Multiple object tracking by stochastic method

Li, Jiang

2006

Multiple Object Tracking by Stochastic Method

Li Jiang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

To my family,
for their encouragement and love.

Acknowledgements

Above all, I would like to express my sincere thanks to my advisor Professor Chua Chin Seng for his insightful guidance, enlightening advice, and endless encouragement throughout my Ph.D. study, which has given me a great opportunity to explore various difficult but interesting problems.

I wish to express my sincere gratitude to the following individual in our research group: Dr. Ren Ying, Dr. Wang Yingjie, Liu Xiaohui, Tao Meng, Gege Putra Kusuma Negara, Tata Tafik Nugraha, Yeo Yong Kiang, Dr. Chen Ying, Dr. Yu Hongchuan and other research students at IMRL for their valuable discussion during the process of this research; Sow Peck Heng and Yuen Sien Huan in IMRL, for their kind assistance in using the laboratory facilities. I would like to acknowledge the School of E.E.E., NTU, Singapore, for awarding me the research scholarship and providing me the excellent research facilities.

Last but not least, I would like to express my deep thanks to my family for their endless love, unconditioned support and understanding through all the time of my study.

Summary

Visual tracking and surveillance has become an active research area of computer vision due to the demand from the public for improved security and safety. The research work of visual tracking and surveillance involves many technical issues, such as motion segmentation, object representation, object tracking and behavior understanding. Among these issues, object representation and tracking are specially important. Being able to maintain tracking of objects in video sequences is not only useful by itself but also a crucial step to higher level video interpretation. The problems are made difficult due to non-stationary environment, persistent and temporary occlusion of multiple interacting objects and low image resolution in cases of distant viewing, which occur frequently in real applications.

A stochastic transductive adaptation method is proposed in this thesis to address the problem of non-stationary object tracking in a complex environment. The proposed stochastic transductive adaptation algorithm combines stochastic transductive learning with a locally exploring particle filter. This adaptive tracker can efficiently and successfully handle non-rigid objects under different appearance changes by its stochastic transductive learning ability. Objects can be tracked well despite severe occlusion or clutter.

An attentive MCMC sampling method has been proposed to localize and track multiple objects. Multiple modes of the joint likelihoods surface are localized and

maintained using attentive sampling, a novel high-dimensional search method based on sampling incrementally on uncertain regions. Experimental results on real sequences show that this is more effective than the traditional MCMC sampler based on random inflated noise, because the proposed algorithm congregates the samples to the most uncertain region. This strategy makes searching more effective in high-dimensional multi-modal distributions.

Maintaining tracks of multiple interacting objects is very difficult where various kinds of mutual occlusions and interactions occur. A large number of interacting objects and frequent occlusions make the problem even worse. In order to achieve robust tracking under severe mutual occlusion, a Color Spatiotemporal MRF model is developed to explicitly model the interactions of objects using color, spatial and temporal correlations. A stochastic search algorithm based on attentive MCMC sampling is used to estimate the optimal states. Experimental results successfully demonstrate the ability to track multiple objects during interactions with occlusions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vision is perhaps the most important part of the human senses. It provides a detailed description of a complex and rapidly changing world. Computer vision, the study of enabling computers to understand and interpret visual information from static image and video sequences, emerged in the late 1950s and early 1960s and is expanding rapidly throughout the world [8, 32].

The growth of the computational power is well characterized by the famous Moore's law for 40 years. The increase of the capabilities of storage devices: electronic storage, magnetic storage and optical storage follows. The advances of sensor technology made CCD (charged couple device) cameras highly available. And the advances in network transmission (e.g., IP camera, wireless LAN), high bandwidth digital interface (e.g., USB, Fireware) and high capability harddisk make setting up a vision system more convenient. All these provide the possibilities of advancing computer vision technology, which needs to manipulate huge amounts of possibly

real-time data, in the hardware aspect. Computer vision is becoming a source of powerful tools for the industry and other disciplines, including multimedia, robot vision, semiconductor manufacturing, medical imaging, image database, virtual reality and surveillance. The potential practical benefits of computer vision systems are immerse. It is anticipated that computer vision systems will soon become popular and the vision technology will be applied across a broad range of products, revolutionizing our lives. Limited by hardware, early vision systems can only perform predefined simple tasks. These systems may not work well in a dynamic, real-world environment. With recent advancements in computing power, today's vision systems are beginning to address some of the real-world issues that their predecessors could not. Complicated algorithms and techniques are continually being developed over the years as the field grows. Even though it is still very far from human performance with respect to adaptively dealing with changing environments, computer vision outperforms human vision in many applications. For example, computer vision systems can provide 24-hour monitoring without tiring or lose of attention. Computer vision systems can also detect small, discontinuously motion while human vision cannot due to the limitation of our eyes. In addition, computer vision systems show their advantages in some unsafe environments where human beings are not suitable to be present. With the availability of high-speed computation facilities, high-accuracy vision sensors and the rapid advancements of video processing algorithms, computer vision is becoming widely used in multimedia applications, video conference, robot vision navigation, image and video retrieval from digital media database, human

computer interaction, medical imaging and visual tracking and surveillance. Among these categories, advancing the techniques of visual tracking and surveillance will be the focus of this thesis.

1.1 Motivation

Visual tracking and surveillance can enable many applications which provide security, information and convenience to our daily lives. Visual tracking by itself can be used directly for intrusion detection and estimation of target trajectory. Further applications involve automatic visual surveillance which facilitates the monitoring of abnormal activities, providing a higher level of security to fight against terrorism and crime; understanding group of objects' behaviors for better assistance in various environment; visual-based human computer interaction for computers to interact with human naturally through vision sensors. Visual tracking and surveillance address the problem of monitoring targets and their activities within secured region using single or distributed cameras. So far most of the surveillance systems employ close circuit TV (CCTV) to provide a large amount of data about the monitored site. However, these traditional surveillance and monitoring system rely heavily on the participation of the human observer. The human observer has to look through the the huge amount of video data captured by the traditional surveillance system and filter out suspicious events and decide whether an event is a potential threat to the security. No matter how highly trained or how dutiful a human observer is, it is impossible to provide full attention to more than one or two things at a

time. A large amount of surveillance video is permanently lost without any useful intelligence being gained from it. The solution to this problem is automatic visual tracking and surveillance [43, 111, 21, 95]. That is, the computer analyses video streams to track targets and determine activities, events or behaviors that might be considered suspicious and provide an appropriate response when such actions occur. Such visual tracking and surveillance systems are still in the stage of research and development in laboratories and is still a long way to go before commercialization. Since the environments of the surveillance applications are real world scenes that are highly variable and irregular, constraints which are usually assumed in general computer vision systems cannot be assumed any more. Keeping track of people, vehicles, and their interactions in an urban or battlefield environment is a difficult task. Many situations, such as illumination changes, unconstrained background, severe and continuous occlusions, which are likely to be present, should be considered in the design of the surveillance system.

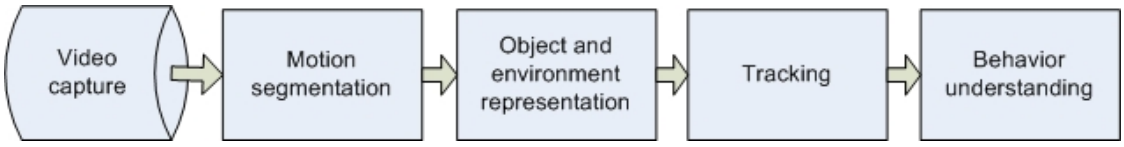


Figure 1.1: The processing flow in an automatic visual tracking and surveillance system.

The processes involved in a typical visual tracking and surveillance system can be outlined in Figure 1.1. Firstly, the foreground objects should be separated from the background environment. Segmented foreground parts are described by some con-

venient representation for reducing the computation and storage, such as silhouette, shape, texture and color. Then the targets are tracked over time based on the targets' representation. Finally, event analysis and activity understanding are carried out based on the information provided by tracking results. Many technical issues are involved in the research work of visual tracking and surveillance, such as motion segmentation, object representation, target tracking, human motion analysis and activity recognition. Among these issues, object representation and target tracking are specially important. Whether the object representation is discriminative and objects can be tracked continuously determine the performance of the surveillance system.

The representation of the target object is mostly a bottom-up process and this needs to cope with changes of appearance of the target. Objects may be represented and tracked using various cues: motion, geometry, shape and color. Among these, the color cue offers many advantages over motion and geometric information which cannot robustly handle partial occlusion, rotation, scale and resolution changes. In addition, color-based tracking can be highly efficient in computation. The color distributions of objects may vary over time due to the changes of illumination, visual angles and camera parameters. The color distribution of the target may not be stable under a non-stationary environment due to the changes of illumination. A static color model would be inadequate to capture the color changes over time. A color model adaptation method that is robust to the changes of environment is required.

Multiple object tracking is another key technology to realize automatic surveillance systems. Being able to maintain track of multiple interacting objects in video sequences is not only useful by itself but also a crucial step to higher level video interpretation. In such a situation, targets are influenced by the behavior of other targets. Such interactions cause problems for traditional approaches to the data association problem. Dealing appropriately with this problem has important implications for tracking of crowd, and is generally applicable to any situation where many interacting objects need to be tracked over time. The basic assumption on which all established data-association methods rely is that objects maintain their behavior before and after the objects visually merge. When these objects encounter each other, some amount of interactions and overlaps occur, and the behavior of involved objects changes after the interaction. Tracking multiple objects in a video sequence requires a modeling of the target objects. The difficulty arises with deformable objects which produce a significant variability in shape and appearance and it is not clear how such variability can be accounted for. The objects may be occluded by static objects in the scene which stands between the camera and the objects. When there are multiple objects in the scene, some of them may occlude each other due to their spatial proximity. When the objects are in a group, the occlusions are often persistent. The occlusions cause missing observations or observations which are collected by multiple overlapping objects. A multiple object tracking technique that is able to handle large number of interacting objects and mutual occlusions is required.

1.2 Objectives of the Research

The goal of this research is to develop an adaptive multiple object tracking method for automatic surveillance systems. Two principle efforts are made in this research. Firstly, adaptive object representation and color tracking are to be investigated, and realized as adaptable to the changes of environment. Secondly, the interactions of multiple objects are to be modeled explicitly. State estimation is achieved by stochastic sampling. The key objectives of this thesis can be summarized as:

- To develop an adaptable model for object representation that is robust to illumination changes and environmental noise.
- To model the interactions of multiple objects explicitly. The modeling method should be able to infer the objects' locations from occluded images.
- To investigate a multiple object tracking method that seeks the optimal state sequence which maximizes the joint state-observation probability. The method should be able to handle inter-occlusions.

1.3 Major Contributions of the Thesis

The research in this thesis focuses on establishing the adaptive object representation and tracking multiple interacting objects. The major contributions of this thesis are:

- A stochastic learning algorithm is proposed for the target model adaptation and color based object tracking. This algorithm employs a stochastic inference

to update the target model dynamically. Combining confidently labelled image data and weighted unlabelled image data, the proposed stochastic adaptation scheme offers an effective way to transduce the object's color model through the given image observations in non-stationary color distributions.

- Attentive Markov chain Monte Carlo sampling (AMCMCS) algorithm is proposed to efficiently search the high-dimensional space associated with multi-object modeling. This method addresses the problem that sample-based Bayesian trackers are often trapped in following incorrect local maxima. This method allows the localization of nearby peaks in the high-dimensional joint-likelihood surface.

- A graphic model based tracking algorithm is proposed for tracking multiple interacting objects. Our major goal is to track and give unique identification to each individual nonrigid object against temporal occlusion and clutter effects which usually occur at intersections. Objects travelling through interactions are moving in various directions. Various parts of these objects may either be occluded by, or themselves occlude, other objects. The interaction potentials of color-spatiotemporal Markov random field (CSTMRF) give the possibility of easily specifying domain knowledge governing the joint behavior of interacting objects. The proposed algorithm follows the assumption that either color, spatial or temporal cues would provide discriminative power at a time. The color, spatial and temporal cues support each other and MCMC iterations sample over joint state space to maximize posterior probability.

1.4 Organization of the Thesis

The organization of this thesis is as follows:

Chapter 2, *Literature Review on Visual Tracking and Surveillance* reviews various approaches to visual tracking and surveillance based on the modules of tracking and surveillance system. These include motion segmentation, object modeling, tracking and behavior understanding.

Chapter 3, *Local Exploring Particle Filter* formulates the general tracking problem as a sequential Bayesian estimation and presents the fundamental of particle filter for Bayesian sequential estimation. An improved particle filter tracking framework which uses mean shift iteration for local maximum exploration is proposed to speed up the convergence of a conventional particle filter.

Chapter 4, *Non-stationary Color Tracking* explains the problem of tracking under non-stationary environment. A stochastic color model adaptation method is presented. The method combines stochastic transductive inference with sample-based particle filtering to make the stochastic color model transduction possible. Experiments on hand and face tracking under dynamic lighting conditions are performed and qualitative and quantitive analyses are provided.

Chapter 5, *Modeling of Multiple Objects* investigates the problems of tracking multiple objects. The method of modeling multiple objects is described, including our hierarchical representation of motion likelihood, color likelihood and multi-object's joint likelihoods, the constraints and prior used.

Chapter 6, *Attentive Markov Chain Monte Carlo Sampling* addresses the issue

of tracking multiple objects when those objects sometimes occlude each other. A stochastic sampling algorithm based on an attentive search strategy is proposed to solve the problem of mutual occlusion. This method can efficiently search the high-dimensional space associated with monocular multi-objects modeling. Experiments are conducted to evaluate the effectiveness of the proposed algorithm.

Chapter 7, *Color-Spatiotemporal MRF-based MCMC Sampling* presents a method which uses a graphic model to represent the interaction of multiple objects. Qualitative and quantitive analysis of the experiments are provided to evaluate the performance of the proposed algorithm.

Finally, Chapter 8, *Conclusion and Recommendations for Future Research* draws conclusions about the theory, implementation and performance evaluation of the designed system with a summary of open problems and perspective for future research.

Chapter 2

Literature Review on Visual Tracking and Surveillance

Over the past two decades, the research progress within the field of visual tracking and surveillance using computer vision has grown significantly. Visual tracking and surveillance in dynamic scenes attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors. The aim is to develop an intelligent visual tracking and surveillance system to replace the traditional passive video surveillance system that is ineffective as the number of surveillance cameras exceeds the capability of human operators to monitor them. The visual tracking and surveillance system can assist us by providing an extended range of monitoring, as well as for providing perception and reasoning capabilities.

A wide range of research projects have been investigated worldwide to improve

the performance of visual tracking and surveillance systems. For example, the Visual Surveillance and Monitoring (VSAM) [21] project was supported by the Defense Advanced Research Projection Agency (DARPA) during 1997-1999. The aim of this project is to develop a multi-sensor surveillance system to support battlefield awareness. There are other surveillance systems under the support of DARPA. The Pfinder system [111] developed by MIT is used to recover the 3-D description of a person in a room. W^4 surveillance system [43] developed by University of Maryland uses shape analysis to detect and track groups of people as well as monitor their behaviors even in the presence of occlusion under the outdoor environment. The Annotated Digital Video for Intelligent Surveillance and Optimized Retrieval (ADVISOR) [77] project was partly supported by the European Community during 2000-2003. ADVISOR was developed by a consortium of industrial and academic partners, including Thales, INRIA, Bull, University of Reading and Kingston University. In this project, new algorithms were developed for motion detection, tracking of people, and behavior recognition.

All of the above research activities are the evidence of the increasing interest in the visual tracking and surveillance technologies. The purpose of this chapter is to present a literature survey of visual tracking surveillance technologies following the processing flow of a video surveillance system as described in Section 1.1. Section 2.1 presents the approaches of motion segmentation which provides a region of interest for further processing. The technologies of object and environment representation are reviewed in Section 2.2. Section 2.3 presents approaches of object tracking which

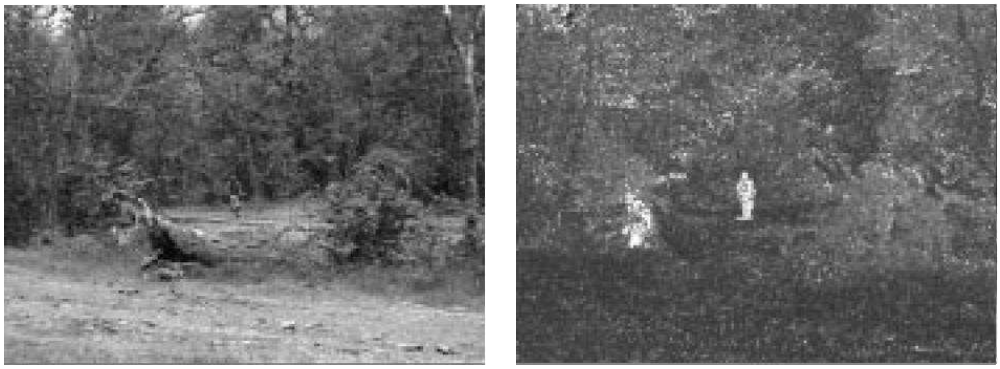
performance decides the effectiveness of an automatic surveillance system. Finally, a short review of behavior understanding approaches is given in Section 2.4.

2.1 Motion Segmentation

Motion segmentation in image sequences aims at detecting regions corresponding to moving objects such as vehicles and humans. Detecting moving regions provides a focus of attention for later processes such tracking and behavior analysis because only moving regions are most interested in surveillance applications. There are many research works being done in the motion segmentation. Most segmentation methods use either temporal or spatial information in the image sequences. Several conventional approaches for motion segmentation are reviewed in the following.

Background subtraction is widely used by simply subtracting the current image from the previous image on a pixel by pixel basis, using either the intensity value or the gradients. If the scene is static, a background image without objects may be recorded and used as reference for subtraction [76, 43]. This method is simple, but extremely sensitive to changes in dynamic scenes derived from lighting and changing environments. The statistical approaches use the characteristics of individual pixels or groups of pixels to extract the figure from the background. This approach is becoming increasing popular due to its robustness compared to the simple subtraction approaches. McKenna *et al.* [71] propose a method combined with the statistics of pixel gradients to remove the shadows cast by subjects. Stauffer *et al.* [99] extend it to a mixture of Gaussians to handle the situations where the value at each pixel

switches between multiple processes such as ripples and tree branches under strong sunlight. Elgammal *et al.* [29, 28] presented nonparametric kernel density estimation techniques as a tool for constructing statistical representation for the scene background and foreground regions in video surveillance. Since the probability density function associated with the background or the foreground does not necessarily follow a known parametric form, kernel estimation methods are shown to be more suitable approach to use in these applications. The background model is based on estimating the probability density function of pixel intensity directly from a set of recent intensity values. The model achieves sensitive detection of moving targets against cluttered backgrounds. The model can handle situations where the scene background is not completely static but contains small motions such as moving tree branches and bushes (see Figure 2.1).



(a) (b)

Figure 2.1: Background subtraction using nonparametric kernel density estimation techniques. (a) Original image. (b) Estimated probability image. (image extracted from Elgammal *et al.* [28]).

Frame differencing makes use of pixel-wise differences between two or three con-

secutive frames in an image sequence to extract moving regions. Frame differencing is very adaptive to dynamic environments, but generally cannot extract all the moving pixels. A commonly known defect is there may be "holes" left inside detected moving regions. Lipton *et al.* [65] detect moving targets in real video sequence using frame differencing. After the absolute difference between the current and previous frame is obtained, a threshold is used to determine changed pixels. By using the connected component labeling method, the extracted moving pixels are clustered into motion regions and small clutters are removed due to the possibility of being noise regions (see Figure 2.2).

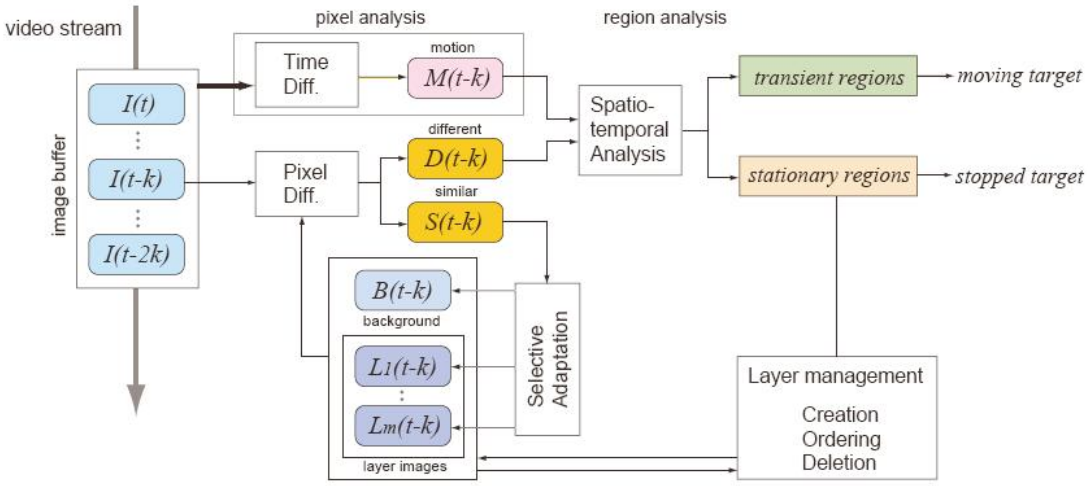


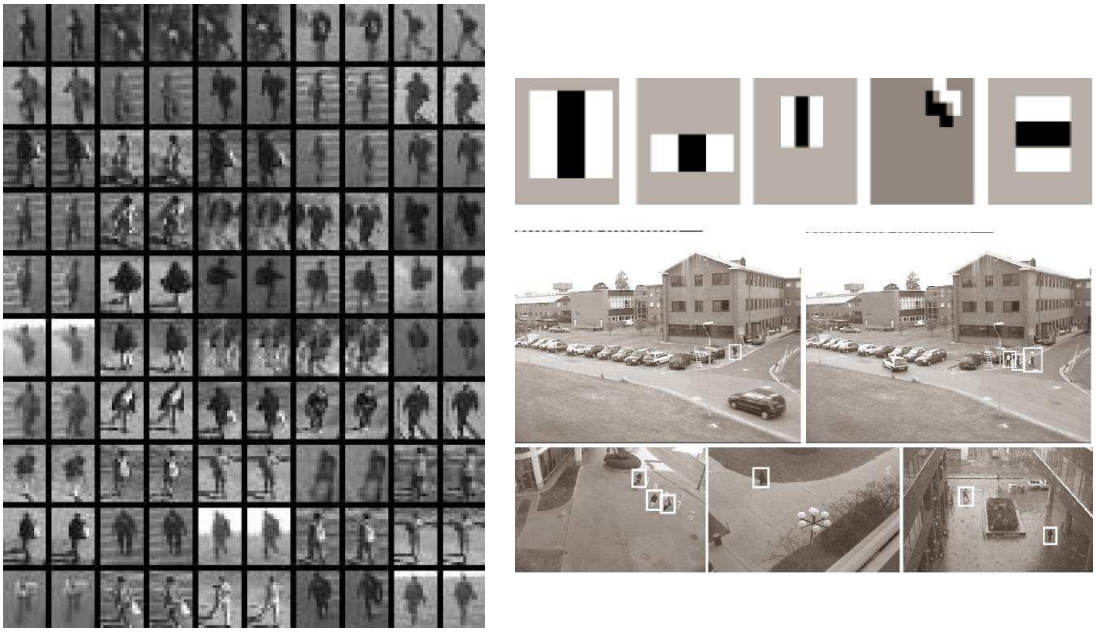
Figure 2.2: Architecture of frame differencing (image extracted from VSAM final report [21]).

Optical flow is used as a general term describing coherent motion of points or features between image frames. Yamamoto and Koshikawa [114] find the motion parameters of a human body part by calculating the optical flow of several points

within this part and compare them to the movements of a model of the part. Optical flow based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise. It may not be suitable for real time applications without specialized hardware support.

Training-based detection is another approach for foreground segmentation which detects foreground based on a prior trained model. Rigoll [90] implements a statistical object tracker. The statistical model is represented by a Pseudo-2D Hidden Markov Model (P2DHMM). This system differs from many other systems in that it does not use any motion information at all for calculating the trajectory of a moving object in an image sequence, resulting in the fact that the proposed approach even makes the tracking of objects possible in the presence of background motions. Viola *et al.* [108] use a learned model that encompasses both the appearance and the motion of pedestrians. The pedestrian model can be evaluated quickly and this makes it feasible to search for pedestrians across the frames of a video sequence. The detector is trained (using AdaBoost) to take advantage of both motion and appearance information to detect a walking person. The implementation described runs at about 4 frames/second, detects pedestrians at very small scales (as small as 20×15 pixels), and has a very low false positive rate (see Figure 2.3).

Motion segmentation under non-stationary viewing sensors has been studied in [75, 47]. In the case of a moving camera, camera motion can first be compensated using a parametric motion model, which is effective for a moving camera, a distant



(a) (b)

Figure 2.3: Detecting pedestrians using patterns of motion and appearance. (a) A small sample of positive training examples. (b) The upper image shows the filters learned for the dynamic pedestrians detector; the lower image shows the example detections for this algorithm. (image extracted from Viola *et al.* [108]).

scene or a near planer scene. In such scenarios the change detection techniques can be applied. Ren *et al.* [88, 89, 19] proposes a spatial distribution of Gaussians (SDG) model to deal with moving object detection having motion compensation that is only approximately extracted. Based on this statistical model, a pixel in the current flame is then classified as belonging to the foreground or background. This method is specially useful when the moving target to be detected/tracked is small, while the traditional motion compensation methods will submerge the small targets.

2.2 Object and Environment Representation

Segmented foreground parts are described by some convenient representation for saving computation and storage. The representations can be categorized as object-based representation, image-based representation and adaptive object representation.

2.2.1 Object-Based Representation

Object-based representation usually uses box, silhouette and blob to represent the foreground region extracted from motion segmentation module. The box representation is widely used. The objects are represented by a set of boundary boxes containing moving pixels or regions. Some systems track these boxes over time [21], while others use the box representation as an intermediate representation [77]. The silhouette representation is popular due to its simplicity. It can be obtained using the threshold or subtraction methods from motion segmentation. It is used for both 2D and 3D images. Haritaoglu *et al.* [41] propose a shape analysis algorithm that determines whether a person is carrying an object and segments the object from the person, so that it can be tracked, such as during an exchange of objects between two people. The method combines periodic motion estimation with static symmetry analysis of the silhouette of a person (see Figure 2.4). The 3D silhouettes can be obtained using combined 2D silhouettes [12]. The blob representation typically follows the results of the motion segmentation approaches. The classified foreground pixels are usually filtered with a median filter and a morphological filter for noise re-

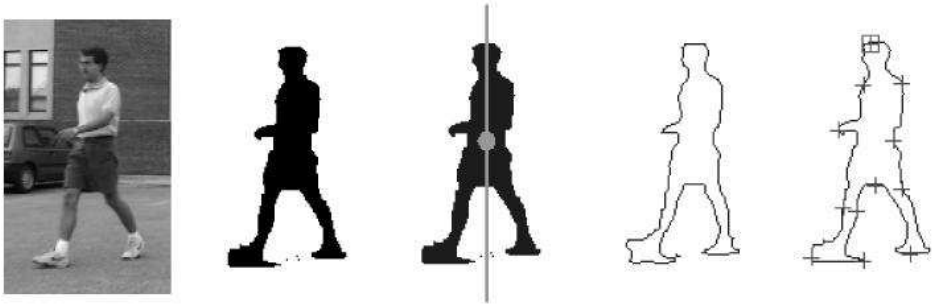


Figure 2.4: Silhouette based shape features used in W^4 (image extracted from Haritaoglu [41]).

duction and to obtain smooth boundaries. Connected components are then labeled on the foreground mask. Each connected component is called a blob. The subject is represented as a blob or a number of blobs each having some similar characteristics. The similarities can be coherent flow [102] or colors [111]. The Pfinder algorithm [111] is a probabilistic method which segments the object into a number of blobs and tracks those blobs over time (see Figure 2.5).

