View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by ScholarBank@NUS

AN EFFECTIVE TRAJECTORY-BASED ALGORITHM FOR BALL DETECTION AND TRACKING WITH APPLICATION TO THE ANALYSIS OF BROADCAST SPORTS VIDEO

YU XINGUO

NATIONAL UNIVERSITY OF SINGAPORE

2004

AN EFFECTIVE TRAJECTORY-BASED ALGORITHM FOR BALL DETECTION AND TRACKING WITH APPLICATION TO THE ANALYSIS OF BROADCAST SPORTS VIDEO

YU XINGUO
(M.Eng, NTU)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

I would like to express my sincere gratitude to Assoc. Prof. Hon Wai Leong, my supervisor, for his time and constant guidance during this research. His invaluable suggestions, honest criticisms, and the constant encouragement were a great resource of inspiration. His immense enthusiasms, high standards for excellence have a great influence to this research and will benefit me all the rest of my life. I also would like to thank my PhD research guidance committee--Assoc. Prof. Wee Kheng Leow, Asst. Prof. Teck Khim Ng, Dr. Qi Tian for their useful comments and suggestions.

I wish to thank Professor Shih-Fu Chang, Professor Jesse Jin, and Dr. Yihong Gong for their suggestions and comments. I consider it my good fortune to have their comments and suggestions when I frequently met them in USA and Singapore during this research.

I am especially grateful to Dr. Changsheng Xu. He has given me his full support to this research and he also constantly gives me his comments and suggestions. Thanks to Dr. Liyuan Li, Mr. Joo Hwee Lim, Dr. Dongyan Huang, Dr. Ruihua Ma, Dr. Loong Fah Cheong, Dr. Xiaofan Liu, Mr. He Dajun, Mr. Mingjiang Yang, Mr. Kong Wah Wan, Mr. Lingyu Duan, Mr. Xin Yan, Miss Min Xu, Miss Jenny Ran Wang, and Mr. Xi Shao for many useful discussions and detailed comments. Thanks to Mr. Tze Sen Hay and Mr. Chern-Horng Sim for his manual work and doing experiments for some algorithms in thesis.

I would like to thank Institute for Infocomm Research for providing a good research environment for my research. Thanks to the library of National University of Singapore for providing rich reference materials for my research.

Finally, I wish to express my gratefulness to my wife, Jing Xia and my son Zhuoran Yu for their love, sacrifice, and encouragement.

Contents

Summary

A trajectory of an object contains more information than a single object. Due to this reason, trajectory analysis has been used in computer vision for some time. In particular, trajectory analysis is useful for ball detection and tracking in sports video as there are some non-ball objects that look like the ball. However, a non-ball object does not form significant trajectories or forms different trajectories from ball trajectories in various aspects. Using these properties, we discriminate the ball trajectory from the ball-like object trajectory. Furthermore, the ball might be occluded, deformed, or out of the camera temporarily. Using trajectory enables suppression of these problems for reliable location of the ball. The ball locations have a close correlation with the ball-related events in the ball game video. Hence, the ball locations significantly facilitate the event detection. The ball is viewers' attention in watching ball games. Therefore, one of the main objectives in generating and enhancing the ball game video is to reconstruct the ball and to illustrate the ball motion. In other words, the ball locations play an important role in the enhancement and enrichment of ball game video.

This thesis addresses three closely-related problems. It first addresses the ball detection and tracking problem in broadcast sports video. It proposes an effective trajectory-based algorithm for detecting and tracking the ball in a broadcast sports video, which can obtain the accurate results for locating the ball in

broadcast soccer/tennis video. The key idea of this approach is as follows: a non-ball trajectory might contain some objects that look like the ball but such objects have a small ratio in the trajectory. On the other hand, a ball trajectory may also contain some objects that do not look like the ball, but most of its objects would be ball-like. Unlike the object-based approach, we do not evaluate whether a sole object is a ball. Instead, we evaluate whether a trajectory is a ball trajectory. As a result, the ball trajectory can be produced reliably. Then, this thesis applies ball detection and tracking to two problems: ball-related event detection and enhancement and enrichment of broadcast soccer video (BSV). For the first application problem, it proposes a trajectory-based event detection approach, which improves the event detection performance because the events closely correlate with the ball location than with the low-level features. More importantly, this approach can detect some events that cannot be detected if one just uses low-level features. For the second application problem, it proposes an enhancement and enrichment system for BSV. This system is better than the existing systems as it automatically approximates the 3D position of the ball, extends the reconstruction range, and enriches the video by illustrating the contents of video. In addition, this thesis proposes a robust ellipse Hough transform and applies it to detect the ellipse in BSV. The detected ellipse is used to estimate the ball size in locating the ball in BSV and provide the feature points for reconstructing the midfield scene of BSV.

List of Figures

List of Tables

Abbreviations

3D	**Three-Dimensional**
AFEHT	**Accumulator-Free Ellipse Hough Transform**
AMF	**Absolute Measure Function**
BV	**Broadcast sports Video**
BSV	**Broadcast Soccer Video**
BTV	**Broadcast Tennis Video**
CEHT	**Combinatorial Ellipse Hough Transform**
CFI	**Candidate Feature Image**
CL	**Central Line**
EHT	**Ellipse Hough Transform**
FCV	**Fixed-Camera Video**
FIFA	**Fédération Internationale de Football Association**
NEHT	**Normalized Ellipse Hough Transform**
NMF	**Normalized Measure Function**
REHT	**Random Ellipse Hough Transform**
RobustEHT	**Robust Ellipse Hough Transform**
RSV	**Real Soccer Video**
SEHT	**Standard Ellipse Hough Transform**
UMF	**Unbiased Measure Function**

Chapter 1

Introduction

1.1 Motivation

Sports video is one of the most popular forms of entertainment in the world, touching many people from various cultures in the world. With consumers' demand and the great technological advances in recent year in video production technology, sports videos are produced in large quantity annually. However, it is well known that large portions of a sport video are routine and fairly boring to watch and few viewers are interested in watching the entire video. Most viewers want to watch only the *interesting events* in the video. In fact, currently consumers can afford the *money* to pay for accessing huge volumes of video (partly because the cost of producing video is now very low), but they cannot afford the *time* to *find and view the portions* of the video that they want. What is needed is a system that allows users to retrieve *only* the segments that they are interested in viewing, thus saving time and money.

In recent years, there has been a great deal of work on the development of efficient *indexing and retrieval systems* for sports video. These systems aim to allow users to *efficiently* and *accurately* search a large database of sports video for the specific segments that they are interested in viewing. By *efficient*, we mean a system that is fast in answering query, and

by *accurate*, we mean that the system will return video segments that satisfy the specification given by a particular user.

Generally speaking, the consumers (or *viewers*) of sports video are interested in the video segments that contain specific "*interesting events*" in a game and not in viewing the entire video. For example, in a soccer game, viewers may be interested in segments where specific soccer events occur such as when (a) goals are scored, (b) a corner kick is given and taken, (c) their favorite player is shown, or (d) ball possession is changed from one team to another. Hence, one key task in building indexing and retrieval system for sports video is that of *identifying the sport-specific events within the video*. These events are specific to and defined by the sports and are usually well-known to both players and viewers of the sport. For example, in soccer these events can be goals, corner kicks, free kicks, penalty shots, etc. In tennis, the examples of these events are scoring, serving, and play/break.

Manual *identification and indexing* of these sports-specific events in the broadcast sports video are being done for some specific purposes. For example, currently media companies employ a group of experts to identify several most interesting events from a just-happened sports game to form a sports news video. However, this manual process is tedious because of the sheer volume of sports video produced nowadays.

Given this scenario, it is not surprising that the problem of automatic detection and indexing of events from sports video became a hotly researched topic in recent years. Although many research and development efforts have been undertaken, the problem of *automatic* event detection and indexing in sports video is still not solved, at least, not well-solved. Current

research efforts on event detection for sports video falls in three main directions as described in the following:

- The first direction is to build the generic framework for semantic shot classification of the sports videos including soccer, basketball, tennis, etc [DXTX2003, DXTX2004]. The framework performs a top-down shot classification, including human identification of shot categories for a specific sports game, visual and auditory feature representation, and supervised learning. The classified shots are further used to facilitate event detection and other semantic analysis.
- The second direction is to detect events based on low-level features [XXCD2001, XCDS2002, XDXT2003, Eki2003]. The above two directions analyze the video in different ways, but they both work on the low-level features[1], which are mainly video features (such as color, texture, and motion) and audio features (such as pitch, whistling, and crowd cheering/excitement).
- The third direction is to detect events based on object-related features associated with the sports. This research direction is motivated by the relatively low accuracy obtained by algorithms that detect events using *only the low-level features*. As a result, researchers have moved to incorporate the detection of object-related features in order to improve the performance of event detection in their algorithms [GLCZ1995, HMSP2002, CHHG2002, CHHG2003]. In many ball games, most of the interesting events closely correlate with the *ball location and*

[1] In these, "low-level features" mean the features derived from the audio, motion, color, and texture. Such low-level features were also called cinematic features in some recent papers [EkTe2003d, YaLC2004]. In contrast to "low-level features", "object-related features" are the features derived from the detected objects. For example, in the soccer video the features derived from goalmouth, ellipse, and the ball are object-related features.

motion. In soccer, for example, *kicking, passing, team possession and goal (scoring)* are all events that are closely related to the motion of the ball. Hence, an increasing interest has been paid to the ball detection and tracking problem for the videos of ball game [DACN2002, DGLD2004, SCKH1997].

In summary, the general problem of designing good indexing and retrieval systems for broadcast sports video remains a challenging research problem. Presently, no system can do a very good job of accurate retrieval from the huge volume of sports video in a short time. The problem is set to grow more complex because of the increasingly fast pace in which these sports video are produced in recent years and in the future.

1.2 Overview of Research

The overall goal of this research is to design better automatic indexing and retrieval systems for broadcast sports video. We aim to do this by improving event detection algorithms to automatically detect events which are then used for indexing the video. As discussed in the preceding section, there are several research directions in doing automatic event detection. In this thesis, we focus on the third direction, namely, the event detection approach that *uses a combination of low-level features and object-related features (such as ball position and motion)*.

We choose to study this approach because it can be used to handle *complex* sports video such as broadcast soccer video (denoted by BSV in this thesis). BSV is generally considered to be complex because of the general lack of “structure of play” during the game unlike games such as tennis. In

addition, the quality of the video is generally low in BSV. As a result, automatic event detection for BSV is generally considered to be harder.

We first apply this event detection approach to BSV. On the one hand, BSV is a complex case and so we believe that solving this case will make it more likely that our methods can be applied to other sports video. On the other hand, soccer is a very popular sport that appeals to audiences around the world, and so, is in great demand. Therefore, it is quite natural to use BSV as a first candidate.

A key observation by many researchers is that in BSV (and other sports video), the information derived from the accurate location of the ball can play a *crucial role* in automatic event detection. It is well-known that this information greatly improves event detection in general [QiTo2001, ABCB2003a]. Many events such as goal, break, and possession closely correlate with the location and motion of the ball and its position relative to nearby objects. For example, in soccer (and many other games, including tennis), to determine if the ball is *in play or out*, the location of the ball relative to the out-of-bound lines is the most crucial determining factor. In a more complex example, to determine *ball possession* in soccer, the location of the ball relative to the players in the frame is very important even if it is not to sole deciding factor. Therefore, we can expect to improve the accuracy of event detection *by first achieving a higher accuracy in the detection and tracking of the ball in broadcast soccer videos*. This motivates our first research problem.

1.2.1 Ball Detection and Tracking for Broadcast Soccer Video

In this thesis, we first study the problem of *ball detection and tracking* in broadcast soccer video (BSV), which plays a very important role in improving event detection and in soccer video analysis in general. More specifically, we want an algorithm to *efficiently* and *accurately* detect and track the ball in a BSV, namely, determine the location of the ball (if it is visible) in each frame of the given BSV. By *efficient*, we mean procedures that are fast (polynomial in complexity) and by *accurate* we mean the usual metrics of low *false negatives* (not identifying a ball when it is visible) and low *false positives* (wrongly identifying a ball when none is visible or wrongly identifying the location of the ball).

The ball detection problem is a deceptively challenging problem to solve *accurately*. Despite much research work done on ball recognition from video images, it is still very challenging to do ball recognition from broadcast soccer video with *high accuracy* (say, in the range of 10% for false positives and 5% for false negatives). Informally, we can see the reason as follows: the image frames in BSV can be classified into "*close-up*", "*middle-view*" and "*far-view*". Ball detection for close-up frames can be done with high accuracy with many existing methods. However, they form only the minority of the frames in a BSV. In the majority of the frames, (namely the middle-views and far-views), the ball is small relative to other objects in the frame and ball detection remains a big challenge.

Existing methods that directly recognizes the ball from video images are good, but they are limited by several inherent difficulties associated with direct recognition methods. Some of these difficulties include (a) the presence

of many ball-like objects in the image, (b) the small size of the ball relative to the image size, (c) occlusion of the ball (say, by players) in many images, and so on. Because of these inherent difficulties associated with broadcast sports video, direct recognition methods are limited in its accuracy. Figure 1.1 shows the ball and the ball-like objects from a frame, testifying to the above-listed difficulties. To overcome these challenges and barriers with high accuracy in ball detection and tracking, we adopt a strategy, which we call a *trajectory-based strategy,* to develop offline detection and tracking algorithms. Originally, the trajectory-based strategy was popularly used in online tracking algorithms [Cox1993, SmBu1975, ZhFa1992]. In this strategy, there are two steps: in the first step, we reduce the rate of false negatives (at the price of a temporarily higher rate of false positives) by extending the search to ball-like objects, thus getting a number of candidate ball-like objects. Then, in the second step, we use information of the *path trajectories* of these candidate objects *over a short sequence of frames* to obtain the ball (and prune off the non-ball candidates). Thus, in this step, we recover from the higher rate of false positives by throwing the non-ball trajectories.

(a) (b) (c)

Figure 1.1 A soccer frame and its ball and ball-like objects. The frame is shown in (a); the ball in the frame is shown in (b); the ball-like objects in the frame are shown in (c).

