Robust Visual Tracking for Multiple Targets

by

Yizheng Cai

B.E., Zhejiang University, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

The University of British Columbia

September 2005

© Yizheng Cai, 2005

Abstract

We address the problem of robust multi-target tracking within the application of hockey player tracking. Although there has been extensive work in multi-target tracking, there is no existing visual tracking system that can automatically and robustly track a variable number of targets and correctly maintain their identities with a monocular camera regardless of background clutter, camera motion and frequent mutual occlusion between targets. We build our system on the basis of the previous work by Okuma et al. [OTdF⁺04]. The particle filter technique is adopted and modified to fit into the multi-target tracking framework. A rectification technique is employed to map the locations of players from the video frame coordinate system to the standard hockey rink coordinates so that the system can compensate for camera motion and the dynamics of players on the rink can be improved by a second order auto-regression model. A global nearest neighbor data association algorithm is introduced to assign boosting detections to the existing tracks for the proposal distribution in particle filters. The mean-shift algorithm is embedded into the particle filter framework to stabilize the trajectories of the targets for robust tracking during mutual occlusion. The color model of the targets is also improved by the kernel introduced by mean-shift. Experimental results show that our system is able to correctly track all the targets in the scene even if they are partially or completely occluded for a period of time.

Key words: Multi-target tracking, Homography, Sequential Monte Carlo methods, Data association, Mean-shift

Contents

Abstract	ii
Contents	iii
List of Tables	v
List of Figures	vi
Acknowledgements	ix
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Challenges	2
1.3.1 Target localization	2
1.3.2 Target representation	3
1.3.3 Non-rigidity of the target	3
1.3.4 Tracking stabilization	3
1.3.5 Mutual occlusion	4
1.4 Contributions	6
1.5 Thesis Outline	6
2 Previous Work	8
2.1 Target Representation and Localization	8
2.1.1 Representation	8
2.1.2 Localization	10
2.2 Filtering and Data Association	10
2.2.1 Filtering	11
2.2.2 Data association	14
2.2.3 Mutual exclusion	18
2.3 Summary	18

3	Particle Filter for Multi-target Tracking	20
3.1	Particle Filter	20
3.1.1	Monte Carlo simulation	21
3.1.2	Sequential importance sampling	21
3.1.3	Resampling	22
3.2	Boosted Particle Filter	23
3.3	Our Improvements	24
4	Target Dynamics Modelling	26
4.1	Rectification	26
4.1.1	Homography	26
4.1.2	Multi-layer state space model	28
4.2	Autoregressive Dynamics Model	30
5	Data Association	33
5.1	Gating	33
5.2	Linear Optimization	35
6	Mean-Shift Embedded Particle Filter	38
6.1	Color Model	38
6.2	Mean-Shift	41
6.3	Adaptive Scaling	45
6.4	Mean-shift Embedded Particle Filter	46
7	Experimental Results and Conclusion	51
7.1	Results	51
7.2	Conclusion	60
7.3	Discussion and Future Extension	61
	Bibliography	64

List of Tables

5.1	Assignment matrix for the assignment problem shown in Figure 5.1 .	36
-----	--	----

List of Figures

1.1	The figure shows that different players can have quite difference shapes so that it is impossible to use a bounding box with a fixed edge ratio.	4
1.2	The above 4 frames show that the tracker of the target drifts significantly over time	5
1.3	The figure shows the situation of complete occlusion and partial occlusion during the process. The rectangles show the regions where occlusions take place.	6
2.1	Graphical model representation of the relationship between an observed variable y_t and a latent variable x_t	11
2.2	Graphical model representation of the Hidden Markov Model, where x_0 is the initial state	12
3.1	The SIR process starts at time $t-1$ with a set of equally weighted particles $\{\tilde{x}_{t-1}^{(i)}, N^{-1}\}_{i=1}^N$, which represent the distribution $p(x_{t-1} y_{0:t-2})$. Importance weights are updated with information at time $t-1$ to represent $p(x_{t-1} y_{0:t-1})$ with $\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}_{i=1}^N$. Resampling is performed to make it a equallyly weighted particle set $\{x_{t-1}^{(i)}, N^{-1}\}_{i=1}^N$ again but still represent the same distribution. Finally, particles are propagated to $\{\tilde{x}_t^{(i)}, N^{-1}\}_{i=1}^N$ at time t by the proposal distribution to represent $p(x_t y_{0:t-1})$ and iterations go forward [vdMAdFW00].	23
3.2	The final proposal is in the form of a mixture of Gaussian which combines the prior and the boosting detection [OTdF ⁺ 04].	24
4.1	This shows how video frames are mapped to the rink. The top left image is the input video frame; the top right is the standard rink model; the bottom image shows how the image looks after being mapped onto the rink.	29
4.2	Graphical model representation of the three-layer structure of the state space model.	30

4.3	All the black rectangles are the boosting detections. Black dots on the bottom edge of each rectangle represent the positions of the players on the rink.	31
5.1	T1, T2, T3 are predicted target positions; O1, O2, O3, O4 are observations. O2 is assigned to T1; O1 is a candidate of T1, T2 and T3; O4 is a candidate of T2 and T3; O3 is not assigned to any existing tracks so that it is a candidate of being a new target.	34
5.2	T1, T2 are predicted positions of targets; V1, V2 are vectors that indicate the velocities of the targets.	35
6.1	Multi-part color histogram of two players from two different teams. .	41
6.2	This figure shows how significantly the color histogram of the same target changes over time.	42
6.3	Comparison between the original track output and the mean-shift stabilized result. Left ones are the original track result and the right ones are the stabilized result.	47
6.4	This figure shows the tracking result of the system in which the particles are biased by the mean-shift algorithm after the deterministic resampling step in particle filtering. The top frame shows the overall view of the tracking result at frame 1. The six frames below show the tracking result with a close-up view of the three players in the center region. Particles from the same target are represented by rectangles with same color.	49
7.1	This figure shows an example of the blur from interlacing.	52
7.2	This figure shows the overall view of the camera at the initial stage. The close-up view of the tracking results will be shown in Figure 7.3 by zooming into the rectangular region.	52
7.3	This sequence is a close-up view of the rectangular region in Figure 7.2. The left column is the tracking results of the system in [OTdF ⁺ 04]. The right column is the tracking results of our system. Different targets are labelled with rectangles of different colors. . . .	53
7.4	This figure shows the tracking result of the system that only has the camera motion compensated and the dynamics model changed but without mean-shift bias. The left column is the results of the system in which the deviation for the scale sampling is 0.02; the right column is with the deviation of 0.01.	55

7.5	This figure shows the particle representation of each target. The left column shows the particles before the mean-shift bias; the middle column shows the particles after the mean-shift bias; the right column shows the particles after the resampling. Each particle is represented with a rectangle and particles that belong to the same target are of the same color.	57
7.6	This figure shows the trajectories of the three targets shown in Figure 7.3. (b)(c)(d)(e) show the key frames during the process. (b) is the starting frame and (e) is the ending frame.	58
7.7	This figure shows the tracking result of the period when partial occlusion happens. (a) is the overall view of the tracking result at frame 78. (b)(c)(d)(e) show the close-up view of the four targets in the region specified in (a).	59
7.8	This figure shows how the video frame is mapped onto the standard rink.	62

Acknowledgements

I would like to acknowledge those who gave me great help during my thesis work. Without their support, I can never accomplish this work. My sincere gratitude goes to my supervisors, Dr. James Little and Dr. Nando de Freitas for their passion and patience in providing me with extraordinary guidance and invaluable ideas. I also would like to thank Dr. Kevin Murphy and Dr. David Lowe for their enlightening suggestions. My appreciation extends to my colleagues, Xiaojing Wu, Kenji Okuma, Fahong Li, and Yongning Zhu for their beneficial discussions. My special thanks go to my family for their indispensable support. Finally, I wish to acknowledge the funding support of the National Science and Engineering Research Council of Canada and GEOIDE, a Canadian Network of Centres of Excellence.

YIZHENG CAI

*The University of British Columbia
September 2005*

Chapter 1

Introduction

1.1 Motivation

Tracking multiple targets, although has its root in control theory, has been of broad interest in many computer vision applications for decades as well. People or vehicle tracking for public surveillance systems, autonomous robots and vehicles steering, and sports game video annotation are all promising applications. Objects of interest for tracking can be of any kind, including landmarks, vehicles, people, and various kinds of visual features. The number of objects can vary over time. The backgrounds of the scenes to be analyzed can either be static or dynamic. All these complexities, which arise in the real world, make the problem non-trivial.

Tracking players in sports game videos is one of the most practical applications for multi-target tracking because of the popularity of the games and the pervasive demand for automatic annotation and analysis of such videos for broadcasting and coaching purposes. It is also a challenging application because of its inherent properties including the varying number of players in the scene, the dynamically changing background due to camera motion, and non-rigid target shapes due to various player poses. In the world of computer vision research, such applications have already been investigated on the soccer games [VDP03, KCM03] and hockey games [OTdF⁺04].

Recently, research on more higher level of motion analysis has also been carried out. Efros et al. [EBMM03] focus on the character of the motion of soccer players (i.e., walking, running, standing) rather than their translational motion and assume that the targets are stably tracked so that the targets are roughly centered within the bounding box of the tracker. Li et al. [Li04] aim to represent and reason about hockey behaviors for an even higher level of motion understanding. Such applications all require accurate trajectory data extracted from lower level tracking systems as their input. Therefore, acquisition of accurate trajectory data over long video sequences becomes crucial. To generate such required trajectory data, one of the most critical properties of the tracking system is its robustness in maintaining

the correct identity of different targets under various conditions. However, existing systems either have difficulties in maintaining a variable number of targets or have problems in keeping the correct identity of targets when occlusion happens between different targets. Being aware of the weaknesses in the existing systems, we aim to build a tracking system that can robustly track multiple targets and keep their identities over long video sequences captured by a monocular camera.

1.2 Problem Statement

As an extension of the previous endeavors devoted to building robust multi-target tracking systems, and a preparation for researches on high level action analysis, this thesis addresses the task of building a robust tracking system which can handle a variable number of targets and maintain the accurate identity of each target regardless of background clutter, camera motion and frequent mutual occlusions between targets. The system is required to have the ability to track targets over long sequences and be robust under perturbations including illumination condition changes, shadows of the targets and so on.

1.3 Challenges

Although extensive research on various tracking systems has already conquered many critical obstacles in target localization and filtering in tracking, there still remains challenging problems in our application. We present in this section the following major challenges: target localization, target representation, non-rigidity of the targets, tracking stabilization, and mutual occlusion.

1.3.1 Target localization

Object detection has been a major focus of research in computer vision. Learning-based object detection [RBK98, VJ04] is one of the most effective and efficient methods. Camera motions, including panning, tilt, and zooming, make the video frame coordinates variable with respect to the coordinate system, which is the hockey rink in this case. Multiple cameras are able to solve the problem of localizing targets in the real world coordinate system [KCM04] so long as the relative location and configuration of different cameras are known and well synchronized. However, because the source data we use for this task is extracted from only one monocular TV broadcasting camera, it is very difficult to derive even the relative positions among the targets in the real world. Although 2D or 3D geometry of the scene is one possible way to assist solving the problem for moving monocular camera, it is still a non-trivial problem.

1.3.2 Target representation

Because of the inherent weakness in video standards of National Television System Committee (NTSC), the video frames extracted from the recorded video clips are highly unsaturated especially in hockey games. Therefore, the representation of the targets in the color space becomes quite noisy and unstable so that it is not salient enough to support tracking as the only information. In addition, the resolution of the video frames is so low that each target is sometimes as small as twenty pixels high. As a result, a significant number of pixels that belong to one target are probably contaminated by the noise from the hardware device because of the limited number of the total pixels of the target. Also, as is well known, hockey is a fast game where players tend to move so quickly that the targets that move in the opposite direction of the camera motion appear blurry in the video frames. Therefore the representation of those blurred targets in the feature space will become much different from the ground truth.

1.3.3 Non-rigidity of the target

Human bodies are one group of objects that are very difficult to detect or track because of their non-rigidity. Research has been done to detect or track human bodies as a whole [POP98, VJS05, IM01] or separately on different parts of the body with some geometric constraints [MSZ04, RF03]. As is mentioned in the previous section, because of the low resolution of the input video clips, tracking parts of the human body separately in the same way as [RF03] does is almost infeasible in this application. In the systems that detect or track people as a whole, human bodies are more or less upright and straight so that most of the bodies fit into a rectangular bounding box with a fixed length ratio of the edges. However, it will cause problems in our application because it is impossible to find such a bounding box to capture the whole body of the hockey players that has drastic pose changes during the game. Examples are shown in Figure 1.1. The optimal way is to use varying shape boxes to capture the targets. However, as we use the detection results from the boosting detectors, we straightforwardly inherit the fixed shape bounding box as well. This brings some disadvantages because, by using such kind of bounding boxes with fixed edge ratio, it is possible that important pixels are excluded from the box and pixels that do not belong to the target would be included. Hence, the representation of the targets in the feature space will be contaminated.

1.3.4 Tracking stabilization

Taking the bounding box tracker as an example, it is very difficult to stably fit the boxes on the targets. The tracker tends to drift off the targets more or less over time. Examples can be found in Figure 1.2. This is the tracking output of the system

Figure 1.1: The figure shows that different players can have quite difference shapes so that it is impossible to use a bounding box with a fixed edge ratio.

by [OTdF⁺04]. Only one track is shown for clear representation. Having a stable tracker is important because as long as we need the trajectories of the targets to predict their locations, it is important that the trajectories are reliable. Otherwise, the prediction would lead the tracker to a region far away from the true target location. In most cases, such errors are irrevocable so that the tracker loses the target or tracks on to a different target. Although, intuitively, people can tell which tracker is more stable than the other one, there is no broadly accepted concrete standard to define the concept of stability. In addition, even though it is possible to define a standard for our particular application, it is difficult to evaluate the stability of the tracking result and perform certain corrections because the ground truth is unknown.

