NORTHWESTERN UNIVERSITY

Multiple Motion Analysis for Intelligent Video Surveillance

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical and Computer Engineering

By

Ting Yu

EVANSTON, ILLINOIS

June 2006

ABSTRACT

Multiple Motion Analysis for Intelligent Video Surveillance

Ting Yu

With the proliferation of camera sensors deployed world widely, video surveillance systems are gradually finding their way into our daily lives. A direct consequence of these technological advancements is the increased demand for intelligent video analysis and understanding techniques.

This dissertation concentrates on the developments of efficient and effective multiple motion analysis techniques that allow automated tracking of multiple targets, which is arguably the most challenging problem and essential component of any intelligent video surveillance systems.

Besides sharing the common challenges faced by visual tracking of single target, including large appearance variations, complex object motions, successful tracking of multiple targets' motions are also confronted by the tremendous difficulties from the theoretical and practical aspects of the problems, such as target occlusions, unknown number of targets, ambiguities induced by multiple target-tracker associations, high computational demanding, and difficulty of training a target detector.

This dissertation presents several effective and computationally efficient techniques to addressing these challenges: a dynamic Bayesian network formulation for the multiple target tracking with explicit occlusion reasoning; a decentralized framework to multiple target tracking based on Markov network that handles the variable number of targets and copes with the tracker coalescence problem with close to linear complexity; a novel two-layer statistical field model to characterize the large shape variability and partial occlusions for nonrigid target detections, especially pedestrian detections; a component-based appearance tracker based on support vector machines to accommodate the large object appearance variations with the extra appealing capacity of automatically selecting trustworthy components while down-weighting the unreliable occluded components; a novel differential tracking approach based on a spatial-appearance model (SAM) formulation to combine the local appearances variations and global spatial structures enabling the continuous tracking of non-rigid objects that exhibit dramatic appearance deformations, large object scale changes and partial occlusions. Extensive experiments and very encouraging results on both the synthetic and real-world data verified the effectiveness and efficiency of the proposed methods.

Dedicated to my dear wife Zigeng

Acknowledgements

I would like to express my deepest gratitude to my advisor Professor Ying Wu for bringing me into the exciting world of computer vision, for all the years' extraordinary support and supervision, not only on the academic side but also on my personal life, and for every effort he makes to drive me to be successful towards my future career endeavors. It is from his personal example as well as daily instructions that I learned how to become an outstanding researcher, which I wish I do not fail Prof. Wu's expectations.

I would like to thank my summer intern mentors, Dr. Cha Zhang, Dr. Michael Cohen and the research manager Dr. Yong Rui, for their fruitful discussions, detailed guidance and close collaborations, which greatly broaden my knowledge and make my intern experience at Microsoft Research a really pleasant memory. I would also like to thank Professor Aggelos K. Katsaggelos and Professor Thrasyvoulos N. Pappas. for serving as members of my committee and providing enlightening suggestions.

I cherish my friendships with Dr. Mei Han, Wei Xu and Wei Hua, made during my intern at NEC Labs America. I also would like to take this opportunity to thank my group fellows at Northwestern University, Dr. Gang Hua, Zhimin Fan, Junsong Yuan, Ming Yang, Shengyang Dai, which really make our group like a rapport big family. I also enjoy the friendships with Xun Xu, Xiaodan Fan, Dr. Junqing Chen, Ning Wen, Dr. Yishen Sun, and Feidu Luo.

I wish to thank my parents and brother for their love and support though all the years of my study. Finally, I dedicate the work presented in this dissertation to my dear wife Zigeng for all her love, support, understanding, encouragement and help, which make everything possible and meaningful.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1. Background

Over the past few decades, the rapid advancement of information technology has lead to significant improvements of the computational capacities of processing hardware with relatively low cost. The development of sensor technology also made low power and on-chip computing-based video cameras widely available. By connecting the cameras to the inexpensive but powerful computing hardware, it now becomes feasible and convenient to set up a camera-computer system to perform perceptual-based intelligent human computer interactions, such as face/human detection, user tracking, gesture recognition, and behavior analysis [141, 101].

In addition, the combination of the sensor technology with the communication network research also enables a new promising research direction on the large scale wired/wireless camera networks, which targets on the potential applications of large area video surveillance [18,22,23,34,72], environment monitoring [91,99], and possible medicare applications such as tele-monitoring of elderly people for assistance in safety [150].

Compared with other sensing modalities, such as radar, infrared (IR) or sonar, camera sensor provides the unique visualization properties, making itself especially appealing to modern surveillance applications. On one hand, video data can be directly interpreted by humans and offer relatively high-resolution measurements that allow the access to many

details about the environment. On the other hand, the side effect of high sensor resolution implies large quantities of video data that require tedious analysis in order to find out specific information of interest.

The relatively easy access to large volume of visual data, captured by the camera sensors and recorded in the storage devices, has lead to an increased demand for intelligent software solutions to automatic video understanding, the essential problems of computer vision research. Intelligent video surveillance embeds a range of image understanding techniques, which automate the analysis of video data in order to extract various semantical level events in accordance with human knowledge.

Depending on the pursued semantical levels, we may roughly categorize the utilized computer vision techniques into the following groups:

- *Background subtraction:* In the lowest semantical level, pixel-based classification may be taken to extract the interesting regions and remove unwanted backgrounds, which generally achieves some attention mechanism, similar to biological vision system, to allow the later more computationally intensive algorithms to focus on more likely regions [38, 45, 57, 89, 109, 120, 125, 132].
- *Object detection and tracking:* In the middle level, one aims to detect and count some specific objects of interest, like humans or vehicles [129, 98, 47, 114, 88, 75, 29, 137, 139, 133]. And more elaborate applications involve analyzing the video sequences, identifying multiple moving regions induced by the interesting objects, and then tracking all these objects during their presences within the field of views of the cameras [6, 5, 11, 55, 107, 63, 86, 123, 160, 95, 73, 151].

- *Action, activity and event recognition:* Based on the extracted relatively low-level knowledge, the more sophisticated semantical understanding of the videos focuses on recognizing object actions, interpreting the behaviors and activities of each object or object groups [15, 65, 87, 119, 16, 81, 161, 92, 143, 13, 31, 36, 117, 149].

The higher semantical level understanding of the video relies on the feasible solutions to extract the lower level knowledge. For example, object level identifications and motion trajectories computed from object detection and tracking, serving as the middle level semantical features, are fed to the event modelling, classification and mining procedures to facilitate the possibilities of high level semantic reasoning. Though the problem of pixel level classification to identify the interesting regions has been solved in some sense, the existing solutions to the middle and high level understanding of the visual data are still far from satisfactions.

With the recent advancements of numerous literatures devoting into the visual tracking research, it might be able to reasonably claim that there are feasible solutions to addressing the robust tracking of single target, however, simultaneously analyzing multiple targets' motions and tracking them in video stay as one of the most challenging problems in computer vision.

Multiple targets' motions, quite often observable from the real data, prevent a simple solution of instantiating multiple independent single target trackers to solve the problem, because both the number of targets that need to be tracked are unknown beforehand, and there are potential ambiguities introduced by multiple target-tracker associations. Besides sharing the common challenges faced by visual tracking of single target, successful tracking of multiple targets' motions are thus confronted by the tremendous difficulties from the

theoretical and practical aspects of the problems, which in generally can be summarized as follows:

- *Target appearance variations:* Visual appearance, as the most important cue of characterizing the target to enable a target tracker, render numerous variations due to both the intrinsic and extrinsic reasons. For example, the object appearance can look quite different when the object is showing large shape deformation and pose changes. In addition, the environmental factors, such as illumination changes, cluttered backgrounds, all bring huge variations to the object appearance.
- *Target occlusion:* The targets may be occluded by other targets of interest or recurrent sources that are not of interest (such as background clutter). When there are multiple targets in the scene, some of them may occlude one another due to their spatial proximities and relative distances to the camera. Such a problem becomes even more severe, when targets, such as people in the surveillance camera, move in a group, where the occlusion is often persistent.
- *Computational demanding:* The heavy computations are mainly due to the large number of targets that need to be tracked simultaneously, where we will show in later chapters, the required computational cost for the existing solutions are mainly at the level of exponential or combinatorial complexity with regard to the number of targets.
- *Difficulty of training target detectors:* Unlike the radar tracking scenario [6,11,94, 162], where a signal processing component is usually available to facilitate the collections of target detections, thus making multiple target tracking problem better

defined. Multiple target tracking from video is also encountered by the general difficulty of training a visual target detector, which itself, if not impossible, is a quite challenging problem.

During the visual tracking of a single target, the large uncertainties of the visual appearances significantly complicate the measurement models and the matching measures. This difficulty is also shared in object detection and recognition, and has been studied extensively. To address this problem in the context of tracking, we should not let the handling of this problem to jeopardize the requirement of computational efficiency of the tracker.

In general, when a target is partially or completely occluded, its visual appearance would dramatically deviate from its appearance template as we set for tracking. Thus, occlusion becomes a special and difficult problem for appearance-based tracking. In terms of multiple target tracking using appearances, explicit handling of occlusion is especially indispensable for tracking, since occlusion would probably occur when different targets interact. The ignorance of the occlusion problems in multiple target tracking will usually result in the tracking failure, where the trackers may lose tracking some of the occluded targets.

The high computational cost of the existing solutions to the multiple target tracking problem is mainly due to the adoption of a centralized methodology by considering a joint data association, which enumerates all the possible associations between targets and observations. Various methods along this line have been developed. The essence of their methods is the introduction of the joint state space representation, which concatenates together all the state spaces of the individual targets such that they can be jointly inferred

based on the exhaustive data association enumerations. However, due to the exploration of a high-dimensional joint state space, these methods are generally computational intensive. For example, the multiple hypothesis tracker (MHT) [108,28,55] and the joint probabilistic data association filter (JPDAF) [43, 6, 107, 17] have to evaluate all possible associations that suffer from the combinatorial complexity, and the sampling-based methods [60,63,86, 123,138,160,32,73] are confronted by the exponential demand of the increase of particles.

The demands for a robust *human* detector, as our primary interest is in the application to intelligent video surveillance, are also confronted by many difficulties. Although the research of object detection has greatly moved forward with the success of face detection, these developed methods for face detection may not apply to human detection. The visual appearances of the human present tremendous variabilities while lacking apparent visual invariants, as the diversified clothing and the body articulation may significantly change the image of a person.

1.2. Motivation

The listed challenges and discussions in the previous section motivate us to address these critical problems before we can move to the higher semantical level understanding of the video content. More specifically:

- We would like to deal with the explicit occlusion reasoning under the multiple target tracking scenario, where the occlusion relationships (i.e. who is occluding whom) between the tracked targets will be explicitly characterized. However, we will show later that a necessary centralized methodology has to be taken to achieve this goal, which implies that the more accurate descriptions about

the multiple target motions are brought out with the trade-off cost of higher computational demands.

- Such a centralized solution might be fine to a powerful processor. However, they are not appropriate for our previously mentioned emerging application of wireless camera networks. In this application setup, there are a large number of camera units that have the functionality of sensing, computing and communicating. However, these units are power-limited to prevent much computation and communication [77]. To make good use of such sensor networks for target tracking, complex computation must be distributed into the network, since once a certain unit takes charge of sensing, its computational load on target tracking needs to be migrated to other idle units. Although this research is being carried out at the computer architecture level, it is more desirable to find a decentralized scheme at the algorithm-level for efficient tracking of multiple targets. When it is available, it will lead to the essential parallelization and distributed computing. This can be a very promising and interesting application setup that strongly motivates us to search for a decentralized formulation to addressing the multiple target tracking problem.
- Although it may be true that the state-of-the-art multiple target tracking algorithms can work through bootstrapping without the help of the target detectors, the practical deployment of a a successful multiple target tracking algorithm, such as multiple pedestrian tracking for intelligent video surveillance in our consideration, may be largely benefited from the availability of a robust object detection component. The existence of a reliable object detector is able to help collect the

valid target observations from video frames. It can not only reduce the number of wasted hypothesis due to inaccurate dynamics model of the tracker, but also provide some informative clues to guide the trackers to searching around more likely areas where the targets may move. The benefits, which can be obtained from the robust human detection research, also stimulate us to investigate along this direction.

- The solution to multiple target tracking problem does not like a simple algorithm of running multiple independent trackers, however, an effective and efficient single target tracker, which is capable of handling the tremendous appearance variations and recovering any complex object motions, can obviously be greatly beneficial to the efficacy and robustness of any multiple target tracking algorithms. Therefore, in view of constructing a comprehensive multiple target tracking framework, a deep investigation on robust single target tracking is also a must.

1.3. Organization

In this dissertation, several novel approaches to addressing the critical problems, such as multiple target motion analysis, human detections, and robust single target tracking will be presented, which serve as the basis for our targeting application of intelligent video surveillance. More specifically, the dissertation is organized as follows:

- Chapter 2 presents a centralized approach based on a dynamic Bayesian network (DBN) formulation to tackle the multiple target tracking problem with explicit occlusion handling, where the term "centralized" implies that a joint state space representation is adopted to concatenate all target states into a joint inference.

In Section 2.2, we firstly briefly review the previous solutions to appearance tracking of multiple targets. Section 2.3 presents the dynamic Bayesian network for occlusion process. The sequential Monte Carlo strategy is described in Section 2.4 to serve as the inference engine of the underlying probabilistic graphical model. We further present a multiple view appearance model in Section 2.5 to accommodate the target appearance variations due to object pose changes. The generative model that combines the occlusion model and multiple view switching is described in Section 2.5. Experiments are given in Section 2.6 and discussions are given in Section 2.7.

- Chapter 3 presents a novel decentralized and linear complexity framework for visual tracking of multiple targets, with simultaneous coalescence problem handling. In addition, we relax the general assumption that the number of targets being tracked is given, where our solution can simultaneously track variable number of targets. Section 3.3 presents our decentralized multiple target tracking framework that is based on Markov network formulation in details. The autonomous trackers with self-evaluation capacity are introduced in Section 3.4, which thus enables the framework to be able to correctly estimate the number of targets moving in the video scene at any time instant, and then track these variable number of targets. Section 3.5 describes our variational analysis of the proposed Markov networks, where loopy structure may present. Based on the mean field fixed point equations derived from the variational analysis, Section 3.6 introduces a sequential Monte Carlo implementation to approximate the variational inference to the loopy Markov network. An importance sampling mechanism is also

presented to effectively combine the available target detectors with the trackers. Extensive experiments on various synthetic and real video sequences are reported in Section 3.7. Finally, Section 3.8 summarizes the chapter and makes a number of suggestions for further improvements.

- Chapter 4 presents a novel object detection framework based on a two-layer statistical field model to effectively capture the large shape variability of deformable object, especially humans. After a brief description of the related work in Section 4.2, we describe the proposed two-layer field model in Section 4.3. The probabilistic variational analysis of this model is given in Section 4.4, and the learning algorithm is presented in Section 4.5. Section 4.6 describes our methods for pedestrian detection and tracking, and our extensive results are reported in Section 4.7. The chapter concludes in Section 4.8.
- Chapter 5 describes one of our proposed novel approaches to dealing with large object appearance variations, with the aiming of developing a robust single target tracker. Motivated by the factor that larger image regions incur more variabilities, while smaller regions are more likely to be manageable, this chapter pursues a robust appearance tracking idea by combining the previous support vector tracking algorithm with a component-based object representation. Section 5.2 provides a brief overview of the original SVT method. Section 5.3 describes the formulation and solution of the proposed component SVT algorithm. The further extension of the model to handling severe occlusions is discussed in Section 5.4. Section 5.5 then shows experimental results. Conclusions and discussions are made in Section 5.6.

- Chapter 6 devotes an additional chapter into the problem of robust single target tracking, where a novel differential tracking approach based on a spatial-appearance model (SAM) that combines local object appearances variations and global spatial structures is proposed. The SAM is in the form of a Gaussian mixture model, as described in Section 6.2. Section 6.3 presents the designed maximum-likelihood estimation for tracking, which is solved by a variant of Expectation-Maximization (EM) algorithm. Section 6.4 shows the exciting results of the proposed approach. And Section 6.5 presents the conclusions and some discussions.
- Chapter 7 summarize the dissertation, and some promising future research topics are given at the end.

1.4. Contributions

Motivated by the existing challenges and insufficient solutions to the different problems under multiple target tracking, this dissertation proposes several original contributions to tackle the problems, including *target occlusion*, *high computational demand*, *unavailability of target detector*, and *large target appearance variations*. As we have illustrated, all of these problems are the critical issues that must be addressed in order to support the successful applications of the intelligent video surveillance. To summarize, the following contributions have been made in the dissertation:

- A novel centralized formulation to tackle the multiple target tracking problem has been developed with explicit occlusion reasoning. A dynamic Bayesian network formulation is proposed as a generative model by accommodating the hidden

process of occlusion in the probabilistic framework, which can stipulate the conditions on which the image observation likelihood is calculated. The statistical inference of such a hidden process can reveal the occlusion relations among different targets, thus making the tracker more robust against partial even complete occlusions. In addition, considering target appearances are affected by views, another generative model for multiple view representation is also proposed by adding a switching variable to select from different view templates. The sampling-based sequential Monte Carlo strategies are developed to achieve the tracking and inferencing from the complicated network formulation [138].

- A novel linear complexity decentralized framework is proposed to address the multiple target tracking problem with coalescence problem handling. The basic idea is a distributed while collaborative inference mechanism, where the motion state of each target, estimated by the tracker covering it, is not only determined by its own observation and dynamics, but also through the interaction and collaboration with the state estimates of its adjacent targets. Markov networks are proposed to model such competition correlations, and variational analysis is employed and reveals a mean field approximation to the posteriors of the trackers, therefore providing a computationally efficient way to this difficult inference problem [151, 152].
- In order to track the variable number of targets, we further propose an autonomous while collaborative tracker network formulation, where each tracker is equipped with an entropy-based performance evaluator, which can switch the tracker between active or inactive status, indicating if they are currently follow

targets or not. Motion estimates of the targets from this set of autonomous trackers can still be distributed into a Markov network formulation. Furthermore, target detectors are effectively combined with each autonomous tracker to help sense the potential newly appearing targets in the dynamic scene, therefore background subtraction is not necessary to our method. The use of object detectors also supports the construction of an effective importance function, which leads to a more efficient variational inference [153, 155].

- A novel two-layer statistical field model is proposed to characterize the large shape variability and partial occlusions for nonrigid target detections, especially pedestrian detections. The complex shape changes are captured by a well trained Boltzmann distribution as a prior model. Probabilistic variational analysis of this model reveals a set of fixed point equations that give the equilibrium of the field, leading to computationally efficient methods for calculating the image likelihood and for training the model. Based on that, effective algorithms for detecting and tracking nonrigid objects are developed [140, 137].
- A novel solution of pursuing a component-based idea in visual tracking is proposed with the emphasis on the handling of the challenges such as partial occlusions in a computationally efficient way. Our solution is based on the optimal integration of a set of correlated simple component support vector trackers, where the collaboration among the set of component trackers enables the better handling of object appearance variations. In addition, the enhanced model, by

introducing a selective mechanism, can automatically select trustworthy components while down-weight the unreliable ones that may be occluded, thus enabling the reliable dealing of object partial occlusions [154].

- A novel differential tracking approach is developed based on a spatial-appearance model (SAM) that combines local appearances variations and global spatial structures. This model can capture a large variety of appearance variations that are attributed to the local non-rigidity. At the same time, this model enables efficient recovery of all motion parameters. Rigorous derivation of the model lead to a closed form solution to motion tracking. The obtained encouraging results demonstrate the effectiveness and efficiency of the proposed method for tracking non-rigid objects that exhibit dramatic appearance deformations, large object scale changes and partial occlusions [156].

It is interestingly noted that although the first two Chapters 2, 3 and the later Chapters 5, 6 are devoted into two seemingly quite different methodologies of visual tracking, where Chapters 2, 3 are under the probabilistic tracking formulations, while Chapters 5, 6 are addressing the visual tracking in the deterministic differential-based tracking setups. However, the powerful importance sampling strategy [153, 155], capable of combining the top down probabilistic sampling procedures and the bottom up deterministic matching processes, serves as the clue to glue different parts together. In addition, the proposed target detection method based on the statistical field model in Chapter 4 can also be effectively bonded into the same multiple target tracking framework with the designing of importance function.

CHAPTER 2

Multiple Target Tracking: A Centralized Solution

2.1. Introduction

Tracking multiple targets based on their appearances play an important role in many applications such as intelligent human computer interaction and video surveillance. For example, before the detailed facial motion can be recovered and before the human identities can be recognized, we need to locate and track faces in video sequences. An effective way is through matching and tracking face appearances. Since image appearances provide more comprehensive visual information to represent the targets, e.g., the faces, appearance-based tracking methods receive more and more attention.

However, if a target is partially or completely occluded, its visual appearance would dramatically deviate from its appearance template as we set for tracking. Thus, occlusion becomes a special and difficult problem for appearance-based tracking. In terms of multiple targets tracking using appearances, explicit handling of occlusion is especially indispensable for tracking, since occlusion would probably occur when different targets interact. The ignorance of the occlusion problems in multiple target tracking will easily result in the tracking failure, where the trackers may lose tracking some of the occluded targets.

This chapter proposes a centralized approach to tackle the multiple target tracking problem with explicit occlusion handling, where the term "centralized" implies that a

joint state space representation is adopted to describe the motion states of all targets being tracked. The theoretical foundation of the approach is based on a dynamic Bayesian network, which functions as a generative model to accommodate the occlusion process explicitly. This generative model consists of multiple hidden Markov processes: the dynamics of each individual target, and the process of the occlusion relation whose transition is characterized by a finite state machine. In addition, the model describes the formation (or generation) of the image observations, jointly conditioned on the targets states and their occlusion relations. Then, tracking is to infer the states of all these hidden Markov chains based on the sequence of image observations.

We investigate two representations for the appearances from single view and multiple views. The single view appearance is represented by an appearance template associated with a transformation that depicts the motion and deformation of the template. Since the appearances change with views, we extend this "view+transformation" representation to the multiple view case, by switching among a set of templates and transformations. This mechanism is also modelled by a generative model which contains a hidden switching process.

The combination of the occlusion modelling and the multiple view representation results in a multilevel dynamic Bayesian network. Due to the complexity in the structure of the generative model, the inference of the model is approximated by the sampling-based sequential Monte Carlo strategies. Various test sequences showed the effectiveness of this approach to handle the occlusion situations.

The proposed approach accommodates the inference of the occlusion relations of multiple targets and the switch of multiple views into a probabilistic tracking framework. Not

limited to multiple face tracking, the proposed generative model is general and valid for many tracking scenarios which need to handle occlusion explicitly.

The chapter is organized as follows. In Section 2.2, we briefly review the previous solutions for appearance tracking of multiple targets. Section 2.3 presents the dynamic Bayesian network for occlusion. The sequential Monte Carlo strategy is described in Section 2.4. Section 2.5 presents the multiple view appearance model. The generative model that combines the occlusion model and multiple view switching can also be found in Section 2.5. Experiments are given in Section 2.6 and conclusions are in Section 2.7.

2.2. Previous Work

The representations for targets affect the effectiveness and efficiency of tracking algorithms. Many approaches have been studied based on different target representations, e.g., image appearances [9, 24, 52, 68, 124] and geometrical shapes [8, 62, 86]. Shape-based approaches are concerned about the matching between shape models and image features. They need to deal with more ambiguities in tracking but are less sensitive to lighting. On the other hand, since massive image appearance data contain very rich information for characterizing targets, appearance-based methods would not sensitive to image resolutions, but special attention need to be taken for deformation and lighting [157, 134].

Many different types of appearance models have been investigated, such as color appearances [24], eigen appearances [9], texture appearances [68], layered image template appearances [124], and the appearances combining image template and lighting [52]. All of these models parameterize the appearances for target representations.

Tracking targets includes the estimation of these parameters. There are two methodologies to this problem: *bottom-up* and *top-down*. The bottom-up approaches generally formulate the problem as nonlinear optimization problems which minimize some error functions, e.g., flow residue [9, 52] and color discrepancy [24]. On the other hand, the top-down approaches adopt the idea of *analysis-by-synthesis*, by directly verifying plenty of hypotheses [62, 86].

Most bottom-up algorithms are computationally more efficient, but they are subject to the validation of the small motion assumption, and it is hard for them to cope with occlusions unless the appearance model itself is robust against occlusions. On the other hand, most top-down algorithms involve more computation, but the motion estimation tends to be more accurate and more robust. In addition, occlusion can be modelled from top-down in the same framework.

The generative model approaches take a top-down methodology, by modelling the hidden factors that would affect the observed data [69]. Once the structure and the parameters of the model are set, those hidden factors can be inferred and the parameters can be learnt from the data. As a special case, dynamic Bayesian networks model dynamic systems and temporal signals [100]. The inference of the networks provides tracking results directly.

To track multiple appearances with occlusions, this chapter describes a class of dynamic Bayesian networks that accommodates the hidden process of occlusion and model the switching of the appearance templates of multiple views.

2.3. A Generative Model for Occlusion

We take a "view+transformation" approach to represent the state of a target, which consist of an appearance template T and a transformation H. The template T can be any kind of templates, such as an image template, an edge map template, or a texture template. The transformation H can be an affine transformation or a homography transformation.

To make the description clearer, we limit to the situation of tracking two targets (i.e., A and B). We denote the *target state* of target k at time t by $\mathbf{X}_t^k$. The tracking task is to infer $\mathbf{X}_t^A$ and $\mathbf{X}_t^B$ based on all the observed image evidence $\underline{\mathbf{Z}}_t = \{\mathbf{Z}_1, \cdots, \mathbf{Z}_t\}$, where $\mathbf{Z}_t$ is the image *measurement* (or *observation*) at time t, i.e., to estimate $p(\mathbf{X}_t|\underline{\mathbf{Z}}_t) = p((\mathbf{X}_t^A, \mathbf{X}_t^B)|\underline{\mathbf{Z}}_t)$, where $\mathbf{X}_t = (\mathbf{X}_t^A, \mathbf{X}_t^B)$.

We are concerned about the occlusions between these targets, i.e., a target is occluded by a known object. This chapter does not investigate a more challenging situation where the target is occluded by a completely unknown object, since no clue from the occluding object can be used for occlusion detection. But it will be part of our future work.

The tracking process can be viewed as the density propagation [62] from $p(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1})$ to $p(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$, and it is governed by the dynamic model $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ and the observation model $p(\mathbf{Z}_t|\mathbf{X}_t)$, since we have

$$p(\mathbf{X}_t|\underline{\mathbf{Z}}_t) \propto p(\mathbf{Z}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1}) d\mathbf{X}_{t-1}$$

In addition, since the motion of two targets are independent, we have $p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_t^A|\mathbf{X}_{t-1}^A)p(\mathbf{X^B}_t|\mathbf{X}_{t-1}^B)$. Then we have

$$p(\mathbf{X}_t|\underline{\mathbf{Z}}_t) \propto p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B) \int p(\mathbf{X}_t^A|\mathbf{X}_{t-1}^A)$$

$$\times p(\mathbf{X}_t^B|\mathbf{X}_{t-1}^B)p(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1})d\mathbf{X}_{t-1}$$

If there is no occlusion between A and B, the observation likelihood $p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B)$ can be uniquely determined. However, when one target occludes the other, the occlusion relation has to be known before the likelihood can be uniquely calculated, i.e., the likelihood should be conditioned on the occlusion relations additionally. Let $\alpha_t \in \{0, 1, 2\}$ denote the occlusion relation, i.e., $\alpha_t = 0$ indicates no occlusion, $\alpha_t = 1$ indicates A $\bigwedge$ B, and $\alpha_t = 2$ indicates B $\bigwedge$ A, where $\bigwedge$ means "occludes". Then based on the joint likelihood $p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t)$, we have

$$p(\mathbf{X}_t, \alpha_t|\underline{\mathbf{Z}}_t) \propto p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t) \int p(\mathbf{X}_t^A|\mathbf{X}_{t-1}^A)$$

$$\times p(\mathbf{X}_t^B|\mathbf{X}_{t-1}^B)p(\alpha_t|\alpha_{t-1})p(\mathbf{X}_{t-1}, \alpha_{t-1}|\underline{\mathbf{Z}}_{t-1})d\mathbf{X}_{t-1} \tag{2.1}$$

where $p(\alpha_t|\alpha_{t-1})$ describes the transition of occlusion relation. Thus, based on Equation 2.1, the probabilistic dynamic system can be illustrated by a factorized graphical model (a factorized dynamic Bayesian network) in Figure 2.1.

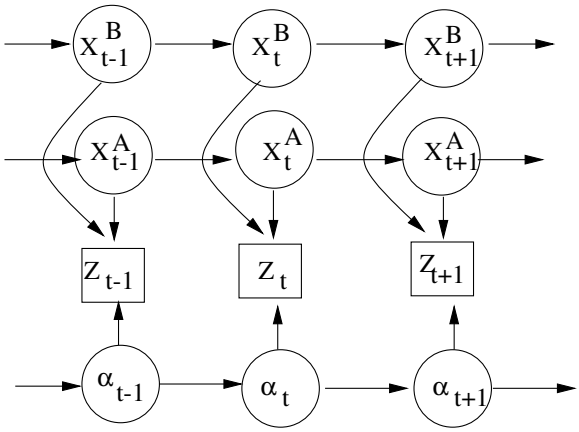


Figure 2.1. A hidden process $\{\alpha_t\}$ is accommodated in the dynamic Bayesian network to present the occlusion relationships.

The posterior density of occlusion can be obtained through integrating out $\mathbf{X}_t^A$ and $\mathbf{X}_t^B$ from the joint posterior probability, i.e.,

$$p(\alpha_t|\underline{\mathbf{Z}}_t) = \int\int p(\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t|\underline{\mathbf{Z}}_t) d\mathbf{X}_t^A d\mathbf{X}_t^B \tag{2.2}$$

As a generative model, this dynamic Bayesian network models the forwarding process of image formation. In the graphical model, there are three hidden Markov processes, $\{\mathbf{X}_t^A\}$, $\{\mathbf{X}_t^B\}$ and $\{\alpha_t\}$, which are to be inferred from the observation data $\underline{\mathbf{Z}}_t$, based on all the conditional probabilities as illustrated by arrows in the graph. Specifically, to characterize the model, we need to model the dynamics of the two targets $p(\mathbf{X}_t^A|\mathbf{X}_{t-1}^A)$ and $p(\mathbf{X}_t^B|\mathbf{X}_{t-1}^B)$, the transition model $p(\alpha_t|\alpha_{t-1})$ of the occlusion process $\{\alpha_t\}$, and the observation likelihood $p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t)$.

We employ a constant velocity model for the target dynamics $p(\mathbf{X}_t^k|\mathbf{X}_{t-1}^k), k \in \{A, B\}$. In addition, the transition $p(\alpha_t|\alpha_{t-1})$ of the occlusion process is described by a finite state

machine, i.e.,

$$\mathbf{T}_\alpha = [T_\alpha(i,j)] = [p(\alpha_t = j|\alpha_{t-1} = i)].$$

The observation likelihood $p(\mathbf{Z}_t|\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t)$ is modelled based on the innovations, i.e, the discrepancies between the predicted appearance and the actual image observations. Denote the predicted region of the k-th target at time t by $R_t^k = R(\mathbf{X}_t^k)$. Then, the predicted region of $\mathbf{X}_t$ is the union of two targets', i.e.,

$$R_t = R(\mathbf{X}_t) = R((\mathbf{X}_t^A, \mathbf{X}_t^B)) = R_t^A \bigcup R_t^B$$

The actual image appearance observation is collected on the predicted region R_t and denoted by $I(R_t)$. Denote the predicated appearance by $T_t = T(\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t)$ which depends on the value of α_t. As illustrated in Figure 2.2, we denote the overlapping region of the

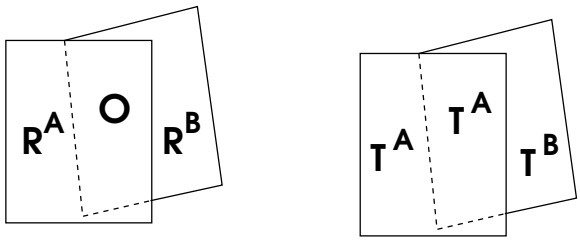


Figure 2.2. The occlusion relations of $\alpha = 1$.

two targets by

$$O_t = O(\mathbf{X}_t) = O((\mathbf{X}_t^A, \mathbf{X}_t^B)) = R_t^A \bigcap R_t^B$$

which is independent of α_t. Then, $\forall u \in R_t$,

$$T_t(u) = \begin{cases} T^A(\mathbf{X}_t^A(u)), & u \in R_t^A - O_t \\ T^B(\mathbf{X}_t^B(u)), & u \in R_t^B - O_t \\ T^C(\mathbf{X}_t^C(u)), & u \in O_t \end{cases}$$

where u is a pixel location in a region, and C indicates the occluding target, i.e.,

$$C = C(\alpha_t) = \begin{cases} \phi, & \alpha_t = 0 \\ A, & \alpha_t = 1 \\ B, & \alpha_t = 2 \end{cases}$$

Then, the observation likelihood is modelled by:

$$p(\mathbf{Z}_t|\mathbf{X}_t, \alpha_t) \propto \exp\left[-\frac{\sum_{u \in R_t} \mathcal{D}(T_t(u), I_t(u))}{M(R_t)}\right] \tag{2.3}$$

where $M(R_t)$ is the number of pixels in the region R_t, and $\mathcal{D}(T_t(u), I_t(u)) = |T_t(u) - I_t(u)|^2$.

Specially attention should be taken for the case where one target is fully occluded by the other one as illustrated in Figure 2.3, since no image evidence can be used to support the existence of the fully occluded target. Consequently, the tracker would not be able to follow the occluded target again. Under this circumstance, the regain of tracking the fully

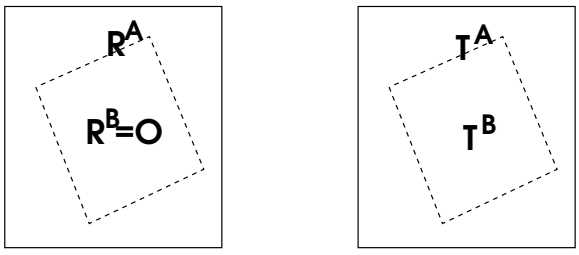


Figure 2.3. Target B is fully occluded by A.

occluded target would depends on motion prediction of target and the detection around the border of the occluding target. Such a mechanism can be implemented by reducing the likelihood of the full occlusion events. Then, we have $p(\mathbf{Z}_t|\mathbf{X}_t, \alpha_t) \propto \exp\left[-H(\mathbf{Z}_t, \mathbf{X}_t, \alpha_t)\right]$, where $H(\mathbf{Z}_t, \mathbf{X}_t, \alpha_t) =$

$$\frac{\sum_{u \in R_t^A} \mathcal{D}(T_t^A(u), I_t(u)) + \sum_{u \in R_t^B} \mathcal{D}(T_t^B(u), I_t(u))}{M(R_t^A) + M(R_t^B)} \tag{2.4}$$

2.4. Sequential Monte Carlo Tracking

The densely-connected structure of the factorized graphical model as shown in Figure 2.1 is complex. The structure variational analysis can be taken to analyze the graphical model [70]. Analytical results of a set of fixed-point equations were obtained based on some simplifications such as linear observation likelihood [50, 70]. In addition, the fixed-point equations reveal a co-inference phenomenon [142]. However, in general, the exact probabilistic inference of the hidden processes would be very difficult especially when the observation likelihood is complicated.

On the other hand, statistical sequential Monte Carlo strategies provide a computational approach to this problem [35, 84, 85], in which a probability density is approximated by a set of weighted particles. The evolution of the set of particles according to the dynamic Bayesian network characterizes the behavior of the dynamic system, and the hidden processes can be recovered from the set of particles. Many particle-based algorithms have been studied for visual tracking [62, 86, 142].