Informally, then, the main idea behind this is that while it is very difficult to achieve high accuracy when locating just the ball, it is relatively easy to

achieve very high accuracy in locating ball-like objects (the first step). This significantly reduces the rate of false negatives. To eliminate the false positives, it is much better to study the trajectory information of the ball since the ball is the “most active” object in soccer video, as well as in most other sports video. For example, a ball-like object (say image of a ball on a T-shirt) is not likely to move significantly during the game. We believe that the strength of our new strategy comes mainly from the careful control of false positives in the first step and the trajectory-based processing in the second. Indeed, our research results show that the trajectory-based strategy can *greatly enhance the accuracy* of ball detection and tracking in BSV. The details of the methods and the results obtained are described further in this thesis.

Ball Detection and Tracking in Tennis: With the encouragement of the success for the case of soccer, we then apply the trajectory-based strategy to the case of tennis. Namely, we consider the *ball detection and tracking* in broadcast *tennis* video (BTV). The problem is very similar, but there are some unique challenges in the case of tennis: the tennis ball is smaller (harder to identify, especially when it is close to the “far” player) and much faster. In this application, we augment our two-step trajectory-based strategy with other game-related features such as player locations, hitting points (and turning points) to improve accuracy of ball candidate locations and in getting greater accuracy based on ball trajectories. Our results show that our trajectory-based strategy can improve the accuracy of ball detection and tracking for broadcast tennis video.

1.2.2 Applications of Ball Detection and Tracking

After achieving a higher accuracy in the detection and tracking of the ball in broadcast soccer videos, we turn to the solution of a number of ball-related problems associated with broadcast sports video analysis. They are event detection and enrichment of broadcast soccer video.

Detection of Ball-Related Events in BSV: Recall that many events in soccer (and other games) are highly dependent on the location of the ball and its position relative to nearby objects (players) and the field of play. Many existing event detection algorithms are based on the low-level features.

We shall focus on *ball-related* events which are events that involve the interaction between player(s) and the ball that usually result in the change of the location of the ball in the soccer field. For example, a *kick* happens when a player kicks the ball and the trajectory of the ball is changed. A *goal* happens when the ball goes past the goalmouth. Other examples are *passing, shooting, play/break,* and *team possession*. Ball-related events cover the majority of interesting events in most games and are usually the focus of viewer's attention.

While these events are closely related to the location of the ball, the ball location alone is not sufficient to characterize many of these ball-related events. We need to augment the trajectory-based approach with other game specific actions and characteristics. Thus, our strategy for ball-related event detection is to first express a ball-related event as a set (or sequence) of simpler (game specific) basic actions (or sub-events). We first define a series of (game specific) basic actions that are based on the location and trajectory of the ball. For example, touching of the ball (a player coming into contact with

the ball physically), kicking of the ball, and passing of the ball. These basic actions can be accurately determined using our trajectory-based approach since they usually define “pivot points” that correspond to changes in the trajectory of the ball. Then, the results of these basic actions can be used in combination with other standard approaches to detect more complex ball-related events.

Enhancement and Enrichment of Broadcast Soccer Video: We then studied the problems of the enhancement and enrichment of broadcast soccer videos. By *enhancement,* we mean to generate the soccer video based on the camera calibration results. In generating frames, we first render the 3D model of soccer field and the ball. Then we superimpose the images of segmented players. By *enrichment,* we mean to augment the generated video with the icons that illustrates video contents. The problem is difficult due to the absence of feature points in the frames. Several existing systems focus on rendering only the goalmouth scene (to determine if a goal has been scored). This sub-problem is made easier by the presence of salient feature points near the goalmouth to aid in the camera calibration process.

In this research, we are interested in extending this to generating video of the *midfield scene*. Our approach is to extract the feature points from the central circle in the midfield to do camera calibration. To do so, we need a highly accurate ellipse detection algorithm and we use the one described in the next subsection for this purpose.

Once we have performed camera calibration, we can approximate the world location of the ball. Furthermore, from the work on ball detection and tracking and event detection, we already know or can easily compute the

apparent velocity or speed, and direction of the ball, the team possession information, the direction of the camera, the event that is happening at the moment, etc. We can then perform “enrichment” by augmenting the frames in the video with these (or other) information as icons or illustration windows, as well as the matched music. The enriched soccer video will enhance the viewing experience.

1.2.3 Ellipse Detection in Broadcast Soccer Video

In the course of this research on soccer video analysis, we discover that it is very important to have an *accurate* and *robust* ellipse detection algorithm. Ellipses are very common in broadcast sports video since all round objects are transformed into ellipses in the video. The ellipses we want to detect in this thesis are the projections of the central circle of the soccer field. Most of these ellipses are only partial (due to the camera angle) and also slightly-oblique (due to depth), as shown in Figure 1.2. Hence, detecting them is harder than detecting normal ellipses.

Figure 1.2 Three typical partial ellipses in broadcast soccer video. The partiality of the ellipses in (a)-(b) and (c) are caused by occlusion and camera view respectively.

In this thesis, we use ellipse detection in solving two problems: first, in the problem of ball detection and tracking where we use the detected ellipse

to estimate ball size, and second, in the problem of generating the midfield scene where we need a highly accurate ellipse detection algorithm.

Our algorithm for ellipse detection is based on the ellipse Hough transform. However, to make it more robust, we generalize the definition of *measure function* to handle partial ellipses. The partial ellipses appear in images when the original ellipses are partial, part of the ellipses are occluded, and/or the camera only covers part of the original ellipses. We also design a new algorithm to compute generalized Hough transform for robust ellipse detection. The algorithm is accumulator-free (uses less memory space) and our experimental results confirm that it is more robust in handling small and/or partial ellipses. Our new robust ellipse detection algorithm is general and can be used for any general ellipse detection applications. It can also integrate with the existing “fast Hough techniques” to form even better algorithms.

1.3 Contributions

The contributions of this thesis are threefold. Its principal contribution applies a trajectory-based approach to locate the ball in broadcast sports video, which is presented in Chapter 3. Unlike the object-based approach, it does not evaluate whether a sole object is a ball. Instead, it evaluates whether a candidate trajectory is a ball trajectory. In this approach, there are two steps: in the first step, we reduce the rate of false negatives by extending the search to ball-like objects, thus getting a number of ball candidates. Then, in the second step, we use information of the *path trajectories* of these candidate objects *over a short sequence of frames* to obtain the ball. Thus, in this step, we recover

from the higher rate of false positives. Empirical studies show that the trajectory-based approach can significantly improve the accuracy in locating the ball in broadcast soccer video and broadcast tennis video.

The second contribution is two successful applications of ball detection and tracking in BSV. The first application is a new approach of event detection in BSV, which is based on the ball trajectory computed. This approach can detect events more accurately than the algorithms using only the low-level features. This approach not only improves play/break analysis and high-level semantic event detection, but also detects the basic actions and analyzes team ball possession, which may not be analyzed based only on the low-level feature. The second application is a video-generating and enrichment system. This system is better than the existing systems in several aspects. This system applies the results of ball detection and tracking to compute *apparent* ball velocity[2], ball direction, team ball possession, etc and these results, together with other results of video analysis, are converted into icons to enrich the video. In addition, it can render not only the goalmouth scene but also the midfield scene, which cannot be rendered by the existing systems.

In the course of this research on soccer video analysis, we discover that it is very important to have an *accurate* and *robust* ellipse detection algorithm. The ellipses we want to detect are the projections of the central circle of the soccer field. Most of these ellipses are only partial (due to the camera angle) and also slightly-oblique (due to depth). Hence, detecting them is harder than detecting the normal ellipses. Our algorithm for ellipse detection is based on the standard ellipse Hough transform. However, to make it more

[2] The apparent ball velocity means the ball velocity relative to the center of the frames, not the ball velocity in the real-world. However, the apparent ball velocity is close to the ball velocity in the real-world when the camera is still.

robust, we propose the unbiased measure function to fairly measure small and/or partial ellipses. *Measure function* is the concept that we propose to understand and classify the existing ellipse Hough transforms. In addition, measure function also unifies the mathematical expressions of the existing ellipse Hough transforms. We also design a new algorithm to compute generalized Hough transform for robust ellipse detection. The algorithm is accumulator-free (uses less memory space) and our experimental results confirm that it is more robust in handling small and/or partial ellipses.

1.4 Thesis Structure

The remaining chapters of this thesis are organized as follows. Chapter 2 first states the problem of ball detection and tracking, the first problem that this thesis addresses. Then it briefly surveys the related work in general object detection and tracking, and ball (*soccer ball and tennis ball*) detection and tracking. Chapter 3 presents the trajectory-based ball detection and tracking algorithms for locating the ball in broadcast soccer/tennis video. Chapter 4 presents two applications of ball detection and tracking: (1) applying the ball locations in detecting the ball-related events in broadcast soccer video, and (2) applying the information derived from ball locations and the results of detecting ball-related events to enrich the reconstructed soccer video. Chapter 5 first surveys the related work in ellipse detection and understands the existing ellipse Hough transforms in the view of measure function. Then it presents an accumulator-free and robust ellipse algorithm and applies it to detect the ellipses in broadcast soccer video. Chapter 6 summarizes the thesis and indicates some future work.

Chapter 2

Ball Detection and Tracking in Sports Video

This chapter describes the problem of ball detection and tracking in broadcast sports video, its importance as a key component of a good event detection algorithm, and the key challenges. Then, it presents a survey of techniques for object detection and tracking, focusing in-depth on techniques in *ball* detection and tracking.

2.1 Problem of Ball Detection and Tracking

The problem of *detecting and tracking the ball* in broadcast sports video (BV) is, simply stated, the problem of locating the ball in each frame of the given broadcast sport video in which the ball is visible. In most broadcast sports video, the image frames can be classified into "*close-up"*, "*middle-view*" and "*far-view*". Ball detection for close-up frames can be done with high accuracy with many existing methods. Thus, the main focus of many research works is on the challenging problem of detecting and tracking the balls in the middle-views and far-views frames.

This ball detection and tracking problem have been studied by many researchers and many algorithms have been developed, for many different

types of videos and different kinds of sports. Generally speaking, the problem of ball detection and tracking is easiest for “fixed-camera” video where the video recorded using a fixed camera while it is most difficult in the case of broadcast video.

2.2 Motivation of Detecting and Tracking the Ball in BSV

The motivation for seeking accurate ball detection and tracking in broadcast soccer videos (BSV) has been mentioned in Section 1.2. To re-iterate, in soccer games, the ball is the focus of the players and is the object that players want to control. Hence, it is very natural that many events relate with the ball location and motion. Thus, the fact that information derived from the ball location in frames can greatly facilitate the event detection, which has been widely reported in the literature [Miy2003, ToQi2001, YLLT2003, YXLT2003]. For example, the ball locations over frames will greatly facilitate the analysis of broadcast soccer video. They could play a crucial role for analyzing the team ball possession, dividing video into play and break segments, evaluating the team tactics, and detecting semantic events.

2.3 Challenges of Locating the Ball in BSV

As it has been mentioned above, detecting and tracking the ball in the broadcast sports video (BV) is a difficult problem. For example, detecting and tracking the ball in BSV is difficult due to the following challenges [SCKH1997, YXLT2003, ABCB2003(a-c)]:

- The appearance of the ball varies irregularly over frames. Its size, shape, color, and speed all change irregularly over frames.
- Many objects are similar in appearance to the ball. For example, many regions of player and the penalty marks look like the ball.
- The ball is very small.
- The ball is often occluded by players.
- The ball is often merged with lines and players.

Here we show more ball and non-ball objects, which extends the illustration in Figure 1.1 that shows the ball and non-ball objects from the same frame. Typical balls in BSV (which are obtained by removing the other objects in the selected frames) are shown in Figure 2.1; typical non-ball objects but look like the ball are shown in Figure 2.2. These typical balls and non-ball objects justify the above-listed challenges. These challenges lead to a fundamental difficulty, which is there is *no ball representation* available to distinguish the ball from other objects *within a frame* as some non-ball objects look like the ball more than the ball itself.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2.1 Typical balls in broadcast soccer video. The ball in (a) is large from a middle view frame; the ball in (b) is small from a far-view frame; the balls in (c) to (g) are flying-balls; the ball in (h) is a ball separated from a line.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2.2 Typical ball-like objects in broadcast soccer video. The objects in (a) and (b) are penalty marks; the objects in (c) and (d) are soccer boots; the objects in (e) and (f) are white particles in the field; the objects in (g) and (h) are legs with white socks.

2.4 Related Work in Ball Detection and Tracking

The ball detection and tracking problem is a special case of the general object detection and tracking problem so we first give the survey in general object detection and tracking. Then we give the survey in ball detection and tracking.

2.4.1 Previous Work on General Object Detection and Tracking

There have been many object detection and tracking algorithms proposed during the past three decades because they have a wide spectrum of applications in many areas such as image/video processing and computer vision. These algorithms can be classified into four categories: (a) feature-based, (b) model-based, (c) motion-based, and (d) data association.

(a) Feature-Based: In feature-based algorithms, some features of object are used to discriminate targets from other objects within a frame. A category of approaches takes into account a reference image of the background. All objects in the difference frame between the current frame and the background frame are targets [KaBG1990, NCRT1998]. To discriminate the target from other objects, features are used to characterize targets in the property state space. For example, parameterized shapes [DACN2002, DGLD2004], color distributions and texture [EkTe2003a, GeSm1999], shape and color together [RaHa1998], are often employed in target representations. Features and labeled targets also can be used to train a neural classifier, and then the trained neural classifier is used to differentiate the targets from other objects [DGLD2004].

(b) Model-Based: Model-based algorithms, including anti-model algorithms, use not only features but also high-level semantic representation and domain knowledge to discriminate targets from other objects [KeOG2001, KoDN1993, NgWB2001, OhMS1999, ZhNe2001].

Algorithms in both the above two categories (feature-based and model-based) locate the targets frame by frame as locating the targets is performed within a frame using measures provided by properties of the targets. These methods could be called object-based because their crucial step is to decide whether a detected object is a target. In these methods there are three main elements: target representation, property extraction, and object discrimination. Generally speaking, a more parameter target representation would incur better chance of successful target detection and tracking. However, the high dimensionality of target's state space also makes estimating values in the representation to be a formidable problem. Hence, the principle of building a target representation is to make it feasible to discriminate the target from other objects and to make it easy to extract the properties used in the representation. Thus, target representation can include appearance features and models to solve the different problems. The representation has to be built up in *initialization*, and then to be updated over frames. These object-based methods implicitly assume that targets are somehow different from other objects within a frame. The intention of these methods is to decide whether a detected object is one of the targets in each frame. A detection and/or tracking problem is called an object distinguishable problem if the targets have some invariant differences from the other objects within a frame. For the tracking procedure of object-based methods, in [WuHu2001] Wu and Huang

commented: “Visual tracking target could be treated as a parameter estimation problem of target representation based on the observations in image sequences.”