1.3.5 Mutual occlusion

Compared to the challenges described in the previous sections, mutual occlusion is a higher level vision problem which requires the understanding of the scene rather than the video frames only. Because the source data is captured by a single fixed monocular camera, only the 2D projection of the real world is captured by the camera. Consequently, there are always occasions when different targets occlude each other in the video frames as they cross over each other. This can be avoided if the camera records the games from the top down view. However, it is not the case in most broadcasting videos. Occlusions can be partial or even complete for a certain period of time. Most detectors fail in those situations because they treat input video frames as independent and static images without any context information. Therefore, it is impossible to detect a target that is completely occluded. Partial

Figure 1.2: The above 4 frames show that the tracker of the target drifts significantly over time

occlusion is also problematic in detectors that use some templates or training sets with only complete objects for detection. Consequently, trackers that only make use of the visual information or the output of the detector to locate targets fail during occlusion. Although tracking systems may make use of spatial and temporal continuity properties of the video clips to predict the location of the targets, a system still has difficulties in correctly tracking the targets if the complete occlusion results in the absence of the target in one spot and its reappearance somewhere else. Partially occluded targets are also difficult to track because of the insufficient visual support by those pixels that are not occluded. Therefore, without understanding of the context and the spatial configuration of the scene, it is impossible to solve this problem. Figure 1.3 shows occasions that partial and complete occlusions take place.

Figure 1.3: The figure shows the situation of complete occlusion and partial occlusion during the process. The rectangles show the regions where occlusions take place.

1.4 Contributions

Challenges have been investigated and described in the previous section. In order to conquer those challenges and build a robust multi-target tracking system to correctly track hockey players during the games, we propose four improvements on the previous systems. Firstly, a rectification technique is employed to compensate for camera motions by mapping the locations of the players from the video frame coordinate system to the time invariant standard hockey rink coordinate system. Secondly, a second order autoregression model is adopted as the dynamics model to predict the location of the targets. Thirdly, a global nearest neighbor data association technique is used to correctly associate boosting detections with the existing tracks. Finally, the mean-shift algorithm is embedded into the particle filter framework [AMGC01] to stabilize the trajectories of the targets. By assembling all the contributions, the new multi-target tracking system is capable of achieving the goal stated in Section 1.2. The details of all the techniques employed will be described in the following chapters.

1.5 Thesis Outline

The remainder of the thesis will organized as follows. In the next chapter, a review of modern tracking systems will be presented with emphasis on multi-target tracking. Previous work on object representation, motion modelling and data association will also be covered. Chapter 3 gives a general explanation of Sequential Monte Carlo Methods [DdFG01], also known as particle filters, because it forms the basic skeleton of our tracking system. Extensions on the conventional particle filter to support multi-target tracking in our system will be introduced as well. Chapter 4

describes how to improve the modelling of the dynamics of each target so that the motion information can provide enough support for the tracker while visual support is absent or unreliable. A sketch of the previous rectification technique [OLL04], which will be used in our work to compensate camera motion, will be shown and the autoregressive dynamics model will be presented in detail as well. Chapter 5 addresses the importance of the data association issue in our system. A global nearest neighbor data association, which is designed particularly for our application, will be introduced and explained in detail. Chapter 6 demonstrates how the representation of the targets in the color space is improved by a kernel function. Mean-shift, known as a gradient descent algorithm, will be introduced and how it is embedded into the particle filter framework will be described in detail. In Chapter 7, experiment results of our new tracking system will be presented. We conclude our work by summarizing the contributions we have made and analyze strengths and weaknesses in our system. Finally, some possible extensions will be discussed for the future work.

Chapter 2

Previous Work

A typical visual tracking system can be divided into two parts [CRM03]: one is *Target Representation and Localization*, which has its root in computer vision; the other is *Filtering and Data Association*, which comes from control theory. In this chapter, we present a brief introduction of previous work in both aspects with emphasis on their strength and weakness.

2.1 Target Representation and Localization

2.1.1 Representation

Target representation can be categorized into two major classes. One is for a group of generic objects, such as human faces or bodies, computer monitors, motorcycles, and so on. The other is for one specific target including a specific person, car, toy, building and so on. The targets can be images, concrete objects or even abstract feature points.

For the first class, the representation should capture the common features among all the objects in the same group. As it is beyond the scope of this thesis, we suggest readers refer to other related works. [YKA02] is one such reference that summarizes the approaches in face detection. However, some approaches also apply to the representation of other objects. In our system, hockey players are detected by a boosting algorithm, which is based on the work by Viola et al. [VJ04]. It treats the hockey players as generic objects rather than specific objects. However, as we assume the boosting detections already exist as the input of our tracking system, we skip the details of the detection algorithm. For the tracking system, in order to distinguish different targets and keep their identity over time, it is necessary to treat each individual as a specific object. Therefore, we focus on the representation of specific objects in this thesis.

For specific object representation, salience and uniqueness are the most important characteristics. One popular approach is to represent an object or an image

by a set of local features, which are unique and distinguish the object from others. The *Harris corners* [HS88], the *Kanade-Lucas-Tomasi (KLT) features* [TK91] and the *Scale Invariant Feature Transform (SIFT)* [Low04] are three typical examples. These features can be applied to specific object recognition and tracking, image matching and rectification, 3D structure reconstruction from 2D images for scene or motion analysis, and even *Simultaneous Localization and Mapping (SLAM)* [DNCC01] for robotics.

In our work, KLT features are used for image rectification [OLL04]. As is mentioned in Section 1.3.1, the locations of the hockey players in the input video are in temporally variant image coordinates because of the camera motion. Therefore it is difficult to model and predict the motion of the players in the image coordinate system. Mapping the locations of players from the image coordinate system to the standard hockey rink coordinates, which is invariant over time, is necessary for the modelling of the dynamics of the players. The KLT feature tracker is used to match the selected KLT features extracted from consecutive video frames so that the mapping between frames can be computed according to the geometrical properties in the 2D plane-to-plane projection. A detailed introduction of the plane-to-plane mapping and the mechanism to map between the frames and hockey rinks will be given in Chapter 4.

A local feature based approach is effective for recognizing targets that have sufficient salient local features because pose changes and self-occlusions may result in the absence of a significant proportion of irreplaceable local features. However, as we mentioned in Section 1.3.2, the resolution of all the players in our application is quite low. Therefore, it is difficult to extract sufficient local features to overcome the negative effect brought by occlusion. In addition, because of the non-rigidity of the targets, the relative geometry between local features can not be modelled with a static template. Taking into account all these factors, a global representation of the target is needed for our application to conquer the obstacles. Color-based representation is one of the most successful approaches in the literature. Both Pérez [PHVG02, VDP03] and Comaniciu [CRM03] have successfully used color histograms to represent targets for tracking.

The algorithm represents a specified image region with a color histogram, which can be extracted from any color space. The histogram is represented by a vector and the value of each entry of the vector reflects the cumulative count of each bin in the histogram. Because the purpose is to track a specific target in the scene, a reference histogram of the particular target is created at the initialization step. A metric based on Bhattacharyya coefficient [Kai67] is adopted to evaluate the similarity between two color histograms.

There are two major weaknesses in this color-based approach. One is the loss of spatial information because the histogram approach only takes into account the value of pixels in the color space rather than the location information. The

other is the interference from the background clutter and other targets. If some background regions or other targets had similar histograms as the target of interest, the tracker would probably get confused. To solve the first problem, a multi-part color histogram representation is used to encode the spatial information. Particle filters set up an excellent framework for the color-based approach to solve the second problem. Proper modelling of the dynamics of the targets can assist in resolving the ambiguity that arises when background clutter and similar targets overlap with the target of interest.

2.1.2 Localization

The aim of localization is to detect the existence of the objects of interest and give their location for each of them. Taking the human face detection as an example, various approaches have been applied to detect generic faces. Template based approaches and learning based approaches are the two main strategies. Statistical learning approaches, such as *Neural Networks*, *Support Vector Machine* and *Boosting*, have become popular. The Viola and Jones [VJ04] approach, which makes use of *AdaBoost*, is known as one of the most successful classifiers and can be applied to many different classes of objects. Okuma et al. [OTdF⁺04] incorporate the boosting detector into a particle filter based tracking system. In the new framework, which they called *Boosted Particle Filter* (BPF), the three major functions of the boosting detector in the tracking system are initializing trackers, adding new targets, and improving the proposal density function for the particle filter. They represent the proposal distribution with a mixture of Gaussian model, which combines the boosting detections and the corresponding tracks. As a result, the data association problem arises when searching for the correspondence between the boosting detections and the existing tracks if multiple targets are in the scene. In [OTdF⁺04], the nearest neighbor association method is used, which only finds the local optimal rather than the global optimal solution. We improve their work by using a global nearest neighbor association method to reach the global optimal observation-to-track assignment. The in-depth introduction can be found in Chapter 5.

2.2 Filtering and Data Association

Filtering and data association have their theoretical basis in control theory, where the signals of the targets are mostly from radar, sonar, laser and so on. In our application, visual information is the observed signal from the targets. Both Bar-Shalom et al. [BSF88] and Blackman et al. [BP99] summarized modern techniques in filtering and data association in their books. In this section, we break down the topic into three parts: filtering, data association and mutual exclusion. The last part discusses problems that arise from visual-based multi-target tracking.

Figure 2.1: Graphical model representation of the relationship between an observed variable y_t and a latent variable x_t

2.2.1 Filtering

In the field of object tracking, *Bayesian Sequential Estimation* [AM79], which is also called *Bayesian Filtering*, is the most widely accepted framework. In this section, we present the fundamental theory of Bayesian filtering as a preparation for its sequential Monte Carlo implementation which is also known as *Particle Filters* [DdFG01]. We adopt it as the basic skeleton of our tracking system. In addition, we introduce some previous work that use two major implementations of Bayesian filtering: *Kalman Filters* and *Particle Filters*.

Bayesian sequential estimation

During the tracking process, the evolution of the targets can be interpreted as a state transition process, where each state can be represented by a vector that includes the location of the target in the reference coordinate system, the scale and velocity of the target, and any values that are required to represent the complete property of the target. Here, we denote the state variable as $x_t \in \mathbb{R}^{n_x}$, where $t \in \mathbb{N}$ indicates the time step and n_x is the dimension of the state variable. In the real world, the state variables are always difficult to observe or measure directly. For example, the aim of a tracking system is to estimate the ground truth (or the state variables) of the targets through the output data of the observation or measuring devices. Therefore, such state variables are called hidden variables or latent variables. The observation of the hidden variables are denoted as $y_t \in \mathbb{R}^{n_y}$, where n_y is the dimension of the observed variable. In most cases, $x_t \in \mathbb{R}^{n_x}$ and $y_t \in \mathbb{R}^{n_y}$ are in different spaces, therefore n_x and n_y are not necessarily the same. The relation between the two variables can be represented by a generative model and the corresponding graphical model can be represented as is shown in 2.1. The directed graph structure shows that the observed variable only depends on its corresponding latent variable and the conditional probability lying on the edge can be represented as $p(y_t|x_t)$.

The *Hidden Markov Model* (HMM) [AM79] is known to be an appropriate

Figure 2.2: Graphical model representation of the Hidden Markov Model, where x_0 is the initial state

model for sequential data (e.g., the tracking data). It links the data of each time step, which can be represented by a single node structure as is shown in Figure 2.1, as a chain to represent the dependence relationship between the nodes. The graphical model to represent such kind of chain structure is shown in Figure 2.2. x_0 is the initial state to start the whole process. In an HMM, the latent variables are discrete so that the transition probability on the arrows between latent variables is an $M \times M$ matrix, if the number of possible values is M . However, in most tracking applications, the locations of the targets are often represented in a continuous 2D or 3D space. Therefore, the State Space Model (SSM) [AM79], which generalizes HMMs to the continuous space but shares the same graphical model structure as HMMs, is used to model the tracking process in our application.

The goal of Bayesian sequential estimation is to estimate the posterior distribution $p(x_{0:t}|y_{0:t})$ ¹ or its marginal $p(x_t|y_{0:t})$. The transition between two consecutive latent variables is given by Equation 2.1 and the observation model is given by Equation 2.2

$$x_t = f_t(x_{t-1}, v_{t-1}), \quad (2.1)$$

$$y_t = h_t(x_t, u_t), \quad (2.2)$$

where $f_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ and $h_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ can either be linear or nonlinear, v_t and u_t are sequences of i.i.d. process noise and measurement noise respectively, and n_v and n_u are the dimensions of the noise vectors.

The marginal posterior distribution can be computed by the following two

¹ $x_{0:t} \triangleq (x_0, \dots, x_t), y_{0:t} \triangleq (y_0, \dots, y_t)$

step recursion based on the Markov assumptions ².

$$\begin{aligned} \text{prediction step: } p(x_t|y_{0:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})dx_{t-1} \\ \text{filtering step: } p(x_t|y_{0:t}) &= \frac{p(y_t|x_t)p(x_t|y_{0:t-1})}{\int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t} \end{aligned} \quad (2.3)$$

where the recursion is initialized by the prior distribution $p(x_0|y_0) = p(x_0)$, the dynamics model $p(x_t|x_{t-1})$ is specified by Equation 2.1, and the likelihood model $p(y_t|x_t)$ is specified by Equation 2.2.

Kalman Filter

The Kalman filter [BSF88] is one of the solutions to the Bayesian sequential estimation if both function $f(\cdot)$ in Equation 2.1 and $h(\cdot)$ in Equation 2.2 are linear and Gaussian. It is widely used in visual tracking [CH96, KCM03] and robotics [Kel94]. In order to handle the nonlinear and non-Gaussian situations, extensions have been made to the standard Kalman filter. The *Extended Kalman Filter* (EKF) [AM79] is one of the extensions which uses a first order Taylor series expansion of the nonlinear functions in Equation 2.1 and Equation 2.2 for the approximation. The EKF has been widely used in the field of robotics in the last decade [DNDW⁺00], especially in the sub-field of SLAM [SSC90]. The *Unscented Kalman Filter* (UKF) [vdMAAdFW00] has even higher accuracy. It approximates the posterior $p(x_t|y_{1:t-1})$ directly instead of approximating the nonlinear functions $f(\cdot)$ and $g(\cdot)$. In particular, it uses a deterministically selected sample set and pass it through the true nonlinear function to capture the true distribution. Because of its capability of handling nonlinear models and its efficiency in computation, it has been successfully used in tracking [CHR02].

Particle Filter

The particle filter gained its popularity because of its ability to handle highly nonlinear and non-Gaussian models in Bayesian filtering with a clear and neat numerical approximation. The key idea is to approximate the posterior distribution with a set of randomly sampled particles that have weights associated to them. The set of particles can be represented as $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$, where $x_t^{(i)}$ is the supporting point, $w_t^{(i)}$ is the weight, N_s is the number of sample points. The posterior distribution is approximated by iteratively propagating the particles with the proposal distribution and updating the weights. The details of particle filtering are covered in Chapter 3.