(a) (b) (c)

Figure 2.5: (a) Video input. (b) Segmentation. (c) A 2D representation of the blob statistics (image extracted from Wren *et al.* [111]).

2.2.2 Image-Based Representation

Image based representations are based on the transformations of the pixels of the image. These images may be transformed into another space, yielding a more compact representation of the image. Transformations used are, for example, Fourier, principle component analysis (PCA) [83, 27, 33], Discrete Cosine Transformation (DCT), and wavelets. Viola *et al.* [107] have introduced haar-like wavelet features for the rapid object detection based on a boosted cascade classifier.

2.2.3 Adaptive Object Representation

One of the challenges of object representation is that the object's appearance would change under different lighting conditions or environmental changes. The fixed object models would be inadequate to represent the changes of the object's appearance. changes. Various adaptable object's models have been proposed. Stauffer *et al.* [99] use an adaptive mixture of Gaussians to model background pixels. The background model is updated using an linear on-line approximation. Nummiaro *et al.* [80] implement an adaptive color tracking framework which integrates color distributions into particle filtering and updates color distributions through a linear function. Wu *et al.* [113] present an approach for non-stationary color-based object tracking which adapts color classifiers via model transduction (see Figure 2.6).

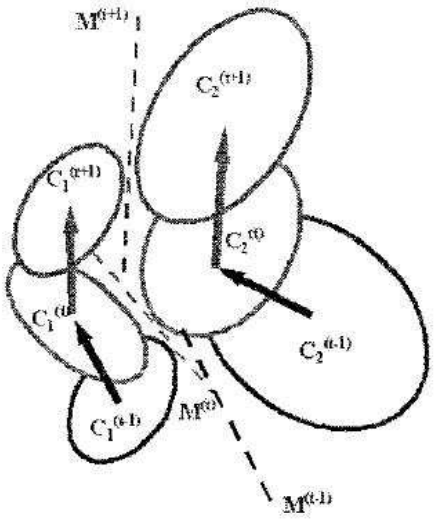


Figure 2.6: An illustration of transduction of classifiers (image extracted from Wu *et al.* [113]).

2.3 Tracking

After motion segmentation and object modeling, tracking and surveillance systems generally track moving objects continually in an image sequence. The early works on target tracking were mainly motivated by radar signal tracking, where the point-like targets against a dark background would be easily segmented. However, for many vision-based tracking problems, the objects to be tracked cannot be assumed as points and the background varies with time. Visual tracking over time typically involves matching objects in consecutive frames using image features.

2.3.1 Region-Based Tracking

Region-based tracking algorithms track objects according to variations of the image regions corresponding to the moving objects. Wren *et al.* [111] adopts a maximum a

posteriori probability (MAP) approach to detection and tracking of the human body using simple 2D models. In their work, a human body is considered as a combination of some blobs respectively representing various body parts such as head, torso and limbs. Both human body and background scene are modeled with Gaussian distributions of pixel values. Individual pixels are then assigned to particular classes: either to the scene texture class or a foreground blob. Each of the individual blobs are then morphologically grown, with the constraint that they remain confined to the foreground region. Therefore, by tracking each foreground blobs, the moving human is successfully tracked.

2.3.2 Active Contour-Based Tracking

Active contour-based tracking algorithms [11] track objects by representing their outlines as bounding contours and updating these contours frame by frame (see Figure 2.7). Active contour-based algorithms describe objects more effectively than region-based tracking algorithm. These algorithms can maintain track under partial occlusion. However, active contour-based algorithms are sensitive to the initialization of contours, which makes it difficult to start automatically.

Alper *et al.* [116] describe a contour-based nonrigid object tracking method. Along with color and texture models generated for the object and the background region, this method can maintain a shape prior for recovering occluded object parts during the occlusion.

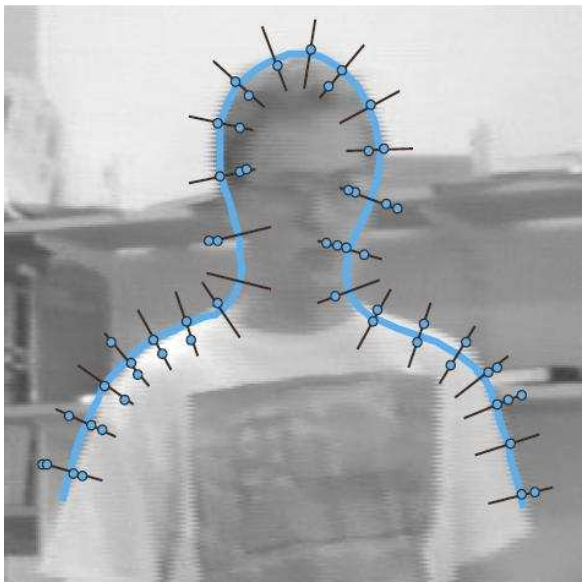


Figure 2.7: A parametric spline curve (image extracted from A. Blake *et al.* [11]).

2.3.3 Model-Based Tracking

Model-based tracking algorithms track objects by matching prerecorded object models, combined with prior knowledge, to image observations.

Tracking over time is finding corresponding objects in consecutive frames. The difficulties of this task are related to the complexity of the scene and the complexity of the tracked objects. Tracking more points in an object is equivalent to tracking multiple objects simultaneously. The points or objects may split and merge into new objects due to occlusion or noise, or the appearance of an object may change due to shadows and change of lighting. The prediction of various state parameters is based on the assumptions of how they evolve over time. A model of velocity and acceleration [74] or more advanced models of movements such as walking [91] may

be used. An alternative approach is to learn probabilistic motion models prior to operation. A commonly used method for prediction is the Kalman filter [5, 50], which is also capable of estimating the uncertainties of the prediction. These uncertainties may be used to determine the regions of interest.

Tracking Based on Deterministic Search

In radar-based tracking, an object is represented by a bright point against a dark background. It can be easily detected after some simple operations. The tracking problem is mainly concerned with the optimal estimation of the object states. A Kalman filter (KF) [5] suffices to provide optimal estimation in a linear system with Gaussian noise. The Extended Kalman filter (EKF) [7] is applied to non-linear systems by linearizing the system equation at each point. The Unscented Kalman filter (UKF) [72] is based on unscented transformations [53], which is a method for generating approximations of Gaussian distributions when they are propagated through nonlinear functions. A classical way of establishing this approximation is linearization as in the case of the EKF, but the unscented transformations provide an alternative which uses a set of discretely sampled points to parameterize the mean and covariance of the posterior density. When the state space is discrete and consists of a finite number of states, Hidden Markov Models (HMM) [86] filters can be applied for tracking. Another approach based on deterministic search is mean shift method proposed by Comaniciu *et al.* [23] which uses the mean shift procedure to perform the optimization.

Tracking Based on Stochastic Search

The Kalman filter is restricted to situations where the probability distribution of the state-parameters is uni-modal. In the presence of occlusion, cluttered background resembling the tracked objects, and complex dynamics, the distribution is likely to be multi-modal. Alternative tracking algorithms have therefore been developed which are capable of tracking multiple hypotheses. The most recognized one of these is the particle filter algorithm [48]. It is based on sampling the posterior distribution estimated in the previous frame and propagating these samples to form the posterior for the current frame. It uses a number of samples to represent the posterior distribution of the estimate of the state instead of a single value (or a single-modal Gaussian as in the Kalman filter case). This increases the robustness of tracking when there is short-term ambiguity. It can also be regarded as a multiple hypotheses technique. However, it suffers from the problem of high dimensionality of the state space, since non-parametric techniques do not scale well with the number of dimensions.

2.3.4 Multiple Object Tracking

Tracking multiple objects in a video sequence is important for a number of tasks such as video surveillance, sports video analysis, human computer interaction and smart conferencing. In those applications, the challenges involved are complex interaction between objects, severe occlusions, unknown number of objects and cluttered background. The model for estimating the interaction is complex especially when

the number of objects is large and mutual occlusions occur frequently.

A multiple hypothesis tracker (MHT) algorithm [5] associates the observation data in a deterministic way, in which possible associations must be exhaustively enumerated. This leads to an NP-hard problem because the number of possible associations increases exponentially with time. In the joint probabilistic data association filter (JPDAF) [5], the association variables are considered to be stochastic variables, and one needs only to evaluate the association probabilities at each time step. However, the above two algorithms do not cope with nonlinear models and non-Gaussian noises, which are common in computer vision applications. Under the assumptions of the stochastic state equation, non-linear state or measurement equation and non-Gaussian noise, particle filters and their extensions [69, 46] are particularly appropriate. The main idea is to propagate a weighted set of particles that approximate the probability density of the state conditioned on the observations. A boosted particle filter is recently developed in [81] which combines mixture particle filter and Adaboost detection to estimate the multi-modal posterior distribution. However, the above work based on the particle filter will suffer computational problem when the dimensionality of state space increases. A Markov chain Monte Carlo (MCMC) based Bayesian human segmentation algorithm in crowded scenario has been proposed in [118]. However, this work uses 3D information in their Bayesian inference. Several methods address the difficulty that the sampling based searches of the original particle filtering converge rather slowly to modes, especially when the observation likelihoods peak in the deep tail of the prior. This is especially

problematic in the high-dimensional state space, where prohibitively long sampling runs are often required for convergence. Cham and Rehg [17] and Merwe *et al.* [72] combine a Condensation style sampling with either local optimization or unscented transformation to speed the convergence. Sminchisescu and Triggs [96, 97] use joint limits and non-self-intersection constraints, and a sample-and-refine search strategy guided by rescaled cost-function covariance to allow monocular tracking of unconstrained human motions in clutter. Recently, a multi-view multi-people detection framework has been proposed in [30] which is able to deal with the occlusions that inevitably occur in a surveillance context.

2.3.5 Human Body Tracking

Human body tracking is an important computer vision challenge in visual tracking and surveillance applications. This involves estimating the human poses in a given image sequence and it is useful for recognition of human gesture, analysis of human activities and understanding of human movement dynamics. Based on the model used in the representation of human body parts, the methods of human body tracking can be categorized in the following styles.

2D model-based approaches

The movements of human body can be represented briefly as the movements of the torso, the head and the four limbs, so the stick figure method is to represent the parts of a human body as sticks and link the sticks with joints. Karaulova *et al.* [57]

use a novel hierarchical model for view independent tracking of the human skeleton figure in monocular video sequences (see Figure 2.8). Ju *et al.* track articulated

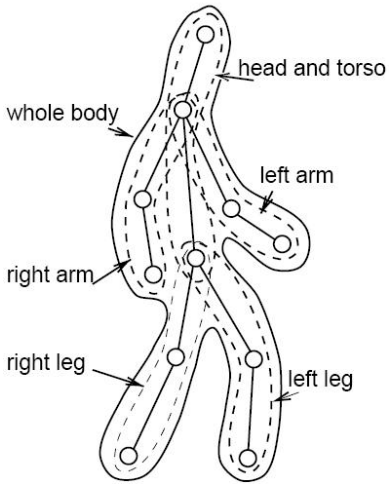


Figure 2.8: The whole body decomposed into parts (image extracted from Karaulova *et al.* [57]).

motion in an image sequence using 2D parameterized models of optical flow [52] (see Figure 2.9).

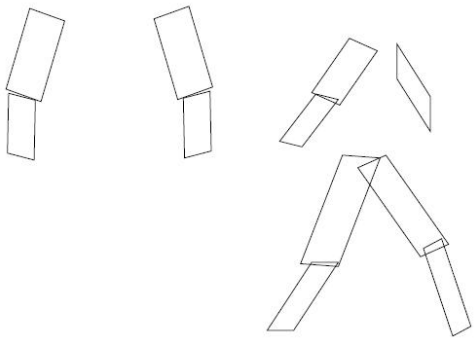


Figure 2.9: The cardboard person model. The limbs of a person are represented by planar patches (image extracted from Ju *et al.* [52]).

3D model-based approaches

The main disadvantages of 2D models is that they require restrictions on the viewing angle. To overcome this disadvantage, many researchers use 3D models such as cylinder and cones [24] (see Figure 2.10). 3D models require more parameters than image-based models and need more computational cost during the matching process. Lee *et al.* [59] estimate 3D human pose in static images using a data-driven MCMC framework. Image observations of different cues provide inferences on the image positions of body joints. This method uses a proposal map as an effective mechanism to consolidate these inferences and generates 3D pose candidates for MCMC (see Figure 2.11).

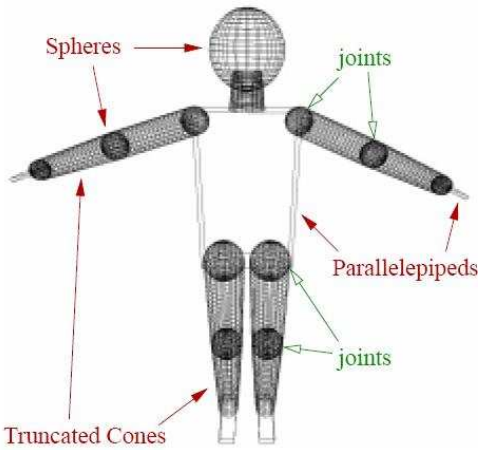


Figure 2.10: 3D model of a person (image extracted from Delamarre *et al.* [24]).

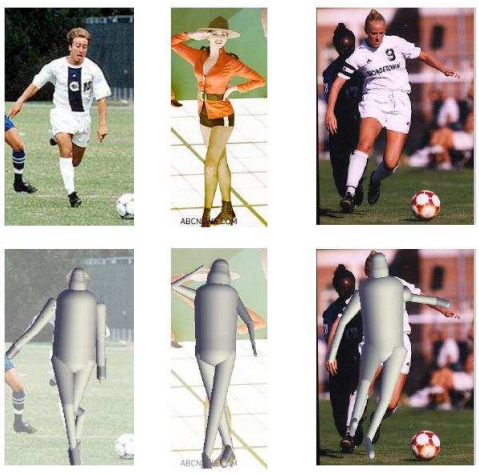


Figure 2.11: Original images and estimated poses (image extracted from Lee *et al.* [59]).

2.4 Understanding Behaviors

After successfully tracking the moving objects from one frame to another in an image sequence, understanding the tracked objects' behaviors is the following stage for an automatic surveillance system. Behavior understanding corresponds to a classification problem of the time-varying feature data that are provided by the preceding stages.

2.4.1 Model-based activity recognition

The hidden Markov model (HMM) is a kind of stochastic state machines [86]. It allows a sophisticated analysis of data with spatio-temporal variability. The use of HMMs consists of two stages: training stage and classification stage. In the training stage, the number of states of a HMM needs to be specified, and the corresponding state transition and output probabilities are optimized in order that the generated

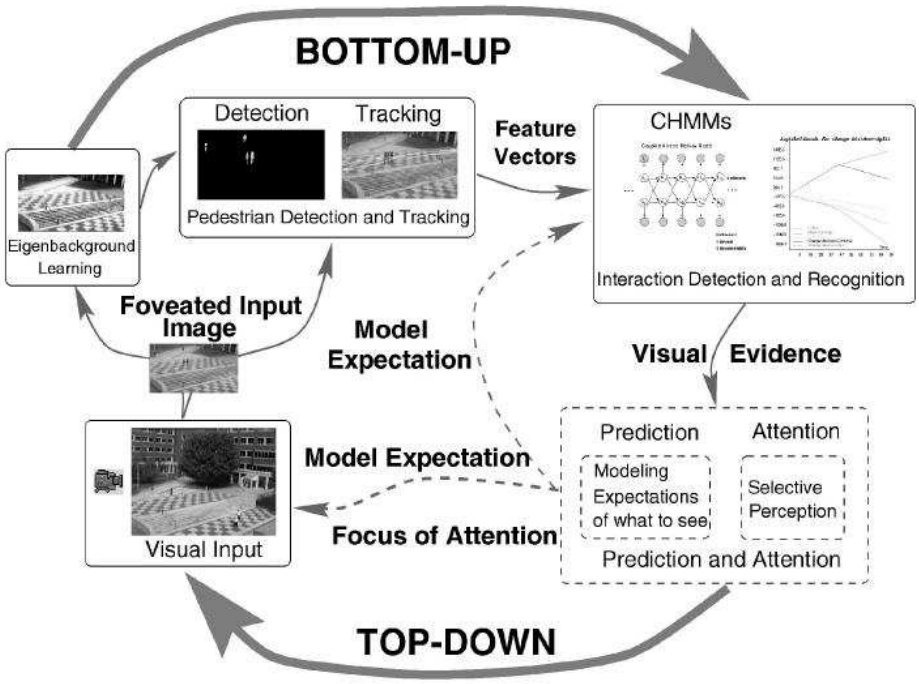


Figure 2.12: Model-based activity recognition using CHMMs (image extracted from Oliver *et al.* [82]).

symbols can correspond to the observed image features of the examples within a specific movement class. In the classification stage, the probability with which a particular HMM generates the test symbol sequence corresponding to the observed image feature is computed. Such behavior understanding systems operate by training a HMM to parse a stream of tracked motions. Starner *et al.* [98] use HMMs for the recognition of sign language. Oliver *et al.* [82] propose coupled hidden Markov models (CHMMs) to classify the interactions between people (see Figure 2.12). The CHMM model is shown to work much more efficiently and accurately than HMMs. Liu *et al.* [67] present an approach to model and recognize multi-agent activities from image sequences based on an improved HMM model which introduces a new

parameter to represent the role of each agent. This improved HMM model can decrease the dimension of the feature space and make the feature space unchanged when the number of agents changes.

2.4.2 Semantic description of behaviors

In some surveillance applications it is important to use semantic description of behaviors. A representative statistical model is the Bayesian network model. This model interprets certain events and behaviors by analysis of time sequences and statistical modeling. Buxton *et al.* [16] describe a system which is controlled by dynamic attention and selective processing. Bayesian networks techniques are used to model dynamic dependencies between parameters involved in the visual interpretation (see Figure 2.13).

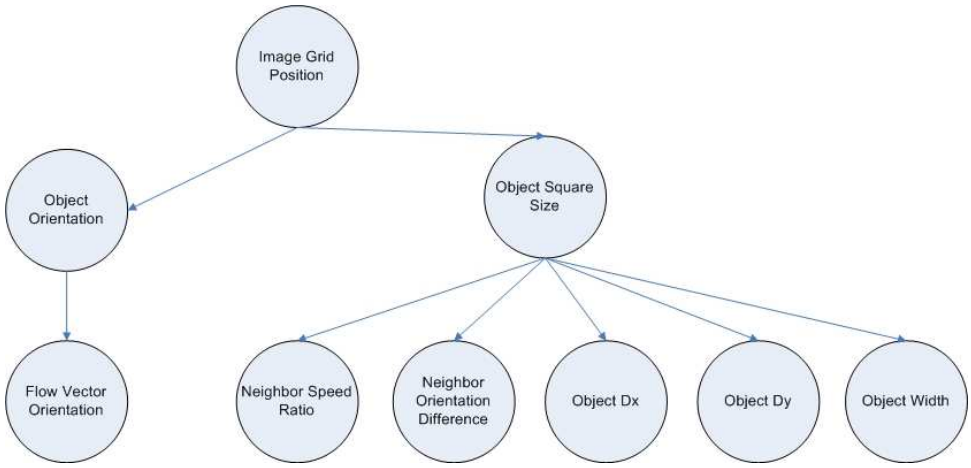


Figure 2.13: A Bayesian network that captures the dependent relationships between the scene layout and relevant measures in motion segmentation and tracking (from Buxton *et al.* [16]).

2.5 Conclusion

Visual tracking and surveillance in dynamic scenes has become an active and important research field recently, which is strongly driven by potential and promising applications, such as intrusion detection, access control in security areas, abnormal behaviors detection and alarming and crowd statistics analysis.

This chapter presents an overview of the state-of-the-art of existing methods in visual tracking and surveillance within a general processing flow for visual tracking and surveillance.

Visual tracking and surveillance systems to date are still in the stage of research and early development. From the above review, it can be observed that the solutions developed are all based on a number of assumptions to make the problems tractable.

Since visual tracking and surveillance is a real-world application, the environments of the visual tracking and surveillance systems are highly variable and unconstrained. The techniques employed are required to be adaptable to the changes of environments. Although some work has been done on adaptive object model, this problem still needs further studies to make the system applicable to more generic environments.

Object tracking for visual surveillance system is still an open problem in computer vision. Robust object tracking techniques are needed which are required to be able to handle temporal occlusions and recover from the failure automatically. The problem of tracking multiple objects is more complex because mutual occlusions occur. Typically, during occlusion, only portions of each object are observable.

Motion segmentation based on subtraction may become unreliable. To reduce ambiguities during occlusion, more complex models need to be developed to cope with the correspondence between image observations and targets.

Chapter 3

Locally Exploring Particle Filter

3.1 Introduction

The tracking of target objects through the frames of an image sequence is gaining more and more attention in many computer vision applications such as video surveillance [71, 43, 40, 20, 21, 56, 42, 94], perceptual user interface [111] and video compression[15]. Robust real-time tracking of non-rigid objects in a dynamic environment is a challenging task. The difficulties lie in complex backgrounds, unknown lighting conditions, complex object movements and occlusions.

The representation of the target object is mostly a bottom-up process and this needs to cope with changes of appearance of the target. Data association and filtering is mostly a top-down approach dealing with the dynamics of the tracked targets, learning of scene priors and evaluation of multiple hypotheses. The way the two major components are combined plays a key role in the design of an effective

and robust tracker.

Objects may be represented and tracked using various cues: motion, geometry, shape and color. Among these, the color cue offers many advantages over motion and geometric information which cannot robustly handle partial occlusion, rotation, scale and resolution changes. In addition, color-based tracking can be highly efficient in computation.

Another major component in a typical tracker is filtering and data association. The most effective formulation of the filtering and data association approach is the state space approach for modeling discrete time dynamic systems [5, 7].

3.1.1 General Tracking Framework

Tracking is the problem of generating an inference about the motion of an object given a sequence of images. Tracking can be seen as a probability inference problem. The key technical difficulty is maintaining an accurate representation of the posterior on object position given measurements, and doing so efficiently.

The general state-space model can be broken into a state transition model $p(x_k|x_{k-1})$ and state measurement model $p(x_k|y_k)$, where $x_k \in R$ denotes the states (hidden variables or parameters) of the system and $y_k \in R$ the observations at time k. It is therefore natural to regard x_k itself as a random variable and the knowledge of the single object is represented by a probability function $p(x)$.

To define the problem of tracking, consider the evolution of the state sequence

$\{x_k, k \in \mathsf{N}\}$ of a target given by

$$x_k = f_k(x_{k-1}, q_k) \tag{3.1}$$

where f_k is possibly a nonlinear function of the state x_{k-1}, q_k is the process noise. The objective of tracking is to recursively estimate the state x_k given the measurements

$$y_k = h_k(x_k, r_k) \tag{3.2}$$

where h_k is possibly a nonlinear non-Gaussian function, and r_k is the measurement noise. From a Bayesian perspective, the tracking problem is to recursively estimate the state x_k at time k, given the data y up to time k. We denote

$$x_{1:k} = \{x_1, x_2, \ldots, x_k\} \tag{3.3}$$

$$y_{1:k} = \{y_1, y_2, \ldots, y_k\}. \tag{3.4}$$

The Bayesian philosophy [36, 9] is that all information about a model is captured by a posterior distribution obtained using Bayes' rule

$$\begin{aligned} posterior &= p(world|observations) \\ &\propto p(observations|world)p(world) \end{aligned}$$

where the prior $p(world)$ is the probability density of the state of the world in the

absence of observations. The term $p(observation|world)$ is called likelihood which encodes the image formation and noise model. The Bayesian philosophy leads to a great deal of recent success in their applications to various problems of computer vision. The Bayesian formulation has been adapted in various areas in computer vision such as image segmentation [105, 119, 106], visual tracking [48, 118], color consistency [14], object recognition [120] and structure from motion [31].

From a Bayesian view, the tracking problem is to recursively estimate the state x_k at time k given the observation $y_{1:k}$ up to time k. Thus, it is required to construct the probability density function $p(x_k|y_{1:k})$. It is assumed that the initial probability density function $p(x_0|y_0)$ is known as *a priori*. Then, the posterior probability $p(x_k|y_{1:k})$ can be obtained recursively by two stages: prediction and correction.

Suppose that the probability $p(x_{k-1}|y_{1:k-1})$ at time $k-1$ is available. The prediction stage involves using the system dynamic model to obtain the prior probability of the state at time k via the following equation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}. \tag{3.5}$$

At time step k, when observation y_k is available, we need to take the prediction as *a priori*, and turn it into *a posteriori* via Bayes' rule. This is the process of correction

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{3.6}$$

where $p(y_k|x_k)$ works as a likelihood function and the evidence

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k \tag{3.7}$$

works as a normalizing constant.

The recurrence relation equation (3.5) and equation (3.6) form the basis for the optimal Bayesian solution. The recursive propagation of the posterior density cannot be determined analytically in general. Solutions do exist in a restrictive set of cases, including Kalman filter and particle filter, which are described in the following paragraphs.

The Kalman filter assumes that the posterior density at every time step is Gaussian and f_k, h_k are known linear functions. If the assumptions hold, the Kalman filter [5] gives an optimal Bayesian solution, which no other algorithm can outperform [4]. However, in a variety of real applications, the assumptions usually do not hold because f_k and h_k are not linear any more and posterior density cannot be assumed as Gaussian distribution in real vision problems. If f_k, h_k are non-linear, approximations are necessary. The Extended Kalman Filter (EKF) [5] uses a local linearization to make a sufficient description of the nonlinearity. However, the EKF always approximates the posterior density $p(x_k|y_{1:k})$ to be a Gaussian.

The main difficulty of tracking in the presence of complicated likelihood functions or of non-linear dynamics is in maintaining a satisfactory representation of $p(x_k|y_{1:k})$. This representation should be able to handle multiple peaks in the distribution. The sequential importance sampling (SIS) algorithm [26] is a Monte Carlo (MC) method.

It is a recursive Bayesian filter by MC simulations. The key idea is to represent the required posterior probability by a set of random samples with weights. As the number of samples is sufficiently large, these samples with weights become an equivalent representation of the posterior probability. Particle filter uses the SIS to approach the optimal Bayesian estimate.

Particle filtering [48] was developed to track objects in dense visual clutter, in which the posterior density $p(x_k|y_{1:k})$ and the observation density $p(y_k|x_k)$ are often non-Gaussian and multi-modal. The key idea of particle filtering is to approximate the required posterior probability distribution by a weighted sample set

$$S = \{(s^{(i)}, w^{(i)})|i = 1, \ldots, N_s\} \tag{3.8}$$

where N_s is the number of samples. Each sample consists of an element s which represents the hypothetical state of an object and a corresponding discrete sampling probability w, where $\sum_{i=1}^{N_s} w^{(i)} = 1$. As the number of samples becomes very large, this estimation becomes an equivalent representation to the usual functional description of the posterior probability and the particle filter approaches the optimal Bayesian estimate. Particle filtering provides a way to model uncertainty. It can keep its options open and consider multiple hypotheses simultaneously.

