5-2014

A Computer-Aided Training (CAT) System for Short Track Speed Skating

Chenguang Liu
Utah State University

A COMPUTER-AIDED TRAINING (CAT) SYSTEM FOR SHORT TRACK SPEED SKATING

by

Chenguang Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Computer Science

Approved:

Dr. Heng-Da Cheng Major Professor	Dr. Lie Zhu Committee Member
Dr. Vicki Allan Committee Member	Dr. Curtis Dyreson Committee Member
Dr. Haitao Wang Committee Member	Dr. Mark McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNVIVERSITY
Logan, Utah

2014

ABSTRACT

A Computer-aided Training (CAT) System for Short Track Speed Skating

by

Chenguang Liu, Doctor of Philosophy

Utah State University, 2014

Major Professor: Dr. Heng-Da Cheng
Department: Computer Science

Short track speed skating has become popular all over the world. The demands of a computer-aided training (CAT) system are booming due to this fact. However, the existing commercial systems for sports are highly dependent on expensive equipment and complicated hardware calibration.

This dissertation presents a novel CAT system for tracking multiple skaters in short track skating competitions. Aiming at the challenges, we utilize global rink information to compensate camera motion and obtain the global spatial information of skaters; apply Random Forest to fuse multiple cues and predict the blobs for each of the skaters; and finally develop a silhouette and edge-based template matching and blob growing method to allocate each blob to corresponding skaters. The proposed multiple skaters tracking algorithm organically integrates multi-cue fusion, dynamic appearance modeling, machine learning, etc. to form an efficient and robust CAT system. The effectiveness and robustness of the proposed method are presented through experiments.

(74 pages)

PUBLIC ABSTRACT

A Computer-aided Training (CAT) System for Short Track Speed Skating

Chenguang Liu

Short track speed skating was adopted by the International Skating Union in 1967, and upgraded to full winter Olympic sport status in 1992. Even though its history is short compared with long track speed skating, it became popular around the world because it is more intense and more entertaining for audiences. The demands of having a CAT system for gathering and analyzing competition data automatically is raising drastically due to its growing popularity all around the world.

There have been some commercial systems for some other sports, which are able to provide extrinsic feedback information to coaches and athletes. However, there is no commercial sports analysis system for short track speed skating yet, and the current commercial sports analysis systems have certain limitations including the requirement of operator intervention to process the video and the necessities of the restricted environments such as multiple cameras with complex camera settings and expensive peripherals.

The proposed CAT system greatly reduces the requirement of hardware settings and the system cost by utilizing only monocular videos captured using a single handycam. Moreover, it automatically tracks multiple skaters and output accurate skater spatial information which provides valuable references to the coaches to improve the skaters' performances in international competitions.

I dedicate this thesis to

my parents, Chunsheng Liu and Yuxia Li,

my wife, Biying Qin,

and my daughter, Jaycee Liu

ACKNOWLEDGMENTS

I would like to express heartfelt gratitude to my PhD advisor, Dr. Heng-Da Cheng, for his strong support during these four years. When I was disoriented in the academic ocean, he guided the direction with insightful advice; when I lost confidence, he kept encouraging me with patience. I would also like to thank my PhD committee members, Dr. Vicki Allan, Dr. Curtis Dyreson, Dr. Haitao Wang, and Dr. Lie Zhu, who have spent a lot of their precious time attending my examinations, reviewing my proposal and dissertation, and providing me with many encouraging and constructive feedbacks.

To my dear comrades Yanhui Guo, Yuxuan Wang, Juan Shan, and Min Xian at the Computer Science Department's CVPRIP lab, Utah State University, I am grateful for the chance to conduct research with you and thank you for helping to develop valuable ideas in this dissertation.

To my parents, thank you for instilling within me a strong curiosity of the unknown and a love of creative pursuits, so that I am able to contemplate this road. This dissertation would also not be possible without the love and support of my wife, Biying. Thank you my beautiful little daughter, Jaycee, who has brought so much happiness to me and my family.

And last, but not least, to Dr. Aravind Dasu, Dr. Hunter Wu, and all team members at Intelligent Occupancy Group and at Wireless Power Transfer Group of the USU Research Foundation, thank you for being so supportive of me.

Chenguang Liu

CONTENTS

Page

CHAPTER

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Short track speed skating is a form of competitive ice speed skating. It originated in the speed skating events held with mass starts and has the history of nearly 50 years. Before it was adopted officially in international competitions, this form of speed skating was mainly practiced in the United States and Canada where skaters skated in pairs. In 1967, the International Skating Union adopted short track speed skating. However, it did not organize international competitions until 1976. World Championships have been held since 1981. After several changes in the name of the competition, the event is now held annually as the World Short Track Speed Skating Championships. Meanwhile, short track speed skating was upgraded to a full Olympic sport in 1992 and has been part of the Winter Olympics since then. In the competitions, the events are the same for both men and women, and include 500 m, 1000 m, 1500 m, 3000 m, and the relay (5000 m (men)/3000 m (women)).

Compared to long track speed skating, which only has pair skaters in the rink, the short track speed skating is more intense and, to some degree, more entertaining to audiences. Due to its popularity, short track speed skating has raised attention in many countries including China, United States, Canada, South Korea, Japan, etc. Many funds are invested to training skaters in order to achieve the best performance in international competitions. Besides, skaters are also seeking every possibility to improve their skills and strategies. Moreover, the speed control and curve strategies of a skater in real competition provide valuable reference for his/her future training, hence a video-based computer-aided training (CAT) system will be very helpful for skaters to analyze their performance in the competition and improve it in the future.

There have been some commercial systems such as TRAKUS, SoccerMan, TRAKPERFORMANCE, Pfinder, etc. which provide extrinsic feedback information to coaches and athletes. However, the current commercial sports analysis systems have certain limitations. They either require operator intervention to process the video or are often limited by the restricted hardware environments such as the necessity of special camera settings with complex calibration or expensive peripherals.

In terms of the proposed CAT system, the necessity of complicated camera settings and expensive peripherals are excluded since it is able to process monocular videos captured by a single handycam. The movement of skaters on the rink provides useful information for coaching or live sports. Unlike the existing sports analysis systems which require manual intervention or annotation, the proposed CAT system is able to track each of the skaters automatically and output the their speeds and trajectories accurately.

The methodologies used in surveillance tasks cannot be directly applied in the context of short track speed skating since a lot of constraints have to be considered, e.g. quick dynamic events, real time analysis, complex situations of occlusions, precise positioning of skaters in the field, and so on. Algorithms for people tracking and object detection have to face difficult situations such as the overlapping of skaters wearing the same uniform, unpredictable trajectories, and a wide and dynamic camera view. Because of these reasons, the automatic short track speed skating video analysis is challenging for scientific studies.

To solve the challenges mentioned above, the author conducted researches in four major aspects: 1) Homography-based image transformation method is applied to eliminating the effect of camera motion, and the spatial information of the skaters is obtained according to the constructed rink panorama and is served as spatial feature for tracking skaters; 2) a random forest (RF) [1] classifying scheme is developed for fusing multiple features and updating appearance model to deal with appearance variation; 3) a blob growing occlusion handling mechanism is developed to classify blobs to different skaters in occlusion; 4) an appearance updating and likelihood voting method is proposed to detect the disappeared skaters.

1.2 Related Works

In this section, we will review the related works published in recent years. There are mainly two categories of related works that we will discuss about: the related team sports video analysis methodologies; and the general visual tracking methodologies aiming at solving challenging problems similar to short track speed skating.

1.2.1 Team Sports Video Tracking and Analysis Methodologies

In sports, the quantitative analysis of team and player activities has become an important aspect of coaching [2]. Accurate positional information about sports players is of interest to coaches to improve the performance of the team strategically, and to assist in the design of better training programs. Furthermore, such information can also be used to understand the coordination dynamics of player activities and the most influential constraints acting upon them.

Different from the general object tracking context, athletes tend to exhibit quick movements with many unpredictable direction changes and frequent collisions with other players. These characteristics of player behavior violate the assumptions of smooth movement which general computer tracking algorithms are typically based on.

There have been several researchers contributing to tracking and analyzing competition videos in short track speed skating context. Liu et al. [3, 4] proposed a single skater tracking method based on Unscented Kalman Filter (UKF), and Wang et al. [5] put forward a single skater tracker based on two-region joint probability. The methods are able to deal with partial occlusion, but cannot be used to analyze competition videos of multiple skaters, and fail when severe occlusion happens.

The work most similar to the proposed method is [6] which presented a multiple skater tracking system based on a combination of GMM and fuzzy membership. Even though the tracker can deal with multiple skaters tracking and partial occlusion, it has several drawbacks: 1) only GMM based color feature is applied to matching the templates initialized in the first frame without updating the appearance model of each skater during tracking. Therefore, the tracker is fragile to appearance variation of the skaters, which may cause less accurate tracking result or even failures; 2) the constructed rink panorama is used to calculate the global spatial information of the skaters in each frame, but it is not used as a feature to improve tracking performance. The rink panorama is only used for rendering the tracking result (similar to the references [3], [4] and [5]); 3) only partial occlusion is handled; 4) all of the skaters are dealt with in the same region of interest (ROI), therefore it fails to process the video with one or more skaters out of the scene. However, complex occlusion, sometimes full occlusion, happens frequently in the competition, and the tracker will fail on tracking the occluded skaters.

In addition to the tracking algorithms on short track speed skating, there have been a number of methods published in recent years which are developed for team sports video tracking and analyzing. As soccer is one of the most popular games around the world, the research on various aspects of soccer video analysis has been growing in the recent years. The methods either uses fixed view [7-9] or utilizes grass color background modeling [10, 11] to obtain player blobs, and then apply constraint of smooth movement to solve occlusion problems. The tracking of soccer players is still facing the occlusion issues, and the demand of tracking accuracy is less important than short track speed skating due to its most popular usage for event recognition based video indexing.

In addition, there has been some notational analysis methods used to investigate the activity patterns and techniques in various sports, such as rugby [12, 13], squash [14, 15, 16], badminton [17], basketball [18, 19], volleyball [20], etc. Notational methods circumvent the challenges in automatic visual tracking and analyzing by human intervention. However, they are too time consuming and fragile to subjective factors.

1.2.2 General Object Tracking Methodologies

Visual object tracking has been a booming research area for decades, and many contributions have been made to solving the open problems such as camera motion handling, appearance modeling, template updating, occlusion handling, etc. Even though the methods are successful in some aspects, the problems remain open in visual object tracking due to their difficulties.

Tracking multi-object in videos captured by moving camera is a challenging task which has been attracting many attentions in computer vision. Choi et al. [21, 22] present a novel way to dynamically calibrate the camera by detecting feature points on the ground. By doing so, the camera motion is represented by camera parameters. Then, a tracking by-detection method is applied to estimating trajectories of pedestrians. Even though successful, it is difficult for the method to handle occlusion because the occluded person is difficult to be detected by object detection algorithms. Moreover, detecting robust features on ice is impractical.

Reilly et al. [23] and Ali et al. [24] build global motion model according to the features detected in each frame, and apply accumulative frame differencing with background modeling to detect motion in the scene captured by UAVs. The proposed method adopts similar approach to detect motion of the skaters and eliminate the effect of the camera motion. The difference is that we also use the constructed rink panorama to calculate the global spatial information of the skaters in each frame, and it will be used as a feature to improve tracking performance. However in [3], [4] and [6], the rink panorama is only used for rendering the tracking result.

In order to achieve robust tracking while subjects are changing appearance, there are two major approaches to adopt. On one hand, multi-cue fusion [25, 26] may be applied to tracking so that more information can be enrolled to improve the robustness. On the other hand, dynamic appearance modeling [27-29] is an important way to handle appearance changes. In this dissertation, we propose a new scheme of RF to fuse multiple features including color, silhouette and spatial information. During tracking, the appearance model of each of the skaters is also updated in each frame to solve the appearance changing problem.

Occlusion handling is another important issue in video-based multi-object tracking. Due to the fierce competition among skaters, inter-skater occlusion happens frequently. Unlike the existing methods [30, 31] dealing with inter-object occlusion in which the objects are relatively large enough to be separated into smaller parts, our method is facing the problem that the skaters are changing scale in a large domain.

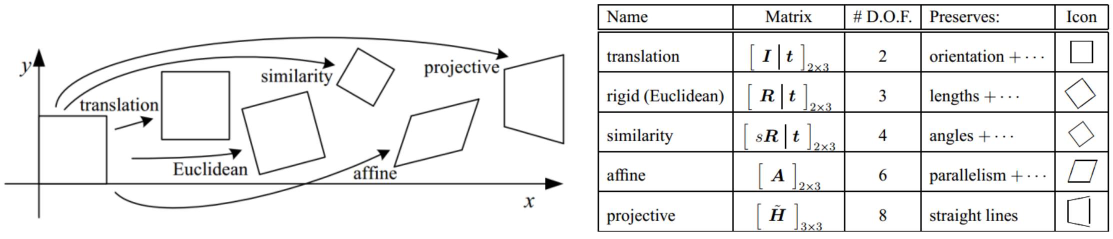
1.3 Research Scope

In this dissertation, we built a CAT system which can automatically track the movements of high speed skaters competing on large-scale rink and provide spatial information of the skaters. However, there are several open challenges we will face:

1) The camera is being panned rapidly in order to capture the fast moving skaters on the large-scale rink.
2) Skaters being tracked are non-rigid fast moving objects.
3) The skaters’ appearances including silhouette, color and scale changes drastically when they are competing on the rink.
4) Severe occlusion happens frequently among skaters, and sometimes even among skaters wearing the same type of uniform.
5) One or more skaters may be out-of-scene and reappear frequently during competition, and should be identified when they reappear.

1.3.1 Camera Motion

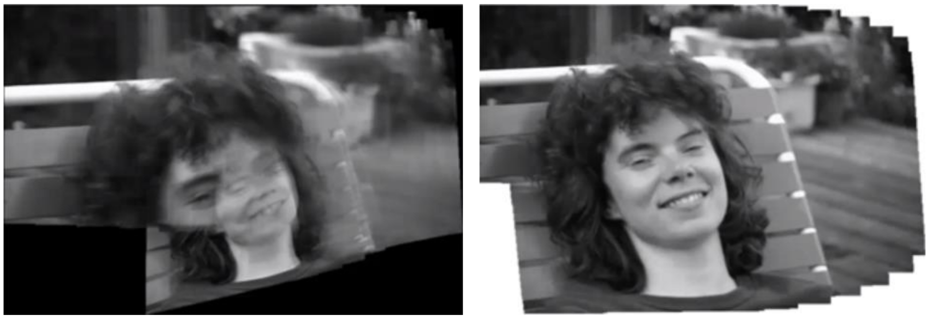
Accurate positional information is of great interest to coaches. In order to obtain precise positional information, the camera motion in short track speed skating video has to be compensated. Global flow has been widely used for compensating camera motion in object tracking. As is shown in Fig. 1.1, there are a set of 2D planner transformations. Due to the nature of projective model, it can better deal with camera rotate motion [32]. An example is shown in Fig. 1.2 where affine model and projective model based image stitching results are compared.



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\, \boldsymbol{I} \mid \boldsymbol{t} \,\right]_{2\times 3}$	2	orientation + ···	□
rigid (Euclidean)	$\left[\, \boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times 3}$	3	lengths + ···	◇
similarity	$\left[\, s\boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times 3}$	4	angles + ···	◇
affine	$\left[\, \boldsymbol{A} \,\right]_{2\times 3}$	6	parallelism + ···	▱
projective	$\left[\, \tilde{\boldsymbol{H}} \,\right]_{3\times 3}$	8	straight lines	⏢

(a) (b)

Figure 1.1. Types of transformation (a) Basic 2D planner transformation set. (b) Hierarchy of 2D coordinates transformations [32].



(a) (b)

Fig. 1.2. Comparison between affine mosaic and projective mosaic. (a) Affine mosaic. (b) Projective mosaic [32].

There have been a number of methods using Homography based projective model to build panorama of the rink [3, 4, 6]. And then, the tracked positional information of skater is projected to the panorama in order to render the tracking results. In these methods, the global positional information is not used for tracking the skaters but only for rendering the results.

We utilized the camera compensation method introduced in [32] in which the projective transformation between consecutive frames is calculated through feature points detection and RANSAC homography estimation. The mosaic corresponding to camera motion will then be built according to the homography transformations. Then, the skater motion can be captured by background subtraction.

1.3.2 Appearance Modeling

Tracking high speed skaters in short track speed skating competition suffers from severe appearance change including color, shape, texture, scale, etc. In order to robustly track the skaters, appearance model updating has to be applied to dealing with the appearance change. In the study, we will conduct researches on finding appropriate features to represent a skater and developing new feature fusion method to achieve robust tracking of skaters.

