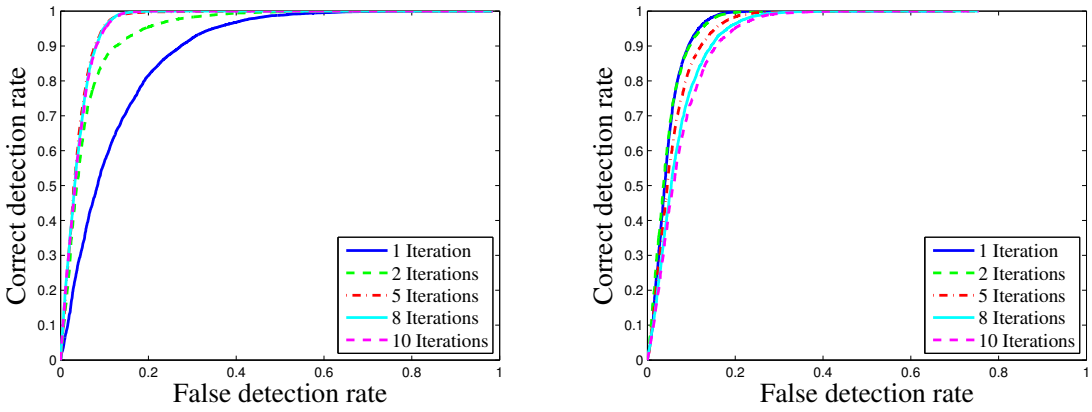
The performances of the two different techniques are compared in Figs. 7.3 and 7.4 with the two different clique potentials $V_c(\mathbf{x}|\boldsymbol{\theta})$ and $V_c(\mathbf{x}|\boldsymbol{\mu})$, respectively. The displays represent the correct detection rate versus the false detection rate (ROC curves) for different values of the parameter σ^2 for the spatial potential (see Section 6.3) and for the two definitions of abnormal behavior introduced in Eqs. (7.16) and (7.17). The top rows display the ROC curves with the implementation of the Euclidian distance $V_c(\mathbf{x}|\boldsymbol{\mu})$ and the bottom rows with the implementation of the Mahalanobis distance $V_c(\mathbf{x}|\boldsymbol{\theta})$, *i.e.* when the covariance matrix is taken into account. The figures show that the correct detection rate for a given false detection rate increases with the value of σ^2. Indeed, the larger the value of σ^2 is, the more weight neighboring locations have in the estimation of the density. In general, this rate also



(a) $\sigma^2 = 0.2$

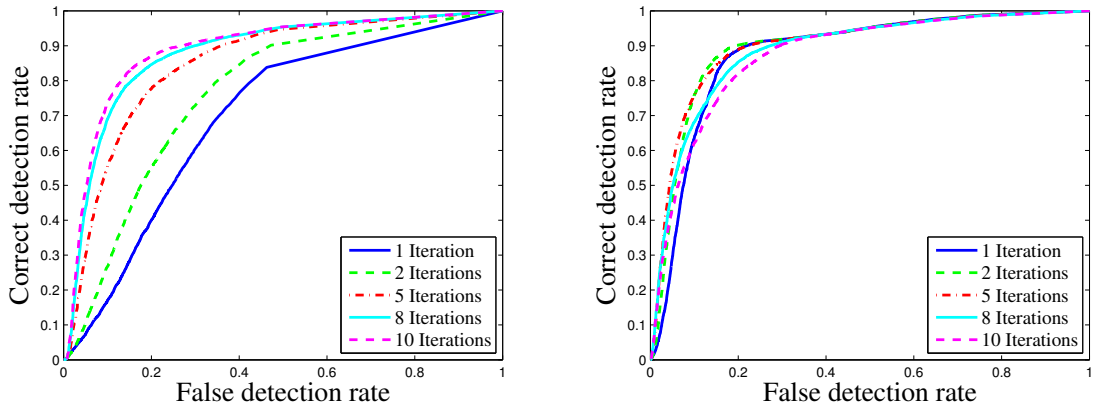


(b) $\sigma^2 = 0.5$

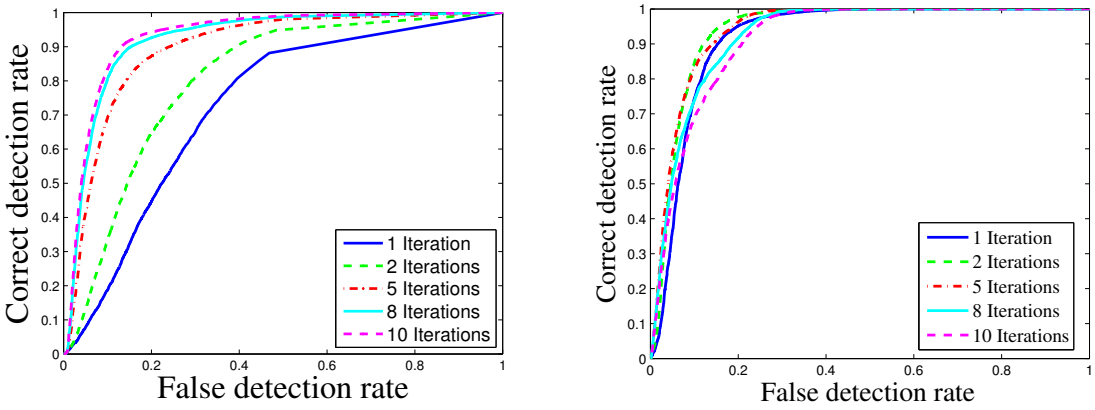


(c) $\sigma^2 = 0.1$

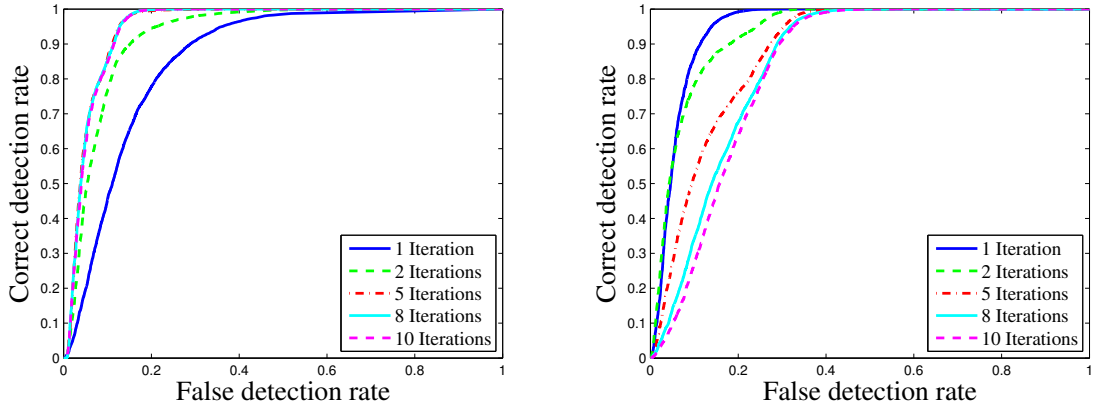
Figure 7.3 ROC curves for the stochastic clustering algorithm and for abnormal behavior detection based on distance from the cluster for different values of neighborhood variance. *Left column*: Euclidian distance ABD-based; *right column*: Mahalanobis distance ABD-based.



(a) $\sigma^2 = 0.2$



(b) $\sigma^2 = 0.5$



(c) $\sigma^2 = 0.1$

Figure 7.4 ROC curves for the stochastic clustering algorithm and for abnormal behavior detection based on probability density $p(r|\boldsymbol{\Theta})$ for different values of neighborhood variance. *Left column*: Euclidian distance ABD-based; *Right column*: Mahalanobis distance ABD-based.

increases with the number of iterations on which the system is trained. However, the right column of Figs. 7.3(c) and 7.4(c) present a decreasing correct rate with the number of iterations. This is due to the faster learning of the stochastic algorithm with the Mahalanobis distance. Indeed, it can be seen that the number of iterations has less influence on the ROC curves when the Mahalanobis distance is implemented than when the Euclidian distance is. The degradation of the performance displayed in the right column of Fig 7.3(c) is imputed to over training of the system with normal behavior. Even though the Mahalanobis distance has the desirable property to scale the density of each mixture component, it decreases the performance of the algorithm when trained with a large number of iterations. Indeed, the covariance matrix does not converge to its true value because the algorithm is trained with samples of the normal behavior subset and not the entire true set, which is not known. With the Euclidian distance, the covariance matrix is ignored and the number of iterations fine tunes the cluster center. The implementation of a variable variance σ^2 addresses the problem of over training. Setting the variance to be large for the first iterations increases the influence of the neighborhood on the estimation of the density; the update of the mixture component parameter is accelerated providing a fast convergence with few iterations. The neighboring variance is then reduced with learning such that $\sigma^2 \leftarrow \sigma^2/(1+c)$, where c is the counter defined in Eq. (7.13). The reduction of the variance limits the over training as shown in Fig. 7.5.

The comparison between Fig. 7.3 and Fig. 7.4 show that the cluster distance approach provides slightly better results than the probability density approach. The detection of abnormal behaviors as normal is accountable for the decrease in performance. Indeed, a behavior deemed to be abnormal with regards to each cluster can be considered as normal when the overall probability density is considered because the probabilities of belonging to each component add up. The poor results shown in Fig. 7.4(c), bottom row, corroborates this comment: the more the neighborhood is integrated in the density, the more components will be added up, leading to an increase of false detection rate for a given correct detection rate.