We take a sequential Monte Carlo approach to inferencing the factorized dynamic Bayesian network in Figure 2.1. The posterior density $p(\mathbf{X}_t^A, \mathbf{X}_t^B, \alpha_t|\underline{\mathbf{Z}}_t)$ is represented

by a set of weighted particles $\{x_t^{A,(n)}, x_t^{B,(n)}, \alpha_t^{(n)}, \pi^{(n)}\}$. The sampling-based algorithm is summarized in Figure 2.4.

Generate $\{x_{t+1}^{A,(n)}, x_{t+1}^{B,(n)}, \alpha_{t+1}^{(n)}, \pi_{t+1}^{(n)}\}$ from $\{x_t^{A,(n)}, x_t^{A,(n)}, \alpha_t^{(n)}, \pi_t^{(n)}\}$.

(1) `Re-sampling.` Resample the particle set $\{x_t^{A,(n)}, x_t^{B,(n)}, \alpha_t^{(n)}\}$ to produce $\{x_t'^{A,(n)}, x_t'^{B,(n)}, \alpha_t'^{(n)}\}$ based on $\{\pi_t^{(n)}\}$.

(2) `Prediction.` For each $(x_t'^{A,(n)}, x_t'^{B,(n)}, \alpha_t'^{(n)})$:

(a) sample the density of the target dynamics $p(x_{t+1}^A|x_t^A)$ to produce $x_{t+1}^{A,(n)}$ from $x_t'^{A,(n)}$;

(b) sample the target dynamics $p(x_{t+1}^B|x_t^B)$ to produce $x_{t+1}^{B,(n)}$ from $x_t'^{B,(n)}$;

(c) sample the finite state machine $\mathbf{T}_\alpha$ of $p(\alpha_{t+1}|\alpha_t)$ to produce $\alpha_{t+1}^{(n)}$ from $\alpha_t'^{(n)}$.

(3) `Correction.` Re-weight each particle by calculating the likelihood

$$\pi_{t+1}^{(n)} = p(\mathbf{Z}_{t+1}|x_{t+1}^{A,(n)}, x_{t+1}^{B,(n)}, \alpha_{t+1}^{(n)}).$$

Then normalize all the new weights so that

$$\sum_n \pi_{t+1}^{(n)} = 1.$$

Figure 2.4. The sequential Monte Carlo algorithm for the factorized dynamic Bayesian network in Figure 2.1.

Based on the weighted particle set at each time instant, we obtain the estimation of the hidden states:

$$\begin{aligned} \hat{\mathbf{X}}_t^k &= \sum_n x_t^{k,(n)} \pi_t^{(n)}, \qquad k = \{A, B\}, \\ \hat{\alpha}_t &= arg\max_\alpha \sum_{\alpha_t^{(n)}=\alpha} \pi_t^{(n)}, \quad \alpha = \{0, 1, 2\}. \end{aligned}$$

2.5. Switching Multiple Views

Most appearance-based methods are sensitive to view changes and large deformations, since appearances are view-based. Subspace-based techniques can be employed to learn the appearance-based representations which are robust to views [78] and large appearance changes [9]. These representations are suitable for target detection and recognition, but the dimensionality of the subspace is high for the tracking tasks.

To model view changes, we simplify the subspace-based approaches, and represent a target by maintaining a finite set of view templates, each of which is associated with a transformation, i.e.,$\{(T_1, H_1), \cdots, (T_V, H_V)\}$. Denote an indicator variable by $\beta \in \{1, \cdots, V\}$. Our representation stipulates that the whole set of appearances under different views can be divided into a set of non-overlapped subsets represented by (T_β, H_β). In other words, for any appearance, a unique view template T_β and a suitable transformation exist. This method extends the "view+transoformation" approach to a "switch view+transformation" representation.

This representation is different from subspace representations. In subspace methods, since an appearance is modelled by a linear/nonlinear combination of a set of appearance basis, the methods are global. On the other hand, our "switch view+transformation" approach identifies a specific "mode" (although it is a special case of linear combination), and it is local like a piece-wise spline in the appearance space. Thus, our approach uses a switch β to switch among different "modes" or views templates.

Accommodating this switching view representation in the generative model, the dynamic Bayesian net for a signal target can be illustrated in Figure 2.5, where $\{\beta_t^A\}$ is the

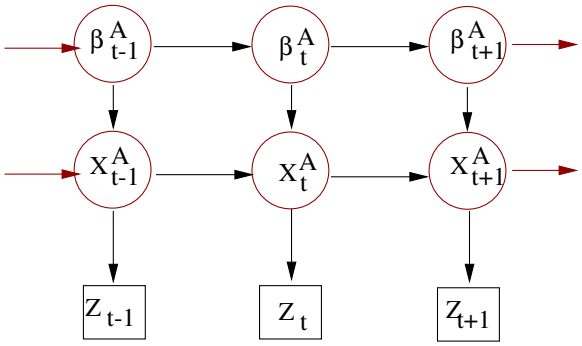


Figure 2.5. A discrete hidden process $\{\beta_t^A\}$ is used to switch among different views of the target A.

hidden process, and we have

$$p(\mathbf{X}_{t+1}^A, \beta_{t+1}^A | \mathbf{X}_t^A, \beta_t^A) = p(\mathbf{X}_{t+1}^A | \mathbf{X}_t^A, \beta_{t+1}^A) p(\beta_{t+1}^A | \beta_t^A)$$

where $p(\mathbf{X}_{t+1}^A | \mathbf{X}_t^A, \beta_{t+1}^A)$ describes the switch of view templates and its dynamics, and $p(\beta_{t+1}^A | \beta_t^A)$ models the transition of the switch event which is stipulated by a finite state machine:

$$\mathbf{T}_\beta^A = [T_\beta^A(i,j)] = \left[p(\beta_t^A = j | \beta_{t-1}^A = i)\right].$$

Although we can perform the structure variational analysis on this graphical model in Figure 2.5 (see [100]), a more flexible approach for inferencing is again the sequential Monte Carlo strategies. Similar to the mixed-state CONDENSATION [64], a particle for the target A is represented as $\{x_t^{A,(n)}, \beta_t^{A,(n)}, \pi_t^{(n)}\}$. The evolution of the set of particles is

generated by the dynamic Bayesian net model. The estimate of the view is given by:

$$\hat{\beta}_t^A = \arg\max_{\beta} \sum_{\beta_t^{(n)}=\beta} \pi_t^{(n)}; \tag{2.5}$$

$$\hat{\mathbf{X}}_t^A = \sum_{\beta_t^{(n)}=\hat{\beta}_t^A} \pi_t^{(n)} x_t^{A,(n)} \pi_t^{(n)}. \tag{2.6}$$

Naturally, the combination of the occlusion model in Figure 2.1 and the model for switching views in Figure 2.5 results in a new dynamic Bayesian network as illustrated in Figure 2.6, which models the occlusion of multiple targets as well as multiple views. Taking the sequential Monte Carol methods similar as those in previous sections, the

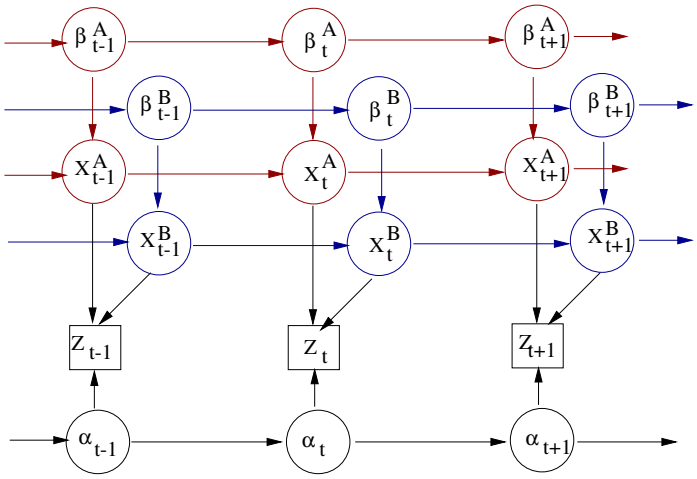


Figure 2.6. A hidden process $\{\alpha_t\}$ controls the occlusion relations among different targets and $\{\beta_t^k\}$ switches among different views for the k-th target, where $k \in \{A, B\}$.

inference of this dynamic Bayesian net is straightforward.

2.6. Experiments

The proposed methods have been applied to the task of tracking two moving and occluding faces. We report the experiments in three tracking scenarios including occlusion, view changes and the combination of the two.

Our first experiment are concerned about the inference of occlusions induced by the interaction of two targets, and the generative model in Figure 2.1 applies. In this case, the appearance of a face is represented by a single pre-trained view template of the face and an affine transformation. The tracking task is to estimate the affine parameters for both templates as well as the occlusion relation when the two faces cross. We have employed two types of view templates: one is the image template, and the other is the texture template based on wavelet transformations.

Since the overlapping can be directly calculated once $\mathbf{X}_t^{A,(n)}$ and $\mathbf{X}_t^{B,(n)}$ are given, the uncertainty remained for occlusion variable α_t is either $\alpha_t = 1$ or $\alpha = 2$. Then the transition of $\{\alpha_t\}$ is reduced to a two-state machine. In the experiment, we set

$$\mathbf{T}_\alpha = p(\alpha_j|\alpha_i) = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \quad i, j \in \{1, 2\}. \tag{2.7}$$

The tracking results can be seen in the sequence named "`occlusion.mpg`" with this chapter. Some sample frames of the tracking results are shown in Figure 2.7. In this experiment, the size of the particle set is 2000. When the two faces do not overlap, the tracker acts like two independent trackers. When the two faces across, the tracker proved to keep locking on the two faces with the right identities, because the occlusion relation is recovered during tracking, which greatly helps to maintain the identities of different

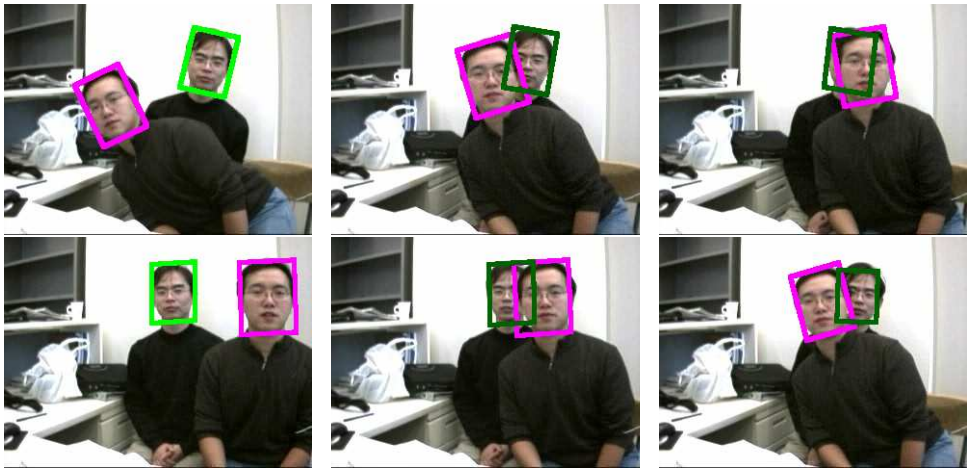


Figure 2.7. Two faces are tracked (in red or green) during the occlusion. One becomes dark if occluded. Their occlusion relations are inferred and the identities of the two faces are maintained. (See "`occlusion.mpg`" for detail.)

targets. The occlusion is estimated by maximizing the *a posteriori* in Equation 2.5. The recovered occlusion process $\{\alpha_t\}$ is shown in Figure 2.8. The estimates of the occlusion

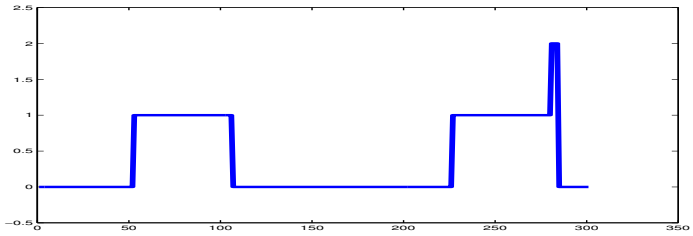


Figure 2.8. The recovered occlusion process $\{\alpha_t\}$.

relations are quite accurate, except for the frames where the occlusion is about to occur or about to finish. But this phenomenon is reasonable since the occlusion relations are weak and uncertain at those time instants. Since a face goes back and forth in front of the other face in the sequence, the occlusion events $\alpha = 1$ occur in two time intervals. This is clearly indicated in Figure 2.8.

The second experiment is about the multiple view model, and the generative model in Figure 2.5 applies. The task is to track a single face but the motion of the face contains out-plane rotation, which results in multiple distinguishable views. In this experiment, we exploit three view templates: one front view, and two profile views, with three Homography transformations associated with each template. The three templates are shown in Figure 2.9. Here, $\beta = 1/2/3$ denotes left profile, front and right profile views, respectively.

Figure 2.9. The three view templates used for the multiple appearances switching.

The transition of the view switching process $\{\beta_t\}$ is a three-state FSM:

$$\mathbf{T}_\beta = p(\beta_j|\beta_i) = \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.15 & 0.8 \end{bmatrix}. \tag{2.8}$$

The result for the single face with multiple views is demonstrated in the sequence "`multiview.mpg`". Some sample frames are shown in Figure 2.10. The size of the particle set in the sequential Monte Carlo inference is 1000. When the face turns, the correct view template is automatically selected and the tracker can switch to this view template and keep tracking. Since the particle set represents the density, it implicitly keeps all the view hypotheses and the priors of these hypotheses are propagated from previous time instants. The calculation of the likelihood of the image observation given these view hypotheses can strengthen or weaken these hypotheses. The one with the maximum posterior probability

Figure 2.10. Tracking one face with out-plane rotations with the switching multiple view model. A suitable appearance template is selected automatically at each time instant. (See "`multiview.mpg`" for detail.)

is selected as the estimation of the view template "mode" at each time instant. The recovered process of mode switching is shown in Figure 2.11. We see clearly from this

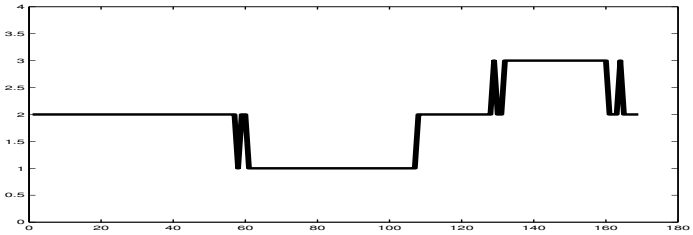


Figure 2.11. The recovered switching process $\{\beta_t\}$.

figure that the person turns his head around when he moves.

In the third experiment, we track two faces under occlusion and multiple views, and the method in Figure 2.6 applies. The same as the second experiment, we use a three-view templates with Homography transformations. And $\mathbf{T}_\alpha$ uses Equation 2.7, and $\mathbf{T}_\beta^A$ and $\mathbf{T}_\beta^B$ are set the same as Equation 2.8.

The sequence "occlu_multiview.mpg" demonstrates the tracking result for the single face with multiple views. Some sample frames are shown in Figure 2.12. Due to the

Figure 2.12. Two faces move across inducing occlusion, and the motion of the faces contains out-plane rotations. The occlusion (the occluded one is shown in dark) are inferred and the suitable view templates are switched. (See "occlu_multiview.mpg" for detail.)

complexity of the dynamic Bayesian net in Figure 2.6 used in this experiment, more particles are needed for effective Monte Carlo. We use 4000 particles to obtain the result. By accommodating the processes of occlusion and view switching, the tracker needs to infer more hidden factors based on the image observations, thus more computation is involved. But the payoff is huge: the tracker becomes more robust and the recovered hidden factors provide quantitative clues for evaluating the tracking performance online.

2.7. Discussions

Appearance-based tracking is useful in many applications such as face tracking, but is confronted by the problem of occlusion, especially when multiple appearances are concerned. This chapter presents a generative model to accommodate a hidden process of occlusion relations among multiple targets. The likelihood of the image observation is conditioned on the configuration of the states of multiple appearances as well as an occlusion relation among them. Graphically, such a generative model is a factorized dynamic Bayesian network with multiple hidden Markov chains. In addition, this chapter also presents a multiple view representation for appearances by a "switch view+transformation" approach. Accommodating multiple views in the dynamic Bayesian network results in a mode-switch model. The inference of the hidden processes is made possible through particle-based sequential Monte Carlo methods, by which the the mode and transformations of different appearances as well as their occlusion relations can be recovered.

The generative models explicitly represent the hidden factors which affect the image observations, thus the recovery of these hidden factors would provide significant interpretation of the image sequences besides tracking. Since analytical results are in general hard to obtain, when more factors are included in the generative model, the computational complexity tends to be more tremendous. Thus, more efficient Monte Carlo methods should be developed to ease these computational issues. In addition, instead of presetting the parameters in the models, learning these parameters from training data would be more plausible. Our future work will include these two issues.

CHAPTER 3

Multiple Target Tracking: A Decentralized Solution

3.1. Introduction

Video based multiple target tracking is an important problem in many emerging applications, such as for intelligent video surveillance systems where robustly tracking multiple persons is a critical step towards some higher level processing and analysis like action recognition and abnormal event detection [119], for sports video analysis where automatically tracking multiple athletes in the field can help coaches for decision making and performance analysis [145], and for video conferencing where effective low bit rate video communication requires the accurate localization and tracking of all attendees.

If the targets demonstrate quite distinctive appearances from each other, they may be robustly tracked by simply instantiating multiple independent trackers (M.i.T.) with least confusion, since the strong discriminative image cues can provide reliable localization power for each target through the likelihood calculations. However, many real application scenarios prevent such a simple M.i.T. solution, where the targets may present more or less the same appearances. For example, when tracking multiple soccer players in a field, since all players are in the uniform sports wears, due to the appearance confusions, the observation models of M.i.T. tracker becomes less effective, which leads to the difficulty of maintaining the correct associations between the image observations with their corresponding soccer player trajectories. Therefore, a major difficulty of multiple target

tracking lies in the fact that the trackers might be insensitive to the differences among the targets such that they may not be distinguishable from each other, which leads to a combinatorial problem on target-tracker association. We can call it as the "identical" targets problem, where "identical" targets imply that the tracked targets are indistinguishable from each other in terms of the tracker observation model. The neglect of this problem (e.g., by using M.i.T.) will generally lead to the tracker coalescence phenomenon [14], i.e., several trackers are associated to one same target while other targets lose track. Coalescence often takes place especially when the targets are close or present occlusions [73, 151].

As a single target tracker, CONDENSATION [62, 12] or particle filtering may also be used for multiple targets, since it can estimate the non-Gaussian posterior density of the targets, where the potential multi-modes may imply the presence of multiple targets. However, when the posterior distribution is propagating over frames by particles, it is likely that the targets that attract more particles will dominate the dispersion of particles, which will gradually reduce the number of particles for the targets that have weaker image observations, and finally lose track of them, i.e., the coalescence problem also happens in this case. Aiming to handle such a problem, a variant of particle filtering algorithm was proposed by [127], which is further extended by [95], to continuously maintain multiple modality of the posteriori distribution using the particle set over time, which may effectively handle the coalescence problem. However, since it is embedded into the single particle filtering framework, it lacks a consistent way to resolve the target identity ambiguities that arise in associating targets with the image observations.

Most existing solutions to this problem are based on the centralized methodology by considering joint data association that enumerates all the possible associations between targets and observations. Various methods along this line have been developed (See Section 2 for details). The essence of their methods is the introduction of the joint state space representation which concatenates together all the state spaces of the individual targets such that they can be jointly inferred based on the exhaustive data association enumerations between targets and image observations. Our centralized solution to the multiple target tracking problem introduced in the previous chapter also belongs to this joint state space category [138]. The coalescence problem may be correctly handled during the joint inference. However, due to the exploring of a high-dimensional joint state space, these methods are generally computational intensive. For example, the multiple hypothesis tracker (MHT) [28, 55] and the joint probabilistic data association filter (JPDAF) [6, 17, 74, 107] have to evaluate all possible associations which suffers from the combinatorial complexity, and the sampling-based methods [60, 63, 86, 123, 138, 160] are confronted by the exponential demand of the increase of particles.

Such centralized solutions might be fine to a powerful processor. However, they may not be appropriate for the emerging application of sensor networks, in which there are a large number of sensing units that have the functionality of sensing, computing and communicating. However, these units are power-limited to prevent much computation and communication [77]. Thus, to make good use of such sensor networks for target tracking, complex computation must be distributed into the network, since once a certain unit takes charge of sensing, its computational load on target tracking needs to be migrated

to other idle units. Although this research is being carried out at the computer architecture level, it is more desirable to find a decentralized scheme at the algorithm-level for efficient tracking of multiple targets, since it will leads to the essential parallelization and distributed computing.

In this chapter, we present a new and linear complexity decentralized framework for visual tracking multiple "identical" targets with coalescence problem handling. The basic idea is a distributed while collaborative inference mechanism, where the motion state of each target, estimated by the tracker covering it, is not only determined by its own observation and dynamics, but also through the interaction and collaboration with the state estimates of its adjacent targets, which leads to a competition mechanism that enables the set of casted trackers to compete for the common image resources, i.e., image observations, to support the motion estimates of their covering targets in the video scene.

The theoretical foundation of this new approach is based on Markov networks, in which each hidden node in the network represents the motion state of an individual target, captured by a tracker, and the links in the network correlate a tracker to those who compete image observations against it. The structure of the Markov network can change according to the spatial relations of the trackers during the tracking process. We call it an *ad hoc Markov network*. Since such a Markov network is likely to contain loops, variational analysis is employed and reveals a mean field approximation to the posteriors of the trackers, therefore it provides a computationally efficient way to this difficult inference problem. In addition, we design a sequential Monte Carlo algorithm that efficiently implements this mean field inference by simulating the competition among a set of low dimensional particle filters.

In order to track the variable number of targets, the proposed decentralized framework is further extended to allow the set of trackers running autonomously and collaboratively by equipping each tracker an entropy-based evaluator, where the individual trackers are autonomous in the sense that they have the freedom to select targets to track and evaluate themselves. The trackers then can be either in active or inactive status, indicating if they are currently follow targets or not. Motion estimates of the targets from this set of autonomous trackers are still distributed into a specifically designed Markov network, where the collaboration mechanism of these netted trackers is again achieved by the information exchanges of these trackers through the variational analysis of the Markov network.

In addition, under the tracking variable number of targets scenario, a roughly trained target detector [128] is equipped to each autonomous tracker to help sense the potential newly appearing targets in the dynamic scene, therefore background subtraction is not necessary to our method, although it can largely help the case of fixed backgrounds. The use of object detectors within each tracker also supports the construction of an effective importance function, which leads to a more effective variational inference.

With linear complexity in terms of the number of targets, the new approach copes with multiple target tracking in a distributed and collaborative fashion. The competition mechanism introduced by the collaborative inference mathematically incorporates the essence of joint data association where one single observation cannot support more than one target, therefore the coalescence problem can be naturally handled. Compared with the existing solutions, the new decentralized approach stands out by its effectiveness and low computational cost to the coalescence problem. The approach also shows its ability to

continuously track variable number targets as long as they are physically existing within the video scene. Extensive experiments on the challenging video sequences are conducted to demonstrate the effectiveness and efficiency of the proposed method.

The chapter is organized as follows. In the next section, we briefly review the existing multiple targets tracking approaches. Section 3.3 presents our decentralized multiple target tracking framework based on Markov network formulation in details. The autonomous trackers with entropy-based self-evaluators are introduced in Section 3.4, which enables the framework to achieve tracking variable number of targets in the video scene. Section 3.5 presents the variational analysis of the proposed Markov networks, where loopy structure may present. Based on the mean field fixed point equations derived from the variational analysis in Section 3.5, Section 3.6 introduces the sequential Monte Carlo implementation to approximate the variational inference from the potential loopy Markov network, where an importance sampling mechanism is also presented, when a roughly trained target detector is available. Extensive experiments on various synthetic and real video sequences are reported in Section 3.7. Finally, Section 3.8 summarizes the chapter and makes a number of suggestions for future research.

3.2. Related Work

Many multiple target tracking methods have been developed during the past few years, where most of them are based on the centralized joint state space inference either under the parametric or non-parametric formulations. The parametric methods, such as multiple hypothesis tracking (MHT) [28, 55] and joint probabilistic data association filtering (JPDAF) [107] handle the coalescence problem by the joint data association

principle in which one image observation can only support a single target hypothesis and one target hypothesis can only occupy a single observation, therefore suffering from the combinatorial complexity due to the exhaustive enumeration for all possible associations. Based on Monte Carlo sampling techniques, non-parametric methods [60, 63, 86, 123, 160] can tackle the coalescence problem in a top-down process that generates and evaluates a large number of hypotheses, thus also confronted by a similar high computational cost due to the exponential demand of the increase of particles. All these approaches are actually dealing with the centralized state space directly, which results in the inevitable combinatorial or exponential complexity in the algorithm level that is hardly scalable.

The existing approaches can also be classified into two categories according to whether a fixed background model is employed. Background subtraction normally offers a strong localization clue for detecting each new target entering the scene. Whenever a new target is appearing, a new tracker can be immediately instantiated to follow it [56]. The fixed background assumption is also the essential reason why the configuration level optimization techniques, such as jump-diffusion Markov chain Monte Carlo in [160] and variants of particle filtering [63, 123], can be applied to inference the number of targets existing in the scene over the union of joint state space of multiple targets, since under this assumption the observation likelihood can be calculated based on the whole image information. Foreground area is evaluated by the foreground target model, background area is also assessed by the maintained background model, which in combination makes the configuration level reasoning feasible. However, this nice property does not exist under the changing background situations, since there is generally no way to maintain a powerful

background model to explain all non-target areas in the dynamic scene. Therefore existing approaches dealing with the dynamic background scenarios are either assuming to track fixed number of targets [60,86,107,151], which obviously limits its generalization, or adopting an target detector to help determine if any new target appears in the scene [95]. However, in [95] it stays unclear how the coalescence problem can be reliably solved under their single particle filtering framework.

Different from these existing methods, we propose a decentralized approach to multiple target tracking with linear computational complexity. Based on this new formulation, a collaborative sequential Monte Carlo algorithm is proposed to allow a set of low dimensional particle filters compete against each other to solve the coalescence problem. Furthermore, by using a set of collaborative autonomous trackers, our approach is also capable of tracking variable number of targets. Compared to the state-of-the-art, this new approach proves to be computationally efficient and algorithm-level parallelizable.

3.3. The Decentralized Representation

In this section, we mainly focus on the model descriptions for tracking fixed number of targets. The discussions of extending the model to track variable number of targets will be delayed to the next section.

Suppose M trackers are casted into the video scene to track M targets. We denote the state of an individual tracker by $\mathbf{x}_i$, the joint state by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ for M trackers, the image observation of $\mathbf{x}_i$ by $\mathbf{z}_i$, and the joint observation by $\mathbf{Z}$.

3.3.1. Conditional Dependency

When multiple targets move close or present occlusions, it is generally difficult to distinguish and segment these spatially adjacent targets from image observations, thus we can not simply factorize the joint image observation, i.e., $p(\mathbf{Z}) \neq \prod_i p(\mathbf{z}_i)$.

As a result, the image observations under this circumstance have to be treated as they are jointly produced by all these targets, i.e., we need to model the joint likelihood $p(\mathbf{Z}|\mathbf{x}_1, \ldots, \mathbf{x}_M)$. In this case, when the joint image observation is given, the posteriors of different targets are conditionally dependent, i.e.,

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_M|\mathbf{Z}) \neq \prod_i p(\mathbf{x}_i|\mathbf{Z}).$$

This conditional dependency of multiple targets is the root of the reason why M.i.T. and CONDENSATION can not cope with the coalescence problem. It also makes clear why the centralized methods that deal with the joint state space are confronted by the high dimensionality, since they have to model $p(\mathbf{Z}|\mathbf{X})$ as a centralized entity.

We present in the next sections a new decentralized model to cope with this problem with linear complexity and a distributed while collaborative algorithm is developed for tracking multiple identical targets.

3.3.2. A Tracker Markov Network Formulation

Since the motions of the multiple targets become dependent when they are spatially adjacent, we can consider to model the prior of the joint target states, i.e., $p(\mathbf{X})$. This prior can be very complicated due to the unknown correlations, but we can approximate

it by a Gibbs distribution in general. Here we present a specific Gibbs model which leads to a theoretically plausible and practically efficient tracking algorithm.

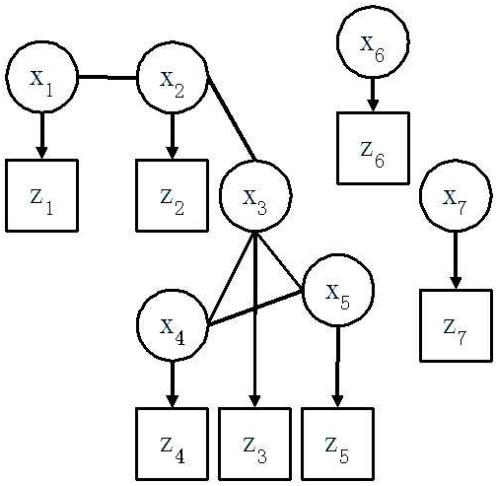


Figure 3.1. The Markov Network for multiple targets.

The theoretical foundation of the new approach is based on Markov networks, as shown in Figure 3.1, which consists of two layers. The hidden layer is an undirected graph $G_x = \{V, E\}$ where each circle node represents the state or motion parameters (such as an affine motion) of a tracker $\mathbf{x}_i$, and the pair-wise link between a pair of trackers represents the motion correlation (of dependency) between them (as described below). In addition, the observable layer includes square nodes that represent the image observations and are individually associated with their corresponding hidden nodes. A directed link from the tracker $\mathbf{x}_i$ to its local image observation $\mathbf{z}_i$ represents the observation likelihood $p(\mathbf{z}_i|\mathbf{x}_i)$. Since the local observation $\mathbf{z}_i$ conditionally independent of others given $\mathbf{x}_i$, we have:

$$p(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{n} p_i(\mathbf{z_i}|\mathbf{x_i}). \tag{3.1}$$

The core problem here is to infer the posterior $p(\mathbf{X}|\mathbf{Z})$.

The structure of the graph in the hidden layer depends on the spatial relations among the trackers' states. At each time instant t, the structure of the tracker network is determined according to the relative positions of the trackers, calculated by the conditional mean state estimator $\overline{\mathbf{x}}_t = \int \mathbf{x}_t p(\mathbf{x}_t|\underline{\mathbf{z}}_t) d\mathbf{x}_t$. The tracker that is not close to others is represented by an isolated vertex in the graph (such as $\mathbf{x}_6$ and $\mathbf{x}_7$ in Figure 3.1). If two trackers are close enough (in the sense the the specific image observer or detector used for tracking is unable to separate their image observations), there is an undirected link between them in the graph to represent their motion dependency (such a $\mathbf{x}_3$ and $\mathbf{x}_4$ in Figure 3.1), and a potential function is associated with this link to parameterize the motion correlation. For example, when several trackers are close, they are generally competing targets for tracking, thus such motion correlation terms enforce an exclusive constraint that a target must not be associated to more than one tracker. It is worth mentioning that the pair-wise state constraint model has also been successfully applied in articulated hand tracking based on a similar Markov network formulation [122].

Since the trackers are moving around to follow the targets, their spatial relations change with time and the structure of the Markov network also change with time. Once the spatial relations of the trackers are roughly determined, the structure of the network is fixed. The neighborhood of a tracker is those that are linked with it, and we denote the neighborhood of $\mathbf{x}_i$ by $\mathcal{N}(i)$.

In this formulation, the prior $p(\mathbf{X})$ is modelled as a Gibbs distribution and can be factorized as:

$$p(\mathbf{X}) = \frac{1}{Z_c} \prod_{c \in \mathcal{C}} \psi_c(X_c) \tag{3.2}$$

where c is a clique in the set of cliques $\mathcal{C}$ in the undirected graph, X_c is the set of hidden nodes associated with the clique and $\psi_c(X_c)$ is the potential function of this clique, and Z_c is a normalization term or the partition function. Our model allows two types of cliques: the first order clique, i.e., $i \in V$, and second order clique, i.e., $(i,j) \in E$, where $\mathcal{C} = V \bigcup E$. The associated potential function ψ_c is denoted by ψ_i and ψ_{ij}, respectively. Thus, Eq. 3.2 can also be written as:

$$p(\mathbf{X}) = \frac{1}{Z_c} \prod_{(i,j)\in E} \psi_{ij}(\mathbf{x_i}, \mathbf{x_j}) \prod_{i \in V} \psi_i(\mathbf{x_i}) \tag{3.3}$$

where $\psi_i(\mathbf{x_i})$ provides a local prior for $\mathbf{x}_i$ which can be the dynamics prior or the prior given by other modalities, and $\psi_{ij}(\mathbf{x_i}, \mathbf{x_j})$ presents the motion dependency between neighborhood nodes $\mathbf{x}_i$ and $\mathbf{x}_j$.

It is critical to model the motion dependency or correlation mentioned above. The motion of two trackers become dependent *a posteriori* only because their image observations can not be separated. But when one tracker has been associated with part of the total image observations, the other tracker can only obtain the rest of the observations, since the same piece of image evidence can not support the existence of two different trackers. Therefore, we can approximate the motion dependency of them by a *competition* correlation, i.e., trackers compete against each other for the common image resources. In other words, if one tracker occupies a region in the state space, it will lower the probability of others to occupy the same region. As a specific example, the competition potential function can be modelled as:

$$\psi_{ij}(\mathbf{x_i}, \mathbf{x_j}) \propto e^{d(\mathbf{x}_i, \mathbf{x}_j)^T \Sigma^{-1} d(\mathbf{x}_i, \mathbf{x}_j)} \tag{3.4}$$

where $d(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i - \mathbf{x}_j$ is the distance between the two trackers in the state space, and Σ characterizes the size of competition region in the state space. The intuition behind Eq. 3.4 is that it favors the situation where the states of the two trackers are not close to each other.

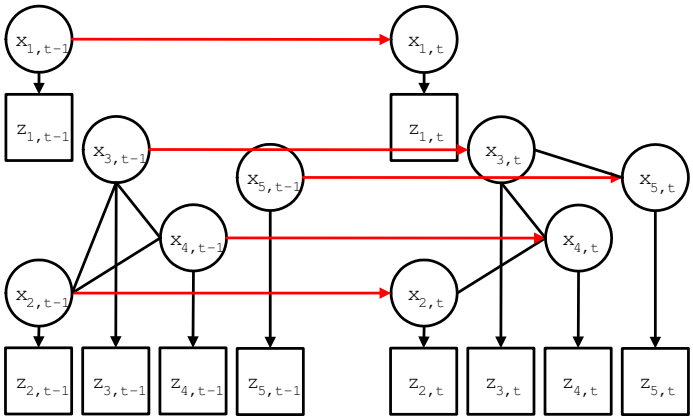


Figure 3.2. Dynamic Markov Network for multiple targets.

Putting the above Markov network in the temporal context by accommodating the dynamics model $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ for each tracker, we can model the visual dynamics of multiple trackers in a more complicated graphical model, which can be called as a dynamic Markov Network, as shown in Figure 3.2. In this figure, the red arrows connecting the trackers in two consecutive frames represent the dynamic prior propagation to model the local prior in the Markov network, i.e., $\psi_i(\mathbf{x}_{i,t}) \propto p(\mathbf{x}_{i,t}|\underline{\mathbf{Z}}_{t-1})$. Note that the structures of the Markov networks in two consecutive time frames are a little different, which illustrates the changes of motion correlations among the trackers, due to the change of the spatial relations among them.