The apparent color of an object is influenced primarily by two physical factors: the spectral power distribution of the illuminant and the surface reflectance properties of the object. Color distributions provide an efficient feature for tracking as they are robust to partial occlusion, rotation and scale invariant. Moreover, color is also computationally efficient [33]. Edge detection is used to identify points in an image at which the image brightness changes sharply or more formally has discontinuities, namely the strong changes in image intensities usually generated by object boundaries. A notable property of edge feature is that it is less sensitive to illumination variation compared to color feature [34, 35]. As skaters are well trained professional athletes who present standard gestures in competitions, silhouette of skaters provides useful information for detecting skaters. Besides studying the color and edge features, we will also apply silhouette feature to automatic skater tracking.

Due to the severe appearance changes of skaters in short track speed skating, and the fact that fusing multiple features have been widely used to achieve robust tracking, we will study on fusing representable features. Current researches on object tracking are mainly using linear fusion for fusing multiple features [36-38].

The general methodology of linear fusion is presented as follows. Given I_k $(1 \leq k \leq n)$ as a feature vector obtained from the kth video source, and w_k $(1 \leq k \leq n)$ as the normalized weight assigned to the kth video source, then the combined feature vector is calculated by using sum (Eq. 1.1) or product (Eq. 1.2) operations, which can be used by the classifiers to provide a high-level decision. Linear fusion based methodologies are less time consuming, but a fusion system needs to determine and adjust the weights for the optimal accomplishment of a task.

$$I = \sum_{k=1}^{n} w_k \times I_k \tag{1.1}$$

$$I = \prod_{k=1}^{n} {I_k}^{w_k} \tag{1.2}$$

Despite fusing multiple features in object tracking, updating the appearance model is also very important to achieve robust tracking. In this dissertation, the author proposed an RF and template matching based multi-cue fusion and dynamic appearance modeling method which combines color, position, silhouette and edge features organically.

1.3.3 Occlusion Handling

Current monocular methods can be categorized into two major classes in terms of dealing with occlusion problems: 1) tracking occluded objects as disappeared [39] or roughly merging the grouped objects [30, 40]; 2) applying a kinematic model to predict the trajectories of occluded objects [41-43]. The former methods are not suitable for accurate object tracking applications such as high speed skater tracking, while the later ones are fragile to abnormal object movements.

In [44], the authors proposed a linear programming relaxation scheme for multiple object tracking. It explicitly models tracking interaction, such as object spatial layout consistency or mutual occlusion, and successfully tracks multiple persons with heavy mutual occlusion. In [45], the algorithm utilized the detected heads' positions to estimate the depth information of objects, and then solved the occlusion problem based on the depth relations among objects. The above two methods are suitable for surveillance video tracking where the subjects are moving relatively slow and the camera is stationary.

However, in our application, the skaters' positions need to be accurately tracked. Therefore, the occluded skaters also need to be tracked. In this dissertation, a occlusion relationship based blob growing method is proposed to handle severe occlusion among skaters.

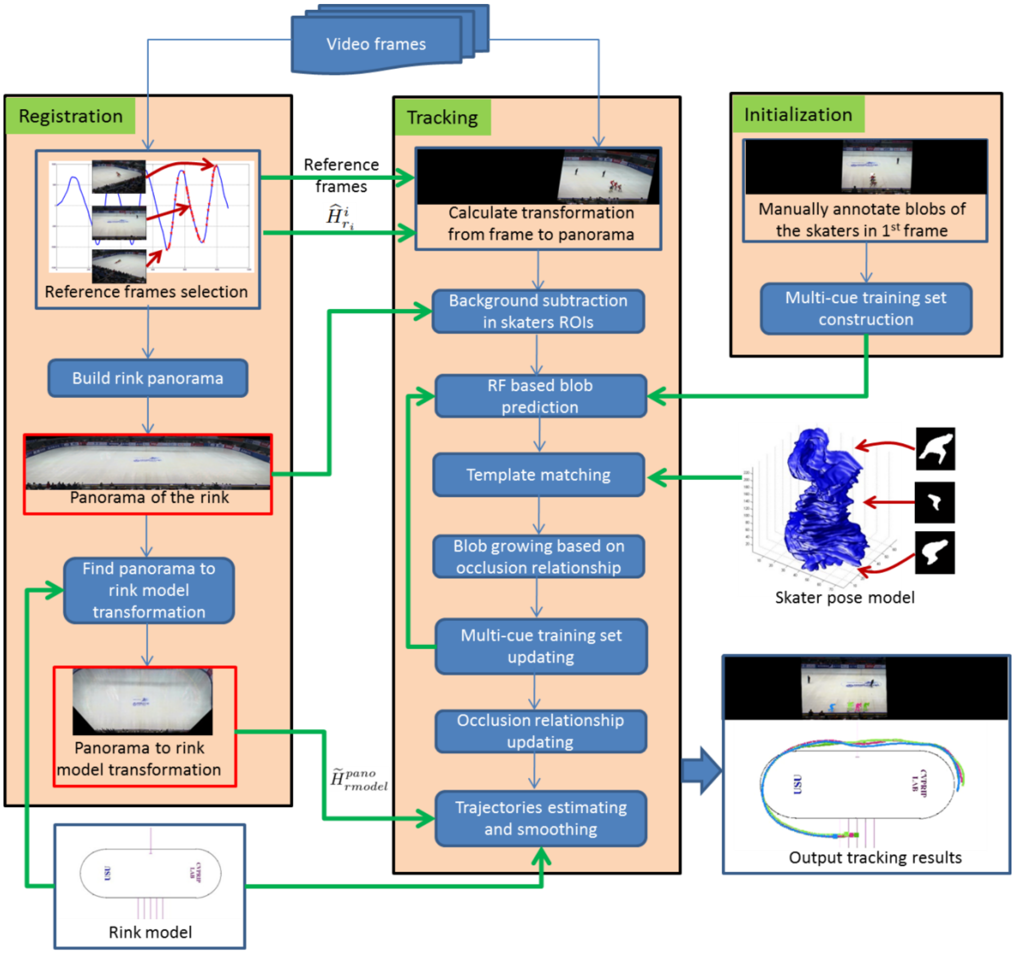

Video frames
Registration
Tracking
Initialization
Reference frames selection
Reference frames
$\widehat{H}^i_{r_i}$
Calculate transformation from frame to panorama
Manually annotate blobs of the skaters in 1st frame
Background subtraction in skaters ROIs
Multi-cue training set construction
Build rink panorama
RF based blob prediction
Panorama of the rink
Template matching
Blob growing based on occlusion relationship
Skater pose model
Find panorama to rink model transformation
Multi-cue training set updating
Occlusion relationship updating
Panorama to rink model transformation
$\widetilde{H}^{pano}_{rmodel}$
Trajectories estimating and smoothing
Output tracking results
Rink model

Figure 1.3. The process flowchart of the proposed method.

1.3.4 Overview of the Methodology

In this section, we will briefly discuss the basic procedure of the proposed methodology. The proposed multi-skater tracking method can be divided into two sub-procedures: the registration procedure and the tracking procedure. The registration is a procedure where homography transformation is employed to achieve three goals: 1) extract reference frames to generate global view of rink mosaics; 2) map each of the frames to the global rink mosaics so that background subtraction can be used to get the foreground blobs of the skaters; 3) find the transformation from the global rink mosaics to the rink model so that the skaters' spatial information on the rink plane can be obtained.

Tracking is a procedure where RF skater predicting, template matching and blob growing method are performed to allocate and refine skater blob.

The basic process of the proposed method is illustrated in Fig. 1.3. The thick green arrows represent data transactions, and the thin blue arrows depict the processing transfers. The registration outputs not only impact on the tracking procedures, but also play important roles in outputting the tracking results. The details of the proposed method will be discussed in the following sections.

The differences between the proposed method and the method presented in STS [6] can be seen in Fig. 1.3. Firstly, our method utilizes the global positional information of skaters for tracking to increase the tracking accuracy, while STS only uses the constructed rink panorama for rendering the tracking results. Secondly, by applying RF, our method is able to combine color and positional information together and update the appearance model. Moreover, edge and silhouette cues are also fused into the algorithm using improved template matching. Because RF and template matching are less complicated and are parallelizable, the proposed method is to be less time consuming and more accurate than STS.

The rest of this dissertation is organized as follows: Section 2 describes the registration procedure. Section 3 shows the design of random forest based multi-cue fusion and dynamic appearance modeling algorithm. Section 4 discusses the skater blob correction and occlusion handling procedure. Section 5 depicts the detection of disappeared skaters using dynamically updated color appearance model. Section 6 exhibits the experimental results of the proposed CAT system.

CHAPTER 2

RANDOM FOREST BASED MULTI-CUE FUSING AND DYNAMIC APPEARANCE MODELING

2.1 *Rink Registration*

Rink registration plays an important role in the proposed method. The basic procedure and outputs of rink registration is illustrated in Fig. 2.1. The output of the registration procedure includes: N_{ref} reference frames automatically selected from the video sequences; the panorama of the rink which is built by calculating the transformations $H_{C_{ref}}^{r}$ $(r = 1, \dots, N_{ref})$ from the *r*th reference frame to the center reference frame $C_{ref} = (N_{ref} + 1)/2$; the transformations $\widehat{H}_{r_i}^{i}$ $(i = 1, \dots, N_{frm})$ from the *i*th frame to the spatially nearest reference frame r_i; and the transaction $\widetilde{H}$ from panorama to rink model.

The transactions mentioned above are all 3×3 homography matrices. In computer vision, homography is a relationship describing two images of the same planar surface according to pinhole camera model [46]. Similar to [3] and [6], we apply robust feature point detection on the two correlated images, and then pass the feature points to RANSAC [47] algorithm to estimate the homography matrix. To calculate the translation $\widetilde{H}$, we manually annotated the pair of corresponding points on the panorama and the rink model, respectively, and then perform RANSAC process. In this study, we adopt the automatic reference frame selection method in [6] and select $N_{ref} = 41$ reference frames. The transformation $N_{C_{ref}}^{r}$ is calculated by accumulating the consecutive transformations from the *r*th reference frame to the center reference frame. For instance, $H^{15}_{C_{ref}} = H^{15}_{16} * H^{16}_{17} *, \dots, * H^{20}_{21}$.

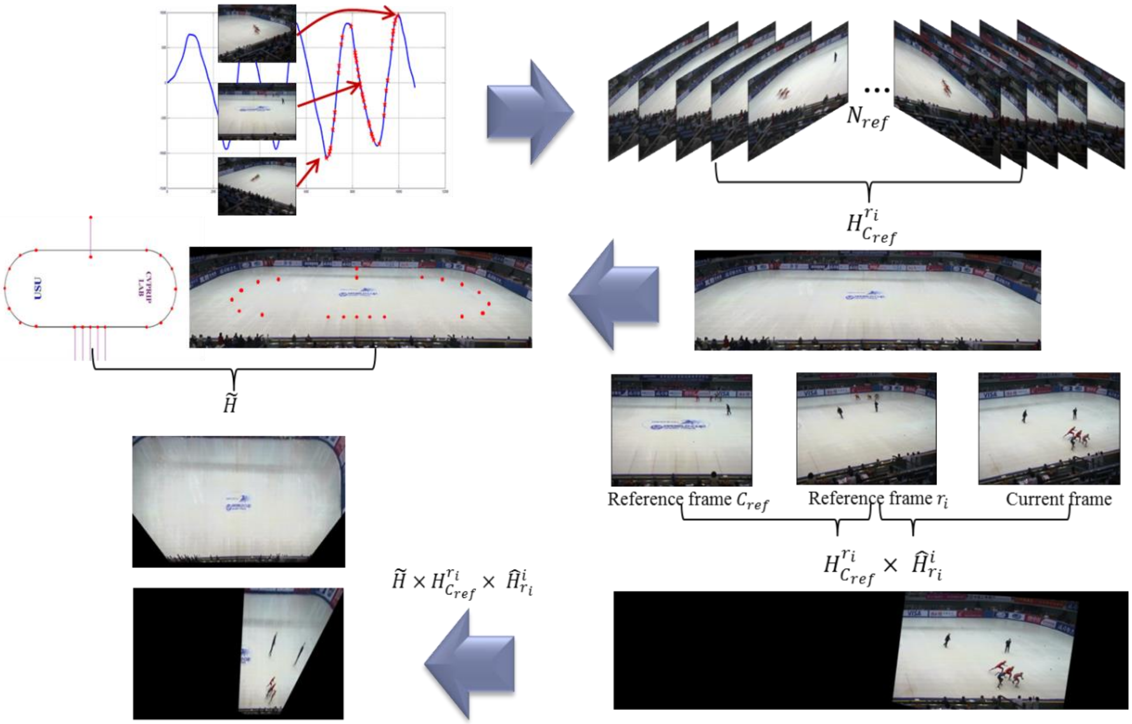

N_{ref}
$H^{r_i}_{C_{ref}}$
USU
CVPRIP LAB
$\widetilde{H}$
Reference frame C_{ref}
Reference frame r_i
Current frame
$H^{r_i}_{C_{ref}} \times \widehat{H}^{i}_{r_i}$
$\widetilde{H} \times H^{r_i}_{C_{ref}} \times \widehat{H}^{i}_{r_i}$

Figure 2.1. Rink registration procedure and outputs.

2.2 *Multiple Skaters Tracking*

Before tracking procedure starts, we apply the transformation $\widehat{H}^{i}_{r_i}$ to transform all pixels on the original frame to the panorama image (Fig. 2.2b) based on Eq. 2.1.

$$u = c\widehat{H}^{i}_{r_i} \cdot (x, y, 1)^T \;\; (i = 1, \dots, N_{frm}) \tag{2.1}$$

where u is the new transformed pixel position of original frame pixel (x, y), $\widehat{H}^{i}_{r_i}$ is the homography transformation from the *i*th frame to the spatially nearest reference frame r_i, and c is a normalization factor.

The ROIs of skaters are either manually annotated in the first frame or automatically generated according to the tracking results of the previous frames. The foreground blobs in each ROI are obtained by background subtraction (Fig. 2.2e).

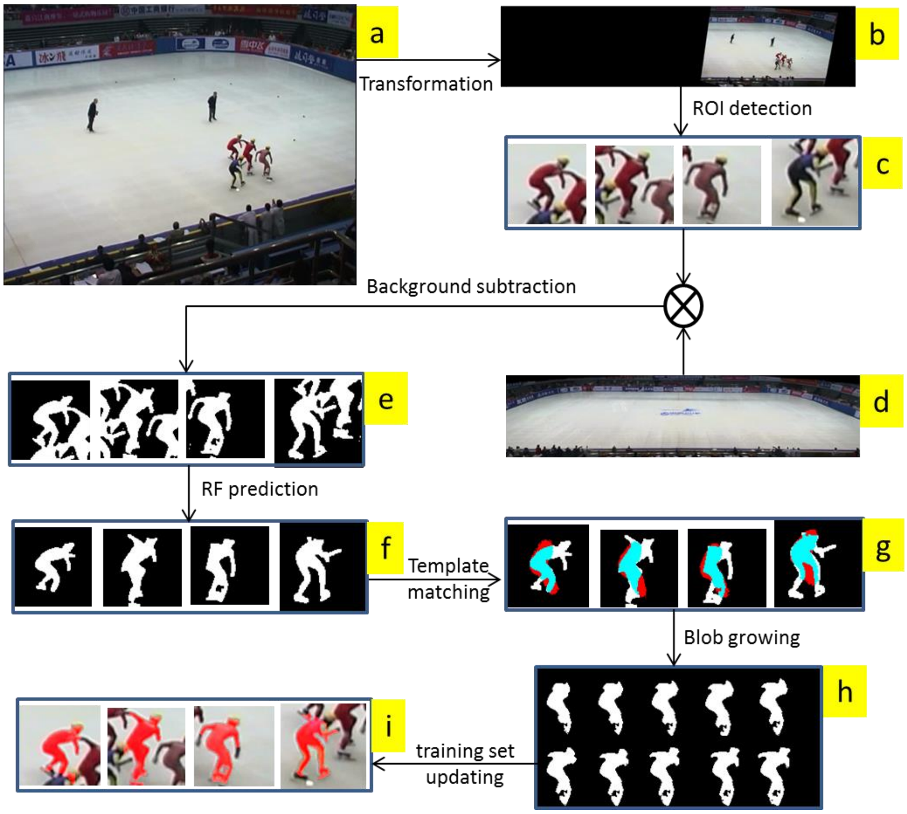


Figure 2.2. The flowchart of tracking. (a) Original frame. (b) The frame panorama after transformation. (c) The ROIs of skaters in (b). (d) The global panorama. (e) The detected blobs in each ROI after background subtraction. (f) The blobs predicted by RF algorithm. (g) Template matching results. (h) An example of blob growing. (i) The pixels (in red) to be used for RF training set updating.

