Automatic Basketball Tracking in Broadcast Basketball Video

by

Shuang Yu

Bachelor of Science, The Chinese University of Hong Kong, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

August 2012

Abstract

We proposed and implemented an automatic basketball detection and tracking system for broadcast basketball video recorded with a single pan-tilt-zoom camera, using knowledge of player tracking information. The task is challenging because the basketball is blurred due to the camera and the ball's fast movements, and broadcast video compression; also the motion pattern of the basketball is complicated and the ball is hard to distinguish from the cluttered background region. We incorporated three independent detection approaches to detect the basketball and tracked the basketball using the Kalman Filter, and then we analyzed the tracklets and selected the passing / shooting tracklets and inferred the player possession information. We tested the system using 830 frames in broadcast basketball video, and our system demonstrated the ability to track some passing / shooting actions and then infer the player who controls the ball. The system is a first attempt to extend the intelligent basketball tracking system to include basketball tracking and player possession inference. Our proposed methodologies can be extended to other intelligent sports analysis systems, even when the ball movement in the sport is not constrained in two dimensional space.

Table of Contents

Abstract . . . . **ii**

Table of Contents . . . . **iii**

List of Tables . . . . **vi**

List of Figures . . . . **vii**

Glossary . . . . **x**

Acknowledgments . . . . **xi**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Challenges 2

1.4 Contributions 3

1.5 Thesis Outline 4

2 Related Work **5**

2.1 UBC Sports Video Analysis System 5

2.1.1 Homography Estimate 5

2.1.2 Sports Player Tracking and Identification 6

2.1.3 Determination of Puck Possession and Location in Hockey Video 7

2.2 Activity Recognition and Pose Estimate 7

2.2.1 Human Activity Recognition 8
2.2.2 Human Pose Estimate 8
2.3 Object Tracking 10
2.4 Ball Tracking 11
2.4.1 Soccer Ball Tracking 11
2.4.2 Sensor Based Tracking 12
2.4.3 3D Ball Tracking with Multiple Cameras 13
2.4.4 3D Basketball Trajectory Reconstruction with Domain Knowledge 14
2.5 Optical Flow 15

3 System Overview 16

4 Basketball Detection 18
4.1 Basketball Color Detection 18
4.2 Basketball Motion Detection 22
4.3 Basketball Differenced Shape Detection using Affine Flow 25
4.4 Combining Detections and Player Region Removal 28

5 Basketball Tracking and Tracks Analysis 30
5.1 Basketball Tracking 30
5.2 Track Analysis 32

6 Player Possession Inference 38

7 Experiments and Results 42
7.1 Ground Truth Data for Evaluation 42
7.2 Experiments and Error Measurements 43
7.2.1 Basketball Detection Experiments 43
7.2.2 Basketball Tracking and Player Inference Experiments . . 45

8 Conclusion and Future Work 48
8.1 Conclusion 48
8.2 Future Work 49
8.2.1 Probabilistic Detections 49

8.2.2 Tracking Methods 49
8.2.3 Splitting the Tracklets 50
8.2.4 Player Inference 50
8.2.5 Player Pose Estimation and Other Information for Inference 51

Bibliography 52

A Supporting Materials 57

List of Tables

Table 7.1 Basketball Detection Candidates Count 44
Table 7.2 Basketball Detected Frames 44
Table 7.3 Basketball Detected Pass / Shoot Frames 45
Table 7.4 Basketball Tracklets Count 45

List of Figures

Figure 2.1 Pose estimate on frontal view players 9
Figure 2.2 Pose estimate on non-frontal view players 10
Figure 2.3 The basketball is not always restricted in the court region (image sequence frame: 013057.jpg) 12
Figure 3.1 System overview 17
Figure 4.1 Ball color red channel frequency count 19
Figure 4.2 Ball color green channel frequency count 19
Figure 4.3 Ball color blue channel frequency count 20
Figure 4.4 Color pixel thresholding using red color in [24, 181], green color in [7, 132], and blue color in [1, 133] (image sequence frame: 013031.jpg) 20
Figure 4.5 Color pixel thresholding using red color in [90, 136], green color in [45, 80], and blue color in [32, 79] (image sequence frame: 013031.jpg) 21
Figure 4.6 Ball center pixels after noise reduction 22
Figure 4.7 Morphological dilation on ball center pixels (marked in red color) (image sequence frame: 013031.jpg) 22
Figure 4.8 When ball is moving fast and blurred in the image, color detection does not work well. (image sequence frame: 013346.jpg) 23
Figure 4.9 The color code for the normalized optical flow (u, v): x-axis is for u from -1 to 1 (left to right), y-axis is for v from -1 to 1 (bottom to top). 24

Figure 4.10 Visualized optical flow result between two frames (camera is almost static, image sequence frame: 013058.jpg) 24
Figure 4.11 Visualized optical flow result between two frames (with moving camera, image sequence frame: 013346.jpg) 25
Figure 4.12 Regions with large motion (basketball motion detection candidates on image sequence frame 013346.jpg) 25
Figure 4.13 Optical flow comparison 26
Figure 4.14 Edges match from the images under affine flow (image sequence frame: 013096.jpg) 26
Figure 4.15 Affine flow warped and differenced images 27
Figure 4.16 Shape detection candidates (image sequence frame: 013096.jpg) 27
Figure 4.17 All basketball candidates (marked with red rectangles) before the player region removal process (image sequence frame: 013061.jpg) 29
Figure 4.18 Basketball candidates (marked with red rectangles) after the player region removal process (image sequence frame: 013061.jpg) 29

Figure 5.1 Kalman filter tracking on the detection results (image sequence frame: 013097.jpg). The trajectories are formed by connecting all x, y coordinates positions of the detections with the same tracking id. 31
Figure 5.2 Kalman filter tracking on the detection results (image sequence frame: 013473.jpg) 31
Figure 5.3 First step track analysis results (image sequence frame: 013097.jpg) 34
Figure 5.4 First step track analysis results (image sequence frame: 013455.jpg) 34
Figure 5.5 Initial fitting on long track C8, kept it for further analysis . . . 36
Figure 5.6 Fitting on short track C12, rejected because of error 36
Figure 5.7 RANSAC fitting on C8, red stars are points draw from the detections on tracklet C8, blue circles are the fitted points using the RANSAC algorithm 37
Figure 5.8 Track analysis results with the three criteria (image sequence frame: 013100.jpg) 37

Figure 7.1 Ground truth bounding box and the trajectory (image sequence frame: 013103.jpg) 43
Figure 7.2 Basketball detection results 43
Figure 7.3 Ground truth trajectories and tracklets 46
Figure 7.4 False tracklets 46
Figure 7.5 Missed ground truth trajectories 47

Glossary

KLT Kanade-Lucas-Tomasi

DLT Direct Linear Transformation

2D Two-Dimensional

3D Three-Dimensional

HSV Hue-Saturation-Value

HOG Histograms of Oriented Gradients

Acknowledgments

First, I would like to show my gratitude to my supervisor, Prof. Jim Little, a responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. I would also like to thank my second reader, Prof. Bob Woodham, as well for reading the thesis, and kindly guiding me as my advisor in my first year of graduate study.

I shall extend my thanks to Wei-Lwun Lu, Kenji Okuma, Xin Duan, Maodi Hu and Shervin Mohammadi Tari for all their kindness and help in the lab.They have been kindly discussing the research problems and providing useful suggestions all the way through my thesis study. Kenji and Wei-Lwun provided the useful sports video analysis display system. I would also like to thank Kevin Schick for helping me annotate some of the basketball player and ball tracking ground truth data.

Last but not least, thank my parents for their emotional and financial support all these years. To them I dedicate this thesis.

Chapter 1

Introduction

1.1 Motivation

Knowing the locations of the players and the ball positions has been one important aspect in intelligent sports video analysis. Intelligent sports video analysis can automatically analyze sporting events and generate statistics about the games, in order to assist the coach to analyze the statics and design strategies accordingly. The system can also enhance the video images so that it can help the referee or audience to perceive the game more easily.

Previous work about ball tracking in broadcast video mainly focused on soccer ball tracking. There were also sensor-based or multiple camera systems to track the soccer ball, the puck in a hockey game, or the basketball in a basketball game: these systems require special hardware and are difficult to achieve in the reality. In broadcast video, the soccer ball or the puck is constrained on the field or hockey rink most of the time; however, the basketball bounces in the court, which makes the basketball less distinguishable from the background and harder to track.

Based on the Lu and Little's basketball player tracking work [Lu11], we want to design an automated system to track the basketball in broadcast video, so that we could gain more knowledge about the game. We could use the basketball passing / shooting tracklets to infer which player possesses the basketball for a time period, automatically generate the scores, and extract the passing / shooting patterns for automated analysis, etc.

1.2 Problem Statement

Our broadcast video used for analysis is broadcast basketball video recorded with a single pan-tilt-zoom camera. We developed a basketball tracking system to automatically find the basketball trajectories when it is being passed or shot, and infer which player possesses the basketball in the broadcast video.

In addition to the broadcast video data, our system also used the ground truth player position information as input. Lu and Little's system [Lu11] demonstrated the ability to track and identify players in broadcast videos. We chose other games and used ground truth player tracking information for demonstration purposes, but theoretically we could also use the player tracking information generated in Lu and Little's system.

The goal in our system is to detect and track the basketball when it is being passed or shot in the broadcast video, and then infer which player possesses the basketball when the basketball is in the player's hand or being dribbled by the player based on the player tracking information. We want to demonstrate that the current automated basketball intelligent video analysis system could be extended to include basketball tracking and player possession inference.

1.3 Challenges

Detecting and tracking the basketball robustly in broadcast sports video is a challenging task.

First of all, the broadcast video is recorded with a single pan-tilt-zoom camera, the camera moves from one side of the court to another frequently, in order to capture the major activities from the players; at the same time, the basketball is a fast moving object when it is being dribbled, passed or shot, which is easy to be blurred in the captured images.

In addition to the blurring caused by the camera and basketball movements, the broadcast video is compressed using MPEG4 format using selected key frames and the other frames in between are interpolated from the key frames. We expect that the basketball will be more seriously blurred in those interpolated frames. In our sequences, sometimes it is quite difficult for a human to see the basketball in some frames.

The blurred basketball is difficult to detect using edge detection methods. The size of the basketball is small with diameters ranging from 13 to 44 in the 1200 (width) * 780 (height) images. The basketball size is not fixed, because of the zooming camera and the basketball relative distance with the camera. The small basketball size and the blurring effects result in the lack of distinctive features on the basketball, which makes the basketball very hard to detect.