In all the notations, the subscript t represents the time index. In addition, we denote the collection of all the image observation up to time t by $\underline{\mathbf{Z}}_t = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_t\}$. In this

formulation, the multiple target tracking problem is to infer the posterior of each target $p(\mathbf{x}_{i,t}|\underline{\mathbf{Z}}_t)$, which will be solved in Section 3.5.

3.4. Autonomous Trackers for Tracking Variable Number of Targets

Tracking variable number of targets requires either a configuration level optimization over the variable dimensional state-space, or a fixed dimensional state-space in conjunction with a set of indicator variables showing which components from the state-space correspond to active trackers (i.e., the trackers that are currently following the targets, which are physically existing in the video scene.). Since our decentralized model prevents a direct estimation of the number of targets, we adopt the second approach, because it deals with fixed dimensional space, and fits into our previously presented Markov network formulation with only some minor modifications.

As before, we denote the motion state of each individual tracker at time t by $\mathbf{x}_{i,t}$, and its associated image observation by $\mathbf{z}_{i,t}$. However, now we allow individual trackers move autonomously, and are able to evaluate themselves. They need to determine and switch the modes (i.e., active or inactive) by themselves. We denote by $\mathbf{r}_{i,t}$ the binary performance indicator for each tracker, and let it be part of the tracker state variables, where $\mathbf{r}_{i,t} = 1$ means active status and vice versa. The details of tracker self-evaluation is given in Section 3.4.1.

A similar Markov network formulation for this set of autonomous trackers is thus illustrated in Figure 3.3, which in comparison with the Markov network in Figure 3.1 clearly shows the difference that some of the nodes are weakly dotted, and correspond to

those inactive trackers, while the solid nodes stay the same as before corresponding to the set of active trackers.

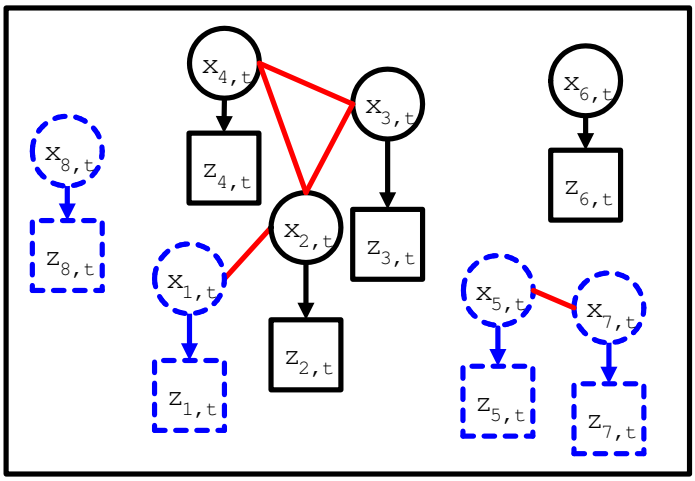


Figure 3.3. Autonomous and collaborative trackers as a Markov network for tracking variable number of targets.

The competition constraints still apply to the individual trackers no matter whether they are currently active or inactive. For example, when the competition presents among active trackers, such a potential term acts to overcome the coalescence problem as described before. When the competition presents among inactive trackers, this potential term helps to force these inactive trackers to search different image regions for newly appearing targets to track. When the competition happens between an active tracker and an inactive one, such an elastically exclusive force becomes unidirectional, i.e. only the active tracker can exclude the inactive one to prevent the case where the inactive tracker "hijacks" the target being tracked by the active one.

3.4.1. Self-awareness and Mode Switching

Based on the inference result $p(\mathbf{x}_t|\underline{\mathbf{z}}_t)$, there may exist different ways to obtain a performance indicator for each tracker. We observe that when each single tracker is experiencing good tracking conditions, the underlying posterior $p(\mathbf{x}_t|\underline{\mathbf{z}}_t)$ will mainly demonstrate some sharp unimode distribution. On the contrary, a more uniform distribution implies larger uncertainty of the motion estimation, i.e., the tracking result is less confident and thus not satisfactory. Therefore, an entropy measure can be used as a good performance metric to evaluate the tracking performance. Specifically, we define the performance indicator $\mathbf{r}_t$ as follows:

$$\mathbf{r}_t = \begin{cases} 1 & \text{if } -\int p(\mathbf{x}_t|\underline{\mathbf{z}}_t)\mathbf{log}p(\mathbf{x}_t|\underline{\mathbf{z}}_t) < \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where $\mathbf{r}_t = 1$ indicates active trackers since the target seems to be successfully followed by the tracker, while $\mathbf{r}_t = 0$ implies inactive trackers otherwise, such as the tracker loses track due to the interferences from the cluttered background or simply because the previously tracked target leaves the video scene. The threshold τ can be empirically determined.

The modes of an autonomous tracker determine its dynamics model $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ (where i indexes the tracker). Thus a track can switch its behaviors autonomously based on the active or inactive mode determined by itself:

$$p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) = \begin{cases} p_a(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) & \text{if } \mathbf{r}_{i,t-1} = 1 \\ p_u(\mathbf{x}_{i,t}|\overline{\mathbf{x}}_{i,t-1}) & \text{otherwise} \end{cases} \tag{3.6}$$

where $p_a(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ is a constant acceleration motion model, and $p_u(\mathbf{x}_{i,t}|\overline{\mathbf{x}}_{i,t-1})$ is an uninformative uniformly random walk around the tracker's previous conditional mean state estimator $\overline{\mathbf{x}}_{i,t-1}$.

Camouflages may affect the tracking performance significantly, no matter whether these camouflages arise from the same types of targets in the nearby region or simply due to the background clutters that resemble the target. Thus, it is not robust when tracking one target. This difficulty can be largely alleviated by the joint tracking of multiple similar targets since the joint data association can largely reduce the risk of loss track of any. Casting this idea into a decentralized methodology, we believe the collaborations among individual trackers act as a distributed way for data association.

3.5. Variational Inference and Decentralization

Belief propagation [44] is generally used to obtain exact Bayesian inference for non-loopy Markov networks. However, this method may not be appropriate for analyzing the Markov networks such as shown in Figure 3.1 and Figure 3.3 introduced in the previous sections for multiple target tracking, because these Markov networks are likely to contain loops when three or more targets are linked together. In contrast to belief propagation, variational analysis methods [70,66,135] are more flexible to the structure of the network. Although only the approximate inference can be obtained, they provide lower bounds of the approximation as a theoretical benefit. Thus we perform variational analysis for the above Markov networks in this section. For clarity, we first analyze the static Markov network and then generalize the results to the dynamic Markov network.

The fundamental idea of the probabilistic variational method is the employment of a variational distribution $Q(\mathbf{X})$ with variational parameters as a variation of the density we want to infer, e.g., the posterior $p(\mathbf{X}|\mathbf{Z})$ in our case. Variational analysis aims at finding the optimal variational distribution $Q^*(\mathbf{X})$ that minimizes the Kullback-Leibler (KL) divergence between them, i.e.,

$$Q^*(\mathbf{X}) = \arg\min_{Q} KL(Q(\mathbf{X})||p(\mathbf{X}|\mathbf{Z})) \tag{3.7}$$

This is feasible when the appropriate forms of the variational densities are adopted. For simplicity, a fully factorized form is usually employed, i.e.,

$$Q(\mathbf{X}) = \prod_{i}^{M} Q_i(\mathbf{x}_i) \tag{3.8}$$

where $Q_i(\mathbf{x_i})$ is an independent distribution of the hidden node $\mathbf{x}_i$. Since Q_i has to be a probability density function, this becomes a constrained optimization problem with the following Lagrangian for each Q_i:

$$L(Q_i) = KL(Q_i) + \lambda(\int_{x_i} Q_i - 1) \tag{3.9}$$

When using the Gibbs model for $p(\mathbf{X})$ in Eq. 3.3, it is easy to show the solution is a set of fixed point equations [151]:

$$Q_i(\mathbf{x}_i) \longleftarrow \frac{1}{Z_i'} p_i(\mathbf{z}_i|\mathbf{x}_i)\psi_i(\mathbf{x}_i)M_i(\mathbf{x}_i), \quad \text{where}$$

$$M_i(\mathbf{x}_i) = \exp\{\sum_{k\in\mathcal{N}(i)} \int_{x_k} Q_k(\mathbf{x}_k) \log \psi_{ik}(\mathbf{x}_i, \mathbf{x}_k)\}, \tag{3.10}$$

where Z_i' is a constant, and $\mathcal{N}(i)$ is the neighborhood of the subpart i, and $i = \{1, \ldots, M\}$. The iterative updating of $Q_i(\mathbf{x}_i)$ decreases the KL-divergence and reaches an equilibrium. These fixed point equations are called *mean field equations*.

The same procedure can also be applied to the dynamic Markov network when tracker temporal dynamics model is available, and the mean field equations can be derived:

$$\begin{aligned} Q_{i,t}(\mathbf{x}_{i,t}) \quad \longleftarrow \quad & \frac{1}{Z_i'} p_i(\mathbf{z}_{i,t}|\mathbf{x}_{i,t}) \\ \times \quad & \int p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) Q_{i,t-1}(\mathbf{x}_i) \\ \times \quad & M_{i,t}(\mathbf{x}_{i,t}) \end{aligned} \tag{3.11}$$

where the second term $\int p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) Q_{i,t-1}(\mathbf{x}_i)$ is actually very similar to the dynamics prediction prior, i.e., $p(\mathbf{x}_{i,t}|\underline{\mathbf{Z}}_{t-1})$.

The mean field equations are very meaningful, since they reveal a collaborative solution to the very difficult Bayesian inference problem: the posterior of a target $\mathbf{x}_i$ is not only determined by its local prior $\psi_i(\mathbf{x}_i)$ (such as the dynamics prediction prior) and its local image likelihood $p_i(\mathbf{z}_i|\mathbf{x}_i)$, but also the beliefs of its neighborhood targets that compete image resources against it. The influence of the competition is summarized in the "message" term, as defined in Eq. 3.10, that is passed to $\mathbf{x}_i$ during the mean field iterations.

In Eq. 3.11, it is clear that the basic computation unit is the posterior estimation for each individual tracker, therefore, the computationally demanding tracking task has been decentralized to the set of netted trackers with the cost of communication and collaboration. The computational complexity of the collaborative tracker is easily figured

out to be linear with respect to the number of trackers and the number of iterations, which is a significant improvement of the methods that deal with the joint state space directly.

During collaborative tracking, when the competition mechanism takes place, the distribution of the trackers that are unlikely to win the competition will be diffused around the tracker that is likely to win, until other image observations become available in the future. Once some trackers do not compete, i.e., without motion correlation, their image observations can be readily separated and thus these trackers will be tracked independently. At this time, the collaborative tracker acts as the same as M.i.T..

3.6. Sequential Monte Carlo Implementation

Since the image observation likelihoods are generally non-Gaussian due to the presence of clutters for example, it is not plausible to express the mean field equations in parametric forms by assuming all the densities are Gaussian. Thus, we describe in this section a sampling-based sequential Monte Carlo implementation of the mean field inference.

A set of particle is employed to represent the variational density $Q_i(\mathbf{x}_i)$ for each tracker $\mathbf{x}_i$, i.e.,

$$q_i^k(\mathbf{x}_i) \sim \{s_i^{(n)}(k), \pi_i^{(n)}(k)\}_{n=1}^N$$

where s and π denote the sample and its weight and N is the number of samples. Based on Eq. 3.11, the Monte Carlo can be summarized as in Figure 3.4.

An equilibrium will be reached after several iterations. Then the optimal variational distributions $Q_{i,t}(\mathbf{x}_{i,t})$ can be treated as the approximation to the posterior $p(\mathbf{x}_{i,t}|\underline{\mathbf{Z}_t})$. In general, mean field equations converge very quickly due to the nature of the fixed point.

Although we have not obtained the rigorous results on the convergence rate, we always observe the convergence in less than five iterations in our experiments.

The significance of the above tracking algorithm is its distributed and collaborative mechanism, where each individual tracker is associated with a particle filter. These particle filters are not independent but competitive through message passing and mean field iterations.

(1) Structure Determination of Markov Network:
At time t, determine the Markov network structure according to the relative positions and performance indicators of the trackers from time $t-1$.
(2) Particle Re-Sampling:
Resample $Q_i(\mathbf{x}_{i,t-1})$ for $\{\widetilde{s}_{i,t-1}^{(n)}, 1\}_{n=1}^{N}$.
(3) Dynamics Propagation:
$\forall\ \widetilde{s}_{i,t-1}^{(n)}$, sample $s_{i,t}^{(n)}$ from $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$.
(4) Observation Likelihood Calculation:
For each $s_{i,t}^{(n)}$, perform likelihood calculation
$w_{i,t}^{(n)} = p(z_{i,t}|s_{i,t}^{(n)})$
(5) Iteration: Initially set $k=0$ and $k=k+1$:
(a) calculate the "message" from neighbors:

$$m_{i,t}^{(n)}(k) = \sum_{j\in\mathcal{N}(i)} \sum_{m=1}^{N} \pi_{j,t}^{(m)}(k-1) \log \psi_{ij}(s_{i,t}^{(n)}, s_{j,t}^{(m)}).$$

(b) Re-weight the particles by:

$$\pi_{i,t}^{(n)}(k) = e^{m_{i,t}^{(n)}(k)} \times w_{i,t}^{(n)}.$$

(c) Normalize to obtain

$$Q_{i,t}^{k}(\mathbf{x}_{i,t}) \sim \{s_{i,t}^{(n)}, \pi_{i,t}^{(n)}(k)\}$$

Figure 3.4. The sequential Monte Carlo implementation for variational inference of the Markov network.

Most recently, Sudderth *et al* [121], Isard [61] and Sigal *et al* [118] have developed algorithms for the interactions among multiple particle sets. These algorithms are based on belief propagation, while the above sequential Monte Carlo algorithm is based on probabilistic variational analysis. Although belief propagation and mean field iteration

share the same paradigm of message passing, the difference between them are the contents of the "messages" and the theoretical analysis for the case of loopy graphs. Theoretically, our sequential Monte Carlo implementation for mean field inference is a very good choice for tracking multiple targets as described in the previous sections, which is also supported by the extensive and very promising experiment results as will be reported in next section.

3.6.1. Mixture Density of Importance Sampling

When facing the problem of tracking variable number of targets, many existing multiple target tracking approaches assume fixed backgrounds [56, 55, 63, 123, 160], since the pixel level likelihood facilitates efficient detection of the appearing and disappearing of the targets. We do not limit our approach to this assumption. Therefore, to make possible the capturing of the new targets in dynamic video scenes, under our autonomous tracker network formulation, each autonomous tracker is equipped with a rough local range detector that only searches its nearby regions. When a new target enters the video scene, it may not be immediately sensed and tracked by any of the trackers due to their limited monitoring areas. But their collaboration will gradually distribute them to cover the entire image region such that the new targets can be eventually detected and tracked. In general, the process of pickup is quick in several frames, depending on the size of the tracker network. This is also validated in our experiments. Although this may induce detection lag, it saves computation significantly. An extreme case is to set the detection range of each tracker to be the entire image to obtain instant detection, but incurring demanding computation. Thus, in practice, we need to balance between the detection lag and the computational cost. When the target detectors are available for each autonomous

(1) Structure Determination of Markov Network:
At time t, determine the Markov network structure according to the relative positions and performance indicators of the trackers from time $t-1$.

(2) Importance Sampling:
For tracker $i, i \in M$, generate new samples $\{s_{i,t}^{(n)}\}_{n=1}^{N}$ from importance function $I_{i,t}(\mathbf{x}_{i,t})$.

(3) Importance Re-weighting:
For each $s_{i,t}^{(n)}$, set its re-weight
$\widetilde{\omega}_{i,t}^{(n)} = [\sum_{m=1}^{N} \pi_{i,t-1}^{(m)} p(s_{i,t}^{(n)} | s_{i,t-1}^{(m)})] / I_{i,t}(s_{i,t}^{(n)})$

(4) Observation Likelihood Calculation:
For each $s_{i,t}^{(n)}$, perform likelihood calculation
$w_{i,t}^{(n)} = p(z_{i,t} | s_{i,t}^{(n)})$

(5) Iteration: Initially set $k = 0$ and $k = k + 1$;
(a) calculate the "message" from neighbors:

$$m_{i,t}^{(n)}(k) = \sum_{j \in \mathcal{N}(i)} \sum_{m=1}^{N} \pi_{j,t}^{(m)}(k-1) \log \psi_{ij}(s_{i,t}^{(n)}, s_{j,t}^{(m)}).$$

(b) Re-weight the particles by:

$$\pi_{i,t}^{(n)}(k) = e^{m_{i,t}^{(n)}(k)} \cdot w_{i,t}^{(n)} \cdot \widetilde{\omega}_{i,t}^{(n)}.$$

(c) normalize to obtain

$$Q_{i,t}^{k}(\mathbf{x}_{i,t}) \sim \{s_{i,t}^{(n)}, \pi_{i,t}^{(n)}(k)\}$$

Figure 3.5. Thė sequential Monte Carlo variational inference of the autonomous tracker Markov network for tracking variable number of targets

trackers, a more effective way to generate the new sample set $\{s_{i,t}^{(n)}, \pi_{i,t}^{(n)}\}_{n=1}^{N}$ of tracker i for the current time instant t in algorithm 3.4 is through the collections of target detections reported from each local region target detectors. The target detections then function as the bottom-up data driven process to design informative importance functions to guide the particle sampling [95, 111].

At time t, tracker i detects $\mathcal{C}$ potential targets within its monitoring area, and each of these detections is depicted by the detected location and scale $\{\mathbf{O}_{c,t}, c \in \mathcal{C}\}$. We construct

the following mixture density as an effective importance function:

$$\begin{aligned} I_{i,t}(\mathbf{x}_{i,t}) &= \alpha[\textstyle\sum_{c=1}^{\mathcal{C}} \omega_{c,t} N(\mathbf{x}_{i,t}|\mathbf{O}_{c,t}, \textstyle\sum_{c,t})] \\ &\quad +(1-\alpha)[p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})] \end{aligned} \tag{3.12}$$

where N is a Gaussian density with mean vector $\mathbf{O}_{c,t}$ and diagonal covariance matrix $\sum_{c,t}$, the Gaussian mixture weights $\omega_{c,t}$ are empirically determined based on the detection's location relative to the current position of the tracker. The parameter α balances between target detection and tracker's dynamics. When the tracker is under good conditions, α should be small and the sensing region for tracker's target detection will also be reduced such that the dynamics model plays the dominant role. On the other hand, if the tracker is experiencing a tracking failure or unable to detect anything, α will become large and the detection region will also expand to facilitate the search of the lost target or any potential new targets.

Therefore, under the scenarios of autonomous tracker network inference for tracking variable number of targets, the sequential Monte Carlo implementation of the proposed collaborative tracker can be summarized as in Figure 3.5.

3.7. Experiments

In this section, we report our extensive experiments of the proposed decentralized Markov network formulation to track fixed and variable number of targets.

3.7.1. Tracking Fixed Number of Targets

Extensive and comparative experiments on both synthetic and real data to track fixed number of "identical" targets are reported in this section. In all these experiments, the individual tracker is a 2D appearance-based region tracker, in which the target state $\mathbf{x}_i$ is modelled by 2D affine parameters, the dynamic model $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ is a 2nd order dynamic model, the likelihood function $p(\mathbf{z}_i|\mathbf{x}_i)$ is calculated by matching a PCA-based appearance model which is trained in advance, and 200 particles are used to represent the posterior of each target state. In all these experiments, our collaborative tracker runs comfortably at 15-20 fps on a PIV 2GHz PC.

3.7.1.1. Proof-of-concept. To clearly demonstrate the basic idea and the correctness of our approach, we firstly test our algorithm by a synthetic video sequence, in which five identical and moving tennis are casted into a real dynamic scene. This synthetic testing case prevents the subtraction of the background to obtain easy detection of the targets. Each tennis presents an independent const velocity motion and is bounced by the image borders. This sequence challenges many existing method due to the frequent presence of occlusions.

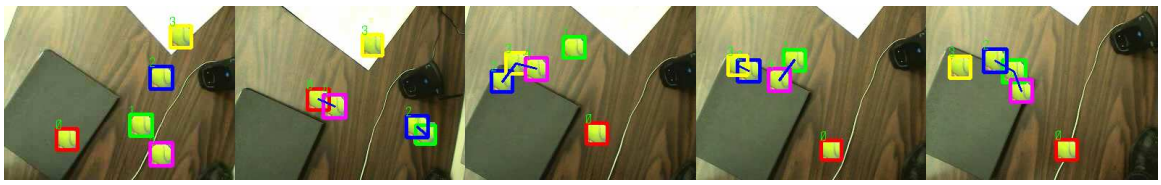


Figure 3.6. MFMC tracker: 5 tennis in a synthetic video. The blue links among the targets illustrate the structure of the *ad hoc* Markov network.

Equipped with the competition mechanism, our collaborative tracker performs excellently. Sample frames from the results are shown in Figure 3.6.We use different colored

rectangles to display the estimated target positions. An index is also attached to each rectangle to identify these tennis uniquely. The blue lines in Figure 3.6 that link the different targets are the visual illustration of the structure of the *ad hoc* Markov network. Therefore, by observing the changing structure of the network over frames, we can clearly learn that which tennis are subject to the collaborative inference and which are simply being tracked by an individual tracker.

Although our collaborative tracker does not deliberately address the identity switching problem, we find in our experiments that our approach seems to have such a capability to nicely handle this problem when combined with motion coherence and dynamic predictions. This can be easily validated by the subjective evaluation on the tracking sequence.

We also compare our results with those obtained by the multiple independent trackers, as shown in Figure 3.7. The number of particles for each target in the M.i.T. algorithm is the same as in our algorithm. However, M.i.T. can not produce satisfactory results, where the coalescence problems always happen during the tracking.

Figure 3.7. M.i.T. tracker: 5 tennis in a synthetic video.

3.7.1.2. Lab Environments. The second and third test sequences are taken in the laboratory environment. In the second sequence, a person is moving a tennis to cross behind a row of other 4 tennis that act as several identical camouflages. Obviously, the occluding tennis increase the burden of correct tracking of the occluded tennis. As expected, our collaborative framework can still effectively handle the difficulty and lead

to a very robust tracking to the occluded target, even successfully keeping the identity of those five tennis. Sample frames of the results are shown in Figure 3.8.

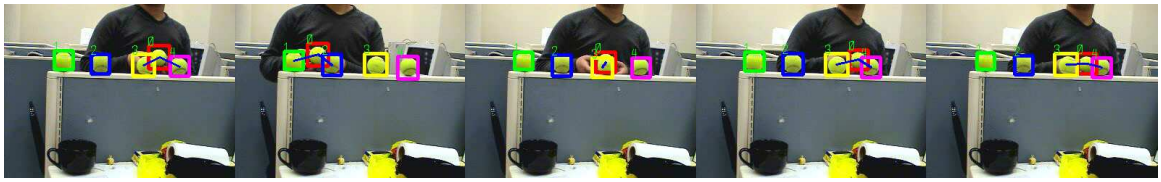


Figure 3.8. MFMC tracker: a tennis moving behind a row of 4 tennis. The blue links among the targets illustrate the structure of the *ad hoc* Markov network.

The third sequence contains 2 moving tennis and 3 still tennis, where different configuration of the structure of ad hoc Markov network is intentionally exploited by changing the positions of the two movable tennis. Once again, our collaborative framework successfully keeps tracking those five tennis. Sample frames are shown in Figure 3.9.

Figure 3.9. MFMC tracker: 2 tennis moving around 3 static tennis. The blue links among the targets illustrate the structure of the *ad hoc* Markov network.

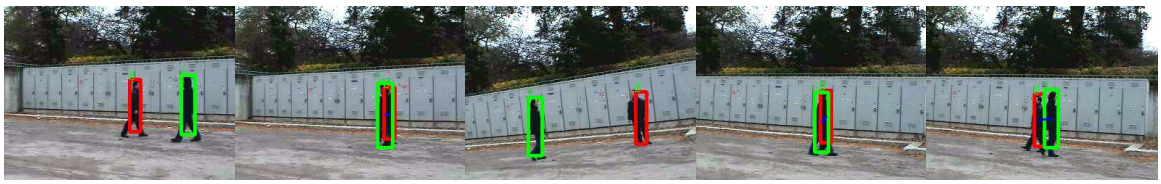


Figure 3.10. MFMC tracker: two people walking. The blue links among the targets illustrate the structure of the *ad hoc* Markov network.

3.7.1.3. Real Scenarios. Both our collaborative tracking framework and M.i.T. trackers have been tested on real scenarios. In the first scenario, two persons are walking

around in the scene and occlusion continuously happens between these two persons. It is easy for our collaborative tracker to obtain very robust results, as shown in Figure 3.10, while M.i.T. can not work well as shown in Figure 3.11.

Figure 3.11. M.i.T. tracker: two people walking.

Finally, three more real sequences that contain many challenging occlusion cases are tested. As expected, our new method provides robust and stable results, as can be seen in Figure 3.12, Figure 3.13, Figure 3.14.

Figure 3.12. MFMC tracker: three women soccer players drilling. The blue links among the targets illustrate the structure of the *ad hoc* Markov network.

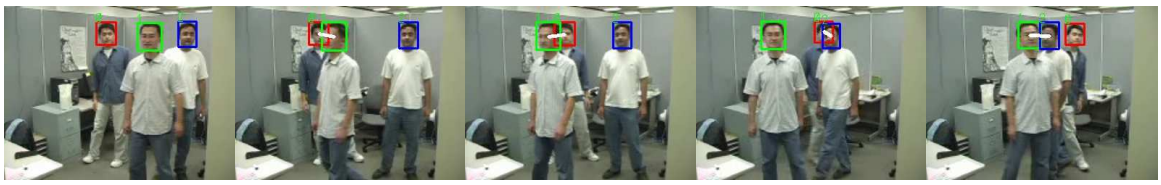


Figure 3.13. MFMC tracker: three faces tracking. The white links among the targets illustrate the structure of the *ad hoc* Markov network.

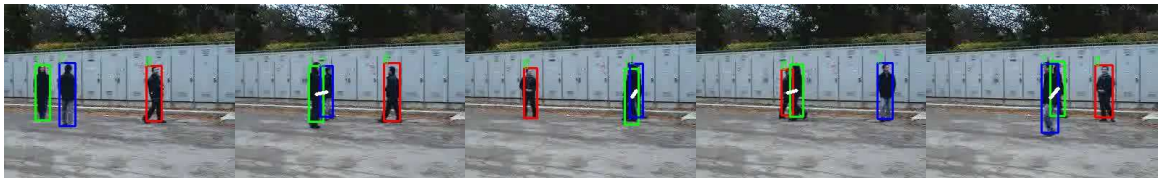


Figure 3.14. MFMC tracker: three human walking around. The white links among the targets illustrate the structure of the *ad hoc* Markov network.

3.7.2. Tracking Variable Number of Targets

To demonstrate the capacity of autonomous tracker network formulation for tracking variable number of targets, the proposed framework discussed in 3.4 is implemented to perform experiments on tracking sports players in real-life video sequences of soccer and hockey games. In both these experiments, a set of 16 trackers is casted to cover the changing background scenes. Each tracker is equipped with an object detector, which is trained using AdaBoost, to help sense the potential appearing sports players within its local range. The training data of the detector for each testing sequence is collected by manually labelling the sports players regions from randomly selected 50 frames of that sequence.

The individual tracker is a rectangle region tracker, where the target state $\mathbf{x}_i$ is modelled by 2D similarity transformation parameters, i.e. translation and scale. The dynamics model $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ is either a constant acceleration model or a uniformly random walk model depending on the performance indicator $\mathbf{r}_{i,t-1}$, as described in section 3.4.1. The likelihood function $p(\mathbf{z}_i|\mathbf{x}_i)$ is a color-histogram based observation model built in HSV color space, which is known insensitive to illumination changes. The histogram model of the target is also trained using the same data set as of training AdaBoost detector. 100 particles are used to represent the posterior of each tracker, which leads to only 1600

particles in total, linear with the number of trackers, to monitor the appearing and disappearing of sports players in the highly dynamic scenes. Under this parameter setting, our collaborative trackers runs around 15 fps on a $P4$ 2GHz PC (Note that our decentralized scheme is theoretically parallel at the algorithm-level, therefore with code optimization, much faster performance can be easily achieved).

We firstly test the proposed approach on tracking multiple soccer players in a video sequence of soccer match, in which the appearing and disappearing of players often happen along the sequence. The maximum number of players simultaneously presenting in the scene is 8. Note that in this sequence there are two team players, one is wearing white sports clothes, while the other is wearing red. Therefore, our 16 trackers are equally divided into two sets, which share the same trained object detector but have different image likelihood functions, each specifically trained for one team.

The gradually detecting of the present soccer players in the viewing scene is shown in Figure 3.15 that are corresponding to the 2, 18, 29 frames of the sequence respectively. The casted autonomous trackers, which are inactive initially, start to roam around the scene with random walk to sense their potential targets, while at the same time forcing their neighboring inactive ones to search around other unchecked areas by communicating with them through variational message, as shown in Figure 3.5. In general it will lead to a roughly uniform coverage over the whole image area as can be seen in Figure 3.15. The red thick rectangles in the Figure illustrate the active trackers which have successfully locked on targets, while the thin blue ones mean they are inactive and still roaming around to search for any potentially new targets. Labels are also displayed to help identify each tracker uniquely.

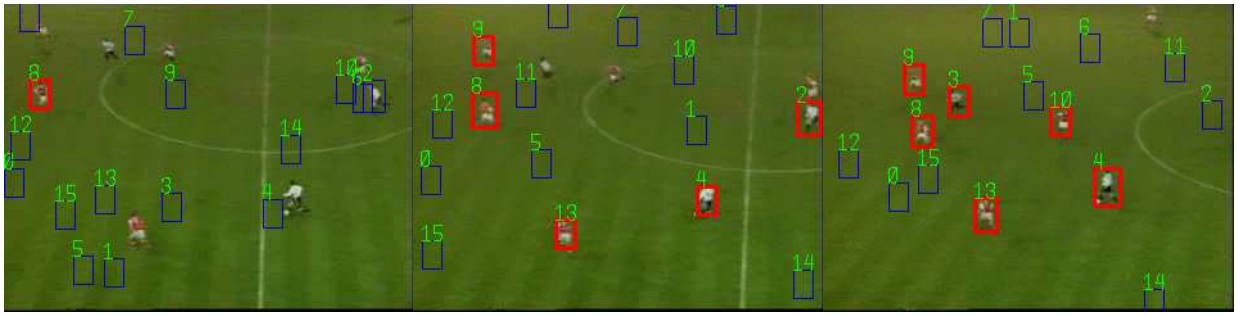


Figure 3.15. Soccer player detections using 16 autonomous trackers with local range AdaBoost detector, frame numbers are 2, 18, 29 respectively. The red thick rectangles illustrate active trackers, while the thin blue means inactive ones. See text for details.

.

Some selected tracking results of the proposed approach are demonstrated in Figure 3.16 (Inactive trackers are not displayed for better illustration). The present soccer players are successfully tracked with uniquely assigned identifiers even under the severe interactions and occlusions. The involved collaborations among targets are depicted by the blue links representing the edges in the underlying Markov network structure (Please note that the network structure changes with time as shown in the Figure). With the help of collaborations among targets, the coalescence problem is successfully handled along the whole sequence.

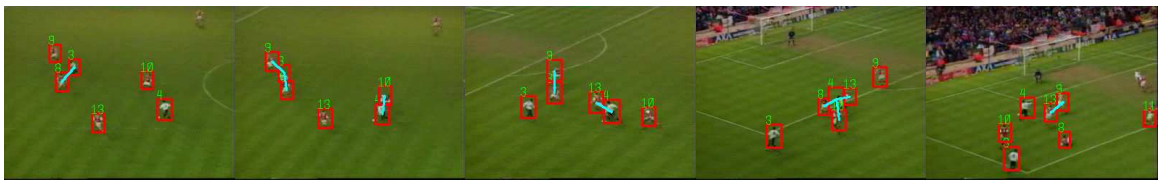


Figure 3.16. Tracking soccer players using the proposed approach, frame number 59, 75, 127, 186, 233. The blue links among the targets illustrate the structure of the Markov network. Please see the attached video for details.

In comparison, 16 multiple independent trackers M.i.T. are also tested to perform detecting and tracking with the same sequence. Every setting is the same as above except the missing of collaborative message passing. The comparison results using the proposed approach and M.i.T. are demonstrated in Figure 3.17, where the left column in the Figure is the original source frames, the middle column corresponds to our proposed approach, and the right one is the results from M.i.T.. For clear illustrations, only the tracking results from the pink areas of the original frames are shown in the middle and right columns. The frame numbers are 141 (top), 215 (bottom) respectively. In the following, for clarification purpose, all the identifiers we described are corresponding to the players in the proposed approach, i.e. the identifiers in the middle column, since there identities are maintained correctly for each player. In frame 141, the M.i.T. loses tracking the red team player 9 due to his previous crossing with the player 8. Actually, this interaction between these two players can be clearly seen in the frame 127 of Figure 3.16. In frame 215, the coalescence problem becomes more severe in the M.i.T. case, where the player 9, although previously lost in frame 141, and then sensed and tracked by other nearby inactive trackers, also "hijack" the tracker of the player 8, which leads to the lose track of player 8. In M.i.T., although the set of trackers equipped with the object detector may successfully cover all appearing targets within the dynamic scene, the inevitable coalescence problem there will result in targets identity switching, then dramatically hurt the tracking performance.

Secondly, a video sequence captured from a hockey game is tested, in which many hockey players appear and disappear in the field and present severe interactions. The tracking results of the sequence are originally reported in [95], which therefore provides

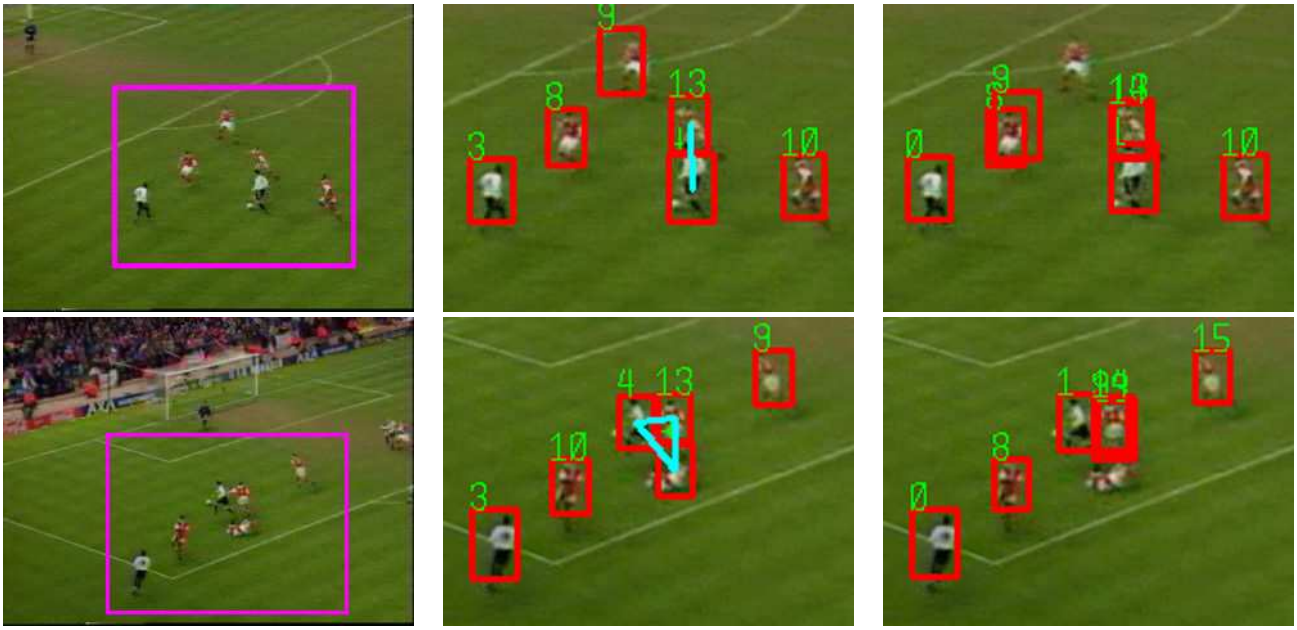


Figure 3.17. The comparison results of tracking soccer players using the proposed approach (middle column) and M.i.T. (right column). Left column is the corresponding source frames, where the pink areas are the actual showing regions in the middle and on the right for better illustration. Frame numbers are 141 (top), 215 (bottom). The blue links among the targets in the middle column illustrate the structure of the Markov network. See text for details.

a direct comparison of our algorithms with theirs. By explicitly introducing the collaborative mechanisms among the spatial adjacent trackers, our proposed approach robustly follows most of the hockey players and handles the coalescence problem satisfactorily, as can be seen in Figure 3.18, while in [95], a simple K-means clustering is proposed to maintain multiple modalities of the underlying particle filtering, therefore may easily result in the identity confusions of hockey players before and after clustering. Please note that the few hockey players are not successfully sensed in the sequence, which are mainly due to the imperfect object detector since it is only trained based the labelled data from 50 frames of the sequence.