The RF will be trained with the training set updated in the previous frame. And then, we construct the testing set using the detected foreground blobs in each ROI for RF classification (Fig. 2.2f). Sequentially, the silhouette and edge-based template matching will be performed on the blobs predicted by RF (Fig. 2.2g). To assure that the detected blobs are allocated to each skater correctly, we perform a blob growing process with respect to the status of inter-skater occlusion (Fig. 2.2h). Finally, the tracked blobs will be used to obtain the features for constructing the training set of RF (Fig. 2.2i). The details of the procedure will be discussed in the following subsections.

2.3 Random Forest Classification

2.3.1 Random Forest

Random forest is an ensemble learning method for classification and regression. It constructs a multitude of decision trees at training time, and outputs the class that is the mode of the classes output by individual tree. More specifically, an RF is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class of input x [1].

The margin function (Eq. 2.2) which measures the extent to which the average number of votes at X and Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence the classification has.

$$mg(X, Y) = av_k I(h_k(X) = Y) - max_{j \neq Y}\ av_k I(h_k(X) = Y) \qquad (2.2)$$

where $I()$ is the indicator function and av_k is the average votes of the kth tree. The generalization error is as below:

$$PE^* = P_{X,Y}(mg(X, Y) < 0) \qquad (2.3)$$

According to theorem 1.2 in [1], when the number of trees increases, the generalization error will converge to $P_{X,Y}(P_\Theta(h(X, \Theta) = Y) - max_{j \neq Y} P_\Theta(h(X, \Theta) = j) < 0)$ for all sequences $\{\Theta_1, \ldots\}$. Hence, RF is guaranteed to converge given enough number of trees.

By comparing with Adaboost algorithm, the advantages of using RF can be addressed as: 1) the accuracy of RF cannot be worse than Adaboost; 2) RF is relatively robust to outliers and noise; 3) RF is less time consuming than bagging or boosting algorithms; 4) RF can provide useful internal estimates of error, strength, correlation and variable importance; and 5) RF is less complicated and is parallelizable.

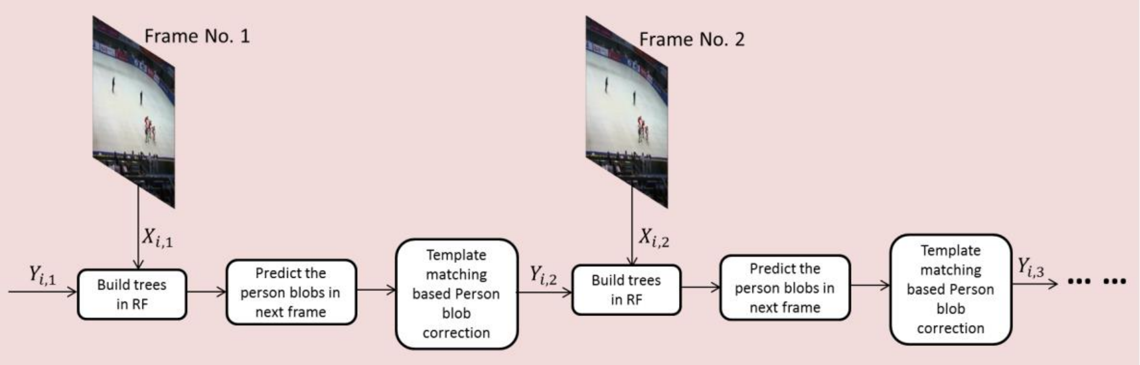

Frame No. 1
Frame No. 2
$X_{i,1}$
$X_{i,2}$
$Y_{i,1}$
$Y_{i,2}$
$Y_{i,3}$
Build trees in RF
Predict the person blobs in next frame
Template matching based Person blob correction
Build trees in RF
Predict the person blobs in next frame
Template matching based Person blob correction
••• •••

Figure 2.3. The basic processing flowchart of the random forest learning and classification.

Due to the state of the art performance and the desirable characteristics mentioned above, RF has attracted many attentions in object tracking [29, 48]. However, because of the drastic scale change of the objects, none of the existing algorithms dividing the objects to smaller parts for feature extraction is suitable for our application. In this paper, we develop a novel multi-cue fusion method based on RF about which we will discuss on two major aspects - the features of RF and how RF works. The main procedure of applying the RF is illustrated in Fig. 2.3.

2.3.2 Feature of Random Forest

In this section, we will discuss about the features being used in the RF. Given a set of training data of the *m*th skater at time step *t*: $D_{n,t}^{m} = \{(X_{i,t}, Y_{i,t})\}_{i=1}^{n}$ where $X_{i,t}$ is the feature vector and $Y_{i,t}$ is the class of the *i*th sample, and *n* is the number of samples, i.e. the number of pixels on the blobs being tracked at time step *t*. There is no doubt that $Y_{i,t}$ is equal to *m* in our application. We will use the position and color features to train the RF and to predict the blobs of the corresponding skater after the foreground of the skater has been obtained through background modeling. Regarding to this manner, we have $X_{i,t} = (\rho_{i,t}, \alpha_{i,t}, R_{i,t}, G_{i,t}, B_{i,t})$ which is a feature vector containing five different elements. $\rho_{i,t}$ and $\alpha_{i,t}$ are predicted polar coordinates of the *i*th sample w.r.t the origin at the upper-left corner of the transformed panorama (Fig. 2.2b). $R_{i,t}$, $G_{i,t}$ and $B_{i,t}$ are the R, G and B values of the *i*th sample, respectively. By doing so, we are fusing position and color features together.

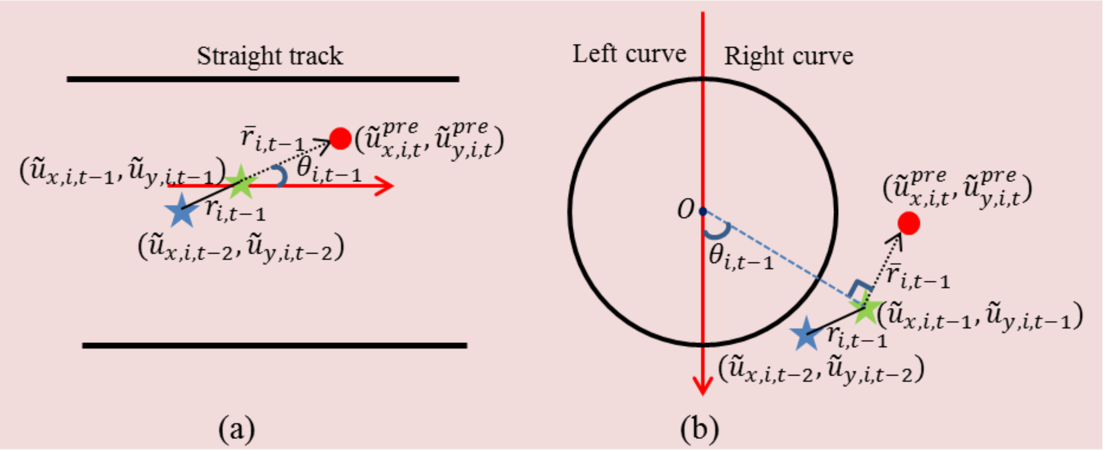

Straight track
$\bar{r}_{i,t-1}$
$(\tilde{u}^{pre}_{x,i,t}, \tilde{u}^{pre}_{y,i,t})$
$(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1})$
$\theta_{i,t-1}$
$r_{i,t-1}$
$(\tilde{u}_{x,i,t-2}, \tilde{u}_{y,i,t-2})$
(a)
Left curve
Right curve
O
$(\tilde{u}^{pre}_{x,i,t}, \tilde{u}^{pre}_{y,i,t})$
$\theta_{i,t-1}$
$\bar{r}_{i,t-1}$
$(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1})$
$r_{i,t-1}$
$(\tilde{u}_{x,i,t-2}, \tilde{u}_{y,i,t-2})$
(b)

Figure 2.4. Dynamic model. The “blue” and “green” stars represent the skater rink model position at time step $t-2$ and $t-1$, respectively. The “red” circle indicates the predicted rink model position at time step t. And the “red” arrow serves as reference direction used to calculate angle $\theta_{i,t-1}$. (a) Dynamic model in straight track. (b) Dynamic model in curves.

The calculation of the location feature $\rho_{i,t}$ and $\alpha_{i,t}$ is illustrated by Eq. 2.4 and Eq. 2.5, respectively.

$$\rho_{i,t} = \sqrt{u_{x,i,t}^2 + u_{y,i,t}^2} \tag{2.4}$$

$$\alpha_{\mathrm{i,t}} = \tan^{-1}\left(\frac{u_{y,i,t}}{u_{x,i,t}}\right) \tag{2.5}$$

where $u_{x,i,t}$ and $u_{y,i,t}$ are predicted x and y coordinates of the ith sample at time step t w.r.t the upper-left corner of the transformed panorama (Fig. 2.2b), respectively. The predict position is calculated by Eq. 2.6.

$$(u_{x,i,t}, u_{y,i,t}) = c\widetilde{H}^{-1} \cdot \left(\tilde{u}_{x,t}^{pre}, \tilde{u}_{y,t}^{pre}, 1\right)^T \tag{2.6}$$

where $\left(\tilde{u}_{x,t}^{pre}, \tilde{u}_{y,t}^{pre}\right)$ is the predicted position of the corresponding skater at time step t which is represented by the centroid of the blob and will be calculated by a dynamic model (Eq. 2.7); $\widetilde{H}^{-1}$ is the inverse of homography transformation $\widetilde{H}$; and c is a normalization factor.

$$\left(\tilde{u}_{x,t}^{pre}, \tilde{u}_{y,t}^{pre}\right) = \left(\tilde{u}_{x,i,t-1} + dx, \tilde{u}_{y,i,t-1} + dy\right) \tag{2.7}$$

$$(dx, dy) = \bar{r}_{i,t-1} \cdot (cos\,\theta_{i,t-1}, -sin\,\theta_{i,t-1}) \tag{2.8}$$

where $\left(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1}\right)$ represent the rink model position of the ith skater at time step $\mathrm{t}-1$, and (dx, dy) are variances on x and y coordinates which will be calculated according to Eq. 2.8. Here $\bar{r}_{i,t-1} = \frac{1}{k}\sum_{j=1}^{k} r_{i,t-j}$ represents the average speed and $r_{i,t-j}$ indicates the distance between rink model positions $\left(\tilde{u}_{x,i,t-j}, \tilde{u}_{y,i,t-j}\right)$ and $\left(\tilde{u}_{x,i,t-j-1}, \tilde{u}_{y,i,t-j-1}\right)$. The predicted direction is determined by angle $\theta_{i,t-1}$ which is defined by Eq. 2.9.

$$\theta_{i,t-1} = \begin{cases} tan^{-1}\left(\dfrac{\tilde{u}_{y,i,t-1} - O_y}{\tilde{u}_{x,i,t-1} - O_x}\right) + \dfrac{3\pi}{2} & \left(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1}\right) \in P_{LC} \\ tan^{-1}\left(\dfrac{\tilde{u}_{y,i,t-1} - \tilde{u}_{y,i,t-2}}{\tilde{u}_{x,i,t-1} - \tilde{u}_{x,i,t-2}}\right) & \left(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1}\right) \in P_S \\ tan^{-1}\left(\dfrac{\tilde{u}_{y,i,t-1} - O_y}{\tilde{u}_{x,i,t-1} - O_x}\right) + \dfrac{\pi}{2} & \left(\tilde{u}_{x,i,t-1}, \tilde{u}_{y,i,t-1}\right) \in P_{RC} \end{cases} \tag{2.9}$$

where $\left(O_x, O_y\right)$ are coordinates of curve center O, and P_S, P_{LC} and P_{RC} are point sets representing the points in straight track, left curve and right curve, respectively.

```
Algorithm: Generating random forest
Input: Training set $D_{n,t}^{m}$, Number of trees $N_B$
Output: Random Forest $F$

FOR b=1,..., $N_B$
    Choose bootstrap sample $D_{b,t}^{m}$ from $D_{n,t}^{m}$.
    Construct the bth tree using $D_{b,t}^{m}$ s.t.
    1) Randomly choose subset of $q$ features s.t. $q < 5$
    2) Split on only chosen q subset of features.
END
```

Figure 2.5. The algorithm of random forest generation.

2.3.3 How Random Forest Works

Given the training set $D_{n,t}^{m}$ constructed for the *m*th skater with *n* samples at time step *t*, the RF is constructed by training procedure. In this dissertation, the RF will be reconstructed in the tracking procedure at each time step according to the algorithm illustrated in Fig. 2.5. The number of trees of the RF is set to 200 through experiments.

After the blobs in each ROI have been detected by background subtraction, the trained RF will be performed to classify the blobs. The testing data set is constructed similarly to the training set. Let $\bar{D}_{n,t}^{m} = \{(\bar{X}_{i,t}, \bar{Y}_{i,t})\}_{i=1}^{n}$ be the testing data set where $\bar{X}_{i,t}$ is the feature vector of the *i*th testing sample at time step *t*; $\bar{Y}_{i,t}$ denotes the class of the *i*th testing sample, which will be set by Eq. 2.10; and *n* is the number of testing samples, namely the number of pixels in the detected blobs.

$$\bar{Y}_{i,t} = \begin{cases} m, & V(m) > \tau \ and \ m = argmax_{j=1}^{N_{obj}}\{V(j)\} \\ 0, & otherwise \end{cases} \tag{2.10}$$

where N_{obj} is the number of skaters; *V*(*m*) denotes the voter value of the *m*th skater; and τ is the threshold indicating if the corresponding voter is reliable or not. The threshold τ is set to 50 through experiments.

The predicted blobs by RF may not be accurate enough to deal with occlusion among skaters, especially when the correlated skaters have similar appearance or even wearing the same uniforms. The template matching based blob correction procedure is applied to refine the blobs of skaters and handle the occlusion among skaters.

CHAPTER 3

TEMPLATE MATCHING BASED SKATER BLOB DETECTION

Each time when the new RF is constructed it utilizes the tracked blobs as the samples for training (Fig. 2.3). If the tracked blobs are not accurate enough, the error will accumulate in the subsequent RF training and predicting procedures, which is so called “drifting” problem in object tracking.

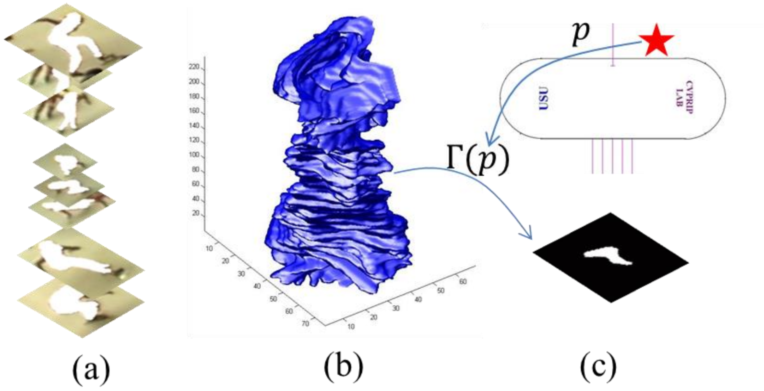
In sports video analysis, many methodologies take advantage of using pose model of sportsman due to the fact that the poses in many sports are usually standardized. Similarly, the skater poses in short track speed skating are basically predictable, and legitimately utilizing pose information will improve the tracking performance. To bootstrap the problem, we propose a template matching based blob detection and occlusion handling method.

3.1 Skater Pose Model Construction

The basic idea of using skater pose model is: the skater’s pose, silhouette from the camera point of view, varies implicitly in a certain range around a position on the rink given the similar view angle and position of the camera. Then, we can build a map from location to skater pose by learning a skating video, and perform template matching to refine the blobs of skaters predicted by RF classification.

To build the skater pose model, we manually annotate the position and blobs of a skater skating in a video of short track speed skating competition for one lap (Fig. 3.1a). We then extract the blobs and construct a set of 3D data. By utilizing volume smoothing with Gaussian filter (Fig. 3.1b), we obtain a skater pose model $\Gamma = \{\mathrm{T_s} | \mathrm{s} \in \mathrm{P}\}$, where P

is the set of positions that have been manually annotated in the training video and transformed to the positions in rink model. We can define the template of skater at time step t using $\Gamma(p) = arg\, min_{T_s, s \in P}\,\{||s-p||\}$ where p is the skater position in rink model, and $\Gamma(p)$ is the template of the skater regarding to position p (Fig. 3.1c).



(a) (b) (c)

Figure 3.1. Pose model construction and utilization. Here, p is the predicted skater position of rink model. (a) Skater blobs annotated manually. (b) Skater pose model smoothed on temporal axis. (c) The silhouette template at position p in rink model.