²First order Markov chain has the following independency assumption

$$\begin{aligned} x_t &\perp y_{0:t-1} | x_{t-1} \\ y_t &\perp y_{0:t-1} | x_t \end{aligned}$$

which means the current state is independent of the past observations given the previous state and the current observation only depends on the current latent state.

Nowadays, particle filtering has been widely used in computer vision and robotics. It was first introduced to visual tracking by Isard and Blake in [IB98], where it was known as the CONDENSATION method. Pérez et al. [HLCP03, VDP03] extended the particle filter framework to track multiple targets. Okuma et al. [OTdF⁺04] further extended it [VDP03] by incorporating a boosting detector [VJ04] into the particle filter initialization and the proposal distribution. Both [VDP03] and [OTdF⁺04] used a *Mixture of Particle Filters* (MPF) as the main framework to maintain multi-modality. The key contribution of MPF is to represent the overall posterior distribution with a M -component mixture model

$$p(x_t|y_{0:t}) = \sum_{m=1}^M \pi_{m,t} p_m(x_t|y_{0:t}) \quad (2.4)$$

where $\sum_{m=1}^M \pi_{m,t} = 1$, M is the number of targets or particle clouds. p_m represents the distribution of each target and evolves independently over time. The only interaction between targets is through $\pi_{m,t}$. Particle clouds are allowed to merge or split if different targets have significant overlap or overlapping targets split into two. Particles can be associated to different targets during merging or splitting while the overall distribution still remains the same. However, when new targets come into the scene, the MPF is not able to associate particles to them because the target initialization only takes place at the first step. Okuma et al. [OTdF⁺04] adopted the framework of MPF and improved it by using boosting detection to initialize targets. However, because the total number of particles in the MPF framework is fixed, part of the particles from each target are allocated to the new targets. As a result, if more and more new targets came into scene, the particles of each target would decrease and not be able to properly approximate the distribution. Another major disadvantage of the MPF framework is that it can not handle a variable number of targets and maintain their identities at the same time, because after the merge and split, there must be at least one target that loses its identity. Therefore, we track each target with a set of particles independently instead of the MPF framework in our system. A global mechanism is used to handle the birth and death of targets and the data association issue.

2.2.2 Data association

Data association arises from the task of multi-target tracking. The aim of data association is to find the correct correspondence among multiple observations $\{y_t^m\}_{m=1}^M$ and the existing tracks $\{x_t^n\}_{n=1}^N$. M and N can be different because there might be outliers or mis-detections. Standard Bayesian filtering does not handle observation-to-track assignments. However, the color-based particle filter tracker [VDP03] solves the problem implicitly because, according to Equation 3.9, the weight of each particle is updated only by evaluating the likelihood of color histogram at the location

specified by the particle. Data association becomes important in our application again because we adopted the idea of boosted particle filter [OTdF⁺04]. BPF uses a mixture of Gaussian that combines the boosting detections with the existing tracks to improve the proposal distribution shown in Equation 3.10. As there are multiple boosting detections and multiple tracks at each time step, the association problem arises again.

Gating is one of the basic techniques to eliminate very unlikely observation-to-track pairs. It defines a certain range around each existing track so that observations that fall outside the ranges are not considered to be associated to them. However, gating is still a preliminary procedure. More sophisticated data association techniques are required.

Linear optimization

Data association can be seen as an auction process where the tracks bid for the observations, or vice versa. Each observation-to-track pair will be associated with a corresponding cost or profit. The goal is to optimize a linear objective function under several linear constraints. Therefore, data association can be considered as a linear optimization problem. Blackman and Popoli give a detailed introduction of various optimization solutions in their book [BP99]. The auction algorithm [BP99] is found to be the most efficient method so far. However, the constraints it requires do not satisfy the requirements of our application. In Chapter 5, we model our application with a set of new constraints and introduce the solution to the new linear optimization setting. We adopt this approach because of three reasons. Firstly, it can handle the birth and death of targets, which is critical in our application. Secondly, the algorithm is easy to understand and implement. Finally, it requires very little computation and is extremely fast. The major drawback of this approach is its inability to handle error association. As it is not a probabilistic method, there is only one optimal solution at each step. Also, because the associations at each step are independent, error association can not be corrected at later stages.

Joint probability data association filter

The Joint Probability Data Association Filter (JPDAF) [BSF88] is one of the most widely accepted probabilistic approach to the data association problem. It can either be combined with Kalman filters [KCM03] or with particle filters [SBFC01, MB00]. The state space of JPDAF is a concatenation of all the states of the existing tracks: $X_t = \{x_{1,t}, \dots, x_{N,t}\}$. The observed variable is also a concatenation of all measurements: $Y_t = \{y_{1,t}, \dots, y_{M,t}\}$. By introducing an association indicator θ , the

joint posterior distribution can be written as

$$\begin{aligned}
P(X_t|Y_{1:t}) &= \int P(X_t, \theta|Y_{1:t})d\theta \\
&\propto \int P(Y_t|X_t, \theta)P(X_t, \theta|Y_{1:t-1})d\theta \\
&= \int P(Y_t|X_t, \theta)P(\theta|X_t, Y_{1:t-1})P(X_t|Y_{1:t-1})d\theta \quad (2.5)
\end{aligned}$$

It should be noted that, in most cases, the association is discrete and finite. Therefore, the integral can be replaced with a sum over all possible associations. With this association indicator θ , all the terms, especially the likelihood term, can be easily factorized and further computed conditioned on the association setting.

A problem arises if the total number of observations and tracks is large, because it results in a huge amount of enumeration and computation. It is even worse if it is combined with particle filtering because sampling in a high dimension space will make the efficiency of sequential Monte Carlo approximation decrease drastically. Another problem with JPDAF is that it does not handle the birth and death of targets explicitly, because the dimension of the joint state space is determined at the initialization step by the total number of targets. Therefore, it does not satisfy the requirement of our application.

Multiple hypothesis tracking

Multiple Hypothesis Tracking (MHT) is a deferred decision logic in which alternative data association hypotheses are formed whenever there are uncertainties about observation-to-track assignments. The algorithm is built on the assumption that uncertainties can be resolved at subsequent stages. MHT was first called Reid's algorithm. An efficient implementation of Reid's approach is later presented by Cox and Hingorani [CH96]. Multiple hypotheses are maintained at each step. New hypotheses are generated according to the previous hypotheses. Each hypothesis in the previous stage will generate one or more children. Therefore, a pruning technique is required so that only the N best hypotheses will be selected. Otherwise, after several steps, the number of total hypotheses will easily increase enormously. The selection process is similar to the linear optimization mentioned in the previous section, however, instead of maintaining the optimal solution only, it finds the N best solutions. All the hypotheses are evaluated based on their scores, which is the sum of all the component observation-to-tracks association.

$$L_{H_j} = \sum_{T_i \in H_j} L_{T_i} \quad (2.6)$$

where L_{T_i} is the score of each observation-to-track assignment and T_i are tracks that belong to hypothesis H_j . It can be evaluated using the likelihood $p(y_m|x_n)$, or any other variations.

Cox’s implementation of Reid’s algorithm [Rei79] is a track oriented solution to MHT. Instead of maintaining a set of hypotheses, it creates a set of tracks. Hypotheses are generated at each step after the tracks are created. Pruning is performed both in the creation of tracks and hypotheses. Details can be found in [CH96]. This implementation is much more efficient because the track-oriented approach does not generate redundant tracks while the hypothesis oriented approach has many redundant observation-to-track assignments in different hypothesis. However, it is still not quite suitable for particle filter trackers. As uncertainty increases, more and more possible potential tracks need to be created. Even if there are at most 15 targets on the rink in our application, the total number of tracks grows exponentially with the increasing possible associations. As each created new track needs a set of particles to support the evolution of track, the required storage space for the particles grows exponentially as well. As a result, the computation for the propagation of all the particles becomes extremely expensive. Therefore, simply combining conventional MHT with particle filters is not practical. The experiments show that the linear optimization approach satisfies our accuracy requirement for data association. Therefore, the computationally more expensive MHT approach is not used.

Monte Carlo method for data association

Unlike the three approaches mentioned above, Monte Carlo methods try to sample in the association space rather than enumerating all possible assignments. *Rao-Blackwellized Particle Filtering* (RBPF) is one of the Monte Carlo methods that are widely used in the literature [SVL04, SFH03]. The key idea is to sample for the data association while the remaining tracking part is computed in an analytical way, for example, the Kalman filter. An in-depth explanation of RBPF can be found in [DdFMR00].

Because the Monte Carlo methods sample in the state space, the “curse of dimensionality” is the major problem if the total number of targets is large. Pérez et al. tried to solve the problem by developing three algorithms: *Monte Carlo Joint Probability Data Association Filter* (MC-JPDF), *Sequential Sampling Particle Filter* (SSPF) and *Independent Partition Particle Filter* (IPPF) in [VGP05]. The fundamental motivation underlying these three algorithms is to factorize the joint state space into individual components.

However, in visual tracking systems, detections at each stage can be considered as multiple signals generated by a single sensor. Therefore, the relationship between signals and sensors in visual tracking is not as significant as it is in robotics and other applications. Consequently, there will not be significant gain to use a state space model to represent the association and use Monte Carlo method for approximation. In addition, as we use the particle filter as basic framework for tracking,

sampling in both the filtering part and the data association part will decrease the efficiency and effectiveness of the particle filter.

2.2.3 Mutual exclusion

Mutual exclusion is important in multi-target tracking, especially when different targets overlap and occlude each other. Some exclusion rules are the basic constraints in data association. For example, one signal can not be generated by multiple targets and one target can generate multiple signals. When targets occlude each other, mutual exclusion plays a critical role for the evaluation of likelihood function. Sullivan et al. [SBIM01] address the global optimal solution to associate pixels to objects and background in a image with a Bayesian approach. It is important for multi-target tracking because it makes the likelihood evaluation of the targets accurate. MacCormick and Blake introduced a probabilistic exclusion principle for tracking multiple objects in [MB00]. MacCormick also introduced a Bayesian multiple-blob tracker [IM01], which explicitly address the issue of mutual exclusion in the 3D space. Wu et al. [WYH03] explicitly model overlapping to handle occlusion. However, all the approaches in [MB00, IM01, WYH03] require explicit modelling of the object shape. As is mentioned in Chapter 1, the non-rigidity of the hockey players in our application makes the shape modelling difficult. While the mutual exclusion is difficult to achieve, multiple cues are used to resolve the ambiguity during occlusion. Wu and Huang [WH01] introduced a co-inference tracking algorithm to utilize both color and shape cues to track single target. Pérez et al. [PVB04] fuse color, motion and sound cues to track multiple objects with a particle filter.

In our application, according to the law of inertia, it is easier to use the physical model to predict the motion of hockey players. Therefore, better modelling of the motion of the targets will help resolve the ambiguity. The stabilized trajectories of the targets are important for accurate modelling of dynamics. Shan et al. [SWTO04] have embedded *Mean-Shift* [Che95, CM02] into the particle filter for hand tracking. Their results show that the mean-shift improves the efficiency of the sampling in particle filter. The mean-shift algorithm is also used in our system to achieve the stabilization by biasing all particles to the true location of the target.

2.3 Summary

The survey in the literature shows that there is no existing system that can satisfy all the requirements of our application. In order to achieve the goal of our robust tracking system, we propose four major extensions on the previous work. Firstly, a better modelling of target dynamics is used to improve the dynamics model of the targets. Secondly, a global nearest neighbor data association method is developed to associate boosting detections with existing tracks for a better proposal distribution

in the particle filtering. Thirdly, the likelihood model is improved by a isotropic kernel brought by the mean-shift algorithm. Finally, mean shift is embedded into the particle filter framework to stabilize the tracking trajectory by biasing each particle. This is critical to both the dynamics model and likelihood model.

Chapter 3

Particle Filter for Multi-target Tracking

As is mentioned in the previous chapter, particle filtering performs well in Bayesian sequential estimation with non-linear, non-Gaussian target dynamics and observation model. In our application, the fast motion of hockey players is highly non-linear and non-Gaussian. In addition, because we employ the observation model in [PHVG02, VDP03, OTdF⁺04], which is an exponential function based on the Bhattacharyya coefficient between two color histograms, the likelihood function is non-linear and non-Gaussian as well. Moreover, particle filtering has been successfully applied to multi-target tracking [VDP03, HLCP03]. Therefore, the particle filter framework is the ideal model to be the basic skeleton of our tracking system. This chapter covers the basic theory of generic particle filters and the boosted particle filters. How the boosted particle filter is modified to fit in our multi-target tracking system will also be introduced.

3.1 Particle Filter

As is shown in Equation 2.3, in order to continue the recursion of the Bayesian sequential estimation, the integration in both the prediction and filtering steps need to be computed at every step. If both the dynamics model and the likelihood model are linear and Gaussian, it becomes the well known analytical Kalman Filter. The details of Kalman Filtering are beyond the scope of this thesis. An in depth introduction can be found in [BSF88]. For the analytically intractable cases, where both the dynamics model and likelihood model are non-linear and non-Gaussian, approximation techniques are required. The most popular numerical approximation technique for this problem is the Sequential Monte Carlo (SMC) method [DdFG01]. It is also known as Particle Filtering [AMGC01] or CONDENSATION method [IB98].

3.1.1 Monte Carlo simulation

The basic idea of Monte Carlo simulation is to represent the whole posterior distribution by a set of random samples with associated weights. For example, the posterior density function can be approximated as

$$p(x_t|y_{0:t}) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(x_t - x_t^{(i)}), \quad (3.1)$$

where N_s is the number of samples, $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$ is the set of particles drawn from the posterior distribution and their weights. Here the weights of all particles are equal to $\frac{1}{N_s}$ because the samples are directly drawn from the true distribution and normalized. Situations where the weights are not equal will be introduced later. $\delta(\cdot)$ denotes the Dirac delta function

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

Therefore, with the Monte Carlo simulation, any integral can be approximated by a discrete form

$$\mathbb{E}(f(x_t)) = \int f(x_t) p(x_t|y_{0:t}) dx_t \approx \frac{1}{N_s} \sum_{i=1}^{N_s} f(x_t^{(i)}). \quad (3.3)$$

According to the law of large numbers, when the number of samples approaches infinity, the approximations of the posterior distribution and the integral are equivalent to the true distribution and integral. Similarly, any query estimation computed from the samples and weights approaches the optimal Bayesian sequential estimate.