Figure 3.18. Tracking hockey players with the proposed approach, frame number 31, 39, 63, 64, 115. The blue links among the targets illustrate the structure of the Markov network. Please see the attached video for details. The authors acknowledge Mr. Kenji Okuma for providing the test data on the website.

3.8. Discussions

Tracking multiple targets is a challenging problem, especially when similar or identical targets move close or occlude with each other. In this case, coalescence that means the tracker associates more than one trajectories to some targets while loses track for others can not be solved by simply instantiating multiple independent trackers. In this chapter, we present a novel decentralized approach with linear complexity to the coalescence problem. The basic idea is the collaborative inference mechanism, in which the estimate of an individual tracker is not only determined by its own observation and dynamics, but also through the interaction and collaboration with the estimates of its adjacent trackers, which leads to a competition mechanism that enables different trackers to compete for the common image observations. The theoretical foundation of the new approach is based on Markov networks, in which the links of the network introduce the competition for image resources among trackers. Variational analysis of this Markov network reveals a mean field approximation to the posterior density of each tracker. Therefore a sequential Monte Carlo algorithm is designed to efficiently implement this mean field approximation inference by simulating the competition among a set of low dimensional particle filters.

To enable the framework to track variable number of targets, we further propose a modified Markov network based on a set of autonomous while collaborative trackers. The autonomous means each individual tracker is self-aware since it can determine its own status, such as following a target or sensing potential new targets by an entropy-based evaluator. In addition, a trained target detector is incorporated to help sense the potential newly appearing targets in the dynamic scene, therefore background subtraction is not necessary to our method. The use of object detectors within each tracker also supports the construction of an effective importance function, which leads to an more effective variational inference.

Since the proposed approach of collaborative tracking multiple targets is a general framework, it does not make any assumptions about the individual autonomous tracker. Therefore we are expecting to incorporate any promising progresses from the robust single target tracking methods into our formulation. One of the representatives is on-line feature selection in [21], where by exploiting the possible disjoint set of discriminative features for multiple targets when they are spatially far away, then when they are coming close, by constraining the corresponding trackers only employ the discovered disjoint feature set, the switching identity problem may be more reliably solved.

CHAPTER 4

Statistical Field Model for Pedestrian Detection

4.1. Introduction

The research of human detection and tracking has received more and more attentions in recent years, due to the drive from many emerging applications such as perceptual interfaces, ubiquitous computing and smart video surveillance [23, 46, 102, 103]. Different applications are concerned with different image resolutions of the subjects, thus requiring different techniques. For example, in perceptual interfaces, the motions of the human body parts need to be determined for action recognition, thus these applications require fairly high resolution for analyzing the articulated motion of the body parts. In contrast, in many video surveillance applications, since the human typically is associated with small image regions, the human needs to be treated as a nonrigid entity for detection and tracking, while the detailed motion of the body parts is no longer the major focus here. This chapter addresses the detection and tracking problem in the latter context.

Remind that we have shown in previous chapters: one of the conclusions we can make for multiple target tracking is that a successful multiple target tracking algorithm, such as multiple pedestrian tracking for intelligent video surveillance in our consideration, may be largely benefited from the availability of a robust object detection component. The existence of a reliable object detector will be able to help collect the valid target observations from each video frame, which will not only reduce the number of wasted

samples due to inaccurate dynamics model, but also provide some informative clues to guide the trackers searching around some more possible areas where the targets may move. In addition, in order to track variable number of targets under a distributed fashion as shown in the previous chapter, an object detector is also indispensable for triggering the possible tracker initialization for every newly appearing target in the video scene. Therefore, in this chapter, we devote ourselves into the investigations of some robust object detector, which, according to our interest, should be particularly suitable for the detections of pedestrians in surveillance video.

A critical issue in object detection and tracking is to calculate the likelihood $p(\mathbf{Z}|\mathbf{y})$ of the image measurements $\mathbf{Z}$ given the rigid motion parameters $\mathbf{y}$ (such as the location, the orientation and the scale) of the target. If the target presents apparent visual invariants (or features), calculating the image likelihood is straightforward. For example, despite the uncertainty in the visual appearances, frontal faces have a similar image pattern that allows the use of the Harr features for face detection [128]. On the other hand, if apparent invariants are not available, we have to break the image likelihood $p(\mathbf{Z}|\mathbf{y})$ into a set of conditionals $p(\mathbf{Z}|\mathbf{y}, \mathbf{X})$ and integrate them, where $\mathbf{X}$ for example can be the nonrigid motion of the target. If $\mathbf{X}$ is complicated, calculating such an integration can be very difficult, leading to the nontrivial nature of detection and tracking in this scenario. Unfortunately, this is the case when treating the human as a nonrigid entity.

Although the research of object detection has greatly moved forward with the success of face detection, these face detection methods may not apply to human detection. The visual appearances of the human present tremendous variabilities while lacking apparent visual invariants, as the diversified clothing and the body articulation may significantly

change the image of a person. In addition, handling partial occlusion is more concerned in detecting humans, because the human detector in practice should be robust to the missing body parts, but most existing methods are limited to cope with this difficulty. Therefore, it is desirable to investigate new human detection methods that cope with the large uncertainty of the visual appearances and are robust to partial occlusions. This chapter addresses these two difficulties.

Because the human-like shapes are more or less unique in the real world, they may provide a powerful clue for human detection and tracking [30, 47, 126]. The difficulty of analyzing human shapes lies in the fact that the local shape nonrigidity has a large number of degrees of freedom thus having very complicated uncertainties, which makes rule-based methods unsuitable. Thus, learning-based methods are generally adopted for learning the shape distributions from training data. If the uncertainty is simple, parametric methods such as Gaussian models or Gaussian mixture models can do a good job, e.g., in face shape [26]. Unfortunately, because of the local nonrigidity, the uncertainty in human shapes is too complex to be sufficiently modelled by reasonable Gaussian mixture models. On the other hand, non-parametric methods, e.g., by using exemplars [47], can be quite flexible. However, we need a huge set of exemplars to represent the concept of the human-like shapes in order to accommodate the possible variations. As a tradeoff, because the Gibbs distribution is flexible to capture a large variety of densities, it can naturally be employed for this task. This idea has been exploited for modeling the face deformations [83], where the face is represented as a random vector and an inhomogeneous Gibbs distribution can be learned from training data. Although this model is complicated

and needs quite involved training, its excellent performance suggests that the Gibbs distribution be useful for characterizing the large and complex variabilities of the human-like shapes.

Unfortunately, it is difficult for the above learning-based methods to handle partial occlusions, because it is not practical, if not impossible, to have the training data that cover all possible situations of partial occlusions. Actually, this difficulty roots in the fact that these methods represent a pattern as a centralized random feature vector. Thus, the missing elements due to occlusion will greatly change the feature vector, thus affecting the classification dramatically. Different from such vectorized methods, the component-based methods [90, 105] divide the entity into parts and take advantage of the structures or correlations of the parts. They have demonstrated excellent performances on detecting partial occluded targets, suggesting that we need to go beyond the vectorized models.

The contribution of this chapter is a new non-vectorized method based on a two-layer field model for detecting and tracking complex targets such as the human. This new method stands out because of its robustness to partial occlusions, which is difficult for the vectorized methods.

Different from most existing methods, this new approach embeds the complicated nonrigid shape prior into a statistical field and distributes the complex image likelihood to the local sites of the field. This new model has two layers. The hidden layer is a hidden Markov field that captures the shape prior. Every node of this Markov field is associated with an observation node describing the conditional likelihood of image observations of this hidden node, thus constituting the observation layer of this field model. We model the prior of the nonrigid human shapes as a Boltzmann distribution. Although it is a special

Gibbs distribution, our method is different from [83], because we do not characterize the Boltzmann distribution directly in a vector space, but distributing it into the Markov field. This treatment results in quite simple inference and learning algorithms in both theory and practice. Although the structure of this field model is similar to that in [44], the difference is apparent, as our method employs probabilistic variational analysis that leads to rigorous and elegant analytical approximations [66, 70]. Another theoretical benefit is that the image likelihood estimates are lower bounded. This new approach enables effective and efficient detection and tracking algorithms for many nonrigid objects such as pedestrians.

This new approach has a number of advantages over many existing methods. Firstly, since this model employs a field rather than a vector to describe a shape, it can sufficiently capture the local variabilities of the shape by the local network structure, thus enabling accurate modeling of the complex shape prior. Secondly, since the model captures shape variabilities and performs image measurements in a distributed fashion, it is more robust against occlusion than the vector-based global approaches (such as PCA) in which image measurements have to be performed in a centralized fashion, i.e., conditioned on all shape parameters. In addition, having an observation layer leads to more flexibility and robustness for handling cluttered backgrounds. Thirdly, the variational approximation provides a computationally efficient way to compute the likelihood of image observations, to infer the hidden states of the model, and to facilitate fast learning. Last but not least, it integrates the top-down and bottom-up methodologies for tracking nonrigid objects. The top-down approaches involve evaluating a large number of hypotheses, and the bottom-up approaches require large efforts in grouping and detection. Given the huge variabilities in

the nonrigid human shapes, neither approach would be satisfactory, because the number of hypotheses would be tremendous and grouping a nonrigid object is very difficult. The proposed tracking method is able to balance these two methodologies and to combine the advantages of both: the global variabilities are handled in a top-down fashion by particle filtering [12,62], while the local nonrigidity is coped by an bottom-up approach by directly evaluating the likelihood of image measurements.

The chapter is organized as the following. After a brief description of the related work in Section 4.2, we present the two-layer field model in Section 4.3. The probabilistic variational analysis of this model is given in Section 4.4, and the learning algorithm is presented in Section 4.5. Section 4.6 describes our methods for pedestrian detection and tracking, and our extensive results are reported in Section 4.7. The chapter concludes in Section 4.8.

4.2. Related Work

In the past decades, many methods have been proposed for object detection, mainly for the human face and cars. They are based on different classification schemes. Neural network has been employed for detecting faces [110] by classifying the candidate image patches into face or non-face classes. A learned histogram model for wavelet coefficients can be used for face/car detection [112], because the histograms approximate the distributions of object features for discrimination. In addition, combined with the Harr features, the AdaBoost classifier has been very successful for frontal face detection [128], and has been extended for pedestrian detection [129] with the help of motion information. Support vector machines are also widely used in the detection tasks [96, 98].

But most existing approaches seem not to be suitable for the detecting targets with large shape variations, such as the pedestrian. For example, methods based on the raw pixel features, such as [97,110], can not handle large variability of the appearances of the pedestrian. It turns out the shape features of these deformable targets need to be used.

The research of nonrigid shape analysis has a long history, and various approaches have been proposed and investigated. For all these methods, three important common issues should be addressed, i.e., the shape representation $\mathbf{X}$, the shape prior $p(\mathbf{X})$ and the conditional likelihood of image observation $p(\mathbf{Z}|\mathbf{X})$. (Here we drop the rigid motion $\mathbf{y}$ for clarity.)

Different shape representations can be categorized into either parametric or non-parametric models. Examples of parametric representations includes Fourier descriptors, B-splines [12, 71], the deformable template [158], etc, where shape deformation is controlled by the shape parameters and smoothness constraints. A typical non-parametric representation is the point distribution model [26] where a shape is described by an ordered and labelled set of landmark points, and the shape deforms when the points change. Although it provides great flexibility, registration of landmark points is not a trivial task. An even radical approach is to use a 2D mask [47,69,126], where the shape deforms when multiplying by a sparse permutation matrix [69], or selecting different exemplars [47,126]. In all these representations, a deformable shape is mapped to a point in a vector space (i.e., the shape space), although the dimensionality varies for different approaches. These vectorized models are global, since the image observations are conditioned on all the shape parameters. Thus, these global methods are generally not likely to be able to

cope with partial occlusions, unless the training data have incorporated all possible occlusion situations. In this chapter, rather than using a global representation, we propose a field representation, with which the complex variabilities of the nonrigid shape and the occlusion difficulty can be handled easily.

Obviously, in reality, a shape can not be allowed for arbitrary deformation, thus we should characterize the allowable shape space by having a shape deformation prior model $p(\mathbf{X})$. An idea is to reduce the correlations among different shape parameters, and model the variance of deformation by a multivariate Gaussian distribution in a lower-dimensional subspace. This is the spirit of the principal component analysis (PCA), and has been widely adopted for learning deformation priors [12, 26]. Since PCA identifies a linear subspace and catches linear correlations, it is powerful to capture and decorrelate certain global deformations, but insufficient for the local nonrigidity. Thus, it motivated the methods that use mixture distributions [69] or exemplar databases [47, 126]. Although mixture distributions can represent arbitrarily complicated densities in theory, it becomes unrealistic when the number of mixtures increase tremendously. To alleviate this difficulty, an inhomogeneous Gibbs model has been proposed and applied successfully to face deformation [83], although the model needs expensive training. The approach proposed in this chapter also models the shape deformation prior, but instead of modeling the prior in a global fashion, our approach is based on the field representation, and the prior, i.e., a Boltzmann distribution (a specific Gibbs distribution), is distributed into a Markov field, and a variational analysis is employed for analytical results (details in Section 4.3 and 4.4). The new approach stands out from the existing methods by this new representation.

Different approaches have been investigated to *fit* a shape model to image observations. This can be done through minimizing an energy function [71], or based on the Bayesian framework where it is important to characterize the conditional likelihood of image observation $p(\mathbf{Z}|\mathbf{X})$. Analytical forms can be obtained by assuming the independence among a set of discrete points on shape contours [12, 86]. To bypass the independence assumption which may be invalid in reality, the conditional likelihood can be modelled as a metric exponential density obtained from the Chamfer distance based on exemplars [126]. When separating global motion from local nonrigidity, the likelihood conditioned on only global motion can be obtained by the mixture (integral) of all exemplar components in the metric mixture model [126]. The proposed approach in this chapter also provides tractable ways to calculate the likelihood only conditioned on global motion, but the differences from [126] are: (a) in our model $p(\mathbf{Z}|\mathbf{X})$ factorizes by independent components, and (b) $p(\mathbf{Z})$ is an integral over almost infinite number of $\mathbf{X}$ instead of a finite set of exemplars, and our method obtains a lower bound of $p(\mathbf{Z})$.

There have been many excellent works on nonrigid shape matching [7, 19, 27, 106, 115]. These methods are more concerned with the matching of extracted shapes for shape registration, where the nonrigid motion needs to be explicitly estimated, while this chapter is more concerned with integrating out the variabilities of the nonrigid motion. In addition, since this chapter is based on field model, it is also related to Markov random fields (MRF) that have been widely used for image restoration [49], texture analysis [163], surface reconstruction [48], etc. This chapter extends MRF to a two-layer model that consists of a random field and an image observation layer. It is more like the Markov network [44, 130]. The detailed differences will be presented in later sections. More

importantly, this chapter deals with the nonrigid target detection and tracking problem that has not been addressed by the above methods.

4.3. The Field Representation

Global methods such as PCA are suitable for capturing the global deformation with a set of uncorrelated deformation basis. But they tend to ignore the detailed local variations induced by the nonrigidity and they are generally vulnerable to partial occlusion. Therefore, it is desirable to have a model that can handle the large number of degrees of freedom of the local nonrigidity and is robust to occlusion. In this chapter, we propose a two-layer field model as the representation. This is not a vectorized and centralized model but a field and distributed model, as shown in Figure 4.1.

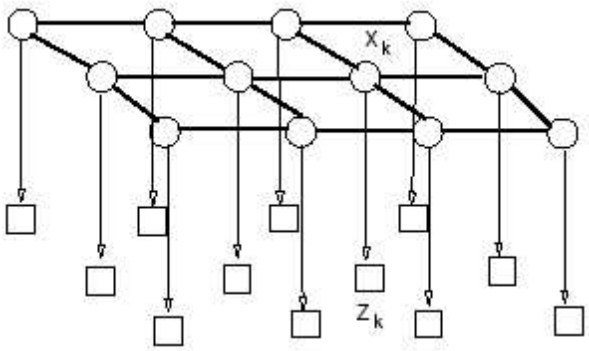


Figure 4.1. A two-layer field representation for nonrigid objects.

This field model consists of two layers. The hidden layer is a hidden random field that represents the shape, modelled as an undirected graph $G_x = \{V, E\}$, where each vertex (or node, or site) represents the hidden shape scene x_k to be inferred. In this model, x_k takes binary values, i.e., $x_k \in \{0, 1\}$, where $x_k = 1$ means that node k is on the object's

contour. Each hidden node is connected to its neighborhood nodes $\mathcal{N}(k)$, thus forming a field.

This hidden random field captures the prior of the shape and needs to be inferred from image observations. The feasible shape changes is described by the joint probability of all hidden nodes, i.e., $\mathbf{X} = \{x_1, \ldots, x_n\}$. We assume $p(\mathbf{X})$ to be a Gibbs distribution. Because we embed this Gibbs distribution in the random field, it can be equivalently factorized as:

$$p(\mathbf{X}) = \frac{1}{Z_c} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j) \prod_{i \in V} \psi_i(x_i) \tag{4.1}$$

where ψ_i and ψ_{ij} are the potential functions associated with site $i \in V$ and the link $(i, j) \in E$, and Z_c is a normalization term or the partition function. Specifically, because x_i is binary in our setting for modeling the shape, $p(\mathbf{X})$ becomes a Boltzmann distribution, i.e.,

$$p(\mathbf{X}) = \frac{1}{Z_c} \prod_{(i,j)\in E} e^{\alpha_{ij} x_i x_j} \prod_{i \in V} e^{\beta_i x_i} \tag{4.2}$$

where $\{\alpha_{ij}, \beta_i : \forall (i, j) \in E, i \in V\}$, are parameters which can be learnt from training data (see Section 4.5).

The other layer is the observation layer, through which the shape is associated with its image measurements. As shown in Figure 4.1, each hidden node x_k is associated with an observation node z_k representing the image observation produced by x_k, which is characterized by the conditional probability $p(z_k|x_k)$. The observation of the shape is the collection of the image observations for all sites, i.e., $\mathbf{Z}(\mathbf{y}) = \{z_1, \ldots, z_n\}$, where $\mathbf{y}$ is the global motion. This is a distributed likelihood model. Without causing confusion, we

denote $\mathbf{Z}(\mathbf{y})$ by $\mathbf{Z}$ for short in later sections. We have:

$$p(\mathbf{Z}|\mathbf{X}) = \prod_{k=1}^{n} p_k(z_k|x_k). \tag{4.3}$$

Thus, the model in Figure 4.1 is fully characterized by $\{\alpha_{ij}, \beta_i, p_i\}$, where $p_i = p_i(z_i|x_i)$, and we denote the model by $\lambda = \{\alpha_{ij}, \beta_i, p_i\}$.

This two-layer field model is suitable for describing local nonrigidity and is robust to occlusion, because of the following reasons. (a) Since the neighborhood sites of the shape are generally correlated, this model captures the correlations and constraints among neighbor sites rather than simply treating them independently, thus resulting in more accurate modeling. (b) The Boltzmann distribution can capture complex distributions which can not be represented by Gaussian or mixture of Gaussian, thus providing more powerful priors. (c) Because the observation model $p(\mathbf{Z}|\mathbf{X})$ is distributed over the field (i.e., the shape), wrong estimates on some part of the field may not ruin the other parts of the shape, thus leading to the robustness to partial occlusion.

Within this model, we need to solve two key problems:

(1) *Calculating the likelihood* $p(\mathbf{Z}|\lambda)$. However, this is not a trivial problem, since it involves the integral of all possible configurations of $\mathbf{X}$, i.e.,

$$p(\mathbf{Z}|\lambda) = \int_{\mathbf{X}\in\mathcal{X}} \prod_{i=1}^{n} p_i(z_i|x_i) p(\mathbf{X}) d\mathbf{X}. \tag{4.4}$$

The key to solve this problem is to design an effective inference algorithm that estimates the posterior $p(\mathbf{X}|\mathbf{Z}, \lambda)$ and its marginals $p(x_i|\mathbf{Z}, \lambda)$.

(2) *Learning model parameters* λ. These parameters need to be estimated from training data. Without causing any confusion, we usually denote $p(\cdot|\lambda)$ by $p(\cdot)$ for short.

The learning problem is closely related to the likelihood problem, because the solution to the learning problem relies on the inference of the model (i.e., the estimation of $p(\mathbf{X}|\mathbf{Z},\lambda)$). Therefore, we present an analytical approximation to the likelihood in Section 4.4, and the solution to the learning problem in Section 4.5.

4.4. Variational Inference

The field model introduced in Section 4.3 is a high dimensional stochastic system, because it consists of a large number of random variables (or nodes). Thus, solving the observation likelihood $p(\mathbf{Z}|\lambda)$ and the posterior $p(\mathbf{X}|\mathbf{Z},\lambda)$ involves computationally intensive multi-dimensional integral over $p(\mathbf{X},\mathbf{Z}|\lambda)$. Although the Markovian property of the structure of $p(\mathbf{X}|\lambda)$ simplifies the problem, the exact analysis for such a model is still prohibitive due to the loopy structures of this field model.

Thus, approximated but computationally efficient analysis methods are of special interests. *Probabilistic variational approximation* is one of these methods. Here, the general approach of the variational analysis for the field model is given in Section 4.4.1, and the deduced Boltzmann field for nonrigid shapes is presented in Section 4.4.2.

4.4.1. Probabilistic Variational Analysis

The core idea of probabilistic variational analysis is to find an analytical and simple variational distribution $Q(\mathbf{X})$ from a variational family to approximate the complicated

posterior probability $p(\mathbf{X}|\mathbf{Z})$, such that the Kullback-Leibler (KL) divergence of these two distributions is minimized.

To see this clearly, we follow Jaakkola & Jordan [66] and formulate an optimization problem to solve $p(\mathbf{Z})$ and $p(\mathbf{X}|\mathbf{Z})$ simultaneously. We can write an objective function as:

$$\begin{aligned} J(Q) &= \log p(\mathbf{Z}) - KL(Q(\mathbf{X})||p(\mathbf{X}|\mathbf{Z})) \\ &= \log p(\mathbf{Z}) - \int_x Q(\mathbf{X}) \log \frac{Q(\mathbf{X})}{p(\mathbf{X}|\mathbf{Z})} \\ &= -\int_x Q(\mathbf{X}) \log Q(\mathbf{X}) + \int_x Q(\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}) \\ &= H(Q) + E_Q[\log p(\mathbf{X}, \mathbf{Z})] \end{aligned} \tag{4.5}$$

where $H(Q)$ is the entropy of $Q(\mathbf{X})$ and $E_Q[\cdot]$ denotes the expectation w.r.t. $Q(\mathbf{X})$. It is easy to see that $\log p(\mathbf{Z})$ is lower bounded by $J(Q)$, since the KL divergence is non-negative. By maximizing the lower bound $J(Q)$ w.r.t. Q, we can obtain an optimal approximation of $p(\mathbf{X}|\mathbf{Z})$ by Q^*, and a closest value of $\log p(\mathbf{Z})$ by $J(Q^*)$.

The spirit of this variational approach is to find the best approximation of $p(\mathbf{X}|\mathbf{Z})$ within a given variational family $Q(\mathbf{X})$. When such a variational family has good analytical properties, such as having independent components, or sparse correlations, or factorized forms, analytical approximation can generally be expected. Although the selection of the variational family $Q(\mathbf{X})$ can be arbitrary, an appropriate $Q(\mathbf{X})$ will make big difference on analyzing. Here, we adopt a fully factorized form:

$$Q(\mathbf{X}) = \prod_i^n Q_i(x_i) \tag{4.6}$$

where $Q_i(x_i)$ is an independent distribution of the hidden node x_i. Then, we can write the entropy of the variational distribution as:

$$H(Q) = \sum_i H(Q_i).$$

Such a fully factorized variation leads to the mean field approximation. To see this clearly, we minimize the KL divergence with respect to $Q(\mathbf{X})$. It can be easily shown (in the Appendix 8) that the optimal approximation is made of a set of interrelated Gibbs distributions:

$$Q_i(x_i) = \frac{1}{Z_i} e^{E_Q[\log p(\mathbf{X},\mathbf{Z})|x_i]}, \quad i = \{1, \ldots, M\} \tag{4.7}$$

where Z_i is a normalization constant, and $E_Q[\log p(\mathbf{X}, \mathbf{Z})|x_i]$ is the conditional expectation given x_i. The set of equations in Eq. 4.7 are fixed point equations. The iterative updating of $Q_i(x_i)$ will monotonically increase $J(Q)$ and eventually reach an equilibrium. These equations can be called as *mean field equations*.

Eq. 4.7 gives a general solution with a very general form of $Q(\mathbf{X})$. Furthermore, when taking advantage of the special factorization property of $p(\mathbf{X})$ in Eq. 4.2 and $p(\mathbf{Z}|\mathbf{X})$ in Eq. 4.3, we can easily obtain a further simplification. Given the structure of this field model, it is easy to shown that:

$$Q_i(x_i) \longleftarrow \frac{1}{Z_i'} p_i(z_i|x_i)\psi_i(x_i)M_i(x_i), \quad \text{where}$$

$$M_i(x_i) = \exp\{\sum_{k \in \mathcal{N}(i)} \int_{x_k} Q_k(x_k) \log \psi_{ik}(x_i, x_k)\}, \tag{4.8}$$

where Z_i' is a normalization constant, and $\mathcal{N}(i)$ is the neighborhood of the site i. The iterative updating of $Q_i(x_i)$ based on these mean field equations will monotonically increase $J(Q)$ as well and eventually reach an equilibrium. From Eq. 4.8, it is interesting to notice that the variational belief of a hidden node x_i is determined by three factors: the local conditional likelihood $p_i(z_i|x_i)$, the local prior $\psi_i(x_i)$, and the neighborhood prior $M_i(x_i)$ from the constraints of the neighborhood nodes $x_{\mathcal{N}(i)}$. This can be treated as a generalized Bayesian rule for the field model.

Thus, we can treat the term $p_i(z_i|x_i)\psi_i(x_i)$ as the local belief of x_i, and treat the term $M_i(x_i)$ as the "message" propagated through the nearby nodes of x_i. This method is actually different from the belief propagation algorithm [44], due to its use of variational analysis and to the different contents in the "messages". In our method, the computation of $M_i(x_i)$ is easier than belief propagation, because of the factorization in the variational distribution. In addition, we can clearly see from this equation that the computation is significantly reduced by avoiding multi-dimensional integral, noticing Eq. 4.8 involves only one dimensional integral.

4.4.2. Boltzmann Field

The derivations described above was only based on the factorization properties of $Q(\mathbf{X})$, $p(\mathbf{X})$ and $p(\mathbf{Z}|\mathbf{X})$. Thus the result is quite general. When using the field model for nonrigid shapes, since x_i are binary random variables (i.e., $x_i \in \{0, 1\}$), we can employ a Boltzmann distribution for $p(\mathbf{X})$, as introduced in Section 4.3 and Eq. 4.2. Since x_i is

binary, we can choose a specific variational distribution here:

$$Q(\mathbf{X}) = \prod_{i}^{n} \mu_i^{x_i}(1-\mu_i)^{(1-x_i)}, \tag{4.9}$$

where $\{\mu_i\}$ are variational parameters to be estimated. Under this variational distribution, the mean field equations Eq. 4.8 can be further simplified as:

$$\mu_i = \frac{p_i(z_i|x_i=1)m_i}{p_i(z_i|x_i=0) + p_i(z_i|x_i=1)m_i}, \tag{4.10}$$

where,

$$m_i = \exp\{\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mu_j + \beta_i\}.$$

We can call it a *Boltzmann field*. This set of mean field equations in Eq. 4.10 are much simpler than Eq. 4.8, since they only involve a finite set of variational parameters, rather than a set of Gibbs distributions. As a result, the computation is quite straightforward. Similar results have also been obtained by Jordan et al. [70], Peterson and Anderson [104].

Then based on this particular variational setting and the result above, Eq. 4.5 becomes:

$$\begin{aligned} J(Q) &= \sum_i H(Q_i) + \sum_{(i,j)\in E} \alpha_{ij}\mu_i\mu_j + \sum_{k\in V} \mu_k\beta_k \\ &+ \sum_{k\in V}(1-\mu_k)\log p_k(z_k|x_k=0) \\ &+ \sum_{k\in V}\mu_k \log p_k(z_k|x_k=1) - \log Z_c \end{aligned} \tag{4.11}$$

We admit that $J(Q)$ can not be fully computed, because of the complexity of calculating $\log Z_c$. Instead, it is simple to compute $\widetilde{J}(Q) = J(Q) + \log Z_c$ in practice. Fortunately,

it is not necessary to calculate $\log Z_c$, because once we find an optimal mean field distribution of Q^*, we readily have:

$$p(\mathbf{Z}) \propto e^{\widetilde{J}(Q^*)},$$

which is enough for our application of detection and tracking in Section 4.6.

4.5. Learning

This section discusses the problem of learning model parameters $\lambda = \{\alpha_{ij}, \beta_i, p_i\}$ from training data. The solution of this model learning problem is in the expectation-maximization (EM) framework, where the core of the expectation step is the inference of the hidden field described in Sec. 4.4. In our method, the training of $\{\alpha_{ij}, \beta_i\}$ and $\{p_i\}$ can be separated. Considering the difficulty of collecting the training data with the known hidden variables (i.e., the annotated training data), we propose a method of using both annotated and un-annotated training data in a semi-supervised fashion. The proposed learning method is based on the Gibbs sampling technique and the Expectation-Maximization iterations.

The initial model is constructed by the following way:

(1) Collecting a set of annotated training examples, $\mathcal{L} = \{\mathbf{X}^k, \mathbf{Z}^k\}_{k=1}^{K_1}$, where $\mathbf{X}^k$ and $\mathbf{Z}^k$ denote the k-th annotated training sample. For each sample, the i-th hidden node of the field model takes binary values $x_i \in \{0, 1\}$, and the observation of this hidden node z_i is the average edge direction over a small image patch associated with x_i in our nonrigid shape applications. We quantize z_i, and use a histogram to model its distribution. If the target shape is very small, z_i simply takes binary value to indicate if it is a detected edge point or not.

(2) Learning $p_i(z_i|x_i)$ for each x_i. Due to the factorization of $p(\mathbf{Z}|\mathbf{X})$, i.e., Eq. 4.3, each individual $p_i(z_i|x_i)$ can be learned independently. Each $p_i(z_i|x_i)$ is represented by a histogram in our experiments.

(3) Learning $\{\alpha_{ij}, \beta_i\}$ by the following steps:

3.a calculating sufficient statistics $S_{ij} = E_p[x_i x_j]$ and $S_i = E_p[x_i]$ from the annotated training data $\{\mathbf{X}^k\}_{k=1}^{K_1}$;

3.b initialize a model $\lambda_b^0 = \{\alpha_{ij}^0, \beta_i^0\}$;

3.c producing synthesized samples of $\{\mathbf{X}_g^k\}_{k=1}^N$ by Gibbs sampling of $p(\mathbf{X}|\lambda_b)$;

3.d calculating sufficient statistics $G_{ij} = E_{\lambda_b}[x_i x_j]$ and $G_i = E_{\lambda_b}[x_i]$ from the synthesized data $\{\mathbf{X}_g^k\}_{k=1}^N$;

3.e adjusting the parameters by:

$$\Delta\alpha_{ij} \propto (G_{ij} - S_{ij}) \tag{4.12}$$

$$\Delta\beta_i \propto (G_i - S_i) \tag{4.13}$$

3.f go to step 3.c;

In our experiments, we select:

$$\alpha_{ij}^0 = \log \frac{S_{ij}}{1 - S_{ij}}, \text{ and } \beta_i^0 = \log \frac{S_i}{1 - S_i}$$

as the initialization. In all of our experiments, we observed the convergence in less than 50 iterations.

Once the model is initialized, we finely tune the model by using a large set of unannotated training examples $\mathcal{U} = \{\mathbf{Z}^k\}_{k=1}^{K_2}$ which are cheaply available. The process is an EM iteration:

- **E-step**: $\forall \mathbf{Z}^k \in \mathcal{U}$, infer the posterior $p(x_i^k|\mathbf{Z}^k, \lambda^t)$ based on variational mean field approximation in Eq. 4.8, i.e., we obtain the set of variational parameters $\{\{\mu_i\}^k\}_{k=1}^{K_2}$.
- **M-step**: estimate the model parameters $\lambda^{t+1} = \{\alpha_{ij}^{t+1}, \beta_i^{t+1}, p_i^{t+1}\}$, given a fixed $\{\{\mu_i\}^k\}_{k=1}^{K_2}$ by a stochastic gradient descent:

$$\Delta\alpha_{ij} \propto \frac{\partial J(Q)}{\partial \alpha_{ij}} \approx \mu_i\mu_j - E_Q[x_i x_j] \tag{4.14}$$

$$\Delta\beta_i \propto \frac{\partial J(Q)}{\partial \beta_i} \approx \mu_i - E_Q[x_i] \tag{4.15}$$

  where $E_Q[x_i x_j]$ and $E_Q[x_i]$ are sufficient statistics calculated w.r.t. the variational distributions. And the method of estimating p_i is the same as the step 2 in the above supervised training.

4.6. Pedestrian Detection and Tracking

A suitable representation for the nonrigidity of a pedestrian is critical for detection and tracking. In this section, we approach to these tasks by the proposed field model.

4.6.1. Pedestrian Detection

Pedestrian detection involves two mean field models: λ_0 corresponds to the negative hypothesis H_0, i.e., no pedestrian presence, and λ_1 to the positive hypothesis H_1, i.e.,

pedestrian presence. The detection algorithm scans different extrinsic shape poses, including locations $\mathbf{u}$, orientations θ, and scales s, denoted by $\mathbf{y} = \{\mathbf{u}, \theta, s\}$. For different scales, we keep the dimension and the number of hidden nodes of the field model the same, but use different sizes of image patches for the observation nodes. In our experiment, we scan all image locations and over 5 scales.