Previous work about soccer or puck tracking in broadcast video also has similar challenges stated above; color was used as a key clue to detect the soccer or puck. In a soccer or hockey game, the ball movements are almost constrained on the grass field or the hockey rink, thus, the ball is relatively distinctive from the background region: the soccer ball is white with mostly green color (the grass) in the background in the soccer game; the puck is black with mostly white color (the ice) in the background in the hockey game.

Sometimes the ball overlaps with the landmarks on logos on the field or rink, and such cases increase the difficulties in detections because of the noise in the ball's background region. In a basketball game, the basketball bounces on the court, most of the time the background color is not uniform, the basketball could be on the court, at a player's hand, or passing though the audience regions, etc. These conditions greatly increase the difficulties to detect and track the basketball.

In addition to these difficulties in detecting the basketball, there are frames where the basketball is partially or totally occluded by the game players or other objects (for example, the hoop or net). The occluded situation is hard to solve and it relies on other prior knowledge to infer the ball's position.

The motion pattern of the basketball is complicated: it can be held at a player's hand, dribbled, passed and shot in many different ways. The complicated motion pattern is a challenge in the basketball tracking, as well.

1.4 Contributions

Despite the challenges, our system is a first attempt to demonstrate the ability to detect and track the basketball using 830 frames in the 2011 National Basketball Association broadcast video.

The system incorporated three independent detection approaches to detect the basketball: the ball color was trained and used to detect the basketball, as used in soccer or puck tracking; in addition, optical flow was computed and the magnitude of the motion was used as a clue to indicate the basketball position, based on the prior knowledge that the basketball is moving fast when being passed / shot; affine flow was also used to compensate for the camera movement, and to find the moving basketball using the compensated information.

A Kalman Filter was deployed based on the detection results, and we selected the tracklets based on the observations of the passing / shooting patterns, and prior knowledge of the games. 6 out of the 8 passing / shooting tracklets were selected using our system, and 2 false passing / shooting tracklets were also kept in our system. We inferred the player possession information based on the tracklets selected by our system.

The system is a first attempt to extend the intelligent basketball tracking system to include the basketball tracking and player possession inference. Although in some difficult situations, especially in occluded case, the passing / shooting actions are not selected by our system, our system demonstrates the ability to track the remaining passing / shooting actions and then infer the player who controls the ball. The methodologies can be extended to other intelligent sports analysis systems, even when the ball movement in the sport is not constrained in two dimensional space.

1.5 Thesis Outline

The thesis is organized as follows. In Chapter 2, we review some related work in intelligent sport video analysis. Chapter 3 overviews the structure of our proposed basketball tracking system; Chapter 4, 5, and 6 describe the implementation details of the system. In Chapter 7, we report our experiment results. We conclude this thesis in Chapter 8 and also provide some suggestions for future extensions.

Chapter 2

Related Work

2.1 UBC Sports Video Analysis System

The Laboratory for Computational Intelligence at the University of British Columbia has worked on several sports video analysis projects, mainly using the hockey and basketball broadcast video. The goal is to understand the sports video semantics using computer vision algorithms.

2.1.1 Homography Estimate

In both hockey and basketball broadcast video, the cameras are not static. In order to analyze the scenes, it is necessary to estimate the camera parameters (pan, tilt, zoom) to map the video frame image onto a unified framework (the hockey rink or the basketball court) with a planar projective transformation. Each frame image and the unified framework in space are related by a homography, assuming a pinhole camera model.

Okuma et al. [OLL04] combined the Kanade-Lucas-Tomasi (KLT) tracking system [Bir99] [ST94] [TK91], RANSAC [FB81], and the normalized Direct Linear Transformation (DLT) algorithm [HZ03] to automatically compute the homographies. However, features are sometimes occluded by players or not seen due to camera movements, resulting in significant challenges for accurately estimating the homographies. Therefore, Gupta [GLW11] [Gup10] proposed a new method to use line and ellipse features along with the key-point based matches to estimate the homographies; the approach demonstrates the ability to track long sequences on the order of 1000 frames. In addition, Tari [Tar11] worked on automatically initializing the homography estimate, so that the initialization is accurate enough to guarantee the convergence to the optimal homography estimate.

With the homography estimate, it is possible to transform the player tracking and puck tracking results into a unified framework in hockey game analysis, with the assumption that the players and puck are roughly constrained on the Two-Dimensional (2D) rink plane. In the basketball video, the players can also be approximated on the 2D plane because they stand or run on the court most of the time; however, the basketball is dribbled and bounced up and down frequently in the Three-Dimensional (3D) world, and the homograhy estimate is for the 2D transformation on the court; therefore, it is not suitable to approximate the basketball trajectories in the unified framework with the 2D homography estimate.

2.1.2 Sports Player Tracking and Identification

In the hockey video, the players are detected, tracked, and their actions are recognized. In the basketball video, the players are detected, tracked, and identified.

Okuma, Little and Lowe et al. [OLL04] [OTF+04] initiated the sports video analysis system with multi-target hockey player tracking, using a multi-color observation model [PHVG02] based on Hue-Saturation-Value (HSV) color histograms and the boosted particle filter. In addition to tracking, Lu, Okuma, Little [LOL09] worked on recognizing actions of multiple hockey players using the Histograms of Oriented Gradients (HOG) to represent the players, an efficient off-line learning algorithm to learn the templates from training data, and an efficient online filtering algorithm to update the templates used by the tracker.

Also, Lu and Little [Lu11] tracked basketball players and recognized their identities. The new tracking system is able to track multiple players over hundreds of frames under severe occlusions.

The basketball player tracking and identification results can be used in our basketball tracking system, as useful domain knowledge for both basketball tracking and player possession inference.

2.1.3 Determination of Puck Possession and Location in Hockey Video

With the homography estimate and player tracking results as input, Duan and Woodham [Dua11] first attempted to extend the automated hockey video analysis system to include puck detection and puck possession.

The puck is small, moves rapidly and lacks distinctive local visual features; Duan used grey scale based blob detection to detect the puck, and used size and shape constraints to filter out some false positive detections.

For the tracking part, an innovative hierarchical graph-based method was used. In addition, he constructed a dense player motion field to estimate the puck location and possession information according to motion convergence points.

To the authors' best knowledge, there was no computer vision based previous work on puck tracking and possession analysis. The previous research about soccer tracking in broadcast video and other ball tracking are reviewed in Section 2.4.

2.2 Activity Recognition and Pose Estimate

As mentioned in Section 2.1.3, it is helpful to use the dense player motion field to infer the puck position. Similarly, we believe that, in basketball broadcast video, we could use player activities or player poses to infer the basketball position and status (being dribbled, passed, or shot into the basket, etc.). Therefore, state-of-the-art activity recognition and pose estimate work are briefly reviewed in this section.

Using the player motion field to infer the puck position does not work for the basketball case. There are mainly two reasons. First of all, the basketball bounces in 3D space, and the player can give force to the ball in any direction in the 2D image when he passes or shoots the ball, so the ball's moving direction has little correlation with the player's moving direction. Second, unlike hockey, only the player controlling the ball runs with the basketball, with one or two opponents around him, while others stay in some fixed region for the game strategy: it is very hard to find the motion pattern under such situation, and infer the basketball position.

2.2.1 Human Activity Recognition

Human activity recognition is an important area of computer vision research. It can be used in surveillance systems, patients monitoring systems, and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces.

Aggarwal and Ryoo [AR11] reviewed various state-of-the-art research papers on activity recognition.

For recognizing the simple actions of a single person, space-time volume approaches and sequential approaches are used. The space-time approaches view an input video as a 3-D (X,Y,T) volume, while sequential approaches interpret as a sequence of observations.

For high-level activities such as human-object interactions and group activities, different hierarchical recognition methodologies such as statistical approaches, syntactic approaches, and description-based approaches are compared. Both the statistical approaches and the syntactic approaches model a high-level activity as a string of atomic-level activities. The statistical approaches construct statistical state-based models concatenated hierarchically to represent and recognize high-level human activities, while the syntactic approaches use a grammar syntax such as a stochastic context-free grammar to model sequential activities. Description-based approaches represent human activities by describing sub-events of the activities and their temporal, spatial, and logical structures.

In basketball broadcast video, it is very hard to recognize the players' activities, because the players in the image do not have very high resolution (usually with width around 80 – 100 pixels, height around 120 – 140 pixels in our video), and they occlude each other severely due to the basketball game characteristics. There is no known previous work about basketball player activity recognition, but we believe it is a useful problem to tackle, and it will be a great information source for the basketball tracking and player possession inference.

2.2.2 Human Pose Estimate

In addition to considering the player activities in the basketball game, we also attempted to see whether it is possible to estimate the player pose in order to infer

Figure 2.1 has been removed due to copyright restrictions. It was a diagram of two basketball players viewed from the rear, and the pose estimate on them with marked head, torso, upper and lower arm, upper and lower leg parts. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 2.1: Pose estimate on frontal view players

the basketball position. The observation is that the basketball is usually either on the player's hand, or in one of the three motion statuses: being dribbled, passed, or shot into the basket. When the ball is in motion status, it is relatively better isolated from the background, and we can try to search in the image to find the ball; when it is in the player's hand, the ball is usually connected to the player's hand and hard to segment: given the pose, we can focus on the players' hand region in order to search for the ball.

The 2D articulated human pose estimate studies by Eichner et al. [EF09] [FMJZ08] [FMJZ09a] [FMjZ09b] have been tried on our basketball image. Their research estimates articulated human pose in still images. The algorithms can operate in uncontrolled images with difficult illumination conditions and cluttered backgrounds. People can appear at any location and scale in the image, and can wear any kind of clothing, in any color / texture. The only assumption in the algorithm is that people are upright (i.e. their head is above their torso) and they are seen either from the front or the rear. Figure 2.1 shows the frontal viewpoint player pose estimate results, and Figure 2.2 shows some trials on non-frontal viewpoint player pose estimates.