When the noise sequences are Gaussian and f_k and h_k are linear functions, the optimal solution is provided by the Kalman filter (KF) [5], which yields the posterior which is Gaussian. When the function f_k and h_k are nonlinear, by linearization, the extended Kalman filter (EKF) [5] is obtained; the posterior density is still being

modeled as a Gaussian. The most general filters are represented by particle filters (PF) [26, 4, 48, 79, 84, 46], which are based on Monte Carlo integration methods [38, 31, 3]. The current density of the state is represented by a set of random samples with associated weights and the new density is calculated based on these random samples and weights.

The annealed sampling strategy [25] is proposed to speed up the particle filter, but the main experimental sequence in this paper uses three cameras and a black background to limit the impact of clutters and depth ambiguities. An importance sampling technique with a strong learned prior is proposed in [93]. This method uses a strongly learned prior walking model or a database of motion snippets to track a walking person in an outdoor monocular sequence. Several algorithms address the difficulty that the sampling based searches of pure particle filter converge rather slowly to modes, especially when the observation likelihood peaks deeply in the tail of the prior. The methods proposed in [45, 17, 72] combine the pure particle filter with either a local optimization or the Kalman filtering.

This chapter presents a particle filter solution for non-stationary color tracking using a local exploration method. The target model is represented by a non-parametric density model and the similarity measure is based on a metric derived from mutual information [109]. Efficient proposal distributions containing new observations are obtained through a method of local exploration. Targets can be tracked well despite occlusions or clutters. In the presented tracking examples, the new approach successfully coped with variations of target appearances, severe oc-

clusions and clutters. Section 3.5 will report some of our experimental results on the proposed particle filter with the local exploration algorithm.

3.2 The Conventional Particle Filter

For the sake of completeness, the conventional particle filter is now briefly reviewed. A non-parametric way to represent a distribution is to use particles drawn from the distribution. Let $\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ denote a random measure that characterizes the posterior probability $p(x_k|y_{1:k})$, where $\{x_k^{(i)}, i = 1, \ldots, N_s\}$ is a set of support points with associated weights $\{w_k^{(i)}, i = 1, \ldots, N_s\}$. The weights are normalized as $\sum_i w_k^{(i)} = 1$. Then the posterior density at k can be approximated as

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_k - x_k^{(i)}). \tag{3.9}$$

The approximation converges in distribution when N_s is sufficiently large. This particle-based distribution estimation is, however, only of theoretical significance. In reality, the posterior distribution is the one that needs to be estimated, which is not known *a priori*. Instead, the particles can be sampled from a known proposal distribution $g(x_k|y_{1:k})$, also called an importance density, and still be able to compute $p(x_k|y_{1:k})$. Equation (3.9) says that an unknown distribution p can be approximated by a set of properly weighted particles drawn from a known distribution g. The more difficult problem of density estimation is converted to an easier problem of weight estimation.

Therefore, if the samples $x_k^{(i)}$ are drawn from an importance density $g(x_k^{(i)}|y_{1:k})$, then the weights in Equation (3.9) are defined as

$$w_k^{(i)} \propto \frac{p(x_k^{(i)}|y_{1:k})}{g(x_k^{(i)}|y_{1:k})}. \tag{3.10}$$

Returning to the sequential case, the weight update equation [4] can then be shown to be

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{g(x_k^{(i)}|x_{k-1}^{(i)}, y_k)}. \tag{3.11}$$

Choosing the right proposal distribution is one of the most important issues in the design of particle filter. As pointed out in [4, 72], the optimal proposal distribution is the one that minimizes the variance of the importance weights conditioned on $x_{k-1}^{(i)}$ and y_k. In practice, however, finding the optimal proposal is very difficult. Instead, the conventional particle filters have chosen to trade optimality with ease of implementation by using the transition prior $p(x_k|x_{k-1})$ as the proposal distribution.

Even though this is simple to implement, this proposal results in a higher Monte Carlo variance and thus degrades performance. Comparing the transition prior $p(x_k|x_{k-1})$ with the general proposal distribution $g(x_k^{(i)}|x_{k-1}^{(i)}, y_k)$, it is easy to find that the state space is explored without considering the current observations. Therefore, this filter can be inefficient and is sensitive to outliers.

A common problem with the sequential importance sampling particle filter is the degeneracy phenomenon [72], where after a few iterations, all but one particle will have negligible weights. In addition to choosing better proposal distributions in the

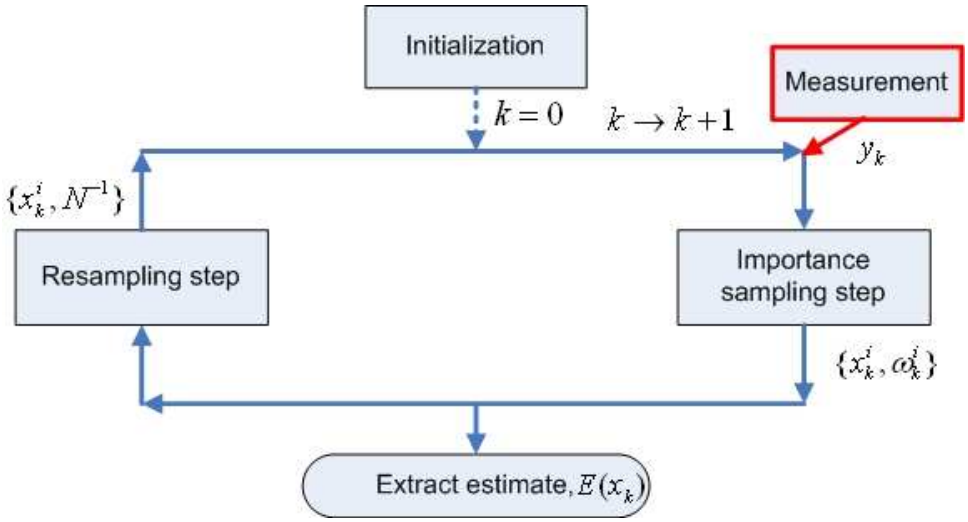


Figure 3.1: The flowchart of the conventional particle filter.

sequential importance sampling step, another crucial step in designing the particle filter is selective resampling. The resampling philosophy [26] is to eliminate particles with low importance weights and multiply particles with high importance weights, thus improving the effective particle size. The flow chart of the conventional particle filter is shown in Figure 3.1. The complete procedure of the conventional particle filter is summarized in Table 3.1.

3.3 Target Representation and Likelihood Measurement

In contrast to the parametric representation of density [27], non-parametric approaches do not assume distribution models and therefore do not require parameter

Table 3.1: The algorithm of the conventional particle filter

Particle Filter Algorithm:

1. Sequential importance sampling

   - Select N_s samples from the set x_{k-1} with proposal distribution $g(x_k|x_{k-1}^{(i)}, y_k)$. The proposal distribution can be the transition prior, as used in the conventional particle filters, or more advanced distributions proposed in Section 3.4.
   - Compute the particle weights by $w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{g(x_k^{(i)}|x_{k-1}^{(i)}, y_k)}$.

2. Resampling

   - Calculate total weights and obtain effective sample size N_{eff}
     - ◇ normalize $w_k^{(i)} \leftarrow w_k^{(i)} \cdot \frac{1}{k}$, $k = \sum_{i=1}^{N_s} w_k^{(i)}$
     - ◇ calculate the effective sample size $N_{eff} = \frac{1}{\sum_{i=1}^{N_s}(w_k^{(i)})^2}$
   - If N_{eff} is lower than a preset threshold, then resample.

3. Extract estimate

   - Estimate the mean state of the set $\{x_k^{(i)}, i = 0, \ldots, N_s\}$, $E[x_k] = \sum_{i=1}^{N_s} w_k^{(i)} x_k^{(i)}$.

estimation. Consequently, it is more flexible, since it is able to model complex data distributions. However, the sacrifice of discriminability is the cost of computational complexity. Non-parametric techniques often require a large number of samples.

3.3.1 Color Space

To characterize the object to be tracked, first a feature space is chosen. The reference object model is represented by its probability density function u in the feature space. For example, the reference model can be chosen to be the color probability density function of the target.

Linear Color Spaces

The perception of color by the human visual system is based on the tristimulus theory. The theory states that color vision results from the action of three cone receptor mechanisms with different spectral sensitivities. When light of a particular wavelength is presented to the eye, these mechanisms are stimulated to different degrees, and the ratio of activity in the three mechanisms results in the perception of a color. Each color is therefore, coded in the nervous system by its own ratio of activity in the three-receptor mechanism.

Since the main variation in human appearance is largely due to luminance change, normalized RGB colors are popularly used, so that the effect of luminance can be filtered out. The normalized colors can be derived from the original RGB components as follows

$$r = \frac{R}{R+G+B} \tag{3.12}$$

$$g = \frac{G}{R+G+B} \tag{3.13}$$

$$b = \frac{B}{R+G+B}. \tag{3.14}$$

The YIQ color space is used in televisions in the United States. One of the main advantages of this format is that gray scale information is separated from color data, so the same signal can be used for both color and black and white sets. In the YIQ format, image data consist of three components: luminance (Y), hue (I), and saturation (Q). The first component, luminance, represents gray scale information, while the last two components make up the chrominance (color information).

The $YCbCr$ color space is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value.

Non-linear Color Spaces

The difficulty with linear color space is that the individual coordinate system does not capture human intuition about the topology of colors. It is common intuition that hues form a circle, in the sense that hue changes from red, through orange to yellow and then green and from there to cyan, blue, purple and then red again. A standard method of dealing with this problem is to construct a color space that reflects these reflections by applying a non-linear transformation to the RGB space. The HSV color space (for hue, saturation and value) is obtained by looking down the center axis of the RGB cube. The family of HSV color spaces is the typical paradigm of perceptual-based color description. The hue (H) is measured by the

angle around the vertical axis and has a range of values between 0^0 (red) and 360^0. It gives us a measure of the spectral composition of a color. The saturation (S) is a ratio that ranges from 0, extending radially outwards to a maximum value of 1 on the triangular sides of the hexcone. This component refers to the proportion of pure light of the dominant wavelength. The value (V) also ranges between 0 and 1 and is a measure of the relative brightness.

The selection of the color space for the object tracking is not trivial. Some color spaces, such as RGB, are more sensitive to the changes of illumination. HSV color space is more stable with the consideration of stability under the changes of illumination as well as reflections and shadows. The HSV color space is exploited in this chapter. In the HSV color space, the influence of the illumination changes, reflections and shadows can be reduced to some extent.

3.3.2 Object Representation

The relationship between an object model and the object's observation is a complex one [109]. Given a target model $u(\xi)$ and an image observation $v(\psi)$ we can formulate the imaging equation

$$v(T_r(\xi)) = F_r(u(\xi)) + \varrho \tag{3.15}$$

where T_r is a transformation that relates the coordinate system of the model to the coordinate system of the target and F_r is the imaging function of the physical world, which can be difficult to model. The reason that it is possible to define F_r is that the observation image do convey information about the model. Clearly if there

are no mutual information between u and v, there will be no meaningful F_r. ϱ is a random variable that models noise in the imaging process.

The entropy of a random variable is defined as $h_e(\psi) \equiv -\int p(\psi) \ln p(\psi) d\psi$. The joint entropy of two random variables z and y is $h_e(y, \psi) \equiv -\int p(y, \psi) \ln p(y, \psi) dy d\psi$.

The mutual information $I_{mi}(u(\xi), v(T_r(\xi)))$ is a measure of the dependence between the two random variables and is defined as

$$I_{mi}(u(\xi), v(T_r(\xi))) \equiv h_e(u(\xi)) + h_e(v(T_r(\xi))) - h_e(u(\xi), v(T_r(\xi))). \tag{3.16}$$

The first step in estimating the entropies from samples is to approximate the underlying probability density $u(\xi)$. In our approach, the target model $u(\xi)$ is estimated by a non-parametric method from a sample A_s drawn from ξ. The kernel density estimation of this distribution $u(\xi)$ is obtained based on a kernel function K.

$$u(\xi) \approx \frac{1}{N_{A_s} h_\sigma} \sum_{\xi_i \in A_s} K_\sigma(\frac{\xi - \xi_i}{h_\sigma}) \tag{3.17}$$

where

$$K_\sigma(\xi) = (2\pi)^{\frac{-t}{2}} |\sigma|^{\frac{-1}{2}} \exp(-\frac{1}{2} \xi^T \sigma^{-1} \xi). \tag{3.18}$$

Next we approximate a statistical expectation over another sample B_s drawn from ξ

$$E_\xi(f_u(\xi)) \approx \frac{1}{N_{B_s}} \sum_{\xi_j \in B_s} f_u(\xi_j). \tag{3.19}$$

An approximation for the entropy of a random variable ξ are calculated as

$$h(\xi) \approx \frac{-1}{N_{B_s}} \sum_{\xi_j \in B_s} \ln \frac{1}{N_{A_s} h_\sigma} \sum_{\xi_i \in A_s} K_\sigma(\frac{\xi_j - \xi_i}{h_\sigma}). \tag{3.20}$$

The approximation of the joint entropy of two random variables ξ and ψ can be obtained over sample A drawn from ξ and another sample C_s drawn from ψ.

$$h_e(\xi, \psi) \approx \frac{-1}{N_{B_s}} \sum_{\xi_j \in B_s} \ln \frac{1}{(N_{A_s} + N_{C_s}) h_\sigma} \sum_{\xi_i \in A,C} K_\sigma(\frac{\xi_j - \xi_i}{h_\sigma}) \tag{3.21}$$

3.3.3 Likelihood Function

Mutual information is only one of the measures of statistical dependence or information redundancy. We have experimented with

$$\rho(u(\xi), v(\psi)) = h_e(u(\xi), v(\psi)) - I_{mi}(u(\xi), v(\psi)) \tag{3.22}$$

which is a metric.

As we want to favor samples containing more information of the target model, the mutual information metric is used for weighting. The probability of each sample is defined as

$$p(y_k|x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\rho^2(u(x_k), v(x_k))}{2\sigma^2}). \tag{3.23}$$

During the following filtering process, the samples with higher weights get higher probability to be chosen, while others with lower weights will be discarded. The

detailed algorithm is discussed in Section 3.4.

3.4 Locally Exploring Particle Filter (LEPF)

The mean shift algorithm [13, 18, 23] is a robust non-parametric technique for climbing density gradients to find the mode (peak) of the probability distribution. For discrete 2D image probability distributions f_m, the mean location within the search window is found as follows:

1. Choose the region windows $R_w^{(i)}$ at size $s^{(i)}$ which is centered at sample point $x^{(i)}$.

2. Compute the mean position within the search window

$$\hat{\overline{a^{(i)}}}(R_w) = \frac{1}{|R_w|} \sum_{j \in R_w} a^j, \tag{3.24}$$

where a is the data point. The mean shift climbs the gradient of $f_m(a)$

$$\hat{\overline{a^{(i)}}}(R_w) - a^{(i)} \approx \frac{f'_m(a^{(i)})}{f_m(a^{(i)})}. \tag{3.25}$$

3. Center the window at location $\hat{\overline{a^{(i)}}}(R_w)$.

4. Repeat steps 2 and 3 until convergence or for a predefined number of iterations.

For particle filters which can model arbitrary distributions, incorporating new observation y_k into the proposal distribution is not an easy task. The conventional

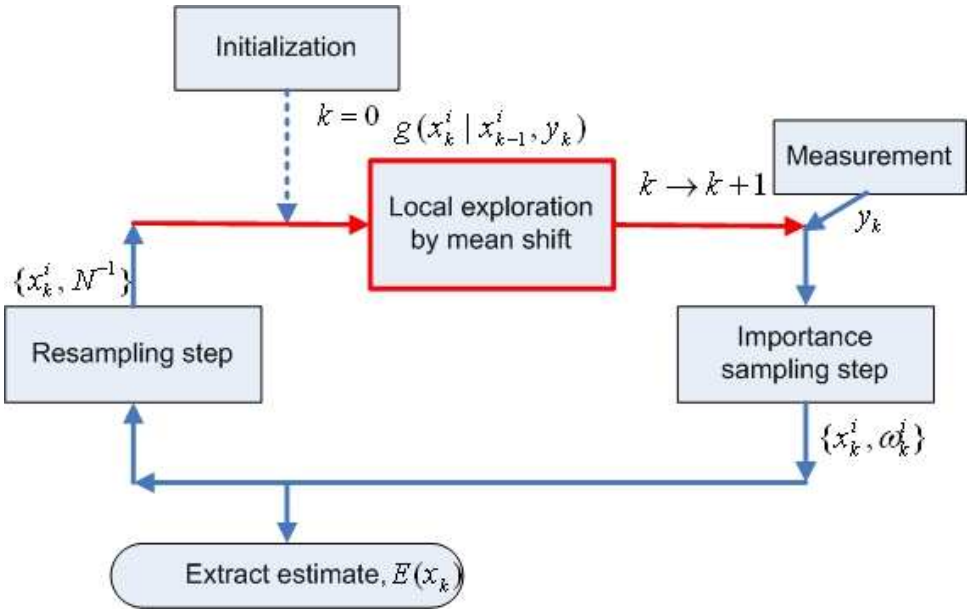


Figure 3.2: The flow chart of our proposed locally exploring particle filter (LEPF).

particle filters simply ignore y_k for ease of implementation. To take advantage of both mean shift and particle filter, and to avoid their limitations, the mean shift is used to generate proposal distributions for the particle filter. Specifically, the proposal distribution for each particle is

$$g(x_k^{(i)}|x_{k-1}^{(i)}) = N(\overline{\hat{a^{(i)}}}(R_w), \sigma_g), i = 1, \ldots, N_s \tag{3.26}$$

where $\overline{\hat{a^{(i)}}}(R_w)$ is the mode of color distribution at each particle's neighboring region and σ_g is the covariance of a. The flowchart of the locally exploring particle filter (LEPF) is shown in Figure 3.2. The iteration steps for a better proposal distribution with local exploration are summarized in Table 3.2.

Table 3.2: The iteration steps for a better proposal distribution with local exploration

Locally Exploring Particle Filter (LEPF) Algorithm:

1. Sequential importance sampling

- Update particles $x_k^{(i)}, i = 1, \ldots, N_s$ with the mean shift calculation (Section 3.4) to obtain $\hat{\overline{a^{(i)}}}(R_w)$.
- Select N_s samples from the set x_{k-1} with proposal distribution $g(x_k|x_{k-1}^{(i)}, y_k) = N(\hat{\overline{a^{(i)}}}(R_w), \sigma_g), i = 1, \ldots, N_s$.
- Calculate the likelihood function $p(y_k|x_k^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\rho^2(u,v)}{2\sigma^2})$
- Compute the particle weights by $w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{g(x_k^{(i)}|x_{k-1}^{(i)}, y_k)}$

3.5 Experimental Results

The LEPF tracker was applied to many sequences. In all the experiments, the HSV color space was taken as the feature space.

Figure 3.3 shows the results of tracking under distraction. The tracked face was moving through a cluttered region which contains another face with a similar color distribution. By propagating a sample-based approximation of the filtering distribution, this locally exploring particle filter (LEPF) can be robust against color clutter.

Figure 3.4 shows a sequence of images where multiple persons interact in a labo-

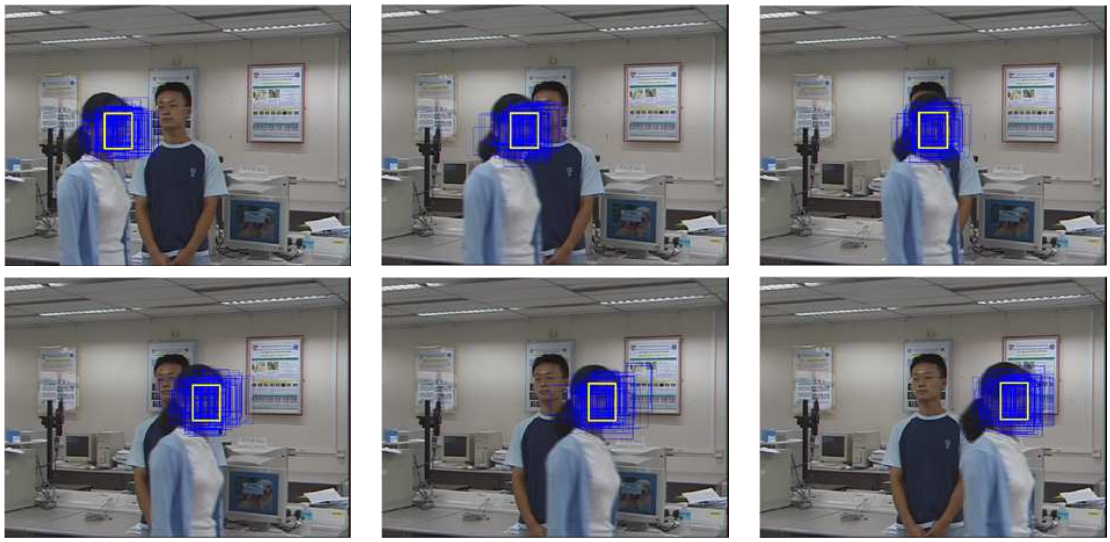


Figure 3.3: Face tracking in clutter sequence. Frames 4, 6, 8, 10, 11 and 13 are shown. The LEPF approach is able to track the identified face despite color clutter. It has the ability to keep track momentarily multiple modes and predict over time. So when the target moves out of the clutter, the tracker can still keep track of the target.

ratory environment. Severe mutual occlusions occurred frequently. The top two rows of Figure 3.4 show the tracking results using a conventional particle filter tracker. When the persons move to a location that is not the same as predicted by the transition prior, the conventional particle filter is easily distracted by severe occlusions and background clutters, because no current observation is taken into account. On the other hand, the LEPF's proposal distribution has the ability to place particles more effectively under a fixed number of particles. It tracks this sequence successfully under severe occlusions, as shown by the two lower rows of Figure 3.4. The quantitative comparisons of LEPF and PF algorithm are shown in Figure 3.5. To evaluate the performance of the tracker, the object tracking error (OTE) is used to indicate the mean distance between the ground truth and the tracked objects'

location.

The locally exploring particle filtering algorithm was tested in the outdoor environment with temporal occlusion. Figure 3.6 shows the tracking results. When the object undergoes temporal occlusion, such as when the person moves behind the pillar (Frame 54), the observation likelihood reduces below a threshold and the model adaptation algorithm will terminate. The ability of particle filter to momentarily track multiple modes of the posterior probability and predict via system model allows us to keep track of the target even after complete occlusion. The results confirm that the locally exploring particle filter tracker allows successful tracking even under temporal occlusion in the outdoor environment.

3.6 Conclusion

We present a locally exploring particle filtering solution of color tracking in a complex environment. The target model is represented by non-parametric density estimation and the similarity measure is based on an integrated computation of mutual information. A better proposal distribution containing new observations is obtained through mean shift iterations. Targets can be tracked well despite severe occlusions or clutters.

conventional particle filter tracker

LEPF tracker

Figure 3.4: Sequence of scenes showing the interaction of multiple persons. Frames 5, 25, 31, 57, 78 and 99 are shown.

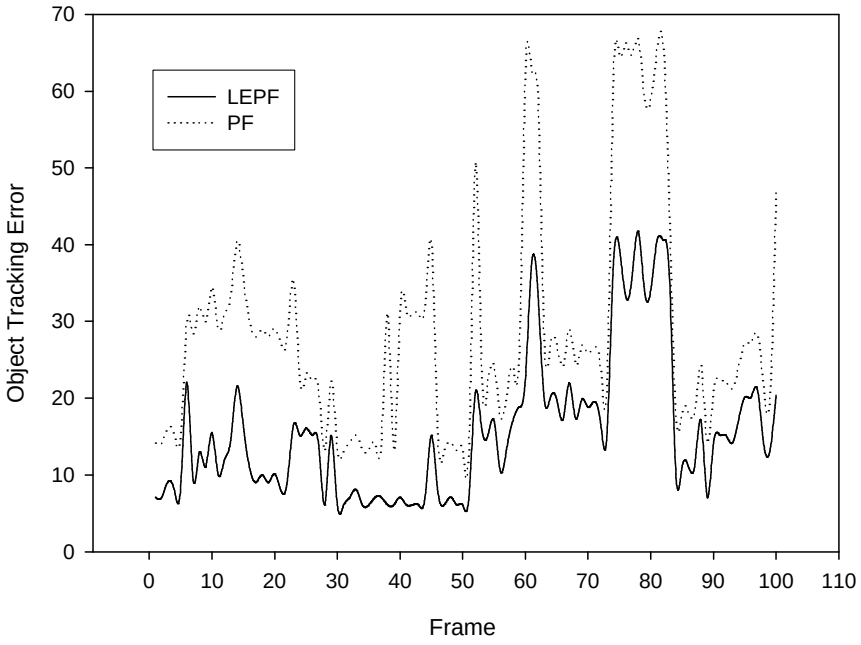


Figure 3.5: The comparison of object tracking errors based on LEPF and PF algorithms. The object tracking error is defined as the mean distance between the ground truth and the tracked objects' location.

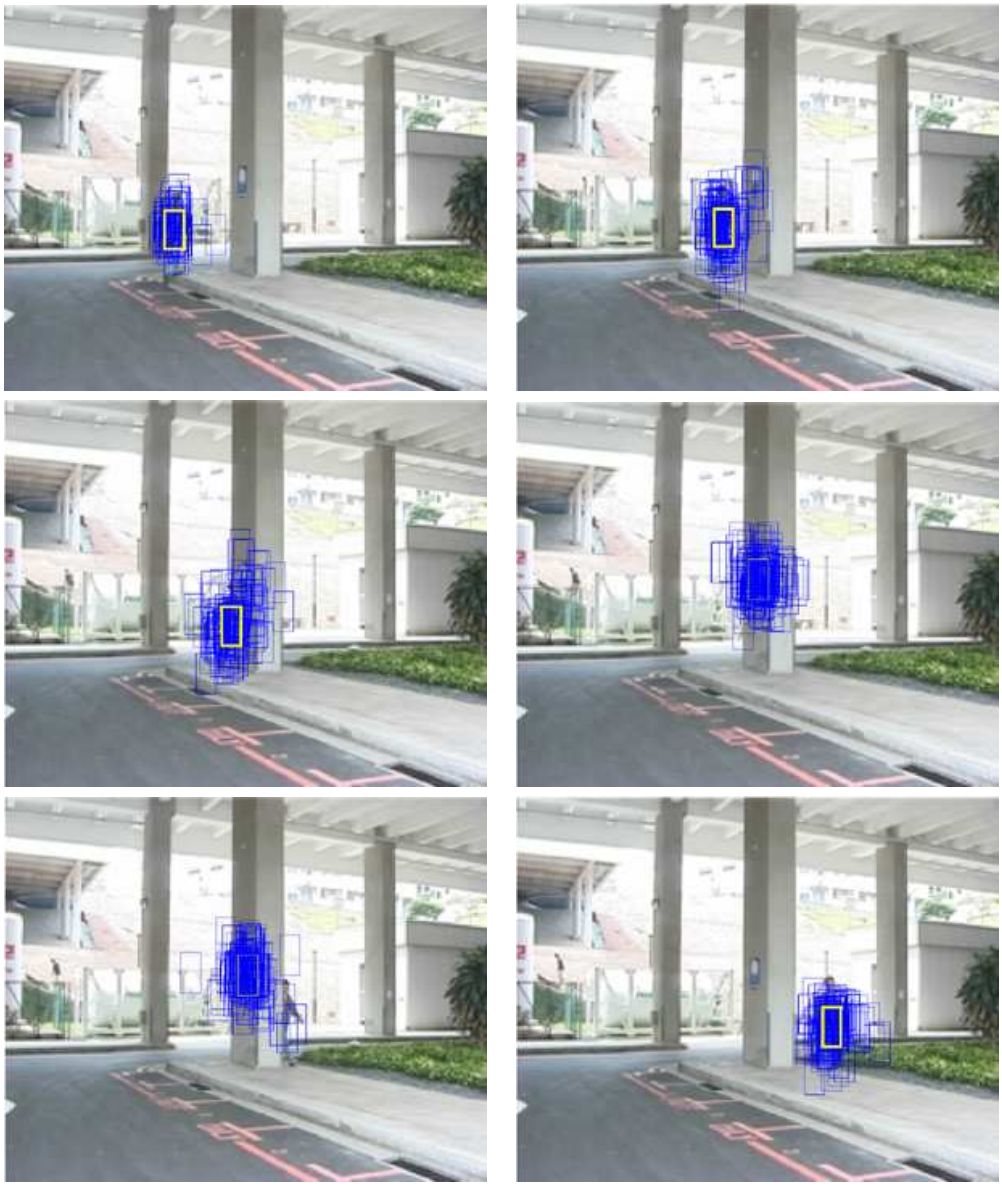


Figure 3.6: Sequence of scenes showing a person walking under complete occlusion in the outdoor environment. Frames 1, 24, 40, 54, 71 and 91 are shown.