For each extrinsic shape pose $\mathbf{y}$, we collect the edge map of the corresponding image patch and treat it as the image observation $\mathbf{Z} = \mathbf{I}(\mathbf{y})$ of the hidden Markov field. We perform likelihood ratio detection on each given $\mathbf{y}$ to determine the pedestrian presence on this particular $\mathbf{y}$:

$$\log p(\mathbf{Z}|\mathbf{y}, \lambda_1) - \log p(\mathbf{Z}|\mathbf{y}, \lambda_0) > \tau_o \geq 0. \tag{4.16}$$

Since it is unrealistic to calculate $p(\mathbf{Z}|\mathbf{y}, \lambda)$ (in Eq. 4.4), the variational analysis in Section 4.4 nicely provides a mean field solution as an approximation, i.e.,

$$\log p(\mathbf{Z}|\mathbf{y}, \lambda) \approx J(Q^*(\mathbf{X}|\mathbf{y}, \lambda)),$$

where $Q^*(\mathbf{X}|\mathbf{y}, \lambda)$ is the optimal mean field approximation of the posterior $p(\mathbf{X}|\mathbf{Z}, \mathbf{y}, \lambda)$. Thus, the detection rule for each given $\mathbf{y}$ becomes:

$$\widetilde{J}(Q^*(\mathbf{X}|\mathbf{y}, \lambda_1)) - \widetilde{J}(Q^*(\mathbf{X}|\mathbf{y}, \lambda_0)) > \tau, \tag{4.17}$$

where $\widetilde{J}(Q^*(\mathbf{X}|\mathbf{y}, \lambda_k)), k = \{0, 1\}$ can be obtained according to Eq. 4.5 once the mean field iteration converges at $Q^*(\mathbf{X}|\mathbf{y}, \lambda_k)$ according to Eq. 4.8.

There are two factors affecting the threshold τ: (a) $J(Q^*|\lambda_k)$ only provides an optimal lower bound of $\log p(\mathbf{Z}|\lambda_k)$, and (b) we generally only calculate $J(Q^*|\lambda_k)$ up to a constant difference, i.e., $\log Z_c^k$ (see Eq. 4.11). Thus, we do not simply set $\tau = 0$, but train this threshold from supervised examples to reduce the rate of false alarm and miss detection.

4.6.2. Pedestrian Tracking

Different from detection, only the pedestrian model λ_1 is involved in tracking, where the task is to estimate the posterior density of $p(\mathbf{y}_t|\underline{\mathbf{I}}_t, \lambda_1)$, where $\mathbf{y_t} = \{\mathbf{u}_t, \theta_t, s_t\}$ is the same as in the detection problem, and $\underline{\mathbf{I}}_t = \{\mathbf{I}_1, \ldots, \mathbf{I}_t\}$. According to Bayesian rule, we have:

$$p(\mathbf{y}_t|\underline{\mathbf{I}}_t, \lambda_1) \propto p(\mathbf{I}_t|\mathbf{y}_t, \lambda_1) \int_{y_{t-1}} p(\mathbf{y}_t|\mathbf{y}_{t-1}) p(\mathbf{y}_{t-1}|\underline{\mathbf{I}}_{t-1}, \lambda_1). \tag{4.18}$$

The dynamic process can be represented as a dynamic Bayesian network in Figure 4.2. Clearly, the hidden factor $\mathbf{X}_t$ of local nonrigidity has been integrated out in the observation

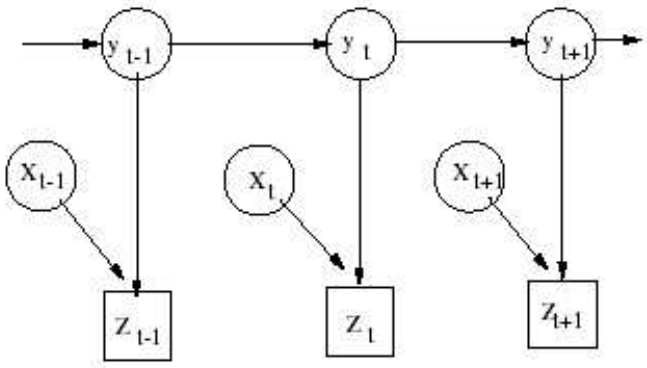


Figure 4.2. The dynamic process for tracking a nonrigid target.

process. This is powerful for tracking since it leaves no extra motion parameters to be estimated.

It is clear that the visual dynamics is governed by the dynamics model $p(\mathbf{y}_t|\mathbf{y}_{t-1})$ and the observation model $p(\mathbf{I}_t|\mathbf{y}_t, \lambda_1)$. Since we have:

$$p(\mathbf{I}_t|\mathbf{y}_t, \lambda_1) = p(\mathbf{Z}(\mathbf{y}_t)|\lambda_1) \propto e^{\widetilde{J}(Q^*(\mathbf{X}_t|\mathbf{y}, \lambda_1))},$$

the local nonrigidity has been absorbed in the calculation of data likelihood which is based on the mean field inference. In our experiments, the dynamics model is characterized as a const acceleration model, and the parameters are learned from an annotated training sequence. Once the MAP solution

$$\mathbf{y}_t^* = \arg\max p(\mathbf{y}_t|\underline{\mathbf{I}}_t, \lambda_1)$$

is obtained, the local nonrigidity can be revealed by

$$p(\mathbf{X}_t|\mathbf{I}_t, \mathbf{y}_t^*, \lambda_1) \approx Q^*(\mathbf{X}_t|\mathbf{y}_t^*, \lambda_1).$$

Because the image likelihood $p(\mathbf{I}_t|\mathbf{y}_t, \lambda_1)$ can be calculated, the tracking algorithm can be easily implemented using particle filtering [12,62], where each particle represents a sample of $\mathbf{y}_t$. Detailed results will be reported in Section 4.7.

4.7. Experiments

In order to validate the proposed approach and demonstrate the applicability of this field model, we performed experiments on pedestrian detection and tracking, and compared the proposed detection method with the AdaBoost detector.

4.7.1. Training and Model Validation

We trained two models, one for the human λ_1 and the other for the background λ_0. In our experiments, the size of the field was set to 12×6, and each of the node covers an image patch, whose size depends on scale. We used 16×16 patches for the finest scale, and coarser scales correspond to smaller patches. We used 5 scales, where the coarsest scale takes 5×5 patches. Please note that neighborhood image patches overlap.

To train λ_1, the training data of various people were collected and their contours were extracted. Then we resized and aligned all the contours by compensating their extrinsic poses. Using the extracted contours and the corresponding image observations, we obtained a set of $3,000$ annotated training data. All training images are aligned to the center of mass. Some examples are shown in upper row of Figure 4.3. Training λ_0 is easier than λ_1, since the alignment step is not needed, and a set of $10,000$ training data were collected randomly from the training sequences to train λ_0. Some of them are shown in the bottom row of Figure 4.3.

In addition to these annotated training data, we also use $30,000$ un-annotated data to tune the model, based on the method described in Section 4.5.

It is important to know if the trained Boltzmann field model really captures the true shape prior $p(\mathbf{X})$. Although there is no quantitative means to validate that, a plausible way for a rough validation is to sample the learned prior Boltzmann distribution $p(\mathbf{X})$ and the learned image likelihood distribution $p(\mathbf{Z}|\mathbf{X})$, and then perform a subjective evaluation. To synthesis an image, we first draw a sample of $\mathbf{X} = \{x_1, \ldots, x_n\}$ by Gibbs sampling from $p(\mathbf{X})$ in Eq. 4.2, then for each x_i, a sample of z_i is drawn from $p_i(z_i|x_i)$. Putting together z_i produces a synthesis image. Through our subjective evaluations, the

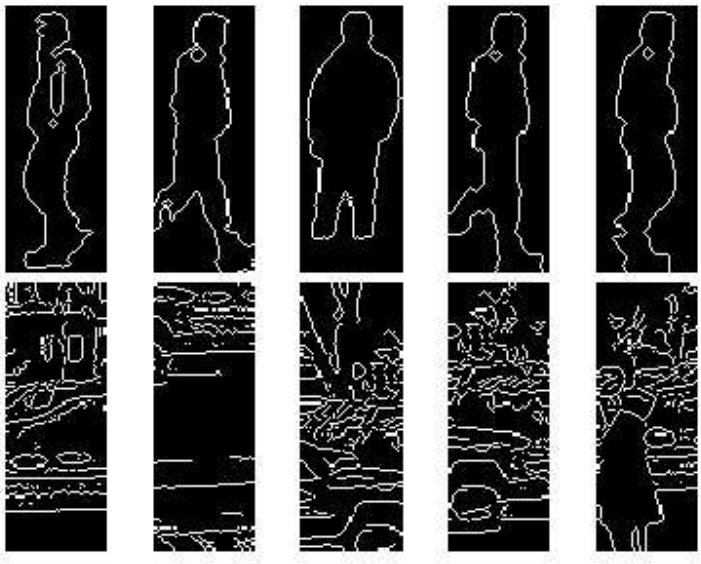


Figure 4.3. The upper row are examples of annotated training data for human λ_1, and the bottom row for nonhuman λ_0.

trained models were able to synthesize reasonably good data. Some synthesized data based on λ_1 and λ_0 are shown in Figure 4.4.

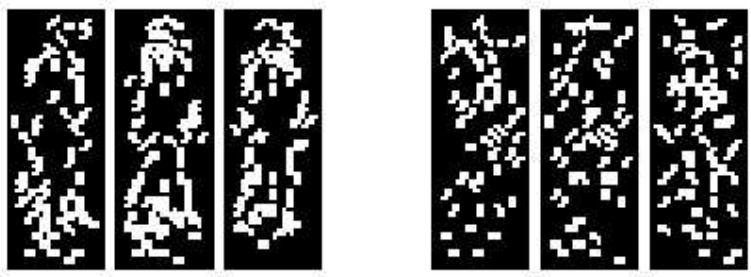


Figure 4.4. Examples of synthesized data. Left ones are sample from λ_1 and right ones from λ_0.

4.7.2. Pedestrian Detection

We performed extensive experiments and quantitative evaluation of the proposed approach to pedestrian detection, and we are particularly interested in the investigation

of the capacity of this field model of capturing the tremendous shape variations and its performance and robustness to partial occlusions.

4.7.2.1. Performance Evaluation. To provide quantitative evaluation of the proposed approach, we constructed a testing database which contains $1,000$ images collected from various occasions. We manually annotated the ground truth detection for each image. The ROC curve is shown in figure 4.5. This curve shows that at 80% detection rate, the detector has a false positive rate of about $1/200,000$ which corresponds to about one false alarm per frame for 320×240 images. This is comparable to the most recent method reported in [129].

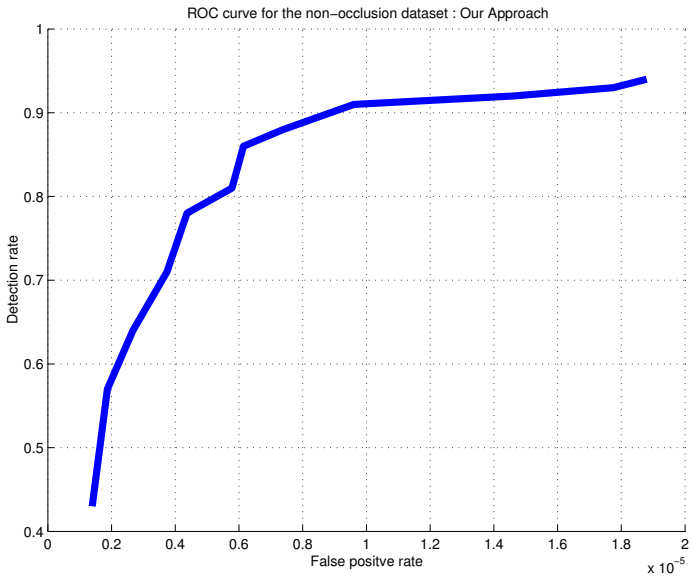


Figure 4.5. ROC curve of the proposed pedestrian detector.

Our extensive experiments show that the proposed field model is capable of capturing the nonrigidity caused by the view changes of the pedestrian. In our test data, there are a large volume of images where the pedestrians present various profiles. Some of the

detection examples are shown in Figure 4.6. The algorithm can also easily detect multiple targets. Some examples are shown in Figure 4.6. In the bottom right image, a false alarm was observed. In these results, the algorithm did not detect the sitting persons and the cyclist. This is reasonable, because the upper body of these examples are largely inclined and our training set did not contain such cases.

In addition, this field model is also able to detect the target from various environments. Some of the results are shown in Figure 4.7. The robustness comes from the observation models of λ_1 and λ_0. We did observe the case where in a region the edge map is pervasive and it is impossible to tell from the edge map where the person is.

Besides detection, a question of great interests is to reveal the value of hidden field. For each detected region, when we display the corresponding mean field $\{\mu_i\}$, a clear pedestrian contour can be seen. Some examples are shown in Figure 4.8.

In addition, the proposed detection algorithm is efficient. Currently, our un-optimized C++ implementation runs about 2 frame/second on a Pentium IV 2GHz PC for 320×240 images. We believe there are much room for improving the implementation. Beyond most existing methods, the proposed field model enables parallel computing, since the mean field updating on the set of sites is intrinsically parallel. In addition, when building real systems, we can easily combine the proposed detection method with background subtraction, motion detection or other remedies to further reduce the false alarm rate while not decreasing the detection rate. (We did not perform these postprocessing in our experiments, in order to provide a true ROC of the new model.)

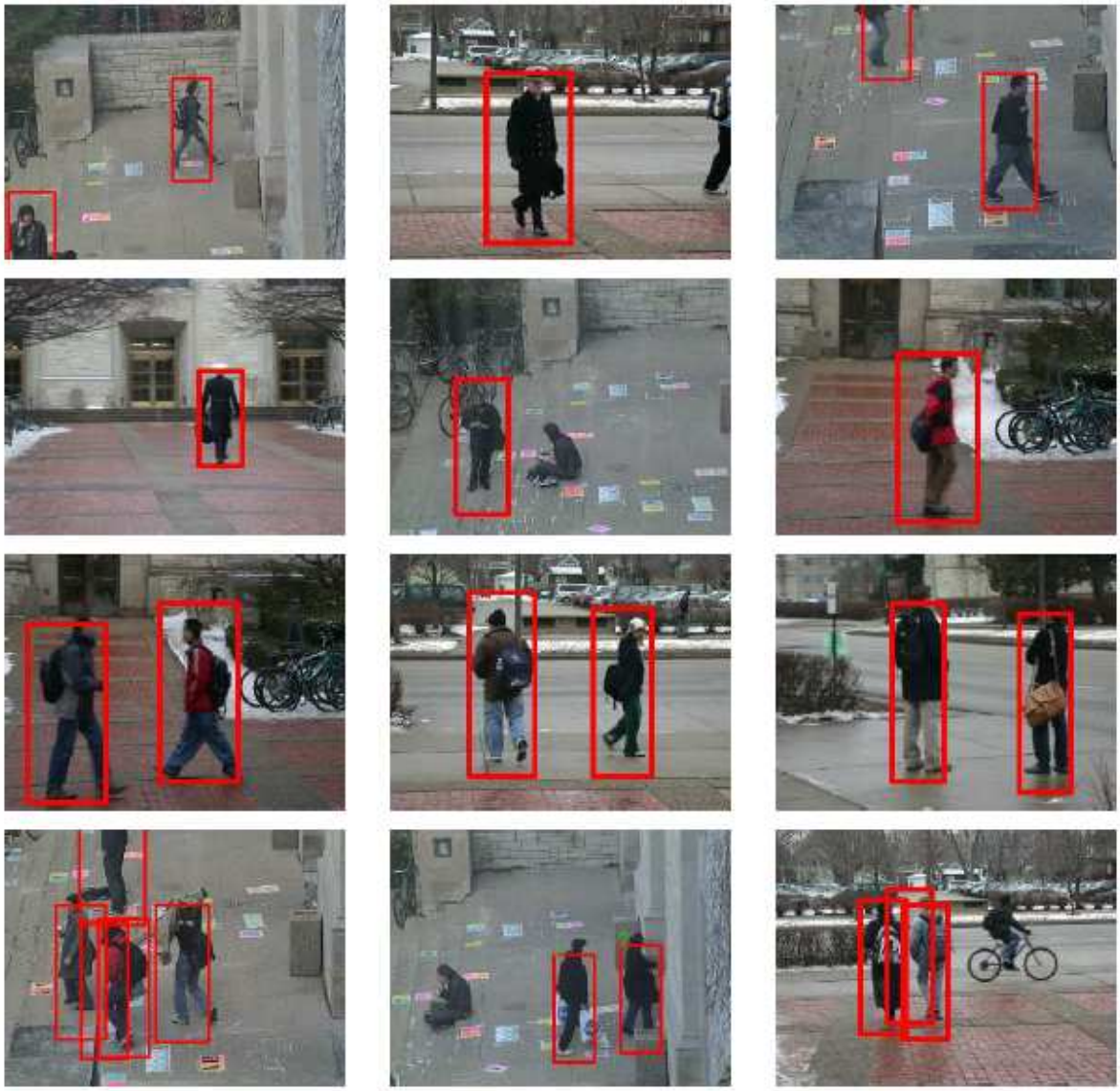


Figure 4.6. Pedestrian detection under various views.

4.7.2.2. Evaluation on Partial Occlusion. More interestingly, the proposed field model works well even when the target is partially occluded. Samples results on the detection under occlusion are shown in Figure 4.9. This feature is unique, since the robustness to partial occlusion is an intrinsic benefit of the proposed field model. This is

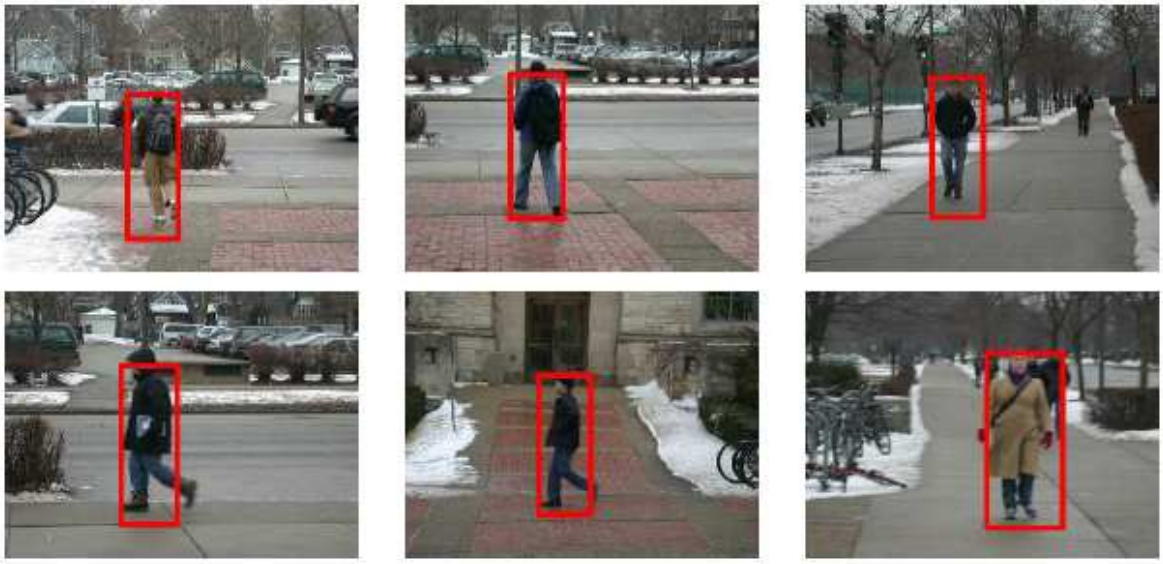


Figure 4.7. Pedestrian detection in various environments.

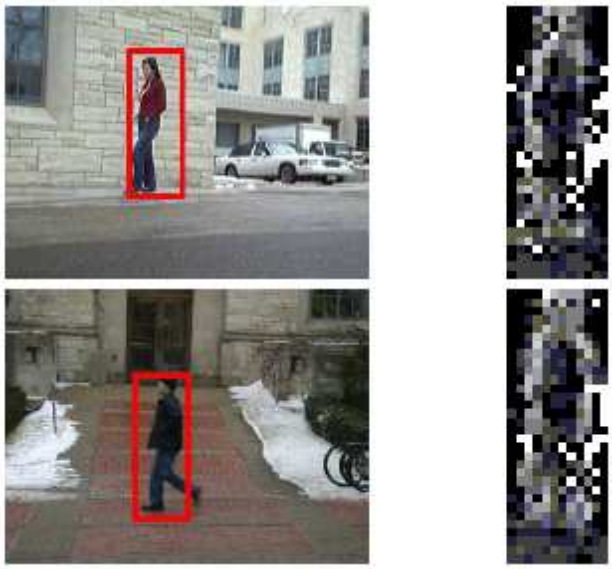


Figure 4.8. The mean field inference of the hidden Markov field. The right column shows the estimated mean field $\{\mu_i\}$ of the detected regions on the left column.

truly owe to the property of the field model, because we did not deliberately include the occlusion cases in training data. On the contrary, vectorized shape models, such as the

active shape models [26], can not cope with this problem, since it is generally infeasible to include all possible occlusion situations in training.

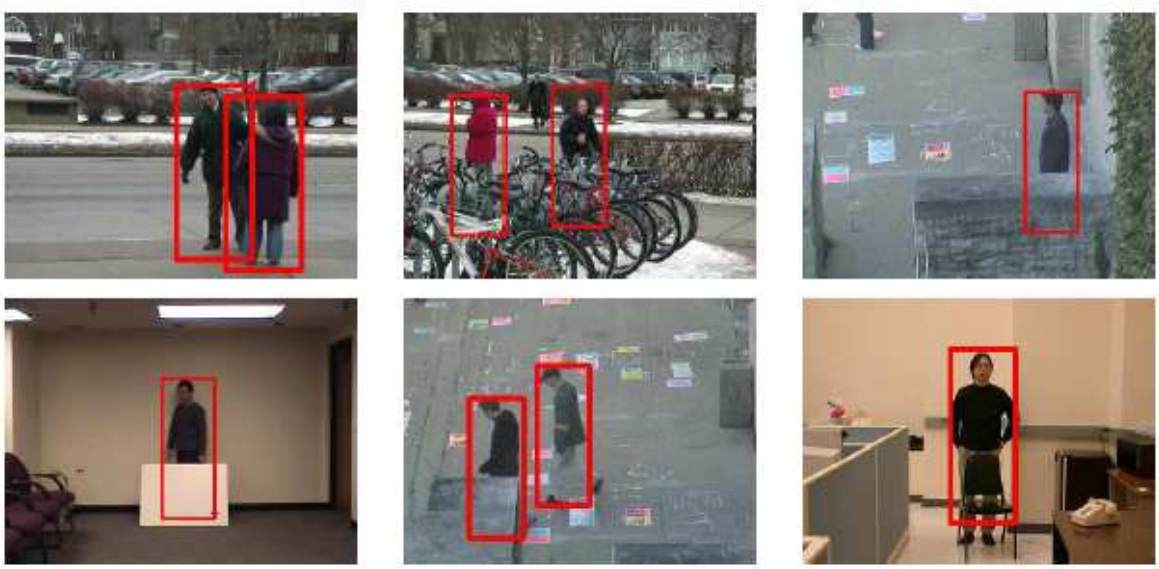


Figure 4.9. Sample results of pedestrian detection under partial occlusions.

To have a quantitative study on the robustness of our method, we created another testing database which consists of 3 subsets, each of which contains 100 images under a certain rough percentage of occlusion (less than 20%, between 20% and 40%, and over 40%, respectively). The ROC curves for these occlusion cases were obtained and shown in figure 4.10.

These ROC curves show that the performance of the proposed method does not degrade much when the percentage of occlusion is under 40%, since 80% detection rate can be achieved with comparable false positive rate as the case without occlusions. But when the occlusion is over 40%, the detection rate drops a lot. Although such quantitative measures are rough, they do verify the robustness of the proposed approach to partial occlusions.

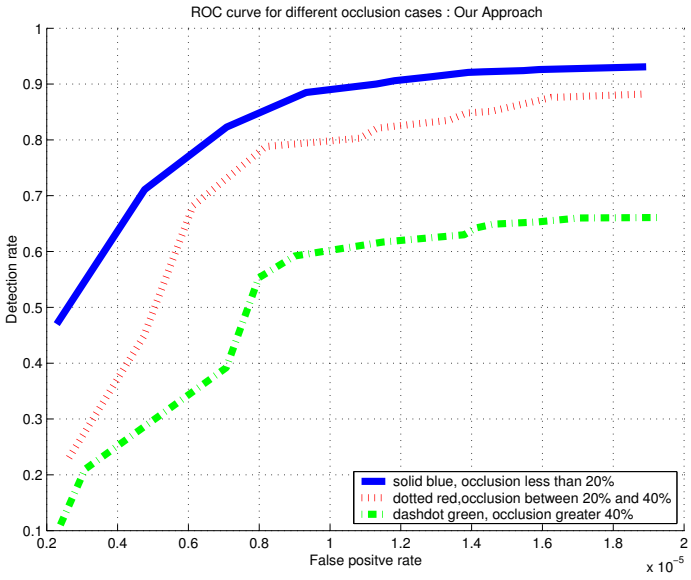
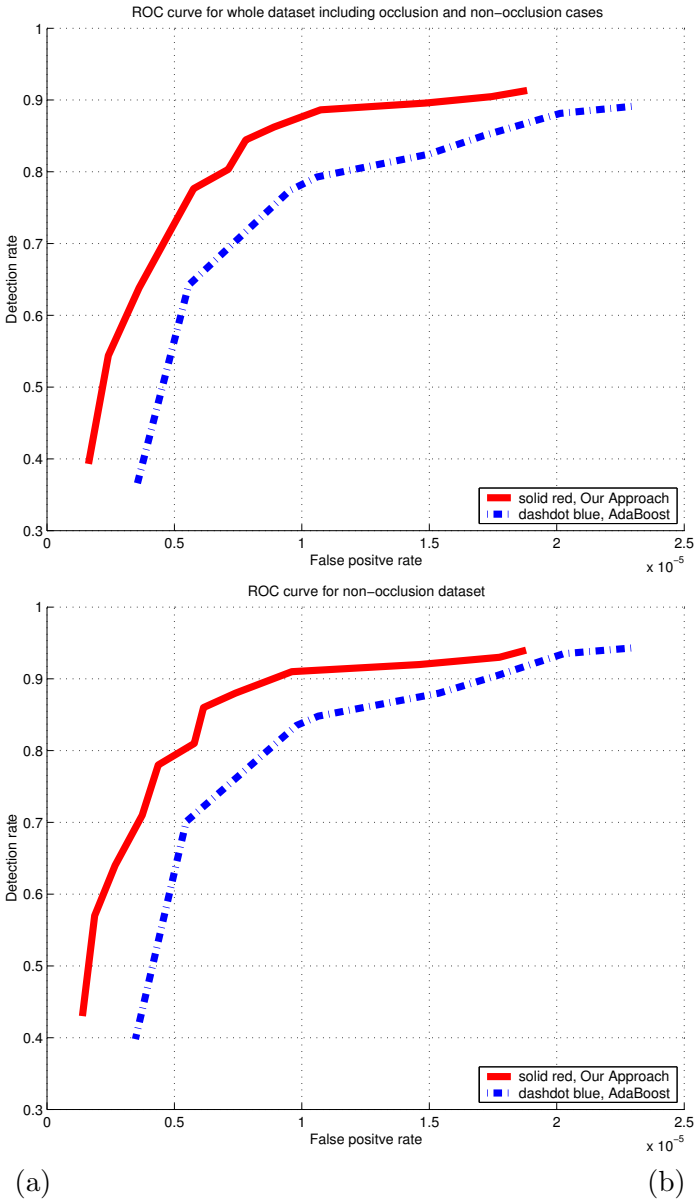


Figure 4.10. ROC curves on the three testing subsets under difference occlusion percentages.

4.7.2.3. Comparison with the AdaBoost Detector. We compare the performance of the proposed method with the AdaBoost detector [128], which is by far one of the best for face detection and is widely used for various object detection tasks. To have a comprehensive comparison, we used six different data sets:

- Data set A is a set of $1,000$ images including both non-occlusion and occlusion cases;
- Data set B is a set of $1,000$ images of non-occlusion cases only;
- Data set C is a set of 300 images of various occlusion cases;
- Data set D is a set of 100 images, each of which presents over 40% occlusion;
- Data set E is a set of 100 images, each of which presents 20% to 40% occlusion;
- Data set F is a set of 100 images, each of which presents less than 20% occlusion;

The ROCs on these data sets are shown in Figure 4.11 and Figure 4.12.



(a) (b)

Figure 4.11. ROC curves on Data set A and B.

Figure 4.11(a) shows the two ROCs on data set A that contains a mixture of non-occlusion and occlusion cases, and Figure 4.11(b) on data set B of all non-occlusion cases. These ROCs show that our method has overall $5\% - 10\%$ higher detection rate than the

AdaBoost detector. With high false alarm rates, both methods have high detection rates (over 90%).

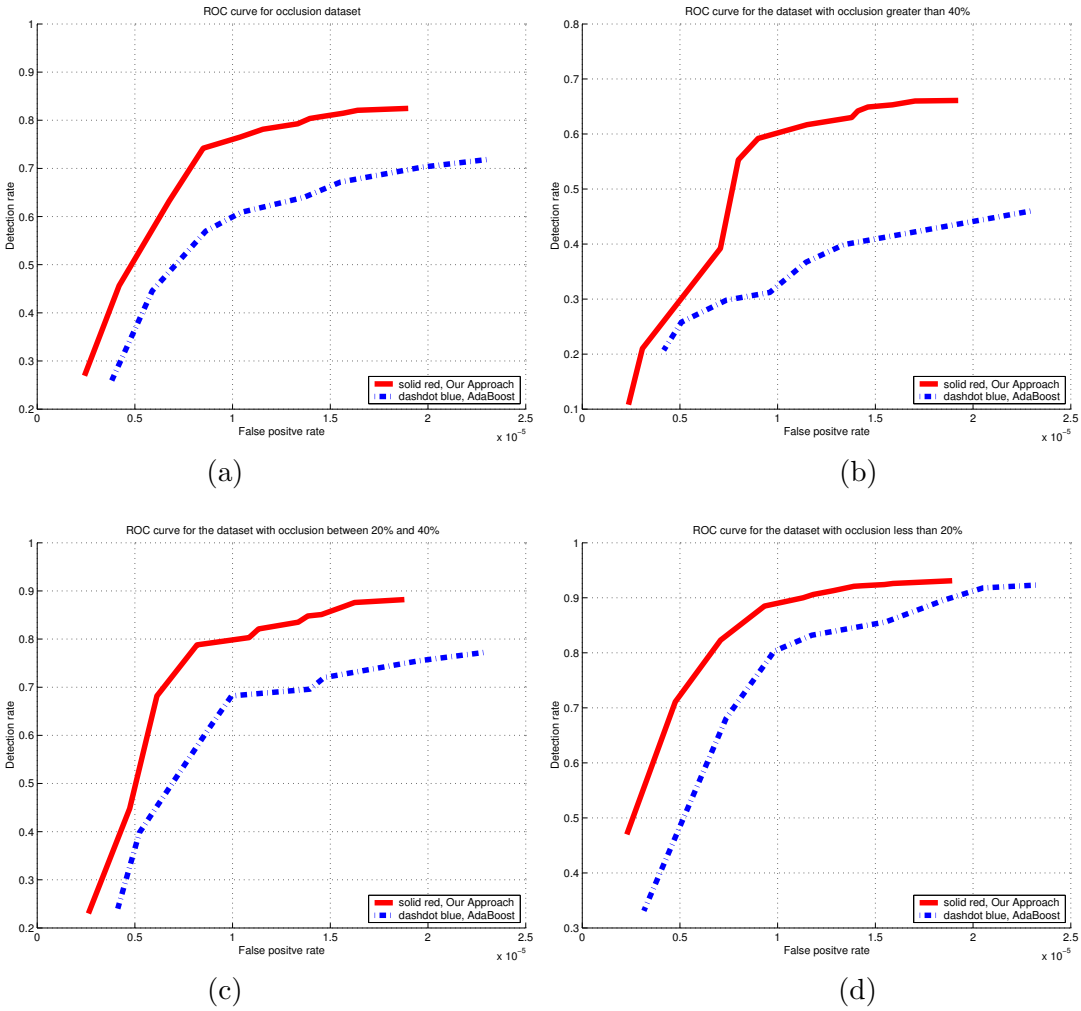


Figure 4.12. ROC curves on Data set C, D, E and F.

Figure 4.12(a) shows the ROCs on data set C, and gives the comparison of the two methods on general occlusions. It is apparent from this figure, our method significantly outperforms the AdaBoost detector. With high false alarm rates, our method can obtain over 80% detection while Adaboost is merely 70%. Figure 4.12(b-d) show the ROCs

on various degrees of occlusions. If the target present over 40% occlusion, AdaBoost detector can hardly work, while our method has over 60% detection. When the target has a moderate occlusion (between 20% and 40%), our method also significantly outperforms AdaBoost. When the occlusion is less than 20%, the two methods are comparable, but our method is slightly better.

4.7.3. Pedestrian Tracking

Tracking nonrigid objects is a challenging problem, especially when the camera is not fixed and the target presents large shape variances, as in the demonstration of this section. Since the mean field approximation also gives the data likelihood (given a global motion) by integrating out all possible local nonrigidity, this is powerful and ideal for tracking nonrigid targets, as described in Section 4.6.2. We did extensive experiments and verified this idea. In our experiments, a particle filter was applied to track the targets, i.e., to estimate the extrinsic pose parameters $\mathbf{y} = \{\mathbf{u}, \theta, s\}$ through the video. We used 400 particles. (One of the tracking sequences ‘‘`Tracking.avi`" is included in this submission.) Some sample frames are shown below in Figure 4.13. Actually, this is a difficult sequence for many tracking schemes. One difficulty is that the camera is not static, then the tracking methods based on background subtraction can not apply. In addition, when the pedestrian walks and rotates, the visual appearances change dramatically and present non-stationary characteristics, which is a very difficult problem for visual tracking in general. This example shows the effectiveness of the proposed field model. Our method can handle such a difficult scenario because the image likelihoods have integrated all the

shape deformations. The C++ implementation of the proposed tracking algorithm runs over 15 frames/second on a Pentium IV 2GHz PC.

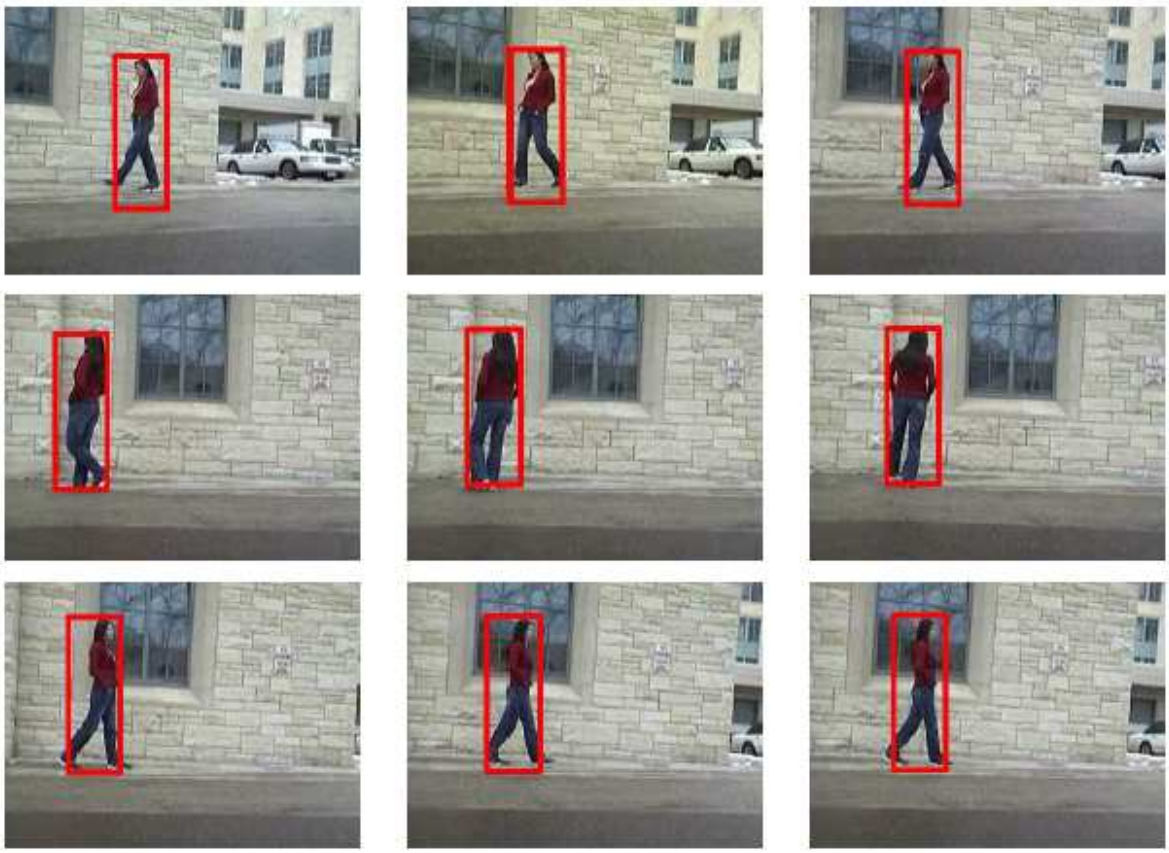


Figure 4.13. Tracking a nonrigid target based on the mean field Boltzmann model.