In the non-frontal pose estimate, all the four cases have some flaws: For case (a), one leg estimate occluded with the other leg, which is not the case in the original image; for case (b), one occluded arm could not be handled and the leg poses are wrongly estimated; for case (c), the basketball was considered as the head, and the arms are both wrongly estimated; for case (d), the occluded arms

Figure 2.2 has been removed due to copyright restrictions. It was a diagram of four basketball players viewed from different angles referred to as case (a), (b), (c), (d), and the pose estimate on them with marked head, torso, upper and lower arm, upper and lower leg parts. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 2.2: Pose estimate on non-frontal view players

between the two players are wrongly estimated. Because it is crucial to estimate the arms correctly in order to search the ball near the player's hand region, when the arm poses cannot be estimated accurately, it cannot provide a lot of help to search the basketball.

In the basketball game, many players are not in frontal view, so the pose estimate procedure still have a lot of room for improvements, and the pose estimate results are not yet used for the basketball tracking in our system.

2.3 Object Tracking

Object tracking, in general, is a challenging problem. Difficulties can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, nonrigid object structures, object-to-object and object-to-scene occlusions. Yilmaz, Javed, and Shah [YJS06] reviewed the state-of-the-art tracking methods.

The first issue for tracking is defining a suitable representation of the object; some common object shape and appearance representations are points, primitive geometric shapes and object contours and appearance representations. The next issue is the selection of image features used (such as color, motion, edges, etc.) as an input for the tracker. Almost all tracking algorithms require detection of the objects either in the first frame or in every frame.

According to Yilmaz et al. [YJS06], the object detection has 4 categories: point detectors, segmentation, background modeling, and supervised classifiers;

the tracking has 3 categories: pointing tracking (deterministic methods, statistical methods), kernel tracking (template and density based appearance models, multi-view appearance models), and silhouette tracking (contour evolution, matching shapes). They listed the representative work in the review paper.

2.4 Ball Tracking

In general tracking is a very broad topic. In this section, we narrow down the focus on reviewing the previous work of ball tracking. In Section 2.4.1 we review the soccer ball tracking research in broadcast video. In Sections 2.4.2 and 2.4.3 we review the sensor based tracking, and tracking with multiple cameras respectively. Finally, we review the 3D basketball trajectory reconstruction with domain knowledge, in Section 2.4.4.

2.4.1 Soccer Ball Tracking

Many have studied soccer ball detection and tracking. Yow et al. [YYYL95] used a template-based approach to detect and track the soccer ball. For the detections, Gong et al. [GSC$^+$95] utilized chromatic and morphological features to detect the ball; a circle detection algorithm based on the Circle Hough Transform was implemented to detect the soccer ball in Orazio et al.'s [DACN02] studies; Atherton and Kerbyson [AK99] proposed a size invariant circle detection method based on Hough transform. In addition, motion information was used in ball detection and tracking in Ohno, Miura, and Shirai's [YSM02] research.

In Seo et al.'s [SCKH97] studies, they first extracted the ground field to find the half line, the side line and the center circle and compute the image-to-model transformation, based on color histogram information under the assumption that the region of the ground is nearly green and occupies almost areas of images. For player tracking, template matching and Kalman filtering is applied. Occlusion reasoning is done by the color histogram back-projection method. For soccer ball tracking, since the ball is too small to track alone, they manually initialized the position and bounding box of the ball, and if a player is running near the ball, the player is marked "has ball". Finally, they map the trajectories in the image on the field with the image-to-model transformation.

Figure 2.3 has been removed due to copyright restrictions. It was an image frame showing a player is shooting the basketball. The ball overlaped with the audience region and was not restricted in the court region. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 2.3: The basketball is not always restricted in the court region (image sequence frame: 013057.jpg)

Tong, Lu, and Liu [TLL04] decided that directly detecting an object and evaluating whether it is a ball is not effective or robust. They proposed a coarse-to-fine ball detection and Condensation-based tracking method. The game field is first extracted and the posterior operations are restricted within it. Then, at the coarse step, some distinct non-ball regions are removed via evaluation of color and shape. And at the fine step, the remaining regions are further examined and the optimal one is determined as the ball. Afterwards, the Condensation algorithm is utilized to track the soccer.

In basketball, we cannot use the field color assumption as used in soccer game, because the basketball color is not very distinctive from the court color. We cannot restrict the ball in the court, either, because the ball moves in 3D space, it can be on the top of the audience region, as shown in Figure 2.3. These increased the difficulties for basketball detection and tracking.

2.4.2 Sensor Based Tracking

To track the puck in a hockey game, in addition to the vision-based approach mentioned in Section 2.1.3, the Fox TV network introduced their sensor based FoxTrax system [Cav97]. Similarly, in a soccer game, the GoalRef system [Hol12] can be used as a sensor-based approach to decide whether the soccer ball has crossed the goal-line.

The FoxTrax puck is a standard National Hockey League puck with a tiny circuit board and a battery placed inside. The circuit board contains a shock sensor and infrared emitters. During the broadcast, infrared pulses emitted by the puck are detected by 20 pulse detectors and 10 modified IR cameras located in the rink rafters. The system can track the puck robustly in real time, but setting up the system has high cost.

Developed at the Fraunhofer Institute, GoalRef requires the football to be chipped and a magnetic field set up in the goal mouth. Sensors detect changes in the field when the ball crosses the line, notifying the referee nearly instantaneously.

2.4.3 3D Ball Tracking with Multiple Cameras

In addition to the sensor-based approach, the Hawk-Eye system [Hol12] uses a minimum of four cameras to track the trajectory of a moving ball (in soccer, cricket, and tennis). Pingali, Opalach and Jean [POJ00] tracked the tennis ball in 3D using multiple cameras in real time. Ruiz and Berclaz [Rui10] tracked a basketball during a basketball match recorded with a multi-camera system.

In the soccer game, the Hawk-Eye system will install six cameras in each goal, each of which will monitor the location of the ball. By triangulating the ball's position from the images, the system can notify the referee if the ball definitively crosses the line. Crucially, it can do this within a second (a FIFA stipulation) by transmitting a radio signal picked up by the referee's wristwatch. Though used by cricket broadcasters since 2001, Hawk-Eye has only been used to help umpires adjudicate leg before wicket decisions since 2008, due to disputes about its accuracy. Hawk-Eye technology also has been used to assist with line decisions in tennis since 2006.

Pingali et al. [POJ00] used six cameras around a stadium, divided into four pairs, to track the tennis ball on serves which sometimes exceed speeds of 225 kmph. A multi-threaded approach was taken. Each thread tracked the ball in a pair of cameras based on motion, intensity and shape, performed stereo matching to obtain the 3D trajectory, detected when a ball goes out of view of its camera pair, and initialized and triggered a subsequent thread.

Ruiz and Berclaz [Rui10] first developed methods to detect a basketball in images based on its appearance, then attempted to track the ball in three dimensions using the cameras calibration. Their algorithm can track the ball and find its 3D position during several consecutive frames.

2.4.4 3D Basketball Trajectory Reconstruction with Domain Knowledge

Both sensor-based tracking and the tracking with multiple cameras require special hardware set up in the games, which could be difficult to achieve without significant expense. Chen et al. [CTC+09] proposed physics-based ball tracking and 3D trajectory reconstruction method in basketball video, in order to estimate the shooting location. Their research mainly focused on the shooting trajectory reconstruction.

2D-to-3D inference is intrinsically a challenging problem due to the loss of 3D information in projection to 2D frames. Their system incorporated domain knowledge and physical characteristics of ball motion into object tracking to overcome the problem of 2D-to-3D inference.

They used the domain knowledge that the court lines and important markers are in white color in their official game rules, to fit the court model; in addition, they needed two more points which are not on the court plane to calculate the 2D-to-3D calibration parameters. The two endpoints of the backboard top-border are selected because the light condition makes them easy to detect in frames.

They detected the basketball based on moving pixels and color. They assumed the camera motion is not so violent in the shooting trajectories, so that they could use motion to detect the ball. They also refined the color detection using shape and size sieve. Then they tracked the trajectories by fitting the detections on parabolic curves, and identified a shooting trajectory by examining if it approaches the backboard. With the 2D trajectories extracted and the camera parameters calibrated, they employed the physical characteristics of ball motion in real world for 3D trajectory reconstruction.

Their proposed system can greatly assist intelligence collection and statistics analysis in basketball game, however, the 2D-to-3D calibration require the backboard top border to be captured in the camera all the time. In our system, we are interested in not only the shooting trajectories but also the passing trajectories and other interactions with the ball in the basketball broadcast video.

2.5 Optical Flow

Optical flow describes the distribution of the apparent movement of the brightness pattern in an image. It is an important tool in image sequence analysis.

Different techniques are used in order to compute the optical flow, according to Mesbah [Mes99], the techniques can be classified into three categories: gradient-based, matching-based, and frequency-based.

Chambolle and Pock developed a first-order primal-dual algorithm for convex problems with applications to imaging [CP11]. They provided a Matlab package for computing dense duality-based TV-L1 optical flow.

We used the Matlab package by Chambolle and Pock when we tried to detect the basketball in the image sequences.

Chapter 3

System Overview

Figure 3.1 overviews our ball tracking and player possession analysis system. The system mainly has three modules: the ball detection module, the ball tracking module, and the player possession inference module.

In the ball detection module, we try to detect the basketball. We use three independent ball detection approaches: ball color detection, ball motion detection, and shape detection using affine flow.

We hand label several basketball regions and learn the basketball color, then search the color range in the images, and filter the regions with size constraints using morphological operation. However, we notice that when the ball is moving fast in the image, it is blurred and hard to detect using color. To compensate, we use the prior knowledge that the basketball and players are the most fast moving object in the image sequences. We compute the optical flow between each pair of consecutive frames, and use Laplacian of Gaussian to find the regions with large motion, which could be the fast moving basketball candidates. In addition, we take the difference between two consecutive frames with compensated camera parameters and find the ball-shape like regions are ball candidates as well.

The three detection candidates are all combined using logical OR operations, so that we could keep the true ball detection in our results, even with quite a few false positive detections. We could filter out the false positive detections in the track analysis part in our ball tracking module. The details are discussed in Chapter 4.