Chapter 4

Non-stationary Color Tracking

4.1 Introduction

Recent color-based trackers, proposed in [13, 22, 62, 23], have been proven to be successful in some applications. However, color-based trackers encounter various difficulties as well. Firstly these trackers rely on a deterministic search of a window whose color content matches a reference color histogram model. The search is deterministic in that the starting search position is based on the last iteration's final position. This deterministic search may run into difficulties when the target moves within a similar color clutter or when the target undergoes momentary occlusion. Secondly, the color distributions of objects may vary over time due to changes of illumination, visual angle and camera parameters. If the object's color model is trained for a specific situation, the tracker may not work well in a dynamically changing situation. It is a good practice to learn a generic color classifier for color-

based tracking by collecting a large labeled data set [51]. If some color invariants could be found, learning such a color classifier would suggest a direct and robust way of color tracking. Generally, this color classifier would be highly nonlinear, and a huge labeled training data set is required to achieve good generalization. However, when we consider the non-stationary color distribution, which is common in real applications, we do not generally expect to find such invariants.

Various methods are available to address the above problem of a non-stationary color distribution. The color model adaptation proposed in [87] uses a Gaussian mixture model to represent color distribution. The color model adaptation is performed by adjusting the parameters of the Gaussian mixture model through a set of labeled training data from the previous frame. An adaptive, color-based, particle filter proposed in [79, 80] uses color histogram to represent the color distribution. The mean state histogram calculated by the particle filter is used as labeled training data to adjust the color model. However, since the training set is not segmented beforehand, the training data is not reliable. Discriminant expectation maximization (D-EM) algorithm, proposed in [112, 113], formulates the non-stationary color tracking problem as a transductive learning problem. Gaussian mixture models are used in this algorithm to model the color distributions. A Gaussian assumption is used in the expectation maximization (EM) while a linear assumption is used in the D-EM. However, the tracking of objects in dense visual clutter usually involves the posterior density and observation density to be non-Gaussian and non-linear where Gaussian assumptions are often invalid.

This chapter presents a stochastic transductive learning method for non-stationary color tracking. The target model is represented by a non-parametric density model and the similarity measure is based on a metric derived from mutual information. We employ a transductive inference to update the target model dynamically. Combining confidently labeled data and weighted unlabeled data, the proposed stochastic transductive inference offers an effective way to transduce object color model through the given observations in non-stationary color distributions. The way the transductive adaptable object model and locally exploring particle filter are combined plays a decisive role in the robustness and efficiency of the tracker. In the presented tracking examples, the new approach successfully coped with illumination changes. Section 4.3 will report some of our experimental results on the proposed stochastic transductive learning algorithm.

4.2 Color Model Adaptation by Stochastic Transductive Inference

4.2.1 Stochastic Color Model Transduction

The approach taken in [79, 22, 84] is an inductive learning approach, by which the color model learned at initialization should be able to classify or weight any color region at any time. In the inductive approach the learner tries to induce a decision which has a low error rate on the whole distribution of samples for a particular learning task. This approach is unnecessarily complex and needs large training

data.

In our approach, we propose a novel stochastic transductive inference [61, 60] to deal with the problem of tracking under unforeseen environment. We assume that the color model u_k at time k can give confident labels to several data in image I_{k+1}, so that the samples in I_{k+1} can be divided into two parts: labeled samples $L_d = \{(x^{(j)}, l^{(j)}), j = 1, \ldots, N_l\}$ and unlabeled samples $U_d = \{x^{(j)}, j = 1, \ldots, N_u\}$, where N_l and N_u are the size of the labeled samples and unlabeled samples respectively, $x^{(j)}$ is the sample set, $l^{(j)}$ is its label, and C is the number of classes. The transductive classification can be written as

$$l^{(i)} = \arg \max_{j=1,\ldots,C} p(l^{(j)}|x^{(i)}, L_d, U_d : \forall x^{(i)} \in U_d). \tag{4.1}$$

From this equation, the color model u_k is transduced into another color model u_{k+1} given the unlabeled data from I_{k+1}.

The color model u_k at time k is only a weak classifier at time $k+1$ because of the changing lighting conditions, such as the changing of illumination and the influence of reflection. Certainly, we cannot expect much from this weak classifier. However, for each unlabeled sampling data $x^{(j)} \in U_d$, the classification confidence $W_u^{(j)}$ can be calculated based on the particle weights assigned by this weak classifier u_k as following

$$W_u^{(j)} = w_{k-1}^{(j)} \frac{p(y_k|x_k^{(j)})p(x_k^{(j)}|x_{k-1}^{(j)})}{g(x_k^{(j)}|x_{k-1}^{(j)}, y_k)}. \tag{4.2}$$

After that, the color model adaptation will be performed on the new weighted data

set

$$D' = L_d \bigcup \{x^{(j)}, W_u^{(j)}) : \forall x^{(j)} \in U_d\}. \tag{4.3}$$

4.2.2 A Stochastic Transduction Algorithm for Adaptive Color Tracking

The image histogram at the mean state location $h_{E[x_k]}$ can be seen as the confidently labeled data at time t. $\{h_{x^{(j)}}, j = 1, \ldots, N_s\}$ are the image histograms located at the sampling region $x^{(j)}$. $W_u^{(j)}$ is used to weight the unlabeled data. The color model is transduced stochastically according to the following equation

$$u_{k+1} = (1 - \alpha - \beta)u_k + \alpha h_{E(x_k)} + \beta \sum_{j=1,\ldots,N_s} h_{x^{(j)}} W_u^{(j)} \tag{4.4}$$

where α and β are scalars between 0 and 1 that allow us to adjust the speed of model transduction. In our experiments, $\alpha = 0.25$ and $\beta = 0.25$ are fixed empirically.

The object model should not be adapted too readily because the object may encounter partial occlusions, outlier clutter or some transient tracking failures. It is easy to adapt erroneously to some parts of background in the above situations. We use an adaptive threshold to address this problem and we update the object color model only when the observation probability is above a pre-defined threshold. The update rule is

$$w_{E(x_k)} > w_{T_w} \tag{4.5}$$

where $w_{E(x_k)}$ is the observation likelihood of the mean state and w_{T_w} is a threshold.

4.2.3 Iteration Steps of the Stochastic Model Transduction Framework

The stochastic model transduction framework is shown in Figure 4.1 and the detailed algorithm is summarized in Table 4.1.

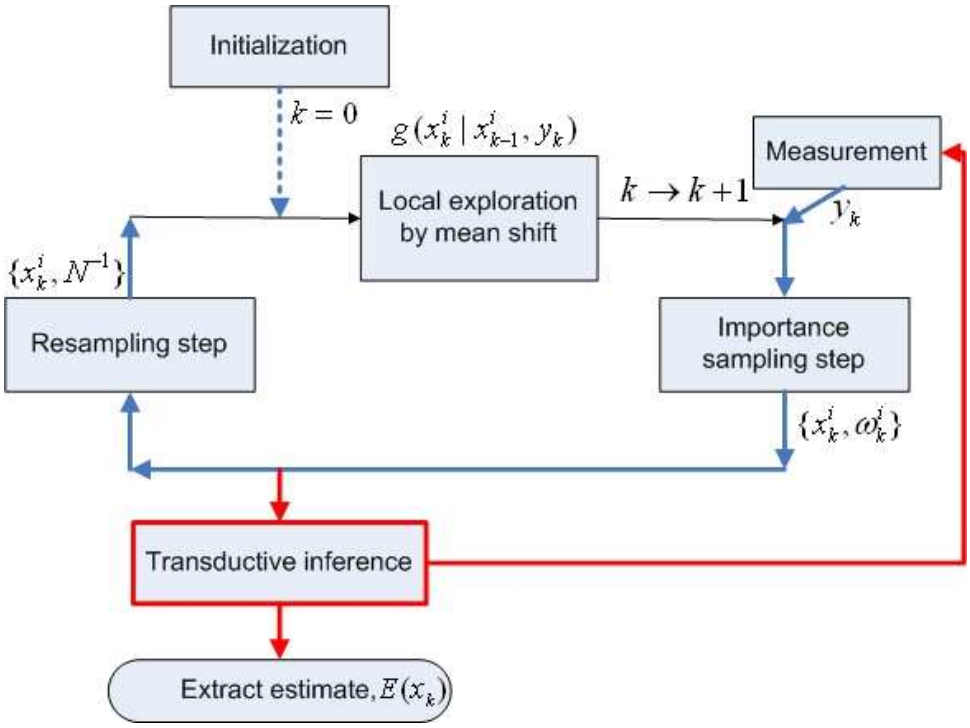


Figure 4.1: The framework of stochastic model transduction for color tracking.

4.3 Experimental Results

The transductive particle filter tracker is applied to many sequences. In all the experiments, the HSV color space is taken as the feature space.

Figure 4.2 shows a comparison of experimental results on hand tracking using a

Table 4.1: The stochastic model transduction framework

1. Sequential importance sampling

   - Update particles $x_k^{(i)}, i = 1, \ldots, N_s$ with the mean shift calculation as Section 3.4 to obtain $\overline{\hat{a}^{(i)}}(R_w)$.
   - Select N_s samples from the set x_{k-1} with proposal distribution $q(x_k|x_{k-1}^{(i)}, y_k) = N(\overline{\hat{a}^{(i)}}(R_w), \sigma_q), i = 1, \ldots, N_s$.
   - Calculate the likelihood function $p(y_k|x_k^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\rho^2(u,v)}{2\sigma^2})$
   - Compute the particle weights by $w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, y_k)}$

2. Resampling

   - Calculate total weights and get effective sample size N_{eff}
     - $\diamond$ normalize $w_k^{(i)} = w_k^{(i)} \cdot \frac{1}{k}$, $k = \sum_{i=1}^{N_s} w_k^{(i)}$
     - $\diamond$ calculate the effective sample size $N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^{(i)})^2}$

3. Extract estimate

   - Estimate the mean state of the set $\{x_k^{(i)}, i = 0, \ldots, N_s\}$, $E[x_k] = \sum_{i=1}^{N_s} w_k^{(i)} x_k^{(i)}$

4. Transductive Inference for Object Model Adaptation

   - if the observation likelihood of mean state is above a threshold $w_{E(x_k)} > w_{T_w}$
   - update object color model by transductive learning $u_{k+1} = (1 - \alpha - \beta)u_k + \alpha h_{E(x_k)} + \beta \sum_{j=1,\ldots,N_s} h_{x^j} W_u^{(j)}$

static color model and our proposed transductive color model. The algorithm was tested on the PETS-ICVS03 [85] smart meeting sequences. The false alarm rate (FAR) is characterizes the tracking performance of the object tracking algorithm which is defined as $FAR = \frac{Total\ False\ Positives}{Total\ True\ Positives + Total\ False\ Positives}$. In Figure 4.2, the hand is rising from the bottom to the top. The hand at the bottom is darker because it is in the shadow of the body and the skin color changes significantly when the hand moves up. The changes in hue and saturation can be seen from Figure 4.3. The left column sequence in Figure 4.2 shows the tracking results of using a static color model in the frame 1, 10, 11 and 19. When the hand moves out of the body's shadow, the static color model approach shows larger errors in localization compared with the right column sequence which shows the results of using our proposed transductive color model adaptation algorithm.

Figure 4.4 shows the experimental results on the face tracking. The method using a static color model gives a localization box containing background pixels when the face moves near the light source. The transductive color model adaptation algorithm can adapt the color model by transductive learning in the framework of particle filter and it can give better tracking results under non-stationary color distributions. Figure 4.5(a) presents the change in the means of hue and saturation for the face tracking experiment. The mean of hue changes significantly along the various frames of the sequence. As a result, the errors for the particle filter approach increase significantly at around frame 55. With the transductive particle filter algorithm, the object color model can be adapted along the video sequences. Even when the

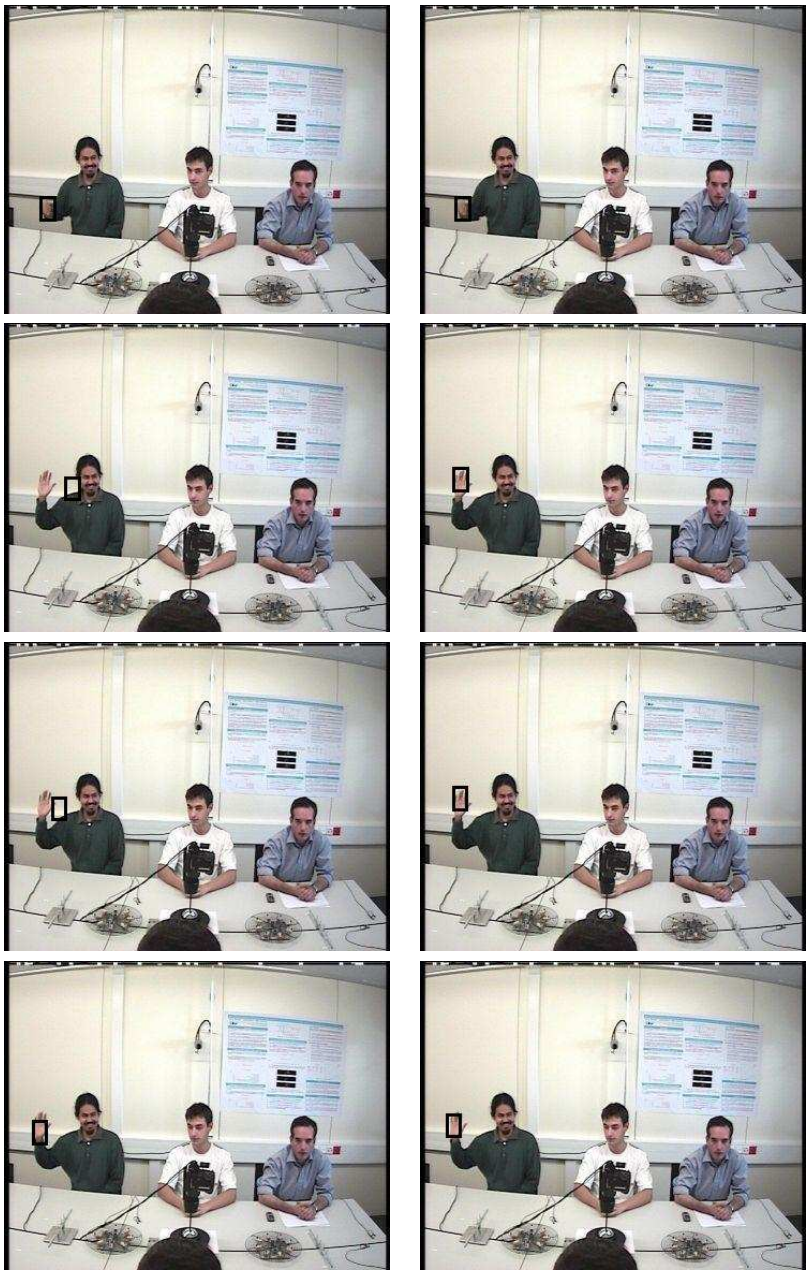
lighting changes significantly, the errors of the transductive particle filter algorithm remain relatively bounded compared with normal particle filter.

4.4 Conclusion

We present a transductive particle filtering solution of non-stationary color tracking in a complex environment. The proposed transductive particle filter algorithm combines the transductive object model with a better proposal distribution particle filter.

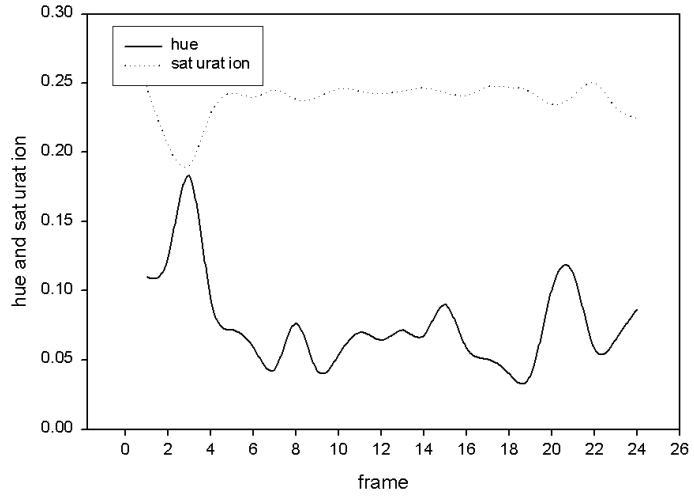
The target model is represented by a non-parametric density estimation and the similarity measure is based on an integrated computation of mutual information.

This color-based tracker can efficiently and successfully handle non-rigid objects under different appearance changes by its transductive learning ability. A better proposal distributions containing new observations are obtained through mean shift iterations. Targets can be tracked well despite severe occlusions or clutter.

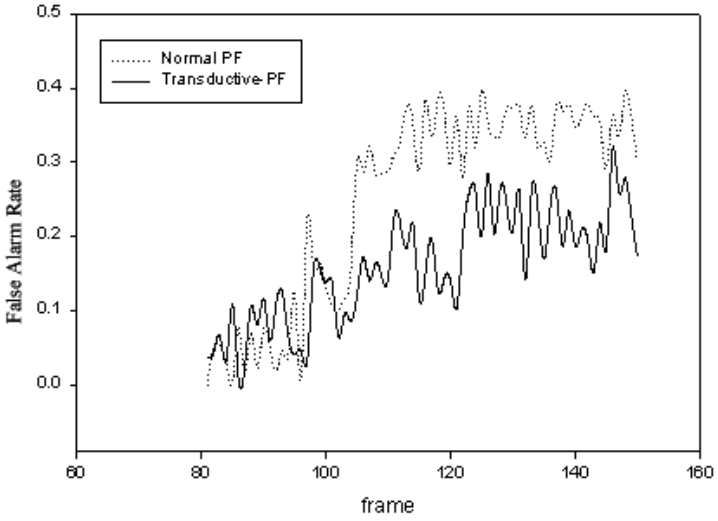


(a) (b)

Figure 4.2: Comparison of experimental results on hand tracking in the frame 1, 10, 11 and 19. The target is the right hand of the person on the left. The left column shows the results from conventional particle filter algorithm while the right column shows the results from transductive particle filter algorithm.



(a)

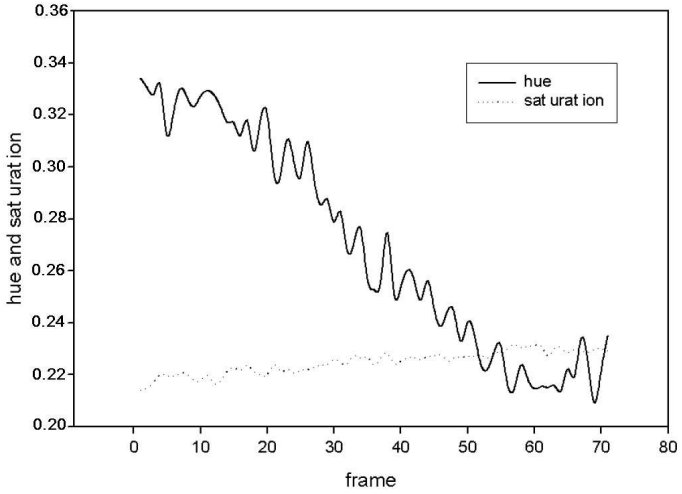


(b)

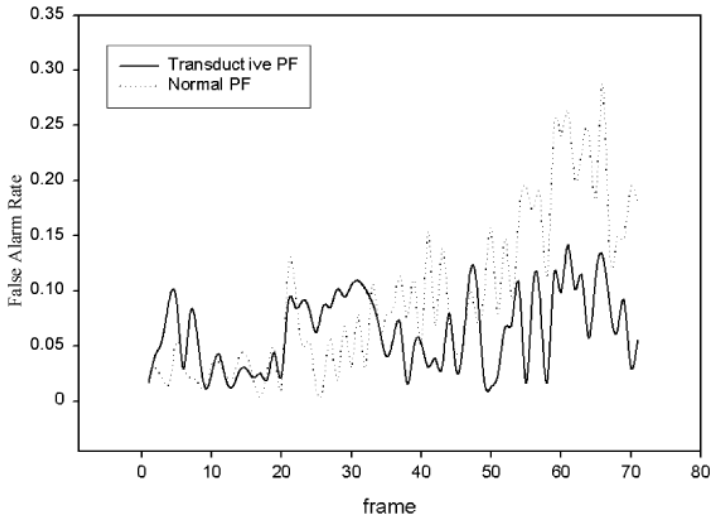
Figure 4.3: Hand tracking experiments (a) The changes in the means of hue and saturation. (b) The error rates of conventional PF and transductive PF.

(a) (b)

Figure 4.4: Comparison of experimental results on face tracking in the frame 3, 60, 65 and 70, the left column shows the results of particle filter algorithm and the right column shows the results of transductive particle filter algorithm.



(a)



(b)

Figure 4.5: Face tracking experiments (a) The changes in the means of hue and saturation. (b) The error rates of conventional PF and transductive PF.

Chapter 5

Modeling of Multiple Objects

Tracking multiple objects in a video sequence is important for a number of tasks such as video surveillance, sports video analysis, human computer interaction and smart conference. In these applications, the challenges involved are complex interactions between objects, severe occlusions and cluttered background. During occlusion, only portions of each object are visible and are often at a very low resolution. This problem is generally intractable, and motion segmentation based on background subtraction may become unreliable. To reduce ambiguities due to occlusion, better models need to be developed to cope with the correspondence between image observations and objects, and thus eliminate correspondence errors that occur during tracking multiple objects. The model for estimating the state spaces of multiple objects is complex especially when the number of objects is large and mutual occlusions occur frequently.

The modeling aspects of the problem of multi-object tracking from monocular

video are discussed in this chapter. In particular, the way of constructing the joint observation likelihood function is described. Given the high-dimensionality of the problem and the presence of clutter, the problem is ill-conditioned. We pay particular attention to the robust and probabilistically consistent integration of color, motion and configuration information as well as prior knowledge that constrains the search space.

A probability distribution is an approach for computing expectations. One useful technique is to represent the posterior by drawing a large number of samples from that distribution. These samples can then be used to estimate any expectation with respect to that posterior.

5.1 Related Work

A well-known early work in multiple object tracking is the multiple hypothesis tracking (MHT) [6]. MHT forms hypotheses of different data associations and calculates the relative probabilities of them. Each hypothesis is scored by its posterior and the algorithm returns a hypothesis with the highest score as a solution. Based on these calculated probabilities it forms a set of most probable data association histories, which is propagated in time and updated using the sensor measurements. Hence, MHT is capable of initiating and terminating a varying number of tracks and suitable for surveillance applications. However, the construction of new hypotheses requires an enumeration of all possibilities and the size of hypotheses grows exponentially with respect to time. The pruning techniques are necessary for real applications

[115]. However, the heuristics are used at the expense of optimality and the algorithm will degrade in a dense environment. The probabilistic multiple hypothesis tracking (PMHT) algorithm [100] is presented to model the data association as random variables which are estimated jointly with state estimation by expectation maximization (EM) iterations. Probabilistic data association filter (PDAF) and joint probabilistic data association filter (JPDAF) [6] are multiple target tracking algorithms that represent the state posteriors as a set of Gaussian distributions, such that each Gaussian distribution represents one target. Given a fixed number of targets, JPDAF enumerates all possible associations between the observations and the known tracks and clutter and computes each association weight. For each association, the conditional expectation of the state of a target is estimated by a filtering algorithm. Then the state of a target is estimated by summing over the conditional expectations weighted by the association weights.

However, the above two algorithms do not cope with nonlinear models and non-Gaussian noises, which are common in computer vision applications. Under the assumptions of the stochastic state equation, non-linear state or measurement equation and non-Gaussian noise, particle filters [48] and their extensions [69, 46] are particularly appropriate. The main idea is to propagate a weighted set of particles that approximate the probability density of the state conditioned on the observations. A probabilistic exclusion principle has been developed in [69] to track multiple objects, but the algorithm is very dependent on the observation model and is only applied for two objects. A Bayesian multiple-blob tracker (BraMBLe) has

been proposed in [70]. BraMBLe deals with a varying number of objects which are depth-ordered due to the use of a 3-D state space. Schulz *et al.* [92] uses particle filters and statistical data association to track multiple moving targets with a mobile robot. This approach uses particle filters to track the states of the objects and applies JPDAFs to assign the measurements to the individual objects. This technique is similar to the one proposed in [69]. However, instead of relying on Gaussian distributions extracted from the sample sets, this approach applies the idea of JPDAFs directly to the sample sets of the individual particle filters to solve the correspondence problem between the detected features and the filters. A boosted particle filter is recently developed in [81] which combines mixture particle filter and Adaboost detection to estimate multi-modal posterior distribution. This method uses a cascaded Adaboost algorithm to learn models of the hockey players. These detection models are used to guide the particle filter. However, the above work based on the particle filter will suffer computational problems when the dimensionality of state space increases and Adaboost can hardly detect occluded targets. A Markov chain Monte Carlo (MCMC) based Bayesian human segmentation algorithm in crowded scenario has been proposed in [117, 118]. However, this work uses 3D information in their Bayesian inference. Several methods address the difficulty of slow convergence from the sampling based searches of the original particle filtering, especially when the observation likelihoods peak in the deep tail of the prior. This is especially problematic in the high-dimensional state space, where prohibitively long sampling runs are often required for convergence. Cham *et al.* [17] and Merwe *et al.* [72]

combine a Condensation style sampling with either local optimization or unscented transformation to speed the convergence. Sminchisescu *et al.* [96] use joint limits and non-self-intersection constraints, and a sample-and-refine search strategy guided by rescaled cost-function covariance to allow monocular tracking of unconstrained human motions in clutter.

5.2 Problem Formulation

In Chapter 3 we have introduced the basic Bayesian tracking inference for single object tracking. Now we extend the Bayesian tracking inference into the tracking of multiple objects. Let M be the number of objects to be tracked. This is not known *a priori* and needs to be estimated. The index i designates one among the M objects. The goal of multi-object tracking is to sequentially estimate the state vector made by concatenating the state vectors of all objects. It is generally assumed that the objects are moving according to independent Markov dynamics. At time k, $X_k = (x_{1,k}, \ldots, x_{M,k})$ follows the temporal evolution which is given by the following state equation

$$x_{i,k} = f_{i,k}(x_{i,k-1}, q_{i,k}) \qquad i = 1, \ldots, M. \tag{5.1}$$

For two objects, i and $i^{'}$, the noises $q_{i,k}$ and $q_{i^{'},k}$ are supposed to be white, both temporally and spatially, and independently for $i \neq i^{'}$.