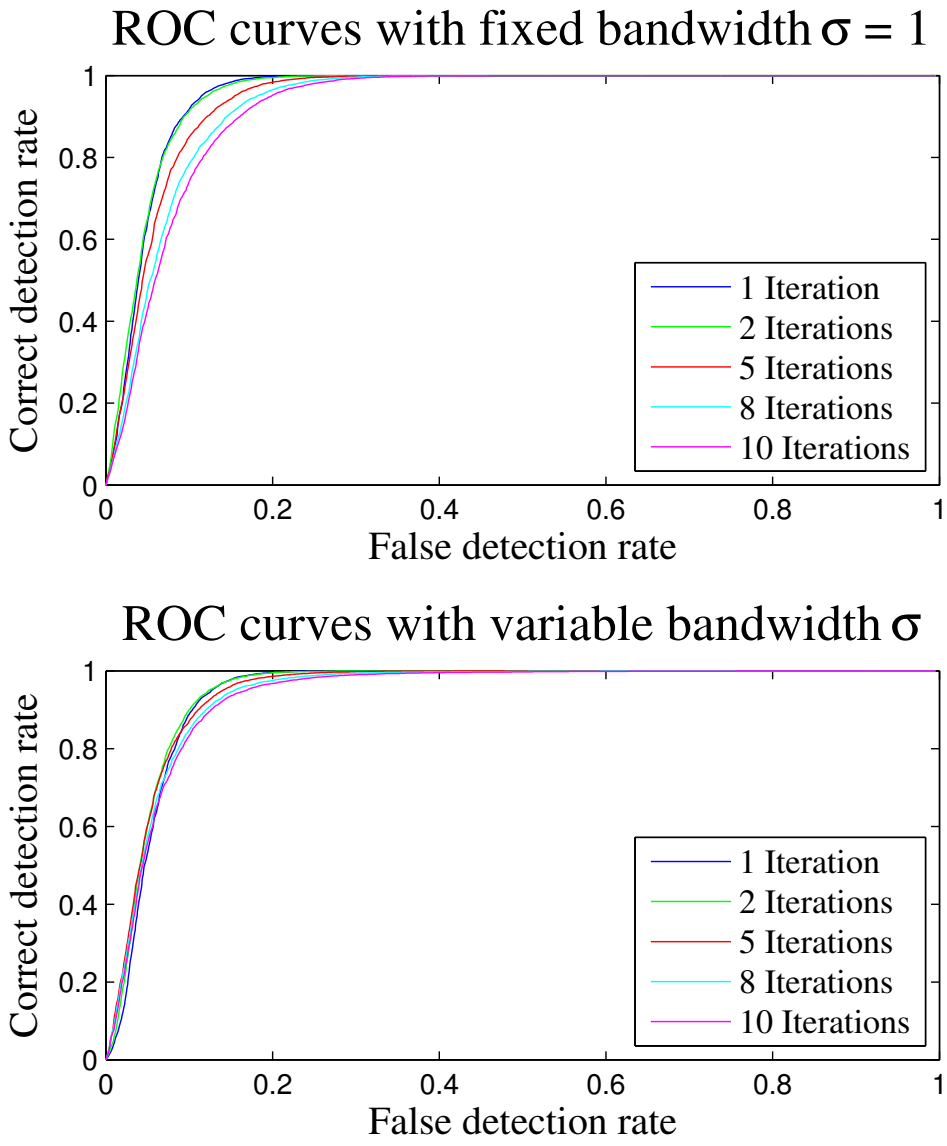


Figure 7.5 ROC curves for the stochastic learning algorithm based on the Mahalanobis distance measure. *Top*: implementation with a fixed neighborhood variance $\sigma^2 = 1$. *Bottom*: implementation with a variable variance.

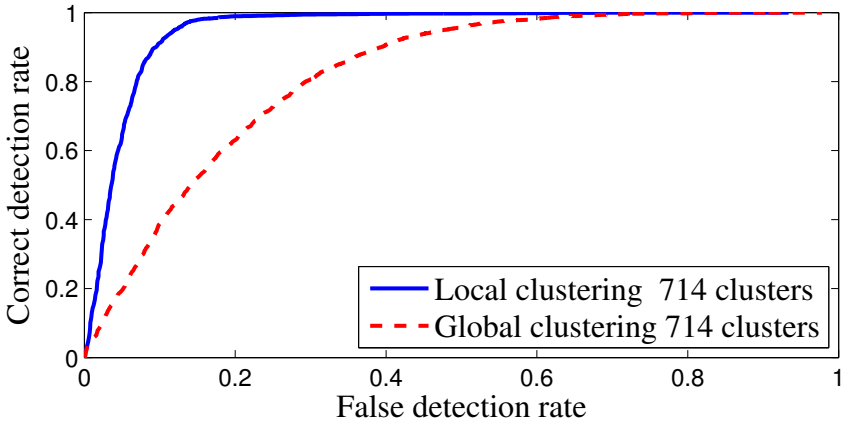


Figure 7.6 ROC curves of stochastic learning algorithm for abnormal behavior detection. Comparison between the local implementation and the global implementation.

7.5.3 Performance Analysis

The algorithm is tested on the dataset described in Subsection 7.5.1. The stochastic learning algorithm using the Mahalanobis distance with a variable rate σ^2 is implemented. Also, since ABD based on the entire density increases the computation load and does not improve the results, the detection based on distance is implemented. The results displayed in this subsection were run on the entire dataset for 10 iterations.

Global versus Local Stochastic Clustering

In this subsection, we propose to compare the performance of the local stochastic learning implemented through the Markov random field with its global counterpart, a mixture of Gaussians modeling the entire feature space as proposed by Johnson and Hogg in [121]. The global stochastic learning consists of a set of 714 clusters. The feature vector $\mathbf{x}$, for the global approach, is composed of the spatial and the behavioral components. Figure 7.6 presents the ROC curve for both implementations. It is clear that the local approach performs better for the entire range of false detection. For instance, a false detection rate of 10% leads to a correct detection rate of 39% for the global approach and 89% for the local approach. The latter performs better because the Markov random field integrates spatial and behavioral

dependencies in a common framework.

For the global approach, an error is introduced during the estimation of the marginal spatial density $p_{\mathcal{S}}(\mathbf{s})$. Indeed, for a given number of clusters, the global approach shows a larger average distance between the cluster centers and the feature vectors than the local approach. The global approach fails to reach a correct detection rate of 90% for a false detection rate of 40%. The inadequate estimation of local feature densities is mostly responsible for such a low performance. The strength of our abnormal behavior detector resides in its ability to model the normal behavior from an incomplete dataset. Indeed, in car traffic surveillance, a large and complete set of training data is not always available. It is then critical that the system adapts quickly. The proposed method is particularly well suited to such a scenario due to the diffusion of probability density modes to neighboring sites.

Local Distribution Learning vs. Self Organizing Maps

The proposed algorithm is compared with the self organizing map (SOM) developed by Dahmane and Meunier [62]. The Euclidian distance is implemented in the clique potential and the behavioral component of the feature vector is taken as the vector flow. The feature vector for the SOM is as in Eq. (7.15). SOMs have proven to give good results on abnormal behavior detection because of their property of topology conservation [62, 151]. This characteristic is particularly desirable when the feature vector is based on position since the neighborhood of the winning neuron is updated with the feature vector. The inclusion of the neighborhood in the modeling process confers the proposed approach with the topology advantage of SOMs, whilst decreasing the feature vector size by the dimensionality of the spatial coordinates.

The performance of the proposed algorithm is compared to that of a SOM which models the global probability density. The proposed approach models the local probability density with a fixed number of clusters K; thus, the total number of clusters required is $K \times N_{\mathbf{s}}$, with $N_{\mathbf{s}}$ being the number of sites. For the SOM, the number of neurons is $h \times w$ where h and w are the height and the width of the map. For comparison purposes, the SOM is composed of 729 neurons (size $[27 \times 27]$) and the proposed algorithm is trained with 714 clusters ($K = 3$ and $N_S = 238$).

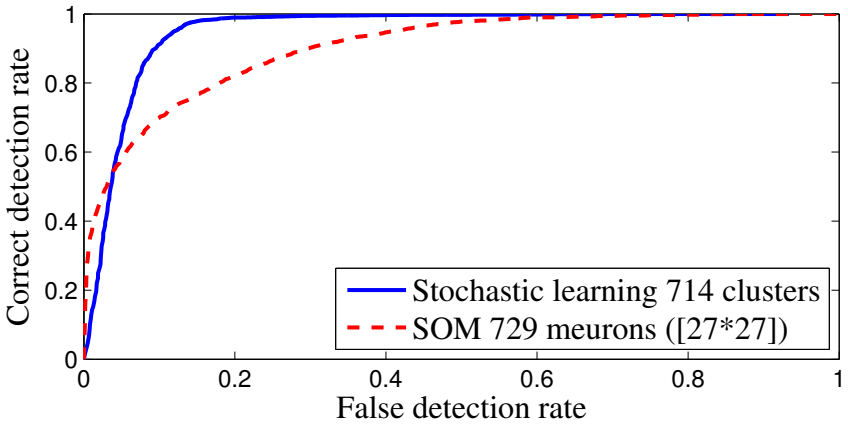


Figure 7.7 ROC curves for the proposed technique and the SOM.

Table 7.1 Correct ABD Rate with MRFs

False Detection	7.5%	10%	12.5%	15%	17.5%	20%
238 clusters	76.8%	83.5%	86.0%	88.0%	89.2%	90.4%
476 clusters	77.7%	85.6%	89.7%	92.0%	93.6%	94.9%
714 clusters	81.0%	89.1%	92.9%	94.7%	95.6%	96.5%
952 clusters	81.7%	89.9%	94.0%	96.2%	97.1%	97.5%
1190 clusters	82.7%	90.2%	95.5%	98.0%	99.0%	99.3%

Figure 7.7 displays the ROC curves of both algorithms. It can be inferred that the proposed algorithm gives better performance, for correct detection rates of 60% and higher. Note that a high rate of correct detection takes precedence over low false detection in most applications. The SOM and the proposed method have also been compared for different number of clusters/neurons; the results are presented in Tables 7.1 and 7.2. The detection rate increases with the number of clusters/neurons for a given false detection rate in both cases. However, the local approach systematically outperforms the SOM, except for a false detection rate of 7.5% with 1190 clusters.