(c) Motion-Based: Motion-based algorithms rely on the methods for extracting and interpreting the motion consistencies over frames (or time) to segment the moving object [BoFr1993, Low1992]. They claim a target is identified when a candidate has accumulated enough confidence to be a target. They are online algorithms so they cannot wait to evaluate trajectories until all the trajectories of a segment of video are formed.

(d) Data Association: Data association algorithms are designed to solve the data association problem, which is a problem of finding the correct correspondence between the measurements for the objects and the known tracks [BoMe2003, CoHi1996, Cox1993, DaHD2003, LeSF2003, RaHa2001]. There are four basic techniques for data association problem: Nearest Neighbor, Track Operation, Joint Probabilistic Data Association, and Multiple Hypotheses Tracking, which are explained further as follows:

- Nearest Neighbor: It assigns the measurement to the nearest track, where the distance between measurement and track normally is measured in the Mahalanobis distance [Cox1993]. It is computationally efficient but unreliable for tracking targets in a highly cluttered environment.
- Track Operation: The existing track operations include track splitting, track merging, and track pruning [Cox1993, SmBu1975, ZhFa1992]. Track-splitting, which was originally proposed by Smith and Buechler [SmBu1975], forks the track into two or more when two candidates (*measurements*) are found inside the validation area, rather than arbitrarily

assigning the closest candidate to the track. Assignment decisions are postponed until additional candidates have been gathered to support or refute earlier assignments. The tracks are restricted to a tractable number by merging similar tracks and pruning unlike tracks.

- Joint Probabilistic Data Association: It enforces a kind of exclusion principle that prevents two or more trackers from latching into the same target by calculating target-measurement association probabilities jointly [BaFo1988, Cox1992, RaHa2001].
- Multiple Hypotheses Tracking: The multiple-hypothesis filter was originally developed by Reid [Rei1979]. Cox and Leonard [CoLe1991] have demonstrated its utility in the context of building and maintaining a map of a mobile robot's environment. However, because it is a multiple scan method both its memory and computation requirement increase exponentially with problem size [Cox1993, ACSS2003]. Some efficient algorithms of multiple-hypothesis were developed to reduce the memory and computation requirement [CoHi1996, CoHi1994].

The algorithms in data association focus on the trajectory generation and management. Most of the techniques of trajectory management, for example trajectory forking, merging, and comparison, will be used in our ball detection and tracking algorithms that will be presented in Chapter 3.

2.4.2 Previous Work on Ball Detection and Tracking

In contrast to general object detection and tracking, there have been many algorithms specially designed for locating the soccer ball and the tennis ball,

which were developed for four kinds of sports videos: (a) fixed-camera video (FCV), which is recorded by fixed camera, (b) real soccer video (RSV), which is recorded by researcher's own camera, (c) broadcast soccer video (BSV), and (d) broadcast tennis video (BTV). Tracking the ball in FCV is relatively easier and successful tracking algorithms were reported. Since RSV is recorded by researcher's own camera, compared with the cameramen recording BSV, cameramen recording the RSV have more freedom of controlling the camera. Hence, they can choose a beneficial place and angle to produce the video in the good quality. Thus, locating the ball in RSV is relatively easier than in BSV. The algorithms for locating the tennis ball in BTV face different challenges from the ones for locating the soccer ball in BSV. The algorithms for locating the ball in four kinds of videos are reviewed separately as follows.

(a) Fixed-Camera Video

Pingali et al [PiJC1998] developed a real-time algorithm to track the ball and players in tennis video recorded by fixed camera. They used four fixed cameras placed in a stadium during an international tennis tournament---each camera covering one half of the court. Their ball tracking algorithm was tested on test sequences in which the players hit tennis balls with tennis racquets. Ball tracking results on these sequences are very encouraging.

Ohno et al [OhMS1999, OhMS2000] developed an algorithm to track the ball and players and to estimate the 3D position of the ball in soccer video recorded by fixed camera. They used 8 fixed cameras to cover the whole

soccer field. Their algorithm can reliably track the ball and players even if the ball and players are occluded temporarily.

Haas et al [HMSP2002] developed an algorithm to decide whether there is a goal in the real-life game for soccer game, which is a kind of computer referee system. They used two fixed cameras to monitor either goalmouth. A search engine performs the ball detection and tracking in each image taken by the camera. Then another procedure computes the world coordinate of the ball from the two images taken at the same time for the same goalmouth. Once the world coordinate of the ball is known, the algorithm can decide whether there is a goal.

Comparing with the video taken by non-fixed camera, FCV has multiple benefits. First, the background image can be accurately obtained. Second, the ball size can be exactly known. Third, the motion in the video taken by fixed camera exactly refers to the physical motion, no still objects will be considered as moving objects. Last, the fixed camera is used only for the game analysis so it can use high definition camera without considering long distance data delivery. The images taken by high definition camera will be much better than the one taken by normal camera and there is no need to use the interpolated images because the high definition camera can take high resolution images in 60 frames per second, which is four times the frame rate of the current broadcast sports video.

(b) Real Soccer Video

D'Orazio et al [DACN2002] proposed a ball recognition algorithm that works on the real soccer image sequences, which were recorded by their own

cameras, with variable light conditions and non-controlled backgrounds (meaning that the camera is not fixed). Their algorithm modified the circle Hough transform (CHT) by considering the self shadow. Thus, their algorithm can detect the ball even with the self shadow caused by the various lighting conditions. However, their algorithm did not consider the case that the ball is deformed into a non-semi circle object.

Leo et al [LeDD2003] studied the automatic ball recognition from the real soccer images. They found that the ball recognition performances applying Wavelet and the *independent component analysis* (ICA) preprocessing techniques are quite the same and that combing the ICA and Wavelet the percentage of pattern recognition can be increased.

D'Orazio et al [DGLD2004] improved the algorithm proposed in [DACN2002] by adding a neural classifier. The improved algorithm consists of two techniques (used together) in order to take advantages of the peculiarity of each of them: a fast circle detection (and/or circle portion) algorithm is applied on the whole image to find the area that is the best candidate to contain the ball considering only edge information; a neural classifier is used on the selected area to validate the ball hypothesis evaluating all the information contained inside the area. The improved algorithm achieved a high percentage of correctness. However, the improved algorithm still does not consider the case where the ball merges with the other objects. The algorithm will fail to identify the ball when the ball merges with an object that has the same color with the ball because their algorithm does not have a procedure to separate the ball from the merged objects. In BSV, there are many instances where balls are merged with other objects. More importantly,

this algorithm will produce a false positive when a non-ball object looks like a circle shape more than the actual ball does. Unfortunately, this often happens in BSV.

(c) Broadcast Soccer Video

Gong et al [GSCZ1995] proposed the first algorithm for identifying the ball from broadcast soccer video (BSV). This algorithm is easy to implement because it used color and shape features without complex representation and reasoning. The algorithm was successful in identifying the ball for the frames that it cares. However, it may have difficulty in identifying the ball in complex frames.

Yow et al [YYYL1995] proposed an algorithm to detect and track the ball in BSV. The detection was an intra-frame approach and is done in the reference frames selected at regular intervals. In a frame, it used a template-based approach to identify the ball. To further reduce the search space, the algorithm produced the difference frame between two frames after the camera motion was compensated. Template matching was performed on these pixels which indicate possible object motions. Between the reference frames, tracking of soccer ball was carried out. The position of the ball in the current frame was used as the starting point for local search of the ball in the next frame. To compensate for zooming action of the camera, the ball in the current frame was first scaled accordingly and then used as a template in the next frame. This is the first paper that tried to regain the benefits possessed by FCV through motion compensation for BSV.

Seo et al [SCKH1997] proposed a ball tracking algorithm to track the ball in BSV by using backprojection to reason the occlusion and a Kalman filter-based template matching procedure to track the ball. This is the first paper that considers the ball occlusion issue in BSV. However, the starting positions of the ball were manually initialized as the ball detection procedure was not available in their paper.

Yamada et al [YaSM2002] proposed an algorithm to detect and track the ball in BSV. The detection was a simple procedure because it did not really identify the ball. It extracted the white regions excluding the player regions and the line regions. Then each white region was considered as a ball candidate. The tracking procedure tracks each ball candidate by searching a neighboring area of the predicted locus of the candidate. If multiple candidates were found, only the one that is the nearest to the predicted position is retained. If no candidate was found, tracking of the candidate was terminated. These steps were repeated until all except one candidate are deleted; the remaining region was determined as the ball region. This procedure effectively uses the length of the trajectories to select the ball trajectory.

Choi et al [ChSL2004] proposed a particle-filter-based algorithm to track the ball in BSV. In their algorithm, the ball is considered as an ellipse with four parameters, the center position and the lengths of the major and minor axes, i.e. the paper assumed that the ellipse is horizontal. The particle filter uses the contour and the histogram of the objects to differentiate the ball from the other objects. This work has a solid base because it used well-

studied techniques. However, the tracking will fail when several consecutive balls are deformed seriously.

The above-mentioned five algorithms have their own advantages respectively. After learning from these algorithms, we proposed a trajectory-based algorithm to detect and track the ball in BSV in the succeeding chapter. Some primary results related to this algorithm are published in our papers [YuTW2003, YXTL2003, YXLT2003, YXTY2003, YSWC2004].

(d) Broadcast Tennis Video

It is based on two reasons that this chapter also discusses the ball detection and tracking algorithms for broadcast tennis video (BTV). First, although the problem of locating the ball in BTV shares some similarities with the one of locating the ball in BSV, they also face different challenges. Second, this thesis will present a trajectory-based tennis-ball detection and tracking algorithm in Section 2.7. This algorithm shows that trajectory-based approach can be applied in different sports games to locate the ball.

Miyamori et al [MiIi2000] and Miyamori et al [Miy2002] proposed a ball detection and tracking algorithm based on template matching in BTV. The ball locations produced by their algorithm greatly helped their main objective---the behavior analysis of a player. However, template matching has problems in identifying the ball when occluded or merged with other objects. In addition, their algorithm only focuses on identifying the ball locations around a player but not all ball locations.

Miyamori [Miy2003] improved the algorithm in [MiIi2000, Miy2002] by adding a competition method to identify the ball trajectory. This technique

works as follows. Detect all candidate locations of the ball that are outside the player's rectangle but within a certain distance from the player's rectangle and that are smaller than a certain area. Repeat this for several successive frames and keep track of the candidates that move spreading out from the player's center position. Repeat this step until it becomes the final single candidate for the ball trajectory. This technique definitely can improve the accuracy of identifying the ball. This algorithm is similar to the algorithm presented in [YaSM2002] for identifying the ball in BSV. However, this improved algorithm still focuses on identifying the ball locations around a player. In addition, it does not deal with occlusion and merging issues.

2.4.3 Other Work Related to the Ball Location

Here we discuss one paper that has relation to the ball location, which is on 3D position of the ball. Shum and Komura [ShKo2004] has done similar works for baseball video.

Kim et al [KiSH1998] proposed an algorithm to compute the 3D position of the ball from monocular image sequence of soccer game. In the algorithm, they adopted ground-model-to-image transform together with physics-based approach that a ball follows the parabolic trajectory in the air. The ball heights are calculated based on the given start and end positions of the ball on the ground using simple triangular geometric relations.

2.5 Summary

This chapter first states the problem of ball detection and tracking. Then it

gives surveys in general object detection and tracking and ball detection and tracking. The algorithms for general object detection and tracking have developed many effective techniques such as the trajectory-based technique. The success of locating the soccer ball in fixed-camera video was reported, but the algorithms may not be directly applied to locate the ball in BSV. The existing algorithms that locate the ball in RSV and BSV have succeeded in solving the problems in the concerning cases, however we can still use trajectory-based techniques to develop a more effective algorithm.

Chapter 3

A Trajectory-Based Ball Detection and Tracking Algorithm

This chapter presents a trajectory-based algorithm for detecting and tracking the ball in broadcast soccer videos, which is capable of obtaining very accurate results in locating the ball. Encouraged by the success of this algorithm, this chapter also applies trajectory-based approach to locate the ball in broadcast tennis videos.

This thesis first focuses on the ball detection and tracking problem in broadcast soccer video (BSV). Then it applies the formed principles and methods to develop an algorithm for detecting and tracking the ball in broadcast tennis video (BTV).

3.1 Overview of the Algorithm

The proposed algorithm aims to detect and track the ball in BSV, whose block diagram is shown in Figure 3.1. This algorithm tackles the challenges (*they are described in Chapter 2*) of locating the ball in BSV with four steps, which belong to two large components: Ball candidate production (including step 1 and 2) and candidate trajectory processing component (including step 3 and 4). In the first component, we use the anti-model approach, which was first

proposed to identify the faces [KeOG2001], to produce the ball candidates for each frame. We built a set of sieves (*anti-models*) to identify and remove non-ball objects and consider the remaining objects to be the ball candidates, which look like the ball in appearance. In the first component, we target to reduce the rate of false negatives (at the price of a temporarily higher rate of false positives) by extending the search to ball-like objects, thus getting several candidates (normally about 5) for each frame.