4.8. Discussions

Characterizing priors of nonrigid shapes is critical for analyzing nonrigid objects. Global or vectorized approaches such as PCA prove to be effective to capture global deformation by reducing global correlations. However, these vectorized models are neither suitable for handling local nonrigidity nor robust to partial occlusion, which are important for many real world applications such as pedestrian detection and tracking. In

this chapter, we proposed a new statistical method to capture the local nonrigidity based on a two-layer field model, where a Boltzmann distribution was employed to characterize the complicated prior for local variability, and a variational mean field approximation was presented for computationally efficient inference, likelihood calculation and model training. Due to the distributed likelihood model, this new field method is robust to occlusion. Based on the framework of this field model, the detection and tracking problems were also investigated and were successfully approached. The success of applying the proposed method to pedestrian detection and tracking showed its effectiveness and general applicability.

Aligning training data in the proposed approach is easier than labelling landmark data in the active shape model [26], but it leaves a problem: how sensitive is the trained model to the alignment errors? We leave it for further studies. In addition, in our future work, we plan to investigate the capacity of the proposed Boltzmann field model, i.e., to what extent the model can capture local nonrigidity. Moreover, better image observation models will be studied to reduce the false alarm rate.

CHAPTER 5

Component-based Robust Support Vector Tracking

5.1. Introduction

One of the major challenges of appearance-based tracking lies in the large uncertainties of the visual appearances that significantly complicate the measurement models and the matching measures of the visual objects. This difficulty is also shared in object detection and recognition, and has been studied extensively. To address this problem in the context of tracking, we should not let the handling of this problem to jeopardize the requirement of computational efficiency of the tracker.

An early work of integrating classification methods and tracking methods is the support vector tracker (SVT) [4] and its more recent advance [131] that have shown their efficacy of long duration tracking. It nicely combines the support vector machine and the differential method, such that the variability of the target can be learned from the training data and the gradient-based search can be used for efficient matching.

SVT works well when the uncertainty of the target's appearances can be managed by a support vector machine, i.e., the target can be well discriminated from non-target by using a reasonable number of support vectors. However, the applicability of SVT is limited in many real applications where targets may exhibit enormous appearance changes. For example, pose changes, partial occlusions, and clutter backgrounds significantly complicate the handling of the uncertainties in the appearances [90, 4]. Even if a SVM can

be trained to cover such tremendous variations, the number of support vectors will be very large, thus making the computation demanding and not practical. Though object pose changes may be dealt with by some special treatment, such as training multi-view SVMs [79, 80], partial occlusion, a more challenging situation to object classification, has plagued SVT as well. As mentioned in [4], since occlusion leads to intractable variability on the appearance, it is infeasible for a SVM to learn the occlusion cases that are virtually limitlessness.

In general, larger image regions incur more variabilities, while the uncertainties of smaller regions are more likely to be manageable. In other words, although the appearances of the entire target have large variabilities, it is likely that its many components or parts do not change much; otherwise, it does not make sense to recognize and track such a target. Therefore, it seems to be a good idea to replace the problem of learning complex uncertainties by the problem of managing a set of learning tasks with much less uncertainties. This treatment is expected to reduce the complexity of learning as well as the enhancement of classification accuracy. There have been many studies on this line in component-based object detection [1, 58, 76, 88, 90, 113, 144]. Similar component-based ideas were also proposed for human pose analysis to divide-and-conquer the complex visual measurement process and high dimensionality of the body configuration through pictorial structure [42, 41, 135].

This chapter proposes a novel solution of pursuing this component-based idea in visual tracking with the emphasis on the handling of the challenges such as partial occlusions in a computationally efficient way. Different from the method of using a single SVT for the entire object, our solution is based on the *collaboration* of a set of correlated simple

component SVTs, each of which has much less support vectors, thus the computation can be greatly saved. Besides this intuition, a central issue of our solution is: how can a set of component SVTs be optimally integrated?

This chapter presents an elegant answer to this central issue by giving analytical results that reveal the mechanism of the integration of the set of component SVTs, and by providing a computationally efficient collaborative SVT (i.e., the CSVT algorithm). Analytically, the motion estimation of a component is determined by two terms (1) the motion estimated by the SVT tracker associated with it, and (2) a compensation term passed by its neighborhood components. The estimations of all components encompass a fixed point system, where the fixed point gives the optimal solution of the motion estimation of the target. In addition, by investigating the behaviors of this CSVT tracker under partial occlusions, we further propose to enhance the model by a selective mechanism, which can automatically select trustworthy components while down-weight the unreliable ones that may be occluded. This new tracking method produces very promising results in our experiments.

The chapter is structured as follows: Section 5.2 provides a brief overview of the original SVT method. Section 5.3 describes the formulation and solution of the proposed collaborative SVT algorithm. The further extension of the CSVT model to handling severe occlusions is discussed in section 5.4. Section 5.5 shows experimental results. Conclusions and discussions are made in Section 5.6.

5.2. Support Vector Tracker, SVT

In the following we briefly review the original SVT algorithm. In SVT, an object model based on SVM is firstly trained from the set of training data, and then applied to object tracking. Given the initial guess of the object location for each frame, which could be the one estimated and predicted from the previous frame, the SVT algorithm aims at finding an updated object location in the current frame, where the cropped image patch from this location maximizes the object SVM score. Instead of taking an exhaustive searching to find the optimal location, the SVT algorithm enjoys an efficient gradient searching strategy, thus achieving a fast computation.

Although the original SVT algorithm in [4] only addresses the translational motion model, here we present the SVT in a general motion formulation, which can be degenerated to any special cases, such as translation, similarity, affine motion model.

Let $I(p,t)$ denote the image intensity at the pixel location $p = (x,y)^T$ of time t. Assume the initial object region at time t_0 is defined by a set of N pixels $\mathcal{R} = \{p_1, p_2, \ldots, p_N\}$, and $p_i = (x_i, y_i)^T$ is the i_{th} pixel location. We denote the motion model of the object region as $f(p;\mu)$, where $\mu = \{\mu_1, \mu_2, \ldots, \mu_n\}^T$ is the motion parameters.

Suppose at time t, we have an initial guess of the object motion parameters as $\mu(t)$, which could be the estimation from previous time $t-1$, then the image patch corresponding

to this initial guess is ready to be defined as

$$I(\mu(t),t) = \begin{bmatrix} I(f(p_1;\mu(t)),t) \\ I(f(p_2;\mu(t)),t) \\ \cdots \\ I(f(p_N;\mu(t)),t) \end{bmatrix} \tag{5.1}$$

SVT algorithm aims to find an optimal update of the motion parameters $\Delta\mu$, which, after adding to the initial guess $\mu(t)$, reports an image region $I(\mu(t)+\Delta\mu,t)$ that maximizes the corresponding SVM score, i.e., we are dealing with the following optimization problem for each frame:

$$\begin{aligned} \max_{\Delta\mu} E(\Delta\mu) &= SVM(I(\mu(t)+\Delta\mu,t)) \\ &= \sum_{j=1}^{l} c_j\alpha_j\mathcal{K}(s_j, I(\mu(t)+\Delta\mu,t)) + b \end{aligned} \tag{5.2}$$

where l is the number of support vectors, and each is denoted by s_j. c_j is the training data label, i.e. $(-1,+1)$, α_j is the lagrange multiplier corresponding to the weight of the j_{th} support vector, and b is the constant intercept. All these parameters are learned from the training data set. $\mathcal{K}(s_j, I)$ is the chosen kernel function. As shown in [4,3], a second order polynomial kernel can lead to a set of linear equations to solve for an optimal $\Delta\mu$.

By taking the small motion assumption, it is easy to show that $I(\mu(t)+\Delta\mu,t)$ can be linearly approximated by first order Taylor expansion as follows [52]:

$$I(\mu(t)+\Delta\mu,t) = I(\mu(t),t) + M(\mu(t),t)\Delta\mu + h.o.t \tag{5.3}$$

where $h.o.t$ stands for the higher-order terms of the expansion, then can be omitted. $M(\mu(t),t)$ is the Jacobian matrix of $I(\mu(t),t)$ with respect to the motion parameters μ, and evaluated at $\mu(t)$. $M(\mu(t),t)$ is an $N \times n$ matrix that has the following column form

$$M(\mu(t),t) = [I_{\mu_1}(\mu(t),t)|I_{\mu_2}(\mu(t),t)|\dots|I_{\mu_n}(\mu(t),t)] \tag{5.4}$$

Each term is the partial derivative $I(\mu(t),t)$ over $\mu_i, i = 1,\dots,n$.

Please note in Eq. 5.3, the Taylor expansion is done in a single frame at time t, i.e. we only care about the Taylor expansion over the spatial domain, which facilitates the iterative gradient ascent updating as shown in [4]. Actually, the similar single frame updating strategy is also taken in several other tracking approaches, such as kernel-based tracking [24]. Therefore, in the later derivations, we drop the argument time t.

Plugging Eq. 5.3 into the SVT objective function Eq. 5.2, we have

$$\begin{aligned} \max_{\Delta\mu} E(\Delta\mu) &= \sum_{j=1}^{l} c_j \alpha_j \mathcal{K}(s_j, (I(\mu) + M(\mu)\Delta\mu)) + b \\ &= \sum_{j=1}^{l} c_j \alpha_j (s_j^T (I(\mu) + M(\mu)\Delta\mu))^2 + b \end{aligned} \tag{5.5}$$

where we adopt a second order polynomial kernel. Taking the partial derivative over $\Delta\mu$ gives

$$\frac{\partial E}{\partial \Delta\mu} = 2P + 2Q\Delta\mu \tag{5.6}$$

where the above two terms P, Q are defined as

$$\begin{aligned} P &= \sum_{j=1}^{l} c_j \alpha_j s_j^T I(\mu) M(\mu)^T s_j \\ Q &= \sum_{j=1}^{l} c_j \alpha_j M(\mu)^T s_j s_j^T M(\mu) \end{aligned} \tag{5.7}$$

and the following linear solution to $\Delta\mu$ can be derived

$$\Delta\mu = -Q^{-1}P \tag{5.8}$$

where $\Delta\mu$ is a n dimensional vector that updates the previous estimation of the motion parameter μ, i.e. $\mu' = \mu + \Delta\mu$. The new μ' then serves as an initial point to start a new cycle of the optimization. This updating is conducted iteratively by continuously increasing the SVM objective function E until the newly derived $\Delta\mu$ becomes sufficiently small or maximum iterative times is reached. The convergent estimation of the object motion $\mu(t)^*$ then becomes an initial guess of the motion for new frame at time $t + 1$.

5.3. Collaborative SVTs

Treating the object as a whole, like many existing image region classifiers did, the original SVT algorithm needs to train a discriminative model that is capable of robustly classifying the object class with diversified appearance patterns. This training task itself is known to remain as a very challenging problem. In addition, illumination variations, object pose changes, clutter backgrounds, and partial occlusions all bring even more challenges to train such a holistic object classifier.

The component-based representation, however, offers the advantages of being able to achieve a more robust detection [58, 90]. It usually suffers less from the tremendous appearance variations, induced by either internal (object itself) or external (environmental) reasons. We are interested in exploiting the extension of this component-based representation to SVT tracking framework, which, when combining them in the right way, will benefit both of them. For example, the advantages of component-based representation, such as insensitive to cluttered backgrounds and partial occlusions, can effectively remedy the defects of original SVT algorithm. On the other hand, the component-based representation may also enjoy the availability of estimation initialization propagated from SVT tracking results of previous frames. Hence, a single stage optimization can be achieved, which means we can simultaneously accomplish the object component detections and geometric verification of these parts. Such a treatment will reduce the information loss encountered by the previous two-stage component-based detection approaches, where the first stage applies the trained component classifiers to collect component candidates, and the second stage is to take a validation scheme using learned geometric configuration to verify the combination of these component candidates.

Assume we are taking a K-component representation for the object class. A corresponding SVM classifier is trained for each component, as did in [58,90]. By following the notations in the previous section, we collect a set of SVM classifiers, each characterized by their parameter set $(l^k, s_j^k, c_j^k, \alpha_j^k, b^k)$, where $k = 1, \ldots, K$ indexes different component.

The motion parameters of each component at time t are denoted by $\mu^k(t)$, whose initial values are the estimation from the same component at previous frame. Please note that although we endow each component a separate motion representation $\mu^k(t)$, considering

the fact that different components are all correlated to the same object during the tracking initialization, there must be some underlying constraint among these parts. Otherwise, if arbitrary motion value is allowed for each component, the whole object may become impractically deformed due to free motions of the components.

If the object being tracked is a rigid one, the motion parameters of different components after each frame updating should be equal to each other, i.e. $\mu^i(t) + \Delta\mu^i = \mu^j(t) + \Delta\mu^j, i, j = 1, \ldots, K$. Based on this observation, we propose a modified SVT objective function to not only incorporate the SVM score term from each component, but also explicitly include a penalty term to pose the geometric constraints of different components, which has the formulation as follows

$$\begin{aligned}
&\max_{\Delta\mu^1,\ldots,\Delta\mu^K} E(\Delta\mu^1, \ldots, \Delta\mu^K) \\
&\quad = E_1(\Delta\mu^1, \ldots, \Delta\mu^K) + E_2(\Delta\mu^1, \ldots, \Delta\mu^K) \\
&\quad = \sum_{k=1}^{K} \{ \sum_{j=1}^{l^k} c_j^k \alpha_j^k \mathcal{K}(s_j^k, I^k(\mu^k + \Delta\mu^k)) + b^k \} \\
&\qquad - \gamma(\Psi(\mu^1 + \Delta\mu^1, \ldots, \mu^K + \Delta\mu^K))^2
\end{aligned} \tag{5.9}$$

where $\sum_{j=1}^{l^k} c_j^k \alpha_j^k \mathcal{K}(s_j^k, I^k(\mu^k + \Delta\mu^k)) + b^k$ is the SVM term for each component k, and we denote the sum of the SVM terms from all components as $E_1(\Delta\mu^1, \ldots, \Delta\mu^K)$. The second term $E_2(\Delta\mu^1, \ldots, \Delta\mu^K)$ contains a geometric structure constraint $\Psi(\mu^1 + \Delta\mu^1, \ldots, \mu^K + \Delta\mu^K)$. γ is a tradeoff factor to balance the relative weights of SVM scores and motion consistency constraint. The constraint may take any form as long as the required structure configuration is embedded, for example, the following is a specific pair-wise component

motion constraint

$$\Psi(\mu^1 + \Delta\mu^1, \ldots, \mu^K + \Delta\mu^K) = \sum_{i,j \in K, i \neq j} \|(\mu^i + \Delta\mu^i) - (\mu^j + \Delta\mu^j)\|^2 \tag{5.10}$$

The above objective function implies that the desired solution of $\Delta\mu^1, \ldots, \Delta\mu^K$ should not only maximize SVM scores of their respective components, but also minimize the motion discrepancy among them.

To solve the above optimization problem, we take the partial derivatives of the objective function over each $\Delta\mu^i$. If a quadratic polynomial kernel is chosen for each component SVM, the partial derivative of the first part of objective function in Eq. 5.9 can be readily written as

$$\frac{\partial E_1(\Delta\mu^1, \ldots, \Delta\mu^K)}{\partial \Delta\mu^i} = 2P^i + 2Q^i\Delta\mu^i \tag{5.11}$$

P^i, Q^i are the compact matrix representations as shown in Eq. 5.7 with added superscript denoting the component index.

In general, the motion constraint term $\Psi(\mu^1 + \Delta\mu^1, \ldots, \mu^K + \Delta\mu^K)$ may take any form with respect to the motion updating parameters $\Delta\mu^k$, either linear or nonlinear. However, considering the fact that $\Delta\mu^k$ is a continuous tracking update from successive frames, then usually takes very small value, a linearization of the motion constraint by first order Taylor expansion is therefore adequate to approximate the original constraint

equation.

$$\begin{aligned}&\Psi(\mu^1+\Delta\mu^1,\ldots,\mu^K+\Delta\mu^K)\approx\\&\Psi(\mu^1,\ldots,\mu^K)+\sum_{k=1}^{K}(\frac{\partial\Psi}{\partial\mu^k})^T\Delta\mu^k\end{aligned}\tag{5.12}$$

Please be aware that in above equations, μ^k is the previous estimation of the component motion from last iteration that has already been computed, thus $\Delta\mu^k$ is the only unknown that needs to be estimated. With Eq. 5.12, the partial derivative over $\Delta\mu^i$ from the second part of the objective function Eq. 5.9 can be written as

$$\begin{aligned}&\frac{\partial E_2(\Delta\mu^1,\ldots,\Delta\mu^K)}{\partial\Delta\mu^i}=\\&-2\gamma[\Psi(\mu^1,\ldots,\mu^K)+\sum_{k=1}^{K}(\frac{\partial\Psi}{\partial\mu^k})^T\Delta\mu^k]\frac{\partial\Psi}{\partial\mu^i}\end{aligned}\tag{5.13}$$

which is also a linear equation with respect to $\Delta\mu^k$.

Combining both Eq. 5.11 and Eq. 5.13 together, and setting the derivative to zero, with some mathematical manipulations the following linear equation of the motion update for each component can be derived

$$\Delta\mu^i=-(Q^i)^{-1}P^i+[-C^iP^i+\gamma((Q^i)^{-1}+C^i)D^i]\tag{5.14}$$

with C^i, D^i respectively defined as follows

$$\begin{aligned}C^i&=\frac{(Q^i)^{-1}(\sqrt{\gamma}\frac{\partial\Psi}{\partial\mu^i})(\sqrt{\gamma}\frac{\partial\Psi}{\partial\mu^i})^T(Q^i)^{-1}}{1-(\sqrt{\gamma}\frac{\partial\Psi}{\partial\mu^i})^T(Q^i)^{-1}(\sqrt{\gamma}\frac{\partial\Psi}{\partial\mu^i})}\\D^i&=\sum_{k=1,k\neq i}^{K}(\frac{\partial\Psi}{\partial\mu^i})(\frac{\partial\Psi}{\partial\mu^k})^T\Delta\mu^k+\Psi\frac{\partial\Psi}{\partial\mu^i}\end{aligned}\tag{5.15}$$

Intuitively, this set of K linear equations Eq. 5.14 define an iterative fixed point updating mechanism for each component motion $\Delta\mu^i, i = 1, \ldots, K$, and is actually very meaningful, which reveals a collaborative algorithm to this general joint multi-component motion estimation problem. The procedure shows that the motion estimation of each component is not only determined by its local SVM score maximization, i.e. the estimation from its own SVT tracker, governed by the term $-(Q^i)^{-1}P^i$, which shares the same form as in Eq. 5.8, but also through a compensation term $[-C^iP^i+\gamma((Q^i)^{-1}+C^i)D^i]$ that is consisted with the motion estimations of other geometric constraint components. One advantage of this motion updating equation is that we can still make use of the original SVT tracking implementation, i.e., the term $(-Q^i)^{-1}P^i$, with only minimum modification due to the introduction of geometric constraint. The compensation term serves as some kind of "message" from other components to tune the motion estimation of the current component to make it become consistent with the estimations of others. The iterative solutions allow the set of SVT trackers to work individually while collaboratively to achieve a more robust object tracking framework, which can better handle large appearance variations and background clutter distractions.

5.4. Partial Occlusion Invariant SVTs

In the previous section, we show that the set of component SVT trackers introduce an interestingly collaborative mechanism, which can hopefully achieve higher robustness to object appearance variations, illumination changes and cluttered backgrounds.

However, when partial occlusion happens during tracking, by closely looking at the SVM term of each component in the objective function Eq. 5.9, it is not difficult to notice

that the SVM scores of those occluded components usually decline to negative values. It implies the situation that the image regions corresponding to those missing components are classified as non-object components, which, from the component classifiers' point of views, are actually not incorrect decisions, because what have been feeded to them are actually the occluding image patches that in most cases are obviously not similar to those training object components. The iterations of SVM maximization for these components are then mainly conducted in image areas that have very slim probabilities of showing the appearance patterns similar to the components due to occlusions. Under these circumstances, it apparently makes no sense to still search for an optimal solution to maximize SVM values around these occluded regions, which, even found, may still be the very negative values, simply meaning non-object components.

A better strategy of dealing with this partial occlusion is to trust more on those unoccluded object components, and propagate their accurate motion estimates to the occluded ones via geometric structure constraint. This way, when it is done in the right manner, will help maintain the roughly correct motion estimates of the occluded components, even there are no positive confidence supports from their own SVM terms. On the contrary, the occluded components should also reduce their distracting influences on the motion estimates of the unoccluded ones.

Such a "message" propagation mechanism with more favoring on the trustworthy information sources calls for some triggering scheme to determine which components of the object are currently valid, and which are not, i.e., we need to predicate the occlusion

situation of every component. It can be naturally thought of that the SVM scores reported from the component classifiers are the most qualifying indicators of these appearing occlusions.

Based on the above analysis, we modify the objective function defined in Eq. 5.9, and explicitly introduce the idea of favoring the SVM terms of the unoccluded components determined by their corresponding SVM scores. It leads to the following new formulation of the objective function

$$\begin{aligned}
&\max_{\Delta\mu^1,\ldots,\Delta\mu^K} E(\Delta\mu^1,\ldots,\Delta\mu^K) \\
&\quad = \sum_{k=1}^{K} \max\{SVM(\Delta\mu^k), 0\} - \gamma(\Psi(\mu^1+\Delta\mu^1,\ldots,\mu^K+\Delta\mu^K))^2 \\
&\quad = \sum_{k=1}^{K} [SVM(\Delta\mu^k)]^+ - \gamma(\Psi(\mu^1+\Delta\mu^1,\ldots,\mu^K+\Delta\mu^K))^2 \qquad (5.16) \\
&\quad = \sum_{k=1}^{K} [\sum_{j=1}^{l^k} c_j^k \alpha_j^k \mathcal{K}(s_j^k, I^k(\mu^k+\Delta\mu^k)) + b^k]^+ - \\
&\qquad \gamma(\Psi(\mu^1+\Delta\mu^1,\ldots,\mu^K+\Delta\mu^K))^2
\end{aligned}$$

where $SVM(\Delta\mu^k)$ is the SVM score term for each component k. In comparison with Eq. 5.9, the modified part is that we add a nonlinear max operation over the component SVM term, i.e. $\max\{SVM(\Delta\mu^k), 0\}$, which is compactly described as $[SVM(\Delta\mu^k)]^+$. This modification implies that during the iterations of the optimization, if the SVM score of any one component is less than 0, the max operation will immediately substitute 0 with that term, i.e. the potential negative contribution from this SVM term will not be included for the next iteration. However, if the SVM score of one component is greater than 0, the optimization will take it into consideration, which also propagates the positive influence of

this component to all others via the geometric structure constraint. The optimization aims at maximizing the collaboration benefits through communicating the valid "messages" among components, and simultaneously reduces the potentially unreliable information propagations from those occluded components.

The introduction of $[x]^+$ operation leads to a general nonlinear optimization problem with non-smooth objective function in Eq. 5.16 [93], where the first order derivative is not continuous, thus limiting the use of gradient-based method to iteratively find an optimal solution. This problem, however, can be combated, if we choose to take the quadratic form of the modified SVM score, i.e. $([SVM(\Delta\mu^k)]^+)^2$. The first order derivative of this modified SVM term does exist, and is the continuous function of $\Delta\mu^k$ with $\frac{\partial([SVM(\Delta\mu^k)]^+)^2}{\partial\Delta\mu^k} = 0$ for $SVM(\Delta\mu^k) \leq 0$. The gradient ascent approach applied in previous sections then is still capable to solve this optimization problem. The objective function becomes the form as follows

$$\begin{aligned} &\max_{\Delta\mu^1,\ldots,\Delta\mu^K} E(\Delta\mu^1,\ldots,\Delta\mu^K) \\ &= \sum_{k=1}^{K}([\sum_{j=1}^{l^k} c_j^k \alpha_j^k \mathcal{K}(s_j^k, I^k(\mu^k + \Delta\mu^k)) + b^k]^+)^2 - \\ &\gamma(\Psi(\mu^1 + \Delta\mu^1,\ldots,\mu^K + \Delta\mu^K))^2 \end{aligned} \tag{5.17}$$

Please note that such a quadratic modification of the original objective function Eq. 5.16 does not change the property of this optimization task. The only thing that may be affected by this modification is the tradeoff factor γ, which is used to balance the contributions of SVM scores and motion consistency constraint. The value of γ, however, can be empirically adjusted in the experiments.

In order to still achieve a linear solution to the optimization problem defined above, we change the previous second order polynomial kernel function of the SVM classifier to the first order, i.e. dot kernel $\mathcal{K}(s_j, I) = s_j^T I$. With this dot kernel, the SVM term $SVM(\Delta\mu^i)$ can be simplified as

$$SVM(\Delta\mu^i) = P^i + Q^i\Delta\mu^i \tag{5.18}$$

with P^i, Q^i defined as:

$$\begin{aligned} P^i &= \sum_{j=1}^{l^i} c_j^i \alpha_j^i (s_j^i)^T I^i(\mu^i) + b^i \\ Q^i &= \sum_{j=1}^{l^i} c_j^i \alpha_j^i (s_j^i)^T M^i(\mu^i) \end{aligned} \tag{5.19}$$

By following the similar mathematical manipulations as done in Section 5.3, the following set of collaborative equations to iteratively update the motion estimate $\Delta\mu^i$ can be derived

$$\begin{cases} \Delta\mu^i = -(A^i)^{-1}B^i + [-C^iB^i + \gamma((A^i)^{-1} + C^i)D^i] \\ \qquad\qquad\qquad\qquad \text{if } P^i + Q^i\Delta\mu^i \geq 0 \\ \Delta\mu^i = -(E^i)^{-1}D^i \\ \qquad\qquad\qquad\qquad \text{if } P^i + Q^i\Delta\mu^i < 0 \end{cases} \tag{5.20}$$

with the coefficient matrixes defined as

$$
\begin{aligned}
A^i &= (Q^i)^T Q^i \\
B^i &= (Q^i)^T P^i \\
C^i &= \frac{(A^i)^{-1}(\sqrt{\gamma}\frac{\partial \Psi}{\partial \mu^i})(\sqrt{\gamma}\frac{\partial \Psi}{\partial \mu^i})^T (A^i)^{-1}}{1 - (\sqrt{\gamma}\frac{\partial \Psi}{\partial \mu^i})^T (A^i)^{-1}(\sqrt{\gamma}\frac{\partial \Psi}{\partial \mu^i})} \\
D^i &= \sum_{k=1,k\neq i}^{K} (\frac{\partial \Psi}{\partial \mu^i})(\frac{\partial \Psi}{\partial \mu^k})^T \Delta\mu^k + \Psi \frac{\partial \Psi}{\partial \mu^i} \\
E^i &= (\frac{\partial \Psi}{\partial \mu^i})(\frac{\partial \Psi}{\partial \mu^i})^T
\end{aligned}
\tag{5.21}
$$

It is interestingly pointing out that the collaborative updating equations defined in Eq. 5.20 clearly reflect our willing of favoring more on the trustworthy information sources and reducing the negative effects from the insecure ones. During the above fixed point iterations, suppose for some iterative step, the SVM term of one component drops down less than zero, meaning that the underlying region being tracked by that SVT tracker is potentially invalid, such as occluded, the updated motion parameters $\Delta\mu^i$ will then be more reliably determined by the "message" propagated from other components. On the other hand, if the SVM score of the component is greater than zero, then its own SVM confidence support will also contribute to the motion estimate of the component. Again, we observe that for these unoccluded components the motion estimates are composed with two terms, one from its own SVM term $-(A^i)^{-1}B^i$, and the other compensation term from geometric constraint $[-C^i B^i + \gamma((A^i)^{-1} + C^i)D^i]$. In comparison with the previous collaborative SVT updating equation shown in Eq. 5.14, the formulation is almost identical, where the only difference is due to the use of dot product kernel here.

5.5. Experiments

The proposed collaborative SVT algorithm is implemented to perform experiments on tracking human faces, and we compared the performance with the original holistic-based SVT tracker and a simple correlation-based SSD tracker. For SVM training, we collected approximately 1000 frontal face training data from different sources, including MIT face database, CVL face database, AT&T database, etc. We manually select three discriminant components of the face to train the SVM face component classifier, which are left eye, right eye, mouth respectively. Some samples of the facial component training data are shown in Figure 5.1. In comparison, a holistic frontal face SVM classifier is

Figure 5.1. Samples of the training data for face parts.

also trained based on the same data set for the single SVT algorithm. The performance records of the trained SVM classifiers for face components and holistic face are reported in the Table 5.1[1]. Consistent with our intuitions, since the smaller image region implies less

	L-Eye	R-Eye	Mouth	Face
Classification Rate	89.2%	91.4%	90.8%	81.2%
Support Vectors	221	198	179	663

Table 5.1. The trained SVM classifiers' performance

appearance variations, the SVM component classifiers require much less support vectors and achieve higher classification rate, as shown in Table 5.1.

[1]we are by no means to train the state-of-the-art SVM classifiers, since our main focus is on the collaborative SVT formulation and solution instead of SVM itself.

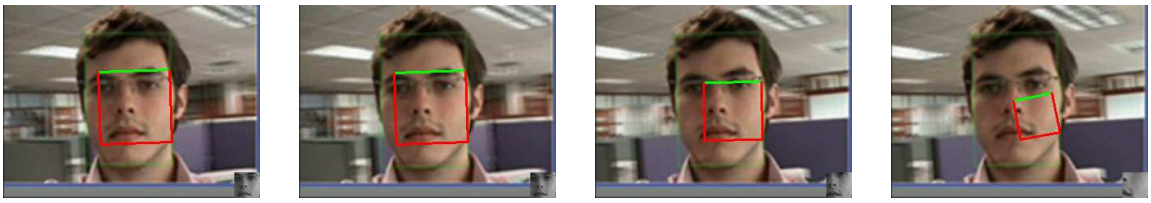
Although our algorithm requires to simultaneously run three SVT trackers with some extra computational efforts on the "message" term calculation, considering the reduced computational cost on the kernel evaluations on the support vectors, we still achieve around 8 frames per second with the non-optimized C++ codes, while the single SVT tracker also runs no more than 12 frames per second in our implementation. In this work, we did not rely on some cyclic support vector selection procedure, as proposed by [3], to further reduce the computational cost, since those steps are essentially heuristic-based algorithm speedup.

Meanwhile, to avoid the problem of getting trapped into a local optimal solution due to the large object movement, which violates the object small motion assumption, a multi-scale searching mechanism is taken to achieve a coarse-to-fine search. The multi-scale algorithm starts the optimization search at the coarsest scale. Once the coarser scale iterations converge, the motion estimate obtained at this scale will be mapped as an initial value of the motion estimate at the next finer scale. Then the procedure continues until the finest scale estimation is arrived. In our experiments, three-scale Gaussian pyramid images are constructed from the input video, and then the collaborative SVT is carried out for all three scales, in a coarse to fine manner.

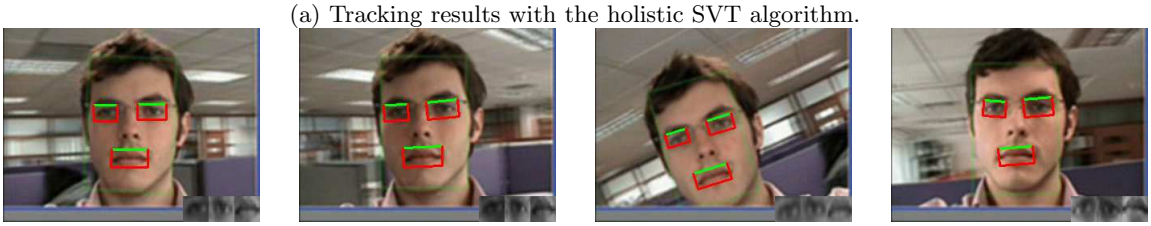
We implemented the trackers to be able to recover the object state up to similarity motion transformation. Please note that our combination of SVM and gradient-based tracking does not complicate the SVM learning. On the contrary, as well known, embedding transformation invariance into SVM classifier itself requires many efforts on training. Therefore, our collaborative SVT greatly reduces the computations on the classifier learning phase. In addition, it also simplifies the object detection process. Instead of

exhaustively looking for all image locations, the object location estimate is speeded up by taking the prediction from the previous frame tracking result.

5.5.1. Tracking Under Illumination Changes



(a) Tracking results with the holistic SVT algorithm.



(b) Tracking results with the proposed collaborative SVT algorithm.

Figure 5.2. Tracking rotating face under illumination changes. Please see the attached video for details.

The first experiment shows a challenging sequence with dramatic illumination changes induced by the continuous camera movement[2]. Some components of the person's face are highly affected by the shadows caused by the changing of the illuminating source direction, especially the areas of two eyes. This uneven illumination situation fails the original holistic-based SVT tracking with the whole face representation. It starts to drift away just after the shadow appears around the left eye area at the 100th frame. Some sample frames of this single SVT tracking are shown in Figure 5.2(a). Please note that we highlight the top border of the object rectangle region by a green bar to explicitly show the object orientation.

[2]The authors acknowledge Mr. Oliver Williams [131] for providing the video data on his website.

On the contrary, the proposed collaborative SVT works fairly well for this case by enjoying the advantages of the introduced geometric structure constraint among different face components. Several samples of the tracking results are shown in Figure 5.2(b). During the illumination change period, because the effect places less influence on the mouth area, it then plays an important role on helping maintain the correct locations of other two components through the "message" propagation.

5.5.2. Tracking Under Large Appearance Variations

Facial region may experience very different appearance patterns, when the person is behaving different facial expressions. Learning all those variations with a holistic-based SVM classifier is challenging. By the divide-and-conquer strategy, our component-based collaborative SVT is able to tolerate such large facial expression changes.

One of the examples showing the efficacy of our approach to dealing with expression variations is demonstrated in Figure 5.3. As illustrated in the Figure, the algorithm is capable of keeping tracking the object, even when the person is showing local face deformations induced by different facial expressions, and large global movement at the same time. Both the open and close patterns of the mouth and eyes are captured by the set of collaborative SVTs. Please note that the imposed geometric structure constraint is not a hard one. It is able to tolerate the local deformations, characterized by the relative distances of different components, to some extent. In this work, we obtained the geometric structure model from the initialization step of the first frame. However, since the proposed approach is a general framework, we may also take a learning step to build the geometric structure model, i.e., Learning the geometric structure constraint

from training data, as the pictorial structure did in [41], then linearizing the function as explained in Eq. 5.12 to obtain the model.