With the combined detections, in the ball tracking module, we first track the

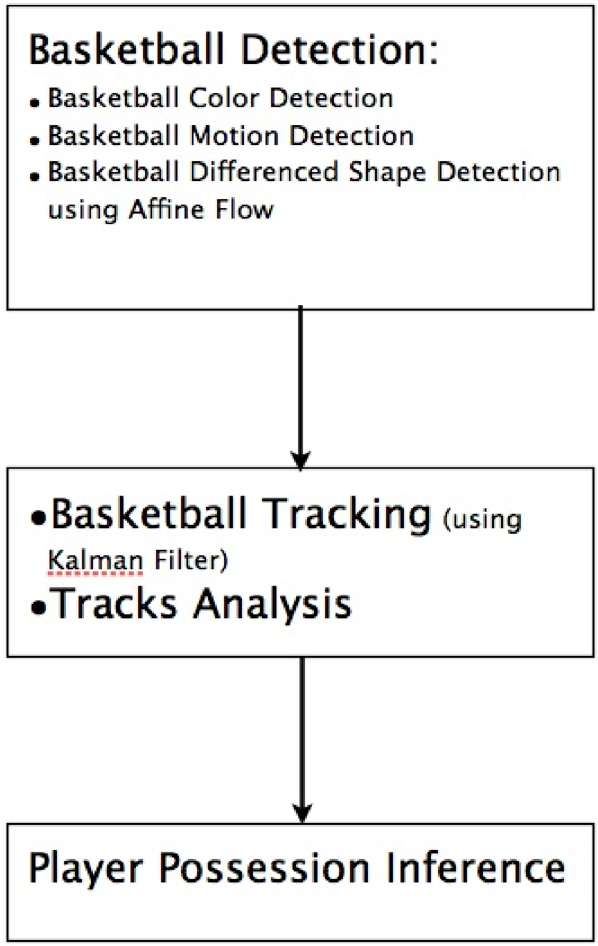

Figure 3.1: System overview

basketball using the Kalman Filter. Then we analyze the tracks we got from the the Kalman Filter tracking code, using the player tracking results and some basketball domain knowledge. Chapter 5 shows the implementation details.

Finally, in Chapter 6, we use the selected tracks obtained in the tracking module to infer which player is controlling the basketball in the image sequences.

Chapter 4

Basketball Detection

This chapter describes how we detect the basketball candidates. In each video frame, there should be zero (occluded case) or one true basketball candidate. In Section 4.1, we use color to find the basketball candidates, in Section 4.2; we use optical flow code and the motion information to identify the ball candidates; and in Section 4.3, we use affine flow to estimate the camera parameters and try to find the ball candidates. The detection results are combined using a logical OR operation.

4.1 Basketball Color Detection

In previous vision-based soccer [SCKH97], puck [Dua11] and basketball [CTC$^+$09] tracking , color was a very important clue. Similarly, we first tried to see how color works on basketball detection.

We selected 38 images and hand labelled the ball center positions (X, Y coordinates), in order to analyze and learn the ball color. The pixels around the ball centers within radius 6 were considered as pixels on the ball, and were taken for the color analysis.

For the ball color pixels, we got the Red Green Blue (RGB) frequency count in Figure 4.1, 4.2, and 4.3. For the red channel the ball color lies in the range [24, 181]; for the green channel the ball color lies in the range [7, 132]; for the blue channel the ball color lies in the range [1, 133]. In all three channels, the model for the color distribution is a Gaussian centered on the mean of the ranges. We tested

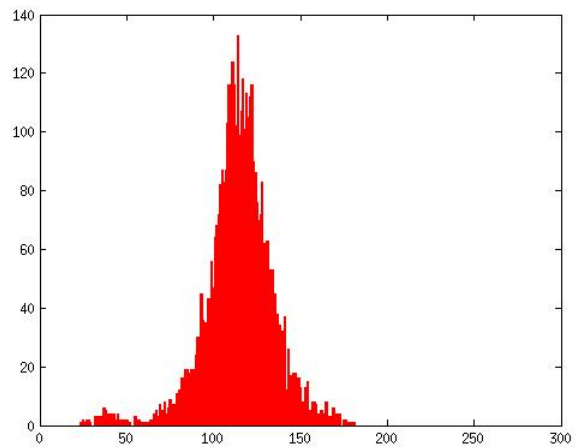

Figure 4.1: Ball color red channel frequency count

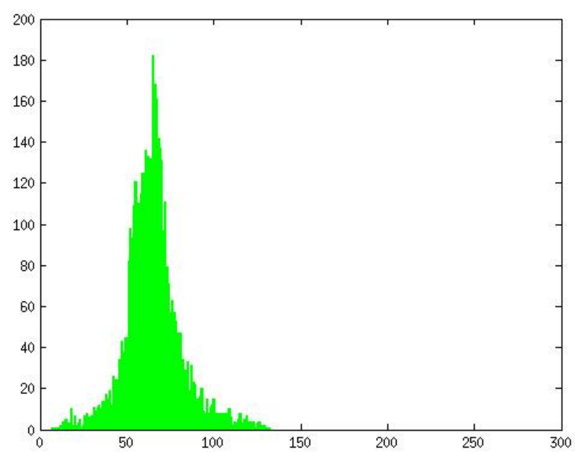

Figure 4.2: Ball color green channel frequency count

using the HSV representation of color and the results were similar.

With the color range we got from the training set, we tested how to find those pixels that look like the ball color. If a pixel is within the red, green, and blue channel range we got from the samples, we keep that pixel in the output image; otherwise, that pixel color is set to [255,255,255] (white color) in the RGB channel

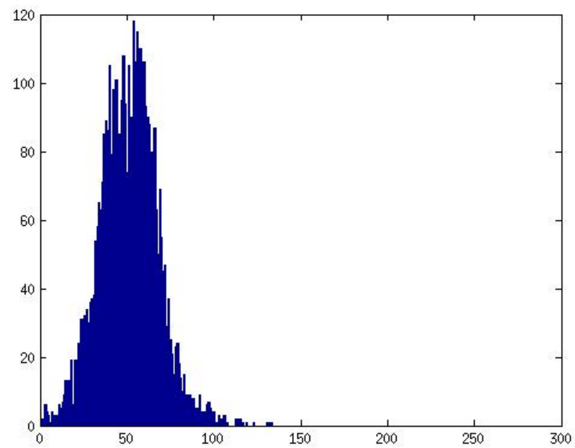

Figure 4.3: Ball color blue channel frequency count

Figure 4.4 has been removed due to copyright restrictions. It was a diagram showing the pixels after applying the loose color range thresholding step. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.4: Color pixel thresholding using red color in [24, 181], green color in [7, 132], and blue color in [1, 133] (image sequence frame: 013031.jpg)

in the output image. One example output image is shown in Figure 4.4.

We observed that too many pixels are considered as ball color pixels from Figure 4.4, including but not limited to the pixels on player skins (heads, arms, and legs), logos on the court, and player clothes.

In the Gaussian distribution, pixels drawn from the tail may be noisy (in Figure 4.4). We manually selected stricter ranges for the RGB channels, eliminating the tail parts of the Gaussian distribution: for the red channel, we changed the color range from [24, 181] to [90, 136]; for the green channel, we changed the color

Figure 4.5 has been removed due to copyright restrictions. It was a diagram showing the pixels after applying the strict color range thresholding step. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.5: Color pixel thresholding using red color in [90, 136], green color in [45, 80], and blue color in [32, 79] (image sequence frame: 013031.jpg)

range from [7, 132] to [45, 80]; and for the blue channel, we changed the color range from [1, 133] to [32, 79]. We ran the same thresholding process as described above, and the output image is shown in Figure 4.5.

The stricter color ranges with removed Gaussian tail parts worked much better than the results shown in Figure 4.4. However, the basketball color is not so distinctive as the soccer or hockey puck. Some logo color on the court, player skin and audience clothes color are still very similar to the basketball color.

We considered each ball-color-like pixel as the potential ball center pixel and verified it with the following constraints: 1. In the nearby 21*21 pixel region (centered around current pixel), there were at least 60 ball-color-like pixels; 2. We took the difference of the pixels centered around the current pixel within radius 10 with the average ball we got in the training part; if the sum of the differences in the RGB channels was smaller than 20000 then we consider the current pixel as a ball center pixel. All of the constants were determined empirically.

If a pixel does not satisfy the constraints, we consider it as a noise pixel. After the noise reduction process, the potential ball center pixels are shown in Figure 4.6.

Because the basketball size is not fixed in the image (it depends on the ball's distance with the camera), we used the ball size with radius 6 in training and ball size with radius 10 in color detection to ensure the basketball is detected in most of the cases. Due to the fixed ball size in detection, many potential ball center pixels are clustered together; they should be actually one ball center.

We tried to find each cluster as one basketball detection candidate. We achieve

Figure 4.6 has been removed due to copyright restrictions. It was a diagram showing the ball center pixels after applying the noise reduction step. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.6: Ball center pixels after noise reduction

Figure 4.7 has been removed due to copyright restrictions. It was a diagram showing the dilated ball pixels based on the results in Figure 4.6. It contained 11 connected regions (possible basketball candidates) in this frame. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.7: Morphological dilation on ball center pixels (marked in red color) (image sequence frame: 013031.jpg)

this by first applying the morphological dilation operation on the ball center pixels with a disk of radius 10 (example shown in Figure 4.7), and then considering each separate connected region as one basketball detection candidate.

4.2 Basketball Motion Detection

When the basketball is not moving very fast, the detection based on color can often detect the true basketball although there are some false detections due to the player skin etc. However, when the basketball is moving fast, it can be blurred significantly in the video image, and detection based on color cannot work well. Figure 4.8 shows one of such cases.

In a basketball game, the basketball is often the most fast moving object, and it is blurred in many frames. Our broadcast video is compressed in MPEG format, with some key frames directly represented and other frames in between are

Figure 4.8 has been removed due to copyright restrictions. It was a frame with the color detection step, the basketball-color-like connected region was only detected on one player's arm region, but not the fast moving and blurred basketball region. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.8: When ball is moving fast and blurred in the image, color detection does not work well. (image sequence frame: 013346.jpg)

interpolated. This also causes further difficulties in finding the basketball. To compensate for these, we try to find out the fast moving regions as possible basketball candidates as well.

We first run the optical flow code mentioned in Section 2.5, with the color code shown in Figure 4.9; some color-coded visualized flow results are shown in Figure 4.10 with almost no camera movement, and Figure 4.11 with moving camera. Most of the time, the camera is not static between two frames.

The true basketball region (annotated with red rectangle bounding box) is color-coded with more saturated color, implying that the optical flow captured the fast brightness change in this region. In addition, as the players are moving fast, they are color-coded with more saturated color as well.

In order to find those regions with fast movement, we first compute the magnitude value of the optical flow (u, v) at each pixel. Then we created a Laplacian of Gaussian filter, with size [25, 25] and sigma = 6, and applied the filter on the magnitude image to identify regions with large motion. We took those pixels with response value greater than 10 as the fast moving pixels. The response map masked on the original image for Figure 4.8 is shown in Figure 4.12.