The observation vector collected at time k is denoted by $Y_k = (y_{1,k} \ldots y_{i,k} \ldots y_{M_k,k})$. The index i, used as the first subscript, refers to one of M_k measurements. The vec-

tor y_k is composed of detection measurements and clutter measurements.

Following the Bayesian philosophy which is discussed in Section 3.1.1, the multi-object tracking problem is formulated as an estimation of the posterior probability which is decomposed into a likelihood term and a prior term,

$$p(X_k|Y_k) \propto p(Y_k|X_k)p(X_k). \tag{5.2}$$

The tracking algorithm solves the estimation problem in a maximum *a posteriori* (MAP) formulation, that is, the tracking result is the current multi-object state with the largest posterior probability based on current and previous observations.

5.3 Model Priors

The prior probability of the state is assumed to be the product of the prior probabilities of all the objects and is defined as

$$p(X_k) = \prod_{i=1}^{M} p(H_i). \tag{5.3}$$

The prior probability of an *ith* object is made up of the prior probability on its hypothesized sample area H_i,

$$p(H_i) = e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i}) \tag{5.4}$$

where the first term penalizes large object sizes which avoids unnecessary over-

lapping. The second term penalizes objects with small image sizes since they are more likely to be due to image noise. The values of λ_1 and λ_2 are adjusted empirically according to the experiment.

5.4 Multi-Cues Observation Likelihood

Whether continuous or discrete, the search process depends critically on the observation likelihood component of the parameter space cost function. Besides the smoothness property, the likelihood function should be designed to limit the number of spurious local minima in parameter space. In order to be able to represent multiple objects, all objects in the image are modeled using a hierarchical representation which includes pixel-level, object-level and configuration-level representations. Pixel-level likelihood explains how the observed motion pixels support the region covered by the hypothesized state space. Object-level likelihood models objects' color similarity with given observations. Configuration-level likelihood is based on geometric information to solve the problem of occluded observations. The integration of multiple visual cues improves the robustness of the system. Each cue has its own constraints to ensure its efficiency and reliability in applications. By integrating multiple visual cues, the reliability of tracking performance is increased.

5.4.1 Pixel-Level Likelihood

The stationary camera gives us the luxury to have a view-based background model without considering most of its geometry. There are many possible approaches

[35, 1, 28, 103] to motion segmentation.

Motion segmentation is based on an adaptive background subtraction method [99] that models each pixel as a mixture of Gaussians and uses an on-line approximation to update the model. The Gaussian distributions are then evaluated to determine which Gaussians are most likely to result from a background process. Each time their parameters are updated, the Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the background process. Pixel values that do not match one of the pixel's background Gaussians are grouped using connected components. Each pixel in the scene is modeled by a mixture of K Gaussian distributions. The probability that a certain pixel has a value of b_k at time k can be written as

$$p(b_k) = \sum_{j=1}^{K} \omega_{j,k} * \eta(b_k, \mu_{j,k}, \Sigma_{j,k}) \tag{5.5}$$

where $\omega_{j,k}$ is the weight parameter of the jth Gaussian component. $\eta(b_k, \mu_{j,k}, \Sigma_{j,k})$ is the Gaussian probability density function

$$\eta(b_k, \mu_{j,k}, \Sigma_{j,k}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_{j,k}|^{\frac{1}{2}}} e^{-\frac{1}{2}(b_k - \mu_{j,k})^T \Sigma_{j,k}^{-1} (b_k - \mu_{j,k})} \tag{5.6}$$

where $\mu_{j,k}$ is the mean and $\Sigma_{j,k} = \sigma_{j,k}^2 I$ is the covariance of the jth Gaussian distribution at time k.

The K distributions are ordered based on the fitness value ω_j/σ_j and the first B distributions are used as a model of the background of the scene where B is

estimated as

$$B = \arg\min_{b}(\Sigma_{j=1}^{b}\omega_j > T_b) \tag{5.7}$$

where the threshold T_b is the minimum fraction of the background model. It is the minimum prior probability that the background is in the scene. Background subtraction is performed by marking a foreground pixel any pixel that is more than a threshold away from any of the B distributions. The first Gaussian component that matches the best value will be updated by the following update equations,

$$\omega_{K,k} = (1-\alpha)\omega_{K,k-1} + \alpha(M_{K,k}) \tag{5.8}$$

$$\mu_k = (1-\rho)\mu_{k-1} + \rho b_k \tag{5.9}$$

$$\sigma_k^2 = (1-\rho)\sigma_{k-1}^2 + \rho(b_k - \mu_k)^T(b_k - \mu_k) \tag{5.10}$$

$$\rho = \alpha\eta(b_k|\mu_k, \sigma_k) \tag{5.11}$$

$$M_{k,k} = \begin{cases} 1, \text{ if } \omega_k \textit{ is the first matched Gaussian component}; \\ 0, \text{ otherwise.} \end{cases} \tag{5.12}$$

where ω_k is the kth Gaussian component. α defines the time constant which de-

termines change. Figure 5.1 shows the motion segmentation results based on this method.

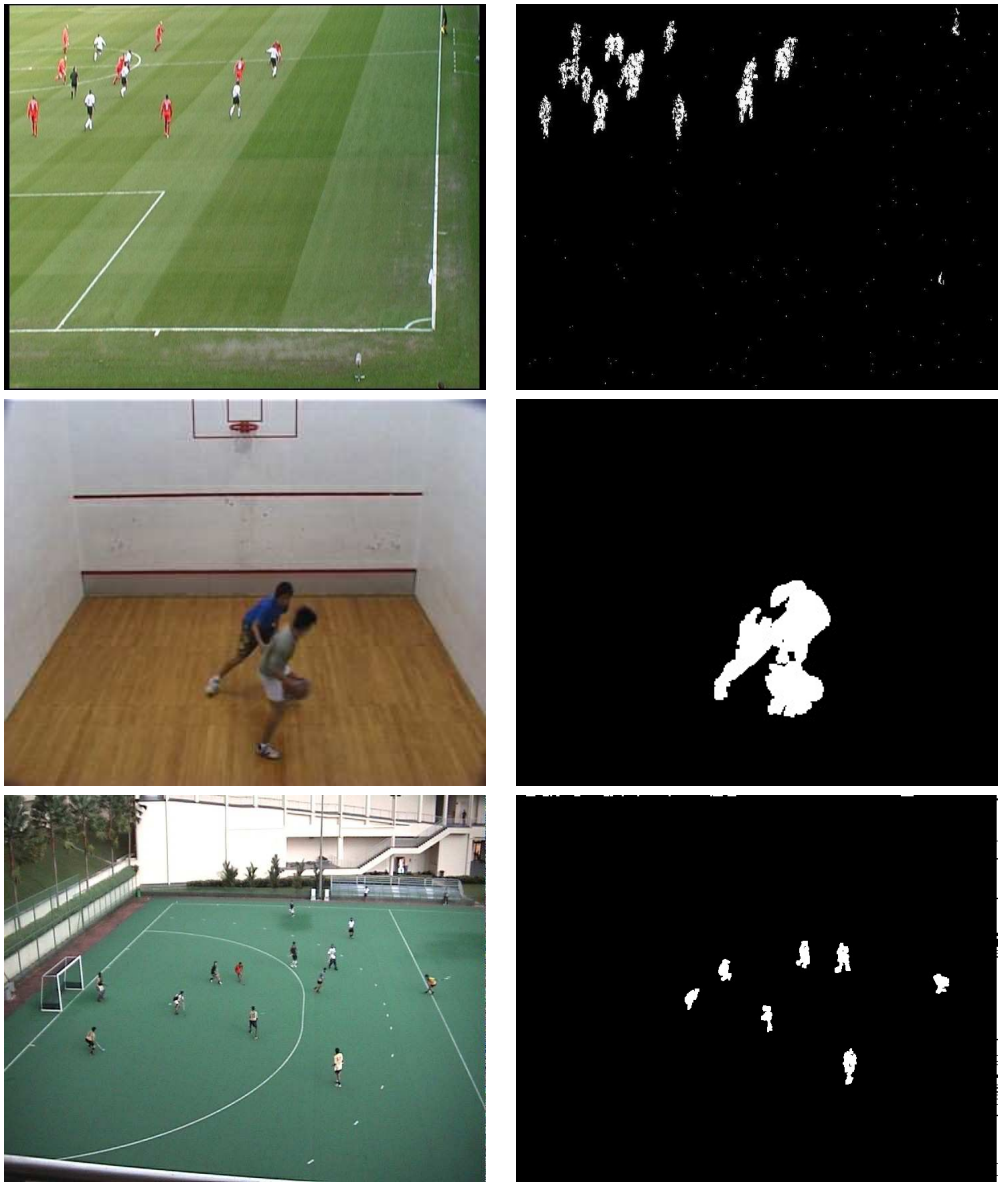


Figure 5.1: The original image frame and its foreground detection result from adaptive background subtraction method.

5.4.2 Object-Level Likelihood

To characterize the color feature of the target objects, a feature space needs to be chosen. The reference target model is represent by its probability density function g in the feature space. The hypothesized candidate object is characterized by h. Both probability density functions are to be estimated from the data. To satisfy the low-computational cost imposed by multi-object tracking, n_b-bin histograms are used.

Target Reference Model and Hypothesized Candidate

The parameters of an individual object i are defined as $x_i = \{c1_i, c2_i, c3_i, c4_i\}$. Parameters represent, respectively, the coordinates of the object's centroid and the width and height of the object's bounding box.

To achieve robustness against rotation, partial occlusion and non-rigidity, color histograms g_i is used to represent the color distribution of target object i.

The pixels in the outlying areas of the target have a higher probability to be considered as clutter than that of the inner part. A weight is employed to the pixels within the target region in such a way that the pixels which are further away from the centroid of the target region are assigned smaller weights. The weighting function is defined as

$$\tau(\xi) = \begin{cases} 1 - \xi^2 & \xi < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.13}$$

where ξ is the normalized distance from the region centroid. Thus, with this weighting function, the target model's robustness against outlier clutter has been enhanced. Let $\{z_i\}_{i=1...n_h}$ be the pixel locations of the target candidate, centered at $\hat{z}$ in the current image. The color distribution $g(\hat{z}) = \{g_{(u)}(\hat{z})\}_{u=1...n_b}$ is calculated as

$$g_{(u)}(\hat{z}) = f \sum_{i=1}^{n_h} \tau(\frac{\|z_i - \hat{z}\|}{a})\delta[h(z_i) - u] \tag{5.14}$$

where δ is the Kronecker delta function and n_b is the number of bins. A function h is defined to associate to the pixel at location z_i. The index $h(z_i)$ of the histogram bin corresponds to the color of that pixel. The normalization factor is

$$f = \frac{1}{\sum_{i=1}^{n_h} \tau(\frac{\|\hat{z}-z_i\|}{a})} \tag{5.15}$$

where a is used to adapt the size of the region and defined as

$$a = \sqrt{c3_i^2 + c4_i^2}. \tag{5.16}$$

This histogram is not the best nonparametric density estimation, but its low-computational cost suits the requirement of multi-object processing.

A geometric constraint for human body is employed for the purpose of human tracking which is shown in Figure 5.2.

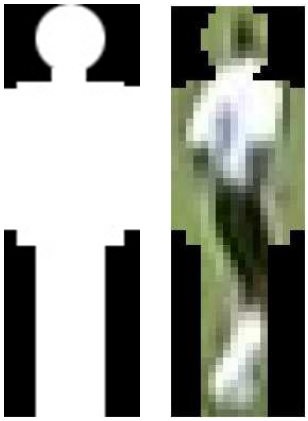


Figure 5.2: Figure model of human body.

Similarity Function

For a given object, the likelihood $L(y_{j,k}, x_{i,k})$ measures how well the image data supports the presence of this object. Histogram intersection [101] is used as the matching measurement

$$\rho[h_j, g_i] = \frac{\sum min(h_j, g_i)}{\sum g_i} \tag{5.17}$$

where h_j is the histogram of sampled image region and g_i is the color model of the tracked object. The larger the value of ρ, the more similar it is between two color histograms. A distance between two color distributions can be defined as the measure

$$d = \sqrt{1 - \rho[h_j, g_i]}. \tag{5.18}$$

The vector K_k is introduced to describe the associations between the measurements and the objects. Each component K_k^j is a random variable that takes its values among $\{0, \ldots, M\}$. Thus, $K_{j,k} = i$ indicates that $y_{j,k}$ is associated with the

ith object. The assumption used here is that one measurement can originate from only one object or clutter. The color likelihood of each observation can be computed as

$$L(y_{j,k}, x_{i,k}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} \qquad if \quad K_{j,k} = i. \tag{5.19}$$

5.4.3 Configuration-Level Joint Likelihood

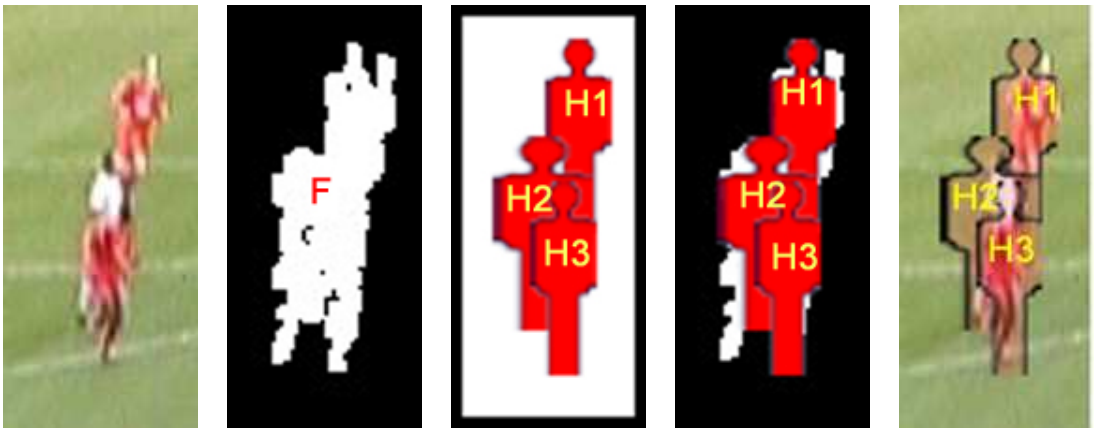
Since multi-object tracking may encounter severe occlusion, the joint-likelihood cannot be decomposed into the product of individual object's likelihood. The joint likelihood can be decomposed into different levels: object level and configuration level with exclusion principle.

For a given object, the object likelihood $L(y_{j,k}|x_{i,k})$ measures how well the observation data supports the presence of this hypothesized object. The joint object-level likelihood is computed as the geometric average of $L(y_{j,k}|x_{i,k})$ as following

$$L(Y_k|X_k) = \sqrt[M]{\prod_{i=1}^{M} L(y_{j,k}|x_{i,k})}. \tag{5.20}$$

Denote F as the foreground motion region and H as the hypothesized object region which are shown in Figure 5.3. All foreground motion regions F are assumed to be caused by moving objects. Υ is used to measure the coverage of hypothesized objects and is defined as

$$\Upsilon = \frac{|F \cap (\cup_{i=1}^{M} H_i)|}{|F|}. \tag{5.21}$$



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

Figure 5.3: The explanation of different regions. (a) the original image (b) the motion area F (c) the hypothesized sample area H (d) the hypothesized sample area matched to motion area F (e) the hypothesized sample area matched to original image.

Similarly,

$$\Phi = \frac{|F \cap (\cup_{i=1}^{M} H_i)|}{|\cup_{i=1}^{M} H_i|} \tag{5.22}$$

is used to measure the ratio between the hypothesized objects that has been supported and the amount of cost.

The multiple objects joint likelihood is calculated as

$$p(Y_k|X_k) = L(Y_k|X_k)^{\alpha} \cdot (\Upsilon\Phi)^{\beta} \tag{5.23}$$

where α and β are constants which control the importance of object-level likelihood and configuration-level likelihood. The values of α and β are adjusted through experiments. The explanation of the multi-object's joint likelihood is shown in Figure 5.4. The white masks are the foreground pixels extracted from motion segmentation

as introduced in Section 5.4.1. The red human-shape masks are the hypothesized samples as explained in Figure 5.3. Highest configuration likelihood is obtained if all hypotheses explain well all foreground motion pixels as shown in image (a) of Figure 5.4. If too many hypotheses are used to explain the same regions or foreground pixels are not explained by the hypotheses, the configuration likelihood is low which is shown in images (b), (c) and (d) of Figure 5.4.

5.5 Problem Formulation

We aim towards a probabilistic interpretation and optimal estimates of the model parameters by maximizing the posterior probability according to the Bayes rule

$$p(X_k|Y_k) \propto p(Y_k|X_k)p(X_k) \tag{5.24}$$

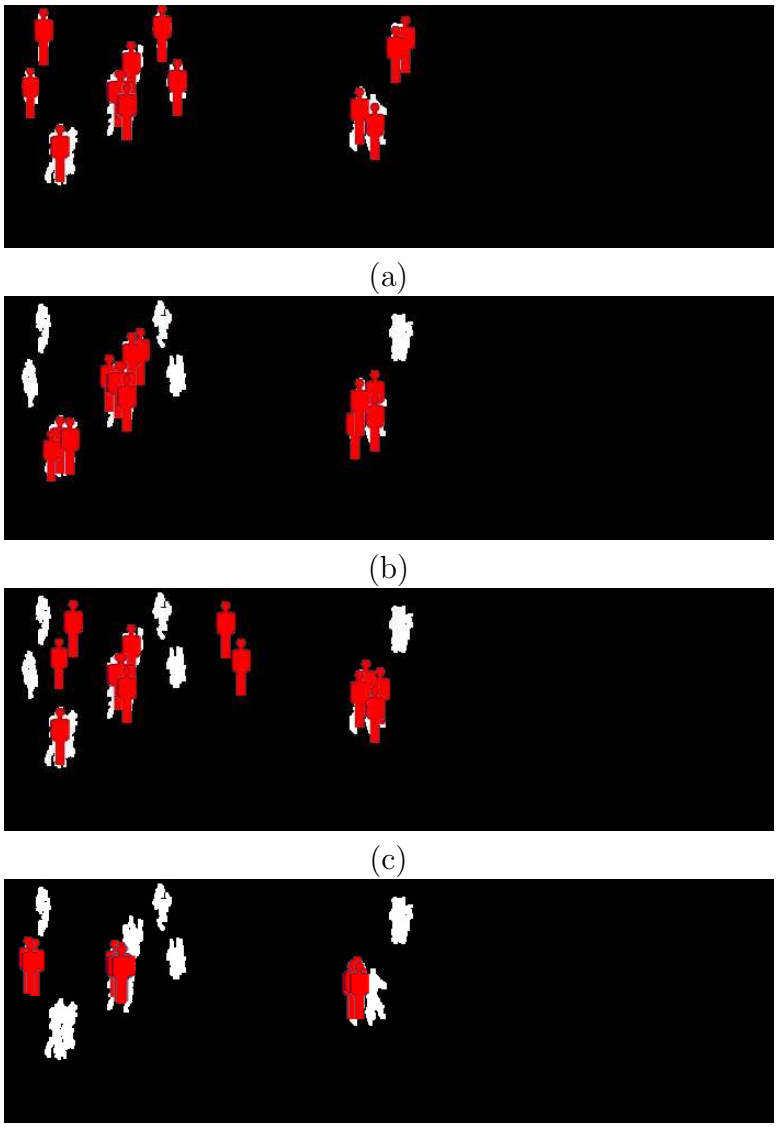
where $p(Y_k|X_k)$ is a likelihood term and $p(X_k)$ is a prior term.

By combining the prior distribution Eq.(5.3) and this hierarchical multi-object joint likelihood, the posterior likelihood can be obtained from Eq.(5.24) as

$$p(X_k|Y_k) \propto L(Y_k|X_k)^{\alpha} \cdot (\Upsilon\Phi)^{\beta} \cdot \prod_{i=1}^{M} e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i}). \tag{5.25}$$

5.6 Conclusion

This chapter describes a modeling framework which is able to deal with the problem of tracking multiple objects. Multiple visual cues such as color and motion cues are integrated seamlessly in this framework to improve the reliability of tracking performance. A hierarchical joint likelihood is used to represent multiple objects. The problem of tracking multiple objects is formulated as the MAP estimation. The methods of solving the MAP estimation are to be presented in the following chapters.



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

Figure 5.4: Likelihood of a configuration. The white masks are the foreground pixels extracted from motion segmentation as introduced in Section 5.4.1. The red human-shape masks are the hypothesized samples as explained in Figure 5.3. (a) highest likelihood configuration (b) low likelihood, too many hypotheses are used to explain the same regions (c) low likelihood, likelihood of individual object is low (d) low likelihood, foreground regions are not covered.

Chapter 6

Attentive Markov Chain Monte Carlo Sampling

6.1 Introduction

A probability distribution is a tool for computing expectations. One useful technique is to represent the posterior by drawing a large number of samples from the distribution. These samples are to be used for estimating the expectation of the posterior. Particle filter is widely used in tracking for low-dimensional problems. Markov chain Monte Carlo (MCMC) [31] has been used to obtain an MAP estimate by random sampling.

The randomization provided by noisy dynamic propagation which is used in the conventional particle filter or MCMC sampling algorithms is effective only when the dimension of state space is relatively low, such that the hypothesized samples

can cover the mode regions fairly densely. As the dimensionality of the state space increases, the state volume increases rapidly with the number of tracked objects. Any hypothesized sample set that is widely distributed to reach nearby maximum has a lower chance of appearing as the modal region. Another difficulty for sample-based Bayesian trackers is that exaggerated random noise is often needed to provide a wider hypothesis coverage to control the accumulated errors caused by the selection of incorrect dynamical model and inaccurate joint-likelihood model. The problem is that even minor errors can significantly pull the state estimation away from its true state, especially when such a situation persists for several sequences. Recovering from such errors requires the state hypotheses to disperse from samples over a wider state space. Boosting the dynamical noise does have undesirable effects of reducing the constraints from past observations and hence increase the uncertainty in each mode. This will cause the state hypotheses to be easily hijacked by neighboring modes or clutter observations. In summary, in multi-modal problems, sample-based Bayesian trackers often trapped by following incorrect local maximum, and some form of explicit global search must be included to rescue them. In this chapter, we model the interaction of multiple objects via a hierarchical joint-likelihood function and propose a novel multiple hypotheses search strategy, attentive MCMC sampling (AMCMCS), which allows the localization of nearby peaks in the high-dimensional joint-likelihood surface.

6.2 Modeling and Estimation

Following the discussion in Section 5.5, the multi-object tracking problem is formulated as an estimation of the posterior probability which is decomposed into a likelihood term and a prior term,

$$P(X_k|Y_k) \propto P(Y_k|X_k)P(X_k). \tag{6.1}$$

The prior probability of the state is assumed to be the product of the prior probabilities of all the objects and is defined as $P(X_k) = \prod_{i=1}^{M} P(H_i)$. The prior probability of an ith object is made up of the prior probability on its hypothesized sample area H_i, $P(H_i) = e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i})$, where the first term penalizes a large overall size (covering all objects) which avoids unnecessary overlapping and the second term penalizes objects with small image sizes since they are more likely to be due to image noise. The value of λ_1 and λ_2 can be adjusted according to the experiment.

Since multiple interacting objects may occlude each other, the image likelihood cannot be decomposed into the product of the image likelihood of individual object hypothesis. We build multi-level image likelihoods hierarchically and use an exclusion principle to compute the joint likelihoods of the interacting objects which has been discussed detailedly in the previous chapter.

6.3 Attentive MCMC sampling algorithm

6.3.1 Introduction of MCMC and Its Applications in Computer Vision

Markov chain Monte Carlo methods [38, 34, 66, 31, 105] are the standard methods for sampling complex distributions. In this method, a Markov chain can be designed to sample a probability distribution. A Markov chain is constructed as the way that the stationary distribution is the target distribution. A new sample can be obtained from an old one, by advancing along the Markov chain.

The Metropolis-Hastings algorithm [73, 44] is a technique for constructing a Markov chain that has a particular desired stationary distribution. Assume that we have a distribution g from which we would like to generate samples. We would like to build a Markov chain which has Q as a stationary distribution.

The algorithm will produce a sequence of samples $X^{(1)}, \ldots, X^{(N)}$, by taking a sample $X^{(j)}$ and proposing a revised version, $X^{(j)*}$. The next element of the sequence $X^{(j+1)}$ will be $X^{(j)*}$ with probability $\alpha_r(X^{(j)}, X^{(j+1)})$; otherwise, it will be $X^{(j)}$. The form of α_r will be given blow.

The proposal process $g(X^{(j)*}|X^{(j)})$ is governed by a proposal distribution which gives the probability of proposing $X^{(j)*}$ from $X^{(j)}$. α_r is defined as

$$a_r = \min(1, \frac{Q(X^{(j)*})g(X^{(j)}|X^{(j)*})}{Q(X^{(j)})g(X^{(j)*}|X^{(j)})}). \tag{6.2}$$

This expression is qualitatively sensible. If the chain is at a point where Q has a very low value and at the new point Q has a very high value and the forward and backward proposal probabilities are about equal, then the new point will be accepted with high probability. If the chain is at a point where Q has a very high value and the proposal process has a high probability of suggesting points with a very low value of Q, it is likely to stay at that point. Finally, if a point which has high value of Q is proposed disproportionately often, it is less likely to be accepted.

A good way of thinking about the Metropolis-Hastings algorithm is that it is an improved version of the hypothesize and test process that is common in computer vision. Metropolis-Hastings proposes various hypotheses which are accepted or rejected. This process yields the sequence $X^{(1)}, \ldots, X^{(N)}$. However, for Metropolis-Hastings the sequence of hypotheses has very significant semantics. Once sufficient iterations have completed, all subsequent $X^{(j)}$ are samples drawn from $Q(X)$.

The advantage of viewing Metropolis-Hastings algorithm as a hypotheses and test process is that it suggests how to build proposal mechanisms. A natural strategy is to take current vision algorithms and make them produce probabilistic outputs. A really attractive feature is that we can use different, possibly incompatible algorithms as distinct sources of proposals, and the samples we obtain represent the posterior incorporating all available measurements. The data-driven techniques [105, 59, 58] can be used to guide the Markov chain search to achieve tremendous speedup in comparison to previous MCMC algorithms [37, 39].

6.3.2 Attentive Sampling Method

The objective is to estimate the multi-object configuration that maximizes the posterior probability which is defined in Eq.(5.25). Markov chain Monte Carlo (MCMC) methods, combined with jump-diffusion dynamics, provide a way to sample the posterior probability in a complex solution space to search for the multiple modes. A Markov chain is defined over the multi-object state space X_k such that the stationary distribution of the chain is exactly the target distribution. For simplification, $Q(X)$ is used to denote $P(X|Y)$. A new sample can be obtained from an old one, by advancing the Markov chain. The Metropolis-Hastings algorithm [2] is a technique for constructing a Markov chain that has a particular desired stationary distribution. This algorithm will produce a sequence of samples $X^{(1)}, \ldots, X^{(N)}$, by taking a sample $X^{(j)}$ and proposing a revised version, $X^{(j)*}$. The next element of the sequence $X^{(j+1)}$ will be $X^{(j)*}$ with probability α_r; otherwise, $X^{(j+1)}$ is kept unchanged as $X^{(j)}$. The proposal distribution g gives the probability of proposing $X^{(j)*}$ from $X^{(j)}$. In our case, we only change the parameter of one object or change one dimension of the multi-object structure at each iteration by sampling from Markov chain dynamics defined in Section 6.3.3. α_r is then defined as

$$a_r = \min(1, \frac{Q(X_k^{(i)*})g(X_{k-1}^{(i)}|X_k^{(i)*})}{Q(X_{k-1}^{(i)})g(X_k^{(i)*}|X_{k-1}^{(i)})}). \tag{6.3}$$

It can be proven that the Markov chain constructed this way has its stationary $Q(\cdot)$, independent of the choice of the proposal probability $g(\cdot)$ and the initial state.