3.1.2 Sequential importance sampling

As it is often impossible to directly sample from the posterior density function, the particles are sampled from a known and easy-to-sample proposal distribution $q(x_t|y_{0:t})$ instead. Then, Equation 3.1 can be rewritten as

$$p(x_t|y_{0:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(x_t - x_t^{(i)}), \quad (3.4)$$

where

$$w_t^{(i)} \propto \frac{p(x_t^{(i)}|y_{0:t})}{q(x_t^{(i)}|y_{0:t})} \quad (3.5)$$

and

$$\sum_{i=1}^{N_s} w_t^{(i)} = 1. \quad (3.6)$$

The new sampling strategy is called *Importance Sampling* and the weights are called *Importance Weights*. As is shown in Equation 3.5, if the proposal distribution is the same as the posterior, the weights reduce to the same as in Equation 3.1.

By replacing the integral of the prediction step in Equation 2.3 with the importance sampling, the one step ahead prediction can be written as

$$p(x_t|y_{0:t-1}) \approx \sum_{i=1}^{N_s} p(x_t|x_{t-1}^{(i)})w_{t-1}^{(i)}. \quad (3.7)$$

It should be noted that the proposal density function just samples the current state from the previous one rather than drawing samples for the whole trajectory of the states because we assume the current state only depends on the previous one.

One step further, the current posterior distribution can be updated as

$$p(x_t|y_{0:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(x_t - x_t^{(i)}) \quad (3.8)$$

where

$$w_t^{(i)} \propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_{0:t})} w_{t-1}^{(i)}. \quad (3.9)$$

Here the term $\int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t$ is omitted for convenience because it is just a normalization constant. Eventually, the estimation evolves by sequentially updating the weights for each supporting particle and the procedure is named *Sequential Importance Sampling* (SIS).

3.1.3 Resampling

One inherent problem in the sequential importance sampling method is the degeneracy phenomenon. It is observed that after several iterations, all but one particle have close to zero weights, which means the majority of the computing is wasted on those particles with negligible weights. Even worse, it has been proved that the problem can not be avoided because the variance of the weights increases over time. Therefore, a resampling technique is required to ignore particles with very low weights and concentrate on the more promising ones, which take an important role in the approximation.

One of the most widely used resampling schemes, which is also used in our work, is named the *Sampling-importance resampling* (SIR) technique. The basic idea is to map a set of weighted particles $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$ to a set of equally weighted samples $\{\tilde{x}_t^{(i)}, N_s^{-1}\}_{i=1}^{N_s}$ through a discrete cumulative distribution of the original sample set. As a result, the original particles that have higher weights would have more replicates while those with small weights would have much less or even no replicates. Details can be found in [Gor94]. The whole process of particle filtering is shown in Figure 3.1 [vdMAdFW00].

Figure 3.1: The SIR process starts at time $t - 1$ with a set of equally weighted particles $\{\tilde{x}_{t-1}^{(i)}, N^{-1}\}_{i=1}^N$, which represent the distribution $p(x_{t-1}|y_{0:t-2})$. Importance weights are updated with information at time $t - 1$ to represent $p(x_{t-1}|y_{0:t-1})$ with $\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}_{i=1}^N$. Resampling is performed to make it a equally weighted particle set $\{x_{t-1}^{(i)}, N^{-1}\}_{i=1}^N$ again but still represent the same distribution. Finally, particles are propagated to $\{\tilde{x}_t^{(i)}, N^{-1}\}_{i=1}^N$ at time t by the proposal distribution to represent $p(x_t|y_{0:t-1})$ and iterations go forward [vdMAdFW00].

3.2 Boosted Particle Filter

In the standard particle filter, there are two major problems that need to be solved: the initialization of the particle filter and proper design of the importance proposal. The Boosted Particle Filter (BPF), which is first introduced by Okumak et al. [OTdF⁺04], properly solves the two problems by incorporating the AdaBoost detection algorithm by Viola et al. [VJ04].

Most tracking systems that use particle filters (i.e., [PHVG02, VDP03, HLCP03]) initialize the particle filters by hand labelling to set the value for $p(x_0)$. The boosted particle filter uses AdaBoost detections for automatic initialization of the starting states.

Figure 3.2: The final proposal is in the form of a mixture of Gaussian which combines the prior and the boosting detection [OTdF⁺04].

A well designed importance proposal distribution should be as close as possible to the posterior distribution. The proposal distribution should be able to shift the particles to the regions with high likelihood if the mode of the prior distribution is far away from the mode of likelihood distribution. The boosted particle filter assumes Gaussian distribution for the dynamics and superimposes Gaussian distributions centered at the Adaboost detection modes as well. Then, it combines both parts to construct a proposal distribution with a mixture of Gaussian model:

$$q_B^*(x_t|x_{t-1}, y_{0:t}) = \alpha q_{ada}(x_t|y_t) + (1 - \alpha)p(x_t|x_{t-1}), \quad (3.10)$$

where α is the parameter which is dynamically updated according to the overlap between the Gaussian distribution of boosting detection and the dynamics prior. If there is no overlap, α is set to zero so that the boosting detection has no effect on the proposal distribution. The value of α can be fine tuned through experiments. In our application, it is set to 0.5. The mixture of Gaussians proposal can be best described by Figure 3.2 [OTdF⁺04].

3.3 Our Improvements

Particle filtering has proved in [IB98, HLCP03] to be successful in tracking single or multiple targets with nonlinear and non-Gaussian dynamics and likelihood models. In the original work of boosted particle filtering, a mixture of particle filters [VDP03]

was adopted and augmented by the AdaBoost detection to automatically track a variable number of targets. However, as in the MPF, the BPF only uses a fixed number of particles for all the targets. This causes critical problems when the total number of targets keeps increasing because new tracks have to steal particles from existing tracks so that the number of remaining particles for each target decreases until it is insufficient to correctly approximate the distribution. Moreover, when occlusion happens, the BPF merges overlapping particle clouds and splits them when they fall apart. Obviously, such a mechanism will lose the unique identity of each target after the occlusion. Therefore, our work only retains the basic idea of the original boosted particle filter for individual target and maintains multiple boosted particle filters to track multiple targets. Therefore, it solves both problems stated above. On one hand, in the new structure, new tracks are created by assigning a new particle set for each of them rather than stealing particles from existing targets. Therefore, the new structure can dynamically handle an increasing number of targets without affecting the approximation accuracy of the existing tracks. On the other hand, because all the tracks are maintained and removed independently by each particle set, there will be no merge or split during occlusion. Both of the new features are critical in maintaining correct identities of all targets in our multi-target tracking task.

Chapter 4

Target Dynamics Modelling

As is mentioned in Section 1.3.5, in visual tracking systems, mutual occlusion between different targets is one of the major factors that cause the tracker to lose the identity of different targets. Accurate modelling of the target dynamics can improve the prediction of the locations of the targets while visual support is insufficient due to occlusion. However, because of the camera motion in our application, the image coordinate system changes over time. Therefore, target motion modelling and prediction in the image coordinates are difficult. Zhao et al. [ZN04] uses camera calibration to provide a transformation from the predefined ground plane in the scene to the image. It helps transform the invariant 3D model of the target to the 2D model in the image to assist human segmentation and further motion modelling. We adopt the approach by Okuma et al. [OLL04] to map the locations of the targets from the image coordinates to the standard rink coordinate system which is consistent over time. Therefore, according to the physical law of inertia, the motions of the players in hockey games are easier to predict with a constant velocity autoregressive model.

4.1 Rectification

In this section, we will give a brief introduction to the concept of homography and how it acts as an invertible deterministic mapping between the image coordinates and the standard rink coordinates. It is important that homography is an invertible transformation because mapping back and forth between the rink and the image coordinate system is frequent during the tracking process.

4.1.1 Homography

Homography, which is also known as *projectivity* or *collineation*, is defined by Hartley and Zisserman in [HZ00] as follows:

Definition. A *projectivity* is an invertible mapping h from $\mathbb{P}^2$ to itself such that

three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lie on the same line if and only if $h(\mathbf{x}_1)$, $h(\mathbf{x}_2)$ and $h(\mathbf{x}_3)$ also lie on the same line.

Images recorded by cameras are 2D projections of the real world through lenses. For any plane in the world, its images from a camera, which can pan, tilt, zoom or even move, are exactly the projection described above because any line in the world is still a line in the images as long as there is no noticeable lens distortion. As the hockey players are always moving in the plane formed by the hockey rink, they are in the same plane both in the real world and the image space. Therefore, it is possible to project their locations between the two planes.

The mathematical way to accomplish such projection is possible because of the the following theorem:

Theorem. *A mapping $h: \mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix $\mathbf{H}$ such that for any point in $\mathbb{P}^2$ represented by a vector $\mathbf{x}$ it is true that $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$.*

Here, the vector $\mathbf{x}$ can be written as $(x, y, w)^\top$, which is a *homogenous* representation of a point in the plane. The corresponding inhomogeneous representation of the point in Euclidean 2D space is written as $(x/w, y/w)^\top$, where x/w and y/w are the x - and y -coordinates of the point. According to the theorem, given the coordinates of a pair of matching points $\{(x, y), (x', y')\}$, in two planes, the transformation can be written as

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \quad (4.1)$$

where $\mathbf{x}'$, $\mathbf{x}$ and $\mathbf{H}$ are defined as:

$$\mathbf{x}' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (4.2)$$

It needs to be noted that for all point pairs $\{\mathbf{x}'_i, \mathbf{x}_i\}$ from the two planes, Equation 4.1 holds for the same matrix $\mathbf{H}$. In order to compute $\mathbf{H}$, at least three points are needed because there are 9 unknown entries. However, the homography between two video frames can be defined with less parameters if the camera rotates about its optical center, which is the case in our application. The camera at each time step can be parameterized by 3 rotation angles $\theta = [\theta_1, \theta_2, \theta_3]$ with focal length f . Then, the pairwise homography $\mathbf{H}_{ij}$ from frame j to frame i can be parameterized by the parameters of the camera in the two time steps, which is 8 in all. Details of how to form the homography with the 8 parameters can be found in [BL03]. With this technique, it is possible to build a whole panorama for the scene, including the audience, from a sequence of video frames. It allows many other techniques to assist in improving the tracking result. For example, background subtraction can be performed to eliminate interference from background clutter.

In [OLL04], a set of point pairs are labelled to compute the initial video frame to rink homography H_0^{fr} . After the initialization, the frame to rink homography at each step is automatically computed in a recursive way

$$H_t^{fr} = H_{t-1}^{fr} \times H_{t-1,t}^{ff}{}^{-1} \quad (4.3)$$

where $H_{t-1,t}^{ff}$ is the transformation matrix from the previous frame to the current one. Different from the initialization, the point pairs are automatically selected by using the KLT feature tracker to match the extracted KLT features from consecutive video frames. In order to improve the accuracy of the computed homography, a template-based correction step is added to further fit the predefined key points with the standard rink model. Therefore, the algorithm can automatically and accurately map a long sequence of video frames to the rink. The whole process is explained in detail in [OLL04].

Notice that this is an online algorithm, which perfectly satisfies the online requirement of the particle filter tracker. Another important property of the homography is that it is an invertible transformation, which makes it easy to map the points in the images and rink in both directions in the following way with the same transformation matrix H_t^{fr} .

$$\begin{aligned} \text{frame to rink: } \mathbf{x}_t^r &= H_t^{fr} \mathbf{x}_t^f \\ \text{rink to frame: } \mathbf{x}_t^f &= H_t^{fr-1} \mathbf{x}_t^r \end{aligned} \quad (4.4)$$

where $\mathbf{x}_t^r$ and $\mathbf{x}_t^f$ are the points in the rink and video frames respectively. Figure 4.1 shows how the video frames are mapped to the standard rink with the homography.

4.1.2 Multi-layer state space model

With the frame to rink homography, it is possible to perform the particle filter tracker only in the rink coordinate system. However, the locations of the targets in the image coordinates are still required because, while evaluating the likelihood function in Equation 3.9, the locations of the particles in the image coordinates are still needed to specify the regions where the color histograms are extracted. Therefore, one more layer, which represents the hidden states in the image coordinates, is added into the original state space model. The graphical model of the new state space model is shown in Figure 4.2.

According to the graphical model, there are two layers of latent variables. On the top level, the variables contain locations of the targets in the rink coordinates. They are the true state variables in the particle filtering process. The middle layer states contain the locations of the targets in the image coordinates. The locations in the states between the top two layers have deterministic mapping function which is the same as Equation 4.4. It should be noted that all state variables in the middle

Figure 4.1: This shows how video frames are mapped to the rink. The top left image is the input video frame; the top right is the standard rink model; the bottom image shows how the image looks after being mapped onto the rink.

layer do not have any arrows between them, which means they are independent of each other. The state dynamics only applies to the top level state variables. The middle level acts as a communication tunnel to bridge the true latent variable and the observed variables, which are in the bottom layer.

In our application, there are two kinds of observations. One is the boosting detection and the other is the color histogram likelihood. At each time step, boosting detections provide a set of detected targets with their locations in the video frame and their scales. Therefore, each target can be represented by a bounding box with a predefined edge length ratio. As is shown in the bottom image in Figure 4.1, players are significantly distorted while only the feet of the players are on the right position relative to the rink because homography only guarantees accurate mapping between points from two planes. Here, we assume that the bounding box fits the target tightly. Therefore, we represent the location of each player in the image, $\mathbf{x}_{k,t}^f$, with the middle point on the bottom edge of the bounding box as is shown in Figure 4.3. Only these points will be mapped to the rink. At the initialization step, these transformed locations are used to create particle filter trackers. At later stages, they are used for the data association between boosting detections and the tracks or the creation of incoming new tracks. There are also situations that require mapping

Figure 4.2: Graphical model representation of the three-layer structure of the state space model.

locations on the rink back to the image coordinates. When the likelihood function needs to be evaluated, the positions of particles in the rink are mapped back to the video frames. They are assumed to be the positions of the players. Therefore, with this location value and the scale, a rectangular bounding box can be created. It is just the inverse of the operations with which the boosting detections are mapped to the rinks. Color histograms are extracted from the regions specified by the bounding boxes and compared with the reference color histograms, which are created at the initialization step.

4.2 Autoregressive Dynamics Model

An autoregressive process is a time series modelling strategy which takes into account the historical data to predict the current state value. In this model, the current state $\mathbf{x}_t$ only depends on the previous states with a deterministic mapping function and a stochastic disturbance. If the deterministic mapping is linear and the current state only depends on the previous p states, the model can be written as

$$\mathbf{x}_t = \sum_{i=1}^p \alpha_i \mathbf{x}_{t-i} + Z_t \quad (4.5)$$

where $\{\alpha_i\}_{i=1}^p$ are the autoregression coefficients, and Z_t is the stochastic disturbance which is normally a white noise process.