In addition to Chambolle and Pock's TV-L1 optical flow code, Brox's high accuracy optic flow using a theory for warping [BBPW04] was also compared, Figure 4.13 shows the color-coded optical flow and the motion detection results on image sequence 013060.jpg using the two approaches. TV-L1 optical flow code

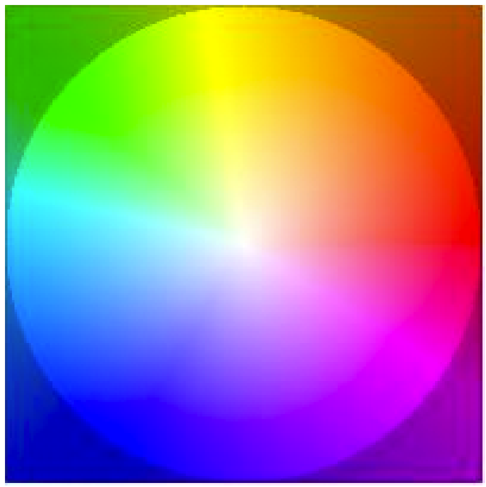

Figure 4.9: The color code for the normalized optical flow (u, v): x-axis is for u from -1 to 1 (left to right), y-axis is for v from -1 to 1 (bottom to top).

Figure 4.10 has been removed due to copyright restrictions. It was a color-coded optical flow image between two consecutive images with almost static camera. There was strong motion evidence in the basketball region for this image. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.10: Visualized optical flow result between two frames (camera is almost static, image sequence frame: 013058.jpg)

Figure 4.11 has been removed due to copyright restrictions. It was a color-coded optical flow image between two consecutive images with moving camera. There was strong motion evidence in the basketball region for this image. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.11: Visualized optical flow result between two frames (with moving camera, image sequence frame: 013346.jpg)

Figure 4.12 has been removed due to copyright restrictions. It was a diagram showing 5 connected regions with large motion as possible basketball candidates. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.12: Regions with large motion (basketball motion detection candidates on image sequence frame 013346.jpg)

works more stably in our image sequences most of the time.

Again, we used each connected region as one basketball candidate if the connected region is within the ball size range.

4.3 Basketball Differenced Shape Detection using Affine Flow

In addition to the color and motion based approaches, David Young [You10] provided efficient Matlab code to compute the affine optic flow between two images, which has 6 parameters describing image translation, dilation, rotation and shear. Figure 4.14 shows the edges matched from the images under affine flow; the green color edges are the first image edges, the blue color edges are the second image

Figure 4.13 has been removed due to copyright restrictions. It was a diagram showing the TV-L1 and Brox optical flow results on two consecutive frames, and the motion response on the 2 optical flow results. The motion response using TV-L1 optical flow result shows better evidence of the basketball region than that using Brox optical flow. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.13: Optical flow comparison

Figure 4.14 has been removed due to copyright restrictions. It was a diagram showing the affine flow mapping between two consecutive frames. It displayed the edges in the first and the second image, as well as the edge warped from the first image using the affine flow. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.14: Edges match from the images under affine flow (image sequence frame: 013096.jpg)

edges, and the red color edges are the edges from the first image after warping by the flow field.

The affine flow idea is somehow similar to the motion detection part, with the assumption that the basketball is moving very fast. The affine flow shape detection approach and motion detection complement each other: sometimes the true basketball ball candidate can only be found by the motion detection part, while sometimes the true basketball can only be found by the affine flow shape detection.

The affine flow provides an approximation to the true flow, though it does not capture its details or exact form. Therefore, we can use the affine flow to compensate for the camera movement between the current frame and its previous frame,

Figure 4.15 has been removed due to copyright restrictions. It was a diagram of one frame and its affine warped images in red, green, blue channels. It also displayed the differenced images between the original and the warped images in the three color channel. In the differenced images there is evidence of the basketball region. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.15: Affine flow warped and differenced images

Figure 4.16 has been removed due to copyright restrictions. It was a frame that displayed the basketball candidate using the affine flow and shape detection. In this example, exactly one basketball candidate region was detected using this approach. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.16: Shape detection candidates (image sequence frame: 013096.jpg)

by warping the previous frame image onto the current frame, and then taking the difference of the current frame and the warped image, as shown in Figure 4.15. The basketball is a small region (the details), so when the basketball is moving fast between the two frames, the region difference should be large.

We computed the difference in the red, green, blue channels separately using the affine flow to compensate for the camera motion and thresholded the large difference region as possible basketball pixels. Then we combined the three channel thresholding results using a logical OR operation. One combined thresholding result masked on the original image is shown in Figure 4.16.

Similarly, we used each connected region as one basketball candidate if the connected region is within the ball size range.

4.4 Combining Detections and Player Region Removal

There is at most one true basketball candidate in each frame; however, we noticed that the three independent detection methods complemented each other: when the basketball is not moving fast and is clear in the image, color detection works well most of the time; when the basketball is moving fast, motion or shape detection can sometimes detect the true basketball candidate. In order to keep the true basketball candidate, we combine the three detection threshold map results using the logical OR operation.

By combining the detections with the logical OR operation, we inevitably introduced many false basketball candidates in our detection result. We noticed many false basketball candidates are on the players, because the skin colors are very similar to the basketball color, and the player motion is large too.

Because there are too many false basketball detections inside the player region, and our tracking should focus on the passing and shooting actions where the basketball is far away from the players most of the time in those actions, we could use the ground truth player tracking results (mentioned in Section 2.1.2) to remove those basketball candidates inside the player regions. When a basketball candidate is totally within one player region, we remove that basketball candidate.

The basketball detection candidates before and after the player region removal process are shown in Figure 4.17 and 4.18 separately.

Figure 4.17 has been removed due to copyright restrictions. It was a frame showing all basketball candidates using the three detection approaches, as well as the marked ground truth basketball player regions. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.17: All basketball candidates (marked with red rectangles) before the player region removal process (image sequence frame: 013061.jpg)

Figure 4.18 has been removed due to copyright restrictions. It was a frame showing all basketball candidates after the player region removal process. It also showed the marked ground truth basketball player regions. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 4.18: Basketball candidates (marked with red rectangles) after the player region removal process (image sequence frame: 013061.jpg)

Chapter 5

Basketball Tracking and Tracks Analysis

With the basketball candidates we got in the detection phase in Chapter 4, we can try to link the candidates and get the basketball tracks. As we mentioned, the ideal case should be that there is only at most one true basketball detection in each frame, and there is only one track for some time period when the basketball is being passed or shot into the basket; however, we detected more than one basketball candidate most of the time in order to keep the true basketball in our detection results, and thus computed multiple tracks for a time period. Therefore, in addition to the basketball tracking, we also analyze all the tracks we derived, to eliminate some false basketball tracks.

5.1 Basketball Tracking

The Kalman filter is an algorithm to use a series of measurements with noise and inaccuracies observed over time to produce estimates of unknown variables. It can be used as a classical tracking algorithm. The Kalman filter we used is from Lu's thesis work [Lu11].

In our basketball case, the measurements are the basketball candidates over time. The Kalman filter implements a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated covariance. Two Kalman filter

Figure 5.1 has been removed due to copyright restrictions. It was a diagram showing connected tracklets based on the detection results after one player's shooting action. The shooting action was tracked, and there are also other 8 false tracklets due to the noisy detections. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 5.1: Kalman filter tracking on the detection results (image sequence frame: 013097.jpg). The trajectories are formed by connecting all x, y coordinates positions of the detections with the same tracking id.

Figure 5.2 has been removed due to copyright restrictions. It was a diagram showing connected tracklets based on the detection results after one player's passing action. The passing action was tracked, and there are also other 7 false tracklets due to the noisy detections. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 5.2: Kalman filter tracking on the detection results (image sequence frame: 013473.jpg)

tracking results are shown in Figure 5.1 and Figure 5.2, with a unique tracking ID associated with each track.

In Figure 5.1 the track with ID C8 is actually the true basketball track, which is a basketball shooting action. All other tracks are noisy tracks because of detections with too many false positive basketball candidates. Similarly, in Figure 5.2, the track with ID C130 is the true basketball track, which is a basketball passing action from one player to another player; other tracks are noisy tracks, and we want to get rid of those false tracks intelligently.

5.2 Track Analysis

With the many tracks we got from the previous section, we analyzed each of them and tried to eliminate the false tracks, using some prior knowledge about the basketball game.

We first selected the passing / shooting actions using Algorithm 1. The input were the tracklets in Section 5.1, the output were some new tracklets, which are the continuous segments of points in the input tracklets (called snippet); some tracklets in the input are removed.

We associated each track with the players, using the ground truth player tracking information. We first loaded the track starting position in the image, and loaded the ground truth player tracking results at the starting time: we did a linear search on the absolute position of the track starting position to see whether the basketball candidate is completely inside any of the player bounding boxes, if yes, we marked the starting player ID with the track, otherwise, we used 'NO' to represent that the starting position is not associated with any player. Similarly, we loaded the track positions in each of the following frames and associated the positions with either the player ID or 'NO' to represent the position is not associated with any players.

The passing and shooting action are actions that are not associated with any players in between, so we are only interested in the track snippet that is not associated with the players for some frames. We selected tracks that have 'NO' associated with them for at least k ($k = 8$) frames. In addition, we allowed for at most d ($d = 2$) frames noise (i.e., track associated with player ID in those frames) in the selection process, because occlusions with other players could occur in the passing or shooting actions – the occlusions should be no more than d frames most of the time though, due to the fast movement in the passing or shooting actions. If a track does not contain a snippet with 'NO' with at least k frames, then it is probably not a true basketball track being passed or shot: we discarded such tracks. The parameters k and d are determined empirically.

In the passing / shooting action selection step, some tracklets are deleted directly, while part of other tracklets are selected for further analysis. Figures 5.3 and 5.4 shows the results after the passing / shooting selection.