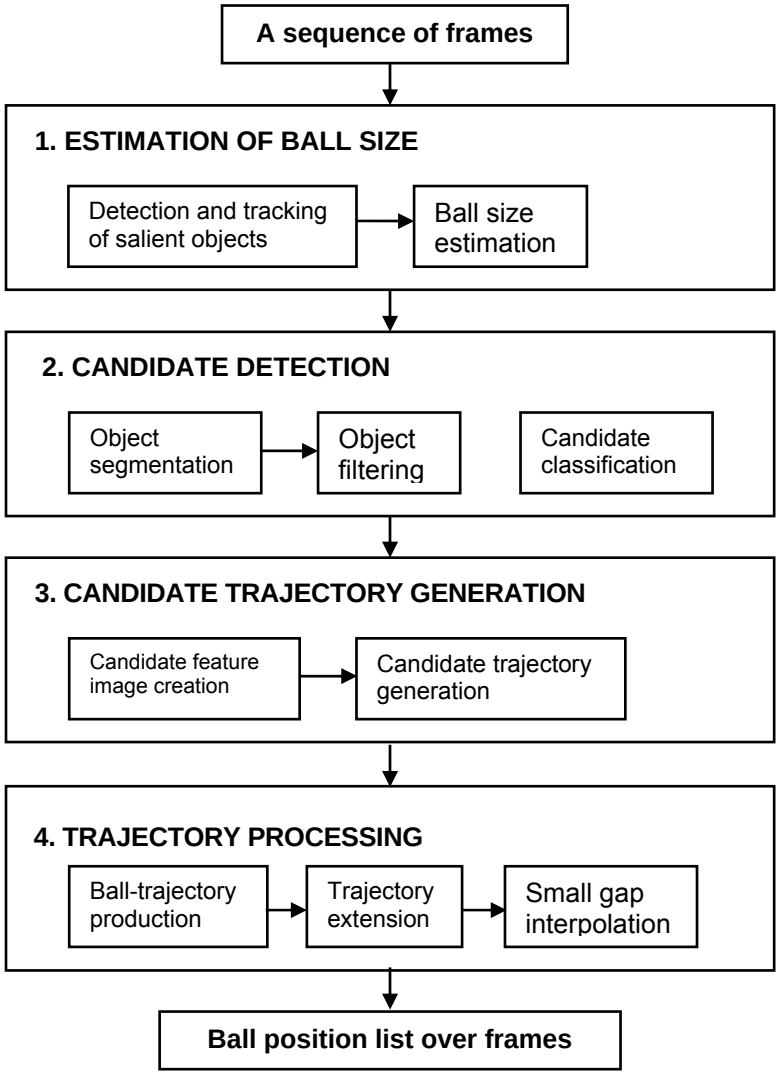


Figure 3.1 Block diagram of the trajectory-based algorithm for detecting and tracking the ball location in broadcast soccer video.

In the second component, we use a trajectory-based approach to identify the ball trajectory. Our algorithm integrates trajectory operation techniques such as *trajectory-splitting, trajectory-merging, joint probabilistic measure for objects and trajectories*, which originally were proposed in the context of multiple-target tracking. To make trajectory processing fast, we introduce candidate feature image (CFI) and process CFIs to reduce the number of candidates for a sequence of frames. Hence, trajectory processing is speeded up. To enhance the reliability of ball selection, we introduce a competitive procedure to select the ball trajectories after we have evaluated all candidate trajectories for a whole sequence of frames. After we obtain ball trajectory, we recover from the higher rate of false positives by throwing the non-ball trajectories.

In other words, our algorithm uses two key ideas. (1) Since identifying the ball within a frame is difficult, we relax the condition and focus on the task of producing a set of ball candidates. (2) Since identifying the ball trajectory is much more reliable than identifying the ball in a frame, we use a trajectory-based procedure to produce the ball trajectory.

In the preprocessing step of the algorithm, a statistical procedure is used to find the field color range for the whole video. A frame is considered to contain a portion of the soccer field if and only if its dominant color falls in the field color range of the video. Thus, the video sequences of the frames showing the soccer field are produced. In this chapter, we will focus on how to find the ball location in each frame of a given sequence of the contiguous frames depicting the soccer field.

The rest of this chapter is organized as follows. Section 3.2 presents the methods of estimating the ball size. Section 3.3 explains the method to generate the ball candidates for each frame. Section 3.4 presents the method to generate the candidate trajectories. Section 3.5 explains the measures and procedures to process the candidate trajectories to produce the ball trajectories. Various experiments and their results are given in Section 3.6. Section 3.7 presents the trajectory-based ball detection and tracking algorithm for broadcast tennis video. We summarize this chapter in Section 3.8.

3.2 Ball Size Estimation

Removing non-ball objects by size necessitates ball size estimation. In broadcast soccer video (BSV), the ball size changes due to camera calibration and ball deformation, apart from the distance change from ball to camera. However, we can estimate the ball size through the sizes of salient non-ball objects in the same frame. In a soccer field, the ellipse, goalmouth, and players (the referee, two goalkeepers and players) are salient so that they can be detected more reliably than the ball. Thus, these objects are first identified in each frame and their heights and locations are used to estimate the size of the ball. Since the ball size varies with locations within a frame, we compute the ball size for each location to build a complete size sieve.

3.2.1 Principle of Ball Size Estimation

We begin with the principle used in ball size estimation. For BSV, the camera can be approximately modeled by a pinhole camera, whose principle of image

generation is illustrated in Figure 3.2. Therefore, we have the following approximated formula.

$$h = H \times d \times D^{-1} \tag{3.1}$$

where *H* and *h* are the heights of the object and its image respectively, *D* and d are the distances from the object and its image to the pinhole respectively.

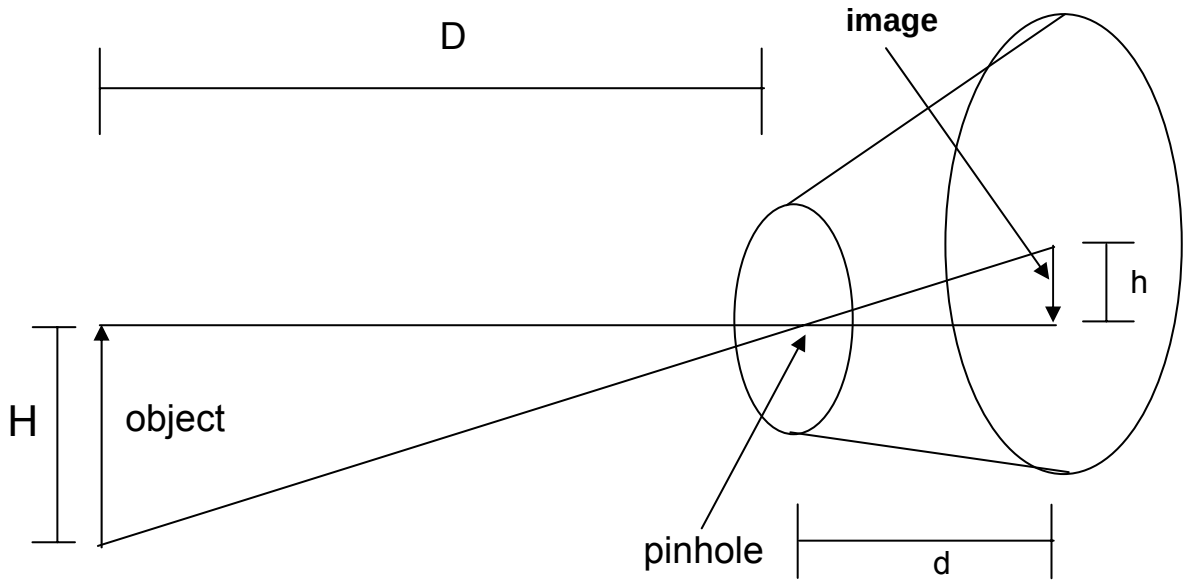


Figure 3.2 Illustration of a pinhole camera.

Let H_0 be the diameter of the physical ball, (i_c, j_c) be the center of the frame and $b(i, j)$ the height of the ball at the location (i, j) in the frame (*i* and *j* are the row and column of the ball respectively). Let $b(i_c, j_c)$ be the ball size if the ball were at (i_c, j_c). Then according to the equation 3.1, we have

$$b(i,j) = \frac{H_0 \times d(i,j)}{D(i,j)} = \left(\frac{d(i,j) \times D(i_c, j_c)}{d(i_c, j_c) \times D(i,j)} \right) \times b(i_c, j_c) \tag{3.2}$$

where $D(i, j)$ and $d(i, j)$ are the distances from the physical ball and the image of the ball to the pinhole respectively, provided that the ball image is at row *i*

and column j in the frame $D(i_c, j_c)$ and $d(i_c, j_c)$ are the values of $D(\mathrm{i},\mathrm{j})$ and $d(i, j)$ respectively if the ball were at (i_c, j_c), the center of the frame. Thus, we have the following ball height variation array.

$$A = (a_{ij})_{h\times w},\ \ a_{ij} = \frac{d(i, j)\times D(i_c, j_c)}{d(i_c, j_c)\times D(i, j)} \tag{3.3}$$

where w and h are the width and height of the frame.

The above formulae (3.2) and (3.3) are approximate because they consider only the main camera parameters affecting the size of the ball image. Nevertheless, they are accurate enough for estimating the ball size. To be precise, each frame has its own variation array. But in soccer video, we could approximately consider that each frame (*far-view or middle-view)* has a height variation array. Thus, we find salient objects, decide the frame type, and further estimate the ball size. We use $s(i, j) = [b(i, j)(1-\Delta_1), b(i, j)(1+\Delta_2)]$, the ball size range, to replace $b(i, j)$, the ball size, in our algorithm, where Δ_1 and Δ_2 are the selected extension to tolerate the estimation and segmentation errors. Thus, $S = (s(i, j))_{h\times w}$ forms a complete ball size sieve.

According to the above discussion, the main task of ball size estimation is to detect the ellipse, the goalmouth, and the players, which are described below one after another.

3.2.2 Salient Object Detection

To estimate the ball size, we need to detect the ellipse, the goalmouth, and the players. For the challenging problem of detecting the ellipse in BSV, this thesis develops a robust ellipse Hough transform that can be applied to detect

the ellipses in BSV well. The robust ellipse Hough transform is presented in Chapter 5 separately because it needs a longer presentation. Here, we only describe how to detect the goalmouth and the players in the soccer field.

Goalmouth Detection: We will try to find the goalmouth if we cannot find the ellipse in the given frame. For the goalmouth, we have observed: (a) the two posts are almost vertical; (b) the goal posts and goal bar are bold line segments; and (c) compared with the middle line and side lines, the goal posts are short line segments. Based on these observations, we first find the two parallel vertical short line segments. For the given frame, we segment pixels with line color and all other pixels are painted in the field color. For each pixel (i, j), we define $\Psi(i, j)$ as follows.

$$\Psi(i, j)=\begin{cases}1, & \text{pixel } (i, j) \text{ is of the line color,}\\ 0, & \text{otherwise.}\end{cases} \tag{3.4}$$

In the segmented image, many pixels do not belong to the vertical lines. According to our statistics, the two posts are almost vertical (the angle between the central line and the vertical line is less than 3°). With this knowledge, we propose a filter $\Re(\bullet)$ to identify the pixels in the long vertical line and to remove the pixels not belonging to the long almost vertical line. $\Re(\bullet)$ paints the pixel (i, j) in the field color if $\Phi(i, j) = 0$.

$$\Phi(i,j)=\begin{cases}1, & \text{if } \left(\sum_{k=0}^{L}\Psi(i, j-k)\right)/L > \text{threshold},\\ 1, & \text{if } \left(\sum_{k=0}^{L}\Psi(i, j+k)\right)/L > \text{threshold},\\ 0, & \text{otherwise.}\end{cases} \tag{3.5}$$

$\Re(\bullet)$ can remove most of the pixels that are not on the two vertical posts in the segmented image, as illustrated in Figure 3.3(c).

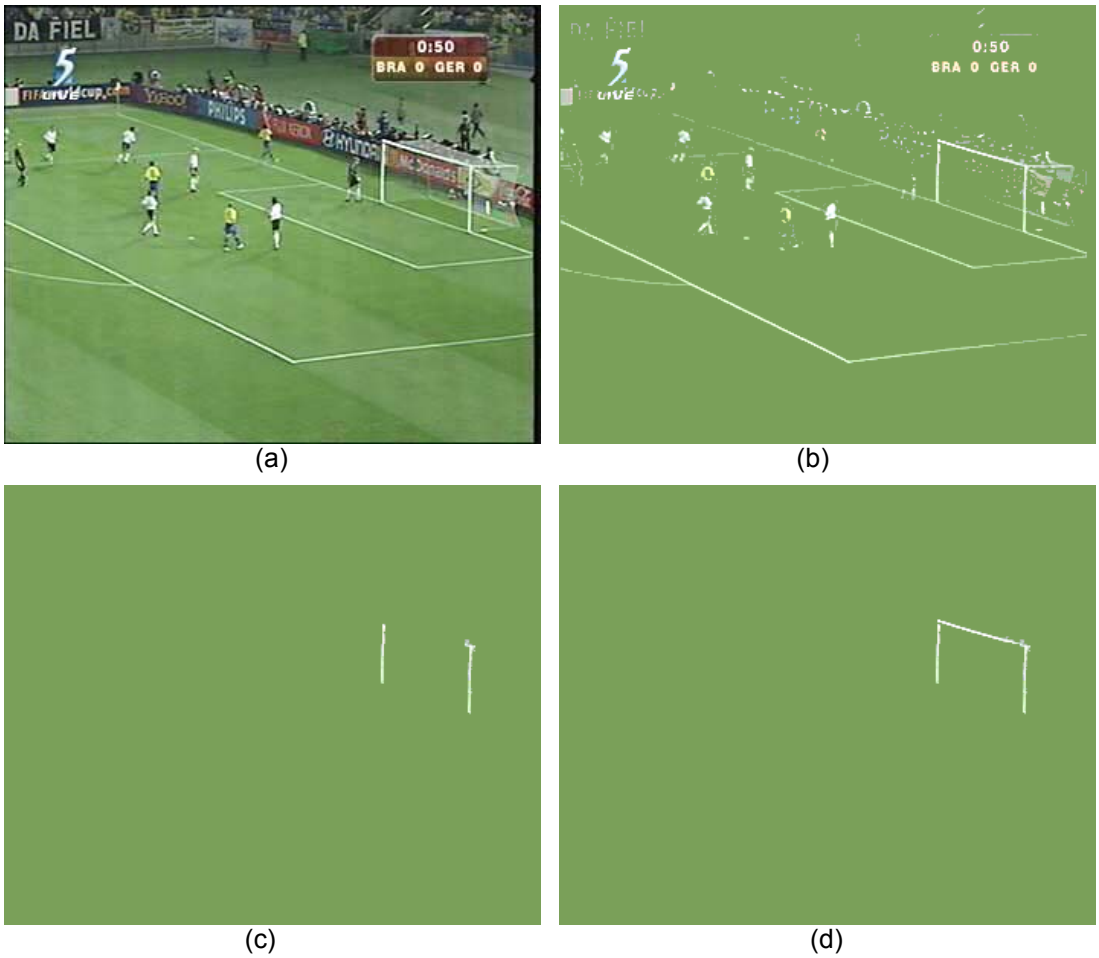


Figure 3.3 Goalmouth detection. (a) the input image, (b) the segmented image by color, (c) the found two goalposts, (d) the found goalmouth.