The autoregression coefficients can either be learned or predefined in an ad-hoc way. North et al. [NBIR00] tried to determine the coefficients by learning the

Figure 4.3: All the black rectangles are the boosting detections. Black dots on the bottom edge of each rectangle represent the positions of the players on the rink.

patterns of different motion classes through a training process where perfect tracking could be achieved. On the other hand, Pèrez et al. [PHVG02] used a second order constant velocity model to predefine the coefficients. However, this only works if the camera is stationary. In the work by Okuma et al. [OTdF⁺04], the camera motion made it impossible to separate the motion of players. Therefore, it is difficult to predict the location of players with the constant velocity autoregressive model in the image coordinates. By mapping the locations of players from the video frame coordinate system to the standard rink coordinate system, the motions of the players on the rink are separated from the camera motion. Therefore, no matter how the camera pans, tilts or zooms, the motions of the players on the rink coordinate system will not be affected. In hockey games, because of the effect of inertia, a constant velocity model is more suitable to model the motion of the players. It is best described by the following second order autoregressive model

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{x}_{t-2} + C\mathcal{N}(0, \Sigma) \quad (4.6)$$

where $\{A, B, C\}$ are the autoregression coefficients, $\mathcal{N}(0, \Sigma)$ is a Gaussian noise with zero mean and standard deviation of 1.

Unlike [PHVG02, OTdF⁺04], state $\mathbf{x}_t$ in our model only contains the location of the player in the x - and y -coordinates in the rink rather than including the scale of player as well because experiments show that incorrect sampling of player scales is more difficult to undo at later stages of tracking in this constant velocity model. Therefore, we use mean shift, which will be introduced in Chapter 6, instead to find the most appropriate scale at each step. Therefore, the states are defined as

$\mathbf{x}_t = (x_t, y_t)^\top$ and the coefficients are defined as

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad B = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad C = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \quad (4.7)$$

where A and B are standard for all second order constant velocity model, C can be fine tuned through experiment because it defines the standard deviation of the noise. In our application, it is defined to allow 3-pixel deviation on both x - and y -coordinates of the rink.

Chapter 5

Data Association

Chapter 2 gives a general introduction to data association and compares several typical approaches in the literature. In this chapter, we discuss in detail the gating technique and the solution to the global nearest neighbor (GNN) approach that solves the association problem in the application we investigate.

In our application, the data association does not take place in a conventional manner because there are two different kinds of observations in the framework. One is the color histogram and the other is the boosting detections. In a standard Bayesian filtering framework, data association is performed to pair the observations and tracks for the evaluation of the likelihood function $p(y_t^m | x_t^n)$. The particle filter framework in our application handles this level of data association in an implicit way because color histograms are extracted from the regions specified by particles. It means observations and tracks are no longer independent because observations are conditioned on the particles. However, it is still necessary to determine the association of the pixels within the specified region because the region might include pixels that do not belong to the corresponding target. However, in practice, as we do not have any concrete shape model of the targets, it is impossible to accomplish this level of data association.

Because the boosting detections are used to improve the proposal distribution in particle filters as in shown in Equation 3.10, we perform data association at this level to assign boosting detections to the existing tracks. Therefore, the proposal distribution can lead the sampling to more promising regions.

5.1 Gating

Gating is a technique to eliminate very unlikely observation-to-track assignments. At each time step during the tracking process, a certain region, also named as gate, is defined around the predicted position of each target. Observations that fall within the region are considered as candidates to update the track. Figure 5.1 depicts the concept of gating. As is shown in the figure, it is very likely that multiple

Figure 5.1: T1, T2, T3 are predicted target positions; O1, O2, O3, O4 are observations. O2 is assigned to T1; O1 is a candidate of T1, T2 and T3; O4 is a candidate of T2 and T3; O3 is not assigned to any existing tracks so that it is a candidate of being a new target.

observations fall in one gate or one observation falls in multiple gates. In addition, there are observations that do not belong to any gate. It remains to be decided later if they are outliers or new tracks. Therefore, gating is just a preliminary process to reduce the computation burden for further data association processing.

The sizes of the gates depend on the motion types of targets and the property of sensors. Because the camera is the only sensor for all targets, the effect from the sensor is homogeneous for all targets. Therefore, here we only consider the motion of targets. Targets that have high speed must have larger potential gates than slower targets. If the tracking were performed in image coordinates, the sizes of the gates of the targets could depend on the relative distances between targets and the camera. For example, players that are closer to the camera apparently have higher speed than those far away even if they are of the same speed in the real world. As a result, both target motion and camera motion affect the size of gate for each target, which makes the problem difficult. However, because of the three-layer state space model in our application, particle filtering is performed in the hockey rink coordinate system after compensating for camera motion. Consequently, the sizes of gates only depend on the motion of targets on the rink regardless of their locations because their speeds are evaluated homogeneously over the whole rink rather than being evaluated with respect to the camera position.

There are different shapes of gates including rectangles and ellipsoids. It is reasonable to use ellipsoids in our application to reflect the direction and magnitude of the velocity of target as is shown in Figure 5.2. However, as the tracking results

Figure 5.2: T1, T2 are predicted positions of targets; V1, V2 are vectors that indicate the velocities of the targets.

at each step might drift, the direction and magnitude computed from the previous steps could be very noisy and unstable. In addition, boosting detections also might be off the targets with several pixels. As a result, ellipsoidal gates might eliminate too many potential observation-to-track pairs that are correct. Therefore, in our application, we adopt the simplest circular gates which place equal deviation on all directions. The radii of the circles are determined by the maximum possible magnitude of target speed, which is consistent for all targets.

5.2 Linear Optimization

In [OTdF⁺04], only *nearest neighbor* association is used locally for the association, which means each track picks up the nearest boosting detection for update. Figure 5.1 shows that one observation can be in multiple gates. As a result, the same observation might be associated to multiple targets for update, which violates the basic constraint of data association that one observation can only be generated by one target. The *global nearest neighbor* method is one of the approaches to solve this problem. It maintains the single optimal solution at each step. The assignment problem can be best represented by an assignment matrix shown in Table 5.1. Each entry in the table is the cost or gain of pairing the corresponding track and observation. Assignments that are forbidden by gating are denoted by $\times$ in the corresponding entries. Observations that are forbidden by the gating to be associated to any track are considered as a new track.

Such assignment problems stem from economic theory and auction theory as well. Examples are assigning personnel to jobs and assigning goods to bidders. The objective of such problems is to minimize the cost or maximize the gain subject to a set of constraints. Given the assignment matrix shown in Table 5.1, the objective is to find a set $X = \{x_{ij}\}$ that maximize or minimize the objective function $C =$

Table 5.1: Assignment matrix for the assignment problem shown in Figure 5.1

Tracks	Observations			
	O1	O2	O3	O4
T1	a_{11}	a_{12}	$\times$	$\times$
T2	a_{21}	$\times$	$\times$	a_{24}
T3	a_{31}	$\times$	$\times$	a_{34}

$\sum_{i=1}^n \sum_{j=1}^m a_{ij}x_{ij}$ subject to the following constraints:

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1, \forall j \\ \sum_{j=1}^m x_{ij} &= 1, \forall i \end{aligned} \quad (5.1)$$

where x_{ij} are binary indicators

$$x_{ij} = \begin{cases} 1 & \text{if observation } j \text{ is assigned to track } i \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

Linear programming was initially used to solve this problem. Later on, it was found that the auction algorithm [BP99] is the most efficient method so far to reach the optimal solution or sub-optimal one without any practical difference. In the auction algorithm, normally, trackers are goods and observations are bidders. The objective is to maximize the global gain to satisfy all bidders. No matter whether the values of all entries in the assignment matrix are for maximizing gain or minimizing cost, it is always easy to convert them to satisfy a maximization problem by adding negative signs to all of them. The auction algorithm is an iterative process that consists of a bidding phase and an assignment phase. Iterations keep going until all bidders are satisfied. The detailed algorithm on a square association matrix, which has the same number of targets as observations, is available in [BP99].

The extended version of auction algorithm [Ber91] is able to solve the rectangular matrix problems with loosened constraints

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1, \forall j \\ \sum_{j=1}^m x_{ij} &\geq 1, \forall i \end{aligned} \quad (5.3)$$

These constraints allow one track to have multiple observations, which often happens in practice. However, in our application, it is very likely that some tracks do not have any observation due to the mis-detection of the boosting detector. Therefore, even if there are some observations within the gate of that track, it is still possible that none of the observations belongs to the track. Hence, the constraints are of the following form:

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1, \forall j \\ \sum_{j=1}^m x_{ij} &\geq 0, \forall i \end{aligned} \quad (5.4)$$

As a matter of fact, because x_{ij} are binary, the second constraint in Equation 5.4 is always satisfied. To simplify the problem, the assignment matrix only has existing tracks and the observations that fall in at least one gate. Observations that are not in any gate are used to create new tracks separately. As a result, the problem is significantly simplified.

In our application, the values of all the entries in the assignment matrix are defined to be the distance between the observations and tracks. It is mentioned in [BP99] that the value of each entry can include both kinematic and attribute information to fuse different types of sensor signals. Color is one important potential attribute in our application. However, because the reference color histogram of each target is created only at the first step and not updated adaptively over time, there are some time steps at which the current color histogram of track i is more similar to the reference color model of track j than the current color histogram of track j itself. It will produce incorrect assignment scores especially when targets with similar reference color model are close to each other or even overlap. Therefore, only the distance is used as the evaluation.

Given the new assignment matrix and the constraints, the solution to this problem becomes straightforward. The goal is to minimize the objective function and the optimal solution is just a collection of observation-to-track pairs that have the minimal entry value in the corresponding column in the assignment matrix.

$$x_{i'j} = \begin{cases} 1 & \text{if } i' = \arg_i \min a_{ij} \quad \forall j \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

Chapter 6

Mean-Shift Embedded Particle Filter

The motivation of embedding the mean-shift algorithm into the particle filter framework of our tracking system is to stabilize the tracking result. It is important for the dynamic model of targets because stabilizing trajectories improves the accuracy of the computed velocity of targets, which is critical for improving the prediction of the location of targets. It is also important for the likelihood model because accurate prediction leads sampling to more promising areas so that the influence from background clutter and mutual occlusion will be reduced. As a result, the particles of one target, which are biased by mean-shift, will be more concentrated on the true location of the targets in the scene. In this chapter, we explain how the mean-shift algorithm works using the color features and how to embed it into our particle filtering process for robust tracking.

6.1 Color Model

Color-based tracking is successful in tracking non-rigid objects with unique colors. It produces robust tracking results even if the targets change shape or are partially occluded by other objects. We adopted the color model in [PHVG02, OTdF⁺04] to represent the hockey players we try to track in our application. The model is originally introduced by Comaniciu et al. [CRM00] for the mean-shift based non-rigid object tracking. In addition, because the color histogram based approach discards the spatial information of the targets, it might have problems when the background clutter has a similar representation to the targets of interest in the color space. Therefore, we adopt the multi-part color model from [PHVG02] with small variation to fit it into the kernel-based color model we use in our application.

The color histogram is created in the Hue-Saturation-Value (HSV) color space because it is robust to the illumination condition changes. The HSV histogram comprises $N = N_h \times N_s + N_v$ bins, which discretize the whole HSV space.

N_h, N_s, N_v are the number of bins in each dimension of the HSV color space and are set to 10 equally in our experiment. The color vector of a pixel located at position k is denoted as $y(k) = [h(k), s(k), v(k)]^\top$ and each entry is mapped to an appropriate bin in the corresponding dimension. The bin representation of the color vector is denoted as $[b_h(k), b_s(k), b_v(k)]^\top$. The bin vector is further mapped to a bin in the histogram with the following index function.

$$b(k) = \begin{cases} b_s(k) \times N_h + b_h(k) & \text{if } s(k) \geq 0.1 \text{ and } v(k) \geq 0.2 \\ b_v(k) + N_h \times N_s & \text{otherwise} \end{cases} \quad (6.1)$$

where $b(k)$ is the index of pixel in the histogram. The saturation and value of a pixel is set to be above 0.1 and 0.2 respectively because the hue information is only reliable under that condition. Pixels that are too dark or unsaturated are classified only according to their value. With this bin indexing function, the color histogram of the candidate region $R(\mathbf{x}_t)$ around the location $\mathbf{x}_t$ at time t is denoted as $Q(\mathbf{x}_t) = \{q(n; \mathbf{x}_t)\}_{n=1, \dots, N}$, where

$$q(n; \mathbf{x}_t) = C \sum_{k \in R(\mathbf{x}_t)} \delta[b(k) - n] \quad (6.2)$$

$$\sum_{n=1}^N q(n; \mathbf{x}_t) = 1$$

where δ is the Kronecker delta function, C is a normalization constant, k is any pixel within the region $R(\mathbf{x}_t)$, $R(\mathbf{x}_t)$ is a 2D region centered at location $\mathbf{x}_t$, which can either be rectangular or ellipsoidal. By normalizing the color histogram, $Q(\mathbf{x}_t)$ becomes a discrete probabilistic distribution.

The color-based tracking searches for a candidate region whose histogram is the most similar to the one of a reference target model. The target model color histogram and target candidate histogram are denoted as follows:

$$\begin{aligned} \text{target model:} \quad & Q^* = q^*(n; \mathbf{x}_0)_{n=1 \dots N} \quad \sum_{n=1}^N q^*(n; \mathbf{x}_0) = 1 \\ \text{target candidate:} \quad & Q(\mathbf{x}_t) = q(n; \mathbf{x}_t)_{n=1 \dots N} \quad \sum_{n=1}^N q(n; \mathbf{x}_t) = 1 \end{aligned} \quad (6.3)$$

where Q^* is constructed at the initialization step. The Bhattacharyya coefficient [Kai67] is adopted to represent the similarity between two sample histograms as

$$\hat{\rho}(\mathbf{x}_t) \equiv \rho[Q(\mathbf{x}_t), Q^*] = \sum_{n=1}^N \sqrt{q(n; \mathbf{x}_t) q^*(n; \mathbf{x}_0)} \quad (6.4)$$

The distance between two histograms is represented as

$$d(\mathbf{x}_t, \mathbf{x}_0) = \sqrt{1 - \rho[Q(\mathbf{x}_t), Q^*]} \quad (6.5)$$

There are three major reasons to use Bhattacharyya coefficient in the similarity function. Firstly, it has a straightforward geometric interpretation because it is the cosine of two vectors. Secondly, $d(\mathbf{x}_t)$ is a *metric*. Finally, because it uses a

normalized discrete distribution as evaluation variable, it is invariant to the scale of the target.