Table 7.2 Correct ABD Rate versus Size of SOM

False Detection	7.5%	10%	12.5%	15%	17.5%	20%
Size [15 16] (240 Clusters)	50.5%	55.6%	61.1%	64.7%	68.2%	70.6%
Size [22 22] (484 Clusters)	63.3%	67.7%	71.6%	74.4%	77.1%	79.4%
Size [27 27] (729 Clusters)	64.6%	69.9%	73.9%	76.7%	79.6%	82.0%
Size [31 31] (961 Clusters)	81.5%	85.0%	87.6%	89.3%	90.7%	91.8%
Size [34 35] (1190 Clusters)	85.8%	89.6%	91.7%	93.2%	94.3%	95.0%

7.6 Abnormal Behavior Detection on Highways

This section is dedicated to the evaluation of the system developed in Section 7.4 on real video sequences of highway traffic. The challenges encountered with vehicle traffic video dataset as well as the experimental setup are described in Subsection 7.6.1. The performances of the algorithm is presented in Subsection 7.6.2. Finally, we discuss the performances of the proposed algorithm and propose further improvements in Subsection 7.6.3.

7.6.1 Experimental Setup

The proposed algorithm is tested on a set of trajectories extracted from the video surveillance dataset described in Subsection 4.6.1. Also, as mentioned previously, the wide range of settings can be a source of errors that reduce the performance of the ABD system. The projective Kalman filter, proposed in Section 4.4, is implemented to reduce the error in trajectory estimation by integrating the camera calibration settings into the tracking algorithm. The trajectories are extracted with this technique and directly fed into the proposed algorithm; no postprocessing is performed on the data because of computation load constraints. The trajectories are learnt for each sequence individually since the settings vary from one video to another. The trajectory-based feature vector is composed of the position and the vector flow of the vehicle (see Eq. (7.15)). The 15 videos in the dataset contain only normal behaviors, which are used to train and test the system. In addition to this data, a video (Video_016) containing both normal and abnormal behaviors is tested. Sample frames from Video_016 sequence, representing abnormal behaviors on a highway, are displayed in Fig. 7.8.

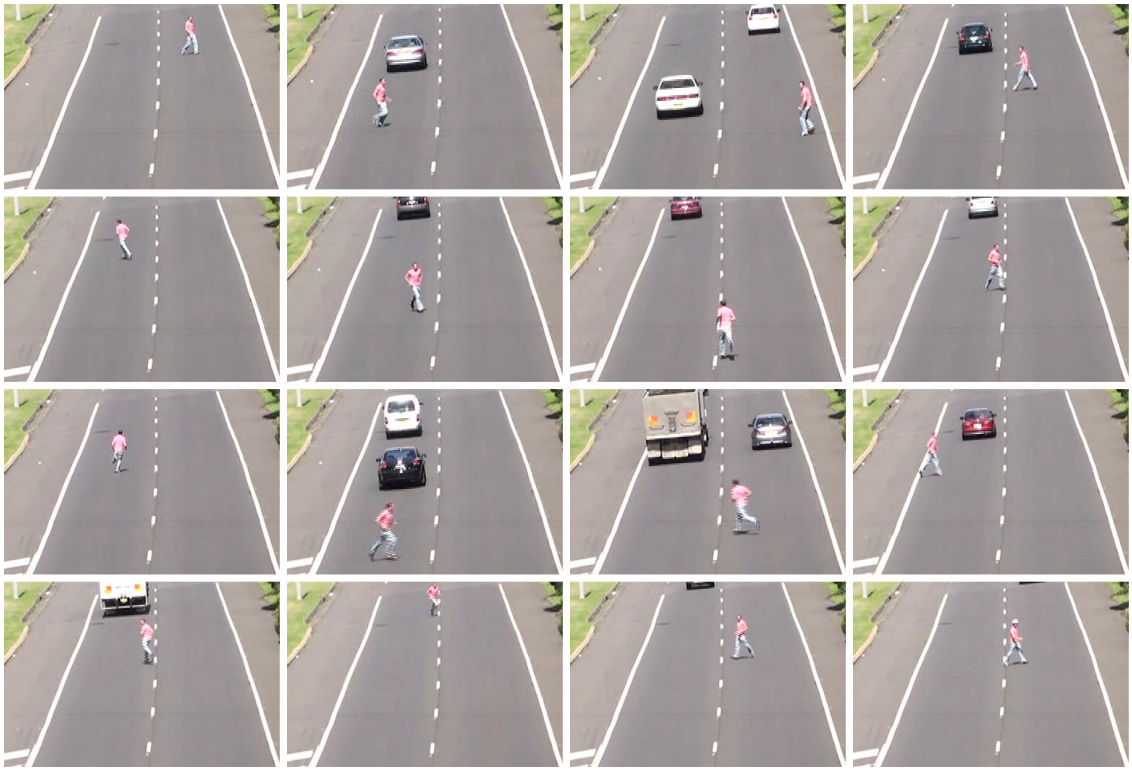


Figure 7.8 Examples of abnormal behavior on highways.

Normal behavior is learnt with cars driving on the highway at normal speed ($\approx 25\,\mathrm{m.s^{-1}}$). Abnormal behavior consists of a person walking or riding a bike on the highway. There are 20 recorded trajectories for abnormal behavior while more than 300 vehicles represent normal behavior in the video sequence. Consequently, normal behavior is modeled and abnormal behavior is detected as defined in Eq. (7.17), that is, trajectories not fitting the learnt model are considered abnormal. The threshold T in Eq. (7.17) considerably simplifies the problem in terms of abnormal behavior definition and computation.

7.6.2 Performance Analysis

In this subsection, the threshold T in Eq. (7.17) is fixed so that an average 10% false detection rate is allowed. The variable neighboring parameter is implemented and the system is trained with 10 iterations. A preliminary experiment on the video sequence containing abnormal behavior showed that a threshold of $T = 0.0344$

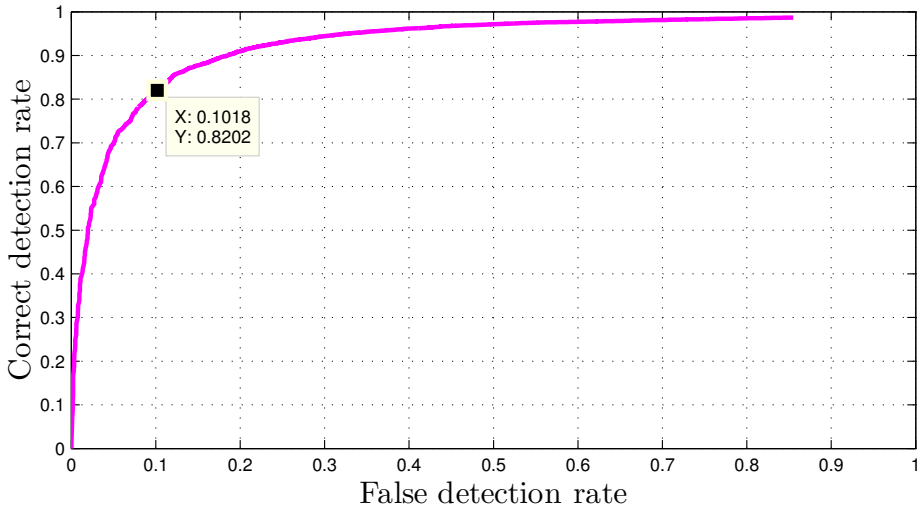


Figure 7.9 ROC curve for the video sequence including abnormal behavior. The curve is explored by tuning the parameter T. The value of 10% false detection rate gives a threshold value $T = 0.0344$.

achieves a 10% false detection rate; the expected correct detection rate is about 82% as shown in Fig. 7.9.

The algorithm is first tested on a pool of 15 videos representing normal behavior. The training for the estimation of the correct detection rate follows a 5-fold cross validation process. More precisely, four fifths of the trajectory dataset for each sequence is used for training and one fifth for testing. The five subsets are shifted around to either train or test the system. The results are then averaged over the 5 runs. The 5-cross validation process ensures that all data have been used in training and test sets. The results are summarized in Table 7.3. The average correct detection rate is 86.2%. The variation in the tracking rate for each video is due to the errors introduced by low-level tasks as described in Section 7.2. Video_004 presents the lowest correct tracking rate. The weak performance of the system on this video is due to the speed variation of vehicles. Indeed, because Video_004 captures a close view of the highway, since $D = 29$m (see Table 4.1), the accuracy of the object position is reduced and the classification by the systems is impaired. Normal behavior is characterized by a specific speed and direction of displacement of the

Table 7.3 Correct ABD Rate on the Video Dataset

Videos	Correct Det.
Video_001	88.4%
Video_002	78.5%
Video_003	80.5%
Video_004	70.5%
Video_005	80.0%
Video_006	88.4%
Video_007	80.8%
Video_008	83.0%
Video_009	90.3%
Video_010	86.6%
Video_011	93.0%
Video_012	96.5%
Video_013	94.4%
Video_014	90.6%
Video_015	91.6%
Average	86.2%

vehicles. After sufficient training every object not matching with these conditions is considered as abnormal.