Once we identified the two vertical posts, we will check whether there is a goal-bar connecting the two upper ends of the two vertical line segments. Then, we further find the slope of the bar and adjust the two ends of the two poles. So far, we have estimated all the parameters of the goalmouth. Let L_1, L_2, and L_3 be the estimated three line segments that comprise the goalmouth. Let L_4 be the estimated goal line. Let $G=\{L_1, L_2, L_3, L_4\}$. We define $M_G(G)$ to evaluate the presence of the goalmouth.

$$M_G(G)=\frac{1}{4}\left(\sum_1^4 M_L(L_i)\right) \qquad (3.6)$$

Sometimes, we can only detect one of the two parallel posts since the other is blocked by players. In this case, we will estimate the location of the bar according to the location of the bar in the previous frame. Then, we confirm the presence of the goalmouth by verifying the presence of the bar. Figure 3.3 illustrates the procedure to detect the goalmouth for a frame.

Players Detection: We use a seed-growing procedure to find the objects (the connected components) that are in the field portion of the frame shown in Figure 3.4. Then, we use feature $f_1, f_2, \cdots, f_k$ to evaluate whether an object is a player. Assume that $\mathrm{K}(F)$ be the set of all detected objects in the field portion of frame F. Let $o \in \mathrm{K}(F)$ be an object and $P(o \mid f_i)$ $(i = 1, 2, \cdots, k)$ be the probability that indicates how likely o is a person with respect to the feature f_i. The choice of features allows us to assume that they are independent with relatively small error. With this assumption the probability that o is a person has a simple formula.

$$P(o) = \prod_1^k P(o \mid f_i) \qquad (3.7)$$

The object o is removed if $P(o)$ is small; otherwise o is kept for estimating the ball size. The features used to evaluate whether an object is a person are mainly the color, the size, and the surrounding. The surrounding means whether the object is surrounded by the pixels in the field color. The probability for color is the ratio of the number of pixels in player color to the area, in the upper half of the object. For shape, we use a predefined table of probability with respect to the ratio of height to width of object.

(a) (b) (c)

Figure 3.4 People detection. (a) the input image, (b) the segmented image, (c) the found people with their bounding boxes.

According to the sizes of the detected players, we can classify the type of the frame. For example, we know that the frame is a far-view frame when the sizes of persons are very small. For each detected person, we normalize its size as if it is at the center of the frame by the size variation array. Then, we find the average of the normalized sizes and compute the ball size according to the ratio of the players' heights to the diameter of the ball.

3.2.3 Ball Size Computation and Adjustment

After knowing the size and location of a found salient object, we can compute its normalized size, which is the size of its image projection provided that it is at the center of the frame according to equation 3.3. Then we can estimate ball sizes at the center of the frame according to equation 3.2. The estimated ball sizes form a function over frames. Among these ball sizes, those estimated by ellipse and goalmouth are very reliable so we do not change them. We adjust the ball size by smoothing the curve of the function of the ball size without changing the reliable sizes to improve the accuracy of the estimated ball sizes.

3.3 Ball Candidate Generation

This section presents a procedure to produce high qualitative ball candidates. The ball candidates comprise both the ball objects and the ball-like objects, which look like the ball but are not the ball objects, because we cannot identify the ball object among the candidates solely by appearance. This ball-candidate production procedure uses the anti-model approach, which removes identified non-ball objects in a frame using the built sieves (or anti-models) while the remaining objects are considered as the ball candidates of the frame. First, the frame can be separated into two objects: field and non-field. The non-field object is eliminated as a non-ball object as the ball is seldom out of the playing field. The field is obtained through a seed-growing procedure by continuously absorbing the neighbor pixels with the field color or with small color difference from the seed pixel.

3.3.1 Object Production

Here we describe the segmentation procedure to produce the objects in the soccer field portion of a given frame. The challenge of the procedure is to separate the ball from the object when they are merged. The importance in separating the merged ball is that the ball will be removed if it cannot be properly separated. The object production procedure has three steps. We first identify shapes such as ellipse, lines, etc. For the identified shapes, we find the function that represents the shapes. Then we segment the shapes according to the representative function. In this segmentation process, another important task is to prevent the ball that merged with the identified

shapes from being removed. The segmented shapes are recorded as the found objects. Then all connected components in the processed frame that the identified shapes are removed are considered to be objects. Next, from the processed frame we produced a segmented frame with the ball color. All connected components in segmented frame are considered to be objects too. From this segmented frame, we can properly produce the ball object even when the ball merges with other objects which are not in the ball color. Besides finding the ball in the field, we also want to find the ball in the area around the goalmouth as locating the ball in this area is very important for detecting goal and other events (this area is termed as the goalmouth area for short). We use a specialized procedure to find the ball objects in the goalmouth area when the goalmouth is detected. Let frame F be the current frame and F_p be its previous frame. Assume that we have detected the goalmouth in F and F_p. Then we transform F_p into F_t to make the goalmouth in F_t match with the goalmouth in F exactly. This transform includes shifting, rotation, and linear resizing. Next we obtain F_d that is the difference frame that frame F is subtracted by frame F_t. During the subtraction, for each pixel (i, j), the pixel of frame F is kept if F is in the ball color and F_t is in the field color at (i, j); otherwise, it is painted in the field color. The salient clusters of the points in non-field color are considered as the objects. Figure 3.5 illustrates the procedure to produce objects in the goalmouth area.

Let F be the considered frame, the set of all the objects in F is denoted as $O(F) = \{o: o \text{ is an object in frame } F\}$.

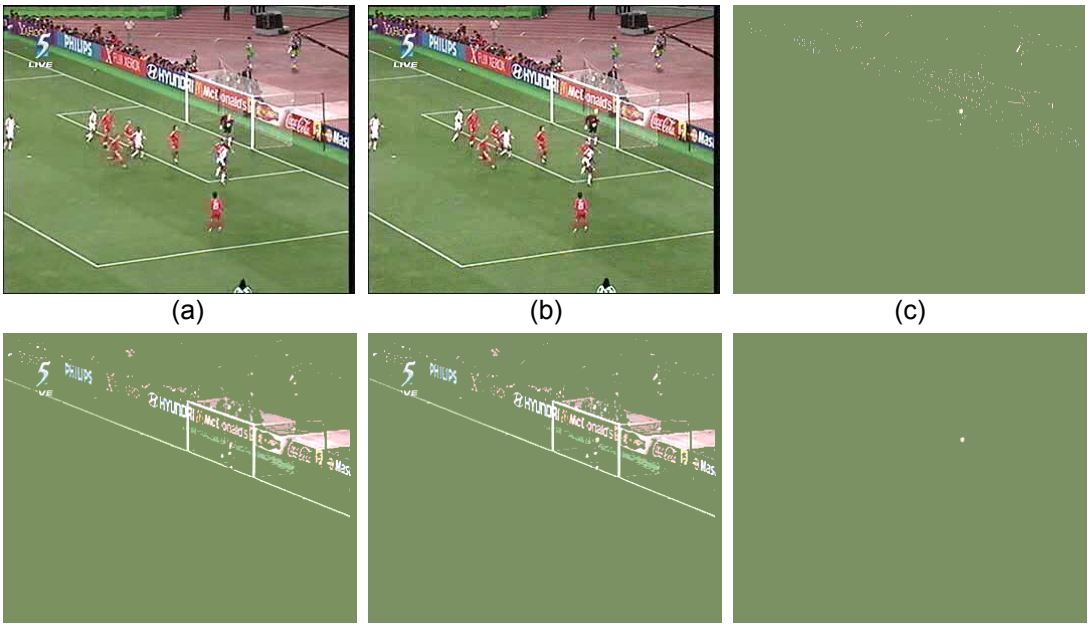


Figure 3.5 Object production in goalmouth area. (a) is the original previous frame, (b) is the original current frame, (d) and (e) is the goalmouth area of segmented frame of (a) and (b) respectively, (c) is the difference that (e) is subtracted by (d), and (f) is the found salient cluster.

3.3.2 Sieves and Candidate Generation

We employ the following six sieves to remove more non-ball objects.

Ball Size Sieve Θ_1**:** The ball size sieve is $S = (s(i, j))_{h \times w}$ as described in the preceding Section 3.2.

Line Sieve Θ_2**:** We remove all long lines, including straight lines and curves, as the ball cannot deform into a long line.

Ball Color Sieve Θ_3: The ball must have some pixels whose colors fall into the ball color range. Hence, we can filter objects with too few ball color pixels.

Shape Sieve Θ_4: The ball object can have a shape far from the circle, but in most frames the ratios of both width-to-height and height-to-width are less than 3 according to our statistical results.

Ball Center Sieve Θ_5 : A ball object may have some pixels whose colors do not fall into the ball color range, but the color of its center should fall into ball color range. Hence, we remove objects whose center colors are not in the ball color range.

Penalty Mark Sieve Θ_6 : When a frame shows the goalmouth or penalty box, it probably shows one of the two penalty marks in broadcast soccer video. Hence, for the frames that we had goalmouth or penalty box detected, we compute the world position of each candidate. As a result, we can find out whether a candidate is a penalty mark according to its world position. We remove the identified penalty mark from the obtained candidates.

Figure 3.6 Candidate generation. (a) original frame, (b) the frame after removing non-field and lines, (c) the frame after removing the objects by size, (d) the frame after removing objects by all the sieves.

Each sieve Θ_i is a Boolean function on the domain $\mathrm{O}(F) = \{o : o \text{ is an object in frame } F\}$.

$$\Theta_i(o) = \begin{cases} 0 & \text{if } \Theta_i \text{ removes it,} \\ 1 & \text{otherwise.} \end{cases} \quad (3.8)$$

After sieving, the remaining objects of $\mathrm{O}(F)$ form the ball candidate set $C(F)$ of frame F.

$$C(F) = \{o : \prod_{i=1}^{6} \Theta_i(o) = 1, o \in \mathrm{O}(F)\} \quad (3.9)$$

The sieved results of a sample frame are shown in Figure 3.6.

3.3.3 Candidate Classification

Let $f_1, f_2, \cdots, f_k$ be all features that we use to evaluate the candidates. Let $P(o \mid f_i)$ $(i = 1, 2, \cdots, k)$ be the probability that o is the ball according to the feature f_i. The choice of features allows us to assume that they are independent with relatively small error. Thus, this assumption is attractive because the probability that o is the ball has a simple formula.

$$P(o) = \prod_{i=1}^{k} P_i(o) \quad (3.10)$$

According to the probability $P(o)$, the candidates in $C(F)$ can be divided into three categories. Categories 1 to 3 contain the objects with high, medium and low probabilities respectively. Two types of features are used to compute the probability. One type is the appearance features such as the circularity, the average color distance to the ball color, and the difference to the estimated size. The other type is the isolation of the candidate, i.e. how far the candidate is from other objects. This feature is important because the

candidate close to a player might be his region due to over-segmentation. For candidate classification, we have a special rule that all candidates in the goalmouth area are assigned as Category 2.

The various conditional probabilities are defined as follows. The probability of the object with respect to the circularity is defined to be the ratio of the number of the pixels in the object to the area $\pi \times r^2$, where r is the radius of the object; the probability with respect to the color is defined as the ratio of the number of pixels in ball color range to the number of all the pixels in the object; the probability with respect to the radius is defined as $(R - |R - r|) : R$, where R and r are the estimated ball size and the radius of the object respectively. The probability with respect to isolation is a predefined table. Classification result of the candidates will be used to evaluate the candidate trajectory in the following trajectory mining procedure.

3.4 Candidate Trajectory Generation

Within a frame, it is hard to identify the ball among the candidates of a frame as the features of the ball are not different from the features of the other candidates. In the worst case, some non-ball candidates look like the ball more than the ball itself. For example, the shape of the ball is far from its ideal rounded shape, when it is half occluded or when it is flying very fast. Hence, we do not decide whether a sole object is a ball. Instead, we decide whether a candidate trajectory is a ball trajectory. Trajectories are generated by a Kalman filter-based procedure, which works properly and fast.

For each frame, we throw away the data that is irrelevant to ball position and retain only several candidates. These candidates have two merits. The first merit is that the candidates contain enough information for finding the ball. The second merit is that the data volume of candidates is very small so that we can process all candidates of a long sequence in one time. When candidates are processed together, rich spatial and temporal information can be obtained and used.

For a sequence of frames, each combination of candidate numeral features can form a candidate feature image (CFI). The samples of numeral features are the size, the locations, and the velocities of the candidates. The various CFIs express the candidates of a long sequence in different aspects. In a CFI, the temporal filters and the trajectory analysis are easily applied. This section uses the Kalman filter to generate the candidate trajectories in CFI. Each element of each candidate trajectory is a ball candidate. This is why it is called a candidate trajectory.

3.4.1 Candidate Feature Image

A candidate feature image is an image that draws a combination of candidate numeral features over the frames in the given sequence. X-image (Y-image, DISTANCE-image) is created in such a way that the width of the created image is the number of the frames in the sequence and the height of the created image is the width (height, the length of diagonal) of the frame in the soccer video. Each candidate draws a pixel in X-image (Y-image, DISTANCE-image) with the column being the frame serial number in the sequence and the row being the column (row, the distance between its center and the top

remaining corner) in the original frame. XY-image, a 3D image, is created in such a way that the length of the image is the number of the frames in the sequence, and the width and the height of the image are the width and the height of frames in soccer video. Each candidate draws a pixel (t, x, y) in XY-image with t being the frame serial number in the sequence and x (y) is the number of the column (row) of the candidate center in the original frame. Figure 3.7 shows a sample DISTANCE-image.