Given the distance function, the likelihood distribution can be computed by passing the distance into an exponential function:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{-\lambda d^2(\mathbf{x}_t, \mathbf{x}_0)} \quad (6.6)$$

where $\lambda = 20$ is determined through experiment.

Multi-part color model splits the region $R(\mathbf{x}_t)$ into r parts so that $R(\mathbf{x}_t) = \sum_{j=1}^r R_j(\mathbf{x}_t)$. The color histogram of each part is constructed independently. In [PHVG02], the likelihood is formulated as

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{-\lambda \sum_{j=1}^r d_j^2(\mathbf{x}_t, \mathbf{x}_0)} \quad (6.7)$$

where $d_j^2(\mathbf{x}_t, \mathbf{x}_0)$ is the distance between the two corresponding parts in the target model and the target candidate. In our work, the mean-shift algorithm will be embedded into the particle filter framework to bias the particles according to the color feature. If the distances are evaluated on the part basis, the mean-shift algorithm is very likely to move them in different directions, which breaks the integrity of the whole target. Therefore, in our work, the color histograms of all parts are concatenated in parallel to form a new color histogram and the distance is evaluated according to the integral histogram. As a result, the concatenated histogram not only encodes the layout of all parts but also guarantees the integrity of the target candidate during the mean-shift procedure. The likelihood function is evaluated in the same way as Equation 6.6. Figure 6.1 shows the comparison of two multi-part histograms of players in two different uniforms. Here, the target region is divided vertically into two parts to represent the upper and lower body of the player. Each histogram is a concatenation of the upper and lower parts.

Figure 6.2 shows how significantly the color histogram of the same target changes over time with respect to the reference model extracted at the first frame. The comparison indicates that it remains a challenging problem to find an optimal model for the likelihood evaluation. Adaptively changing the reference color model with the current color histogram of the target is one possible approach. However, as there is no background subtraction in our system, the bounding box of the tracker may include pixels from the background clutter and make the color model unstable. Drift of the tracker and the mutual occlusion will also introduce undesired pixels from background or other targets, which make the color model stable. Therefore we give up the attempt to use any adaptive color model. Instead, we incorporate both the mean-shift based stabilization and strong dynamics model of the targets to assist the color histogram based likelihood model and improve the overall tracking performance of our system.

Figure 6.1: Multi-part color histogram of two players from two different teams.

6.2 Mean-Shift

Mean-shift is a nonparametric statistical method which seeks the mode of a density distribution in an iterative procedure. It was first generalized and analyzed by Cheng in [Che95] and later developed by Comaniciu et al. in [CM02]. It can be seen as a clustering technique, of which the k -means clustering algorithm is a special case. It is also a mode seeking procedure on a surface created by a "shadow" kernel in the feature space. Therefore, it is popular and successful in many applications including image segmentation [CM02] and object tracking [CRM03, Bra98, Col03]. Because of its ability to analyze a discrete feature space, it is a perfect tool to analyze color histogram features in our application.

The objective of the mean-shift algorithm is to iteratively compute an offset from the current location $\mathbf{x}$ to a new location $\mathbf{x}' = \mathbf{x} + \Delta_{\mathbf{x}}$ according to the following relation

$$\Delta_{\mathbf{x}} = \frac{\sum_{i=1}^M a_i w(a_i) k\left(\left\|\frac{a_i - \mathbf{x}}{h}\right\|^2\right)}{\sum_{i=1}^M w(a_i) k\left(\left\|\frac{a_i - \mathbf{x}}{h}\right\|^2\right)} - \mathbf{x} \quad (6.8)$$

where $\{a_i\}_{i=1}^M$ are points within the region $R(\mathbf{x})$ around the current location $\mathbf{x}$, $w(a_i)$ is the weight associated to each pixel a_i , and $k(x)$ is a kernel profile of kernel K that can be written in terms of a profile function $k : [0, \infty) \rightarrow R$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$. According to [Che95], the kernel profile $k(x)$ should be nonnegative, nonincreasing, piecewise continuous, and $\int_0^\infty k(r) dr < \infty$.

Mean-shift is a mode seeking algorithm because the theory guarantees that

Figure 6.2: This figure shows how significantly the color histogram of the same target changes over time.

the mean-shift offset $\Delta_{\mathbf{x}}$ at location $\mathbf{x}$ is in the opposite direction of the gradient direction of the convolution surface

$$C(x) = \sum_{i=1}^M G(a_i - \mathbf{x}) w(a_i) \quad (6.9)$$

where kernel G is called the shadow of kernel K and satisfies the following relationship

$$g'(r) = -ck(r) \quad (6.10)$$

where g is the kernel profile of G , g' denotes the derivative of kernel profile g , and c is a constant. Therefore, it is clear that mean-shift is a gradient ascent algorithm that converges to the mode of the convolution surface.

In order to utilize mean-shift to analyze a discrete density distribution, i.e. the color histogram, an isotropic kernel G with a convex and monotonically decreasing kernel profile $g(\mathbf{x})$ is superimposed onto the candidate region $R(\mathbf{x}_t)$. The target is represented by an ellipsoidal region bounded by a rectangle specified by the tracker. The target model is normalized to a unit circle by rescaling the row and column dimensions so that the shape of the ellipsoid does not influence the model representation. The normalized pixel locations in the region of the target model are denoted as $\{a_i^*\}_{i=1,\dots,M}$. With this kernel G , the target model element $q^*(n; \mathbf{x}_0)$ can be rewritten as follows:

$$q^*(n; \mathbf{x}_0) = C \sum_{i=1}^M g(\|a_i^*\|^2) \delta[b(a_i^*) - n] \quad (6.11)$$

where the region center is 0, C is the normalization constant to guarantee the probabilistic property of the discrete distribution. The nonincreasing kernel assigns lower weights on the pixels that are on the edge of the target. This is reasonable because most edges are blurry or might not actually belong to the target. They are much less reliable than those pixels that are closer to the center.

As in the target model, the normalized pixel locations of the target candidate centered at $\mathbf{x}_t$ are denoted as $\{a_i\}_{i=1,\dots,M_h}$. Each element of the target candidate histogram can be rewritten as

$$q(n; \mathbf{x}_t) = C_h \sum_{i=1}^{M_h} g\left(\left\|\frac{a_i - \mathbf{x}_t}{h}\right\|^2\right) \delta[b(a_i) - n] \quad (6.12)$$

where C_h is also a normalization constant that depends on h , and h is the bandwidth that determines the scale of the target candidate.

The similarity measure and the distance between the new target models and target candidates are the same as in Equation 6.4 and 6.5. The goal of a tracking process is to localize the position of the target in the current frame. In other words, it aims to search the neighborhood for a location so that the new candidate feature

extracted from that location has the minimum distance to the model feature. We assume that, at each time step, the search starts at location $\mathbf{x}$ and the normalized discrete distribution $Q(\mathbf{x})$ is already computed. $\mathbf{x}$ can be the target location in the last time step or any kind of location specified by the system. The goal is to iteratively update $\mathbf{x}'$ to reach the mode location. According to Equation 6.5, minimizing the distance is equivalent to maximizing the Bhattacharyya coefficient $\rho[Q(\mathbf{x}'), Q^*]$. Here, we assume that the new candidate feature does not have significant change from the feature at the starting location, which is true in most situations in practice. With this assumption, we can use the Taylor expansion around the value $Q(\mathbf{x})$ to have a reasonably accurate linear approximation of the Bhattacharyya coefficient

$$\rho[Q(\mathbf{x}'), Q^*] \approx \frac{1}{2} \sum_{n=1}^N \sqrt{q(n; \mathbf{x}) q^*(n; \mathbf{x}_0)} + \frac{1}{2} \sum_{n=1}^N q(n; \mathbf{x}') \sqrt{\frac{q^*(n; \mathbf{x}_0)}{q(n; \mathbf{x})}}. \quad (6.13)$$

Substitute $q(n; \mathbf{x}')$ with its kernel representation in Equation 6.12, the above equation can be rewritten as

$$\rho[Q(\mathbf{x}'), Q^*] \approx \frac{1}{2} \sum_{n=1}^N \sqrt{q(n; \mathbf{x}) q^*(n; \mathbf{x}_0)} + \frac{C_h}{2} \sum_{i=1}^{M_h} w(a_i) g \left(\left\| \frac{a_i - \mathbf{x}'}{h} \right\|^2 \right) \quad (6.14)$$

where

$$w(a_i) = \sum_{n=1}^N \sqrt{\frac{q^*(n; \mathbf{x}_0)}{q(n; \mathbf{x})}} \delta[b(a_i) - n]. \quad (6.15)$$

According to Equation 6.14, the first term does not depend on $\mathbf{x}'$. Therefore, to maximize the whole equation, we maximize the second term. It is noticed that the second term has the same formulation as Equation 6.9, which means the mode of the density estimation can be reached by applying mean-shift procedure, which iteratively moves the current location $\mathbf{x}$ with offset Δ_t according to Equation 6.8.

In our application, the Epanechnikov profile is selected to be the profile function of kernel G

$$g(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (6.16)$$

where d is the dimension of the geometrical space of the kernel, c_d is the volume of the unit d -dimension sphere. Because kernel G in the second term in Equation 6.14 is the shadow kernel of kernel K in the mean-shift process in Equation 6.8, they satisfy the relationship specified in Equation 6.10. Because the derivative of Epanechnikov profile is constant, the mean-shift offset reduces to

$$\Delta_{\mathbf{x}} = \frac{\sum_{i=1}^{M_h} a_i w(a_i)}{\sum_{i=1}^{M_h} w(a_i)} - \mathbf{x}. \quad (6.17)$$

In conclusion, the mean-shift iterative procedure can be presented as follows.

- **Input** target model $Q^* = q^*(n; \mathbf{x}_0)_{n=1, \dots, N}$ and the start location $\mathbf{x}$
- **Initialize** new location $\mathbf{x}'$ so that $\mathbf{x}' = \mathbf{x}$
- **Do**
 - $\mathbf{x} = \mathbf{x}'$
 - At location $\mathbf{x}$, evaluate Bhattacharyya coefficient
$$\rho[Q(\mathbf{x}), Q^*] = \sum_{n=1}^N \sqrt{q(n; \mathbf{x})q^*(n; \mathbf{x}_0)}$$
 - For each normalized pixel location a_i within region $R(\mathbf{x})$, compute the weights using Equation 6.19.
 - Find the new location
$$\mathbf{x}' = \frac{\sum_{i=1}^{M_h} a_i w(a_i)}{\sum_{i=1}^{M_h} w(a_i)}$$
 - At new location $\mathbf{x}'$, evaluate Bhattacharyya coefficient
$$\rho[Q(\mathbf{x}'), Q^*] = \sum_{n=1}^N \sqrt{q(n; \mathbf{x}')q^*(n; \mathbf{x}_0)}$$
 - While $\rho[Q(\mathbf{x}'), Q^*] < \rho[Q(\mathbf{x}), Q^*]$
 - * $\mathbf{x}' = (\mathbf{x} + \mathbf{x}')/2$
 - * Evaluate $\rho[Q(\mathbf{x}'), Q^*]$
- **While** $\|\mathbf{x}' - \mathbf{x}\| \geq \epsilon^1$
- **Output:** Location $\mathbf{x}'$ that minimizes the distance $d(\mathbf{x}', \mathbf{x}_0)$

As mean-shift is able to seek the mode of the density distribution, it is also capable of stabilize the tracking result by shifting the tracking output to the location with the maximum similarity to the model in the feature space. The algorithm is exactly the same as the one shown above. The start location $\mathbf{x}$ for the stabilization is given by the output of the tracking system.

6.3 Adaptive Scaling

As is shown in Equation 6.12, the bandwidth h determines the scales of the target candidate. In our application, the scales of players in the video frames is determined by their relative distance to the camera and the zooming effect of the camera. Therefore, it is important to adaptively change the scale through out the tracking process. Wu et al. [Wu05] tried to stabilize the scales of the targets through a pyramid based template matching, which is a coarse-to-fine scale search process to

¹ ϵ selects the stop criteria for the mean-shift iteration. Any value is appropriate as long as it is small enough relative to the application. It is chosen to be 1 pixel in our application.

find the best scale that match the template. In addition, rather than using only one top match, it takes multiple top matches to generate a best estimate for a consistent scale to avoid sudden jump. As we do not have any template to match, we need to search for the best estimate of the scale on the fly through the mean-shift algorithm.

In [OTdF⁺04], target scale is one of the dimensions in the the state space. Experiments show that the sampling process might generate scales that have a drastic change from the previous scale. However, in the real situation, because of the temporal and spatial continuity, the scales of players seldom change drastically. One of the solutions is to separate the scale from the state space and evolve it independently for each particle. The adaptive strategy searches several bandwidths around the previous bandwidth h_{prev} to find the optimal bandwidth h_{opt} at current step. To reduce the computational cost, only three bandwidth options are computed in our application: h_{prev} , $h_{prev} - \Delta_h$, $h_{prev} + \Delta_h$. Taking into consideration of the scale continuity, Δ_h is defined to be $\Delta_h = 0.1h_{prev}$. The optimal bandwidth h_{opt} selects the bandwidth that outputs the minimum distance (or the maximum Bhattacharyya coefficient). The new bandwidth h_{new} is updated with the following smoothed adaptation strategy

$$h_{new} = \gamma h_{opt} + (1 - \gamma)h_{prev} \quad (6.18)$$

where γ is a smoothing factor that controls the adaptation speed to avoid drastic scale change. It is chosen to be 0.1 in our application.

By combining the mean-shift algorithm and the adaptive scaling strategy, the tracking results can be well stabilized. Figure 6.3 compares the the raw tracking output and the stabilized result. It shows that the scale of the original tracker often is too small so that it cannot capture the major body of the target. Also, it drifts off the targets in some time steps. Meanwhile, the mean-shift stabilized result is much more stable. The scale changes more smoothly and fits the bounding box onto the target more properly.