Figure 7.10 displays the classification of each displacement in Video_016. It can be observed that the tracks of the vehicles (vertical) are considered normal (*blue*) in most cases. The false positive detections (normal behaviors considered abnormal) are due to tracking errors. Two cases can be differentiated: track loss and track uncertainty. In the first case, the tracker on the vehicle undergoes large variations in position when the track is lost because the mean-shift does not converge to the vehicle center with the projective Kalman filter. This results in displacements that do not fit the estimate of the density, hence abnormal classification. The second occurs if the bandwidth and the center estimates of the vehicle position are not accurate. This leads to smaller errors because the track is not lost. However, these variations are sufficient to misclassify the behavior as abnormal. Solutions to overcome these temporary misclassifications are discussed in Subsection 7.6.3. On

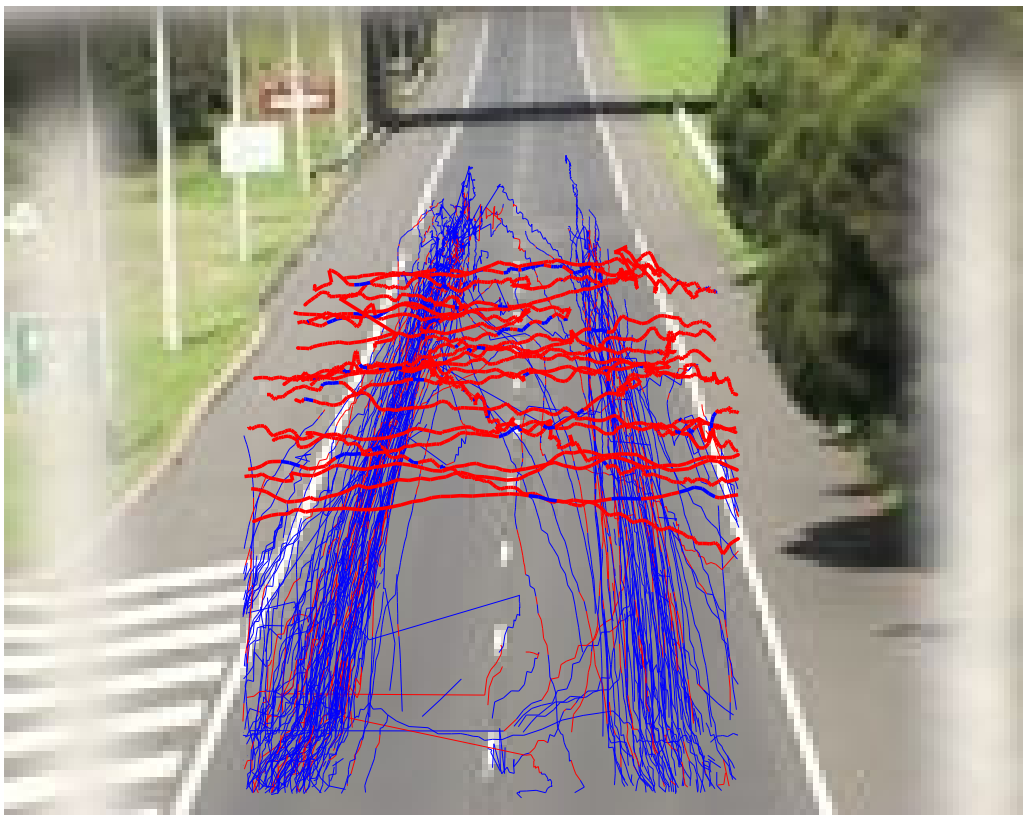


Figure 7.10 Abnormal behavior detection rendering for a system trained and tested on real data. *Blue* represents the normal behavior; *red* the abnormal behavior.

the other hand, the person walking and cycling on the highway has an abnormal behavior. The system correctly detects the abnormality because the trajectories do not fulfill the conditions on speed and direction.

7.6.3 Discussion

The performance on the video sequence dataset shows the efficiency of the proposed technique based on local mixture models. The variable neighboring variance σ^2 limits the over training and reduces the number of normal behavior considered as abnormal. The correct detection rate of 82% obtained from Fig. 7.9 for Video_016 is surpassed, on average, when the entire dataset is tested. However, it can be observed that the performance varies largely from one video sequence to another. Because the parameters are identical, the difficulty of trajectory extraction and the number of vehicle tracks available for training are the primary causes of such

variations. Indeed, even though the Projective Kalman filter improves the tracking of vehicles, some noise is still present in the trajectory extraction, hence reducing the performance of ABD system. Despite the error accumulated from lower level processing, the system is still highly discriminative and provides good results for such a challenging problem. It can be noted from Fig. 7.8 that the pedestrian is running along a path tangential to the vehicle track in some cases. However, the system still detects the path as abnormal because the speed is too low.

The local approach to detecting abnormal behavior addresses the issue of severe distortion when the scene is projected onto the camera plane. In the application of ABD on highways, the projection induces large variation in apparent speed that could not be efficiently modeled in a global approach. The local approach can also provide the framework for more advanced discrimination. In this chapter, the feature vector has been limited to the position and the vector flow of objects for comparison purposes. However, an increase in the feature vector can improve the discrimination and lead to even more accurate decisions. Typically, the feature vector can be augmented with a color representation providing information on the context. A concrete example is the integration of the traffic light color at an intersection. Regarding abnormal behavior detection, the study was restricted to the classification of elementary displacements. Elementary displacements provides low level analysis of the behavior. In the same way as data can be filtered to remove the noise or clustered to remove outliers, the elementary behavior can be post-processed to classify the overall object behavior as normal or abnormal. Techniques described in Section 7.3 can be used for this purpose. For instance, filtering can be applied if online behavior analysis is required or maximum-likelihood for batch analysis, *i.e.*, when the entire track is already available. Post-processing would increase the discrimination and thus improve the rate of correct detection while decreasing the rate of false detection.

The local modeling of densities with the Markov random field also presents specific characteristics. First, the local modeling is scalable. If each pixel of the camera plane is a site $\mathbf{s}$, the system requires a very large amount of memory preventing the implementation on embedded systems. This is clearly a limitation of the proposed

algorithm. To handle this problem, the arrangement of the 2D-lattice set of sites can be restricted to a subset of positions through scaling; the memory requirement will thus be reduced. On the other hand, the local property offers some advantages compared to the global model. For instance, the local model required less computation than its global counterpart because only the neighborhood $\eta_{\mathbf{s}}$ is updated in the MRF. The algorithm can also be implemented on a distributed system and, in particular, sensor networks, because the density of behavior is updated locally. In this case, the computation and storage requirements for each node are small. Distributed systems widen the field of applications for the local model. Not only can it be generalized to all sorts of tracking (*e.g.*, vehicles, people, objects, etc.) but it can also open prospects to new applications of Markov random fields such as abnormal event detection on networks and grid-based systems (*e.g.*, attacks, unusual power surge detection, etc.).

7.7 Summary of Abnormal Behavior Detection

Abnormal behavior detection has been in increasing demand in a broad range of fields, including vehicle traffic monitoring. Nevertheless, the problem remains open due to the inherent high level tasks and the difficulty of defining abnormal behaviors. A framework dividing the abnormal behavior into four main steps, namely feature selection, dimensionality reduction, feature vector density modeling and behavior classification, has been proposed. The algorithm introduced includes these four steps through a local modeling of abnormal behavior by a stochastic mixture model. The density of the feature vector is learnt locally via the implementation of a Gaussian Markov random field. The modes of the density are represented by cluster centers estimated by a stochastic learning algorithm.

The system was tested on a synthetic dataset modeling the trajectory of vehicles with occupants driving under the influence of alcohol. This experiment showed that the type of distance measure, the criterion for abnormal behavior classification and the neighboring variance play an important role in the performance of the algorithm. It has been inferred from the experiments that the right combination of these factors, namely the Mahalanobis distance in the clique potential and a variable neighboring

variance, provide the best results. The system has then been tested on traffic video sequences. The correct classification reaches 86.2%. Abnormal behavior has been tested to evaluate the suitability of the system to detect illegal crossings on highways. Finally, it has been suggested that the scope of the algorithm can be extended to a wider range of problems due to its high adaptability.

Chapter 8

Conclusions and Future Research

The thesis has been dedicated to the development of a contextual Bayesian inference for visual object tracking. The research conducted integrates the information pertaining to tracking in the Bayesian framework in order to improve the tracking robustness. The work makes use of broad assumptions on the nature of the video signal to set up the framework. The tracking was therefore inscribed in a Gaussian or near Gaussian noise environment with a video framerate allowing the capture of motion for analysis. The research has been limited to fixed cameras and small object size to make possible the implementation of site-based (*e.g.*, pixel-based) techniques such as background subtraction with mixture of Gaussians or Markov random fields. The project has involved the development of tracking algorithms for vehicle and pedestrian tracking in order to support the basis of our research. The aim of the thesis was to set a new path in the tracking chain, from the low level task of illumination-invariant background subtraction to tracking with integration of local context. Ultimately, abnormal behavior detection was performed in order to prove the efficiency of the techniques developed. This chapter presents conclusions on the research conducted in the thesis in Section 8.1 and proposes directions for future research in Section 8.2.

8.1 Thesis Summary

Low level tasks are crucial in the development of a tracking system since their reliability and quality impact the entire bottom-up chain. One of the most challenging problems encountered when segmenting objects with background subtraction is the robustness in varying illumination environments. Chapter 3 addresses this issue by providing a new update technique of the Gaussian mixture model. We first show the existence of *saturated pixels*, caused by a local variance degeneracy when abrupt changes in the background occur. Intuitively, the variance could be controlled by slowing down the update rate. Unfortunately, this yields a slower update of the model and, therefore, a decrease in illumination adaptation. The trade-off was resolved by using two separate update rates: a variable learning rate for the mean and a semi-constrained learning rate for the variance. The results show that the update of the model could be accelerated while the degeneracy of the variance is prevented. Indoor and outdoor changes in illumination are thus handled better than with existing techniques. Moreover, the foreground extraction for subsequent tasks is improved since the artefacts from illumination changes are well suppressed.

Traffic videos present specific characteristics due to the constrained nature of the environment such as slowly-varying vehicle speed, bounded trajectories and projection of the real-world scene on the camera plane. In Chapter 4, a projective Kalman filter is proposed, which integrates these characteristics through the projective transformation, the mean-shift algorithm and the foreground mask to provide fine and robust vehicle trajectory extraction. The projective Kalman filter was tested on an extensive traffic monitoring dataset including more than 2,600 vehicles. The results show that the technique achieves a tracking rate of 98% at 30 fps and 89% at 3 fps, whereas the extended Kalman filter reaches only 84% and 7%, respectively. In terms of computation, it was shown that the projective Kalman filter reduces the number of mean-shift iterations by 67%. The developed system therefore provides outstanding tracking performance on vehicle tracking.