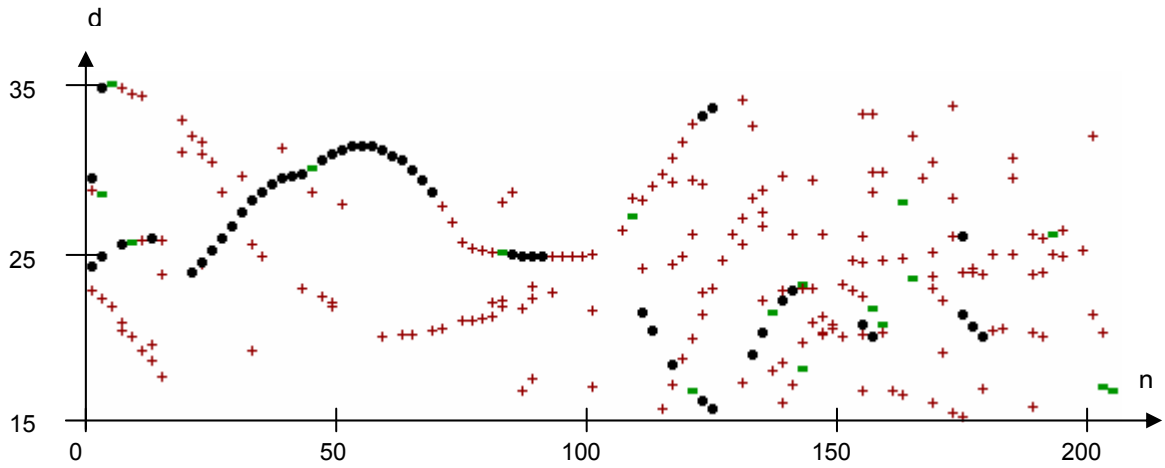


Figure 3.7 Partial DISTANCE-image of the obtained candidates for the sequence of the frames from 48957 to 49167 of FIFA 2002 final. Black dots, green rectangles and red crosses stand for candidates in category 1 to 3. In the figure, n is the serial number of the frame in the sequence and d is the distance between the candidate and the top left corner of the original frame. For the legibility, only the frames with even serial number are drawn.

3.4.2 Candidate Trajectory Generation

We use a candidate verification procedure based on the Kalman filter to find candidate trajectories from a candidate feature image. Figure 3.8 gives the flowchart of the procedure.

A trajectory seed is first obtained, which then initializes the Kalman filter. Then we grow the trajectory. The ball position in the next frame is estimated by the Kalman filter; the candidate that is close to the estimate is found; the ball position is the center of the found candidate if it is close

enough to the estimate. Notice that the trajectory will be forked into more trajectories when more candidates are close to the estimate. Theoretically, the number of forked trajectories can be the exponent of the number of candidates in the CFI. Fortunately, in our procedure the forking occurs sparsely. Finally, the filter is updated. A frame is called a candidate missing frame and the estimate will be considered the ball position if the frame has no candidate close to the estimate. The growing procedure terminates when the number of consecutive candidate missing frames reaches a given threshold. In our implementation, this threshold is 7. This procedure produces a candidate trajectory set for a candidate feature image.

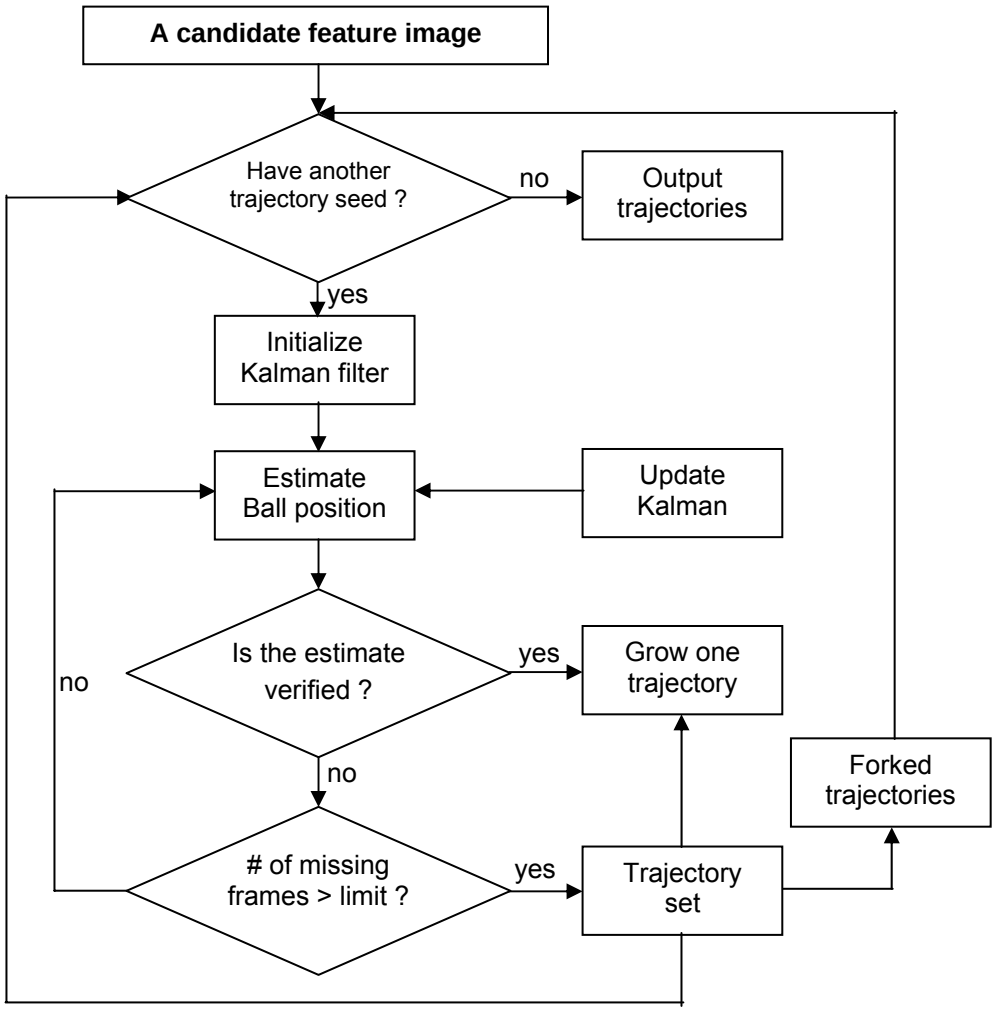


Figure 3.8 Flowchart of candidate trajectory generation.

In this chapter, Kalman filter is used in several places, but we only describe how to use the Kalman filter to generate candidate trajectory in Appendix A to avoid repetition. We use the Kalman filter implemented in OpenCV b2.1 from Intel^{TM}.

3.4.3 Trajectory Joint

Let T_1 and T_2 be two trajectories. Assume that the start of T_2 is just after the end of T_1 in time. Let O_1 be the last object of T_1 and O_2 be the first object of T_2. We join T_1 and T_2 into T if O_1 and O_2 are close in location and size. We extend the definition of a trajectory to include the join of two trajectories, in which case we still call T a trajectory.

3.5 Trajectory Processing

We define a confidence index to indicate the likelihood that a candidate trajectory is a ball trajectory. Then, we first remove the candidate trajectories with which the confidence index is too low. For the remaining candidate trajectories, we produce the ball trajectories using two procedures: ball trajectory production and trajectory extension. After trajectory extension, we use an interpolation procedure to patch the small gap between each pair of the obtained ball trajectories.

3.5.1 Confidence Index

Let $\Gamma = \{T : T$ *is a candidate trajectory*$\}$ be the trajectory set of a given sequence in its XY-image. Let $\lambda_1, \lambda_2, \cdots, \lambda_m$ be all properties of a trajectory T. A function $\Omega_i(\lambda_i)$ computes the confidence index that T is a ball trajectory with respect to λ_i. The confidence index that T is a ball trajectory $\Omega(T)$ is defined below:

$$\Omega(T) = \sum_{i=1}^{m} \Omega_i(\lambda_i) \qquad (3.11)$$

The confidence index $\Omega_i(\lambda_i)$ is computed by using three types of features.

Type I: The first type of features is the numbers of the candidates in Category 1 to 3 in the trajectory.

Type II: The second type of features is the ratio of the sum of the numbers of candidates in Category 1 to 2 to the number of candidates in Category 3.

Type III: The third type of features is the straightness of the trajectory and the straightness of the trajectory with respect to x or y.

Normally $\Omega_i(\lambda_i)$ is a positive number that shows how likely T is a ball trajectory with respect to λ_i. But $\Omega_i(\lambda_i)$ can be negative when λ_i indicates that T is not a ball trajectory. For example, let λ_i be the straightness of the projection of T in Y-image (Candidate feature Y-image). We set an infinite negative number to $\Omega_i(\lambda_i)$ if there is a significant length of the projection of T resembling a straight line in Y-image. The above rule is based on the fact that the straight line portion of T indicates that the candidate is a stationary object. Such a trajectory can be discarded even if it is a ball trajectory as the ball trajectory containing the stationary ball indicates that the game is stopped.

3.5.2 Overlapping Index

Roughly speaking, after we identified a ball trajectory we can remove all trajectories that overlap with the identified ball trajectory as there is only one ball in each frame. However, we cannot directly apply this rule as a ball trajectory might contain some non-ball objects. Hence, we define an overlapping index to indicate how much a trajectory overlaps with an identified ball trajectory. Then, we decide what to do according to the overlapping index. Let T_b be a ball trajectory and T be a candidate trajectory. The overlapping index T to T_b is denoted by $\Xi(T_b, T)$, which is computed as follows.

$$\Xi(T_b, T) = \frac{L(T, T_b)}{|T| \times \Omega(T)} \tag{3.12}$$

where $L(T, T_b)$ is the number of the frames that both T and T_b cover.

3.5.3 Ball Trajectory Production

Let Γ *be the set of all candidate trajectories in XY-image.*
SET *the ball trajectory set* B *to be empty.*
WHILE *(*Γ *is not empty)* ***DO***
Move the trajectory T *with the highest index into* B***.***
Discard the trajectories that seriously overlap with T *in* Γ*.*
Remove the overlapped portions of the trajectories that slightly overlap with T *in* Γ*.*

Figure 3.9 Ball trajectory selection procedure.

After we identify a ball trajectory T_b, we process the candidate trajectories that overlap with it as follows. Assume that a candidate trajectory T overlaps

with T_b . We discard T as non-ball trajectory when $\Xi(T_b, T)$ is large; we discard the overlapped portion of T when $\Xi(T_b, T)$ is small. With this preparation, we propose the procedure described in Figure 3.9 to produce the ball trajectories.

The whole procedure to produce the ball trajectories from the all candidate trajectories of a sequence of frames is termed as the trajectory mining procedure. This procedure consists of trajectory evaluation and ball trajectory selection. Figure 3.10 shows the result of the trajectory mining procedure on the sequence of frames from 48957 to 49167 of FIFA 2002 final.

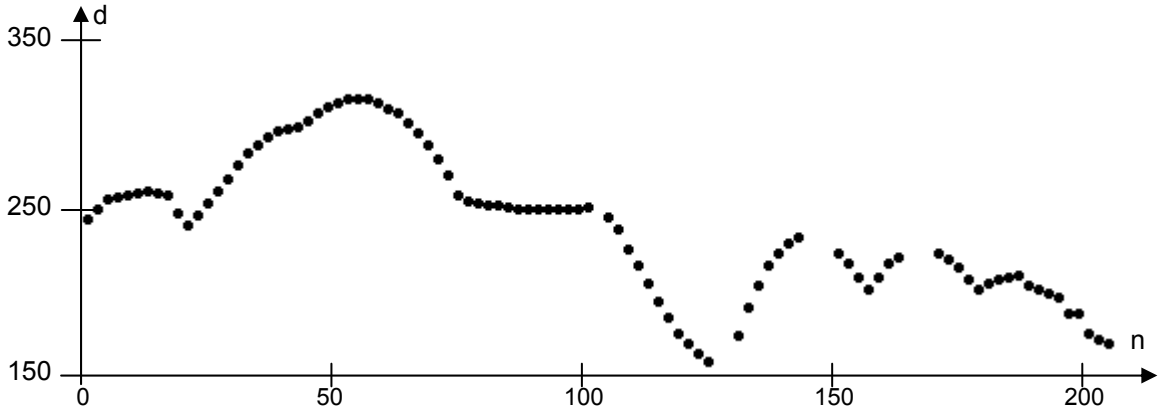


Figure 3.10 Ball trajectories after trajectory mining for the sequence of frames from 48957 to 49167 of FIFA 2002 final. In the figure, n is the serial number of the frame in the sequence and d is the distance between the candidate and the top left corner of the original frame. For the legibility, only the frames with even serial number are drawn.

3.5.4 Ball Tracking

For each obtained ball trajectory, we use a Kalman filter based procedure to track the ball to extend the ball trajectory.

Features for Ball Tracking: Three features are used for ball tracking since our tracking is a local tracking.

- The color range: The color of the ball might change irregularly, but it resides in a range.
- The ball radius upper bound: The radius of the ball might vary irregularly, but it has an upper bound.
- The ball location: It is predicted by the Kalman filter.

Feature Prediction and Extraction: Let T be the trajectory to be extended and $\mathrm{B}(c, r, l)$ be the ball model. $\mathrm{B}(c, r, l)$ is initialized according to the objects in T. Once the ball in frame k is obtained, $\mathrm{B}(c, r, l)$ for frame k+1 can be built. First, an estimator predicts the location of the ball. Another estimator $\mathrm{H}(\bullet)$ predicts the color range of the ball. In the area enclosing the predicted location, we segment the objects with color range $\mathrm{H}(k+1)$. Then, we remove the objects whose radii are larger than the predicted radius upper bound $\mathrm{Q}(k+1)$, which $\mathrm{Q}(\bullet)$ is another estimator. Here, the estimators are Kalman filter-based (see Section 3.4.2 for the use of Kalman filter).

For the detected object O, its probability of being the ball is denoted by $\mathrm{M}(O)$, which is the ratio of the number of pixels in $\mathrm{H}(k+1)$ to πr^2, where r is the estimated ball radius. Finally, the object with the highest probability is considered as the ball if the probability is larger than the given threshold.

3.5.5 Gap Interpolation

Let T_1 and T_2 be two obtained ball trajectories. Let e_1 be the frame number of the last frame in T_1 and s_2 be the frame number of the first frame in T_2. We compute the ball positions between e_1 and s_2 by linear interpolation if $s_2 - e_1 < 13$ (13 means half a second). This interpolation procedure will give

the ball positions when the ball is occluded temporarily or it is out of the camera view temporarily.

The ball tracking and the gap interpolation together form a procedure termed as trajectory refinement. The trajectory processing comprises trajectory mining and trajectory refinement. Figure 3.11 shows the result of trajectory refinement for the sequence of frames from 48957 to 49167 of FIFA 2002 final.