However, the choice of the proposal probability $g(\cdot)$ works as a decisive role in the efficiency of the MCMC. A random proposal probability will lead to slow convergence rate while a proposal probability, designed with domain knowledge and containing information to guide the Markov chain, can traverse the solution space more efficiently.

The objective is to find the configuration that maximizes the multi-modal posterior probability defined in previous section. However, the solution space contains high-dimensional subspaces when the number of tracked objects increases and these may contain many local minima. The difficulty with estimating high dimensional distributions is finding a proposal sampling density that concentrates on the areas where most of their probability mass is concentrated. Effective focusing is the key for high-dimensional state space search. Basically, a reasonably large proportion of the samples should be focused on the current track, while the others should be scattered fairly widely in the region where other good matches may be found. Also, the hyper-volume increases very rapidly with radius for high dimensions; so there is no chance to sample densely enough to provide an effective search coverage with large sample inflation factors. An attentive search factor is used to expand the searching region when the object observation likelihood is low. The regions with low observation likelihood often means occlusions or interactions of multiple objects had occurred. The regions with low observation likelihood deserve more search efforts so as to localize and keep track of those nearby modes. The local attentive

sampling factor with normalization is defined as

$$\varrho_{i,k} = \frac{\log(1/L(y_{j,k}|x_{i,k}))}{\sum_{i=1}^{M} \log(1/L(y_{j,k}|x_{i,k}))} \tag{6.4}$$

which describe the uncertainty of the hypothesized object. The number of samples for estimating the object $X_{i,k}$ at time step k is proportional to $\varrho_{i,k}$. The main insight is that the hypotheses with lower image matching likelihoods are more uncertain in the local search and therefore more samples are needed in these uncertain regions.

6.3.3 Attentive MCMC Scaled Dynamics

Denote the state at iteration $k-1$ as $X_{k-1} = \{M, \{X_{1,k-1}, \ldots, X_{M,k-1}\}\}$. The following Markov chain dynamics are applied to X_{k-1} which results in X^*.

1. Birth and death move: A new hypothesis is generated from motion detection and blob formation. Following the assumption that all objects will appear or disappear near the boundary of the image scene, The domain-specific information is used to model the birth and death move. The birth and death weights are defined as a function of the image coordinates. Higher weights are assigned for objects near the boundary of the image scene. With the appearance of new objects, motion blobs, constructed from pixel-level likelihood and combined with the domain-specific information, are used to hypothesize new object positions. Assuming that a detected blob's position is at $(c1_0, c2_0)$, sample position $(c1_s, c2_s)$ can be constructed in a Gaussian density

$N((c1_0, c2_0), \Sigma(\sigma_{c1}^2, \sigma_{c2}^2))$. The rest of the parameters, $c3_s$ and $c4_s$, are samples from the motion blob information. The color model g_s defined in Eq.(5.13) is constructed from the motion region inside the hypothesized bounding box. A birth hypothesis is generated as $x_{M+1} = \{c1_s, c2_s, c3_s, c4_s\}$ and X is updated as $X^* = \{M+1, \{x_1, \ldots, x_M, x_{M+1}\}\}$. The procedure of death move is similar to birth move. $X^* = \{M-1, \{x_1, \ldots, x_{M-1}\} - \{x_s\}\}$.

2. Propagation move: The frequency of selecting object i is proportional to the attentive sampling factor ϱ. The parameters of one hypothesized object are updated by its temporal evolution Eq.(5.1) with incremental scaled Gaussian noise. $x_i^* = f_i(x_i, q_i)$, $q_i = N(\varrho\Sigma_i)$.

3. Trajectory prediction move: The frequency of selecting object i is proportional to the attentive sampling factor ϱ. The parameters of a hypothesized object is updated in the direction of the object's trajectory plus attentive scaled stochastic noise.

4. Interchange move: The frequency of selecting object i is proportional to the attentive sampling factor ϱ. The model x_i is randomly switched with one of the neighboring objects.

The first two are referred to as jump dynamics and the rest are diffusion dynamics. In various diffusion moves, the frequency of selecting object i to move is proportional to the attentive sampling factor ϱ_i and the inflation noise is also scaled with ϱ_i. Multiple ways of generating samples increase the robustness of the tracker. Attentive

Table 6.1: The iteration steps of attentive MCMC sampling algorithm

- Initialize the MCMC sampler for time k by drawing $X_k^{(s)}$ from temporal evolution Eq.(5.1).
- Calculate attentive sampling factor with Eq.(6.4)
- Perform Metropolis-Hastings iterations to obtain N_s samples from the posterior Eq.(5.25). Discard the first N_d samples to account for sampler burn-in.
  - ◇ Generate hypotheses according to Section 6.3.3
  - ◇ Compute the acceptance ratio $a_r = \min(1, \frac{Q(X_k^{(i)*})g(X_{k-1}^{(i)}|X_k^{(i)*})}{Q(X_{k-1}^{(i)})g(X_k^{(i)*}|X_{k-1}^{(i)})})$.
  - ◇ If $a_r \geq 1$ then accept $X^{(i)*}$, update the ith object in X_k to $X^{(i)*}$. Otherwise, we leave the ith object parameters in X_k unchanged.
- The sample set $\{X_k^{(s)}\}_{s=N_d}^{N_s}$ at time k represents the estimated multi-modal joint posterior probability.

sampling guarantees that the uncertain regions receive increased attention and effort in the search.

6.3.4 Algorithm Summarization

The iteration steps of attentive MCMC sampling algorithm are shown in Table 6.1.

6.4 Performance Evaluation of Multi-object Tracking System

For the experiments, a multi-object model is used which incorporates a motion and color based image matching likelihoods function as explained in Section 2. Figure

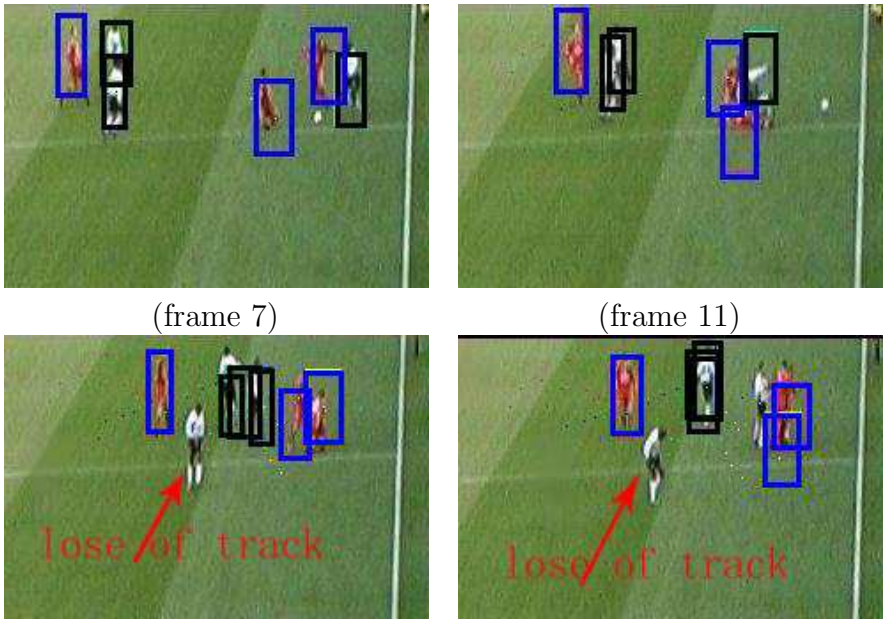
(frame 7) (frame 26)

(frame 31) (frame 162)

Figure 6.1: Results of tracking multi-person interaction.

6.1 shows the results of tracking the interactions of multiple persons. In this test sequence, the images are 360 by 280 24-bit RGB pixels. The attentive MCMC tracker was executed with 30 samples. The color models of three persons are initialized and assigned a label when they enter the scene separately. The three persons walk

interactively and occlude each other as shown in Frame 26. The proposed attentive sampling algorithm is able to keep track and recover the correct label after occlusion which is shown in Frame 31. When person #1 leaves the scene, the algorithm performs a death move which can be observed from Frame 162.

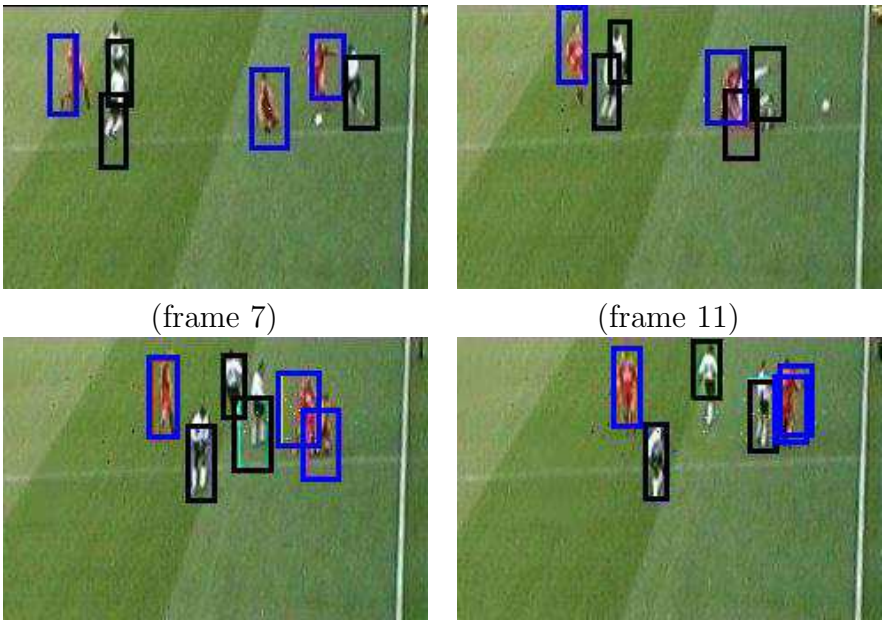


(frame 7) (frame 11)

(frame 20) (frame 25)

Figure 6.2: The results of AMCMCS tracker with independent likelihoods function. When objects come close and occlude, samples are easily trapped and contained by the most likely object.

A more challenging sequence where six soccer players approach, merge, occlude each other and split is shown in Figure 6.2 and Figure 6.3. The attentive MCMC tracker is run with 100 samples and the first 50% samples are discarded to let the sampler burn in. We compare the tracking results with independent likelihoods function and our hierarchical joint likelihoods function and show them in Figure 6.2 and Figure 6.3.



(frame 7) (frame 11)

(frame 20) (frame 25)

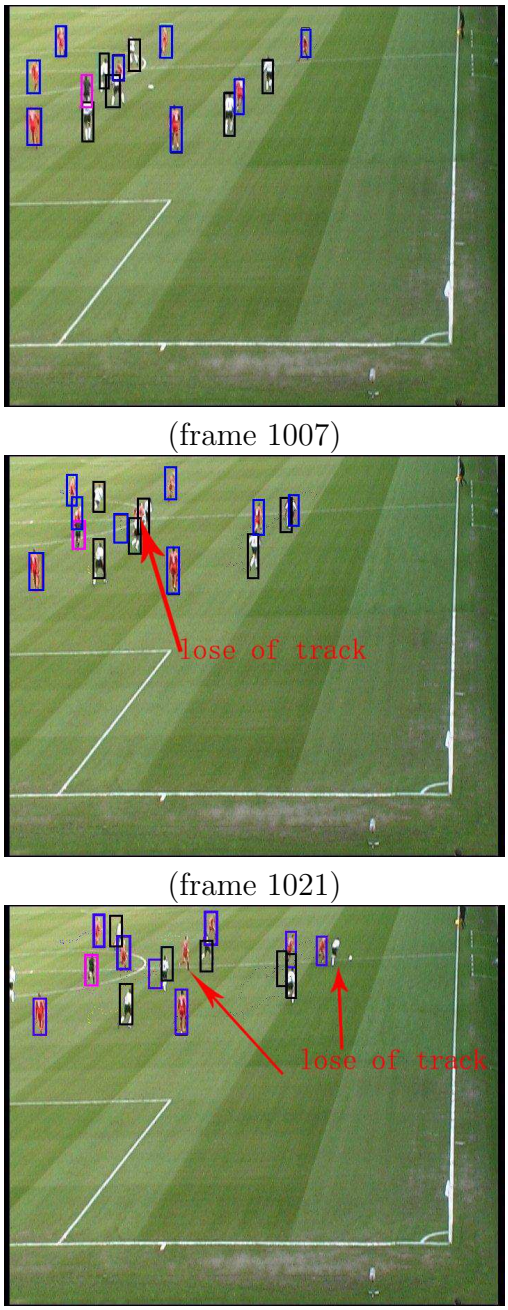
Figure 6.3: The results of AMCMCS tracker with hierarchical joint likelihood model. With the exclusion principle, the sampler can keep track of all the objects.

The evaluation results of outdoor soccer players tracking with ground truth are given in Figure 6.4 and Figure 6.5. The images are 720 by 576 24-bit RGB pixels from PETS-ICVS03 [85] database. The number of tracked players are up to 16. Figure 6.5 shows the tracking results with our attentive MCMC tracker, 100 samples. The players are classified into three types, based on color, which represent the two different team members and the referee. The attentive MCMC sampler is able to track of all interacting players with only 100 samples. Qualitative comparison results of our attentive MCMC sampler and the traditional MCMC sampler are shown in Figure 6.6. The tracker detection rate (TDR) characterizes the tracking performance of the object tracking algorithm and is defined as $TDR = \frac{Total\ True\ Positives}{Total\ Number\ of\ Ground\ Truth\ Points}$.

The track detection rate indicates the tracking completeness of a specific ground truth track. Figure 6.7 compares the performance with varying number of runs. We can conclude that the attentive MCMC sampler performs much better than the traditional MCMC sampler especially when the tracking persists for a long time.

6.5 Conclusion

This chapter presents a new method for localizing and tracking multiple interacting objects, based on attentive MCMC sampling on a hierarchical joint likelihoods surface. Multiple modes are localized and maintained using a novel attentive sampling which is a high-dimensional search method based on sampling incrementally in uncertain regions. Our experiments on real sequences show that this is more effective than the traditional MCMC sampler based on random inflated noise, because it congregates the samples to the most uncertain region. This strategy makes searching more effectively in high-dimensional multi-modal distributions.

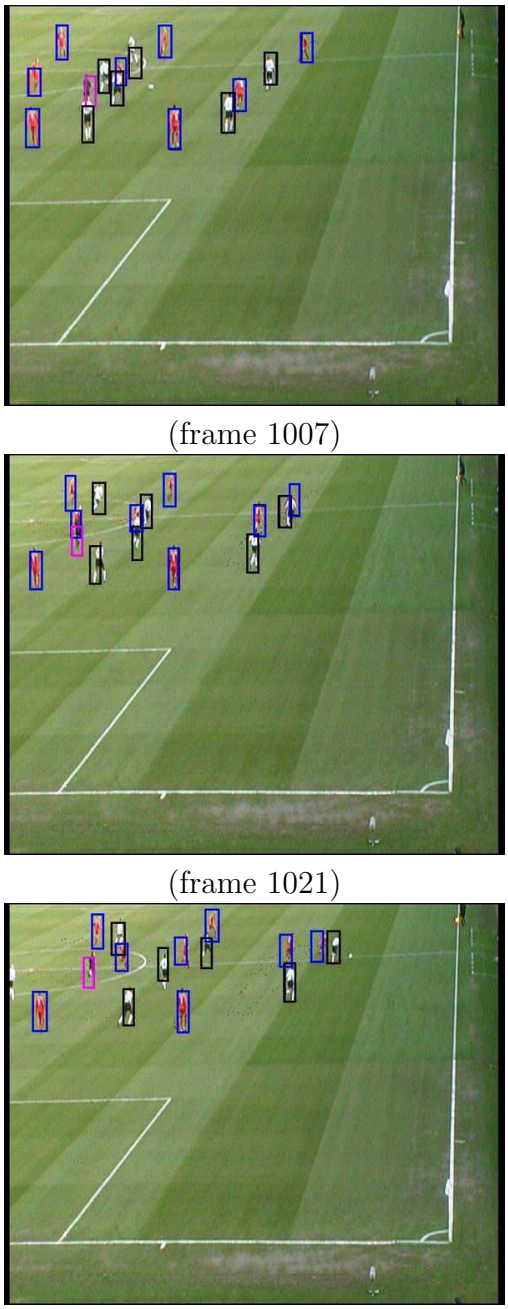


(frame 1007)

(frame 1021)

(frame 1034)

Figure 6.4: Comparison results are demonstrated using frames 1007, 1021, 1034. The figure shows the results of the traditional MCMC sampling. The tracks of two targets were lost in frame 1034.



(frame 1007)

(frame 1021)

(frame 1034)

Figure 6.5: Comparison results are demonstrated using frames 1007, 1021, 1034. This figure shows the results of our attentive MCMC sampling. The AMCMCS tracker is able to keep tracking all the objects because of its ability of extensively exploring the state space.

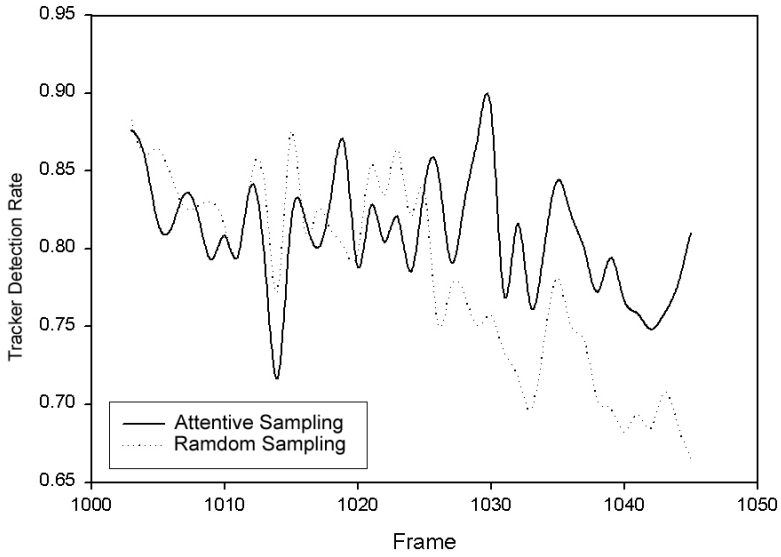


Figure 6.6: Qualitative comparison of attentive MCMC sampler and random MCMC sampler in the soccer players tracking sequence which is shown in Figure 6.5. Both are with 100 samples, with 50% discarded for burn in. The attentive MCMC sampler performs better than the traditional MCMC sampler especially when the tracking persists for a long time because our AM-CMCS can recover from accumulated errors through a more global search.

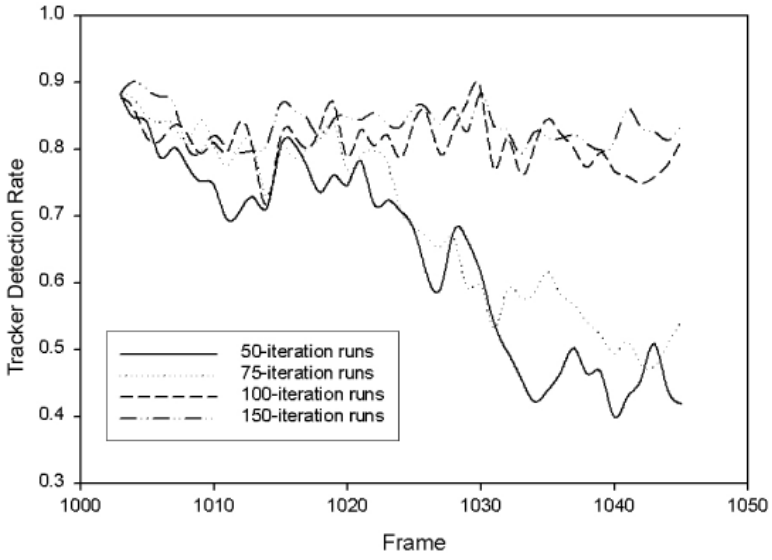


Figure 6.7: Qualitative comparison of attentive MCMC tracker with different number of samples, with 50% samples discarded for burn in.

Chapter 7

Color-Spatiotemporal MRF-Based MCMC Sampling

Chapter 5 is concerned about the modeling aspects of multiple object tracking. In Chapter 6, attentive MCMC sampling method is presented to estimate the optimal states of multiple objects. This chapter presents the method of maintaining the track of multiple interacting objects. During the interactions, targets are influenced by the proximity and/or behavior of other targets. Such interactions cause the data association problem for traditional approaches. Dealing appropriately with this problem has important applications for tracking of sports players, and is generally applicable to any situation where many interacting objects need to be tracked over time.

The basic assumption on which all established data-association methods rely on is that targets maintain their behavior before and after the objects visually merge.

However, consider the video example shown in Figure 7.1, which shows various multiple interacting objects. In these cases, objects do not behave independently. When the objects encounter each other, some amount of interactions and overlaps occur, and the behavior of involved objects changes significantly before and after interactions.

Figure 7.1: Examples of various multiple interacting objects. In these cases, objects do not behave independently. When the objects encounter each other, some amount of interactions and overlaps occur, and the behavior of involved objects changes significantly before and after interactions.

For the purpose of maintaining the tracking under the above situations, a multi-object tracking method which is robust against severe mutual occlusions is required. The major interest in this chapter is to track objects against mutual occlusions which usually occur during interactions. Because objects appear in various kinds of forms and they move in random manners during interactions, occlusions and clutters occur in complex manners during interactions. In contrast to traditional methods, the proposed approach in this chapter relies on the modeling of explicit interaction, which is able to adequately describe object behavior throughout the interaction process even when experiencing severe mutual occlusions and overlapping.

The major goal is to track individual nonrigid objects robustly against the occlu-

sion and clutter effects which usually occur during interactions. Objects travelling through interactions are moving in various directions. Various parts of these objects may either be occluded by, or themselves occlude, other objects. A tracking algorithm utilizing the color-spatiotemporal Markov random field (CSTMRF) model is developed in this chapter in order to overcome such situations.

7.1 Color-Spatiotemporal MRF Model

We model the interaction among objects using a graph-based MRF [63, 110, 104, 54, 64, 55]. A prior model should properly define the interactions between the multiple objects. MRF is well suited for that purpose. An MRF is a graph with undirected edges between nodes where the joint probability is factored as a product of local potential functions at each node, and interactions are defined on neighborhood cliques.

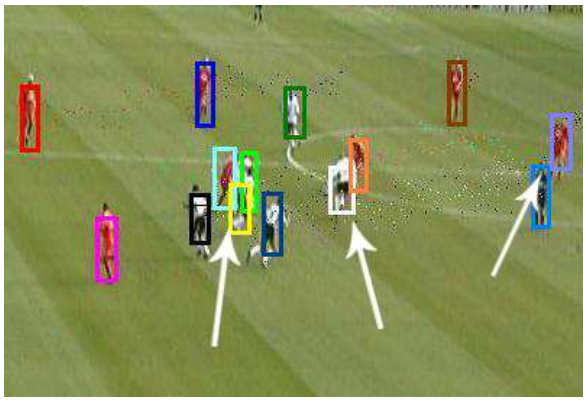


Figure 7.2: An example of tracking multiple interacting soccer players. The places where interactions occur are indicated using white arrows.

The interaction potentials of MRF give the possibility of easily specifying the

domain knowledge governing the joint behavior of interacting objects. Our approach to addressing tracking failures resulting from interactions and occlusions is to introduce a graphic model, based on Markov random fields. Soccer players tracking domain contains severe mutual occlusions and complex interactions which are shown in Figure 7.2. An example MRF for our test domain is illustrated in Figure 7.3. The interaction potentials of the MRF can provide the possibility of governing multiple interacting objects. At the same time, the absence of an edge in the MRF indicates there are no interactions.

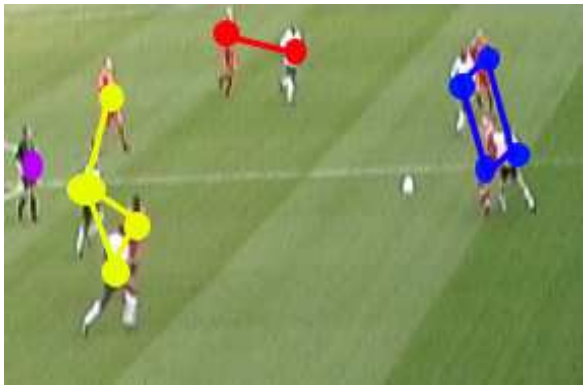


Figure 7.3: To model the interaction, a Markov random field prior model is constructed. An example is shown here for soccer players' interaction. Players which are close to each other are linked by an edge, denoting that there are occluding interactions.

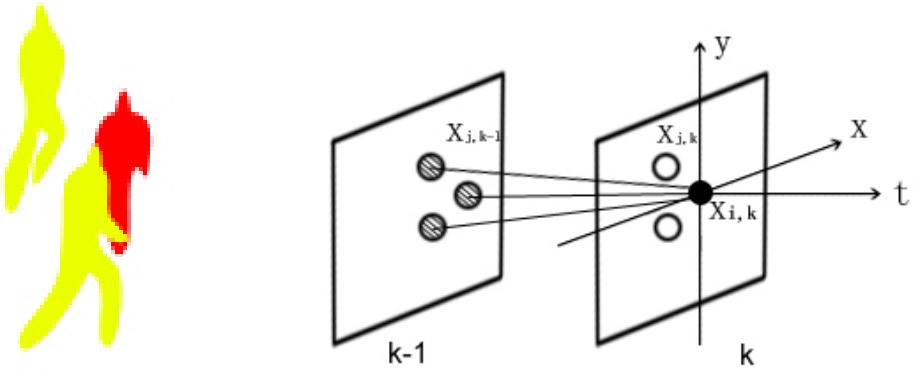
The multiple objects can be represented by a set of nodes in a connected graph. Formally stated, the graph model $G = (m, \eta(m))$ is an undirected graph such that

- $M_k = \{m_{1,k} \ldots m_{j,k} \ldots m_{N,k}\}$ is the set of nodes in the graph at time step k, where node $m_{j,k}$ corresponds to object $x_{j,k}$.

- $(m_{i,k}, m_{j,k}) \in \eta(m)$ if the corresponding objects $x_{i,k}$ and $x_{j,k}$ are spatially

adjacent.

The label field is supposed to verify the main MRF property related to that neighborhood, namely the node m_i of the current object x_i depends only on its neighbors $m_j \in \eta(m)$.



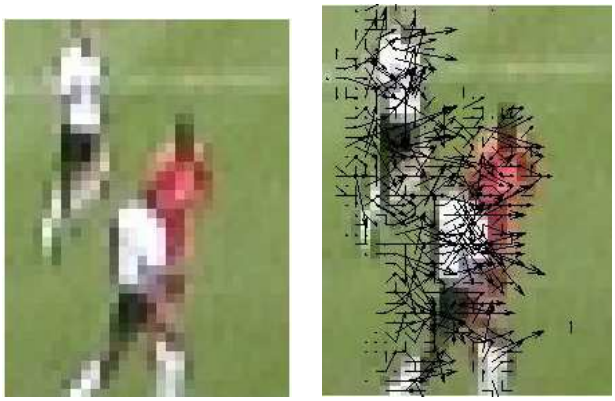
(a) (b)

Figure 7.4: (a) Spatial neighbors in image. (b) Spatiotemporal neighborhood η_m: in black, the current object $x_{i,k}$; in white, the spatial neighbors $x_{j,k}$; in gray, the temporal neighbors $x_{j,k-1}$.