We can then retrieve a silhouette template $\Gamma(p)$ of the mth skater given its position p in rink model. However, in order to deal with the change of view point when the skater's position is different from the training sequence, we have to obtain a subset of templates (Eq. 3.1) around the position p_m and then conduct template matching. Before tracking procedure is finished at current time step, we can only use $p = (\tilde{u}_{x,t}^{pre}, \tilde{u}_{y,t}^{pre})$ to obtain the set of templates.

$$\Gamma(p) = \{T_s \mid s \in P \; and \; ||s-p|| \leq \lambda\} \tag{3.1}$$

where λ is the threshold confining the scope of the region centered at position p. Different from the previous one, here $\Gamma(p)$ is a set of silhouette templates instead of one template w.r.t rink model position p. Without loss of generality, we have $\Gamma(p) = \{T_i | i =$

$1, \ldots, N_{tpl}\}$ for the mth skater, where N_{tpl} is the number of templates of the skater at predicted position p. The threshold λ is set to 50 through experiments.

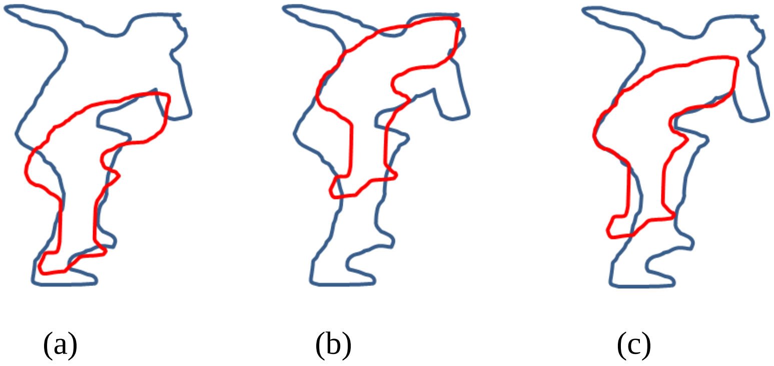


Figure 3.2. Problems of traditional template matching based on edge and silhouette. The red shape represents template, and the yellow region represents target. (a) Template matching based only on edge. (b) Template matching based only on silhouette. (c) Template matching based on both edge and silhouette.

3.2 Template Matching Based on Silhouette and Edge

We will apply template matching based on edge and silhouette, and combine the two matching results to find the optimally matched blobs for each skater. The reason behind this is: the trained skater templates are more or less different from the current skater pose due to the high deformability of the skater in competition. Moreover, because the blob size may vary among skaters, we will apply scaling to the templates while processing template matching. Edge-based matching only takes into account the boundary pixels (Fig. 3.2a) while silhouette-based matching considers only the ratio of overlapping pixels (Fig. 3.2b), hence both may not be able to find the optimal solution. However, by combining the two features, we may have better opportunity to find the optimal (Fig. 3.2c).

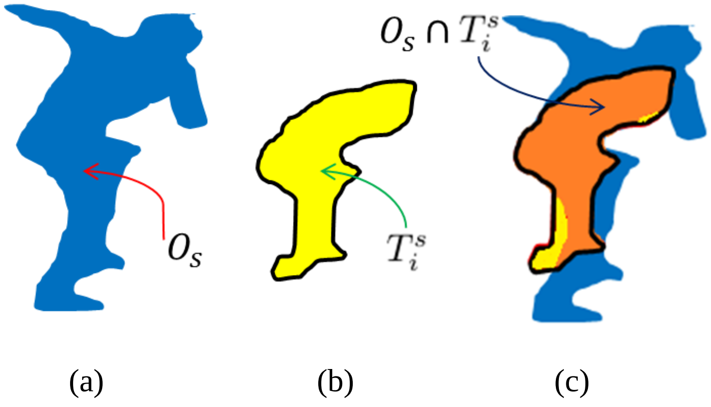


Figure 3.3. Silhouette-based template matching. (a) Observation silhouette. (b) Template silhouette. (c) Silhouette matching.

3.2.1 Silhouette-based Template Matching

Given a silhouette observation O_s and a silhouette template T_i^s, we will drift the template on the observation, and calculate each move of the template (Fig. 3.3). We perform pyramid processing to lower the matching complexity. The likelihood between the observation and the template is proportional to the overlapping rate of the two silhouettes, which can be measured by Eq. 3.2.

$$L_s(O_s, T_i^s) = \frac{1}{\sqrt{2\pi}\sigma_s} exp\left(-\frac{(W_s(O_s, T_i^s) - 1)^2}{2\sigma_s^2}\right) \tag{3.2}$$

where $L_s(O_s, T_i^s)$ is the likelihood of matching O_s and T_i^s; σ_s is the standard deviation which is set to 0.05 through experiments; and W_s is the overlapping rate which is calculated according to Eq. 3.3.

$$W_s(O_s, T_i^s) = \frac{|O_s \cap T_i^s|}{|O_s \cup T_i^s|} \tag{3.3}$$

3.2.2 Edge-based Template Matching

The edge template can be easily obtained by performing edge detection algorithm on a silhouette template. Given $\hat{\Gamma}(p) = \{T_i^e | i = 1, \dots, N_{tpl}\}$ is the set of edge templates corresponding to the silhouette template set $\Gamma(p)$, the likelihood of edge-based template matching (Eq. 3.4) is measured by using a revised Chamfer matching [49] algorithm.

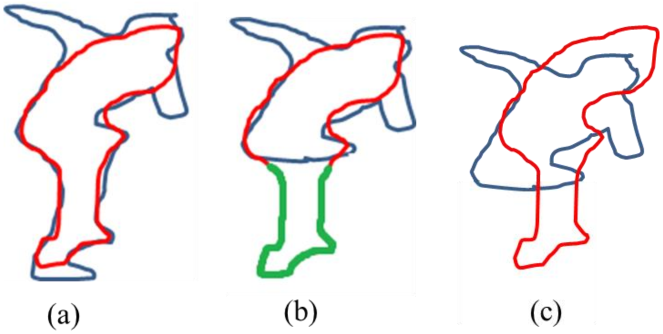


Figure 3.4. Occlusion problem in traditional Chamfer matching. Blue edge represents the observation while the red edge represents the template. (a) Chamfer matching without occlusion. (b) Chamfer matching with occlusion. (c) Mismatch caused by occlusion. The edge shown in green will generate big Chamfer matching score which leads to mismatch as shown in (c).

To handle partial occlusion between skaters, we will revise the standard Chamfer matching which is able to deal with small occlusion but fail on severe occlusion. The problem is illustrated in Fig. 3.4 where severe occlusion of the observation causes mismatch because the occluded part in distance transform image corresponds to big distance score.

We design a variant of Chamfer matching in which only the edge pixels close enough to the observation edge will be used to calculate the Chamfer distance. Thus, the green edge in Fig. 3.4 will not be employed while calculating the Chamfer distance. Moreover, the Chamfer distance should be scaled according to the number of pixels on the template edge which are close enough to the observation edge, here we use N_{sel} to represent the number of selected pixels of which the chamfer distances are smaller than a threshold. Because the bigger N_{sel} is the more confident the optimal matching is achieved, and the scaled Chamfer distance should be inverse proportional to N_{sel}.

Given the edge observation O_e which is obtained by performing edge detection algorithm on the silhouette observation O_s, the likelihood of matching the observation with the template is calculated using Eq. 3.4.

$$L_e(O_e, T_i^e) = \frac{1}{\sqrt{2\pi}\sigma_e} exp\left(-\frac{\left(W_e(O_e, T_i^e)\right)^2}{2\sigma_e^2}\right) \tag{3.4}$$

where L_e is the likelihood of matching O_e and T_i^e; σ_e is the standard deviation which is set to 0.25 through experiments; and W_e is determined by Eq. 3.5 given the observation set $O_e = \{o_j\}$ and template set $T_i^e = \{t_i^k\}$.

$$W_e(O_e, T_i^e) = \frac{1}{\left|\hat{T}_i^e\right|} \sum_{t_i^k \in \hat{T}_i^e} \min_{o_j \in O_e} \left|t_i^k - o_j\right| \tag{3.5}$$

where $\hat{T}_i^e = \left\{t_i^k \,\middle|\, \min_{o_j \in O_e} \left|t_i^k - o_j\right| < \epsilon\right\}$ and ϵ is the threshold (set to 5 through experiments) determining if the two edge pixels t_i^k and o_j are close enough.

3.2.3 Combination of Edge and Silhouette

In order to lower the computational complexity of matching, we will apply a pyramid processing in which the optimal location of a template is found in the top-down layers (Fig. 3.5). The solution searching space is greatly reduced by doing so. To combine the edge and silhouette features, we calculate the combined likelihood of the two likelihood measurements for each of the templates according to Eq. 3.6.

$$L^\delta(T_i) = L_s^\delta(O_s, T_i^s) \cdot L_e^\delta(O_e, T_i^e) \tag{3.6}$$

where $L^{\delta}(T_i)$ is the likelihood of the ith template T_i with the pyramid scale δ; L_s^{δ} and L_e^{δ} are likelihood measurement of silhouette and edge with pyramid scale δ. In pyramid processing, the observation and the template are scaled to $1/2^{\delta}$ of the original size, and then likelihood measurements are performed based on Eq. 3.2 and Eq. 3.4. And the optimal position of each template corresponds to the maximum of the likelihoods.

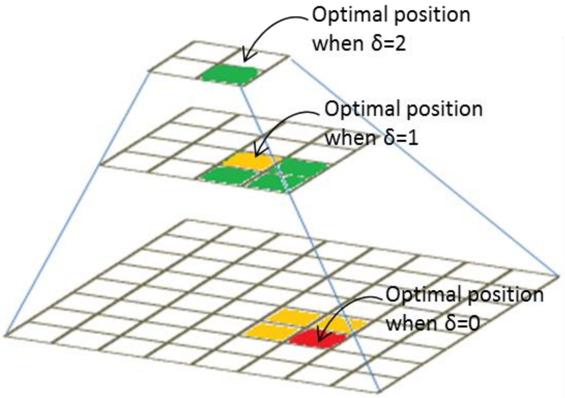

Optimal position when δ=2
Optimal position when δ=1
Optimal position when δ=0

Figure 3.5. Pyramid processing of template matching.

After template matching process, we have a set of triplets of the mth skater $\Psi_m = \{(T_i^m, S_i^m, \hat{L}_i^m) | i = 1, \dots, N_{tpl}^m\}$ where T_i^m, S_i^m and $\hat{L}_i^m$ are template, state of the template (optimal position of pixels of the template on the observation) and the likelihood corresponding with the optimal state of the template, respectively. Based on the set Ψ_m obtained above, we propose a voting-based method to find the semi-optimal state $\hat{S}_m$ of the mth skater which is a set of pixels on the skater. Given that the dimension of the mth skater's observation O_m is $r_m \times c_m$, we create a set of $r_m \times c_m$ matrices $\{A_i^m | i = 1, \dots, N_{tpl}^m\}$ where we set each matrix according to the Eq. 3.7.

$$(A_i^m)_{x,y} = \begin{cases} \hat{L}_i^m & if\ (x,y) \in S_i^m \\ 0 & if\ (x,y) \notin S_i^m \end{cases} \tag{3.7}$$

Then, we will accumulate the matrices and get the vote map $\hat{A}_m = \sum_{i=1}^{N_{tpl}^m} A_i^m$ where pixels on the skater will have higher value (e.g. more votes), while pixels outside skater will have lower value (e.g. less votes). The basic procedure is illustrated in Fig. 3.6 where pixels with different colored blobs in each matrix represent different likelihood (e.g. votes). The semi-optimal state of the *m*th skater is estimated according to Eq. 3.8.

$$\hat{S}_m = \{(x, y) | (\hat{A}_m)_{x,y} > \omega \max(\hat{A}_m)\} \tag{3.8}$$

where $\hat{S}_m$ is a set of pixels with position (x, y) in the observation of the skater. The threshold ω is set to 0.5 through experiments, the larger ω is the more pixels in $\hat{A}_m$ will be selected.

The blob set $\hat{S}_m$ is as the semi-optimal because it is the optimal blob set voted by all the templates, but may either contain the blobs that do not belong to the object or miss the blobs that are part of the object due to occlusion and internal difference between the template and observation. In order to estimate the optimal state $\tilde{S}_m$, we propose a blob growing method to correct the blobs found according to Eq. 3.8 and handle the occlusion among skaters.

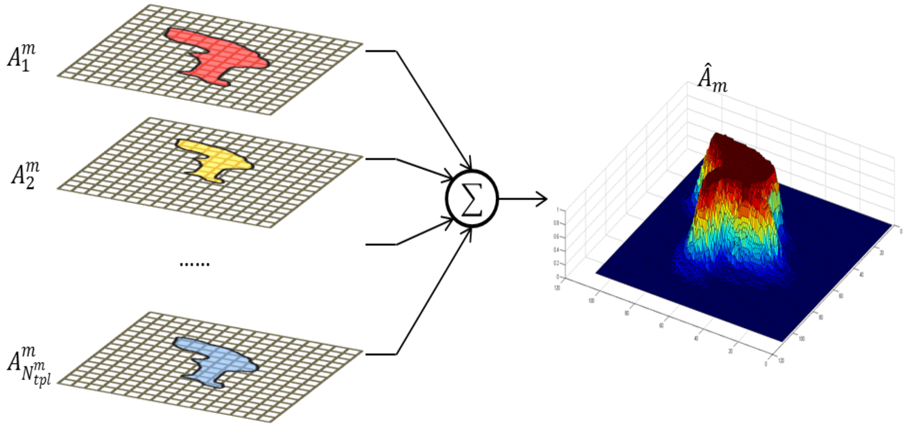


Figure 3.6. Voting-based skater semi-optimal state estimation.

CHAPTER 4

OCCLUSION AND SKATER REAPPEARANCE HANDLING

Short track speed skating is a form of game where several players compete at very high speed and occlusion happens among skaters. Because only one handycam is used in the CAT system to capture the competition video, the complexity of setting up the system is greatly reduced. However, the slower skaters are sometimes out of the scene while the camera is panning. In this section, we will discuss about the occlusion handling based on blob growing and the reappeared skater capturing based on color voting.

4.1 Occlusion Handling Based on Blob Growing

4.1.1 Acquisition of Seed Blob

As mentioned in the previous section, it is necessary to perform blob correction to obtain the optimal state of each skater. The basic idea of blob correction is to get the seed blob of a skater which is most likely to belong to the object and grow it to find the optimal state of a skater. Blob growing method includes two aspects which are inspecting the neighboring pixels and merging the pixels with similar gray level. Meanwhile, the occlusion will be handled in the blob growing procedure based on the occlusion relationship determined using the global spatial information.

The seed blob is obtained by applying “and” operation between the semi-optimal states (Eq. 4.1). However, occlusion has to be considered when obtaining the seed blobs because the adjacent parts between two occluding skaters may affect the blob growing procedure. We perform a dilate operation on the correlated seed blobs to remove the overlapped blobs and make the seed blobs grow according to the occlusion relationship.

$$B_m = \hat{S}_m \cap O_s^m \tag{4.1}$$

where the seed blobs B_m of the mth skater is a set of pixels belonging to the intersection of sets $\hat{S}_m$ and O_s^m which are the semi-optimal state and the silhouette observation of the mth skater, respectively.

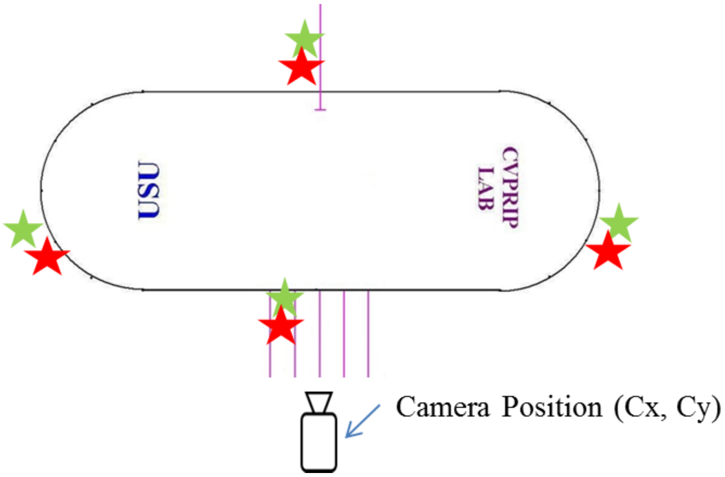


Figure 4.1. Occlusion model. In any case, the red skater is occluding green skater because it is closer to the camera.