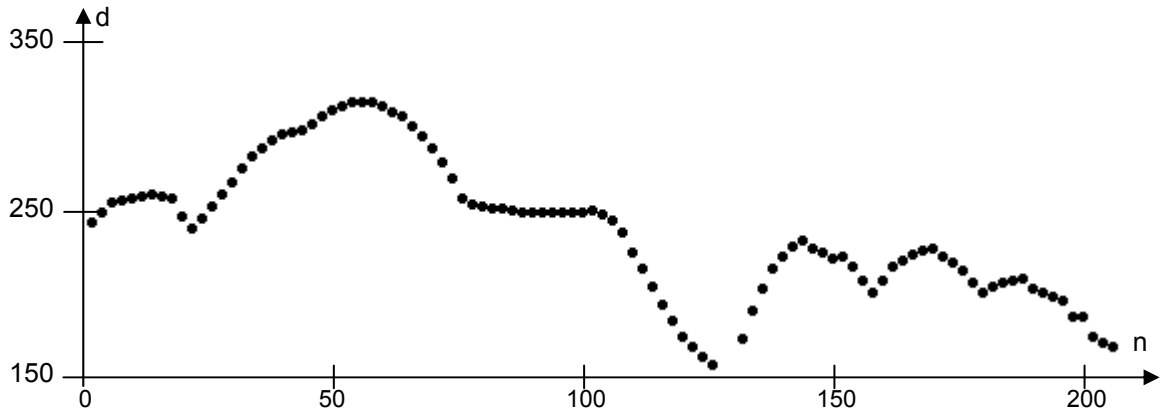


Figure 3.11 Ball trajectories after the trajectory refinement for the sequence of frames from 48957 to 49167 of FIFA 2002 final. In the figure, n is the serial number of the frame in the sequence and d is the distance between the candidate and the top left corner of the original frame. For the legibility, only the frames with even serial number are drawn.

3.6 Experiments on the Ball Detection and Tracking in BSV

This section describes various experiments to explore the different aspects of the algorithm for locating the ball in broadcast soccer video (BSV), which is presented in the preceding sections. We first do experiments to evaluate the performance of the algorithm. Then we do experiments to evaluate some key techniques used in the algorithms. We also compare our algorithm with the algorithm proposed by D'Orazio et al [DACN2002]. The test has been conducted on two MPEG-1 videos of FIFA 2002, which were recorded by a

WinTV^{TM} card connected to a TV antenna. One of the video is the final match (Brazil vs Germany) and the other is a quarter-final (Senegal vs Turkey).

3.6.1 Performance of the Soccer Ball Detection and Tracking

The term *ball location* refers to the coordinates of the center of the ball object in frame. The objective of an algorithm for detecting and tracking the ball is to locate the ball in the ball frames that are visible. In addition, the algorithm aims to infer the ball location when the ball is temporally occluded.

A frame is said to be with the ball if viewers know there is a ball in the frame according to the previous and posterior frames, even though the ball is occluded or is flying very high but the ball must be still in play. However, the closed-up frames, which can be identified by the players' size, are considered as the non-ball frames. This is because the balls in closed-up frames can be detected by template-matching, which is not in the interest of our algorithm. Another case of non-ball frames occurs when a player holds the ball for a long time (longer than 1 second) before starting the ball or just after a break starts. A frame is said to be detected correctly if the algorithm identifies the presence of the ball and its position correctly. As shown in Figure 3.7, our ball candidate generation procedure does obtain a lot of non-ball candidates as these candidates resemble the ball. Fortunately, the ball trajectory mining procedure, including candidate trajectory generation and ball-trajectory production, can identify the ball trajectories correctly as shown in Figure 3.10. In a ball trajectory there are some non-ball objects---false alarms (or false positive detections), as these non-ball objects are very close to the real balls. These false alarms appear only when a player wearing a pair of white soccer boots

or socks is dribbling the ball. However, these non-ball objects do not misguide the tracking procedure even though they are very close to the real balls. Therefore, the ball trajectory mining does not obtain any non-ball trajectory. This is because to decide whether a candidate trajectory is a ball trajectory is much more reliable than to decide whether a sole object is a ball. Figure 3.11 shows the ball trajectories after trajectory refinement, which includes ball tracking and interpolation for small gaps between two produced ball trajectories. Figures 3.7, 3.10 and 3.11 show the candidate distance images for the same sequence in the different process stages. However, some ball trajectories cannot be found as they are too short (less than 6 frames).

It is a hard and tedious job to obtain the groundtruth of the ball location for a soccer video because a video of half a soccer game is about 47 minutes long and has about 70,000 frames. To reduce the difficulty of this job, we first compute the ball locations using our algorithm. Then we obtain the groundtruth through adjusting the obtained ball locations in a visual tool that we built for this task. Thus, we can produce the groundtruth very fast because our algorithm achieves very good accuracy in locating the ball.

Table 3.1 Detection and tracking results for the nine sequences (FIFA 2002 Final, Germany vs Brazil)

Sequences	Ground truth			Detected result			Det'ed & tracked result		
	# frame	# ball	# no ball	# detect	# F	accuracy	# det. traked	# F	accuracy
002900-003001	102	102	0	98	0	96.1%	102	0	100%
003143-003308	266	108	158	266	0	100%	266	0	100%
005368-005503	136	130	6	113	0	83.1%	136	0	100%
005623-005834	212	212	0	196	0	92.5%	212	0	100%
008026-008424	399	323	76	281	3	70.0%	385	3	96.1%
008805-008834	30	0	30	30	0	100%	30	0	100%
008950-009069	110	0	110	110	0	100%	110	0	100%
048957-049167	211	211	0	192	3	91.0%	209	3	99.1%
049256-049974	719	608	111	678	2	94.3%	702	4	97.6%

The detection and tracking results for nine representative sequences from the video of the final match, in which the game starts at the first frame, are shown in the Table 3.1. In Table 3.1, # means “the number of” and # F means “the number of the false alarms”. These nine sequences are good representative because they include long, short, and ball-less sequences.

After we have tested the algorithm on nine sequences from FIFA 2002 Final, we turn to test our algorithm on a whole video of the first half of the game between Senegal and Turkey (FIFA2002), called the test video for short, whose distribution of the various types of sequences is given in Table B.1 and B.2 in Appendix B.

Table 3.2 Performance of the Algorithm on successive 10045 frames of the test video. (FIFA2002 Quarter-Final Senegal vs Turkey)

Seq.	# frm	# ball	#~ball	# B	# ~B	# P	# pos.	accu.	% pos.
L02	3046	2486	560	1723	543	58	17	74.4%	2.46%
X10	42	0	42	0	42	0	0	100%	0%
R03	262	190	72	40	72	150	0	42.8%	0%
M13	441	218	223	184	223	16	0	92.3%	3.63%
X11	71	0	71	0	71	0	0	100%	0%
M14	487	244	243	195	243	4	0	89.9%	0.82%
X12	95	0	95	0	95	0	0	100%	0%
X13	24	0	24	0	24	0	0	100%	0%
M15	690	310	380	179	380	0	0	81.0%	0%
X14	154	0	154	0	154	0	0	100%	0%
M16	420	177	243	108	243	2	0	83.6%	0.48%
M17	796	660	136	532	135	33	1	83.8%	4.27%
X15	48	0	48	0	48	0	0	100%	0%
X16	73	0	73	0	73	0	0	100%	0%
X17	59	0	59	0	59	0	0	100%	0%
L03	1472	1282	190	933	190	27	0	76.3%	1.83%
X18	26	0	26	0	26	0	0	100%	0%
X19	27	0	27	0	27	0	0	100%	0%
X20	27	0	27	0	27	0	0	100%	0%
M18	566	268	298	147	297	14	1	78.5%	2.65%
~F(19)	1218	0	1184	0	1184	0	0	100%	0%
Total	10045	5835	4210	4041	4191	304	19	82.0%	0.19%

Table 3.2 shows the results on the successive 10045 frames of the test video. In Table 3.2, “# frm” means “the number of frames”; “# ball” means “the

number of frames having the ball"; "# ~ball" means "the number of frames not having the ball"; "# B" means "the number of frames that the algorithm correctly spots the ball"; "# ~B" means "the number of frames that the algorithm correctly tells that there are no ball"; "# P" means "the number of frames that we wrongly tell the locations of the ball"; " # pos." means "the number of frames that we wrongly say that the ball appears"; "~F(19)" means "19 sequences in which no frame shows the field".

Table 3.2 also shows that the algorithm does not perform well on the replay sequences. There are at least two factors to explain this. First, most of the frames in replay sequences are from the side cameras, not the main camera. Thus, the ball is more often occluded and its size relation to the salient objects is not considered in our algorithm. Second, the scene change is fast in replay sequences. As a result, it is hard for the ball to form a trajectory. In the following experiments, we will discard the replay sequences because they are not the objective of our algorithm. Table 3.2 shows that our algorithm performs very well on the ball-less sequences. To be concise, we will not include the ball-less sequences as well. The number of the remaining sequences is 68.

Table 3.3 shows the experimental results for the remaining 68 sequences, which is much better than the results shown in Table 3.4 obtained with the algorithm in [DACN2002, DGLD2004]. In all sequences, the number of false alarms is very low, hence, reflecting that our detection results are very reliable. There are two types of false alarms. One type is that the algorithm reports some balls for frames without the ball. Another type is that the algorithm reports the wrong locations of the ball for the frames with the ball.

For each ball trajectory, the false alarms are a small portion of all elements of the trajectory. In other words, there is no ball-trajectory containing only false alarms or a main portion of false alarms.

Table 3.3 Detection and tracking results of the 68 sequences.

Segment	Ground truth			Detection			Detection and tracking		
	#frame	#ball	#~ball	#right	#false	Accuracy	#right	#false	Accuracy
S01-S14	2512	1567	945	1803	22	71.78%	2173	45	86.50%
M01-M40	24465	16563	7902	17881	194	73.09%	19721	598	80.61%
L01-L14	25460	20945	4515	19592	402	71.29%	20450	591	80.32%
Total	52437	39075	13362	37834	618	72.15%	42344	1234	80.75%

Table 3.4 Comparison on the detection results between the detection procedure of our algorithm and the CHT algorithm.

Segment	Ground truth			CHT			Our detection procedure		
	# frame	#ball (n1+n2)	# ~ball	#right	# false	accuracy	#right	# false	accuracy
S0-S14	2512	933+634	945	1321	309	52.59%	1803	22	71.78%
M01-M40	24465	10924+5639	7902	12729	2862	52.03%	17881	194	73.09%
L01-L14	25460	14796+6149	4515	10043	4390	39.45%	19592	402	71.29%
Total	52437	26653+12422	13362	24093	7561	45.95%	37834	618	72.15%

D'Orazio et al proposed a Circle Hough Transform (CHT) algorithm to detect the ball in soccer video in [DACN2002, DGLD2004]. Their algorithm identifies the ball using CHT and neural classifier, which works well for the videos recorded by their own camera [DACN2002, DGLD2004]. However, in BSV, a lot of the balls are not circle in shape due to ball deformation and the ball merging with other objects. As a result, their algorithm produces bad results for locating the ball. The detection comparison between the detection procedure of our algorithm and the CHT is shown in Table 3.4. In the column "#ball" of Table 3.4, "n1" is the number of the balls that are neither occluded nor merged with other objects and "n2" is the number of the balls that are either occluded or merged with other objects. Actually, the CHT works quite well as it has detected 24093 balls from 52437 frames if we realize that 24093

equals to 90.40% of 26653, which are the number of the balls that are neither occluded nor merged with other objects.

3.6.2 Experiments on Ball Size Estimation

The ball sizes are estimated from the ellipse, the goalmouth, and the players. Here, some experiments are done to explore the performances through three types of salient objects. We chose a video sequence of frames from 68340 to 69098 for our experiments. Firstly, the ball locations and sizes are obtained manually. Then we compute the ball sizes as though the balls are located at the center of the frame, which are computed from the real ball sizes by the following formula.

$$S = \left(\frac{5\mathrm{h}}{4\mathrm{h} + 2\mathrm{r}} \right) \times S_0 \qquad (3.13)$$

where h is the height of the frame, r is the row of the ball location, and S_0 is the manually obtained ball size.

The above formula is the simplified version of ball size variation array, in which the simplification is based on the observation that the changes in the ball size along the row are very small and can be ignored. The accuracy comparison of the ball size estimation through ellipses, goalmouth, and players are given in Table 3.5. Let F be the considered frame. Assume the true ball size of the frame is S_0, the estimated ball size is the *S*. The standard error in ratio for the frame is the absolute value of $1 - S \times (\mathrm{S}_0)^{-1}$.

Table 3.5 Comparison on estimating the ball size in three types of salient objects for the sequence of the 68340 to 69098 frames of Senegal vs. Turkey.

Type of salient objects	Number of applied frames	Average standard error in pixel	Average standard error in ratio
Ellipse	423	0.831	0.159
Goalmouth	31	0.967	0.179
Player	305	1.232	0.214

3.6.3 Experiments on Ball Size Filter

Here we vary ball sizes to be used in the filter to find the relation with the number of the obtained ball candidates and the relation with the number of all the produced candidates, which are shown in Figures 3.12 and 3.13. Figure 3.12 shows that the number of total candidates decreases when the used ball size increase. This indicates that the objects with a size slightly smaller than the ball are more than the objects with a size slightly larger than the ball size. Figure 3.12 shows that we can produce the best results by setting the ball size used in the filter to be 1.7 times of the real ball size.

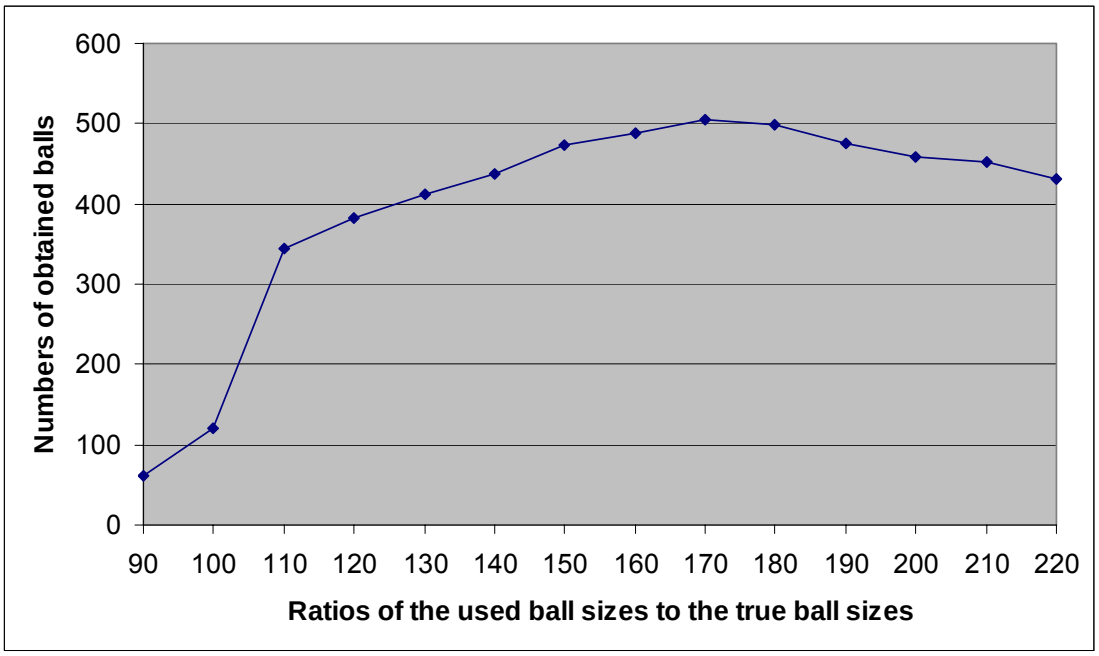


Figure 3.12 Relation between the number of the true-ball candidates and the used ball sizes in the ball size filter.