The projective Kalman filter provides the optimal solution to vehicle feature estimation in Gaussian environment. This constraint could be relaxed with the use of

the particle filter, based on Monte Carlo simulations. However, the main issue with particle filters is the computation load since the accuracy is a function of the number of particles. In Chapter 5, we integrated the projective transformation into the importance density to improve the distribution of particles into the feature space. The technique was tested on the traffic monitoring dataset. The experimental results led to two conclusions. First, they show that the integration of the projective transformation into the PPF improves the tracking error compared the standard particle filter and that the number of particles necessary for a given tracking error is reduced. Second, we evaluate the projective particle filter in terms of tracking rate. For this purpose, the system developed in Chapter 4 was used with the PPF replacing the PKF. The results show an improvement in the performance compared to the standard particle filter.

Chapter 6 generalized the integration of contextual information with the implementation of Markov random fields. The constraint on the *prior* knowledge of the trajectory was relaxed to allow the learning of patterns in unknown environments. The local information was therefore learnt through a mixture of Gaussian Markov random fields. In turn, the patterns were used to distribute the particles in the feature space for tracking with the particle filter: the local distribution provided an accurate model of the importance density. The system was compared with the CONDENSATION algorithm on the traffic monitoring dataset and a pedestrian dataset. The results proved that the distribution of particles provided by the importance density is improved with the inference from the Markov random fields, leading to a reduction in tracking error. The adequate modeling of the importance density also led to robust recovery of prolonged spatio-temporal occlusions, where the CONDENSATION algorithm fails.

The mixture of Gaussian Markov random fields introduced in Chapter 6 was adapted and trained to detect abnormal behavior. The technique relied solely on the position and the displacement of the object, two features directly accessible from the track extraction. Contrary to traditional techniques, the displacements were modeled locally, providing fast and efficient estimate of the distribution for each position. A stochastic clustering algorithm was adapted to train the Markov random fields.

The experiments were conducted on synthetic and real data containing abnormal behavior. The detection of abnormal behavior was found to be maximum when the clique potential is modeled with the Mahalanobis distance and when the detection is based on clusters. The proposed technique was compared with its global counterpart and with a SOM. The experiments showed that a higher accuracy in the detection of abnormal behavior is achieved with the proposed method. On traffic monitoring dataset, results showed that abnormal behavior, represented by people walking, running or cycling on a highway, was detected with 86% accuracy for a 10% false detection rate.

The thesis explored the different steps of object tracking and abnormal behavior detection, offering improvements in terms of accuracy and robustness. Throughout the bottom-up chain, the development of new techniques contributed to the particular area of research and to the overall improvement of abnormal behavior detection.

8.2 Suggestions for Improvements and Future Research

The thesis addressed several issues pertaining to visual object tracking which were summarized in the previous section. However, new topics of research can be defined with the work proposed herein serving as starting point. The use of Markov random fields for behavior modeling and tracking improvement with contextual inference is a breakthrough in the proposed form. Although Markov random fields are extensively used for image processing (*e.g.*, segmentation, noise reduction, image restoration, etc.), they have seldom been used for activity modeling. Markov random fields provide contextual information for both tracking and abnormal behavior detection. We investigated local stationary inference in this thesis, based on a parametric model. The rest of this section outlines the research that can be conducted to further improve the robustness of tracking and behavior modeling with contextual information.

First, the learning of contextual information in this thesis was limited to the displacement of objects (position and speed). The approach was suitable for recovery

after occlusion and detection of abnormal behavior which required the modeling of trajectories. With the development of powerful architectures, the feature vector can include other characteristics of the object such as color and size. The interest of increasing the dimension of the feature vector is to provide more complex decisions such as the detection of small vehicles in bus lanes, etc. However, the complexity of the algorithm would increase.

Second, a theoretical analysis of non-stationary inferences should be conducted. Indeed, the underlying local densities can evolve through time, leading to problems concerning transient phase, statistical convergence, etc. Markov random fields are suitable for the modeling of non-stationary inferences. Although the transient state of the random fields is out of the scope of the thesis, tracking and abnormal behavior detection in crowds require an accurate modeling of the distributions for this phase. Indeed, in the presence of dense flow of objects, the realizations cannot be considered sparse and independent anymore. The non-stationary case occurs in a large number of situations. For instance, behaviors can be normal or abnormal depending on the time in the day; and local changes in the scene such as roadworks can induce a temporary modification of the underlying distribution.

Finally, and most importantly, we believe that this thesis has opened an area of research in Bayesian inference with contextual information, where the MRFs are adapted to model local information. The representation can be completed with a global modeling of event detection. Information fusion could then be used to draw more complex decisions with low computation cost. A concrete example is the integration of traffic context with global information: traffic congestion, based on the average speed, could justify small vehicle displacements. Even more elaborated, the detection of traffic light color brings a global context to the monitoring of an intersection. With the current system, pedestrians crossing a road are either considered as a normal or an abnormal event, at all time. Contextual information could discriminate between green lights for vehicles, that is, when pedestrian crossing is abnormal, and red lights, when it is normal. The integration of a global model would enable the weighting of different fields in the mixture. Ultimately, a multi-resolution collection of Markov random fields can provide multi-scale inference.

The two major fields suggested for future research are non-stationary and multi-scale inferences. However, another field of investigation, pertaining to practical implementation is the computation load. The accuracy of tracking systems is always limited by the complexity of the implementation. One of the advantages of a local approach which has not been explored in this work is the use of distributed architectures to manage the entire system. The possibility of dispatching the computation load on different nodes in a mesh is eased by the structure of the existing Markov random field. For example, neighboring sites can be clustered and managed by one single node in the mesh of computers, communicating with others only when local update is necessary. A distributed Markov random field can thus be designed.

Bibliography

[1] *The Macquarie Dictionary*. 2005.

[2] S. Ali and M. Shah. An integrated approach for generic object detection using kernel pca and boosting. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2005.

[3] S. Ali and M. Shah. A supervised learning framework for generic object detection in images. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1347–1354, 2005.

[4] J. Allen, R. Xu, and J. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7, 2004.

[5] I. P. Alonso, D. F. Llorca, M. A. Sotelo, L. M. Bergasa, P. R. De Toro, J. Nuevo, M. Ocana, and M. A. G. Garrido. Combination of feature extraction methods for svm pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):292–307, 2007.

[6] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 175–178, 2006.

[7] D. Angelova and L. Mihaylova. Extended object tracking using monte carlo methods. *IEEE Transactions on Signal Processing*, 56(2):825–832, 2008.

[8] L. Angrisani, M. D'Apuzzo, and R. S. L. Moriello. Unscented transform: a powerful tool for measurement uncertainty evaluation. *IEEE Transactions on Instrumentation and Measurement*, 55(3):737–743, 2006.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[10] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

[11] R. Azriel and L. P. John. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 13(4):471–494, 1966.

[12] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. In *Readings in computer vision: issues, problems, principles, and paradigms*, pages 714–725. 1987.

[13] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking*. 1995.

[14] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[15] A. Bartesaghi and G. Sapiro. Tracking of moving objects under severe and total occlusions. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 301–304, 2005.

[16] E. Bas, M. Tekalp, and F. S. Salman. Automatic vehicle counting from video for traffic flow analysis. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 392–397, 2007.

[17] R. Bastos and J. M. S. Dias. Fully automated texture tracking based on natural features extraction and template matching. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 180–183, 2005.

[18] R. Ben-Ari and N. Sochen. Variational stereo vision with sharp discontinuities and occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–7, 2007.

[19] S. Benk. Psycho-visual model of the human optical system as base for lossy-lossless compression of visual information. In *Proceedings of the International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service*, volume 2, pages 430–433, 2001.

[20] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1158–1163, 2005.

[21] H. Bischof, H. Wildenauer, and A. Leonardis. Illumination insensitive eigenspaces. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 233–238, 2001.

[22] C. M. Bishop. *Neural networks for pattern recognition*. 1995.

[23] W. D. Blair. Design of nearly constant velocity track filters for tracking maneuvering targets. In *Proceedings of the International Conference on Information Fusion*, pages 1–7, 2008.

[24] F. Boussaid, A. Bouzerdoum, and D. Chai. Vlsi implementation of a skin detector based on a neural network. In *Proceedings of the International Conference on Information, Communications and Signal Processing*, pages 1605–1608, 2005.

[25] P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi. Abnormal behavior detection using a multi-modal stochastic learning approach. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 121–126, 2008.

[26] P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi. Vehicle tracking by non-drifting mean-shift using projective kalman filter. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 61–66, 2008.

[27] A. Bouzerdoum. The elementary movement detection mechanism in insect vision. *Journal Information for Philosophical Transactions: Biological Sciences*, 339:375–384, 1993.

[28] A. Bouzerdoum. A stochastic competitive learning algorithm. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pages 908–913, 2001.

[29] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. *Computer Vision and Image Understanding*, 84(1):77–103, 2001.

[30] R. Bradski. Computer vision face tracking for use in a perceptual user interface. Technical report, Intel Corporation, 1998.

[31] H. Breit and G. Rigoll. A flexible multimodal object tracking system. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 133–136, 2003.

[32] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. 2009.

[33] A. Buchanan and A. Fitzgibbon. Combining local and global motion models for feature point tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[34] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[35] S. Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 141–146, 2007.

[36] S. Calderara, R. Cucchiara, and A. Prati. A distributed outdoor video surveillance system for detection of abnormal people trajectories. In *Proceedings of the International Conference on Distributed Smart Cameras*, pages 364–371, 2007.

[37] F. M. Candocia. Simultaneous homographic and comparametric alignment of multiple exposure-adjusted pictures of the same scene. *IEEE Transactions on Image Processing*, 12(12):1485–1494, 2003.

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

[39] B. P. Carlin and T. A. Louis. *Bayes and empirical Bayes methods for data analysis*. 1996.

[40] B. S. Carlson. Comparison of modern ccd and cmos image sensor technologies and systems for low resolution imaging. In *Proceedings of the IEEE Conference on Sensors*, volume 1, pages 171–176, 2002.