6.4 Mean-shift Embedded Particle Filter

Applying the mean-shift algorithm directly to the tracking output only gives one deterministic offset at each step. It might not be able to capture the true location of the targets due to background clutter or mutual occlusion between targets in the image. Embedding it into the particle filter framework brings uncertainty to the deterministic method so that the statistical property can improve the robustness of the algorithm. The key idea of the mean-shift embedded particle filtering is to insert one more operation into the conventional particle filter framework.

There are two places where we can insert the mean-shift biasing step. One is after the particles are propagated by proposal distribution and the other is the after the deterministic resampling step, which is introduced in Section 3.1.3. There

(a) Frame 24

(b) Frame 40

(c) Frame 63

(d) Frame 84

Figure 6.3: Comparison between the original track output and the mean-shift stabilized result. Left ones are the original track result and the right ones are the stabilized result.

is one fatal disadvantage that undermines the second option. Because mean-shift is only based on the similarity between the reference model and the candidate model in the feature space, it might bias particles to other similar targets that are around the target of interest or some background clutter that are similar to the target of interest in the feature space. As a result, those wrongly biased particles that are far away from true location of the targets will have low weights because of the penalty from the prior distribution. Without the resampling step, those low weight particles are still propagated in the next step and lead the proposal distribution to sample in the wrong region and the tracker quickly loses the targets. Figure 6.4 shows the tracking result of inserting the mean-shift after the resampling step in the particle filtering. It can be noticed that, in Subfigure (b)(e)(f), some particles are moved to some locations that are far away from the true target location or the other targets. Although they are assigned low weights, they keep on propagating without being penalized by the resampling. As a result, they increase the variance of the particle set of one target and cause the target to be eliminated in the next frame because of the high uncertainty in particle hypotheses.

In our application, the mean-shift operation biases all the particles right after the sampling from the mixture of Gaussians proposal distribution and before the resampling step in the particle filter framework. In our multi-level state space model, particles propagated by the proposal distribution are mapped back to the video frame coordinate system. Then, each particle is biased individually and independently by the mean-shift algorithm to seek the mode of the density distribution in the neighborhood of each particle. The locations of particles in the video frame coordinates are set to be the starting locations for all the independent mean-shift procedures. After the bias step, new samples are drawn from a Gaussian distribution which is centered at the biased particle with a predefined covariance matrix.

The mean-shift searching step combined with the old proposal distribution can be considered as a new proposal distribution $\check{q}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$. Mean-shift biases the samples $\{\hat{\mathbf{x}}_t^{(i)}\}_{i=1,\dots,N}$ that are propagated by the old proposal distribution to a new particle set $\{\tilde{\mathbf{x}}_t^{(i)}\}_{i=1,\dots,N}$. We denote mean-shift searching with function $\varphi(\cdot)$ such that $\tilde{\mathbf{x}}_t = \varphi(\hat{\mathbf{x}}_t)$. The new proposal, which combines the old proposal and the mean-shift bias, superimposes a Gaussian distribution on the biased particle $\tilde{\mathbf{x}}_t^{(i)}$ for sampling new particles. Therefore, the weight is updated as follows:

$$\check{w}_t^{(i)} \propto \frac{p(\mathbf{y}_t|\check{\mathbf{x}}_t^{(i)})p(\check{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\check{q}(\check{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)} w_{t-1}^{(i)} \quad (6.19)$$

where $\check{q}(\check{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = \mathcal{N}(\check{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_t^{(i)}, \Sigma)$. Here, Σ is a diagonal 2×2 matrix and the value of the two entries are chosen to be the same, which is 0.3, in our application. Note that we use a sample $\check{\mathbf{x}}_t^{(i)}$ instead of the biased particle $\tilde{\mathbf{x}}_t^{(i)}$. This ensures that the sequential importance sampler remains unbiased and valid.

(a) Frame 1

(b) Frame 19

(c) Frame 20

(d) Frame 22

(e) Frame 24

(f) Frame 25

(g) Frame 26

Figure 6.4: This figure shows the tracking result of the system in which the particles are biased by the mean-shift algorithm after the deterministic resampling step in particle filtering. The top frame shows the overall view of the tracking result at frame 1. The six frames below show the tracking result with a close-up view of the three players in the center region. Particles from the same target are represented by rectangles with same color.

The following pseudo-code depicts the overall structure of our tracking system, which includes all the contributions in our work.

- **Initialization:** $t = 0$
 - Map boosting detections from the video frame coordinates to the rink coordinates to get $\{\mathbf{x}_{k,0}\}_{k=1,\dots,M_0}$.
 - For $k = 1, \dots, M_0$, create particle set $\{\mathbf{x}_{k,0}^{(i)}; \frac{1}{N}\}_{i=1}^N$ by sampling from prior distribution $p(\mathbf{x}_{k,0})$.
 - $K_0 \leftarrow M_0$.
- **For** $t = 1, \dots, T$,
 1. Targets adding and removing
 - Remove D_t targets with large particle set variance.
 - Map boosting detections from the video frame to the rink.
 - Data association
 - * For $s = 1, \dots, S_t$, create particle set $\{\mathbf{x}_{K_{t-1}+s,t}^{(i)}; \frac{1}{N}\}_{i=1}^N$ by sampling from prior distribution $p(\mathbf{x}_{s,t})$ for each new target.
 - * Associate boosting detections to the existing K_{t-1} tracks to construct Gaussian mixture proposal distribution $q(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}, \mathbf{z}_{k,t})$, where $\mathbf{z}_{k,t}$ is boosting detection.
 - * $K_t \leftarrow K_{t-1} + S_t - D_t$.
 2. Importance sampling
 - For $k = 1, \dots, K_t$, $i = 1, \dots, N$, sample $\hat{\mathbf{x}}_{k,t}^{(i)} \sim q(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$.
 3. Mean-shift biasing
 - For $k = 1, \dots, K_t$, $i = 1, \dots, N$, bias the particles as $\tilde{\mathbf{x}}_{k,t}^{(i)} = \varphi(\hat{\mathbf{x}}_{k,t}^{(i)})$.
 - For $k = 1, \dots, K_t$, $i = 1, \dots, N$, sample $\check{\mathbf{x}}_{k,t}^{(i)} \sim \check{q}(\mathbf{x}_{k,t}|\tilde{\mathbf{x}}_{k,t}^{(i)})$.
 4. Weight update
 - For $k = 1, \dots, K_t$, map particle set $\{\check{\mathbf{x}}_{k,t}^{(i)}\}_{i=1}^N$ from rink to frame coordinates.
 - For $k = 1, \dots, K_t$, $i = 1, \dots, N$, update weights $\check{w}_{k,t}^{(i)}$ according to Equation 6.19.
 - Normalize weights.
 5. Deterministic resampling
 - Resample particles to get new sample set $\{\mathbf{x}_{k,t}^{(i)}\}_{i=1}^N$.
 - Update weights: $w_{k,t}^{(i)} = \frac{1}{N}$.
 6. Output
 - For $k = 1, \dots, K_t$, $E(\mathbf{x}_{k,t}) = \sum_{i=1}^N w_{k,t}^{(i)} \mathbf{x}_{k,t}^{(i)}$

Chapter 7

Experimental Results and Conclusion

As all the important components of our tracking system have been introduced and explained in detail, they are assembled to construct a robust tracking system that is capable of tracking multiple targets accurately regardless of background clutter, camera motion and mutual occlusion between targets. In this chapter, tracking results of our system will be presented and analyzed. They will be compared with the results from previous work to support the validity of our contributions. A summary of our work will address the strengths and weaknesses of our work together with some possible extensions for future work.

7.1 Results

Firstly, it should be noted that, in order to gather more visual information for the color model, the original video tape is resampled in our experiment to produce video frames with the size of 640×480 pixels while, in [OTdF⁺04], the tracking is performed in video sequences with the frame size of 320×240 pixels. It is necessary to mention the issue of interlacing. The interlacing results from the inherent scanning mechanism of TV. In a TV, the whole picture is not in the frame, but instead all the odd rows are shown first, then all the even rows. Therefore the complete picture is only created after every second field. This causes significant edge blur if the object has significant offset between consecutive frames. Figure 7.1 shows an example of the blur from interlace. This will affect the analysis of the local motion of the targets. However, it does not significantly affect our application because we only care about first order motion and we only use the color histogram to represent the target, which does not emphasize spatial information.

Figure 7.2 shows the initial configuration of our multi-target tracking system. Each target is labelled with a colored rectangle. In order to have a clear view of the tracking process, we present the tracking result with close-up view of the rectangular

Figure 7.1: This figure shows an example of the blur from interlacing.

Figure 7.2: This figure shows the overall view of the camera at the initial stage. The close-up view of the tracking results will be shown in Figure 7.3 by zooming into the rectangular region.

region in Figure 7.2. Figure 7.3 shows the close-up view of the tracking results which focus on the three targets in the region. Images on the left column in the figure are the tracking results of the system in [OTdF⁺04]. Those on the right column are from our system.

Notice in Subfigure (c) and (e) of Figure 7.3 that two tracks merge into one track in both frames because the MPF structure in [OTdF⁺04] merges particle sets when they have significantly overlap. Therefore, two targets are lost in the tracking process. In Subfigure (g), a new target is created because the boosting detection detects the target and it is not in the gate of any other existing tracks. It is also noticed that the tracker of the player in the bottom part of Subfigure (a) in Figure 7.3 migrates to another one, which was in the left part of Subfigure (a) in Frame 30, in Subfigure (g). In systems that do not have a strong prior dynamics model, such migration happens frequently during cross-over because, without a strong dynamics model, color likelihood dominates the weights of particles and forces them

Figure 7.3: This sequence is a close-up view of the rectangular region in Figure 7.2. The left column is the tracking results of the system in [OTdF+04]. The right column is the tracking results of our system. Different targets are labelled with rectangles of different colors.

to migrate onto targets with similar color features. However, the color model alone is not competent enough to distinguish different targets in our application because of the similarity between players and the mutual occlusion that makes the color model unreliable. The targets lose their identity after the cross-over in the BPF-based tracking system. However, in our tracking system, with the assistance of the prior dynamics model and the mean-shift stabilization mechanism, all the targets are accurately tracked and all their identities are well kept after the cross-over. In our system, particles do not easily migrate to nearby targets with similar color histograms because the dynamics model takes into account the trajectory of the target and assigns low probability to particles that are far away from the predicted location. Combining both the visual and motion information from the targets significantly improves the robustness of the tracking system.

In order to demonstrate the advantages brought by the new dynamics model, Figure 7.4 shows the tracking result of the system in which the camera motion is compensated and the dynamics model is replaced with the new second order autoregressive model. The particles in this experiment are not biased by the mean-shift algorithm and the scale is still one dimension in the state space. In the sampling stage of the particle filtering, particles are sampled from a proposal distribution which is a mixture of Gaussians. It superimposes Gaussian distribution on the dynamics model and the boosting detection results. The value of the entry in the covariance of the Gaussian distribution that corresponds to the scale controls the deviation of the scale. The larger the value, the more possible the scale has a sudden jump. Figure 7.4 shows the tracking results of the experiment. The output scale is an average of the scale value in all the particles. It can be noticed that the two improvements do improve the accuracy of the tracking result. However, it can also be noticed that the result is very sensitive to the value in Gaussian covariance matrix that corresponds to the scale. The left column in Figure 7.4 shows the result with the value of 0.02 in the covariance, which makes the scale of the players getting smaller quickly over time. The right column shows the result with the entry value of 0.01, which is more reasonable because of the temporal continuity. However, in the experiment with the more reasonable value, one tracker migrates onto another one and miss the target. This indicates that, even with a strong dynamics model, if the particles are led to wrong regions because of the large variance on the trajectory of the particle cloud, it is still possible that the particles are trapped by another target with similar color histogram. Although the result on the left shows correct identity of all tracks, their scales no longer reflect the true scale of the targets. Therefore, there is not enough evidence to completely trust the system without mean-shift bias. Separating the scale from the state space can reduce the dimension of the state space so that, with the same number of particles, the particles can capture the true distribution more accurately. Moreover, the mean-shift algorithm with adaptive scaling is able to evolve the scales in a more reasonable and effective way.

Figure 7.4: This figure shows the tracking result of the system that only has the camera motion compensated and the dynamics model changed but without mean-shift bias. The left column is the results of the system in which the deviation for the scale sampling is 0.02; the right column is with the deviation of 0.01.

Figure 7.5 shows the particle representation of the tracking results of our system. Although the particles are propagated in the rink coordinate system, they are mapped to the video frames for representation purposes. In the pseudo-code in Section 6.4, the evolution of particle sets in each iteration of propagation can be divided into three steps in our tracking system. The figure compares the difference between the particle sets after each step. The left column in the figure shows the particles that are directly sampled from the old mixture of Gaussian proposal. The middle column shows the particles that are biased by the mean-shift algorithm. The right column shows the resampled particles after the deterministic resampling stage in our particle filtering process. Generally, the mean-shift algorithm moves particles from different locations around the target to locations in the neighborhood that are most similar to the reference model in the color space. Therefore, particle sets appear more condensed after the mean-shift bias. The resampling technique duplicates more children for particles with higher weights and particles with much lower weights might not have any child. Therefore, the resulting particle sets become even more compact. It does not mean that the number of particles for one target is reduced. Instead, they just overlap with each other on a limited number of locations. The difference between Subfigure (h) and (i) in Figure 7.5 indicates that mean-shift might move particles to some other targets because of the similarity between the two targets in the color space. However, particles that are shifted to the wrong targets will be assigned low weights because of the regularization of the dynamics model of targets. As a result, those particles will have much fewer or no children after the resampling stage. For the same reason, particles that are biased to regions without any target, as are shown in Subfigure (b) and (c) in Figure 7.5, will be penalized as well. In summary, both the mean-shift algorithm and the dynamics model penalize erroneous particle hypotheses and improve the robustness of the overall tracking system.

Because the particles evolve in the rink coordinates in our system, it is easy to acquire the trajectories of the targets on the rink. Figure 7.6 shows the trajectories of the same three targets that are shown in Figure 7.3. Subfigure (a) shows the trajectories of the three players on the standard hockey rink. Different trajectories are represented with different shapes on the nodes. Subfigures (b)(c)(d)(e) are the key frames during the process. Only the three targets we focus on are labelled in the key frames for the purpose of clear presentation. However, in the real tracking results, all targets are well tracked and labelled. It is easy to judge from the key frames that the trajectories well demonstrate the actual locations of the three targets, and hence support the validity of the three-layer state space model described in Section 4.1.2.