Usually, an MRF model handles only spatial $x - y$ directional distribution. We extend it to be able to handle not only spatial distribution but also temporal distribution. An image sequence has correlations at each object between consecutive images along the time axis. Each object has its own temporal behavior consistency between consecutive images. Our MRF also consider this temporal correlation.

Optical flow is used to provide temporal correlation of interacting objects. Optical flow is computed at each frame using the Lucas-Kanade [68] algorithm (see Figure 7.5). The optical flow vector field F is first split into two scalar fields corresponding to the horizontal and vertical components of the flow, F_x and F_y, each of

which is then half-rectified into four non-negative channels F_x^+,F_x^-, F_y^+, F_y^-. These are each blurred with a Gaussian and normalized to obtain the final four temporal motion features, $\hat{F}b_x^+$, $\hat{F}b_x^-$, $\hat{F}b_y^+$, $\hat{F}b_y^-$. These temporal motion features are shown in Figure 7.6.



(original image) (optical flow $F_{x,y}$)

Figure 7.5: Constructing the optical flow descriptor.

These temporal motion features are compared using normalized correlation. The four temporal motion features for frame k of sequence S are $s_{1,k}$, $s_{2,k}$, $s_{3,k}$ and $s_{4,k}$. Matrix S_1 is defined as the vector of $s_1^{'}s$ for each frame stringed as column vectors, and the other three features are similarly formulated. The temporal consistence of two consecutive frames is defined as the conjunctive frame similarity:

$$C(k, k-1) = (S_{1,k})^T S_{1,k-1} + (S_{2,k})^T S_{2,k-1} + (S_{3,k})^T S_{3,k-1} + (S_{4,k})^T S_{4,k-1}. \quad (7.1)$$

Given this neighborhood structure as shown in Figure 7.4, the prior energy is expressed as a sum of potential functions V that model color-spatiotemporal inter-

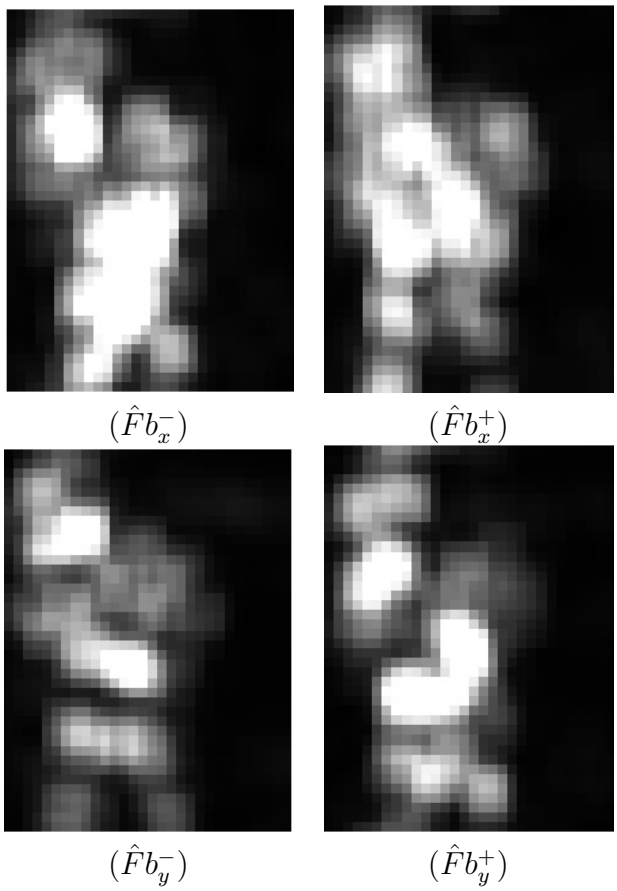


Figure 7.6: Constructing temporal motion descriptor. The optical flow vector field F is first split into two scalar fields corresponding to the horizontal and vertical components of the flow, F_x and F_y, each of which is then half-rectified into four non-negative channels F_x^+, F_x^-, F_y^+, F_y^-. These are each blurred with a Gaussian and normalized to obtain the final four temporal motion features $\hat{F}b_x^+$, $\hat{F}b_x^-$, $\hat{F}b_y^+$, $\hat{F}b_y^-$

actions within the neighborhood $\eta(m)$ of node m_i

$$p(X) = \frac{1}{Z} e^{-U(X)} \tag{7.2}$$

where $U(X)$, the energy function, is given by

$$U(X) = \sum_{c \in C} V_c(X) \tag{7.3}$$

and the term Z, often called the partition function, is given by

$$Z = \sum_{X} e^{-U(X)}. \tag{7.4}$$

C is the set of all cliques in the graph G. The partition function Z is simply a normalizing constant, so that the sum of the probabilities of all realizations X, add to unity. The functions $V_c(X)$ are called the clique potentials. The only condition on $V_c(X)$ is that it depends on the nodes within the cliques c. Clique functions provide a mechanism to express soft constraints between neighboring labels.

We define the energy function U_p as a composition of two terms

$$U_p = \sum_{j \in \eta(i)} V_C(x_{i,k}, x_{j,k}) + \sum_{j \in \eta(i)} V_T(x_{i,k}, x_{j,k}). \tag{7.5}$$

- The first term $V_C(x_{i,k}, x_{j,k})$ is the spatial color term.

$$V_C(x_{i,k}, x_{j,k}) = \prod_{i=1}^{M} L(y_{j,k}|x_{i,k}) \qquad if \quad K_{j,k} = i. \tag{7.6}$$

- The second term $V_T(x_{i,k}, x_{j,k})$ is the temporal motion term

$$V_T(x_{i,k}, x_{j,k}) = \lambda_a C(S_{i,k}, S_{i,k-1}) - \Sigma_{j=1}^{N_r} \lambda_b C(S_{i,k}, S_{j,k-1}). \tag{7.7}$$

The color-spatiotemporal prior probability in Eq.(7.2) follows a Gibbs distribution

$$p(X) = \frac{e^{-U_p(X)}}{Z_p} \tag{7.8}$$

where $Z_p = \sum_X e^{-U_p(X)}$.

Furthermore, MRF used in conjunction with statistical criteria for optimization, such as maximum a posterior (MAP), are used to formulate an objective function in terms of an MAP optimization principle. A probabilistic interpretation and optimal estimates of the model parameters can be achieved by maximizing the posterior probability. The optimal estimator is the maximum a posterior (MAP) estimator

$$\hat{X}_k = \arg\max_{X_k} p(X_k|Y_k). \tag{7.9}$$

According to the Bayes rule, we know that

$$p(X_k|Y_k) \propto p(Y_k|X_k)p(X_k). \tag{7.10}$$

where $p(Y_k|X_k)$ is a likelihood term and $p(X_k)$ is a prior term.

$$p(Y_k|X_k) = \frac{1}{Z} e^{-U(X_k)} \cdot (\Upsilon\Phi)^{\beta} \tag{7.11}$$

By combining the prior distribution and this hierarchical multi-object joint like-

lihood, the posterior likelihood can be obtained from Eq.(7.10) as

$$p(X_k|Y_k) \propto \frac{1}{Z} e^{-U(X_k)} \cdot (\Upsilon\Phi)^{\beta} \cdot \prod_{i=1}^{M} e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i}). \tag{7.12}$$

7.2 Tracking using CSTMRF-based MCMC Sampling

Our state estimation technique combines CSTMRF model with attentive MCMC sampling method.

7.2.1 CSTMRF-based Attentive MCMC Sampling

The objective is to estimate the multi-object configuration that maximizes the posterior probability which is defined in Eq.(7.12). Markov chain Monte Carlo (MCMC) methods, combined with jump-diffusion dynamics, provide a way to sample the posterior probability in a complex solution space to search for the multiple modes.

7.2.2 Algorithm Summarization

The iteration steps of CSTMRF-based attentive MCMC sampling algorithm are shown in Table 7.1.

Table 7.1: The iteration steps of CSTMRF-based attentive MCMC sampling algorithm

- Initialize the MCMC sampler for time k by drawing $X_k^{(s)}$ from temporal evolution Eq.(5.1).
- Calculate attentive sampling factor with Eq.(6.4)
- Perform Metropolis-Hastings iterations to obtain N_s samples from the posterior Eq.(7.12). Discard the first N_d samples to account for sampler burn-in.
  - ◇ Generate hypotheses according to Section 6.3.3
  - ◇ CSTMRF model
  - ◇ $Q(X_k^{(i)}) = p(X_k^{(i)}|Y_k) \propto \frac{1}{Z} e^{-U(X_k^{(i)})} \cdot (\Upsilon\Phi)^{\beta} \cdot e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i})$
  - ◇ $Q(X_k^{(i)*}) = p(X_k^{(i)*}|Y_k) \propto \frac{1}{Z} e^{-U(X_k^{(i)*})} \cdot (\Upsilon\Phi)^{\beta} \cdot e^{-\lambda_1 H_i}(1 - e^{-\lambda_2 H_i})$
  - ◇ Compute the acceptance ratio $a_r = \min(1, \frac{Q(X_k^{(i)*}) g(X_{k-1}^{(i)} | X_k^{(i)*})}{Q(X_{k-1}^{(i)}) g(X_k^{(i)*} | X_{k-1}^{(i)})})$.
  - ◇ If $a_r \geq 1$ then accept $X^{(i)*}$, update the ith object in X_k to $X^{(i)*}$. Otherwise, we leave the ith object parameters in X_k unchanged.
- The sample set $\{X_k^{(s)}\}_{s=N_d}^{N_s}$ at time k represents the estimated multi-modal joint posterior probability.

7.3 Performance Evaluation of Tracking Systems

There are many ways in which the performance of a video tracking system can be evaluated [10, 49, 78]. For evaluating the performance of multi-object tracker, the aim is to evaluate how well a tracker is able to determine the position of each object.

The proposed approach is evaluated by tracking through several video sequences of sports players, and present both quantitative results as well as a graphical comparison of the different tracker methodologies. The test sequences consist of basketball, hockey and soccer match videos which involve extensive interactions and occlusions of multiple players. The frame rate was 10Hz, and images are 720 by 576 24-bit RGB pixels.

7.3.1 Metrics and Statistics for Trajectory Comparison

A trajectory is a sequence of positions over time. The general definition of a trajectory $\hat{T}$ is a sequence of positions $(\hat{x}_k, \hat{y}_k)$ and corresponding times k:

$$\hat{T} = \{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_k, \hat{y}_k)\}. \tag{7.13}$$

To evaluate the performance of the tracker, metrics comparing two trajectories need to be devised. We have two trajectories $\hat{T}_A$ and $\hat{T}_B$ which represent the trajectory of a target from the tracker, and the ground truth trajectory which is usually marked manually from footage. Metrics comparing the trajectories allow us to identify how similar, or how different, they are.

Consider two trajectories composed of 2D positions in an image sequence. Let positions on trajectory $\hat{T}_A$ be $(\hat{x}_i, \hat{y}_i)$, and on trajectory $\hat{T}_B$ be $(\hat{x}'_i, \hat{y}'_i)$. The object tracking error (OTE) indicates the mean distance between the ground truth and the

tracked object trajectory.

$$OTE = \frac{1}{N_{AB}} \sum \sqrt{(\hat{x}_i - \hat{x}_i')^2 + (\hat{y}_i - \hat{y}_i')^2} \tag{7.14}$$

where N_{AB} is the total number of frames in the trajectory.

The reliability of the video tracking algorithm can be associated with a number of criteria: the frequency and complexity of dynamic occlusions, the overlapping of the objects, the distinctiveness of the targets, and the maximum number of tracked objects. We use a measure for estimating the perceptual complexity of the sequence based on the occurrence and duration of dynamic occlusions, since this is the event most likely to cause the tracking algorithm to fail. A true positive is defined as a ground truth point that is located within the bounding box of an object detected and tracked by the tracking algorithm. A false negative is a ground truth point that is not located within the bounding box of any object tracked by the tracker. These conditions are illustrated in Figure 7.7. In Figure 7.7 the player in the middle of the image has not been tracked correctly. The ground truth for the player is classified as false positive. The other two players are tracked correctly and counted as true positives.

The following metrics are used to characterize the tracking performance

$$TDR = \frac{Total\ True\ Positives}{Total\ Number\ of\ Ground\ Truth\ Points} \tag{7.15}$$

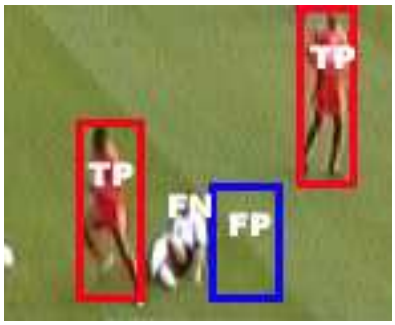


Figure 7.7: Accuracy definition.

$$FAR = \frac{Total\ False\ Positives}{Total\ True\ Positives + Total\ False\ Positives}. \tag{7.16}$$

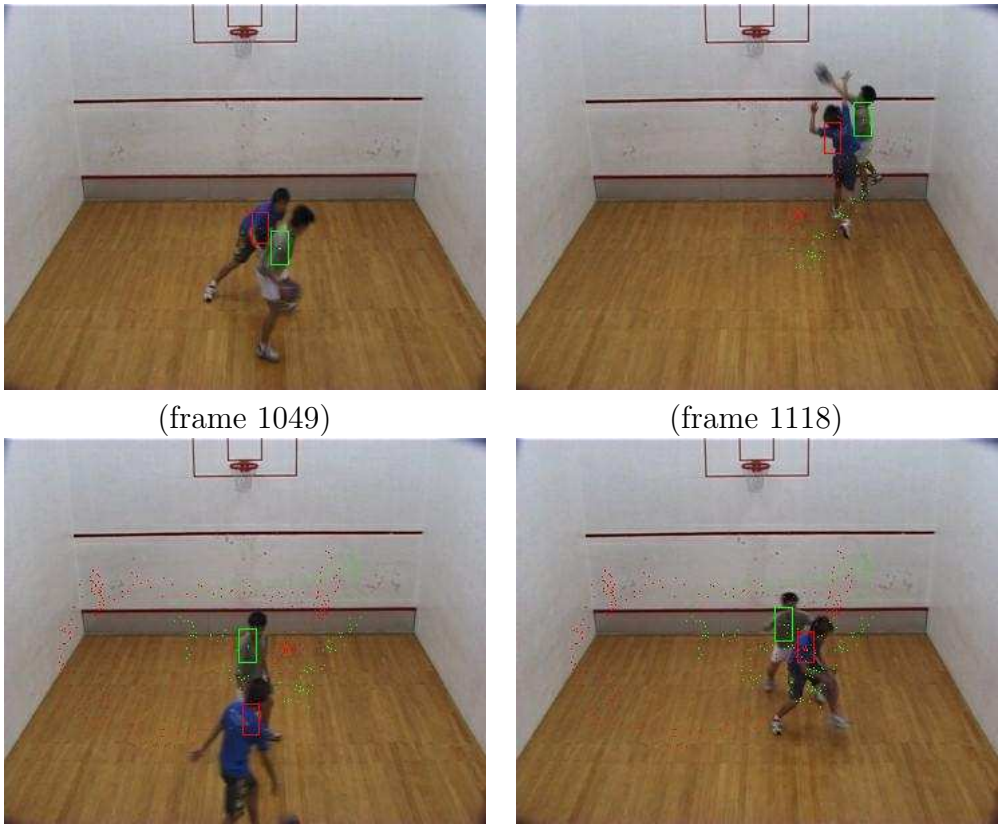
The tracker detection rate (TDR) and false alarm rate (FAR) characterize the tracking performance of the object tracking algorithm. The track detection rate indicates the tracking completeness of a specific ground truth track.

7.3.2 Evaluation, Results and Discussion

Figure 7.8 shows the result of an indoor two-person basketball sequence. The inter-occlusion is under extended long duration in this sequence.

Figure 7.9 shows the results of hockey players tracking with our CSTMRF algorithm. All the interacting players are successfully tracked with CSTMRF.

Performance evaluation is performed on a PETS03 soccer tracking sequence [85]. Figure 7.10 shows the comparison results on a soccer match video sequence, "seq1". Figure 7.11 and Figure 7.12 compare the results of our CSTMRF MCMC sampling and conventional MCMC sampling on a soccer match video sequence, "seq2", which



(frame 1049) (frame 1118)

(frame 1284) (frame 1318)

Figure 7.8: Two-person basketball tracking result.

is taken from a different view of "seq1". The quantitative comparisons of tracker detection rate (TDR), false alarm rate (FAR) and object tracking error (OTE) are shown in Figure 7.13, 7.14 and 7.15.

The total computation time for processing a frame is the product of the number of iterations and the computation time of each iteration. The scenes containing less objects and less occlusions require less number of iterations. The computation time per iteration is affected by the complexity of the image, that is, the number of objects and the interaction of objects. The tracking system can achieve around 10Hz

speed when a 150-iteration is run on the above tracking sequences which contain 10-20 interacting objects, with a Pentium IV 2.4G Hz PC and C++ un-optimized code.

From the qualitative comparison in Figure 7.11 and Figure 7.12 and quantitative comparison in Figure 7.13, Figure 7.14 and Figure 7.15 we draw the following discussions. The most difficult problem associated with multiple object tracking is the occlusion among interacting objects. Color-spatiotemporal MRF model is developed in order to solve this problem. This algorithm models a tracking problem by determining the state of each object in an image and its transit, and how such states transit along both image axes as well as the time axes. In the example of soccer players tracking, soccer players are of various shapes and they move in random manners and therefore leading to occlusions during interactions. During the tracking of multiple soccer players, there are two types of occlusions. One is occlusion among persons with different color uniforms. The other one is occlusion among persons with similar color uniforms. In the case of occlusion among persons with different color uniform, color offers a significantly discriminative cue to infer the occlusion. The hypothesis with correct color pattern and correct spatial configuration gets higher probability. In the case of occlusion among persons with similar color uniform, color cue may not be reliable in this case. Temporal consistence works as a major cue to provide discriminative power. MCMC method samples over joint state space and maximizes the posterior probability to maintain the tracking of multiple interacting objects under occlusion situation. The assumption that either

color, spatial or temporal cues would provide discriminative power at a time is valid at most situations. However, in case the assumption is violated, for example, the players with same color uniform undergoes complete occlusion and moving in the same direction, the tracker would fail to give correct label because there is no visual discriminative power available any more.

7.4 Remarks

7.4.1 Comparison with Particle Filter-Based Tracking Algorithm

- Particle filter is known to suffer from the dimensionality problem due to the poor scalability of non-parametric approaches in high dimension.

- The sequential nature of MCMC can make more in-depth analysis of the solution distribution. The sampling methods used in MCMC may be more flexible than the one-pass sampling in particle filter. Various data-driven techniques can be easily incorporated in MCMC framework.

- One important shortcoming of particle filters, and Monte Carlo methods in general, is that these methods show inability in consistently maintaining the multi-modality in the target distribution. Multiple modes arise if there is ambiguity about the object state due to insufficient measurements of clutters, or measurements from multiple objects. In tracking multiple objects, it is

required to track all the objects present. In a particle filter implementation, however, it often happens that all the particles quickly merge to one of the most significant modes, subsequently discarding all the other modes. In contrast, our approach uses a Color-Spatiotemporal MRF model to explicitly model the interaction of multiple objects. The multi-modality of state is explicitly modeled and well maintained.

7.5 Conclusions

We have presented an approach to track multiple interacting objects. Generally, it is difficult to track multiple objects without missing tracking them. In particular, maintaining the track of multiple objects is very difficult during interactions where various kinds of mutual occlusions and clutters occur. A large number of interacting objects and frequent occlusions make the problem even worse. In order to achieve robust tracking under severe mutual occlusion, we propose a Color Spatiotemporal MRF (CSTMRF) model to reason the interactions among objects. A stochastic search algorithm based on attentive MCMC sampling is used to estimate the state. Experimental results successfully demonstrate the ability to track multiple objects during interactions with occlusions. The real experiments of various sports matches showed a good performance when multiple players go through interaction and occlusions.

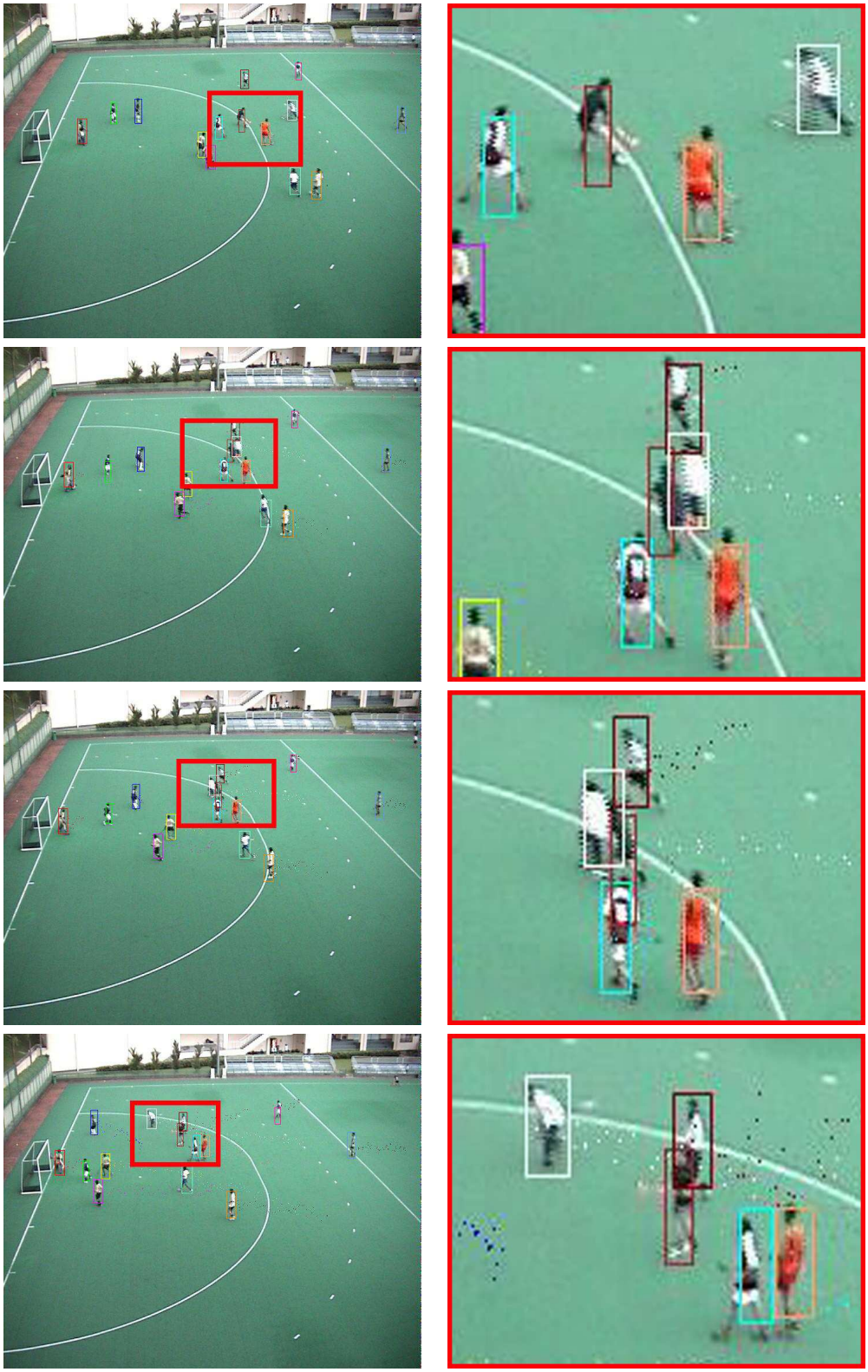
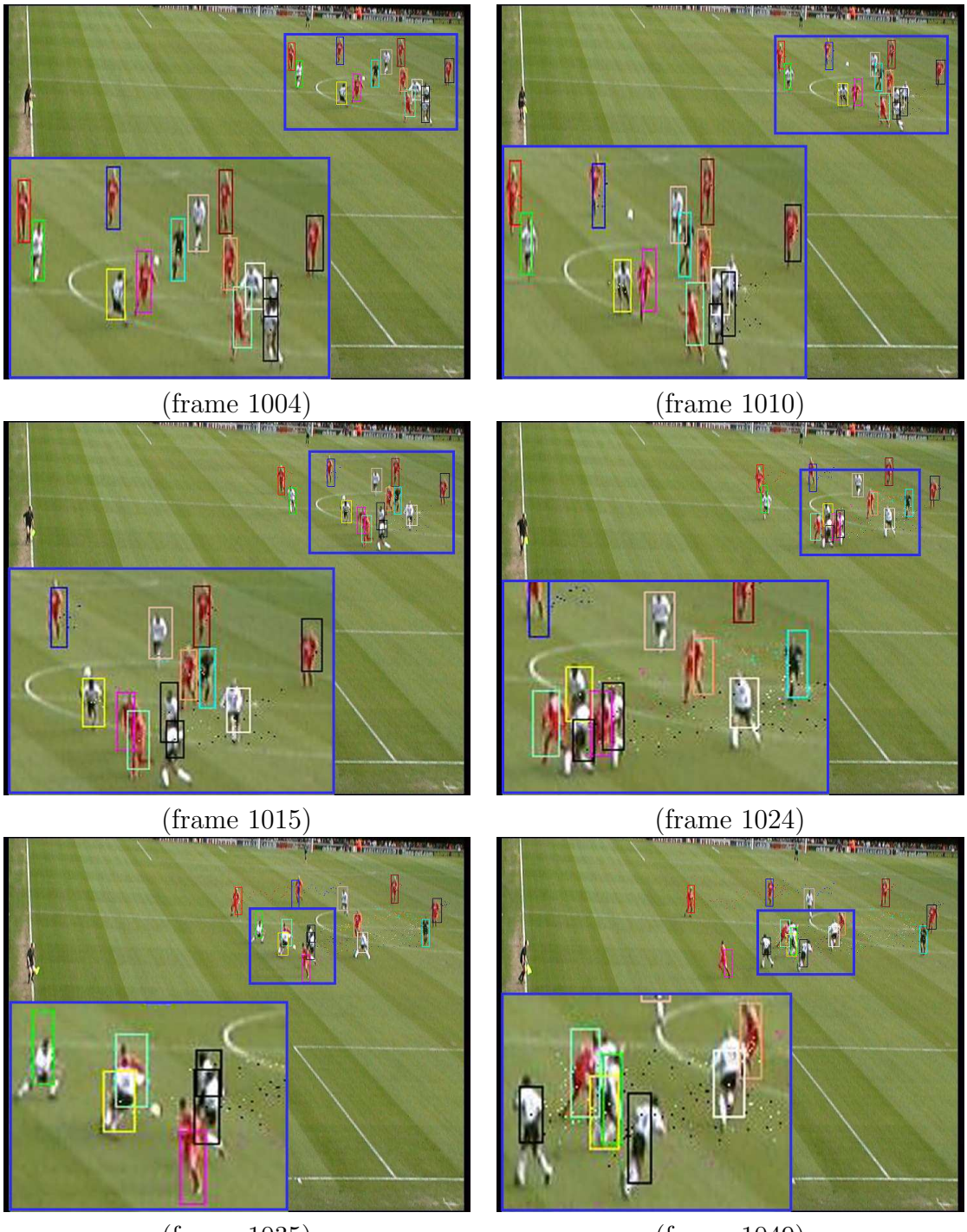


Figure 7.9: Hockey player tracking result.