Many tracklets in Figure 5.1 and Figure 5.2 are discarded because they do not

Algorithm 1 Passing / Shooting Track Selection

```
for each track do
    for each frame in the current track do
        associate track position in current frame with player ID or 'NO'
    end for
end for
k = 8
d = 2
for each track do
    count = 0
    output_track = []
    noise_count = 0
    for each frame in the current track do
        if track position association is 'NO' & noise_count ≤ d then
            count = count + 1
            output_track.append(current frame)
            noise_count = 0
        else
            if count < k & track position association is 'NO' & noise_count > d
then
                count = 1
                output_track = [current frame]
                noise_count = 0
            else
                noise_count = noise_count + 1
            end if
        end if
    end for
    if count ≥ k then
        update track with output_track
    else
        delete current track
    end if
end for
```

Figure 5.3 has been removed due to copyright restrictions. It was a diagram showing connected tracklets after the first step track analysis. The shooting action was tracked, and there is 1 remaining false tracklet left. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 5.3: First step track analysis results (image sequence frame: 013097.jpg)

Figure 5.4 has been removed due to copyright restrictions. It was a diagram showing connected tracklets after the first step track analysis. The passing action was tracked, and there is no other false tracklet. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 5.4: First step track analysis results (image sequence frame: 013455.jpg)

satisfy the passing / shooting selection criteria. Part of track C8 and C12 snippets were selected as new tracklets from Figure 5.1 to Figure 5.3; similarly, part of track C130 snippet was selected as new tracklet from Figure 5.2 to Figure 5.4.

In the remaining passing / shooting tracklet candidates, we further selected the tracks using mainly three criteria: 1. The true basketball track is always from one player (A) to another player (B). 2. For the true basketball track, player A and B are from the same team most of the time, unless in the occasional case the passing action failed because the passing is intercepted by the player in the other team. 3. The track trajectories in between are either passing or shooting trajectories, and the curve looks like a parabola.

We can assume the passing or shooting trajectory curves look like a parabola, because usually in those actions, the camera does not move very fast. In addition, the time duration of a shot is about 1 – 3 seconds (20 – 90 frames) which is very fast, so the camera movement does not change the parabola curves a lot.

We took the remaining passing / shooting tracklet candidates and analyzed them using the three criteria mentioned above. First, we took the track snippet start and end positions, and saw which players the track snippet started and ended from by searching the nearby players. If the track snippet started and ended from players in different team in a short track, we discarded the track snippet.

For the remaining tracks, we first fit the whole track on the parabola curve, if it is a short track and has large error on the parabola fitting, it is discarded; if it is a short track and has small error, it could be a valid passing action; if it is a long track, we keep it for further analysis. Figure 5.5 shows the initial fitting on the shooting action C8, it is a long track and we keep it for further analysis; Figure 5.6 shows a fitting on invalid short track C12, the error value on the fitted curve is large, therefore the tracklet is discarded. The lines on the Figures are the trajectories and the circles are the parabola curve estimated points.

For the long tracks like C8, we used the RANSAC algorithm [FB81] to fit the trajectories with a parabola. The RANSAC process picked 3 points randomly and estimate the parabola parameters, and tested how all the points fitted the parabola. This process was repeated 100 times, so that there was a high chance to get a good fit if the trajectory was like parabola. If the trajectory fitted well on the parabola, we considered the whole tracklet as a true basketball track (as shown in Figure 5.7 as an example); otherwise, we discarded the track as a noisy track.

We keep the whole tracklet as the true basketball track, so that we can use it to connect to the rest of the play. However, the part of the tracklets using RANSAC algorithm could be used to further analyze the trajectory between the shooting player and the basketball, and the falling trajectory after the basketball hits the backboard. This could be used in the further work.

After all the tracks analysis steps, one result is shown in Figure 5.8, where the identifier C8-0.64-0.12-M1-D3 means that the track C8 starts from player M1, ends at player D3 as a long track, with the initial fitting confidence 0.12, and the RANSAC refitting confidence value 0.64. The false tracklet C12 was discarded from Figure 5.3 to Figure 5.8 when we selected the tracklets using the three criteria

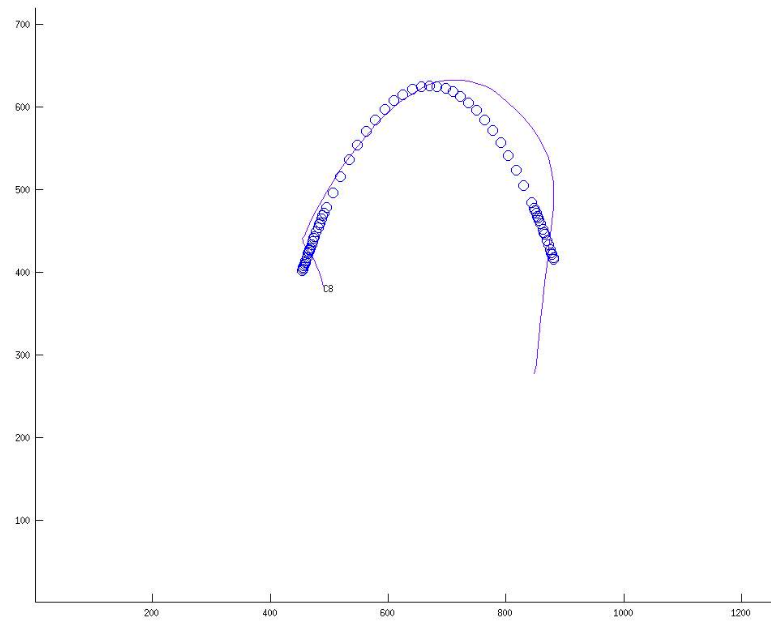

Figure 5.5: Initial fitting on long track C8, kept it for further analysis

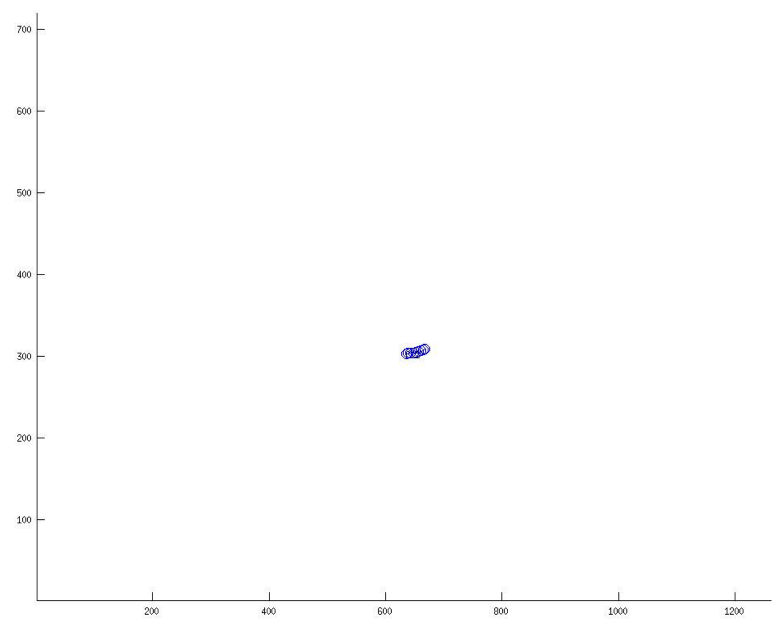

Figure 5.6: Fitting on short track C12, rejected because of error

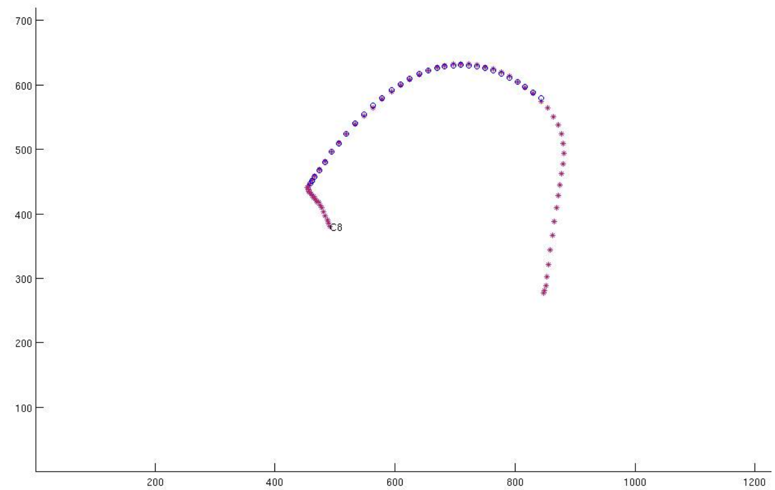

Figure 5.7: RANSAC fitting on C8, red stars are points draw from the detections on tracklet C8, blue circles are the fitted points using the RANSAC algorithm

Figure 5.8 has been removed due to copyright restrictions. It displayed the tracklets after the analysis with the three criteria. The tracklet displayed had track ID C8-0.64-0.12-M1-D3 and it was a shooting tracklet. In addition, the ground truth player regions were also shown on the image. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 5.8: Track analysis results with the three criteria (image sequence frame: 013100.jpg)

mentioned above.

Chapter 6

Player Possession Inference

If we have all the true basketball tracks, which are either shooting or passing trajectories, then it is easy to infer which player is controlling the ball by using the track starting and ending information from Chapter 5. However, there are still some false positive tracks in our tracks analysis results (though we already eliminated many), especially when the camera is moving fast; also, there are some missing true ball tracks, due to occlusion or the low basketball detection rate in some frames. We tried to infer the player possession information from the tracks we obtained in Chapter 5.

We first took the data file we obtained from Chapter 5, and processed them into formats that were easy to analyze for player possession inference.

The algorithm for the preprocessing is shown in Algorithm 2 and 3. The new track file (prep1) item format after the preprocessing in Algorithm 2 is:

track_id track_start_frame number_of_following_frames

The new track file (prep2) item format after the preprocessing in Algorithm 3 is:

track_id track_start_frame number_of_all_following_frames boolean_has_overlap_with_other_tracks

As we mentioned in Chapter 5, a true basketball track does not overlap with other tracks in time. If a track overlaps with one or more other tracks after the preprocessing steps, we consider those tracks as false tracks due to false basketball detections and camera movements. We only keep those track items with

Algorithm 2 Preprocess Tracks Step 1 (write tracks with continuous track_id)

```
# Tracks file item format: image_seq_name ball_start_pos_x ball_start_pos_y
ball_end_pos_x ball_end_pos_y track_id
# The tracks items are sorted according to image_seq_name
Open and read tracks file
Open a prep1 file to write the preprocessing results
for track_item in tracks file do
    number_of_following_frames = 0
    while track_item.track_id has continuous same following track_id do
        increase number_of_following_frames by 1
    end while
    write item: track_id track_start_frame number_of_following_frames
end for
```

Algorithm 3 Preprocess Tracks Step 2 (Check track overlaps)

```
Open and read prep1 file
Open a prep2 file to write the preprocessing results
idList = empty idList # keep records of all written track_id
for track_item in prep1 file do
    current_id = track_item.track_id
    if not current_id in idList then
        # create a new item in the prep2 file
        count = number_of_following_frames
        boolean_has_overlap_with_other_tracks = False
        # count and add all together
        for remaining track_item in prep1 file do
            if remaining track_id == current_id then
                count = count + number_of_following_frames
                boolean_has_overlap_with_other_tracks = True
            end if
        end for
        # write the item to prep2 file
        idList.append(current_id)
        write item with format: current_id track_start_frame count
boolean_has_overlap_with_other_tracks
    end if
end for
```

boolean_has_overlap_with_other_tracks equal to False for the player inference.