[41] B. Carswell and V. Chandran. Automated recognition of drunk driving on highways from video sequences. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 306–310, 1994.

[42] C. Chang and R. Ansari. Kernel particle filter for visual tracking. *IEEE Signal Processing Letters*, 12(3):242–245, 2005.

[43] C. Chang, R. Ansari, and A. Khokhar. Multiple object tracking with kernel particle filter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 566–573, 2005.

[44] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi. Heterogeneous fusion of omnidirectional and ptz cameras for multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1052–1063, 2008.

[45] L. Chen, J. Zhou, Z. Liu, W. Chen, and G. Xiong. A skin detector based on neural network. In *Proceedings of the IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions*, volume 1, pages 615–619, 2002.

[46] Y. Chen, P. Gross, V. Ramakrishna, H. Rabitz, and K. Mease. Competitive tracking of molecular objectives described by quantum mechanics. *The Journal of Chemical Physics*, 102(20):8001–8010, 1995.

[47] Y. Chen, G. Liang, K. K. Lee, and Y. Xu. Abnormal behavior detection by multi-svm-based bayesian network. In *Proceedings of the International Conference on Information Acquisition*, pages 298–303, 2007.

[48] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[49] H. Choi and R. G. Baraniuk. Multiscale image segmentation using wavelet-domain hidden markov models. *IEEE Transactions on Image Processing*, 10(9):1309–1321, 2001.

[50] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang. Multiple vehicles detection and tracking based on scale-invariant feature transform. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 528–533, 2007.

[51] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 150–155, 2002.

[52] R. T. Collins. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

[53] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

[54] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[55] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 438–445, 2001.

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

[57] D. Cremers, T. Kohlberger, and C. Schnorr. Nonlinear shape statistics in mumford-shah based segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 93–108, 2002.

[58] F. Crow. Summed-area tables for texture mapping. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1984.

[59] P. Cui, L.-F. Sun, Z.-Q. Liu, and S.-Q. Yang. A sequential monte carlo approach to anomaly detection in tracking visual events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[60] D. Culibrk, O. Marques, D. Socek, H. A. Kalva, and B. A. Furht. Neural network approach to background modeling for video object segmentation. *IEEE Transactions on Neural Networks*, 18(6):1614–1627, 2007.

[61] J. Czyz. Object detection in video via particle filters. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 1, pages 820–823, 2006.

[62] M. Dahmane and J. Meunier. Real-time video surveillance with self-organizing maps. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 136–143, 2005.

[63] K.-X. Dai, G.-H. Li, and Y.-L. Gan. A probabilistic model for surveillance video mining. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, pages 1144–1148, 2006.

[64] D. Datcu and L. J. M. Rothkrantz. Automatic bi-modal emotion recognition system based on fusion of facial expressions and emotion extraction from speech. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–2, 2008.

[65] F. Dellaert, D. Pomerlau, and C. Thorpe. Model-based car tracking integrated with a road-follower. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1889–1894, 1998.

[66] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society - Series B*, 39(1):1–38, 1977.

[67] B. Deutsch, H. Niemann, and J. Denzler. Multi-step active object tracking with entropy based optimal actions using the sequential kalman filter. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 105–108, 2005.

[68] M. Dewan and G. D. Hager. Toward optimal kernel-based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 618–625, 2006.

[69] J. R. Diebel, S. Thrun, and M. Brunig. A bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25(1):39–59, 2006.

[70] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Universit of Cambridge, 1998.

[71] R. O. Duda. *Pattern classification*. 2001.

[72] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[73] T. V. Duong, H. H. Bui, D. Q. Phung, and S. A. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 838–845, 2005.

[74] D. Duque, H. Santos, and P. Cortez. Prediction of abnormal behaviors for intelligent video surveillance systems. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pages 362–367, 2007.

[75] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of the European Conference on Computer Vision*, pages 751–767, 2000.

[76] R. Elias. Wide baseline matching through homographic transformation. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 4, pages 130–133, 2004.

[77] Z. Fan, M. Yang, and Y. Wu. Multiple collaborative kernel tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1268–1273, 2007.

[78] Z. Fan, M. Yang, Y. Wu, G. Hua, and T. Yu. Efficient optimal kernel placement for reliable visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 658–665, 2006.

[79] W. Fang, L. Qing, L. Lin, X. Ying-Qing, and S. Heung-Yeung. Color sketch generation. In *Proceedings of the international symposium on Non-photorealistic animation and rendering*, pages 47–54, 2006.

[80] R. Feghali and A. Mitiche. Spatiotemporal motion boundary detection and motion boundary velocity estimation for tracking moving objects with a moving camera: a level sets pdes approach with concurrent camera motion compensation. *IEEE Transactions on Image Processing*, 13(11):1473–1490, 2004.

[81] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, 1995.

[82] C. Fu, C. Chun-Jen, and L. Chi-Jen. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.

[83] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[84] D. Gamerman. *Markov chain Monte Carlo : stochastic simulation for Bayesian inference*. 2006.

[85] G. Garibotto and C. Cibei. 3d scene analysis by real-time stereovision. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 105–108, 2005.

[86] G. Garibotto, M. Corvi, C. Cibei, and S. Sciarrino. 3dmods: 3d moving obstacle detection system. In *Proceedings of the IEEE International Conference on Image Analysis and Processing*, pages 618–623, 2003.

[87] C. Gentile, O. Camps, and M. Sznaier. Segmentation for robust tracking in the presence of severe occlusion. *IEEE Transactions on Image Processing*, 13(2):166–178, 2004.

[88] I. Giakoumis, N. Nikolaidis, and I. Pitas. Digital image processing techniques for the detection and removal of cracks in digitized paintings. *IEEE Transactions on Image Processing*, 15(1):178–188, 2006.

[89] B. Gloyer, H. K. Aghajan, K.-Y. Siu, and T. Kailath. Vehicle detection and tracking for freeway traffic monitoring. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 970–974, 1994.

[90] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[91] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, 2002.

[92] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[93] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 790–797, 2004.

[94] B. Han, D. Comaniciu, Z. Ying, and L. Davis. Incremental density approximation and kernel-based bayesian filtering for object tracking. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 638–644, 2004.

[95] B. Han and L. Davis. On-line density-based appearance modeling for object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1492–1499, 2005.

[96] B. Han, Z. Ying, D. Comaniciu, and L. Davis. Kernel-based bayesian filtering for object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 227–234, 2005.

[97] Z. Hao and Q. Lei. Vision-based interface: Using face and eye blinking tracking with camera. In *Proceedings of the International Symposium on Intelligent Information Technology Application*, volume 1, pages 306–310, 2008.

[98] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. 1992.

[99] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[100] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[101] A. Haselhoff and A. Kummert. A vehicle detection system based on haar and triangle features. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 261–266, 2009.

[102] H. T. Ho and R. Goecke. Optical flow estimation using fourier mellin transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[103] S. A. Hojjatoleslami and J. Kittler. Region growing: a new approach. *IEEE Transactions on Image Processing*, 7(7):1079–1084, 1998.

[104] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1988.

[105] K. P. Horn, B. and G. Schunck, B. Determining optical flow. *Artifical Intelligience*, 17:185–203, 1981.

[106] W. Hu, T. Tieniu, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):334–352, 2004.

[107] S. Huang. *Neural network control : theory and applications*. 2004.

[108] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1051–1058, 2005.

[109] S. S. Intille, J. W. Davis, and A. F. Bobick. Real-time closed-world tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 697–703, 1997.

[110] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[111] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 34–41, 2001.

[112] M. A. Isard. *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*. PhD thesis, University of Oxford, 1998.

[113] K. Jack. *Digital video and DSP : instant access*. 2008.

[114] R. C. Jain and H. H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):206–213, 1979.

[115] S. Jayaram, S. Schmugge, M. C. Shin, and L. V. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 813–818, 2004.

[116] C. Jian, D. M. Dawson, W. E. Dixon, and A. Behal. Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. *IEEE Transactions on Control Systems Technology*, 13(5):814–825, 2005.

[117] F. Jiang, Y. Wu, and A. K. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *Proceedings of the IEEE International Conference on Image Processing*, volume 5, pages 145–148, 2007.

[118] L. Jianguang, T. Tieniu, H. Weiming, Y. Hao, and S. J. Maybank. 3-d model-based vehicle tracking. *IEEE Transactions on Image Processing*, 14(10):1561–1569, 2005.

[119] M. Jogan and A. Leonardis. Robust localization using eigenspace of spinning-images. In *Proceedings of the IEEE Workshop on Omnidirectional Vision*, pages 37–44, 2000.

[120] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[121] N. Johnson and D. Hogg. Representation and synthesis of behaviour using gaussian mixtures. *Image and Vision Computing*, 20(12):889–894, 2002.

[122] M. J. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–4, 2008.

[123] S.-W. Joo and R. Chellappa. Attribute grammar-based event recognition and anomaly detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, pages 107–107, 2006.

[124] N. Joshi, S. Avidan, W. Matusik, and D. J. Kriegman. Synthetic aperture tracking: Tracking through occlusions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[125] W. Jue, T. Bo, X. Yingqing, and C. Michael. Image and video segmentation by anisotropic kernel mean shift. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 238–249, 2004.

[126] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[127] C. R. Jung. Multiscale image segmentation using wavelets and watersheds. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 278–284, 2003.

[128] Y.-K. Jung and Y.-S. Ho. Traffic parameter extraction using video-based vehicle tracking. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 764–769, 1999.

[129] F. Jurie and M. Dhome. A simple and efficient template matching algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 544–549, 2001.

[130] R. E. Kalman. A new approach to linear filtering and prediction problems. *Basic Engineering*, 82:34–45, 1960.

[131] H. Kamel and W. Badawy. A real-time multiple target tracking algorithm using merged probabilistic data association technique and smoothing particle filter. In *Proceedings of the IEEE Conference on Radar*, pages 218–223, 2006.