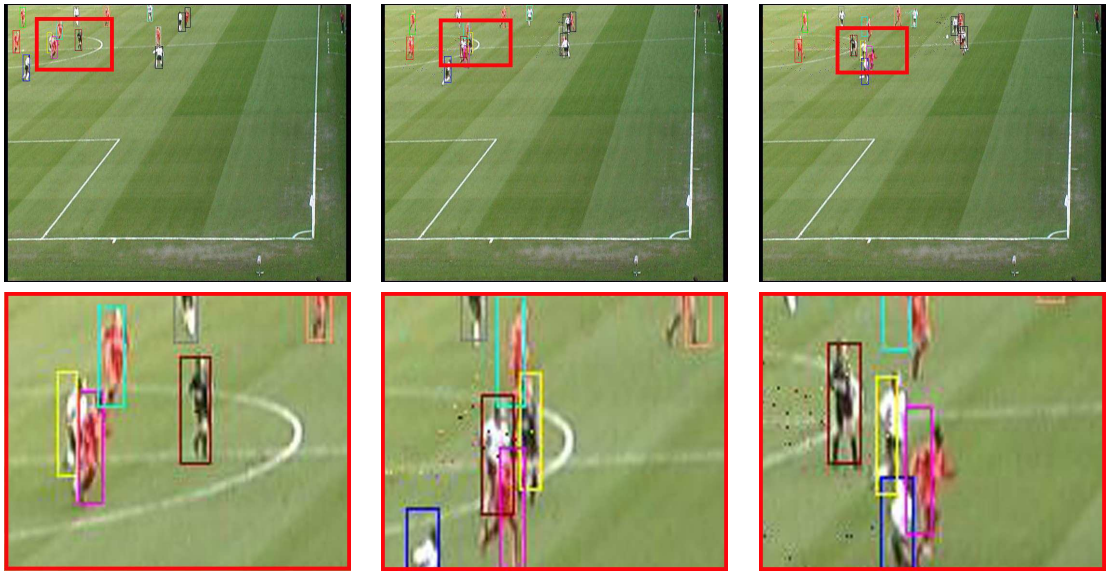


(frame 1004) (frame 1010)

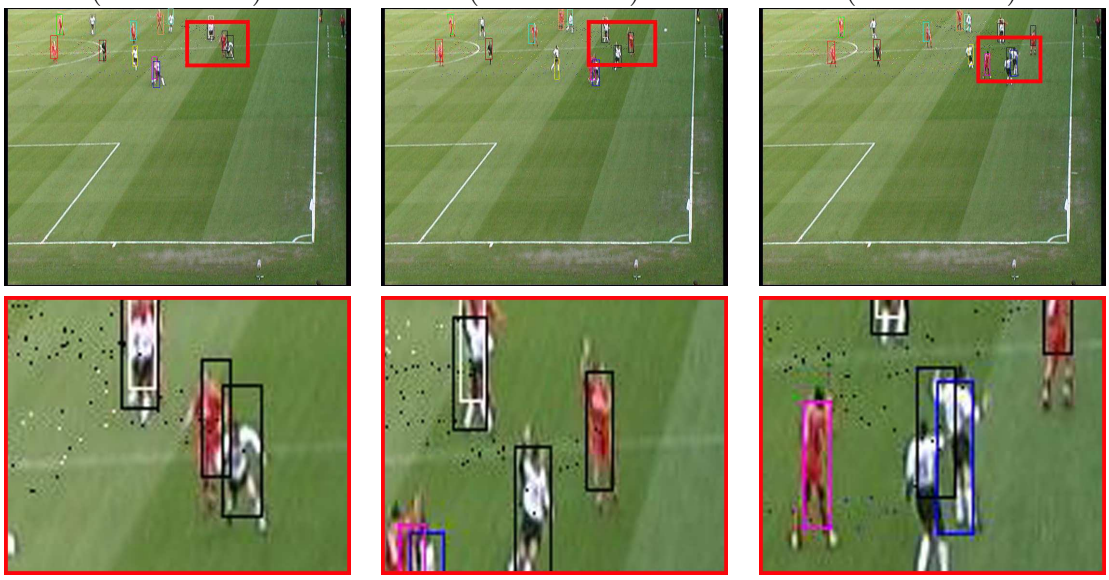
(frame 1015) (frame 1024)

(frame 1035) (frame 1049)

Figure 7.10: Tracking results of a soccer match video sequence, "seq1".

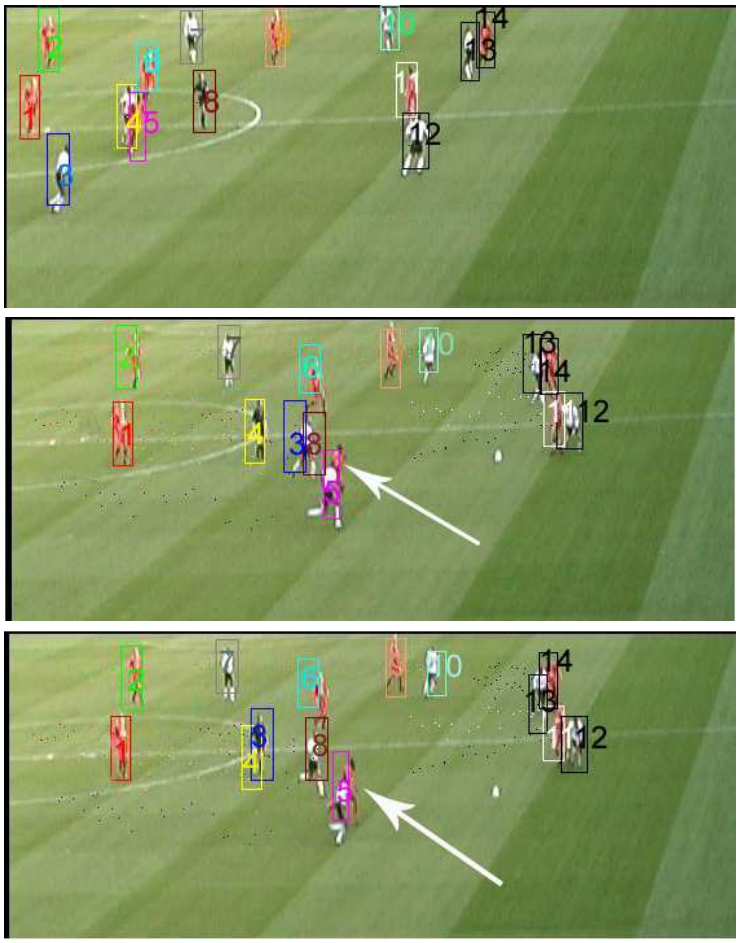


(frame 1004) (frame 1021) (frame 1030)



(frame 1039) (frame 1054) (frame 1070)

Figure 7.11: Tracking results of a soccer match video sequence, "seq2".



(frame 1005, 1033 and 1034)

Figure 7.12: Tracking results of a soccer match video sequence, "seq2" using a conventional MCMC method. Miss detections are marked with white arrows.

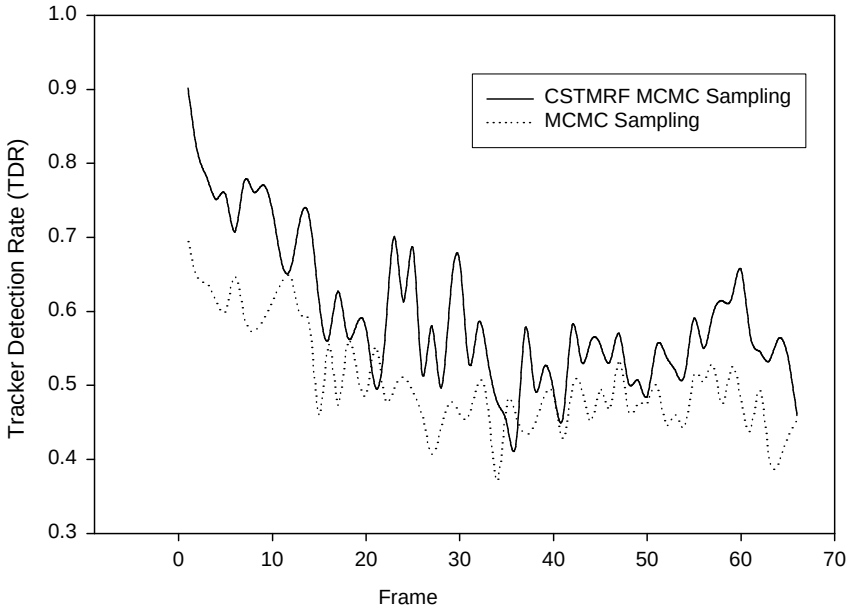


Figure 7.13: Tracker detection rate (TDR) of "seq2"

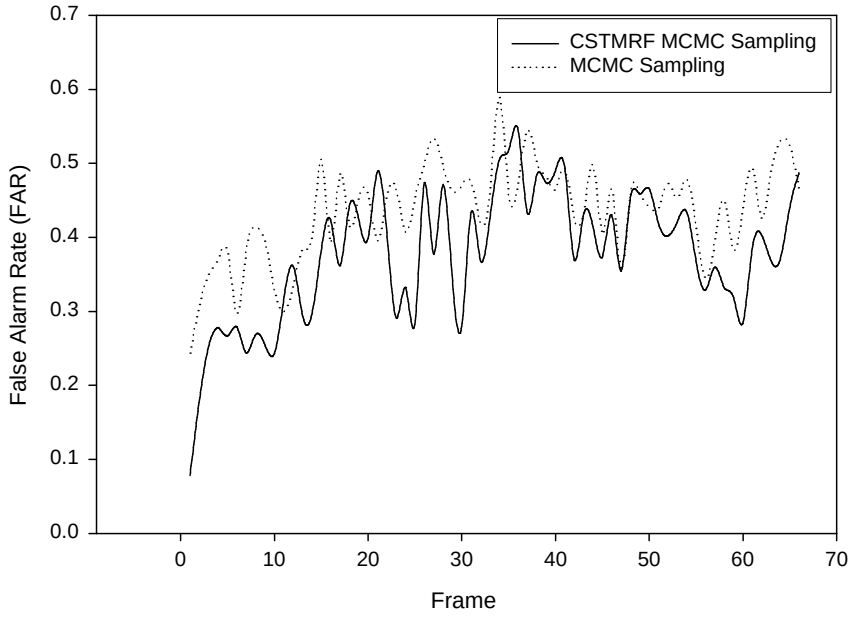


Figure 7.14: False alarm rate (FAR) of "seq2"

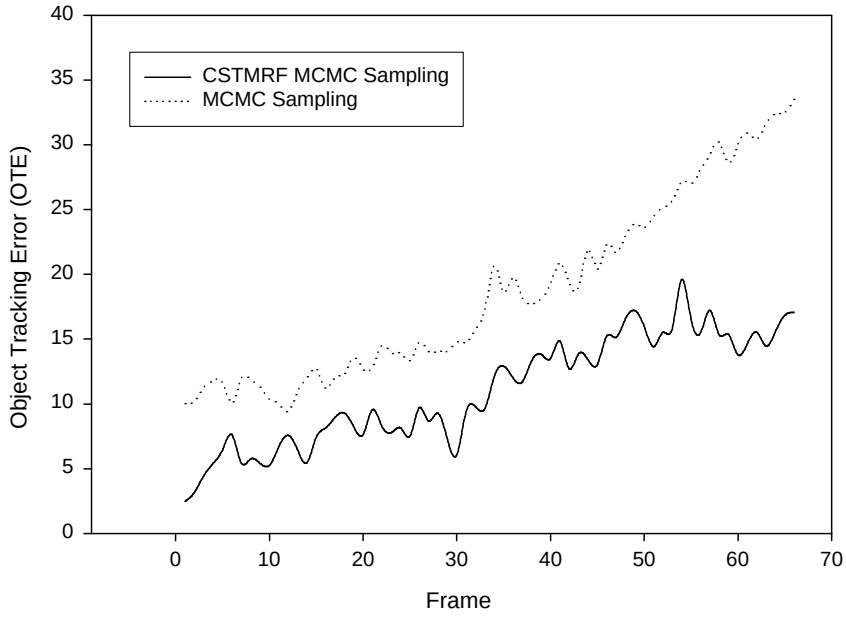


Figure 7.15: Object tracking error (OTE) of "seq2"

Chapter 8

Conclusion and Recommendations for Future Research

8.1 Conclusions

Visual tracking and surveillance has become an active application of computer vision. Many technical issues are involved in the research work of visual tracking and surveillance, such as motion segmentation, object and environment representation, object tracking and behavior understanding. Among these issues, tracking in non-stationary environment and tracking multiple interacting objects are especially important due to the requirement of practicable visual tracking and surveillance systems. Whether the system can keep tracking objects in a dynamic environment and keep tracking multiple interacting objects is a prior request for the following behavior understanding module and determines the performance of the visual tracking

and surveillance system.

In this thesis, approaches have been proposed to track objects in a non-stationary environment and track multiple interacting objects in monocular video. Experiments are demonstrated with satisfactory results, both qualitatively and quantitatively, on very challenging situations.

A transductive particle filter algorithm has been proposed to track objects in a non-stationary environment. The proposed transductive particle filter algorithm combines the transductive adaptable object model with a locally exploring particle filter. This color-based tracker can efficiently and successfully handle non-rigid objects under different appearance changes by its transductive learning ability. Targets can be tracked well despite severe occlusions or clutter.

An attentive MCMC sampling method has been proposed to localize and track multiple interacting objects. Multiple modes of the joint likelihoods surface are localized and maintained using attentive sampling, a novel high-dimensional search method based on sampling incrementally on uncertain regions. Our experiments on real sequences show that this is more effective than the traditional MCMC sampler based on random inflated noise, because our algorithm congregates the samples to the most uncertain region. This strategy makes searching more effective in high-dimensional multi-modal distributions.

Generally, it is difficult to track multiple interacting objects without missing tracking them. In particular, maintaining tracking of multiple objects is very difficult at interactions where various kinds of mutual occlusions and clutters occur.

Large number of interacting objects and frequent occlusions make the problem even worse. In order to achieve robust tracking under severe mutual occlusion, a Color Spatiotemporal MRF model is proposed to reason the interactions among objects. A stochastic search algorithm based on attentive MCMC sampling is used to estimate the state. Experiments on various sports video demonstrate the ability to track multiple objects at interactions with occlusions.

8.2 Recommendations for Future Research

The research work on visual tracking and surveillance is still in its infancy. There are several directions to undertake for future research.

Nonparametric color model stochastic adaptation: Further research will be carried on the analysis of the convergence and stability of the approach. Combining spatial constraints in the transductive inference warrants further investigations as well.

Coherent, high-level likelihood construction: Despite our efforts, modeling multi-object similarity and matching model features to image ones, remains a major challenge. It would be useful to be able to build more globally consistent likelihood surfaces by finding tractable approximations that can account for the interactions of objects.

Trajectory based analysis: Tracking based on Markov assumption, which assumes the state estimation is only related the current and previous frame,

is a very local view and inevitably contains ambiguities. There are also ambiguities that can only be resolved by future observations or the history of observations. For example, whether a completely occluded object disappears or is hiding behind other objects cannot be inferred based only on current and previous observations. Trajectory-based approaches can solve such problems via a more global view, which can give a more meaningful inference.

Integration with behavior understanding modules: One of the objectives of visual tracking and surveillance is to analyze and interpret individual behaviors and interactions among objects to make decisions. The results of tracking are stored and used by high-level behavior analysis modules. It is an interesting research direction that tracking and behavior understanding modules form an integrated system.

Author's Publication List

[1] J.Li and C.S.Chua, "Transductive Local Exploration Particle Filter for Object Tracking", accepted by *Image and Vision Computing*, 2006.

[2] J.Li and C.S.Chua, "Tracking Multiple Interacting Objects With Attentive MCMC Sampling", resubmitted to *IEEE Transactions on Circuits and Systems for Video Technology*, 2005.

[3] J.Li and C.S.Chua, "Transductive Inference for Color-Based Particle Filter Tracking", *International Conference on Image Processing*, pp. 949-952, 2003.

[4] J.Li and C.S.Chua, "Transductive Inference for Color-Based Particle Filter Object Tracking", *Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 17-23, 2003.

[5] J.Li, C.S.Chua and Y.K.Ho, "Color Based Multiple People Tracking", *7th International Conference on Control, Automation, Robotics and Vision*, pp. 309-314, 2002.

Bibliography

[1] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73:428–440, 1999.

[2] C. Andrieu, J. Freitas, and A. Doucet. Sequential bayesian estimation and model selection applied to neural networks. *Technical Report CUED/F-INFENG/TR 341, Cambridge University Engineering Department*, 1999.

[3] C. Andrieu, N.D. Freitas, A. Doucet, and M.I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

[4] 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*, 50:174–188, 2002.

[5] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, Inc., 1988.

[6] Y. Bar-Shalom and X.R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.

[7] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[8] G. Bebis, D. Egbert, and M. Shah. Review of computer vision education. *IEEE Transactions on Education*, 46:2–21, 2003.

[9] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.

[10] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003.

[11] A. Blake and M. Isard. *Active Contours : The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer, 2000.

[12] A. Bottino, A. Laurentini, and P. Zuccone. Toward non-intrusive motion capture. In *Asian Conference on Computer Vision*, 1998.

[13] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2nd Quarter, 1998.

[14] D.H. Brainard and W.T. Freeman. Bayesian color constancy. *Journal of the Optical Society of America*, 14:1393–1411, 1997.

[15] A.D. Bue, D. Comaniciu, V. Ramesh, and C. Regazzoni. Smart cameras with real-time video object generation. In *IEEE International Conference on Image Processing*, pages 429–432, 2002.

[16] H. Buxton and S. Gong. Advanced visual surveillance using bayesian networks. In *International Conference on Computer Vision*, 1995.

[17] T.J. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–245, 1999.

[18] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

[19] C.S. Chua, Y. Ren, and Y.K. Ho. Motion detection with non-stationary background. *International Journal of Machine Image and Applications*, 13:332–343, 2003.

[20] R.T. Collins, A.J. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:745–746, 2000.

[21] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring.

[22] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 142–149, 2000.

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

[24] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *IEEE International Conference on Computer Vision*, pages 716 – 721, 1999.

[25] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 126–133, 2000.

[26] A. Doucet, N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York, Inc., 2001.

[27] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[28] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163, 2002.

[29] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Eur. Conf. on Computer Vision*, pages 751–767, 2000.

[30] F. Fleuret, R. Lengagne, and P. Fua. Fixed point probability field for complex occlusion handling. In *International Conference in Computer Vision*, pages 694–700, 2005.

[31] D.A. Forsyth, J.A. Haddon, and S. Ioffe. The joy of sampling. *International Journal of Computer Vision*, 41:109–134, 2001.

[32] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[33] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[34] D. Gamerman. *Markov Chain Monte Carlo*. Chapman and Hall, New York, 1997.

[35] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[36] A. Gelman, J.B. Carlin, H.S. Stern, and D.R. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

[37] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[38] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[39] P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrica*, 82:711–732, 1995.

[40] M. Greiffenhagen, D. Comaniciu, H. Niemann, and V. Ramesh. Design, analysis, and engineering of video monitoring systems: an approach and a case study. *Proceedings of the IEEE*, 89:1498 –1517, 2001.

[41] I. Haritaoglu, R. Cutler, D. Harwood, and L.S. Davis. Backpack: Detection of people carrying objects using silhouettes. *Computer Vision and Image Understanding*, 81:385–397, 2001.

[42] I. Haritaoglu, D. Harwood, and L.S. Davis. Ghost: A human body part labeling system using silhouettes. In *IEEE International Conference on Pattern Recognition*, 1998.

[43] 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*, 22:809–830, 2000.

[44] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[45] T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *IEEE International Conference on Computer Vision*, pages 334–349, 1998.

[46] C. Hue, J. Cadre, and P. Perez. Sequential monte carlo filtering for multiple target tracking and data association. *IEEE Transactions on Signal Processing*, 50:309–325, 2002.

[47] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:577–589, 1998.

[48] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[49] C. Jaynes, S. Webb, M. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. In *IEEE Workshop on Performance Analysis of Video Surveillance and Tracking (PETS'2002)*, 2002.

[50] A. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[51] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46:81–96, 2002.

[52] S. Ju, M. Black, and Y. Yaccob. Cardboard people: a parameterized model of articulated image motion. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[53] S. Julier and J. Uhlmann. A general method of approximating nonlinear transformations of probability distributions. *Technical report, Robotics Rsearch Group, Department of Engineering Science, University of Oxford*, 1995.

[54] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1:108–118, 2000.

[55] S. Kamijo and M. Sakauchi. Segmentation of vehicles and pedestrians in traffic scene by spatio-temporal markov random field model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 285–288, 2003.

[56] T. Kanade, R.T. Collins, and A.J. Lipton. Advances in cooperative multisensor video surveillance. In *Proceedings of DARPA Image Understanding Workshop (IUW)*, pages 3–24, 1998.

[57] I.A. Karaulova, P.M. Hall, and A.D. Marshall. A hierarchical model of dynamics for tracking people with a single video camera. In *the British Machine Vision Conference*, 2000.

[58] M.W. Lee and I. Cohen. Human upper body pose estimation in static images. In *European Conference on Computer Vision*, pages 126–138, 2004.

[59] M.W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *International Conference on Computer Vision and Pattern Recognition*, pages 334–341, 2004.

[60] J. Li and C.S. Chua. Transductive inference for color-based particle filter object tracking. In *International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 17–23, 2003.

[61] J. Li and C.S. Chua. Transductive inference for color-based particle filter tracking. In *International Conference on Image Processing*, pages 949 – 952, 2003.

[62] J. Li, C.S. Chua, and Y.K. Ho. Color based multiple people tracking. In *International Conference on Control, Automation, Robotics and Vision*, pages 309–314, 2002.

[63] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001.

[64] M. Lievin and F. Luthon. Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video. *IEEE Transactions on Image Processing*, 13:63–71, 2004.

[65] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop Applications of Computer Vision*, pages 8–14, 1998.

[66] J.S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[67] X. Liu and C.S. Chua. Multi-agent activity recognition using observation decomposed hidden markov model. In *International Conference on Vision System*, pages 247–256, 2003.

[68] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[69] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39:57–71, 2000.

[70] J. MacCormick and M. Isard. Bramble: A bayesian multiple-blob tracker. In *IEEE International Conference on Computer Vision*, pages 34–41, 2001.

[71] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[72] R. Merwe, A. Doucet, N. Freitas, and E. Wan. The unscented particle filter. *Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department*, 2000.

[73] N. Metropolis, A. Rosenbluth, R. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[74] T.B. Moeslund and E. Granum. Multiple cues used in model-based human motion capture. In *The Fourth International Conference on Automatic Face and Gesture Recognition*, 2000.

[75] D. Murray and A. Basu. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:449–459, 1994.

[76] A. Nakazawa, H. Kato, and S.Inokuchi. Human tracking using distributed vision systems. In *International Conference on Pattern Recognition*, pages 1–4, 1998.

[77] M. Naylor and C.I. Attwood. Annotated digital video for intelligent surveillance and optimized retrieval. *Final Report IST1999-11287: ADVISOR, The ADVISOR Consortium*, 2003.

[78] C.J. Needham and R.D. Boyle. Performance evaluation metrics and statistics for positional tracker evaluation. In *Computer Vision Systems Third International Conference*, pages 278–289, 2003.

[79] K. Nummiaro, E. Koller-Meier, and L.V. Gool. A color-based particle filter. In *First International Workshop on Generative-Model-Based Vision*, pages 53–60, 2002.

[80] K. Nummiaro, E. Koller-Meier, and L.V. Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21:99–110, 2003.

[81] K. Okuma, A. Taleghani, N. Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39, 2004.

[82] 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*, 22:831–843, 2000.

[83] N. Papamarkos, A. Atsalakis, and C. Strouthopoulos. Adaptive color reduction. *IEEE Transactions on Systems, Man and Cybernetics*, 32:44–56, 2002.

[84] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675, 2002.

[85] PETS-ICVS03. http://petsicvs.visualsurveillance.org.

[86] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 77:257–285, 1989.

[87] Y. Raja, S. McKenna, and S. Gong. Color model selection and adaptation in dynamic scenes. In *Proc. of European Conference on Computer Vision*, pages 460–474, 1998.

[88] Y. Ren, C.S. Chua, and Y.K. Ho. Motion detection from time-varied background. In *Asian Conference on Computer Vision*, pages 222–227, 2002.

[89] Y. Ren, C.S. Chua, and Y.K. Ho. Statistical background subtraction for foreground detection with non-stationary background for human-tracking. *Pattern Recognition Letters*, 24:183–196, 2003.

[90] G. Rigoll, S. Eickeler, and I. Yalcin. Performance of the duisburg statistical object tracker on test data for pest2000. In *First IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PEST2000)*, pages 1–7, 2000.

[91] K. Rohr. *Human Movement Analysis Based on Explicit Motion Models*. Kluwer Academic, Dordrecht/Boston, 1997.

[92] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE International Conference on Robotics and Automation*, pages 1665–1670, 2001.

[93] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages 702–718, 2000.

[94] N. Siebel and S. Maybank. The advisor visual surveillance system. In *ECCV 2004 workshop on Applications of Computer Vision*, pages 103–111, 2004.

[95] N.T. Siebel. Design and implementation of people tracking algorithms for visual surveillance applications. *Phd thesis, the University of Reading*, 2003.

[96] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 6:371–393, 2003.

[97] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *International Conference on Computer Vision and Pattern Recognition*, pages 69–76, 2003.

[98] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1371–1375, 1998.

[99] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transaction on Pattern and Analysis and Machine Intelligence*, 22:747–757, 2000.

[100] R.L. Streit and T.E. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceedings of SPIE International Symposium, Signal and Data Processing of Small Targets*, pages 394–405, 1994.

[101] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, pages 11–32, 1991.

[102] M.J. Black S.X. Ju and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[103] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, pages 255–261, 1999.

[104] Y. Tsaig and A. Averbuch. Automatic segmentation of moving objects in video sequences: A region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:597–612, 2002.

[105] Z. Tu and S.C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:657 – 673, 2002.

[106] Z.W. Tu, X.R. Chen, A.L. Yuille, and S.C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *IEEE International Conference on Computer Vision*, pages 18–25, 2003.

[107] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *International Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[108] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *International Conference on Computer Vision*, pages 734–741, 2003.

[109] P. Viola and W.M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24:137–154, 1997.

[110] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer Verlag, 1995.

[111] C. Wren and A. Azarbayejani. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

[112] Y. Wu and T.S. Huang. Color tracking by transductive learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 222–227, 2000.

[113] Y. Wu and T.S. Huang. Nonstationary color tracking for vision-based human-computer interaction. *IEEE Transactions on Neural Network*, 13:948–960, 2002.

[114] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–665, 1991.

[115] M. Yeasin, E. Polat, and R. Sharma. A multiobject tracking framework for interactive multimedia applications. *IEEE Transactions on Multimedia*, 6:398–405, 2004.

[116] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1531– 1536, 2004.

[117] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[118] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[119] S.C. Zhu, C.E. Guo, Y.Z. Wang, and Z.J. Xu. What are textons? *International Journal of Computer Vision*, 62:121–143, 2005.

[120] S.C. Zhu, R. Zhang, and Z.W. Tu. Integrating top-down/bottom-up for object recognition by data driven markov chain monte carlo. In *International Conference on Computer Vision and Pattern Recognition*, 2000.