Given the ball tracks without any overlaps with each other, and the knowledge of which player each track starts with and ends at from Chapter 5 (saved in the track_id in each track item), we can simply fill in the time gaps between the tracks by inferring the controlling player of the ball, using Algorithm 4.

If all the ball tracks are true basketball tracks, then the inference algorithm works 100% correctly, and the track ending player is always the next track's starting player. However, in our tracks result, it is not always correct because of some missing tracks (due to occlusions, low detection rate, etc.) and some false positive basketball tracks. We handled the case by simply dividing the gap between current track and next track into 2 parts, with the assumption that the first part being controlled by current track ending player, and the second part being controlled by the next track starting player.

Algorithm 4 Player Inference

```
Open and read prep2 file
Open a player_inference file to write the inference results
Set image_seq_start_frame, image_seq_end_frame
prev_track_start_player = ”
prev_track_end_player = ”
for track item in prep2 file do
    if track_item.boolean_has_overlap_with_other_tracks == False then
        # get the values
        Set start_player, end_player from track_item.track_id
        Get track_start_frame
        Compute track_end_frame

        # link the gaps
        if prev_track_end_player == ” then # the initial gap
            Fill initial gap from image_seq_start_frame to track_start_frame with start_player
        else
            if prev_track_end_player == start_player then
                Fill the gap from prev_track_end_frame to track_start_frame with start_player
            else# split the gap in 2 parts
                middle_frame = (prev_track_end_frame + track_start_frame) / 2
                Fill the gap from prev_track_end_frame to middle_frame with prev_track_end_player
                Fill the gap from middle_frame to track_start_frame with start_player
            end if
        end if

        # set current track item to be the previous track item
        prev_track_start_player = start_player
        prev_track_end_player = end_player
        prev_track_start_frame = track_start_frame
        prev_track_end_frame = track_end_frame
    end if
    # Fill in the last gap
    Fill the gap from track_end_frame to image_seq_end_frame with end_player
end for
```

Chapter 7

Experiments and Results

7.1 Ground Truth Data for Evaluation

We evaluated our system on 830 frames of the National Basketball Association Broadcast Video in 2011. We annotated the basketball ground truth on these frames for the purpose of evaluation.

The video was originally in MPEG4 format and was converted into consecutive JPEG images using the FFMPEG tool, and the image sequences were divided into sequences of passing and shooting actions. We took 830 frames (013031.jpg - 013860.jpg) and evaluated the system performance on them.

We hand labelled the position of the basketball if it is visible in these frames. In each frame, if the basketball is visible, we use a bounding box to indicate the basketball position. In addition, we specify the basketball motion status using one of the three labels: *player*, *pass*, or *shoot*. The *player* label indicates the basketball is either in a player's hand, or being dribbled by the player. The *pass* label indicates the basketball is being passed from one player to another player. The *shoot* label indicates the basketball is being shot into the basket. If the basketball is occluded by the player or too blurred to be identified, we do not label it in that image frame.

Figure 7.1 shows one image frame with the bounding box and label *shoot*, and the trajectory of the previous frames with the *shoot* label.

Figure 7.1 has been removed due to copyright restrictions. It was a frame showing the hand-labelled bounding box of the basketball and its ground truth basketball shot trajectory used for evaluation. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 7.1: Ground truth bounding box and the trajectory (image sequence frame: 013103.jpg)

Figure 7.2 has been removed due to copyright restrictions. It showed detection results on 4 frames with cases (a), (b), (c), (d). The detection results using the three different detection approaches were shown in different colors, and the detection results after the player region removal step were also shown seperately. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 7.2: Basketball detection results

7.2 Experiments and Error Measurements

7.2.1 Basketball Detection Experiments

Figure 7.2 shows some basketball detection results: the basketball candidates detected using color (mentioned in Section 4.1) are shown in red bounding boxes; the candidates detected using motion information (mentioned in Section 4.2) are shown in yellow bounding boxes; the candidates detected using differenced affine flow image (mentioned in Section 4.3) are shown in green bounding boxes. The combined bounding boxes with player region removal step mentioned in Section 4.4 are shown in the second column in the table.

Table 7.1: Basketball Detection Candidates Count

Method	Candidates #
Basketball Color Detection	5091
Basketball Motion Detection	2108
Basketball Differenced Shape Detection using Affine Flow	142
Sum of All Detections	7341
Combined Detections after Player Region Removal	3746

Table 7.2: Basketball Detected Frames

	Count	Recall
Total Number of Frames	830	
Number of Frames with True Basketball	696	
Number of Detected Frames using Color	247	35.5%
Number of Detected Frames using Motion	91	13.1%
Number of Detected Frames using Affine Flow	28	4.0%
Total Number of Detected Frames	314	45.1%
Number of Detected Frames after Player Region Removal	211	30.3%

In the 830 image frames, there are 696 ground truth basketball detections. Table 7.1 counts the number of detections after each step. On average, there are 8.84 basketball detection candidates in each frame before the player region removal step, and 4.51 candidates in each frame after the player region removal step.

If the combined detection after the player region removal step contains a basketball candidate that overlaps with the labelled ground truth bounding box, we consider the true basketball is detected in that frame. The number of detected frames using different approaches are shown in Table 7.2. Recall is computed using # of detected frames divided by # of frames with true basketball.

The false basketball candidates generated using the color detection approach are mainly due to the players' / audience's skin color on the head, arms, and legs, as well as the reddish logo color on the basketball court. The false candidates generated using the motion / affine flow detection approach are mainly due to the fast player motions; when the camera is moving fast from one side of the court to the other side, false candidates are generated in some noisy parts in the audience

Table 7.3: Basketball Detected Pass / Shoot Frames

	Count	Recall
Total Number of Frames	830	
Number of Pass / Shoot Frames with True Basketball	94	
Number of Detected Pass / Shoot Frames	74	78.7%

Table 7.4: Basketball Tracklets Count

	Count
# of tracklets using Kalman Filter	73
# of tracklets after passing / shooting action selection	70
# of tracklets after the three criteria	15
# of non-overlapping tracklets in player inference	8

region as shown in Figure 7.2 (d).

It is hard to distinguish the false candidates with the true basketball under small scale and the compression and camera blurring effects, but we can use the fact that the false candidates are mostly inside the player regions to eliminate all candidates that are totally inside the player region, and focus on finding the passing and shooting actions in the video, then infer the subsequences where the basketball is being controlled by players. We listed the detection rates on those frames with pass / shoot labels in Table 7.3. Recall here is computed using # of detected pass / shoot frames divided by # of pass / shoot frames with true basketball.

7.2.2 Basketball Tracking and Player Inference Experiments

We manually counted the number of passing / shooting trajectories in the 830 testing frames. There is one long shot and 7 passing actions in these frames. We counted the number of tracklets we got after each step after the Kalman Filter tracking and track analysis steps, as shown in table 7.4.

In the 8 non-overlapping tracklets we obtained before the player inference step, the one shot trajectory is included, and 5 out of the 7 passing actions are also included in the tracklets. The hand labelled ground truth trajectories and the tracklets we obtained in the analysis are shown in Figure 7.3. The remaining 2 tracklets are

Figure 7.3 has been removed due to copyright restrictions. It showed the 1 ground truth shooting and 5 passing trajectories and the corresponding tracklets generated using our algorithms. The tracklets generated by us were similar to the ground truth trajectories. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 7.3: Ground truth trajectories and tracklets

Figure 7.4 has been removed due to copyright restrictions. It showed the two remaining false tracklets generated using our algorithms. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 7.4: False tracklets

false tracklets, shown in Figure 7.4.

Two passing actions were not tracked in our algorithm. The ground truth trajectories are shown in Figure 7.5. In the left missed trajectory (a), one player made a bounce pass to another player: the basketball was occluded by other players when it hit the court, so it could not be detected in those frames; the remaining visible frames are too short to form a tracklet to be considered as a valid passing action. In the right missed trajectory (b), although the basketball is visible in most frames, it passes on top of another player, those detections were removed in the player region removal step; therefore, no tracklet was formed.

The major reason that caused a missing tracklet is the low basketball detection rate. In our algorithm, occlusions, overlaps with other players, and image / motion blur all can result in a low basketball detection rate. These problems remains to be tackled.

Figure 7.5 has been removed due to copyright restrictions. It showed the two ground truth passing trajectories which were not successfully tracked using our algorithms. Original source: National Basketball Association Broadcast Video (2011), ESPN and ABC Network.

Figure 7.5: Missed ground truth trajectories

The player inference step highly relies on the accuracy of the track analysis results: if the track analysis generate tracks with 100% accuracy, then the player inference results are 100% correct too. Since we have already reported our track analysis results in the Figures 7.3, 7.4, and 7.5, we do not see the necessity to set up a qualitative measurement for the player inference accuracy here.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Our proposed basketball tracking system demonstrated the ability to extend the current intelligent sports video analysis system for broadcast video to include ball tracking and player possession inference even when the ball movement in the sport is not generally constrained to the two dimensional space of the court.

Despite the small basketball with few features and the cluttered background, we proposed three independent detection approaches to try to find the true basketball candidate. Training and using color can detect the basketball with the 35.5% recall accuracy; in addition, because we know the basketball is usually the fastest moving object in the broadcast video, including the detections using the optical flow and affine flow increased the recall by 17.1%.

If we focus the detections on the ball passing / shooting frames, the recall accuracy is 78.7%; it is harder to detect the basketball on the remaining frames, because the ball can be connected with player's hand, overlaps with player's skin color, or occluded – color and size filtering cannot work when the ball overlaps with regions with similar color, and the ball is not moving very fast when it is controlled by a player. Our strategy to deal with the situation is to track the passing / shooting actions in our system, and then infer which player controls the ball before or after the passing / shooting actions.