Given the seed blobs B_i and B_j of two skaters who are occluding with each other, the spatially conflicting pixels in the seed blobs will be removed and new seed blobs will be generated. The procedure of finding and removing conflicting pixels in the two sets is described by Eq. 4.2 and Eq. 4.3.

$$\dot{B}_i = B_i \backslash ((B_i \oplus D) \cap (B_j \oplus D)) \tag{4.2}$$

$$\dot{\mathrm{B}}_\mathrm{j} = \mathrm{B}_\mathrm{j} \backslash ((\mathrm{B}_\mathrm{i} \oplus \mathrm{D}) \cap (\mathrm{B}_\mathrm{j} \oplus \mathrm{D})) \tag{4.3}$$

where D is a structuring element and its size is set to 5 through experiments; and $\oplus$ is dilate operator which is defined as $\mathrm{X} \oplus \mathrm{D} = \bigcup_{\mathrm{d} \in \mathrm{D}} \mathrm{X}_\mathrm{d}$ given the binary image X.

```
Algorithm: Blob growing
Input: Seed blobs $\dot{B}_m$ $(m = 1, \dots, N_{obj})$ of skaters
Output: Optimal blobs $\tilde{S}_m$ $(m = 1, \dots, N_{obj})$ of skaters

Initialize $\tilde{S}_m = \dot{B}_m$ $(m = 1, \dots, N_{obj})$
FOR  each iteration $i$ = 1,…, $I_{max}$
   FOR  each skater $m$ = 1,…, $N_{obj}$
      Update edge pixel set $E_m^i$ using $\tilde{S}_m^{i-1}$;
      Update the 8-neighbors set $\Psi_m^i = \{N_8(C_j)|\ C_j \in E_m^i\ (j = 1, \dots, |E_m^i|)\}$;
      IF  $i$ is divisible by 3
         Check all pixels $X \in \Psi_m^i$, $X \notin \tilde{S}_m^{i-1}$ and $X \in \tilde{S}_k$ $(k = Occ(m))$;
         And, $\tilde{S}_m^i = \cup X$ if $|G(X) - G(Center(X))| < \varepsilon$
      ELSE
         Check all pixels $X \in \Psi_m^i$, $X \notin \tilde{S}_m^{i-1}$ and $X \notin \tilde{S}_k$ $(k = Occ(m))$;
         And, $\tilde{S}_m^i = \cup X$ if $|G(X) - G(Center(X))| < \varepsilon$;
      END
      IF $\left||\tilde{S}_m^i| - |\tilde{S}_m^{i-1}|\right| < \omega$
         Set finish growing flag to the mth skater;
      END
   END
END
```

Figure 4.2. The algorithm of blob growing.

4.1.2 Occlusion Relationship

The occlusion relationship provides useful information describing the spatial relationship between two objects occluding with each other. It is analyzed based on the idea that the object closer to the camera is more likely to occlude the others. We define a camera position according to the rink model (Fig. 4.1).

Moreover, according to the process discussed in the previous section, we perform dilate operation on the blobs detected in template matching, and check if there is conflict between the blobs of two skaters. If so, the skaters are spatially conflicting, and the one closer to the camera is occluding the other.

4.1.3 Blob Growing

Blob growing algorithm will be performed based on the seed blobs found in previous procedure to estimate the optimal state $\tilde{S}_m$ $(m = 1, \ldots, N_{obj})$ of each skater. The blob growing procedure is illustrated in Fig. 4.2 where $Occ(m)$ retrieves the skater who is occluding the mth skater; the threshold ε determines if the two pixels have similar gray level; and the threshold ω indicates when to stop growing the blobs of a skater. The thresholds ε and ω are set to 30 and 10 through experiments, respectively.

The boundary pixels of the seed blob are obtained first, and the 8-neighbors of each pixel on the boundary will be studied and the ones whose gray level is close enough to the center pixel will be merged into the blob set of the corresponding skater. The neighboring pixels belonging to the conflicting area of two occluding skaters are dealt with specially. If an occluded skater is under consideration, it will grow slower in the conflicting area than in the non-conflicting area. The boundary will then be updated, and the study of boundary pixel neighbors will be executed again for the next iteration. The procedure will be iterated until most of the boundary pixels have no more neighboring pixels.

4.2 Skater Reappearance Handling

When capturing the competition videos, the camera is panned quickly to follow the fast skaters, and the slowest skaters may be lagged too far to be captured by the camera. However, it is important to track the skaters when they reappeared. The difficulties of tracking the reappeared skaters include: 1) a skater may disappear at one location and reappear at another location with different appearance; and 2) multiple skaters may be out of the scene and reappear at different times, therefore the algorithm needs to not only detect the reappeared skater, but also assign it with correct identity.

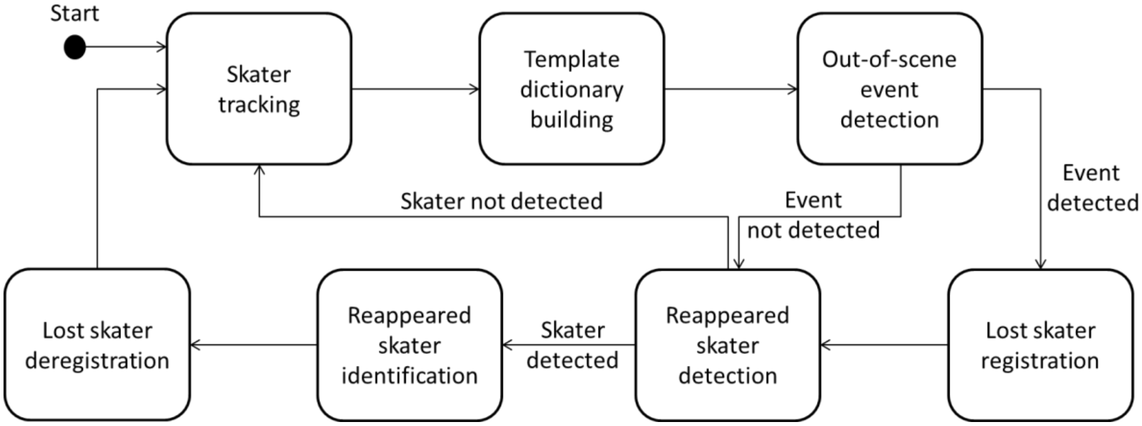

Start
Skater tracking
Template dictionary building
Out-of-scene event detection
Event detected
Skater not detected
Event not detected
Lost skater deregistration
Reappeared skater identification
Skater detected
Reappeared skater detection
Lost skater registration

Figure 4.3. Skater disappearance and reappearance handling state machine.

In this section, we proposed a template dictionary updating and color voting-based method to detect and identify the skater reappeared from outside of the scene. We employ a state machine for each skater to handle the disappearance and reappearance (Fig. 4.3). After the skater tracking process introduced in previous sections is executed, the template dictionary of each skater is updated. Then, an out-of-scene event detection procedure is performed to detect if a skater is out of the scene. The tracking procedure will be continued for the skaters still in the scene, and the out-of-scene skater will be registered as lost. Meanwhile, the template dictionary updating procedure of the lost skater will be stopped, and the reappeared skater detection process will be initiated. The reappeared skater identification procedure will not be conducted until a lost skater is detected. The identified skater will be deregistered from the lost skater table before transferred back to skater tracking procedure.

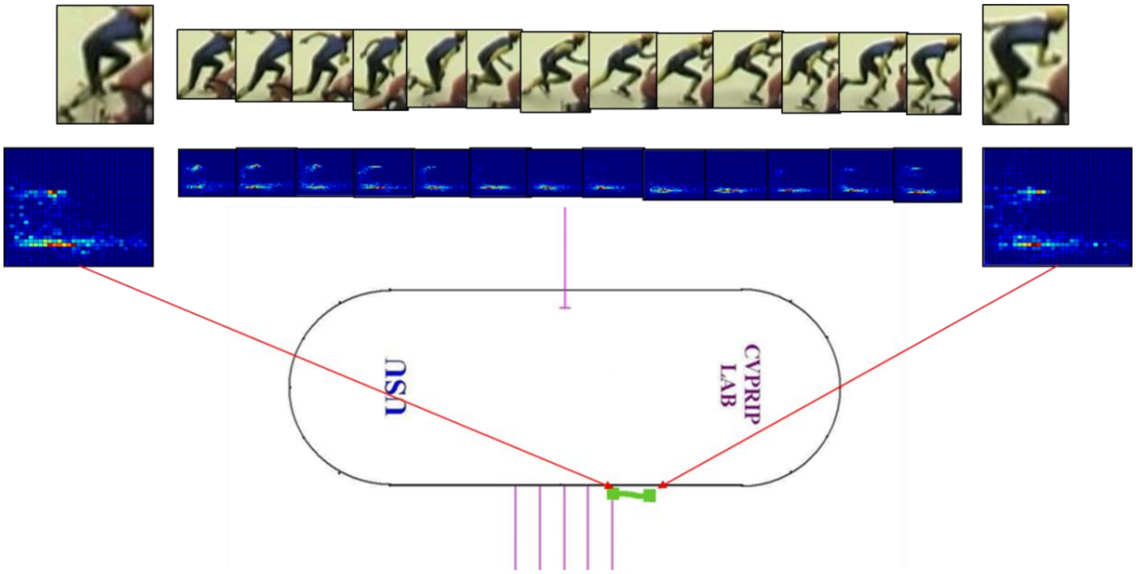


Figure 4.4. Online template dictionary construction of a skater. The top row represents the tracked skater bounding box. The second row denotes the HS 2D histogram generated according to the pixels of the skater in the bounding box. The third row indicates the rink position of the corresponding templates.

4.2.1 Template Dictionary

Due to the fact that a skater's appearance varies greatly during competition, we build a template dictionary for each skater dynamically while tracking, and record the tracked rink position of each skater which is served as the index for the corresponding template dictionary. We represent the template dictionary of the mth skater with Φ_{m} and

$$\Phi_m = \{\tau_p | p \in \mathrm{I}_{\mathrm{rink}}\} \tag{4.4}$$

where τ_p indicates the hue-saturation (HS) 2D color histogram of the ith skater, and p is the skater position in rink image I_{rink}. In this paper, the hue-saturation-value (HSV) color space was used to build the histogram template, without using channel V, to make the algorithm less sensitive to the lighting condition.

When tracking the skaters, the tracked blob of each skater $\tilde{S}_m$ will be used to obtain the pixels on the original image which are utilized for generating the HS 2D color histogram. The online template dictionary construction of a skater is illustrated in Fig. 4.4. In the template dictionary, each template τ_p can be retrieved by a rink position p.

4.2.2 Reappeared Skater Detection and Identification

After the tracking procedure, the out-of-scene event detection will be performed to determine if a skater is out of the scene by checking if the tracked skater bounding box lies on the boundary of the camera scene. If a skater m is detected on the boundary of the scene, the skater identity will be registered to a lost skater set by $S_l = S_l \cup \{m\}$. When the set S_l is not empty, there is at least one skater out of the scene and the reappeared skater detection procedure will be performed to detect if a skater reappears.

In the reappeared skater detection procedure, background subtraction is performed to obtain the candidate blobs. A track model is utilized to check if the candidate blob is on the track. If so and if the detected blob is large enough to be a skater, a color voting based reappeared skater identification procedure will be executed to assign correct skater identity to the detected blob (Eq. 4.5).

$$k = \underset{m \in S_l}{arg\min}\, W_B(h_s, \underset{\tau_p \in \Phi_m}{arg\min} \|s - p\|) \tag{4.5}$$

where k is the estimated reappeared skater identity; h_s is the HS 2D histogram of the detected candidate skater at rink position s; τ_p is the skater template in the skater template dictionary; and $W_B()$ is the Bhattacharyya distance [50] of two HS 2D histograms, which is defined by Eq. 4.6.

$$W_B(h_s, \tau_p) = \sqrt{1 - \sum_{u=1}^{U} \sqrt{h_s(u), \tau_p(u)}} \tag{4.6}$$

where W_B is the Bhattacharyya distance and the smaller the it is, the more similar the histograms are; U is the number of bins of each histogram.

Based on Eq. 4.5, we perform the reappeared skater identification. After the candidate blob of a skater is detected by background subtraction, the rink position of the blob centroid s is calculated. For each template τ_p in dictionary Φ_m where $m \in S_l$, the distance between p and s will be examined, and the template closest to s will be selected for color voting. Therefore, there will be only one selected voter for each lost skater identity m in S_l. The Bhattacharyya distance W_B between the HS 2D histograms of the detected reappeared skater and each of the voters will be calculated, and the one with smallest distance will be chosen whose dictionary index m will be used as the reappeared skater identity. Finally, the reappeared skater identity will be deregistered from S_l and the tracking procedure will be performed to the reappeared skater.

CHAPTER 5

EXPERIMENTS AND ANALYSIS

The effectiveness and robustness of the proposed methodology are proved through experiments. We design the experiments to prove that the algorithm works robustly; and to prove that it works better than state of the art methods. Two video sequences of skaters in competition are utilized for experiments. We apply the proposed method to tracking multiple skaters in the testing video sequences, and analyze the outputs including skaters' trajectories, velocities, tracked blobs, etc. We also execute the existing published state of the art algorithms on the video sequences, and compare the tracking results with that of the proposed algorithm qualitatively and quantitatively.

5.1 Tracking Result

We test the proposed method on three video sequences including "Men's 500 meters," "Women's 500 meters" and "Men's 500 meters semifinal" containing 1071 frames, 1156 frames and 1084 frames, respectively. The "Men's 500 meters semifinal" contains the scenario where two skaters skate out of the scene and reappear frequently. The method is executed on a desktop with Intel(R) Core(TM) I5 2.20GHz CPU and 4GB memory. The average processing time of the video sequences is 5.7 seconds per frame. However, as a comparison, STS takes 18.7 seconds per frame on average.

The tracked blobs of skaters are presented in Fig. 5.2 and Fig. 5.3. We observe that the scale of skaters change drastically in the video sequences, and complicated occlusion happens frequently among skaters. In order to better exhibit the capability of the proposed method to resolve the challenges, we present the skaters in each of the original frames. In Fig. 5.2, the frames 256, 337, 471, 537, 565, 643, 687, 906, etc. show severe occlusion among skaters, and we find that the proposed method can accurately track the blob of each skater even though when the skater is occluded by another skater wearing the same uniform. In Fig. 5.3, similar situation is exhibited in frames 81, 113, 144, 201, 271, 373, etc. Moreover, we notice that sometimes occlusion happens among skaters when their scales are very small referring to frames 175, 643 and 1043 in Fig. 5.2, and frames 201 and 727 in Fig. 5.3. But the proposed tracker is still able to achieve satisfactory tracking result.

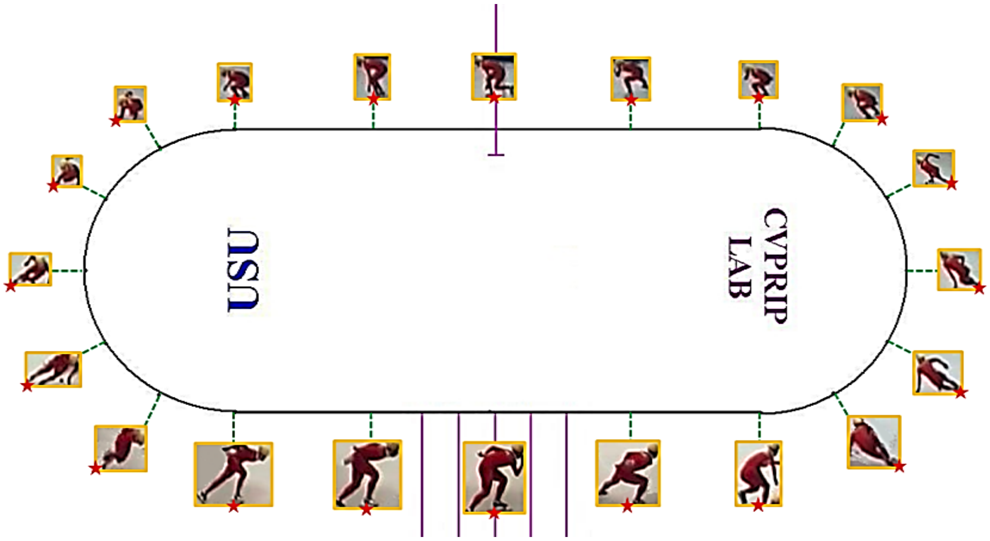


Figure 5.1. Skater image position selection based on rink position. The red star represents the image position selected to calculate the corresponding rink model position. The orange box indicates the skater bounding box tracked in image.

As mentioned previously, the trajectories of skaters convey valuable information to improve their strategies. We calculate skater location in rink model and generate the trajectory for each of the skaters. Instead of using center location of a target, we take into account the skater location on ice (Fig. 5.1). After a skater's bounding box is obtained through tracking, the image position u_t^{image} is figured out by considering the previous location u_{t-1}^{rink} in rink model (Eq. 5.1).