[132] S. Kamran and O. Haas. A multilevel traffic incidents detection approach: Identifying traffic patterns and vehicle behaviours using real-time gps data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 912–917, 2007.

[133] N. K. Kanhere and S. T. Birchfield. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):148–160, 2008.

[134] C. S. Kenney, B. S. Manjunath, M. Zuliani, G. A. Hewer, and A. Van Nevel. A condition number for point matching with application to registration and

postregistration error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1437–1454, 2003.

[135] J. Klein, C. Lecomte, and P. Miche. Fast color-texture discrimination: Application to car tracking. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 546–551, 2007.

[136] D. Koller, J. Weber, and J. Malik. Towards realtime visual based tracking in cluttered traffic scenes. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 201–206, 1994.

[137] A. Kong, J. S. Lui, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Assocation*, 89(425):278–288, 1994.

[138] M. Korhonen, J. Heikkila, and O. Silvcn. Intensity independent color models and visual tracking. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 600–604, 2000.

[139] J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2612, 2003.

[140] D. Kragic and H. I. Christensen. Tracking techniques for visual servoing tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1663–1669, 2000.

[141] I. Kukenys and B. McCane. Classifier cascades for support vector machines. In *Proceedings of the International Conference Image and Vision Computing New Zealand*, pages 1–6, 2008.

[142] C. Kwok, D. Fox, and M. Meila. Adaptive real-time particle filters for robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 2836–2841, 2003.

[143] C. Kwok, D. Fox, and M. Meila. Real-time particle filters. *Proceedings of IEEE*, 92(3):469–484, 2004.

[144] V. Kyrki and D. Kragic. Integration of model-based and model-free cues for visual object tracking in 3d. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1554–1560, 2005.

[145] L. J. Latecki and R. Miezianko. Object tracking with dynamic template update and occlusion detection. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 1, pages 556–560, 2006.

[146] S. M. LaValle and S. A. Hutchinson. A bayesian segmentation methodology for parametric image models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):211–217, 1995.

[147] C.-K. Lee, M.-F. Ho, W.-S. Wen, and C.-L. Huang. Abnormal event detection in video using n-cut clustering. In *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 407–410, 2006.

[148] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, 2005.

[149] X. Lei, Z. Guangxi, T. Miao, X. Haixiang, and Z. Zhenming. Vehicles tracking based on corner feature in video-based its. In *Proceedings of the IEEE International Conference on ITS Telecommunications*, pages 163–166, 2006.

[150] J. Li and S. Chin-Chua. Transductive inference for color-based particle filter tracking. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 949–52, 2003.

[151] X.-X. Li, J.-B. Zheng, Y.-N. Zhang, and H.-J. Yuan. Activity analysis based on som. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, volume 7, pages 3975–3979, 2007.

[152] Z. Liang and C. E. Thorpe. Stereo- and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):148–154, 2000.

[153] L. Liang-qun, J. Hong-bing, and L. Jun-hui. The iterated extended kalman particle filter. In *Proceedings of the IEEE International Symposium on Communications and Information Technology*, volume 2, pages 1213–1216, 2005.

[154] C.-P. Lin, J.-C. Tai, and K.-T. Song. Traffic monitoring based on real-time image tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 2091–2096, 2003.

[155] X. Lin, Y. Liu, and W. Dai. Study of occlusions problem in stereo vision. In *Proceedings of the World Congress on Intelligent Control and Automation*, pages 5062–5067, 2008.

[156] M. Liu, C. Wu, and Y. Zhang. Multi-resolution optical flow tracking algorithm based on multi-scale harris corner points feature. In *Proceedings of the Chinese Conference on Control and Decision*, pages 5287–5291, 2008.

[157] Y. C. Liu and C. H. Ho. The effects of different breath alcohol concentration and post alcohol upon driver's driving performance. In *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, pages 505–509, 2007.

[158] Z. Liu, K. Huang, T. Tan, and L. Wang. Cast shadow removal with gmm for surface reflectance component. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 1, pages 727–730, 2006.

[159] B. D. Lucas. *Generalized image matching by the method of differences*. PhD thesis, Carnegie Mellon University, 1984.

[160] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):541–547, 2008.

[161] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20:895–903, 2002.

[162] L. Malagon-Borja and O. Fuentes. An object detection system using image reconstruction with pca. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 2–8, 2005.

[163] S. Mann and R. W. Picard. Video orbits of the projective group a simple approach to featureless estimation of parameters. *IEEE Transactions on Image Processing*, 6(9):1281–1295, 1997.

[164] G.-Z. Mao, Y.-L. Wu, M.-K. Hor, and C.-Y. Tang. Real-time hand detection and tracking against complex background. In *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 905–908, 2009.

[165] N. Martel-Brisson and A. Zaccarin. Moving cast shadow detection from a gaussian mixture shadow model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 643–648, 2005.

[166] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multidistribution approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133–1146, 2007.

[167] N. Martel-Brisson and A. Zaccarin. Unsupervised approach for building non-parametric background and foreground models of scenes with significant foreground activity. In *Proceedings of the workshop on Vision networks for behavior analysis*, 2008.

[168] R. Martins, P. Pina, J. S. Marques, and M. Silveira. Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127–131, 2009.

[169] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. 1996.

[170] D. L. McLeish. *Monte Carlo simulation and finance*. 2005.

[171] E. B. Meier and F. Ade. Tracking cars in range images using the condensation algorithm. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 129–134, 1999.

[172] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor. Detection and classification of highway lanes using vehicle motion trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):188–200, 2006.

[173] W. P. Menzel. Cloud tracking with satellite imagery: From the pioneering work of ted fujita to the present. *Bulletin of the American Meteorological Society*, 82(1):33–48, 2001.

[174] O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, 16(11):2787–2801, 2007.

[175] L. Mingxiang and J. Yunde. Trinocular cooperative stereo vision and occlusion detection. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 1129–1133, 2006.

[176] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 302 – 309, 2004.

[177] H. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1977.

[178] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatio-temporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, 1998.

[179] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[180] C. Musso, N. Oudjane, and F. LeGland. Improving regularised particle filters. In *Sequential Monte Carlo Methods in Practice*. 2001.

[181] H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, Image Processing*, 21(1):85–117, 1983.

[182] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough. A joint particle filter for audio-visual speaker tracking. In *Proceedings of the International Conference on Multimodal Interfaces*, pages 61–68, 2005.

[183] K. Nickels and S. Hutchinson. Weighting observations: the use of kinematic models in object tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1677–1682, 1998.

[184] M. Niethammer and A. Tannenbaum. Dynamic geodesic snakes for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 660–667, 2004.

[185] L. M. Novak. Optimal target t designation techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 17(5):676–684, 1981.

[186] A. Nowakowski and W. Skarbek. Lens radial distortion calibration using homography of central points. In *Proceedings of the IEEE International Conference on "Computer as a Tool"*, pages 340–343, 2007.

[187] F. Oberti, S. Calcagno, M. Zara, and C. S. Regazzoni. Robust tracking of humans and vehicles in cluttered scenes with occlusions. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 629–632, 2002.

[188] L. Oisel, E. Memin, L. Morin, and F. Galpin. One-dimensional dense disparity estimation for three-dimensional reconstruction. *IEEE Transactions on Image Processing*, 12(9):1107–1119, 2003.

[189] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[190] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[191] N. Paragios, O. Mellina-Gottardo, and V. Ramesh. Gradient vector flow fast geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):402–407, 2004.

[192] V. Parameswaran, V. Ramesh, and I. Zoghlami. Tunable kernels for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2179–2186, 2006.

[193] J. Park, A. Tabb, and A. C. Kak. Hierarchical data structure for real-time background subtraction. In *Proceedings of the International Conference on Image processing*, pages 1849–1852, 2006.

[194] S. Park and M. M. Trivedi. Homography-based analysis of people and vehicle activities in crowded scenes. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 51–51, 2007.

[195] S. L. Phung and A. Bouzerdoum. Detecting people in images: An edge density approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 1229–1232, 2007.

[196] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005.

[197] M. Pieper and A. Kummert. Image prediction for virtual environments by means of kalman filter based 3d object tracking. In *Proceedings of the International Workshop on Multidimensional Systems*, pages 30–35, 2005.

[198] Y. Pingkun, S. M. Khan, and M. Shah. 3d model based object class detection in an arbitrary view. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–6, 2007.

[199] J. C. Principe. *Neural and adaptive systems : fundamentals through simulations*. 2000.

[200] Z. Qiu, D. An, D. Yao, D. Zhou, and B. Ran. An adaptive kalman predictor applied to tracking vehicles in the traffic monitoring system. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 230–235, 2005.

[201] W. Qu, D. Schonfeld, and M. Mohamed. Decentralized multiple camera multiple object tracking. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 245–248, 2006.

[202] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 467–474, 2003.

[203] K. Rangarajan and M. Shah. Establishing motion correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–108, 1991.

[204] S. Rao and P. S. Sastry. Abnormal activity detection in video sequences using learnt probability densities. In *Proceedings of the Conference on Convergent Technologies for Asia-Pacific Region*, volume 1, pages 369–372, 2003.

[205] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):560–576, 2001.

[206] Y. Rathi, N. Vaswani, and A. Tannenbaum. A generic framework for tracking using particle filter with dynamic shape prior. *IEEE Transactions on Image Processing*, 16(5):1370–1382, 2007.

[207] N. Ray and S. T. Acton. Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, 23(12):1466–1478, 2004.

[208] H. Rehreuer, K. Seidel, and M. Datcu. Multiscale image segmentation with a dynamic label tree. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, volume 4, pages 1772–1774, 1998.

[209] H. L. Ribeiro and A. Gonzaga. Hand image segmentation in video sequence by gmm: a comparative analysis. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 357–364, 2006.

[210] N. Richard, B. Bringier, and E. Rollo. Integration of human perception for color texture management. In *Proceedings of the International Symposium on Signals, Circuits and Systems*, volume 1, pages 207–210, 2005.

[211] Y. Ricquebourg and P. Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):797–808, 2000.