The Kalman Filtering generated 73 tracklets in our experiment. We performed track analysis and selected 15 parabola-like tracklets. In the player inference part, we further selected 8 non-overlapping tracklets out of the 15 tracklets to infer the player possession. One shooting trajectory and 5 out of 7 true passing trajectories were found by our system, while 2 true trajectories were still missed in our system. The player possession inference accuracy depends on the accuracy of the selected passing / shooting tracklets.

8.2 Future Work

The following are some possible directions to improve the ball tracking system in the broadcast video.

8.2.1 Probabilistic Detections

In our current proposed system, all the three basketball detection approaches make binary decisions about the basketball candidates, and the candidates are combined using the logical OR operation.

We can consider adopting some confidence measures in all the three detection approaches, and track the basketball with the confidence measures. In addition, when we combine the detections using the different detection approaches, instead of using the logical OR operation, we can also train and assign some weighting in different detection approaches and then combine the detections with a new weighted confidence.

8.2.2 Tracking Methods

We used the Kalman Filter to track the basketball, under the assumption that the basketball motion correspondence is straightforward. Predictions are made as to the expected locations and the predictions are matched to actual measurements. Ambiguities may arise here because predictions may not be supported by measurements. The basketball might move outside the frame or be occluded; or, for example, more than one measurement might match a predicted location, etc.

To deal with the ambiguous motion correspondences, Kalman Filtering and a cross correlation measure to compare the basketball measurements [SWB92] could be adopted. The tracking splitting filter [SB75] or multiple hypothesis tracking

[CH96] can use track trees to delay correspondence decisions until more measurement evidence is available. If we adopt one of these tracking methods or others such as [SBF$^+$11], the tracking results should be able to handle frames with the occluded basketball better.

8.2.3 Splitting the Tracklets

In our track analysis, we used RANSAC to refit a long tracklet and generated more accurate confidence value to indicate how similar the tracklet is compared with a parabola. However, instead of using the refitted parabola-like part of the tracklet, we kept the whole tracklet in order to use it to connect to the rest of the play.

We can try to split the long tracklet into 3 parts: the parabola-like part of the tracklet, and the remaining two parts. The parabola-like part of the tracklet is probably the passing / shooting trajectory, and we can possibly infer some more information using the remaining parts. For example, if the part after the parabola-like part is a falling trajectory down to the court, then this tracklet is probably a shooting tracklet (Figure 5.7). It is possible to consider the falling part as a category of motion, after a shot.

8.2.4 Player Inference

Our current player possession inference algorithm is a simple inference approach based on the assumption that the tracklets we selected are mostly correct tracklets. Errors are introduced if there is any missing passing / shooting trajectory, or any false passing / shooting tracklet.

We assumed that those tracklets overlaping with other tracklets are false passing / shooting tracklets. Those tracklets are usually generated because of the fast camera movement, and therefore the false basketball detections. However, a true passing / shooting trajectory might also exist among those tracklets. One possible way to deal with the situation is to split those long tracklets into small segments (several new tracklets) and re-run the inference step.

The player possession inference part can use more prior knowledge or adopt some advanced algorithms to increase the inference accuracy.

8.2.5 Player Pose Estimation and Other Information for Inference

Detecting and tracking the basketball itself is a hard problem, because the ball is small and lacks distinctive features. However, prior knowledge is a great assistance in determining where the ball is. We used the player tracking information in our system. More information could be used in the future work.

As mentioned and attempted in the related work, player pose estimation could be a very helpful clue to detect the basketball. Current difficulties in player pose estimation include the low player resolution and the many non-frontal poses. If the difficulties are tackled, we could focus on the player's hand region in order to find the basketball. In addition, we could possibly train, learn, and infer the player's action and then predict the trajectory of the ball.

In addition, we can also consider analyzing the score display in the broadcast video, and the which half court the basketball is in to infer the defense and offense team, and use the information to help with basketball detection and tracking analysis.

Bibliography

[AK99] T. J. Atherton and D. J. Kerbyson. Size invariant circle detection. *Image and Vision computing*, 17(11):795–803, 1999. → pages 11

[AR11] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16:1–16:43, April 2011. → pages 8

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004*, pages 25–36, 2004. → pages 23

[Bir99] S.T. Birchfield. *Depth and motion discontinuities*. PhD thesis, Stanford University, 1999. → pages 5

[Cav97] Rick Cavallaro. The FoxTrax hockey puck tracking system. *IEEE Computer Graphics and Applications*, 17:6–12, 1997. → pages 12

[CH96] I.J. Cox and S.L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(2):138–150, 1996. → pages 50

[CP11] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. → pages 15

[CTC+09] Hua-Tsung Chen, Ming-Chun Tien, Yi-Wen Chen, Wen-Jiin Tsai, and Suh-Yin Lee. Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video. *J. Vis. Comun. Image Represent.*, 20(3):204–216, April 2009. → pages 14, 18

[DACN02] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1 - Volume 1*, ICPR '02, pages 10210–, Washington, DC, USA, 2002. IEEE Computer Society. → pages 11

[Dua11] Xin Duan. Automatic Determination of Puck Possession and Location in Broadcast Hockey Video. Master's thesis, the University of British Columbia, August 2011. → pages 7, 18

[EF09] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *British Machine Vision Conference*, September 2009. → pages 9

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. → pages 5, 35

[FMJZ08] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. → pages 9

[FMJZ09a] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. → pages 9

[FMjZ09b] Vittorio Ferrari, Manuel Marn-jimnez, and Andrew Zisserman. 2D human pose estimation in TV shows. In *Dagstuhl post-proceedings*, 2009. → pages 9

[GLW11] A. Gupta, J.J. Little, and R.J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 32–39. IEEE, 2011. → pages 5

[GSC+95] Yihong Gong, Lim Teck Sin, Chua Hock Chuan, Hongjiang Zhang, and Masao Sakauchi. Automatic parsing of TV soccer programs. In *Multimedia Computing and Systems, 1995., Proceedings of the International Conference on*, pages 167 –174, May 1995. → pages 11

[Gup10] Ankur Gupta. Using Line and Ellipse Features for Rectification of Broadcast Hockey Video. Master's thesis, the University of British Columbia, December 2010. → pages 5

[Hol12] James Holloway. FIFA trial goal-line technology in international soccer match ahead of key vote (last date accessed 05/31/2012), May 2012. Available from: http://www.gizmag.com/goal-line-technology/22686/. → pages 12, 13

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. → pages 5

[LOL09] Wei-Lwun Lu, Kenji Okuma, and James J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(12):189 – 205, 2009. → pages 6

[Lu11] Wei-Lwun Lu. *Learning to track and identify players from broadcast sports videos*. PhD thesis, the University of British Columbia, October 2011. → pages 1, 2, 6, 30

[Mes99] M. Mesbah. Gradient-based optical flow: a critical review. In *Signal Processing and Its Applications, 1999. ISSPA '99. Proceedings of the Fifth International Symposium on*, volume 1, pages 467 –470 vol.1, 1999. → pages 15

[OLL04] Kenji Okuma, Jim Little, and David G. Lowe. Automatic acquisition of motion trajectories: Tracking hockey players. *In Proc. of SPIE Internet Imaging V*, 2004. → pages 5, 6

[OTF+04] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *the European Conference on Computer Vision*, pages 28–39, 2004. → pages 6

[PHVG02] P. Prez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *In Proc. ECCV*, pages 661–675, 2002. → pages 6

[POJ00] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *Pattern Recognition, 2000.*

Proceedings. 15th International Conference on, volume 4, pages 152–156. IEEE, 2000. → pages 13

[Rui10] Neus Agelet Ruiz. Tracking of a Basketball Using Multiple Cameras. Master's thesis, Ecole Polytechnique Federale De Lausanne, July 2010. → pages 13

[SB75] P. Smith and G. Buechler. A branching algorithm for discriminating and tracking multiple objects. *Automatic Control, IEEE Transactions on*, 20(1):101–104, 1975. → pages 49

[SBF+11] H. Ben Shitrit, J. Berclaz, F. Fleuret, , and P. Fua. Tracking Multiple People under Global Appearance Constraints. *International Conference on Computer Vision*, 2011. → pages 50

[SCKH97] Yongduek Seo, Sunghoon Choi, Hyunwoo Kim, and Ki-Sang Hong. Where are the ball and players? soccer game analysis with color based tracking and image mosaick. In *Proceedings of the 9th International Conference on Image Analysis and Processing-Volume II*, ICIAP '97, pages 196–203, London, UK, UK, 1997. Springer-Verlag. → pages 11, 18

[ST94] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994. → pages 5

[SWB92] L.S. Shapiro, H. Wang, and J.M. Brady. A matching and tracking strategy for independently moving objects. In *Proceedings of the British Machine Vision Conference*, pages 306–315, 1992. → pages 49

[Tar11] Shervin Mohammadi Tari. Automatic initialization for broadcast sports videos rectification. Master's thesis, the University of British Columbia, December 2011. → pages 6

[TK91] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon University, 1991. → pages 5

[TLL04] Xiao-Feng Tong, Han-Qing Lu, and Qing-Shan Liu. An effective and fast soccer ball detection and tracking method. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04*, ICPR '04, pages 795–798, Washington, DC, USA, 2004. IEEE Computer Society. → pages 12

[YJS06] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006. → pages 10

[You10] David Young. Affine optic flow (last date accessed 05/31/2012), March 2010. Available from: http://www.mathworks.com/matlabcentral/fileexchange/27093. → pages 25

[YSM02] A. Yamada, Y. Shirai, and J. Miura. Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 303–306. IEEE, 2002. → pages 11

[YYYL95] D. Yow, B.L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *Asian Conference on Computer Vision*, pages 499–503, 1995. → pages 11

Appendix A

Supporting Materials

List of parameters decided empirically:

- Ball RGB color detection ranges: red channel [90, 136], green channel [45, 80], blue channel [32, 79]
- Ball center pixel selection: there are at least 60 ball-color-like pixels in the nearby 21*21 pixel region; sum of the difference of the pixels centered around the current pixel within radius 10 minus the average ball is smaller than 20000.
- Laplacian of Gaussian filter in motion detection: size [25, 25] and sigma = 6
- Passing / shooting action selection: tracklets that have NO associated at least k (k = 8) frames, with at most d (d = 2) frames noise