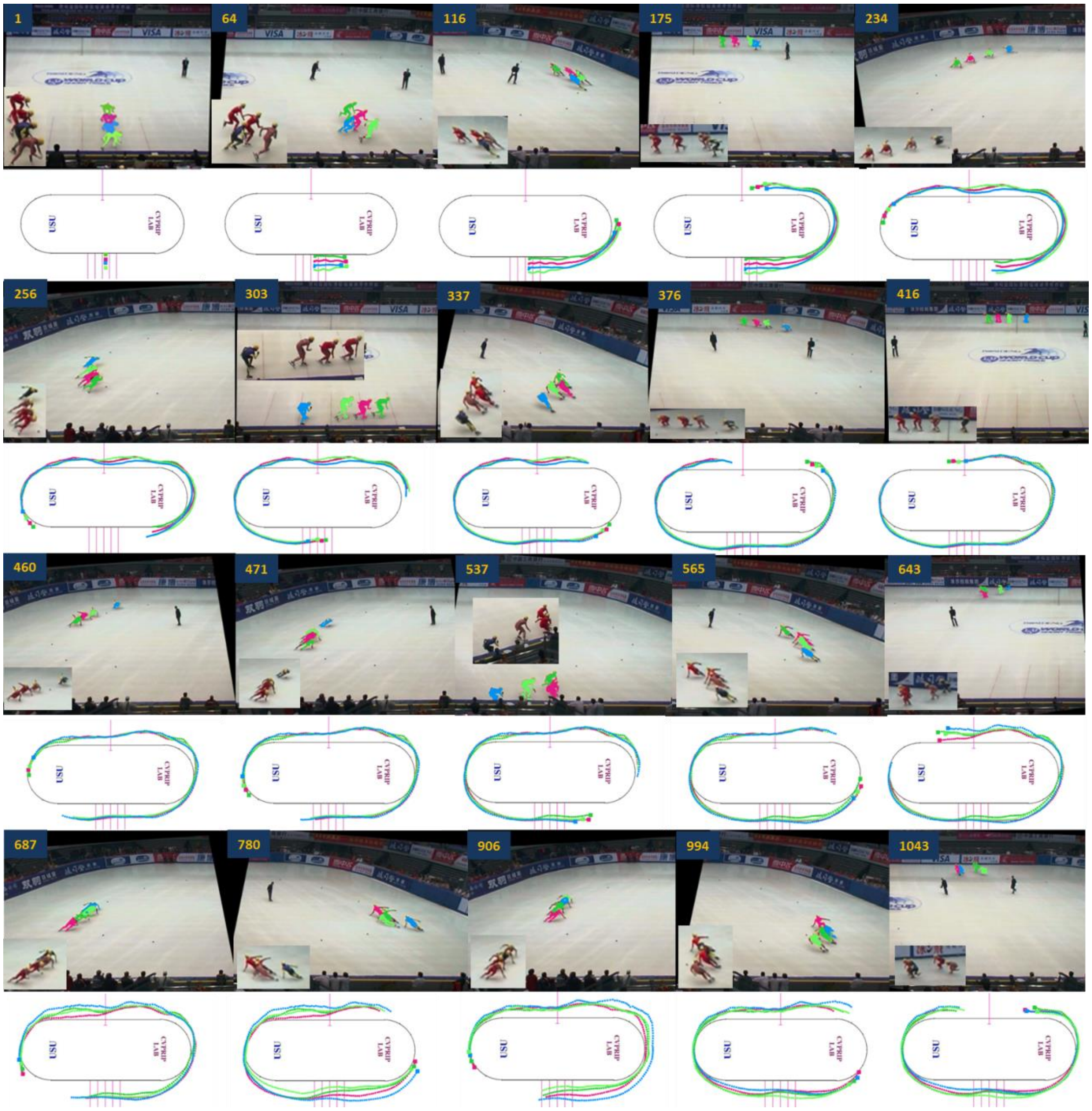


Figure 5.2. Tracking results of the video Men’s 500 meters competition. The video contains 1071 frames. The upper row is the tracked skater blobs with the reference of skaters in the original frame. The lower row represents the trajectories of each of the skaters.

$$u_t^{image} = \begin{cases} (x_{bb}\,, y_{bb} + h_{bb}) & u_{t-1}^{rink} \in P_{LC} \\ \left(x_{bb} + \dfrac{w_{bb}}{2}, y_{bb} + \dfrac{h_{bb}}{2}\right) & u_{t-1}^{rink} \in P_S \\ (x_{bb} + w_{bb}\,, y_{bb} + h_{bb}) & u_{t-1}^{rink} \in P_{RC} \end{cases} \tag{5.1}$$

where $(x_{bb}, y_{bb}, w_{bb}, h_{bb})$ represents the skater bounding box in image, (x_{bb}, y_{bb}) is the coordinates of the top left corner; (w_{bb}, h_{bb}) indicates the dimension of the bounding box; and P_S, P_{LC} and P_{RC} are point sets representing the points in straight track, left curve and right curve, respectively. The procedure is illustrated in Fig. 5.1. The skater rink model location at current time step is calculated according to Eq. 5.2.

$$u_t^{rink} = c\widetilde{H} \cdot \left(u_{x,t}^{image}, u_{y,t}^{image}, 1\right)^T \tag{5.2}$$

where $\widetilde{H}$ is translation from panorama to rink model. We see in Fig. 5.2 and Fig. 5.3 that the trajectories of skaters are smooth and accurate.

Besides, we calculate the velocities of each of the skaters according to Eq. 5.3.

$$V_t = c\left|u_{t+1}^{rink} - u_t^{rink}\right| \cdot f \tag{5.3}$$

where f is the frame rate of the video in frames per second; u_t^{rink} is the tracked skater location in rink model at the current time step t; $\left|u_{t+1}^{rink} - u_t^{rink}\right|$ is the image distance in pixel; and c is a constant ratio indicating the number of pixels per meter.

We present the calculated velocities of skaters in Fig. 5.4 where we observe several facts: 1) the skaters' directions vary drastically at the beginning because they are trying to accelerate by quickly pushing their legs one after another. However, they still need to finish the first lap to reach the maximum speed; 2) they always slow down when they are going into the corner, and accelerate when they are going out of the corner; 3) the speed of a male skater is obviously greater than that of a female skater by comparing the skater speed in Fig. 5.4a and Fig. 5.4b.

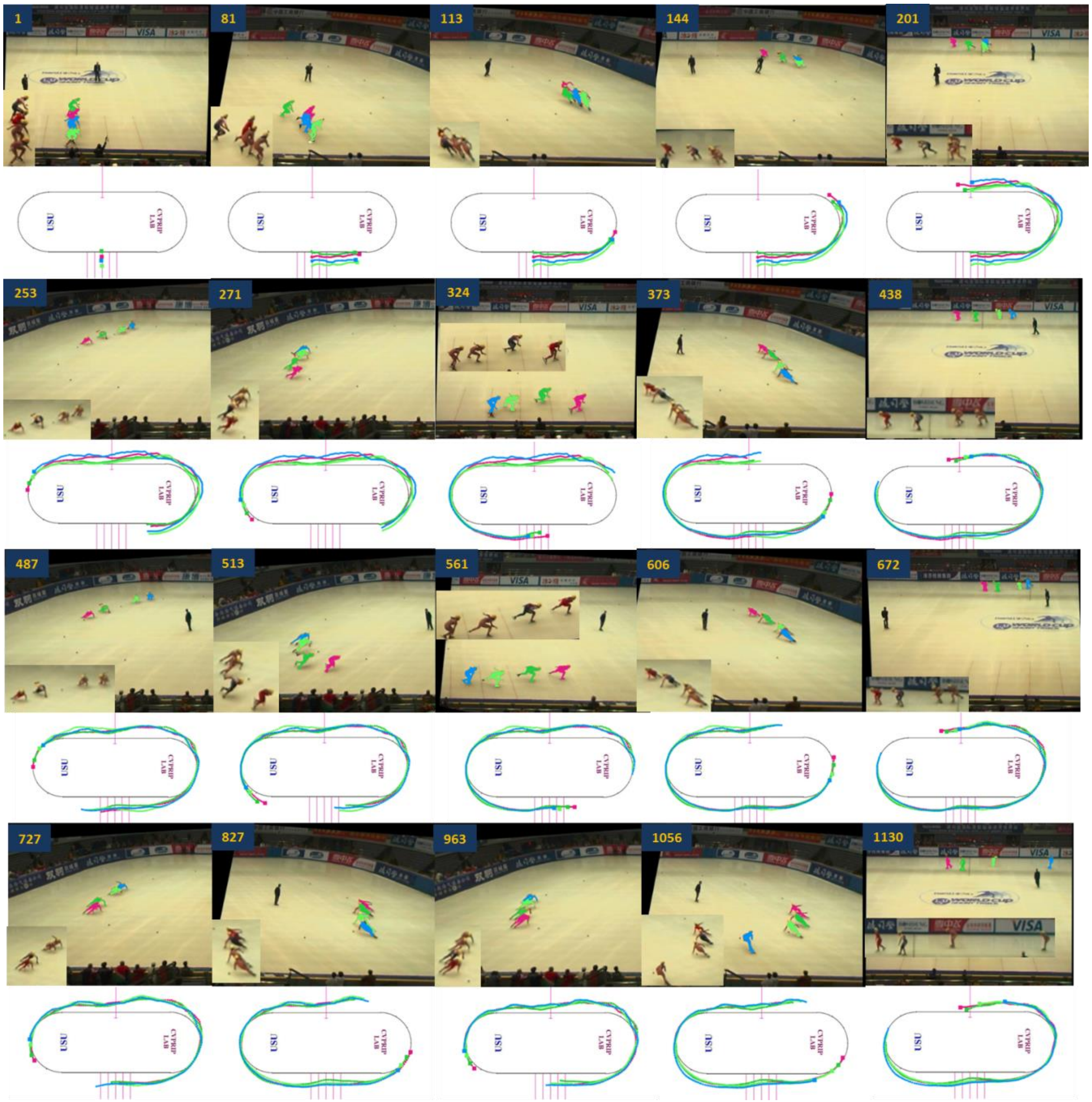
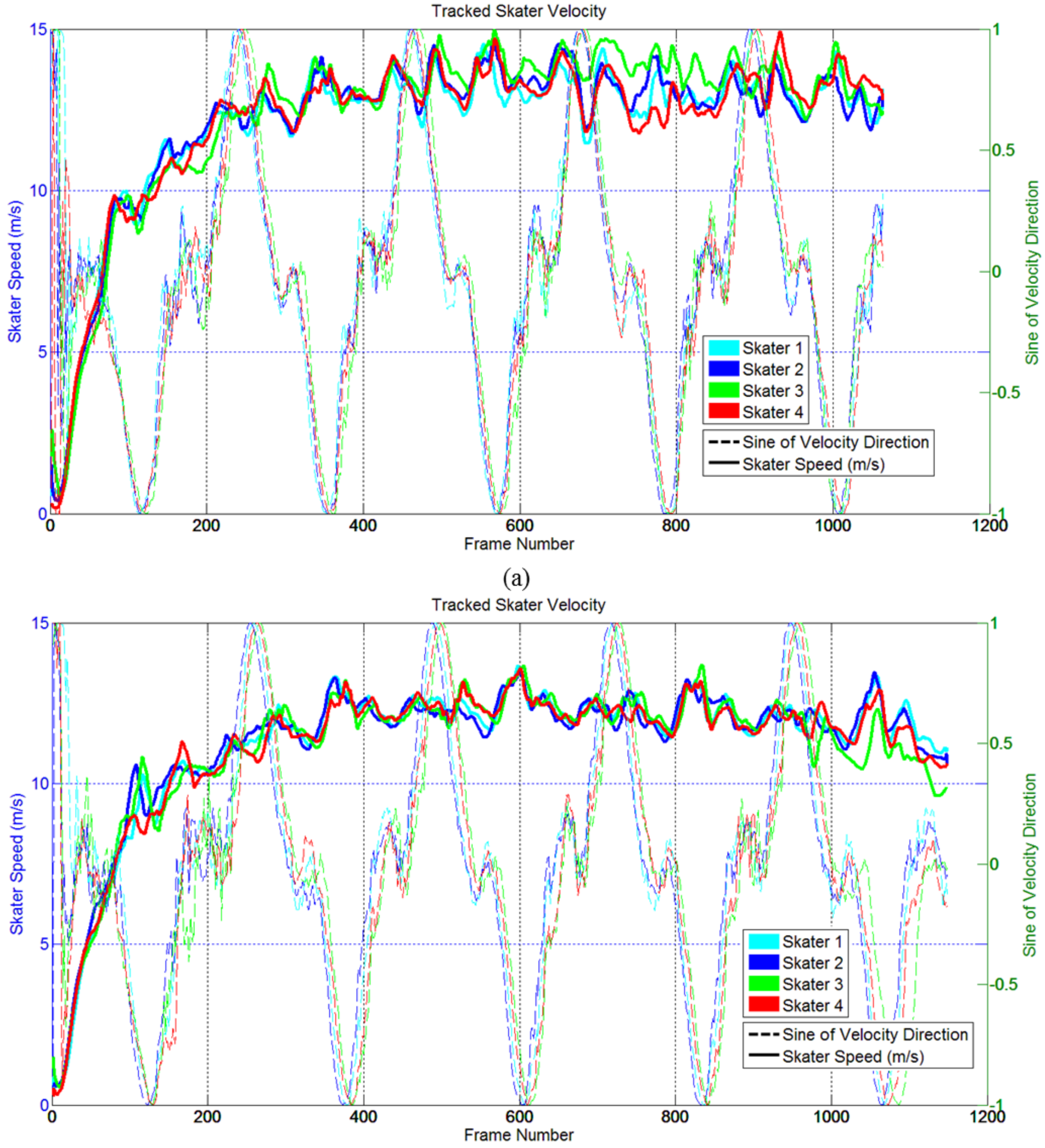


Figure 5.3. Tracking results of the video Women’s 500 meters competition. The video contains 1156 frames. The upper row is the tracked skater blobs with the reference of skaters in the original frame. The lower row represents the trajectories of each of the skaters.

Moreover, we calculate the average speed of each skater and exhibit the result in Fig. 5.5. We notice that the skater maintaining higher average speed in competition tends to have better standings. Therefore, we may utilize the average speed calculated in the past laps and the current position of the skater to estimate the remaining time to the end point by “$remaining\ time = remaining\ distance/average\ speed$.” In Fig. 5.5a, the predicted standings at the end of the fourth lap is “Skater 2, Skater 3, Skater 1 and Skater 4,” while the correct standings at the end point is “Skater 2, Skater 3, Skater 4 and Skater 1.” The error is because the skaters in Men’s video are so close to each other and their speeds are similar too. In Fig.5.5b, the predicted standings at the end of the fourth lap is “Skater 2, Skater 1, Skater 4 and Skater 3” which is identical to the standings at the end point.

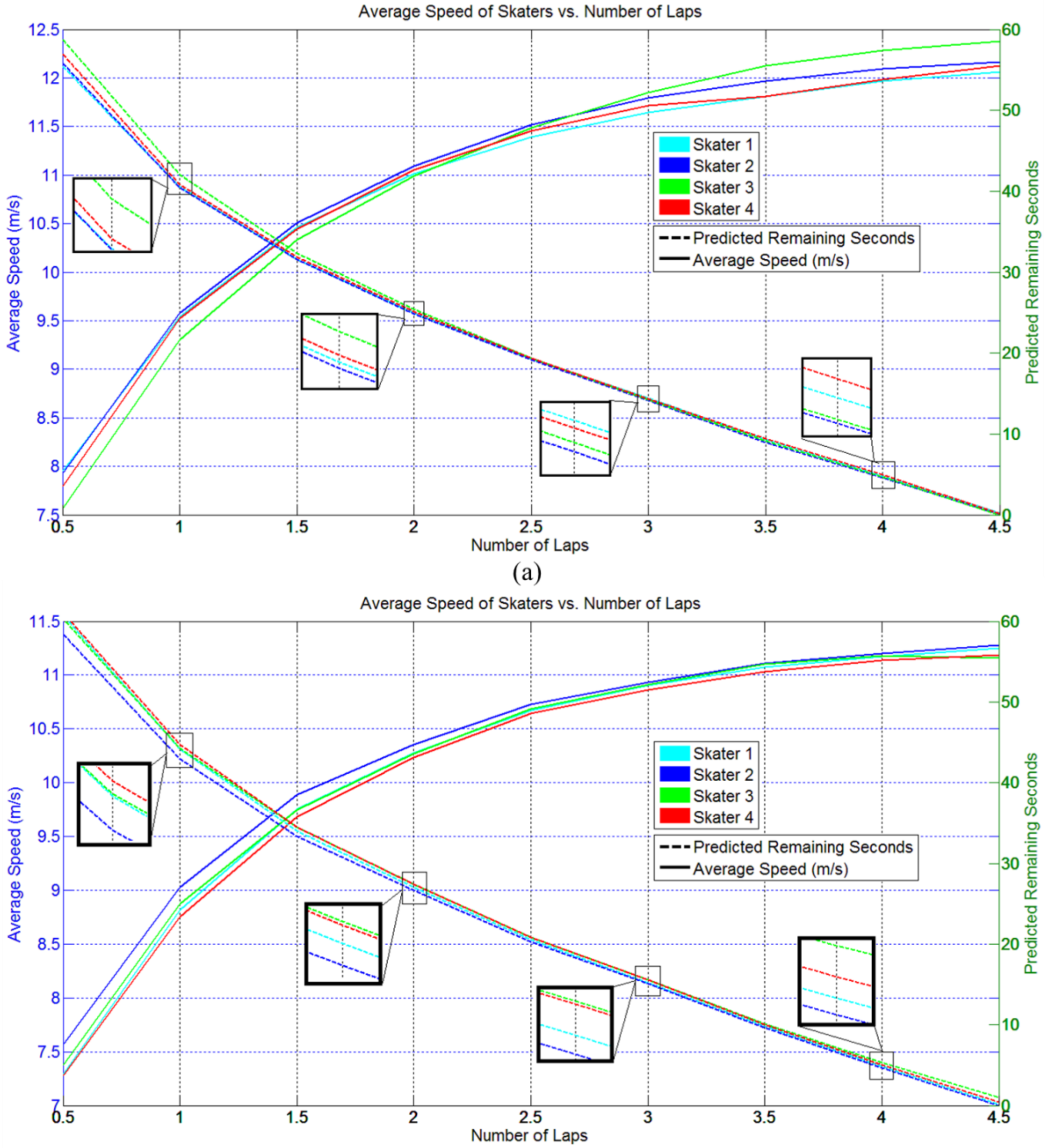
We also tested the reappeared skater detection and identification algorithm on the video “Men’s 500 meters Semifinal” where two skaters disappear and reappear frequently. The tracking result regarding to this manner is shown in Fig. 5.6. We see that the deep green skater is out of the scene in the frames 493, 543, 620, 702, 774, etc., and reappear in the frames 573, 808 and 1044. The blue skater is out of the scene in frame 744 and reappears in frame 774. The proposed algorithm is able to identify the reappeared skaters correctly. In frame 744, both deep green skater and blue skater are out of the scene, and the proposed algorithm can still successfully identify the skaters when they reappear sequentially.