Figure 7.7 shows the correct tracking result of our system during the period when partial occlusion happens. The color of each target is different from their color in Figure 7.2 because they are changed for easy representation here. Both the

Figure 7.5: This figure shows the particle representation of each target. The left column shows the particles before the mean-shift bias; the middle column shows the particles after the mean-shift bias; the right column shows the particles after the resampling. Each particle is represented with a rectangle and particles that belong to the same target are of the same color.

(a) Trajectories of three players.

(b) Frame 1

(c) Frame 39

(d) Frame 65

(e) Frame 97

Figure 7.6: This figure shows the trajectories of the three targets shown in Figure 7.3. (b)(c)(d)(e) show the key frames during the process. (b) is the starting frame and (e) is the ending frame.

(a) Frame 78

(b) Frame 79

(c) Frame 83

(d) Frame 88

(e) Frame 98

Figure 7.7: This figure shows the tracking result of the period when partial occlusion happens. (a) is the overall view of the tracking result at frame 78. (b)(c)(d)(e) show the close-up view of the four targets in the region specified in (a).

results shown in Figure 7.3 and Figure 7.7 provide evidence to support the fact that our new tracking system is able to accurately track multiple targets regardless of partial or complete occlusion.

7.2 Conclusion

In this thesis, we devote our endeavors to building a tracking system that is able to robustly track multiple targets and correctly maintain their identities regardless of background clutter, camera motions and mutual occlusion between targets. Although there has been extensive research in the field of multi-target tracking, it is still a tough journey to achieve the goal we set because of various visual challenges discussed in Section 1.3. Unreliable target representation, unpredictable camera motion, and frequent mutual occlusion between targets are the major challenges.

We adopt the boosted particle filter framework from Okuma et al. [OTdF⁺04] as the basic skeleton of our tracking system because it is successful in tracking a variable number of targets with fully automatic initialization. However, we replace the MPF structure in BPF with a set of independent particle filters so that particle clouds of different targets will not merge or split because the two actions result in the loss of targets. The new structure provides a simple mechanism to handle a varying number of targets by just adding or removing particle sets. The advantages of the new structure become more prominent when more and more targets enter the scene. In BPF framework, as more tracks are created, new targets steal more particles from existing tracks so that the average number of particles of each target decreases and becomes insufficient to accurately capture the overall distribution.

In addition, after analyzing the particle weight update mechanism shown in Equation 3.9, we try to improve all the three terms in the equation to build a better model for our system. Firstly, by using the rectification technique, the location of the targets can be mapped between the video frame and rink coordinates through homography. Therefore, camera motions can be compensated and the motion of targets are easier to model and predict in the rink coordinate system. With the camera motion compensated, a second order auto-regression model is further adopted to predict the location of targets by taking into account the history of the trajectories of the targets. Experiment results shown in Subfigure (h) and (i) in Figure 7.5 support the fact that the dynamics model plays an important role in eliminating incorrect particle hypotheses especially when the visual information is unreliable during occlusion. Secondly, a global nearest neighbor data association technique is employed to assign boosting detections to the existing tracks or create new tracks. It improves the proposal distribution during the mutual occlusion because the nearest neighbor association fails to reach global optimal association when detections are in multiple gates of tracks. Finally, the mean-shift algorithm is embedded into the particle filter framework to stabilize the trajectory of the targets

and improve the dynamics model prediction. On one hand, mean-shift superimposes a kernel onto the target patch in the image and improves the color histogram model by down weighting unreliable pixels on the edges of the targets. On the other hand, the mean-shift algorithm biases particles to new locations with high likelihood so that the variance of particle sets decrease significantly. Experiment results shown in Subfigure (d) and (e) in Figure 7.5 support the fact that mean-shift is able to bias particles back onto more promising regions rather than background.

With all the improvements stated above, the new tracking system is able to keep robust tracking on the targets under all the challenging situations. Experiment results presented in the last section show the accurate tracking results of our system and support the validity of our contributions.

7.3 Discussion and Future Extension

The experiments demonstrate accurate tracking results under all the tough visual challenges mentioned in Section 1.3 to support the conclusion that our contributions in the new tracking system do improve the robustness and accuracy of the system. However, there are still some weaknesses in our approach. Here we focus on two major problems: one is result from the heterogeneous distances in the video frame coordinates and the rink coordinates; the other is caused by the second order autoregressive dynamics model.

As is shown in Figure 7.8, the five big circles on the rink are of the same size while the two that are closer to the camera appears much larger than the middle one in Subfigure (a). It indicates that distances that are the same in the image coordinates might be different in the rink coordinates, and vice versa. It affects the mean-shift bias of the particles because it is performed in the image coordinates. It is possible that mean-shift bias two particles with the same distance from their origins in the image coordinates while the distances in the rink differs significantly. As the dynamics prior is evaluated in the rink coordinates, the Gaussian distribution assigns low probability to particles that are further away from the centroid. Therefore, although the biased particle appears eligible in the images, they are penalized by the prior. Therefore, one possible extension is to take into account the heterogeneous distance in both the two coordinate systems while evaluating the prior probability of the biased particles. The covariance of the Gaussian distribution should vary according to the distances between the targets and the camera. The further the targets, the larger the covariance should be.

Although the second order autoregressive dynamics model takes into account the history of the trajectory of the target, it does not handle noisy data properly. It uses the velocity computed during the last time interval as the current velocity because of the temporal continuity property. However, without any smoothing technique, noisy data in the trajectory might produce abnormal velocity so that

(a) Original video frame.

(b) Mapped video frame blended with the rink

Figure 7.8: This figure shows how the video frame is mapped onto the standard rink.

the current prediction that use this abnormal velocity leads to incorrect regions. It would affect the importance sampling as the the proposal distribution in our system is a mixture of Gaussian which comprises both the prior distribution and the boosting detection. It will not be easily corrected by the mean-shift algorithm because the dynamics prior assigns low probability to particles that are shifted further away from the centroid. If the sampling is misled to regions near another target with similar color histogram, the particles tend to migrate onto that target and keep on tracking. If the particles are propagated out of the scene, the target will be eliminated. One possible extension is to replace the second order autoregressive model with a normal constant velocity model that maintains the history of the velocity instead of the trajectory and adaptively updates the velocity in the way similar to the adaptive scaling shown in Section 6.3 to prevent drastic velocity change.

Another critical inherent disadvantage of our tracking system is brought by

the online algorithm we use. Our system is sub-optimal because it is only a two frame system that only keeps the record of the current state and previous state. The tracking can be improved if it is performed over the entire sequence or over predefined episodes—windows in time. The information from the future may either smooth the trajectories of the tracks or resolve the uncertainties in the past. Forward filtering and backward smoothing can be performed on the particle filter framework to smooth the trajectories. MHT can also be used to help eliminating error tracks given the data in a longer (more than two time steps) window in time.

Bibliography

- [AM79] B.D. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, New Jersey, 1979.
- [AMGC01] S. Arulampalam, S.R. Maskell, N.J. Gordon, and T. Clapp. A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–189, 2001.
- [Ber91] D.P. Bertsekas. *Linear Network Optimization: Algorithms and Codes*. The MIT Press, Cambridge, 1991.
- [BL03] M. Brown and D.G. Lowe. Recognising Panoramas. In *International Conference on Computer Vision*, pages 1218–1225, 2003.
- [BP99] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, 1999.
- [Bra98] G.R. Bradski. Real Time Face and Object Tracking as a Component of a Perceptual User Interface. In *Workshop on Application of Computer Vision*, pages 214–219, 1998.
- [BSF88] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic-Press, Boston, 1988.
- [CH96] I.J. Cox and S.L. Hingorani. An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.
- [Che95] Y.Z. Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [CHR02] Y. Chen, T.S. Huang, and Y. Rai. Parametric Contour Tracking Using Unscented Kalman Filter. In *International Conference on Image Processing*, volume III, pages 613–616, 2002.

- [CM02] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [Col03] R.T. Collins. Mean-shift blob tracking through scale space. In *International Conference on Computer Vision and Pattern Recognition*, volume II, pages 234–240, 2003.
- [CRM00] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *International Conference on Computer Vision and Pattern Recognition*, pages 2142–2149, 2000.
- [CRM03] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.
- [DdFG01] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [DdFMR00] A. Doucet, J.F.G. de Freitas, K.P. Murphy, and S.J. Russell. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Uncertainty in Artificial Intelligence*, pages 176–183, 2000.
- [DNCC01] M.W.M.G. Dissanayake, P. Newman, Durrant-Whyte H.F. Clark, S., and M. Csorba. A Solution to the Simultaneous Localisation and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [DNDW⁺00] M.W.M.G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba. An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building. In *International Symposium on Experimental Robotics*, pages 265–274, 2000.
- [EBMM03] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *International Conference on Computer Vision*, volume II, pages 726–733, 2003.
- [Gor94] N.J. Gordon. *Bayesian Methods for Tracking*. PhD thesis, Imperial College, University of London, 1994.
- [HLCP03] C. Hue, J.P. Le Cadre, and P. Pèrez. Tracking Multiple Objects with Particle Filtering. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 38, pages 313–318, 2003.

- [HS88] C. Harris and M.J. Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, pages 147–152, 1988.
- [HZ00] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [IB98] M. Isard and A. Blake. CONDENSATION-Conditional Density Propagation for Visual Tracking. *Internatinal Journal on Computer Vision*, 29(1):5–28, 1998.
- [IM01] M. Isard and J.P. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *Internatinal Conference on Computer Vision*, volume II, pages 34–41, 2001.
- [Kai67] T. Kailath. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Transactions on Communication Technology*, 15:52–60, 1967.
- [KCM03] J. Kang, I. Cohen, and G. Medioni. Soccer Player Tracking across Uncalibrated Camera Streams. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS) In Conjunction with ICCV*, pages 172–179, 2003.
- [KCM04] J. Kang, I. Cohen, and G. Medioni. Tracking Objects from Multiple Stationary and Moving Cameras. In *Asian Conference on Computer Vision*, 2004.
- [Kel94] A. Kelly. A 3D Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles. Technical Report CMU-RI-TR-94-19, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [Li04] F. Li. Analysis of Player Actions in Selected Hockey Game Situations. Master’s thesis, Department of Computer Science, University of British Columbia, 2004.
- [Low04] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Internatinal Journal on Computer Vision*, 60(2):91–110, 2004.
- [MB00] J.P. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *Internatinal Journal on Computer Vision*, 39(1):57–71, 2000.

- [MSZ04] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human Detection Based on a Probabilistic Assembly of Robust Part Detectors. In *European Conference on Computer Vision*, pages 69–82, 2004.
- [NBIR00] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and Classification of Complex Dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.
- [OLL04] K. Okuma, J.J. Little, and D.G. Lowe. Automatic Rectification of Long Image Sequences. In *Asian Conference on Computer Vision*, 2004.
- [OTdF⁺04] K. Okuma, A. Taleghani, J.F.G. de Freitas, J.J. Little, and D.G. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *European Conference on Computer Vision*, volume I, pages 28–39, 2004.
- [PHVG02] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *European Conference on Computer Vision*, volume I, pages 661–675, 2002.
- [POP98] C.P. Papageorgiou, M. Oren, and T. Poggio. A General Framework for Object Detection. In *International Conference on Computer Vision*, pages 555–562, 1998.
- [PVB04] P. Pérez, J. Vermaak, and A. Blake. Data Fusion for Visual Tracking with Particles. *Proceedings of the IEEE*, 92(3):495–513, 2004.
- [RBK98] H. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
- [Rei79] D.B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [RF03] D. Ramanan and D.A. Forsyth. Finding and Tracking People from the Bottom Up. In *International Conference on Computer Vision and Pattern Recognition*, volume II, pages 467–474, 2003.
- [SBFC01] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association. In *International Conference on Robotics and Automation*, pages 1665–1670, 2001.

- [SBIM01] J. Sullivan, A. Blake, M. Isard, and J.P. MacCormick. Bayesian Object Localisation in Images. *Internatinal Journal on Computer Vision*, 44(2):111–135, 2001.
- [SFH03] D. Schulz, D. Fox, and J. Hightower. People Tracking with Anonymous and ID-Sensors Using Rao-Blackwellized Particle Filters. In *International Joint Conference on Artificial Intelligence*, pages 921–928, 2003.
- [SSC90] R. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. pages 167–193, 1990.
- [SVL04] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-Blackwellized Monte Carlo Data Association for Multiple Target Tracking. In *International Conference on Information Fusion*, volume I, pages 583–590, 2004.
- [SWTO04] C. Shan, Y. Wei, T. Tan, and F. Ojardias. Real Time Hand Tracking by Combining Particle Filtering and Mean Shift. In *International Conference on Automatic Face and Gesture Recognition*, pages 669–674, 2004.
- [TK91] C. Tomasi and T. Kanade. Shape and Motion from Image Streams: A Factorization Method Part 3 - Detection and Tracking of Point Features. Technical Report CMU-CS-TR, 1991.
- [vdMAAdFW00] R. van der Merwe, Doucet A, J.F.G. de Freitas, and E. Wan. The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge, England, 2000.
- [VDP03] J. Vermaak, A. Doucet, and P. Pérez. Maintaining Multi-modality through Mixture Tracking. In *Internatinal Conference on Computer Vision*, volume II, pages 1110–1116, 2003.
- [VGP05] J. Vermaak, S.J. Godsill, and P. Pérez. Monte Carlo Filtering for Multi-Target Tracking and Data Association. *Accepted for publication in the IEEE Transactions on Aerospace and Electronic Systems*, 2005.
- [VJ04] P. Viola and M.J. Jones. Robust Real-Time Face Detection. *Internatinal Journal on Computer Vision*, 57(2):137–154, 2004.
- [VJS05] P. Viola, M.J. Jones, and D. Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. *Internatinal Journal on Computer Vision*, 63(2):153–161, 2005.

- [WH01] Y. Wu and T.S. Huang. A Co-inference Approach to Robust Visual Tracking. In *International Conference on Computer Vision*, volume II, pages 26–33, 2001.
- [Wu05] X. Wu. Template-based Action Recognition: Classifying Hockey Players' Movement. Master's thesis, Department of Computer Science, University of British Columbia, 2005.
- [WYH03] Y. Wu, T. Yu, and G. Hua. Tracking Appearances with Occlusions. In *International Conference on Computer Vision and Pattern Recognition*, volume I, pages 789–795, 2003.
- [YKA02] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.
- [ZN04] T. Zhao and R. Nevatia. Tracking Multiple Humans in Complex Situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1208–1221, 2004.