[212] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. 2004.

[213] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.

[214] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[215] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[216] V. Salari and I. K. Sethi. Feature point correspondence in the presence of occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):87–91, 1990.

[217] R. Santiago-Mozos, J. M. Leiva-Murillo, F. Perez-Cruz, and A. Artes-Rodriguez. Supervised-pca and svm classifiers for object detection in infrared images. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 122–127, 2003.

[218] S. Savarese and F.-F. Li. 3d generic object categorization, localization and pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[219] I. J. Schoenberg. Contribution to the problem of approximation of equidistant data by analytic function. *Quarterly of Applied Mathematics*, 4(45-99), 1946.

[220] D. Schreiber, B. Alefs, and M. Clabian. Single camera lane detection and tracking. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 302–307, 2005.

[221] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.

[222] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 110–115, 2003.

[223] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[224] C. Shen, J. Brooks, M., and A. v. d. Hengel. Fast global kernel density mode seeking: Applications to localization and tracking. *IEEE Transactions on Image Processing*, 16(5):1457–1469, 2007.

[225] C. Shen, M. J. Brooks, and A. van den Hengel. Augmented particle filtering for efficient visual tracking. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 856–859, 2005.

[226] C. Shen, M. J. Brooks, and A. van den Hengel. Fast global kernel density mode seeking with application to localization and tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1516–1523, 2005.

[227] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[228] C. Shu-Ching, S. Mei-Ling, S. Peeta, and Z. Chengcui. Learning-based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems. *IEEE Transactions on Intelligent Transportation Systems*, 4(3):154–167, 2003.

[229] F. Soares and F. Muge. Watershed lines suppression by waterfall marker improvement and line-neighbourhood analysis. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 604–607, 2004.

[230] M. Soga, S. Hiratsuka, H. Fukamachi, and Y. Ninomiya. Pedestrian detection for a near infrared imaging system. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 1167–1172, 2008.

[231] G. E. J. Sotak and K. L. Boyer. The laplacian-of-gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics, and Image Processing*, 48(2):147–189, 1989.

[232] S. Spors, R. Rabenstein, and N. Strobel. Joint audio-video object tracking. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 393–396, 2001.

[233] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[234] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[235] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, 2006.

[236] K. B. Sun and B. J. Super. Classification of contour shapes using class segment sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 727–733, 2005.

[237] K. Suzuki, I. Horiba, and N. Sugie. Fast connected-component labeling based on sequential local operations in the course of forward raster scan followed by backward raster scan. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 2, pages 434–437, 2000.

[238] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Image Processing*, 6(5):642–655, 1997.

[239] Z. Tang and Z. Miao. Fast background subtraction and shadow elimination using improved gaussian mixture model. In *Proceedings of the IEEE International Workshop on Haptic, Audio and Visual Environments and Games*, pages 38–41, 2007.

[240] J. Tao, M. Turjo, and Y.-P. Tan. Quickest change detection for health-care video surveillance. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 505–508, 2006.

[241] Y. Tao, P. Quan, L. Jing, and S. Z. Li. Real-time multiple objects tracking with occlusion handling in dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 970–975, 2005.

[242] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.

[243] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*, 9(4):735–744, 2000.

[244] G. Unal, H. Krim, and A. Yezzi. Fast incorporation of optical flow into active polygons. *IEEE Transactions on Image Processing*, 14(6):745–759, 2005.

[245] M. Unser. Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999.

[246] P. Vadakkepat and L. Jing. Improved particle filter in sensor fusion for tracking randomly moving object. *IEEE Transactions on Instrumentation and Measurement*, 55(5):1823–1832, 2006.

[247] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. The unscented particle filter. Technical report, Cambridge University Engineering Department, August 16, 2000.

[248] D. Van der Weken, M. Nachtegael, and E. Kerre. Using similarity measures for histogram comparison. In *Fuzzy Sets and Systems*, pages 1–9. 2003.

[249] N. Vaswani and R. Chellappa. Non-stationary "shape activities". In *Proceedings of the IEEE Conference on Decision and Control*, pages 1521–1528, 2005.

[250] N. Vaswani and R. Chellappa. Principal components null space analysis for image and video classification. *IEEE Transactions on Image Processing*, 15(7):1816–1830, 2006.

[251] N. Vaswani, A. R. Chowdhury, and R. Chellappa. Statistical shape theory for activity modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 493–496, 2003.

[252] N. Vaswani, A. K. Roy-Chowdhury, and R. Chellappa. "shape activity": a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection. *IEEE Transactions on Image Processing*, 14(10):1603–1616, 2005.

[253] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[254] S. Velipasalar and W. Wolf. Multiple object tracking and occlusion handling by information exchange between uncalibrated cameras. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 418–421, 2005.

[255] K. L. Vincken, A. S. E. Koster, and M. A. Viergever. Probabilistic multiscale image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):109–120, 1997.

[256] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[257] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 734–741, 2003.

[258] N. V. Vladimir. *The nature of statistical learning theory*. 1995.

[259] B. N. Vo and W. K. Ma. The gaussian mixture probability hypothesis density filter. *Transactions on Signal Processing*, 54(11):4091–4104, 2006.

[260] C. K. Wan, B. Z. Yuan, and Z. J. Miao. A new algorithm for static camera foreground segmentation via active coutours and gmm. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[261] Y. Wang, K. Huang, and T. Tan. Abnormal activity recognition in office based on r transform. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 341–344, 2007.

[262] D. M. Weber and D. P. Casasent. Quadratic gabor filters for object detection. *IEEE Transactions on Image Processing*, 10(2):218–230, 2001.

[263] M. Weser, D. Westhoff, M. Huser, and Z. Jianwei. Multimodal people tracking and trajectory prediction based on learned generalized motion patterns. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 541–546, 2006.

[264] P. Wilmott. *Paul Wilmott introduces quantitative finance*. 2007.

[265] M. W. Woolrich and T. E. Behrens. Variational bayes inference of spatial mixture models for segmentation. *IEEE Transactions on Medical Imaging*, 25(10):1380–1391, 2006.

[266] N. Woonhyun and H. Joonhee. Motion-based background modeling for foreground segmentation. In *Proceedings of theinternational workshop on Video surveillance and sensor networks*, 2006.

[267] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[268] X. Wu, Y. Ou, H. Qian, and Y. Xu. A detection system for human abnormal behavior. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1204–1208, 2005.

[269] X. Wu, Y. Wang, and X. Zheng. Monocular video foreground segmentation system. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[270] Y.-J. XiaHou and S.-R. Gong. Adaptive shadows detection algorithm based on gaussian mixture model. In *Proceedings of the International Symposium on Information Science and Engieering*, volume 1, pages 116–120, 2008.

[271] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):893–908, 2008.

[272] L. Xingzhi and S. M. Bhandarkar. Multiple object tracking using elastic matching. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 123–128, 2005.

[273] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[274] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 212–219, 2005.

[275] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[276] A. Yilmaz, L. Xin, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

[277] J. Yin, Q. Yang, and J. J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, 2008.

[278] Z. Ying and S. Schwartz. Efficient face detection with multiscale sequential classification. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 121–124, 2002.

[279] T. Ying-Li, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1182–1187, 2005.

[280] H. Ying-Tung, C. Cheng-Long, J. Joe-Air, and C. Cheng-Chih. A contour based image segmentation algorithm using morphological edge detection. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2962–2967, 2005.

[281] S. Yong, Y. Fan, W. Runsheng, and Z. Feng. A tracking model with occlusion handling based on information fusion. In *Proceedings of the IEEE International Conference on Computer-Aided Design and Computer Graphics*, pages 517–520, 2007.

[282] P. Yuxin, J. Yuxin, H. Kezhong, S. Fuchun, L. Huaping, and T. Linmi. Color model based real-time face detection with adaboost in color image. In *Proceedings of the International Conference on Machine Vision*, pages 40–45, 2007.

[283] Q. Zang and R. Klette. Robust background subtraction and maintenance. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 90–93, 2004.

[284] S. Zehang, G. Bebis, and R. Miller. Monocular precrash vehicle detection: features and classifiers. *IEEE Transactions on Image Processing*, 15(7):2019–2034, 2006.

[285] H.-C. Zeng and S.-H. Lai. Adaptive foreground object extraction for real-time video surveillance with lighting variations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 1201 – 1204, 2007.

[286] Z. Zeng and S. Ma. Head tracking by active particle filtering. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pages 82–87, 2002.

[287] B. Zhang and Y. F. Li. A method for calibrating the central catadioptric camera via homographic matrix. In *Proceedings of the IEEE International Conference on Information and Automation*, pages 972–977, 2008.

[288] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 611–618, 2005.

[289] L. Zhao and C. Thorpe. Qualitative and quantitative car tracking from a range image sequence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 496–501, 1998.

[290] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1208–1221, 2004.

[291] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 819–826, 2004.

[292] Z. Zhong, W. Ye, S. Wang, M. A. Yang, and Y. A. Xu. Crowd energy and feature analysis. In *Proceedings of the IEEE International Conference on Integration Technology*, pages 144–150, 2007.

[293] D. Zhou and H. Zhang. Modified gmm background modeling and optical flow for detection of moving objects. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2224–2229, 2005.

[294] J. Zhou, D. Gao, and D. Zhang. Moving vehicle detection for automatic traffic monitoring. *IEEE Transactions on Vehicular Technology*, 56(1):51–59, 2007.

[295] L.-J. Zhu, J.-N. Hwang, and H.-Y. Cheng. Tracking of multiple objects across multiple cameras with overlapping and non-overlapping views. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 1056–1060, 2009.

[296] A. Ziadi and G. Salut. Non-overlapping deterministic gaussian particles in maximum likelihood non-linear filtering: phase tracking application. In *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems*, pages 645–648, 2005.

[297] D. Ziou and S. Tabbone. Edge detection techniques - an overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559, 1998.

[298] M. Zobel, J. Denzler, and H. Niemann. Entropy based camera control for visual object tracking. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 901–904, 2002.