(a)

(b)

Figure 5.4. Velocities of tracked skaters. Velocity consists of quantity (speed) and direction which are represented by solid line and dashed line, respectively. (a) Skaters' velocities of video "Men's 500 meters." (b) Skaters' velocities in video "Women's 500 meters."



(a)

(b)

Figure 5.5. Average speed of skaters calculated every half of a lap with predicted remaining time in seconds. (a) Average speed of video “Men’s 500 meters.” (b) Average speed of video “Women’s 500 meters.”

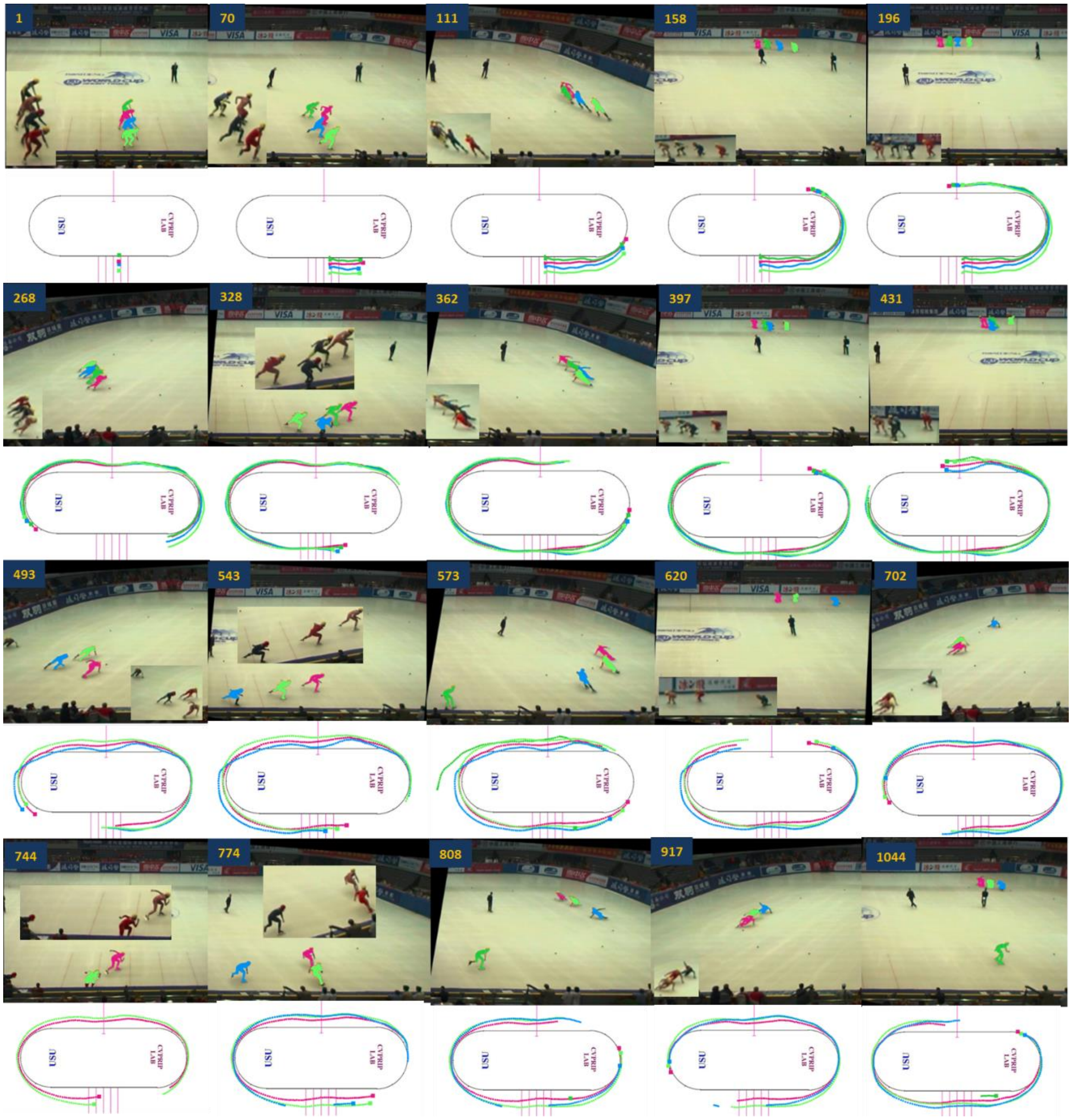


Figure 5.6. Tracking results of the video Men's 500 meters semifinal competition. The video contains 1084 frames. The upper row is the tracked skater blobs with the reference of skaters in the original frame. The lower row represents the trajectories of each of the skaters.

5.2 Comparison

In this section, we will compare the tracking results of the proposed algorithm with state of the art tracking algorithms. To the best of our knowledge, the only multiple skaters tracking algorithm published in recent years is STS [6]. We made a qualitative comparison to it with respect to the tracked blobs and the corresponding result is shown in Fig. 5.7. In frames 48, 153 and 211, the STS allocates large amount of pixels wrongly to the skater wearing the same type of uniform when the two skaters are close to each other. The skaters in frame 111 are close to each other, and the skaters in frame 288 are far from each other. However, the STS still classifies some pixels wrongly to a skater wearing a different uniform. As a comparison, the proposed method classifies pixels perfectly to each skater.

In order to more accurately demonstrate the advantages of the proposed method, we adopt the evaluation methodologies introduced in [51] and quantitatively compare our results with STS as well as recently published general purpose tracking methods. In [51], the authors compare many state of the art tracking methods on the same public data sets. Among these methods, we choose the well performed ones including ASLA [52], CSK [53], CXT [54], CT [55], SCM [56], Struck [57], etc.

Location error is a widely used evaluation metric in tracking. The skater location is calculated according to Eq. 19, and the location error is defined as the Euclidean distance in pixels between the locations of the tracked skaters and the manually labeled ground truths. Then the average location error over all the frames of all skaters in one sequence is used to summarize the overall performance for that sequence. We adopt precision plot as the evaluation metric which shows the percentage of frames whose location error is within the given threshold.

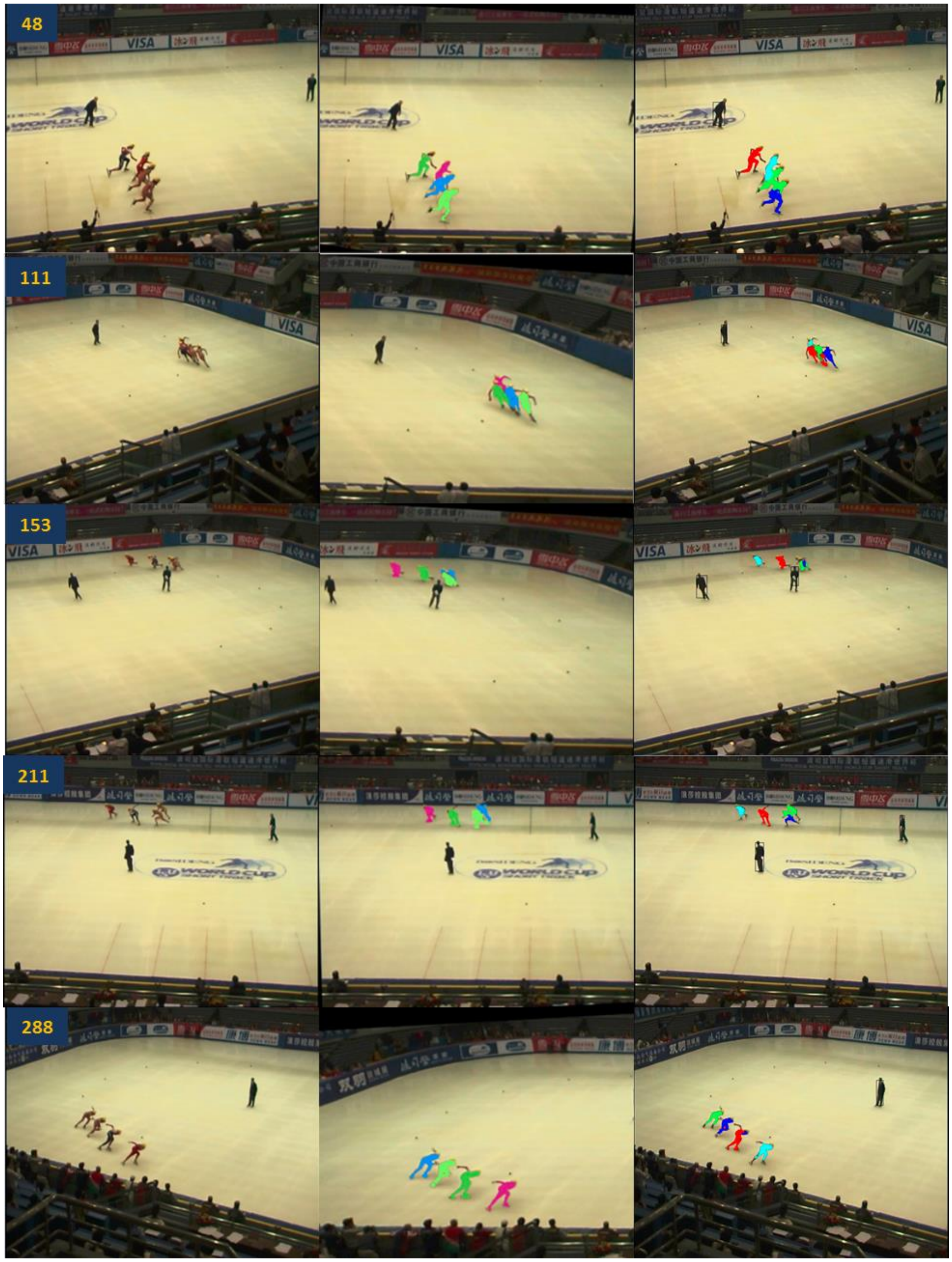


Figure 5.7. Qualitative comparison with method STS. The first column is original frame. The second column is the tracked blobs of the proposed method. The third column is tracked blobs of STS.

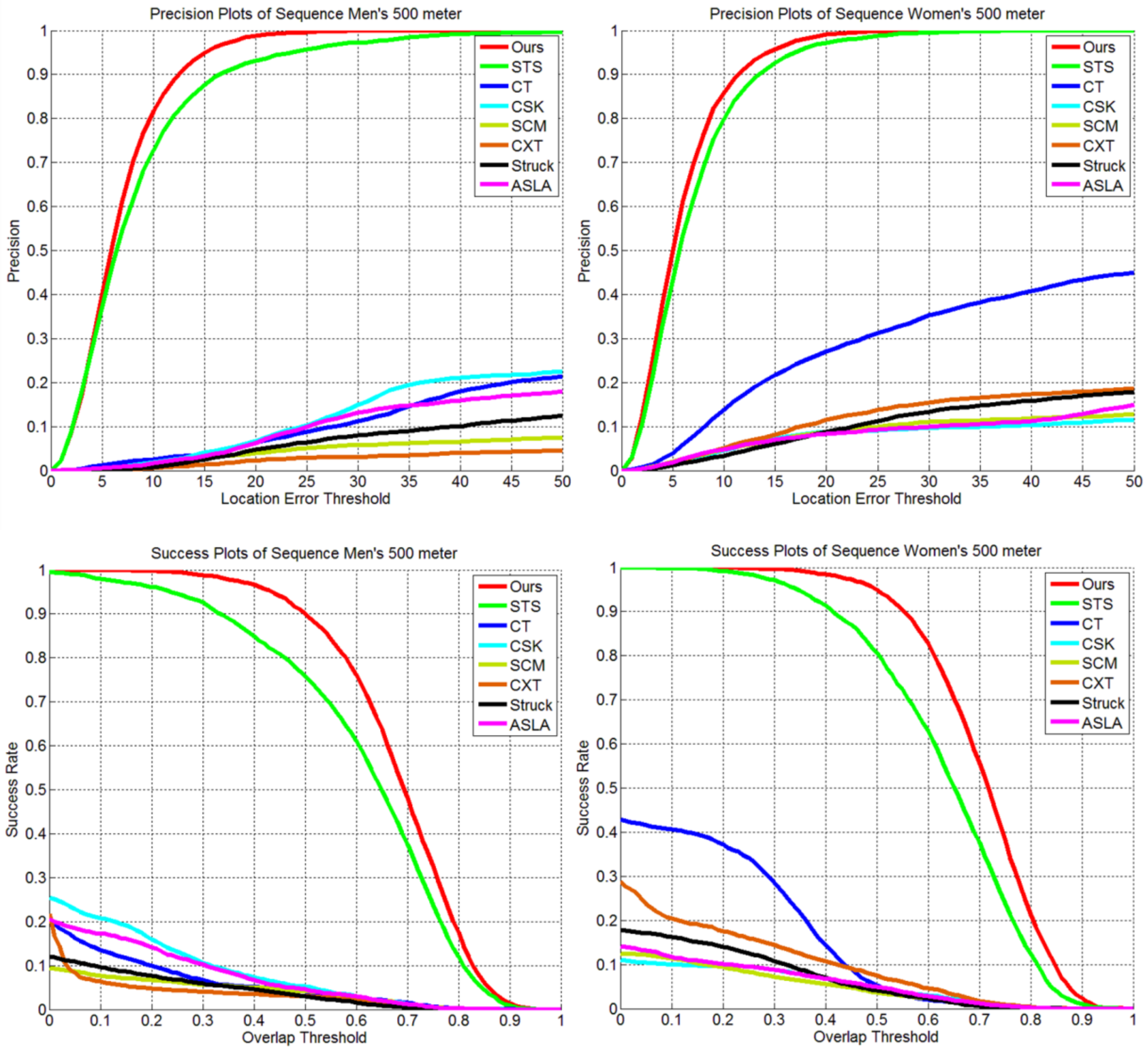


Figure 5.8. Plots of testing sequences. The performance score for different tracker is shown in the legend. The left column and right column illustrate the precision and success plots of sequences “Men’s 500 meters” and “Women’s 500 meters,” respectively.

Bounding box overlap is another evaluation metric which measures the performance based on the overlapping ratio between the bounding boxes of tracked targets and the manually annotated ground truths. The overlap score is defined as $S = \frac{|G \cap T|}{|G \cup T|}$, where G and T represent the ground truth bounding box and tracked bounding box, respectively; $\cap$ and $\cup$ denote the intersection and union of two regions, respectively; and $|R|$ indicates the number of pixels in region R. The success plot exhibits the ratios of successful frames at the thresholds varies from 0 to 1, and “successful frame” represent the frame of which the average overlapping score over all skaters is bigger than the threshold.

The precision plots and success plots of the testing sequences are shown in Fig. 5.8. The testing sequences “Men’s 500 meter” and “Women’s 500 meter” contain 1071 and 1156 frames, respectively. Therefore, ratio 0.01 of frames approximately corresponds to 10 frames.

In precision plots, the proposed method performs slightly better than STS when threshold is smaller than 5 pixels (representing 25 cm in real rink), but is significantly better than STS when threshold is bigger than 5 pixels. For those general purposed tracking methods, CSK and CT perform the best on Men’s and Women’s sequences, but can only achieve around 0.2 and 0.4 at error threshold 35, respectively. The average errors of the proposed method and STS over all frames of both sequences are 6.35 and 7.69, while the corresponding standard deviations of the two methods are 2.54 and 3.98, respectively.