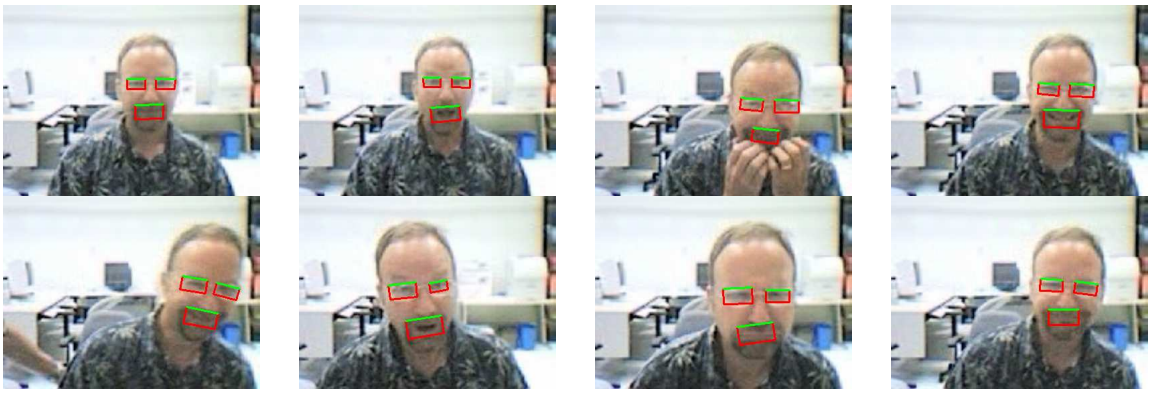


Figure 5.3. Tracking a face with the large expression change and appearance variations by the proposed Collaborative SVT algorithm. Please see the attached video for details.

Figure 5.4 and Figure 5.5 give another two examples of the facial region tracking, where the objects are showing large expression changes. Although our component SVM models are trained only from the frontal face data. The decomposition of the whole facial area into several components enables the algorithm to continuously track the faces even when they demonstrate out-of-plane rotations, as shown in Figures. Note that the state-of-the-art face detectors, such as [128] is usually unable to detect those out-of-plane rotating faces, unless the detector is also trained from profile faces. In comparison with the brute-force detections over every translation and rotation, our collaborative SVT also enjoys the advantages of simplified SVM classifiers training and the computationally efficient gradient search.

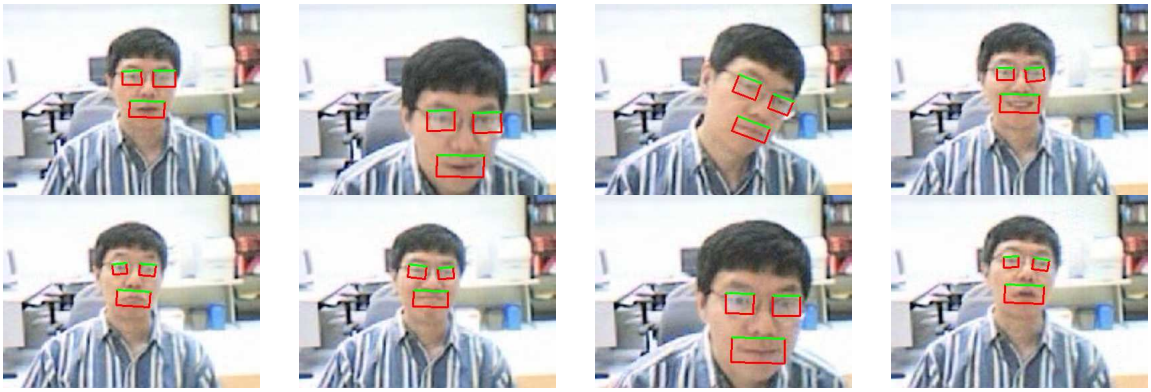


Figure 5.4. Tracking a face with the large expression change and appearance variations by the proposed Collaborative SVT algorithm. Please see the attached video for details.

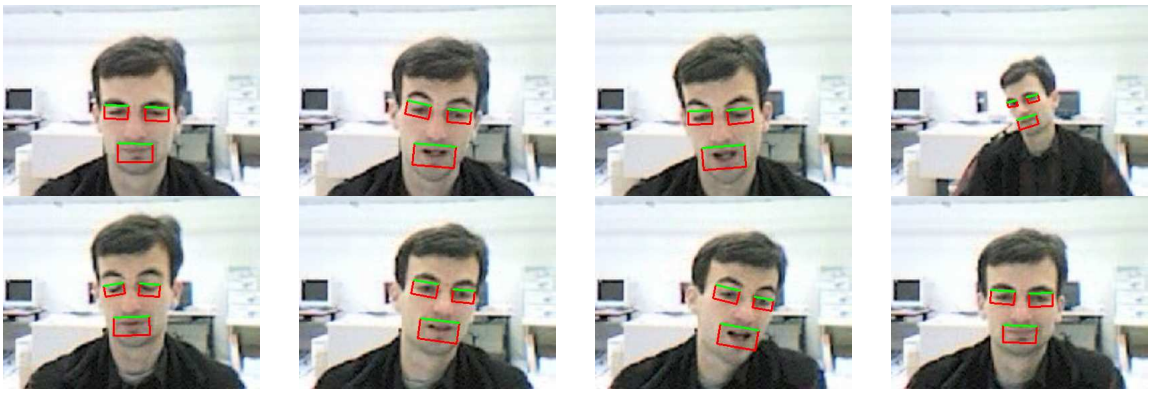


Figure 5.5. Tracking a face with the large expression change and appearance variations by the proposed Collaborative SVT algorithm. Please see the attached video for details.

5.5.3. Tracking Under Partial Occlusions

We demonstrate the efficacy of the occlusion invariant collaborative SVT algorithm proposed in section 5.4 by the following face tracking sequences, where serious occlusions happen frequently, making some parts of facial regions invisible, and thus challenging any holistic-based appearance trackers.

Due to the incapability of handling occlusions, single SVT algorithm fails at the early stages of these sequences. Our approach, because of being equipped with the well designed "message sifting" mechanism as discussed in section 5.4, successfully tracks all the partially occluded faces.

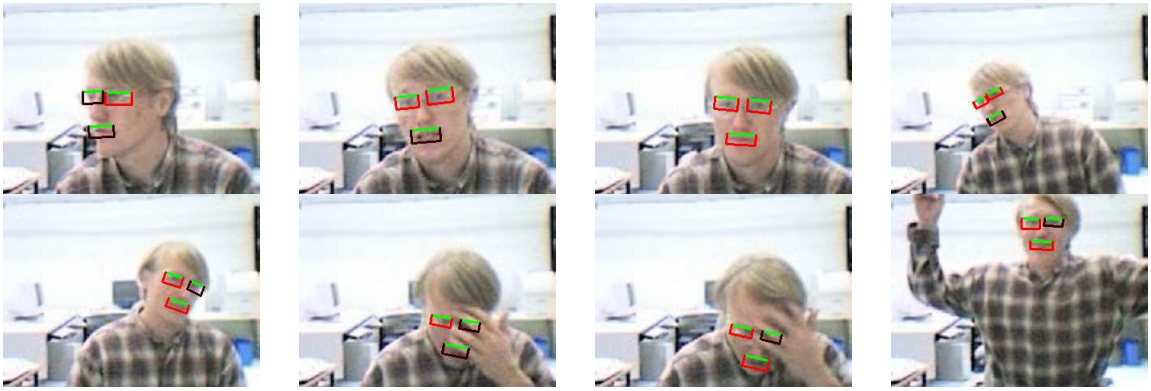


Figure 5.6. Tracking partial occluded face with the proposed occlusion invariant CSVT algorithm. Weakly red-colored components illustrate the negative SVM score response. Please see the attached video for details.

We show the sample frames for one of these testing sequences in Figure 5.6. During some challenging periods, two of the three components (left eye and mouth) are both occluded by a hand, where the SVM scores of these components decrease to negative values and the corresponding tracking boxes become weakly red-colored. The only available and reliable information source (the right eye) became critical to maintain a correct tracking of the whole face. The plotted SVM scores of the three components are shown in Figure 5.7, where the left figure illustrates the component SVM scores, and the right one demonstrates the overall SVM score. Please note that when computing the overall SVM score, the negative SVM scores are truncated out due to the explicit max term as in Eq. 5.17. As can be seen from the figure, though the component SVMs may drop to negative, the overall SVM responses from all parts remain positive, implying that at least one of

the component SVTs are still functioning well, which thus guarantees the continuously successful tracking of the whole facial area. Therefore, when doing iterative updating as in Eq. 5.20, the components with negative SVM response will begin to gradually trust the propagated information from those reliable components.

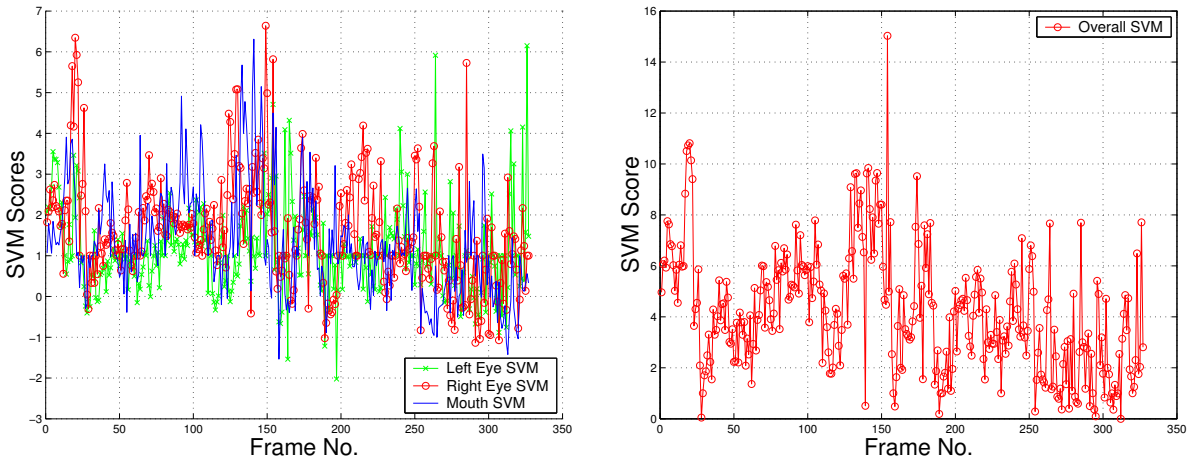


Figure 5.7. The SVM scores of three face components measured from the second sequence. Left figure shows the component SVM scores, and right figure represents the overall SVM score.

Another facial tracking results are shown in Figure 5.8, where when the person is wearing his glasses, both eyes are occluded. Thanks to the selective information propagation mechanism, the mouth successfully maintains the rough locations of both eyes until they are reinforced by their own positive SVM responses later.

Figure 5.8. Tracking partial occluded face with the proposed occlusion invariant CSVT algorithm. Weakly red-colored components illustrate the negative SVM score response. Please see the attached video for details.

5.6. Discussions

In this chapter, we propose a collaborative SVT tracking framework, where the objective function includes the SVM term from each component with an additional term to explicitly model the geometric constraints among the components. The optimization of this objective function reveals a computationally efficient iterative algorithm, which mathematically decomposes the estimation of the joint multiple object component motion states into a set collaborative SVT solvers. In addition, we further introduce an occlusion invariant model, where when occlusion happens can achieve automatic favoring selection to pay more attention on the trustworthy object components while down-weighting the unreliable components.

One of the promising future directions is that instead of manually choosing the components, can we find a principled criterion to automatically select the set of discriminative components? We are also investigating the direct extension of the current framework to multiple object tracking scenario.

CHAPTER 6

Differential Tracking based on Spatial-Appearance Model (SAM)

6.1. Introduction

One of the major challenges of appearance-based tracking lies in the large variations of the visual appearances, which may be caused by many reasons, such as non-rigid deformations, and partial occlusions, etc. Such large uncertainties in the visual appearance significantly complicate the matching of the visual appearances. Inappropriate matching results in the inability of motion recovery and tracking failure.

Existing solutions to appearance-based tracking have different treatment and exploitation of the spatial structure of the appearance. Two opposite extremes are template matching that requires a fine localized match [52, 68, 4, 54], and histogram matching that completely discards the spatial structure [25, 20, 136, 53].

Template tracking with SSD measure requires strict pixel-wise alignments between the object template and the candidate object region [52]. This is fine to handle rigid objects, while having a very limited power to handle non-rigid objects. To allow more appearance variations, improvements have been made by generalizing the template to be a template manifold, which can be linearly expanded by a set of eigenvectors [10], or support vectors [4]. Such a template manifold has to be learned off-line.

Histogram, on the other hand, completely discards the spatial information, thus allows dramatic appearance changes. Histogram-based tracking methods have demonstrated

their superb performance in handling the non-rigid deformation, pose change and partial occlusions [25, 136]. However, the ignorance of the spatial layout also brings difficulties, e.g., less discriminative to appearance changes and thus less sensitive to certain motions. For example, the mean-shift tracker is awkward to handle scaling and rotation. Improvements have been made by using multiple kernels [53, 39, 40].

This chapter presents a novel differential approach based on a spatial-appearance model (SAM) that combines local appearances variations and global spatial structures, thus integrating the advantages of both. SAM is in the form of a Gaussian mixture model. This model can capture a large variety of appearance variations that are attributed to the local non-rigidity. At the same time, this model enables efficient recovery of all motion parameters. A maximum-likelihood estimation is defined for tracking, and is solved by a proposed variant of Expectation-Maximization (EM) algorithm. The analytical derivations lead to a closed-form solution for motion estimation. The proposed EM iterations guarantee the continuous increase of the likelihood, and result in a differential approach to motion recovery. The physical meaning of our solution indicates that the exact pixel-wise alignment is relaxed and the pixels in the candidate object region are weighted by their nearby spatial-appearance Gaussian components in motion estimation.

Besides the ability of handling the appearance variations of non-rigid objects, another advantage of the proposed method is its ability of estimating various motions (e.g., translation, rotation, scaling, and affine) in a unified and principled manner, rather than having different mechanisms to handle them individually. It is actually a very appealing property comparing with mean-shift that only copes with translation in a principal manner. The new method proves very powerful to handle non-rigid objects.

The proposed method is different from some recent approaches that also make use of spatial and appearance models. For example, a model based on the pixel spatial-color features is proposed and is constructed by kernel density estimation [37]. An entropy-based similarity measure between two kernel densities is used for matching. A recent study [159] showed that this approach might not be suitable for the handling of complex motions and the entropy-based similarity measure is difficult to compute. Our approach differs greatly in the matching criteria, the analysis and thus the solutions.

6.2. Spatial-Appearance Model (SAM)

Recall the two extremes of appearance modelling vary from the approaches that strictly obey the spatial layout of object appearance (rigid template representation) to the ones where spatial locations of appearance features are completely discarded (histogram-based representation). Both of the modelling approaches have their merits and limitations. We choose to seek a tradeoff between the two approaches, and arrive at an intermediate level appearance modelling, which not only maintains a rough global spatial structure of object appearance as in template representation, but also preserves the simplicity of the histogram-based representation by only keeping some dominant feature values in the object region.

Given an initial object region $R_0 = \{x_i, i = 1, \ldots, N\}$, selected manually or automatically, a d dimensional spatial-appearance feature vector is extracted from each pixel and denoted by x_i. N is the total number of pixels within the initial object region. A K-component Gaussian mixture model (GMM) is adopted to fit to the collected data points, leading to a spatial-appearance model characterized by GMM with parameters

$\theta = (p_k, \mu_k, \Sigma_k), k = 1, \ldots, K$. p_k, μ_k, Σ_k represent the prior probability, mean and variance of Gaussian component k in the mixture model. Each Gaussian component is denoted by $g(x; \mu_k, \Sigma_k)$. The likelihood of a pixel x within a candidate object region is simply the mixture probability as:

$$p(x|\theta) = \sum_{k=1}^{K} p_k g(x; \mu_k, \Sigma_k) \tag{6.1}$$

Depending on different features, the model dimension d could take different values with the first two dimensions occupied by the pixel spatial coordinate features (u, v). For example, we may take $d = 3$ by augmenting the spatial features with the intensity feature, or when color features are preferred, we may add dimensions with pixel feature values from (r, g, b) color channels.

Similar to the de-correlation strategy of spatial-appearance features as in [37, 146], we assume the spatial and appearance dimensions of the GMM model are decoupled, i.e., the covariance matrix of the Gaussian component takes the block diagonal form, $\Sigma_k = \begin{pmatrix} \Sigma_{k,s} & 0 \\ 0 & \Sigma_{k,c} \end{pmatrix}$, where s and c stand for spatial and appearance features respectively. Thus the joint feature x of each pixel can be written as $x = (x_s, c(x_s))$, with the spatial x_s and the appearance $c(x_s)$ features of a pixel at the location x_s. Each GMM Gaussian component then has the following factorized form:

$$g(x; \mu_k, \Sigma_k) = g(x_s; \mu_{k,s}, \Sigma_{k,s}) g(c(x_s); \mu_{k,c}, \Sigma_{k,c}) \tag{6.2}$$

The appearance feature $c(x_s)$ is actually the function of pixel location x_s, implying the intrinsic correlations between the spatial and appearance features although the decoupled Gaussian distribution.

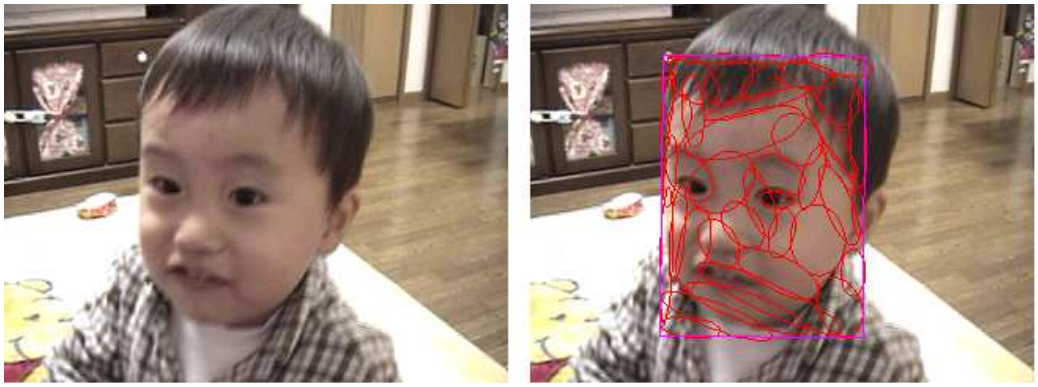


Figure 6.1. The fitted spatial-appearance Gaussian mixture model to the object region.

An illustrative example of fitting the spatial-appearance model to the object region (a kid face) with 40-component mixture model is shown in Figure 6.1, where the left image is the original video frame, and in the right image each red ellipse represents a spatial Gaussian component fitted using Expectation-Maximization (EM) [33].

6.3. Expectation-Maximization (EM) Tracking

6.3.1. Maximum-Likelihood Formulation

Assume the object undergoes a motion transform characterized by a general motion model $T(x_s; a_t)$. a_t is the transform parameter at time t that warps a pixel at location x_s from reference frame to the location $T(x_s; a_t)$ in the current frame. Without losing the generality, we can assume the considered motion model having a general linear form as follows:

$$T(x_s; a_t) = \begin{pmatrix} a_{1,t} & a_{2,t} \\ a_{3,t} & a_{4,t} \end{pmatrix} x_s + \begin{pmatrix} a_{5,t} \\ a_{6,t} \end{pmatrix} \tag{6.3}$$

$$= A_t x_s + B_t$$

which can actually cover a broad spectrum of object motions, such as translation, scaling, rotation, and affine motion, etc.

With the SAM object model initialized in the reference frame, the likelihood of an object pixel x_i, warped from the reference frame to the current frame by motion transform $T(x_s; a_t)$, is evaluated as:

$$\begin{aligned} & p(T(x_i; a_t)|\theta) \\ & = p(T(x_{i,s}; a_t), c(T(x_{i,s}; a_t))|\theta) \\ & = \sum_{k=1}^{K} p_k g(T(x_{i,s}; a_t); T(\mu_{k,s}, \Sigma_{k,s}; a_t)) \\ & \qquad \times g(c(T(x_{i,s}; a_t)); \mu_{k,c}, \Sigma_{k,c}) \\ & = \sum_{k=1}^{K} p_k g(x_{i,s}; \mu_{k,s}, \Sigma_{k,s}) g(c(T(x_{i,s}; a_t)); \mu_{k,c}, \Sigma_{k,c}) \end{aligned} \tag{6.4}$$

Note from Eq. 6.4 that not only the pixel spatial coordinate $x_{i,s}$ is transformed to $T(x_{i,s}; a_t)$, but also the Gaussian parameter values on the spatial dimension are changed from $(\mu_{k,s}, \Sigma_{k,s})$ to $T(\mu_{k,s}, \Sigma_{k,s}; a_t)$. With the general linear motion model defined in Eq. 6.3, such that $T(x_{i,s}; a_t) = A_t x_{i,s} + B_t$, and $T(\mu_{k,s}, \Sigma_{k,s}; a_t) = (A_t \mu_{k,s} + B_t, A_t \Sigma_{k,s} A_t^\tau)$, this generally leads to a cancelled out effect on the Gaussian function evaluations on the spatial GMM components. However, since the appearance features of each pixel are coupled with the transformed position of the pixel in the current frame, i.e., $c(T(x_{i,s}; a_t))$, it essentially correlates the pixel likelihood evaluation with the unknown object motion estimation a_t.

To ease the derivations, define the data component probability $q(k, x_i; a_t)$ as

$$q(k, x_i; a_t) = p_k g(x_{i,s}; \mu_{k,s}, \Sigma_{k,s}) g(c(T(x_{i,s}; a_t)); \mu_{k,c}, \Sigma_{k,c}) \tag{6.5}$$

We propose a matching criterion to recover the object motion a_t based on the integration of the pixel data logarithm likelihood over the object region.

$$\begin{aligned} E(a_t; \theta) &= \sum_{x_i \in R_0} \log p(T(x_i; a_t)|\theta) \\ &= \sum_{x_i \in R_0} \log\{\sum_{k=1}^{K} q(k, x_i; a_t)\} \end{aligned} \tag{6.6}$$

This joint data likelihood term measures the data fitness of a candidate object region R_t at current time t, warped from the reference object region R_0, to the object SAM model characterized by model parameter θ. Thus the problem of object tracking becomes an essential optimization problem, where the objective is to look for an optimal value a_t^* that maximizes the joint likelihood energy function $E(a_t; \theta)$, i.e.,

$$a_t^* = \max_{a_t} E(a_t; \theta) \tag{6.7}$$

6.3.2. Closed-Form Tracking with EM

Treating the motion transform parameter a_t as the only unknown value in the above maximum likelihood estimation of Eq. 6.6, the Expectation-Maximization (EM) algorithm is well suitable to be adopted here to recover the unknown value of a_t for the current frame, with simultaneous achievement of energy function maximization.

Unlike the general EM algorithm for the parameter fitting of GMM model, where the objective is to find an optimal model parameter set θ^* that best explains the training data set. Here we assume that the GMM model parameter θ remains unchanged during this optimization process, while only deriving a solution to incrementally update the motion parameter a_t embedded into the EM iterations. We should clarify that our assumption that the GMM model parameter stays constant during this one frame EM iteration is a quite valid assumption. It actually has been intrinsically utilized by most existing tracking approaches, where the object model, once firstly initialized, will generally remain fixed during the whole tracking sequence, unless some online updating mechanism is adopted in order to handle the non-stationary visual process [68, 54, 147].

In fact, it is interesting to point out that our mixture framework does allow a straightforward incorporation of an online updating process to handle the problem of tracking non-stationary object appearance. Although the current version of the algorithm does not take such a further step, we leave this issue for the future improvements. All the experiments reported in this chapter do not take an online updating step, while still achieving very encouraging tracking results.

Similar to the general EM algorithm, an initial value for the unknown parameter must be specified in order to start the EM iterations. In our case we simply take the recovered motion estimation a^*_{t-1} from previous frame as the initialization of a_t, i.e., $a_t^{(0)} = a^*_{t-1}$. The superscript indexes the EM algorithm iteration. Assume that we have already obtained an estimation of a_t during the j_{th} EM iteration, i.e., $a_t^{(j)}$, the E-step involves the computation

of pixel assignment probability to each Gaussian component as

$$p^{(j)}(k|x_i;a_t^{(j)}) = \frac{q(k,x_i;a_t^{(j)})}{\sum_{m=1}^{K} q(m,x_i;a_t^{(j)})} \tag{6.8}$$

with the data component probability $q(k,x_i;a_t^{(j)})$ defined in Eq. 6.5.

From the Jensen's inequality, we have the following lower bound to the original energy function $E(a_t;\theta)$:

$$\begin{aligned}
&E(a_t;\theta) \\
&= \sum_{x_i \in R_0} \log\{\sum_{k=1}^{K} q(k,x_i;a_t)\} \\
&= \sum_{x_i \in R_0} \log\{\sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \frac{q(k,x_i;a_t)}{p^{(j)}(k|x_i;a_t^{(j)})}\} \\
&\geq \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \log \frac{q(k,x_i;a_t)}{p^{(j)}(k|x_i;a_t^{(j)})} \\
&= \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \log q(k,x_i;a_t) - \\
&\sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \log p^{(j)}(k|x_i;a_t^{(j)}) \\
&= E^{(j)}(a_t;\theta)
\end{aligned} \tag{6.9}$$

Maximizing $E(a_t;\theta)$ can be achieved by maximizing the lower bound function $E^{(j)}(a_t;\theta)$, and subsequently maximizing the first term of $E^{(j)}(a_t;\theta)$ in Eq. 6.9, since the old pixel assignment probabilities $p^{(j)}(k|x_i;a_t^{(j)})$ are known provided the value $a_t^{(j)}$ at j_{th} EM iteration, thus the second term is unrelated to the objective function maximization over a_t.

we define the first term of lower bound function by $\widetilde{E}^{(j)}(a_t;\theta)$,

$$\widetilde{E}^{(j)}(a_t;\theta) = \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \log q(k,x_i;a_t) \tag{6.10}$$

Iteratively maximizing $\widetilde{E}^{(j)}(a_t;\theta)$ by finding an updated estimation $a_t^{(j+1)}$ has the same effect on the incremental maximization of the original objective function $E(a_t;\theta)$. Rather than the logarithm of a sum as in $E(a_t;\theta)$, the derived $\widetilde{E}^{(j)}(a_t;\theta)$ only contains a linear combination of K logarithms, which breaks the coupling of the equations when setting the derivatives of $\widetilde{E}^{(j)}(a_t;\theta)$ over the parameter a_t to zero.

We take an incremental updating form by assuming that $a_t^{(j+1)} = a_t^{(j)} + \Delta a_t$, then the above maximization can be written as

$$\begin{aligned} &\max_{\Delta a_t} \widetilde{E}^{(j)}(\Delta a_t;\theta) \\ &= \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i;a_t^{(j)}) \log q(k,x_i;a_t^{(j)} + \Delta a_t) \end{aligned} \tag{6.11}$$

Taking the partial derivative of $\widetilde{E}^{(j)}(\Delta a_t;\theta)$ over Δa_t and setting it to zero, we can obtain a series of linear updating equations to incrementally maximize the objective function depending on what motion model is used.

To ease the exposition, we firstly show the updating equations for the simple case, where a translational motion model is adopted, and object appearance feature is simply the pixel intensity. Then we generalize the discussions to handle more complex motion model, including scaling, rotation, or affine transform, and multi-dimension appearance features such as pixel values in (r,g,b) color channel are also considered there.

Recall that the motion parameter $a_t = \{A_t, B_t\}$ as defined in Eq. 6.3, when translational motion model is taken, A_t becomes an Identity matrix, we only need to consider the second term, i.e., $a_t = B_t$. By taking the incremental updating form, we have $\Delta a_t = \Delta B_t$.

$$\begin{aligned} &\widetilde{E}^{(j)}(\Delta B_t; \theta) \\ &= \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) \log q(k, x_i; B_t^{(j)} + \Delta B_t) \end{aligned} \tag{6.12}$$

Taking the partial derivative over ΔB_t

$$\begin{aligned} &\frac{\partial \widetilde{E}^{(j)}(\Delta B_t; \theta)}{\partial \Delta B_t} \\ &= \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) \frac{\partial \log q(k, x_i; B_t^{(j)} + \Delta B_t)}{\partial \Delta B_t} \\ &= \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) \\ &\qquad \times \frac{\partial \log g(c(x_{i,s} + B_t^{(j)} + \Delta B_t); \mu_{k,c}, \Sigma_{k,c})}{\partial \Delta B_t} \end{aligned} \tag{6.13}$$

where the spatial component probability $g(x_{i,s}; \mu_{k,s}, \Sigma_{k,s})$ and component priori p_k in $q(k, x_i; B_t^{(j)} + \Delta B_t)$ disappear due to their uncorrelation with motion update ΔB_t. However, please note that their effects on the motion estimation do reflect on the computation of pixel assignment probability $p^{(j)}(k|x_i; B_t^{(j)})$.

Following the small motion assumption, $c(x_{i,s} + B_t^{(j)} + \Delta B_t)$ can be linearized by taking the first order Taylor expansion as

$$c(x_{i,s} + B_t^{(j)} + \Delta B_t) = c(x_{i,s} + B_t^{(j)}) + H_{i,t}^{(j)^\tau} \Delta B_t \tag{6.14}$$

where $H_{i,t}^{(j)}$ is the Jacobian matrix of the appearance feature over motion estimation evaluated at its current value $B_t^{(j)}$. When the appearance feature is simply the pixel intensity, $H_{i,t}^{(j)}$ takes the form as

$$H_{i,t}^{(j)} = \begin{pmatrix} c_u(x_{i,s} + B_t^{(j)}) \\ c_v(x_{i,s} + B_t^{(j)}) \end{pmatrix} \tag{6.15}$$

where $(c_u(x_{i,s} + B_t^{(j)}), c_v(x_{i,s} + B_t^{(j)}))$ are the horizontal and vertical intensity gradients at location $x_{i,s} + B_t^{(j)}$ of the current frame.

Recall that the probability distribution in appearance dimension $g(c(x_{i,s} + B_t^{(j)} + \Delta B_t); \mu_{k,c}, \Sigma_{k,c})$ also takes the Gaussian form, in combination with the linearized form of $c(x_{i,s} + B_t^{(j)} + \Delta B_t)$ in Eq. 6.14, the partial derivative of the objective function $\widetilde{E}^{(j)}(\Delta B_t; \theta)$ over ΔB_t can be eventually reached as

$$\begin{aligned} & \frac{\partial \widetilde{E}^{(j)}(\Delta B_t; \theta)}{\partial \Delta B_t} \\ = & \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) \\ & \qquad \times H_{i,t}^{(j)} \Sigma_{k,c}^{-1} [(c(x_{i,s} + B_t^{(j)}) - \mu_{k,c}) + H_{i,t}^{(j)^\tau} \Delta B_t] \\ = & 0 \end{aligned} \tag{6.16}$$

i.e., the following linear system equation can be derived to solve ΔB_t

$$\begin{aligned} U \Delta B_t &= V \\ \Delta B_t &= U^{-1} V \end{aligned} \tag{6.17}$$

where matrix U and V are defined as follows

$$U = \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) H_{i,t}^{(j)} \Sigma_{k,c}^{-1} {H_{i,t}^{(j)}}^{\tau} \tag{6.18}$$

$$V = - \sum_{x_i \in R_0} \sum_{k=1}^{K} p^{(j)}(k|x_i; B_t^{(j)}) H_{i,t}^{(j)} \Sigma_{k,c}^{-1} (c(x_{i,s} + B_t^{(j)}) - \mu_{k,c}) \tag{6.19}$$

The form of linear system equation implies that the contribution of each pixel to motion estimation is weighted by its nearby spatial-appearance Gaussian components, through assignment probability $p^{(j)}(k|x_i; B_t^{(j)})$, and appearance mean $\mu_{k,c}$ and variance $\Sigma_{k,c}$. Thus exact pixel-wise alignment between initial object region R_0 and warped candidate R_t is relaxed, leading to a more flexible framework of tolerating large appearance deformation during tracking. The extent of deformation tolerance is governed by the variance coverage of each mixture component. The contributions of all pixels to motion estimation are combined and voted for the optimal solution of motion update.

With the estimated motion update ΔB_t solved from Eq. 6.17, a new circle of EM iteration starts with the updated estimation of the motion parameter $B_t^{(j+1)}$ as

$$B_t^{(j+1)} = B_t^{(j)} + \Delta B_t \tag{6.20}$$

In summary, the proposed EM tracking approach takes the following two-step iterative procedure.

E-Step: compute the mixture component assignment probability for each pixel x_i by Eq. 6.8.

M-Step: obtain a motion update estimation Δa_t by solving the linear system equation as in Eq. 6.17.

The above EM iterations are iteratively computed to increase the joint data likelihood until convergence.

6.3.3. Tracking under General Motion Transform

The proposed EM tracking procedure could be easily generalized to handle more complex motion model, and incorporate more informative appearance features, while the same M-Step updating equation as in Eq. 6.17 could still be derived. The only difference between these variations lies on the computations of Jacobian matrix of the appearance feature over motion estimation, i.e., $H_{i,t}^{(j)}$. For example, for similarity motion model, handling translation, scaling, and rotation, the 4-dimensional motion vector $a_t = (a_{1,t}, a_{2,t}, a_{3,t}, a_{4,t})^\tau$ has the following form:

$$A_t = \begin{pmatrix} a_{1,t} & -a_{2,t} \\ a_{2,t} & a_{1,t} \end{pmatrix}, B_t = \begin{pmatrix} a_{3,t} \\ a_{4,t} \end{pmatrix} \tag{6.21}$$

the corresponding Jacobian matrix $H_{i,t}^{(j)}$ with intensity feature is defined as

$$H_{i,t}^{(j)} = \begin{pmatrix} c_u(A_t^{(j)} x_{i,s} + B_t^{(j)}) u_{i,s} + c_v(A_t^{(j)} x_{i,s} + B_t^{(j)}) v_{i,s} \\ -c_u(A_t^{(j)} x_{i,s} + B_t^{(j)}) v_{i,s} + c_v(A_t^{(j)} x_{i,s} + B_t^{(j)}) u_{i,s} \\ c_u(A_t^{(j)} x_{i,s} + B_t^{(j)}) \\ c_v(A_t^{(j)} x_{i,s} + B_t^{(j)}) \end{pmatrix} \tag{6.22}$$

When color appearance features are used, the Jacobian matrix $H_{i,t}^{(j)}$ becomes multi-columns with each column having the same form as in Eq. 6.22 but in a different color channel. More complex motion model, such as affine transform, can be derived in a similar manner, therefore we omit the discussion here.

It is clear that our framework allows tracking object under any general linear motion transforms that are solved in a unified way. The more complex motion recovery puts no more computation overhead than the simple ones. It is actually a very appealing property in comparison with the kernel-based tracking approaches using mean-shift, where only object translation is principally handled.

As guaranteed by the Jensen's inequality in Eq. 6.9, the lower bound optimization in the proposed EM iterations will subsequently lead to a continuous maximization of the original objective function, i.e., the data likelihood, thus driving the motion estimation towards the optimal candidate object region. Compared with the Kernel-based tracking [25], where a line search procedure is usually required for the optimal step length decision of mean shift iteration, our closed-form linear solution derived in Eq. 6.17 enjoys a Newton-style iteration as in template matching [52] and Kernel-based tracking with SSD [53]. It reaches a local optimum in an one-step jump, thus avoiding the tedious process of line search.

Figure 6.2 shows an illustrative example of one-frame EM iterations. The left figure represents the iterative motion estimations, illustrated by a series of colorized quadrangles overlapping on the original frame, with pure red to pure yellow depicting this sequential iteration procedure. The right figure clearly demonstrates the continuous increase of the data logarithm likelihood, as provably guaranteed in Eq. 6.9. In this example, 10

EM iterations are performed to reach the algorithm convergence, where we declare a convergence when there is no significant change between the motion estimations in two consecutive iterations.

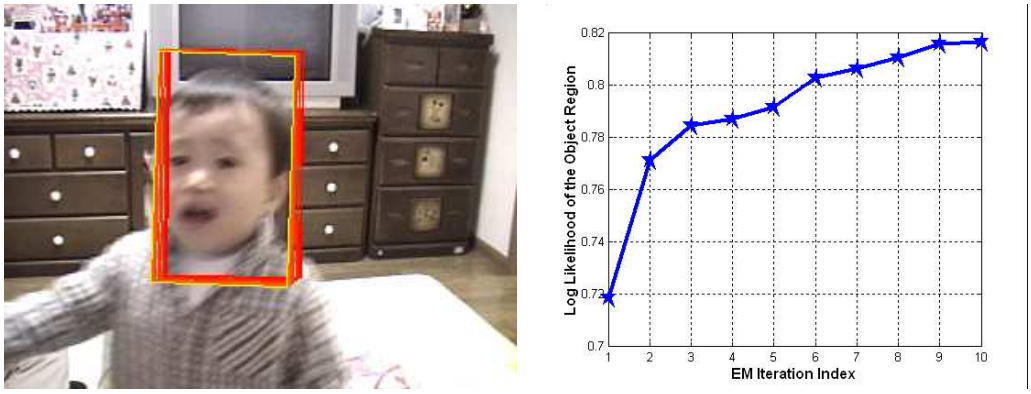


Figure 6.2. Logarithm likelihood evaluation of the candidate object region during one frame iterations.

6.4. Experiments

In this section, we present extensive experiments tested under challenging real-world sequences. A differential tracker based on the proposed approach is implemented, capable of handling object translation, scaling, and rotation. Comparisons are made with simple template tracker and Kernel-based tracker, demonstrating the very encouraging performance of our unified approach for tracking non-rigid objects under dramatic appearance deformations, large object scale changes and partial occlusions.[1]

Depending on the availability of color channels from the input video, the appearance features in the SAM model vary from intensity feature to color features in the RGB color space. The number of mixture components used to model objects may take different values depending on the relative size of objects. It is actually a trade-off factor to govern

[1]Please see the supplemental video for the detailed tracking results.

the model flexibility to appearance deformation, where more components imply more localized component coverage, thus more strict observance of rigid structure assumption, while less components allow more relaxed alignment between the candidate region and object model. Our experience shows that 20-40 components usually work well for a broad spectrum of non-rigid objects we are testing on. We leave the investigation on optimal number of components selection for future study. Some related work along this direction includes [2, 54]. To speed up the model initialization, we take the tracked object regions in the first 50 frames of each sequence to update the mixture model, with one frame one EM iteration to obtain the model. After that, the model is fixed without further updating, and used for tracking the rest of video frames. The current unoptimized C++ implementation of the algorithm runs comfortably around 5-10 fps on average on Pentium 3G.

6.4.1. Large Appearance Deformation

Figure 6.3 shows the tracking results over a home video sequence, where a kid presents significant expression changes, thus dramatic appearance deformations. Considering the relative large size of object, a 40-component mixture model is adopted here to initialize the differential tracker with similarity transform motion model. The first row gives the result from a template matching tracker. It loses a tight tracking of the kid face at the early stage of the sequence, when the kid starts to behave his exaggerating expression and simultaneously shows the significant head movement. The second row of the Figure 6.3 shows the iterative motion estimations in each frame via the proposed differential tracker, with colorized quadrangles from pure red to pure yellow depicting the series of updating

as before. The thickened boundaries due to multiple iterations clearly reflect the large motion effects, which are not only from translation, but also through rotation and scaling. Albeit the difficulties, the proposed differential tracker successfully keeps localizing the non-rigid face with correct motion estimations until the kid completely turns his head to the right side, thanks to the intrinsic deformation tolerance of the proposed approach.

Figure 6.4 demonstrates our tracking results on the famous but challenging *Dudek* sequence[2], which has been tested over several approaches addressing online tracking adaptation [68,82]. The person in this sequence presents not only large appearance variations by changing pose during movement, but also several short periods of severe occlusions. Without counting on the online adaptation, which is acknowledged hard to find the balance between the model adaptability and resistance to noise [54], our approach still achieves very encouraging results, that the improved robustness to partial occlusions could be attributed to the some extent model tolerance on spatial-appearance misalignments in the SAM model.

6.4.2. Large Scale Change

Figure 6.5 shows a real-world surveillance video to demonstrate our tracker capacity of handling large object scale change[3]. A person enters the scene distantly with a quite small scale. Our tracker is initialized on this small object region, and robustly tracks the person for the remaining 1000 frames. Note the accurate scale estimations of the person during most of the tracking period. Two severe occlusions happen when the person is coming

[2]We acknowledge Dr. El-Maraghi [68] for allowing us to download this sequence from his website for testing.

[3]We acknowledge the source data is provided from the EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

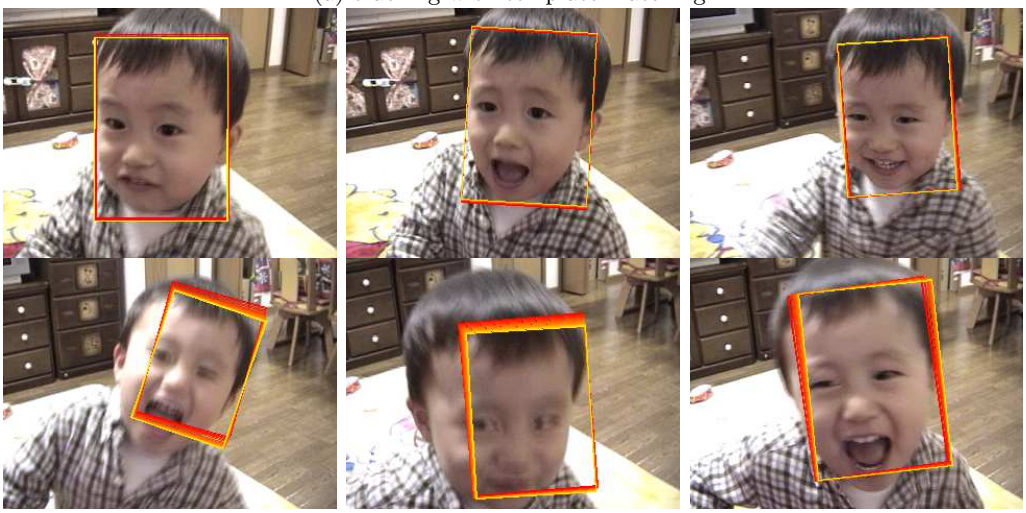
across with other pedestrians, which shortly affects the tracker's scale estimations during occlusion. After the person re-appearance from occlusion, the tracker recovers itself and starts to report the accurate motion estimations again.

The last example in Figure 6.6 also shows a home video filming the same kid as in Figure 6.3. Now the kid demonstrates a dramatic scale change, and also brings the trouble to the tracker by intentionally presenting serious occlusion. Our tracker again robustly tracks the kid face with the correct scale estimations for the whole sequence as shown in Figure 6.6 (b). In comparison, the results obtained from a color-based mean-shift tracker in Figure 6.6 (a) reports an incorrect scale estimation, and consequently loses tracking the object.

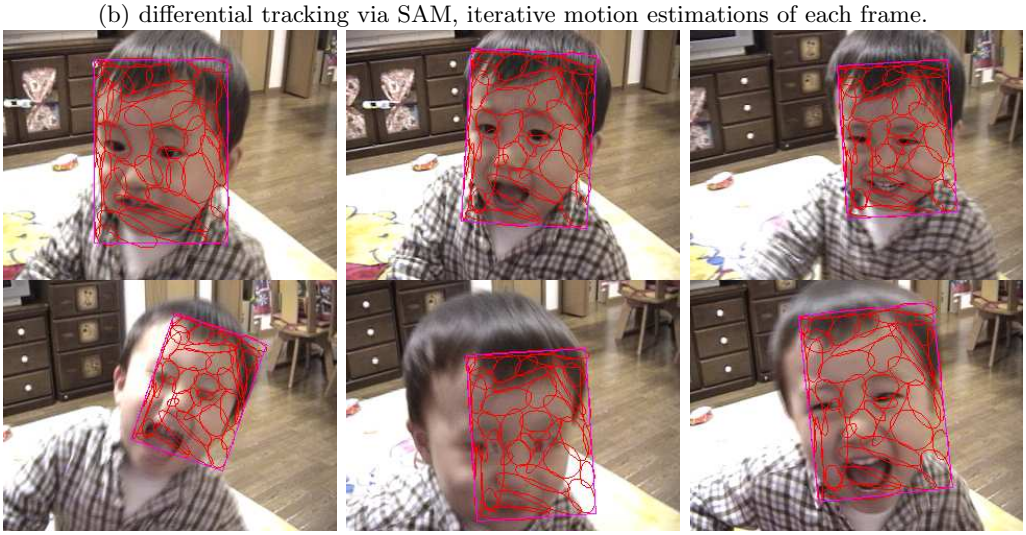
6.5. Discussions

In summary, this chapter presents a novel differential approach for non-rigid object tracking under the general motion transform. A spatial-appearance model (SAM) is introduced to model both the object appearance variations and its global spatial structures. A maximum likelihood matching criterion is defined and rigorous analytical results are obtained through Expectation-Maximization (EM) algorithm, leading to a closed form solution to motion tracking. The derived linear system equation also suggests us to take a new view to look at the connections between the two standard tracking paradigms, template tracking and Kernel-based tracking. Our ongoing research will mainly focus on a deeper investigation on the intrinsic relations of the proposed approach with them, and their more recent advances, such as [53, 39].

(a) tracking with template matching.



(b) differential tracking via SAM, iterative motion estimations of each frame.



(c) differential tracking via SAM, final tracking result of each frame overlapped by spatial mixture components.

Figure 6.3. Tracking a kid face under large appearance deformation. (560 Frames)

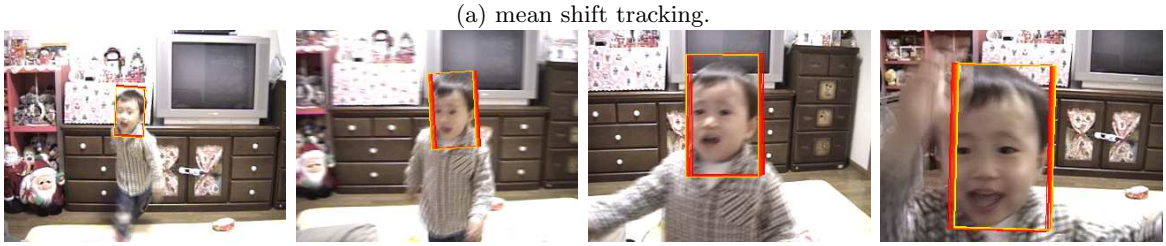
(a) differential tracking via SAM, iterative motion estimations of each frame.

(b) differential tracking via SAM, final tracking result of each frame overlapped by spatial mixture components.

Figure 6.4. Tracking a human face under large scale change and severe occlusions (1145 Frames).

Figure 6.5. Tracking a pedestrian under large scale change and partial occlusions with the proposed differential tracker via SAM. Results overlapped by spatial mixture components. (1230 Frames)

(a) mean shift tracking.



(b) differential tracking via SAM, iterative motion estimations of each frame.

Figure 6.6. Tracking a kid face under large scale change. (690 Frames)

CHAPTER 7

Conclusion and Future Research

With the proliferation of camera sensors deployed world widely, video surveillance systems are gradually finding their way into our daily lives. Security enforcement [55], traffic monitoring [22,99] and daily assistance for elderly [150] are all active applications of the video surveillance systems, to mention a few. A direct consequence of the technological advancements in camera sensor networks is the increased demand for intelligent video analysis and understanding techniques.

This dissertation concentrates on the developments of efficient and effective visual motion analysis techniques that allow automated tracking of multiple targets, which is arguably the most essential problem and component of any state-of-the-art intelligent video surveillance systems.

Due to the unknown number of targets that need to be tracked, and the potential ambiguities introduced by multiple target-tracker associations, a simple solution of instantiating multiple independent trackers is far from enough to solve the problem. Besides sharing the common challenges faced by visual tracking of single target, successful tracking of multiple targets' motions are also confronted by the tremendous difficulties from the theoretical and practical aspects of the problems, such as target appearance variations, target occlusions, high computational demanding, and difficulty of training a target detector.

7.1. Summary

In this dissertation, we present several novel effective and computationally efficient solutions to addressing the above mentioned problems in multiple motion analysis, with the aim of driving the state-of-the-art motion analysis algorithms to fulfill the ever increasing challenges from the real world data. In summary, we have made the following novel contributions to tackle the above challenging problems:

- A novel centralized formulation to tackle the multiple target tracking problem with explicit occlusion handling, where the extra hidden process of occlusion is embedded into a dynamic Bayesian network formulation. The successful inference of this hidden process can reveal the explicit occlusion relations among different targets, which makes the tracker more robust against partial even complete occlusions.
- A novel linear complexity decentralized framework to address the multiple target tracking problem. The basic idea is a distributed while collaborative inference mechanism based on Markov network formulation, where a probabilistic exclusive constraint is added to the targets to allow the set of neighboring targets to compete for the common visual data accounting for the target appearances. Variational inference is employed on the Markov network analysis and reveals an essential parallelization and distributed computing paradigm for multiple target tracking.
- A principled extension of the decentralized framework to enable the tracking of variable number of targets through the entropy-based tracker performance self-evaluator. Discriminative target detector is also bonded into the framework to

enable the construction of effective importance functions to collect informative bottom up image features for more efficient probabilistic inference.

- A novel two-layer statistical field model is proposed to characterize the large shape variability and partial occlusions for nonrigid target detections, especially pedestrian detections. Probabilistic variational analysis reveals a set of fixed point equations that give the equilibrium of the field, leading to computationally efficient methods for calculating the image likelihood and for training the model.
- A component-based appearance tracker based on support vector machines is introduced to accommodate the large object appearance variations, enabling the development of a robust single target tracker. The designed component selective mechanism brings the algorithm the capacity of automatically selecting trustworthy components while down-weighting the unreliable ones, thus making the robust handling of object partial occlusions possible.
- A novel differential tracking approach is developed based on a spatial-appearance model (SAM) that combines local appearances variations and global spatial structures. Rigorous derivation of the model can lead to a closed form solution to motion tracking under any linear motion transformations. The performance evaluation shows that the developed tracker is able to continuously track non-rigid objects that exhibit dramatic appearance deformations, large object scale changes and partial occlusions.

7.2. Potential Future Research Directions

There are a few directions which are interesting to explore in the future research endeavors.

- Single target robust tracking with valid model adaptations. Although our proposed algorithms **??** are able to handle the large non-rigid appearance variations, the other issue remaining to be explored is how to achieve the robust tracking of a target with non-stationary appearances? Due to the inexistence of invariant appearance features for target modelling, model adaptation has to be addressed, i.e., an online model updating mechanism must be explored in order to successfully fitting the model to the continuously changing target appearance. Actually, both our collaborative support vector tracker and differential spatial-appearance tracker do not exclude such a direct extension to accommodate this incremental model updating procedure. Some of the promising results along this direction have been proposed in [59,148,54], but we believe that a more rigorous treatment is expected to justify the validity of model adaptations over time.
- Tracking using camera networks. All the tracking algorithms presented in this dissertation are based on the input from single camera setup, although the proposed approaches are theoretically general enough to be compatible with inputs from multiple camera setups. The visual measurements collected from multiple cameras also enable the possibilities of the 3D target model construction in the tracking formulation. We believe that not only this 3D extension will lead to more accurate target model characterizations, but also the motion competitions introduced in the decentralized multiple tracker formulation will also be better

beneficial from this 3D setting, since practically any two targets can not physically occupy the same locations in 3D space.

- Target re-identification under the non-overlapping camera network. It is very often that the number of cameras set up to cover a large surveillance area may not be enough to achieve an overlapping coverage between any spatial neighboring camera sensors, while the target being tracked may show the extensive movements across these non-overlapping cameras. It implies that we are facing the problem of establishing the correct correspondences of the target tracks even under the existence of target disappearing during some noticeable period of time. Some promising results for this problem are reported in [67, 116, 51], but they are still in some heuristic nature. It is interesting to investigate that whether some more theoretical sound solution can be discovered.
- Towards trajectory-based event detection and mining. When multiple target detection and tracking have successfully extracted the motion trajectories of the targets from large video data set, the next question we are naturally willing to ask is that what semantically interesting contents we can discover for analysis, interpretation, and action. In general, learning-based approaches must be adopted to achieve supervised or unsupervised classifications based on the trajectories data set. We deem this as our long term goals to fulfill the desired properties of the high level semantical descriptions of the visual data for truly "intelligent" video surveillance applications.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans on Pattern Analysis and Machine Intelligence*, pages 1475–1090, Nov. 2004.

[2] O. Arandjelovic and R. Cipolla. Incremental learning of temporally-coherent gaussian mixture models. In *In Proc. British Machine Vision Conference (BMVC)*, 2005.

[3] S. Avidan. Subset selection for efficient svm tracking. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2003.

[4] S. Avidan. Support vector tracking. *IEEE Trans on Pattern Analysis and Machine Intelligence*, pages 1064–1072, Aug. 2004.

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

[6] Yaakov Bar-Shalom and Thmoas Fortmann. *Tracking and Data Association*. Academic Press, Orlando, FL, 1988.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE trans. on PAMI*, 24:509–522, 2002.

[8] Stan Birchfield. Ellitical head tracking using intensity gradient and color histograms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 232–237, Santa Barbara, California, June 1998.

[9] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. European Conf. on Computer Vision*, pages 329–342, 1996.

[10] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *In Int'l Journal of Computer Vision (IJCV)*, 26(1):63–84, 1998.

[11] S. Blackman and R. Popolo. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[12] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, London, 1998.

[13] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. IEEE International Conf. on Computer Vision (ICCV)*, 2005.

[14] A. Bloem. Joint probabilistic data association methods avoiding track coalescence. In *Proc. IEEE Int'l Conf. on Decision and Control*, 1995.

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

[16] A. Yilmaz C. Rao and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision (IJCV)*, 50:203–226, 2002.

[17] C. Chang, R. Ansari, and A. Khokhar. Multiple object tracking with kernel particle filter. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[18] E. Chang. Event sensing on distributed video-sensor networks. In *Proc. ACM/IEEE Conf. on Broadband Networks*, San Jose, CA, Oct 2004.

[19] H. Chui and A. Rangarajan. A new algorithm for nonrigid point matching. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 44–51, Hilton Head Island, SC, Jun. 2000.

[20] R. T. Collins. Mean-shift blob tracking through scale space. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, Madison, Wisconsin, June 2003.

[21] R. T. Collins and Y. X. Liu. On-line selection of discriminative tracking features. In *Proc. IEEE Int'l Conf. on Computer Vision*, Nice, France, 2003.

[22] Robert Collins, Alan Lipton, Hironobu Fujiyoshi, and Takeo Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of IEEE*, 89:1456–1477, Oct. 2001.

[23] Robert Collins, Alan Lipton, and Takeo Kanade. Special issue on video surveillance and monitoring. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:745–746, 2000.

[24] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid targets using mean shift. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2000.

[25] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *In IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2003.

[26] T. F. Cootes, C. J. Taylor, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61:38–59, Jan. 1995.

[27] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation. In *European Conf. on Computer Vision*, volume III, pages 453–468, 2002.

[28] I.J. Cox and S.L. Hingorani. An efficient implementation of reid's multiple hypotheses tracking algorithm and its evaluation for the purposes of visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 18:138150, Feb 1996.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[30] L. Davis, I. Haritaouglu, and D. Harwood. Ghost: A human body part labeling system using silhouettes. In *Int'l Conf. on Pattern Recognition*, 1998.

[31] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *Proc. British Machine Vision Conference (BMVC)*, 2004.

[32] F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun. Em, mcmc, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 2003.

[33] A. P. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *In Journal of the Royal Statistical Society, Series B (Methodological)*, 1(39):1–38, 1977.

[34] Shiloh Dockstader and A. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proceedings of IEEE*, 89:1441–1455, Oct. 2001.

[35] Arnaud Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[36] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. IEEE International Conf. on Computer Vision (ICCV)*, 2003.

[37] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, Madison, Wisconsin, June 2003.

[38] A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proc. European Conf. on Computer Vision (ECCV)*, Dublin, Ireland, June 2000.

[39] Z. M. Fan, Y. Wu, and M. Yang. Multiple collaborative kernel tracking. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 502–509, San Diego, CA, 2005.

[40] Z. M. Fan, M. Yang, Y. Wu, G. Hua, and T. Yu. Efficient optimal kernel placement for reliable visual tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, June 2006.

[41] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, Jan. 2005.

[42] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. on Computer*, 1:67–92, Jan. 1973.

[43] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal Oceanic Eng.*, OE-8:173184, July 1983.

[44] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *In Int'l Journal of Computer Vision*, 40:25–47, 2000.

[45] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proc. of the Thirteenth Conf. on Uncertainty in Artificial Intelligence*, pages 246–252, 1997.

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

[47] D. M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *IEEE Int'l Conf. on Computer Vision*, pages 87–93, Corfu, Greece, Sept. 1999.

[48] D. Geiger and F. Girosi. Parallel and determinstic algorithms from MRFs: Surface reconstruction. *IEEE trans. on PAMI*, 13:401–412, 1991.

[49] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE trans. on PAMI*, 6:721–741, 1984.

[50] Zoubin Ghahramani and Michael Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.

[51] N. Gheissari, T. Sebastian, P. Tu, J. Rittscher, and R. Hartley. Person reidentification using spatiotemporal appearance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, June 2006.

[52] G. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1025–1039, Oct. 1998.

[53] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, Washington, D. C., June 2004.

[54] B. Han and L. Davis. On-line density-based appearance modeling for object tracking. In *In Proc. IEEE Int'l Conf. on Computer Vision (ICCV)*, Beijing, China, Oct. 2005.

[55] M. Han, W. Xu, H. Tao, and Y. H. Gong. An algorithm for multiple target trajectory tracking. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., June 2004.

[56] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. In *Proc. IEEE Int'l Conf. on Face and Gesture Recognition*, Nara, Japan, April 1998.

[57] M. Harville. A framework for high-level feedback to adaptive per-pixel, mixture-of-gaussian background models. In *Proc. European Conf. on Computer Vision (ECCV)*, 2002.

[58] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2001.

[59] J. Ho, K. C. Li, M. H. Yang, and D. Kriegman. Visual tracking using learned linear subspace. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, volume I, pages 782–289, Washington, D.C., June 2004.

[60] C. Hue, J. Cadre, and P. Perez. Tracking multiple targets with particle filtering. *In IEEE Transactions on Aerospace and Electronic Systems*, 38:791–812, March 2002.

[61] M. Isard. Pampas: Real-valued graphical models for computer vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 613–620, Madison, WI, June 2003.

[62] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conf. on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[63] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 34–41, Vancouver, Canada, 2001.

[64] Michael Isard and Andrew Blake. A mixed-state condensation tracker with automatic model-switching. In *Proc. of IEEE Int'l Conf. on Computer Vision*, pages 107–112, India, 1998.

[65] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22:852–872, 2000.

[66] T. S. Jaakkola. Tutorial on variational approximation methods. *MIT AI Lab TR*, 2000.

[67] O. Jave, Z. Rasheed, K. Shafique, and M. Shan. Tracking accross multiple cameras with disjoint views. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume II, pages 952–957, 2003.

[68] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust on-line appearance models for visual tracking. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 415–422, 2001.

[69] Nebojsa Jojic, Nemanja Petrovic, Brendan Frey, and Thomas S. Huang. Transformed hidden Markov models: Estimating mixture models and inferring spatial transformations in video sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000.

[70] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 2000.

[71] M. Kass, A. Witkin, and D. Terzopoulos. Snake: Active contour models. In *Int'l Conf. on Computer Vision*, pages 259–268, 1987.

[72] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[73] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *Proc. of European Conf. on Computer Vision*, 2004.

[74] T. Kiribarajan, Y. Bar-Shalom, and K. R. Pattipati. Multiassignment for tracking a large number of overlapping objects. *In IEEE Transactions on Aerospace and Electronic Systems*, 37:2–21, 2001.

[75] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 878–885, 2005.

[76] T. Leung, M. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 637–644, June 1995.

[77] D. Li, K. Wong, Y. Hu, and A. Sayeed. Detection, classification and tracking of targets in distributed sensor networks. *In IEEE Signal Processing Magazine*, 19, March 2002.

[78] S. Z. Li, X. G. Lv, and H. J. Zhang. View-based clustering of object appearances based on indepedent subspace analysis. In *Proc. IEEE Int'l Conf. on Computer Vision*, Vancouver, Canada, July 2001.

[79] S. Z. Li, L. Zhu, Z. Q. Zhang, A. Blake, H. J. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *Proc. European Conf. Computer Vision*, 2002.

[80] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22:413–427, May 2004.

[81] Z.-M. Lihi and I. Michal. Event-based analysis of video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[82] J. W. Lim, D. Ross, R. S. Lin, and M. H. Yang. Incremental learning for visual tracking. In *In Proc. Neural Information Processing Systems 17 (NIPS)*, 2005.

[83] Ce Liu, Song Chun Zhu, and Heung-Yeung Shum. Learning inhomogeneous gibbs model of faces by minimax entropy. In *IEEE Int'l Conf. on Computer Vision*, Vancouver,Canada, July 2001.

[84] Jun Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93:1032–1044, 1998.

[85] Jun Liu, Rong Chen, and Tanya Logvinenko. A theoretical framework for sequential importance sampling and resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2000.

[86] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 572–578, Corfu, Greece, 1999.

[87] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23:873–889, 2001.

[88] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proc. European Conf. Computer Vision*, 2004.

[89] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, Washington, D. C., June 2004.

[90] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 349–362, April 2001.

[91] S. A. Murray. Human-machine interaction with multiple autonomous sensors. *Navy Command, Control and Ocean Surveillance Center, RDT&E Division, San Diego, California*.

[92] A. K. Roy-Chowdhury N. Vaswani and R. Chellappa. shape activity: a continuous-state hmm for moving/deforming shapes with applications to abnormal activity detection. *IEEE Trans. on Image Processing (TIP)*, 14:1603–1616, 2005.

[93] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.

[94] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Proc. of the IEEE International Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004.

[95] K. Okuma, A. Taleghani, N. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: multitarget detection and tracking. In *Proc. of European Conf. on Computer Vision*, 2004.

[96] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[97] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[98] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38:15–33, 2000.

[99] Ioannis Pavlidis, Vassilios Morellas, Panagiotis Tsiamyrtzis, and Steve Harp. Urban surveillance systems: From the laboratory to the commercial world. *Proceedings of IEEE*, 89:1456–1477, Oct. 2001.

[100] V. Pavlovic, J. Rehg, T. Cham, and K. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume I, pages 94–101, Corfu, Greece, Sept 1999.

[101] Vladimir Pavlovic. *Dynamic Bayesian Networks for Information Fusion with Application to Human-Computer Interfaces*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 1999.

[102] Vladimir Pavlović, R. Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human computer interaction: A review. *IEEE Trans. on PAMI*, 19:677–695, July 1997.

[103] Alex Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:107–119, Jan. 2000.

[104] C. Peterson and J. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, pages 995–1019, 1987.

[105] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 467–474, Madison, WI, June 2003.

[106] Anand Rangarajan, James Coughlan, and Alan Yuille. A bayesian network framework for relational shape matching. In *IEEE Int'l Conf. on Computer Vision*, volume I, pages 671–678, Nice, France, Oct. 2003.

[107] C. Rasmussen and G. Hager. Probabilistic data association methods for tracking complex visual objects. *In IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:560–576, June 2001.

[108] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, AC-24:843854, Dec. 1979.

[109] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *Proc. European Conf. on Computer Vision (ECCV)*, 2000.

[110] Henry Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, Jan. 1998.

[111] Y. Rui and Y. Q. Chen. Better proposal distributions: target tracking using unscented particle filter. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001.

[112] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 746–751, Hilton Head Island, SC, 2000.

[113] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Int'l Journal of Computer Vision*, 2003.

[114] Henry Schneiderman and Takeo Kanade. Object detection using the statistic of parts. *Int. J. Computer Vision*, 56(3):151–177, February 2004.

[115] S. Sclaroff and A. Pentland. Model matching for correspondence and recognition. *IEEE trans. on PAMI*, 17:545–561, 1995.

[116] Y. Shan, H. S. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, 2005.

[117] Y. Sheikh and M. Shah. Exploring the space of an action for human action recognition. In *Proc. IEEE International Conf. on Computer Vision (ICCV)*, 2005.

[118] L. Sigal, M. Isard, B. Sigelman, and M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Proc. Neural Information Processing Systems*, 2004.

[119] C. Stauffer, W. Eric, and L. Grimson. Learning patterns of activity using real-time tracking. In *In IEEE Trans on Pattern Analysis and Machine Intelligence*, volume 22, pages 747–757, 2000.

[120] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, page 246252, 1999.

[121] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 605–612, Madison, WI, June 2003.

[122] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Proc. Neural Information Processing Systems*, June 2004.

[123] H. Tao, H. Sawhney, and R. Kumar. A sampling algorithm for detecting and tracking multiple objects. In *Proc. ICCV'99 Workshop on Vision Algorithm*, Corfu, Greece, 1999.

[124] H. Tao, H. S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 134–141, 2000.

[125] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. Int'l Conf. on Computer Vision (ICCV)*, 1999.

[126] Kentaro Toyama and Andrew Blake. Probabilistic tracking in a metric space. In *Proc. IEEE Int'l Conf. on Computer Vision*, Vancouver, Canada, July 2001.

[127] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. IEEE Int'l Conf. on Computer Vision*, Nice, France, 2003.

[128] P. Viola and M. Jones. Rapid target detection using a boosted cascade of simple features. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2001.

[129] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. IEEE Int'l Conf. on Computer Vision*, Nice, France, 2003.

[130] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[131] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Proc. IEEE Int'l Conf. Computer Vision*, 2003.

[132] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real time tracking of the human body. *In IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1997.

[133] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *IEEE Int'l Conf. on Computer Vision*, pages 90–97, 2005.

[134] Y. Wu, G. Hua, and T. Yu. Switching observation models for contour tracking in clutter. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 295–302, Madison, WI, June 2003.

[135] Y. Wu, G. Hua, and T. Yu. Tracking articulated body by dynamic markov network. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1094–1101, 2003.

[136] Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking. In *In Proc. IEEE Int'l Conf. on Computer Vision (ICCV)*, pages 26–33, 2001.

[137] Y. Wu and T. Yu. A field model for human detection and tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28, 2006.

[138] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 789–795, Madison, WI, June 2003.

[139] Y. Wu, T. Yu, and G. Hua. A statistical field model for pedestrian detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 20–26, San Diego, CA, June 2005.

[140] Y. Wu, T. Yu, and G. Hua. A statistical field model for pedestrian detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[141] Ying Wu. *Vision and Learning for Intelligent Human-Computer Interaction*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2001.

[142] Ying Wu and Thomas S. Huang. Robust visual tracking by co-inference learning. volume II, pages 26–33, Vancouver, July 2001.

[143] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision (IJCV)*, 67:21–51, 2006.

[144] B. L. Xie, D. Comaniciu, V. Ramesh, M. Simon, and T. Boult. Component fusion for face detection in the presence of heteroscedastic noise. In *Annual Conf. of the German Society for Pattern Recognition*, pages 434–441, 2003.

[145] A. Yamada, Y. Shirai, and J. Miura. Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games. In *Proc. IEEE Int'l Conf. on Pattern Recognition*, volume I, pages 303–306, 2002.

[146] C. J. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measurement. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, 2005.

[147] M. Yang and Y. Wu. Tracking non-stationary appearances and dynamic feature selection. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1059–1066, San Diego, CA, 2005.

[148] M. Yang and Y. Wu. Tracking non-stationary appearances and dynamic feature selection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[149] A. Yilmaz and M. Shah. Actions as objects: a novel action representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[150] T. Yu, M. Han, and Y. H. Gong. Home care/monitor system - unusual event detection. In *NEC Labs America Technical Report*, 2004.

[151] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington D.C., June 2004.

[152] T. Yu and Y. Wu. Collaborative visual tracking of multiple identical targets. In *Proc. SPIE Conf. on Storage and Retrieval Methods and Applications for Multimedia*, San Jose, Jan 2005.

[153] T. Yu and Y. Wu. Decentralized multiple target tracking using netted collaborative autonomous trackers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[154] T. Yu and Y. Wu. Collaborative support vector tracking. *under submission to the IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[155] T. Yu and Y. Wu. Decentralized multiple target tracking. *submitted to the IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[156] T. Yu and Y. Wu. Differential tracking based on spatial-appearance model (sam). In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, June 2006.

[157] T. Yu and Y. Wu. Stochastic video partition. *submitted to the IEEE Trans. on Image Processing*, 2006.

[158] A. Yuille. Deformable templates for face recognition. *J. of Cognitive Neuroscience*, 3, 1991.

[159] H. H. Zhang, W. M. Huang, Z. Y. Huang, and L. Y. Li. Affine object tracking with kernel-based spatial-color representation. In *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, 2005.

[160] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Washington D.C., June 2004.

[161] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 819–826, 2004.

[162] B. Zhou and N.K. Bose. An efficient algorithm for data association in multitarget tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 31:458–468, Jan. 1995.

[163] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Frame: Filters, random field and maximum entropy: - towards a unified theory for texture modeling. *In Int'l Journal of Computer Vision*, 27:1–20, 1998.

CHAPTER 8

Appendix

This appendix gives the derivation of the mean field approximation of Eq. 4.7. Based on Eq. 4.5 and 4.6, we have:

$$J(Q_i) = H(Q_i) + \sum_{k \neq i} H(Q_k) + \int_{x_i} Q_i E_Q[\log p(\mathbf{X}, \mathbf{Z}) | x_i]$$

Since Q_i is a distribution, we can construct a Lagrangian:

$$L(Q_i) = J(Q_i) + \Lambda(\int_{x_i} Q_i - 1)$$

Then, the derivative of $L(Q_i)$ w.r.t. Q_i gives:

$$\frac{\partial L(Q_i)}{\partial Q_i} = -\log Q_i - 1 - E_Q[\log p(\mathbf{X}, \mathbf{Z}) | x_i] + \Lambda$$

Once we set the derivative to zero, we obtain:

$$Q_i = e^{-1+\Lambda+E_Q[\log p(\mathbf{X},\mathbf{Z})|x_i]} = \frac{1}{Z_i} e^{E_Q[\log p(\mathbf{X},\mathbf{Z})|x_i]}$$

Vita

Ting Yu received the BS and MS degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2000 and 2002. He is currently a PhD candidate in the Department of Electrical and Computer Engineering at Northwestern University, Evanston, Illinois. During the summers of 2004 and 2005, he was a research intern with the NEC Labs America, Cupertino, California, and Microsoft Research, Redmond, Washington, respectively. Starting from Oct. 2006, he will join the Visualization and Computer Vision Lab as a research scientist at GE Global Research, Niskayuna, NY. His research interests include computer vision, image/video processing and analysis, statistical learning, pattern recognition and data mining. He received the Walter P. Murphy Fellowship at Northwestern in 2002, and the Motorola Graduate Scholarship and Excellent Student Scholarships at Tsinghua in 2001, 1999, and 1997. He is a student member of the IEEE.