In success plots, the proposed method is significantly better than STS on both sequences at any thresholds. Similarly to precision plots, CSK and CT are the best performing general purposed tracking methods. However, the success rates of them drop below 0.1 at overlap threshold 0.5, while STS and the proposed method have more than 0.75 and 0.9 success rate, respectively.

In summary, from the aspects of precision and bounding box overlap, we found that the proposed method is able to track the skaters more accurately and more robustly than STS. The general purposed state of the art methods fail quickly on the testing sequences due to the challenges mentioned early in this paper.

CHAPTER 6

CONCLUSION

Because of its popularity, short track speed skating has been raising continuous attentions in many countries. A video-based computer-aided training (CAT) system is very helpful for skaters to analyze their performance in the competition and improve their strategies in the future. In response to this demand, we proposed a random forest (RF) based multi cue fusion multiple skaters tracking and analyzing methodology.

The main contributions of the proposed method are: 1) global positional information of skaters is utilized for tracking to increase the tracking accuracy; 2) a new RF based color and positional information fusing and appearance updating method is developed; 3) an improved template matching algorithm is developed to fuse edge and silhouette cues; and 4) the challenging problems including object scale change, severe occlusion, camera motion and non-rigid object tracking are solved in the context of short track.

The proposed CAT system greatly reduces the requirement of hardware settings and the system cost by utilizing only monocular videos captured using a single handycam. Thorough experiments have been made to illustrate the effectiveness and robustness. Both qualitative and quantitative comparisons are made between the proposed CAT system and the state of the art methodologies, and the outperformance of the proposed method has been exhibited. The spatial information of eat skater is accurately generated, which is valuable for both skater training and sports broadcasting.

REFERENCES

[1] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] H. Eom and R. Schutz, “Transition play in team performance of volleyball: a log-linear analysis,” *Res Q Exerc Sport*, vol. 63, no. 3, pp. 261–269, 1992.

[3] G. Liu *et al*., “A novel approach for tracking high speed skaters in sports using a panning camera,” *Pattern Recognition*, vol. 4, pp. 2922–2935, 2009.

[4] G. Liu *et al*., “Hierarchical model-based human motion tracking via unscented Kalman filter,” *IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[5] Y. Wang *et al*., “Observation and analysis of high-speed human motion with frequent occlusion in a large area,” *Measurement Science & Technology*, vol. 20, pp. 17, 2009.

[6] Y. Wang, “A novel and effective short track speed skating tracking system,” PhD dissertation, Utah State Univ., Logan, 2012.

[7] P. J. Figueroa *et al*., “Background recovering in outdoor image sequences: an example of soccer player segmentation,” *Image and Vision Computing*, vol. 24, no. 4, pp. 363–374, 2006.

[8] J. R. Renno *et al*., “Shadow classification and evaluation for soccer player detection,” *British Machine Vision Conference, Kingston University, London*, pp. 839–848, 2004.

[9] N. Vandenbroucke *et al*., “Color image segmentation by pixel classification in an

adapted hybrid color space. Application to soccer image analysis," *Computer Vision and Image Understanding*, vol. 90, no. 2, pp. 190–216, 2009.

[10] S. H. Khatoonabadi and M. Rahmati, "Automatic soccer players tracking in goal scenes by camera motion elimination," *Image and Vision Computing*, vol. 27, no. 4, pp. 469–479, 2009.

[11] M. Naemura *et al.*, "Morphological segmentation of sport scenes using color information," *IEEE Trans. Broadcast.*, vol. 46, no. 3, pp. 181–188, 2000.

[12] G. Duthie *et al.*, "The reliability of video-based time motion analysis," *J. Hum. Move Stud.*, vol. 44, pp. 259–272, 2003.

[13] D. Docherty *et al.*, "Time-motion analysis related to the physiological demands of rugby," *J. Hum. Move Stud.*, vol. 14, pp. 269–277, 1988.

[14] Y. Hong *et al.*, "Notational analysis on game strategy used by the world's top male squash players in international competition," *Aust. J. Sci. Med. Sport*, vol. 28, no. 1, pp. 18–23, 1996.

[15] F. Sanderson, "A notational system for analysing squash," *Phys. Educ. Rev.*, vol. 6, pp. 19–23, 1983.

[16] F. Sanderson and K. Way, "The development of an objective method of game analysis in squash rackets," *Br. J. Sports Med.*, vol. 11, pp. 188, 1977.

[17] P. Blomqvist *et al.*, "Validation of a notational analysis system in badminton," *J. Hum. Move Stud.*, vol. 35, pp. 137–150, 1998.

[18] L. Mendes and M. Janeira, "Basketball performance: multivariate study in Portuguese professional male basketball teams," *Cardiff: UWIC,* 2001.

[19] F. Tavares and N. Gomes, "The offensive process in basketball: a study in high

performance junior teams," *Int. J. Performance Analysis*, vol. 3, no. 1, pp. 34–39, 2003.

[20] M. Hughes and R. Daniel, "Playing patterns in elite and non-elite volleyball," *Int. J. Performance Analysis Sport*, vol. 30, no. 1, pp. 50–56, 2003.

[21] W. Choi and S. Savarese, "Multiple target tracking in world coordinate with single, minimally calibrated camera," *European Conference on Computer Vision*, pp. 553–567, 2010.

[22] W. Choi *et al.*, "A general framework for tracking multiple people from a moving camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–15, 2012.

[23] V. Reilly *et al.*, "Detection and tracking of large number of targets in wide area surveillance," *European Conference on Computer Vision*, pp. 186–199, 2010.

[24] S. Ali and M. Shah, "Cocoa - tracking in aerial imagery," *SPIE*, 2006.

[25] C. Liu *et al.*, "Monocular video-based markerless 3d human pose estimation by using local multiconnected belief propagation with multi-cue fusion," *IEEE International Conference on Intelligent Computing and Intelligent Systems*, pp. 478–482, 2010.

[26] D. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *Int. J. Comput. Vision*, vol. 1, no. 73, pp. 41–59, 2007.

[27] Y. Wang *et al.*, "Dynamic appearance model for particle filter based visual tracking," *Pattern Recognition*, vol. 45, pp. 4510–4523, 2012.

[28] J. Kang *et al.*, "Object reacquisition using geometric invariant appearance model," *International Conference on Pattern Recognition*, vol. 4, pp. 759–762, 2004.

[29] J. Kang *et al*., “On-line random forests,” *International Conference on Computer Vision Workshops*, pp. 1393–1400, 2009.

[30] T. Yang *et al*., “Real-time multiple objects tracking with occlusion handling in dynamic scenes,” *Computer Vision and Pattern Recognition*, vol. 1, pp. 970–975, 2005.

[31] B. Yang and R. Nevatia, “Online learned discriminative part based appearance models for multi-human tracking,” *European Conference on Computer Vision*, pp. 484–498, 2012.

[32] R. Szeliski, “Image alignment and stitching: a tutorial,” *Foundations and Trends in Computer Graphics and Computer Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[33] K. Nummiaro *et al*., “Color Features for Tracking Non-Rigid Objects,” *Chinese Journal of Automation*, vol. 29, no. 3, pp. 345–355, 2003.

[34] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[35] K. Bowyer *et al*., “Edge detector evaluation using empirical roc curve,” *Computer Vision and Image Understand*, vol. 10, pp. 77–103, 2001.

[36] G. L. Foresti and L. Snidaro, “A distributed sensor network for video surveillance of outdoor environments,” *International Conference on Image Processing*, Rochester, vol. 1, pp. 525–528, 2002.

[37] J. Wang *et al*., “Experiential sampling for video surveillance,” *ACM Workshop on Video Surveillance*, Berkeley, 2003.

[38] M. T. Yang *et al*., “A multimodal fusion system for people detection and tracking,” *Int. J. Imag. Syst. Tech.*, vol. 15, pp. 131–142, 2005.

[39] A. Yilmaz *et al*., “Contour-based object tracking with occlusion handling in video acquired using mobile cameras,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 1531–1536, 2004.

[40] A. Senior *et al*., “Appearance models for occlusion handling,” *Image and Vision Computing,* vol. 24, no. 11, pp.1233–1243, 2006.

[41] M. Hotter and R. Thoma, “Image segmentation based on object oriented mapping parameter estimation,” *Signal Processing*, vol. 15, no. 3, pp. 315–334, 1988.

[42] K. P. Lim *et al*., “Estimation of occlusion and dense motion fields in a bidirectional Bayesian framework,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 712–718, 2002.

[43] J. Berclaz *et al*., “Robust people tracking with global trajectory optimization,” *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 744–750, 2006.

[44] H. Jiang *et al*., “A linear programming approach for multiple object tracking,” *Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[45] T. Zhao and R. Nevatia, “Tracking multiple humans in complex situations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1208–1221, 2004.

[46] R. Hartley and A. Zisserman, “Multiple view geomerty in computer vision,” *Cambridge University Press (Second Edition)*, 2003.

[47] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[48] A. Wang *et al*., “An incremental extremely random forest classifier for online

learning and tracking," *International Conference on Image Processing (ICIP)*, pp. 1449–1452, 2009.

[49] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, pp. 849–865, 1988.

[50] N. Thacker *et al*., "The bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 32, pp. 1–7, 1997.

[51] Y. Wu *et al*., "Online object tracking: a benchmark," *Computer Vision and Pattern Recognition*, vol. 1, pp. 2411–2418, 2013.

[52] X. Jia *et al*., "Visual tracking via adaptive structural local sparse appearance model," *Computer Vision and Pattern Recognition*, pp. 1822–1829, 2012.

[53] J. F. Henriques *et al*., "Exploiting the circulant structure of tracking-by-detection with kernels," *European Conference on Computer Vision*, pp. 702–715, 2012.

[54] T. B. Dinh *et al*., "Context tracker: exploring supporters and distracters in unconstrained Environments," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1177–1184, 2011.

[55] K. Zhang *et al*., "Real-time compressive tracking," *European Conference on Computer Vision*, pp. 864–877, 2012.

[56] W. Zhong *et al*., "Robust object tracking via sparsity-based collaborative model," *Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012.

[57] S. Hare *et al*., "Struck: structured output tracking with kernels," *International Conference on Computer Vision*, pp. 263–270, 2011.

CURRICULUM VITAE

Chenguang Liu

Education

Ph.D in Computer Science *2014*

Utah State University, Logan, Utah, USA *GPA: 3.85*

Research interests include: Computer Vision, Image Processing, Machine Learning, Object tracking, Human Pose Tracking, etc.

Master in Computer Science *2008*

Harbin Institute of Technology, Harbin, China *Top 6%*

Bachelor in Computer Science *2003*

Harbin Institute of Technology, Harbin, China *Top 14%*

Achievements

- As a member of Wireless Power Transfer team, won the Energy Technology Innovation of the Year Award in 2013 at the annual Utah Governor's Energy Development Summit *2013*
- Awarded *Annual Reviewer Certificate* of Appreciations from Journal of Electric Imaging and Journal of Optical Engineering *2012*
- Won *Annual Achievement Award*, Energy Dynamics Lab, USU Research Foundation *2011*
- Filed *Three patents* with USU Research Foundation *2011*
- Won *Merit Student of Computer Science Department Award*, Harbin Institute of Technology, China *2008*
- As *monitor* of the class, won the *Top Ten Model Classes of Computer Science Department Award,* Harbin Institute of Technology, China *2002*
- Won the *First-class Scholarship* at computer science department, Harbin Institute of Technology, China *2002*

Work Experiences

Research Assistant, WAVE Inc., USU Commercial Enterprises *2012 - present*

- Worked directly with Chief Scientist of efficient wireless power transferring pads design and simulation using JMAG
- Designed the capacitor pre-charge circuitary
- Put forward the state machine of the bus control system, and participated the high level firmware system design
- Designed and implemented the firmware of power charging controller on vehicle
- Developed GUI with C# for firmware inspection, data logging and function tests

Research Assistant, Energy Dynamics Lab, USU Research Foundation *2010 - 2012*

- Worked directly with Chief Scientist of developing an Intelligent Occupancy System which aims at turning raw data from optical and/or infrared sensors into intelligent information and automatically controlling lights, thermostat, and air conditioner in order to save energy
- Developed a real-time face tracking and head gesture estimating algorithm, and further implemented a face orientation driven auto advertising system with C++ and OpenCV on a platform of embedded Windows7
- Developed a large scale gaze estimation algorithm which is able to produce 360 degree yaw face orientation with C++ and OpenCV
- Designed and implemented the software of occupant detection and lights control with C++ on Linux, which is able to turn on, turn off, dim or brighten different lights in the room according to the occupant's location
- Designed and implemented a robust people counting algorithm based on single webcam with C++ and OpenCV

Graduate Research Assistant, CVPRIP Lab, Computer Science Department, USU *2010 – 2014*

- Developed a fast road crack detection and classification algorithm using C++ and OpenCV
- Designed and implemented automatic road crack detection software with MFC, C++ and OpenCV, which
- Developed an efficient multi-cue fusion and dynamic appearance modeling based multiple skaters tracking algorithm with Matlab
- Developed several efficient algorithms for human pose tracking using C++ and OpenCV

Graduate Research Assistant, Computer Science Department, Harbin Institute of Technology, China *2006 - 2009*

- Participated in the development of automatic bill sorting machine (CF3000) for Harbin Billsorter Co., Ltd, and developed a firmware for gate control module and the CAN BUS driver for different chip boards in the system with C
- Managed software development (three students involved) of a vision system of patrolling robots for Shandong Luneng Int. ech. Co. ltd., conducted the system design and detailed design, and implemented main control module and scene matching module of the system with C++ and MFC

Software Engineer, Embedded Software R&D Center, Neusoft Group Ltd. *2003 - 2004*

- Participated in the R&D of two versions of Sony-Erricson multimedia cellphone
- Worked on high level design, low level design, implementation and unit test of multimedia modules including movie, camera and radio
- Took charge of the software release version control of the whole R&D team including software revision control, release documents review and submission, etc.
- Led the multimedia testing team (four employees involved), and participated in test case design, grading test and stress test

Leadership & Activities

Monitor, Harbin Institute of Technology, Harbin, China *2005 - 2008*

- Class monitor of 30+ graduate students
- Initiated money raising in the class for Sichuan Earthquake (100,000 casualties) in 2008
- Organized a Torch Relay activity of the class members for Beijing Olympics

Monitor, Harbin Institute of Technology, Harbin, China *2000 - 2003*

- Class monitor of 30+ undergraduate students
- Managed the class members to design and make a large poster for a China National Day Poster Design Competition in Computer Science Department, Harbin Institute of Technology, and won the second

Selected Publications

Conferences：

[1] *Chenguang Liu*, Heng-da Cheng, Yuxuan Wang. “A Novel Multi-cue Fusion Algorithm for Tracking Multiple Skaters in Short Track Speed Skating Competition.” IEEE Winter Conference on Applications of Computer Vision (WACV), Steamboat Springs CO. March 24-26, 2014 (Accepted)

[2] *Chenguang Liu*, Heng-da Cheng and Vicki Allan. “Articulated Human Pose Tracking based On Game Theory.” 20th IEEE international Conference on Image Processing (ICIP), Melbourne, Australia. Sep. 2013 (Oral)

[3] *Chenguang Liu*, Jiafeng Liu, Jianhua Huang, Xianglong Tang. “Monocular video-based markerless 3D human pose estimation by using Local Multi-connected Belief Propagation with multi-cue fusion.” IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), 478-482, Xiamen, China. Oct. 2010

[4] *Chenguang Liu*, Jiafeng Liu, Jianhua Huang and Xianglong Tang, Grid Particle Filter for Human Head Tracking Using 3D Model, Joint Conference on Information Sciences, Shenzhen, China, 2008

Journals：

[1] *Chenguang Liu*, Hengda Cheng, Aravind Dasu. “Scale Robust Head Pose Estimation Based on Relative Homography Transformation,” New Mathematics and Natural Computation, 10(1): 1-22, 2014

[2] *Chenguang Liu*, Peng Liu, Jiafeng Liu, Jianhua Huang, Xianglong Tang, "2D Articulated Pose Tracking Using Particle Filter with Partitioned Sampling and Model Constraints,” Journal of Intelligent and Robotic Systems, 58(2): 109-124, 2010

[3] *Chenguang Liu*, Hengda Cheng. “Multi-Object Tracking and Occlusion Reasoning Based on Adaptive Weighing Particle Filter.” Journal of Electronic Imaging. (Under Review)

[4] *Chenguang Liu*, Hengda Cheng. “Robust Multiple Cue Fusion Based High-Speed and Non-Rigid Object Tracking Algorithm for Short Track Speed Skating.” Pattern Recognition. (Under Review)

Computer Skills

- Proficient: C/C++, Matlab, MFC, OpenCV, MS Visual Studio, Freescale Code Warrior, LaTex, MS Office, Windows (OS)
- Good: C#, UML, JMAG, Linux (OS)
- Familiar: JAVA, XML, Microsoft Kinect for Windows SDK