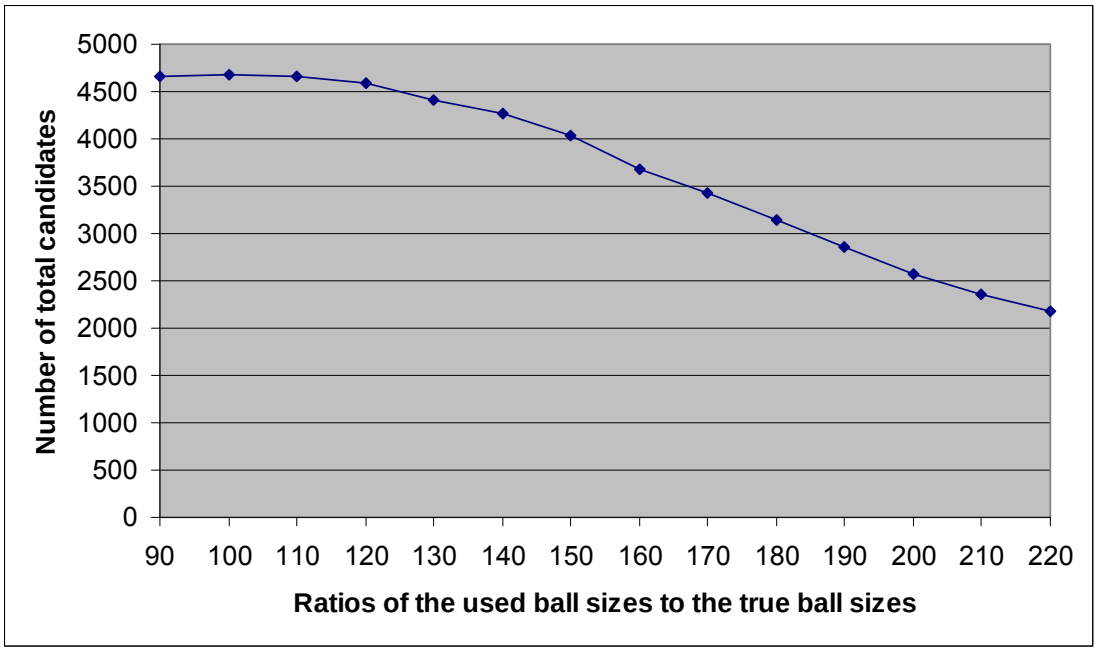


Figure 3.13 Relation between the number of all the candidates and the used ball sizes in the ball size filter.

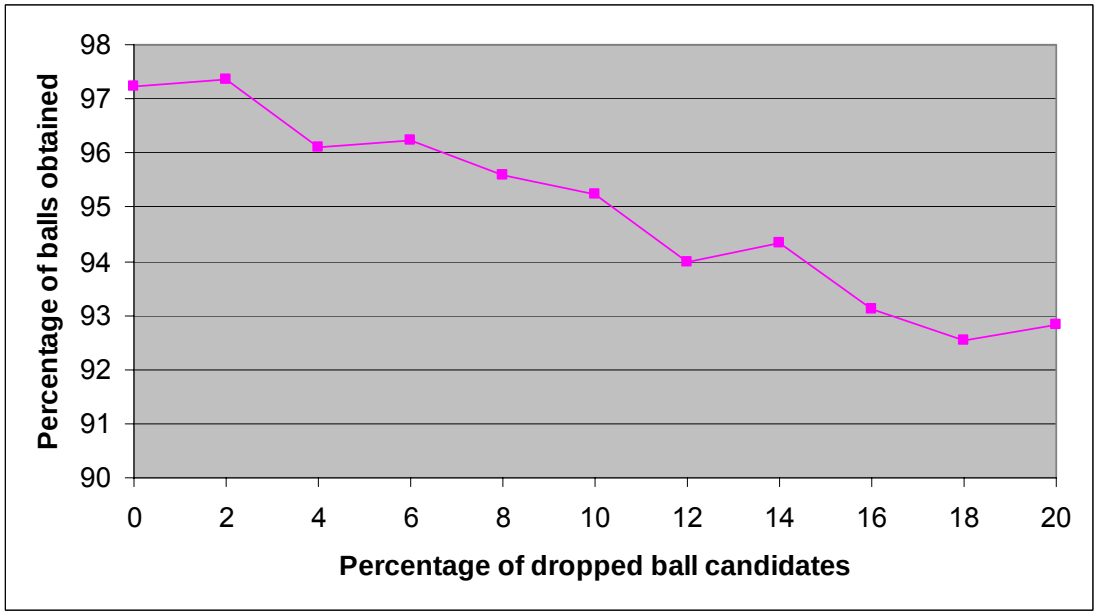


Figure 3.14 Relation between the percentages of the found ball and the dropped true-ball candidates in the ball trajectory mining procedure.

3.6.4 Experiments on the Robustness of Ball Trajectory Mining

Here we test the robustness of the ball trajectory mining by dropping the true ball candidates in various percentages. The experimental results are shown in Figures 3.14 and 3.15, they demonstrated that our procedure is very robust.

Figure 3.14 shows that our trajectory mining procedure can achieve a high percentage of the ball correctly even though the ball candidates have dropped by 20%. Figure 3.15 shows that the false alarms are kept to less than 5% until the ball candidates are dropped by 12%. The percentage of false alarms keeps increasing when more ball candidates are dropped.

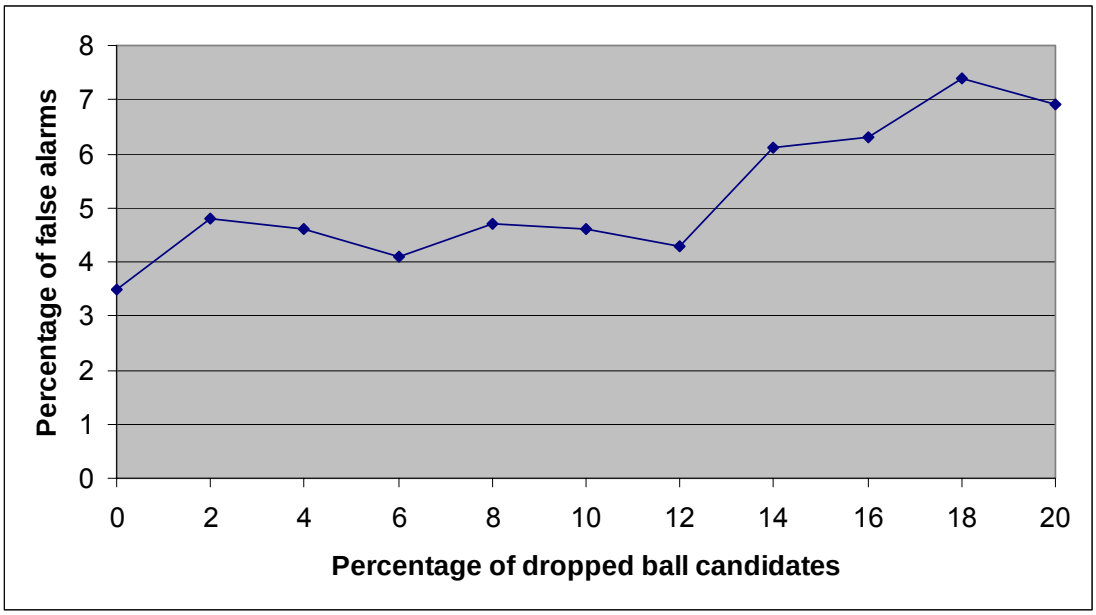


Figure 3.15 Relation between the percentages of the false balls and the dropped true-ball candidates in the ball trajectory mining procedure.

3.6.5 Contribution of Penalty Mark Filter

The penalty mark filter is a special filter, whose task is to identify the penalty marks and remove them. The penalty mark filter is a necessary filter because the algorithm will consider the trajectories of penalty marks to be ball trajectories if we do not have this filter.

We chose a video sequence including frames from 36890 to 36970 of the final match of FIFA 2002 to illustrate the contribution of the penalty mark filter. For this video sequence, the candidates with and without penalty mark filter are shown in Figure 3.16. Then the mined ball trajectories from Figure

3.16 are shown in Figure 3.17. The results showed that the algorithm may identify the trajectory of penalty marks as a ball trajectory.

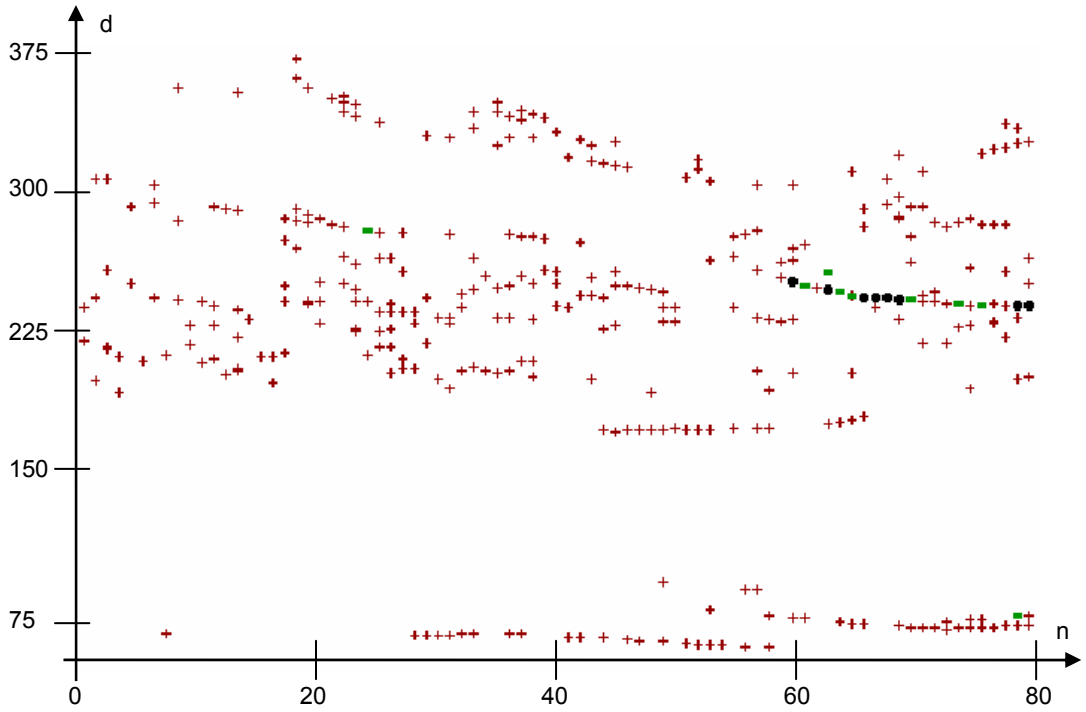
3.7 Application of the Trajectory-Based Approach to BTV

The preceding sections have successfully developed a trajectory-based algorithm for locating the ball in BSV. Under the encouragement of this success, this section applies the trajectory-based approach to develop an algorithm for locating the ball in broadcast tennis video (BTV).

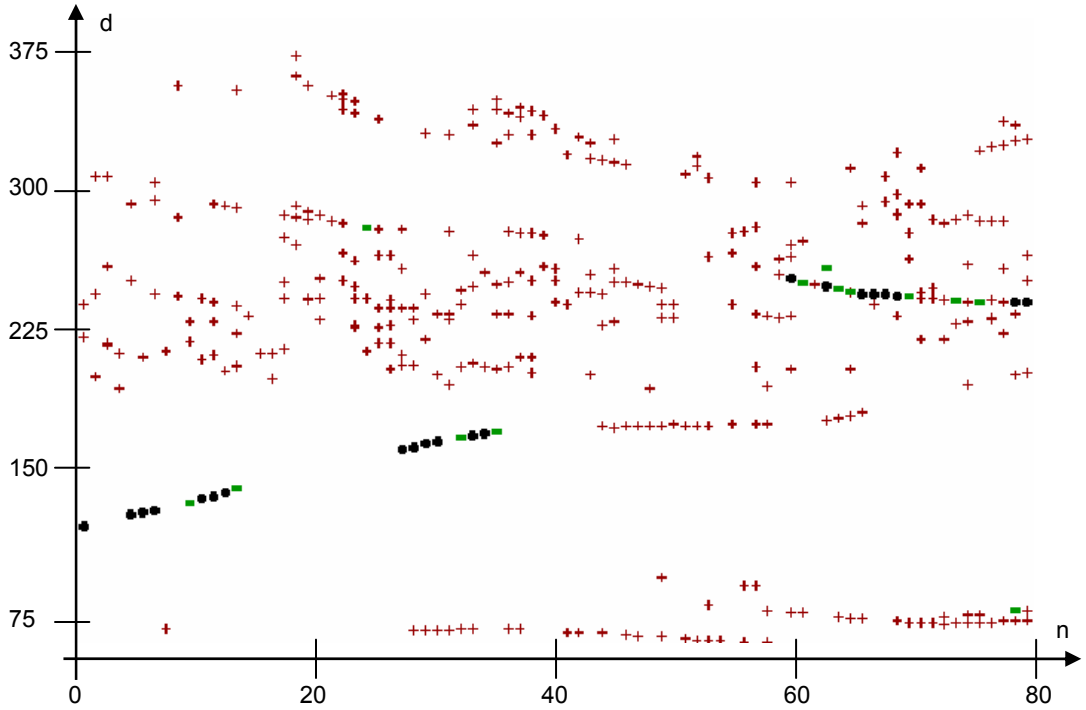
3.7.1 Challenges of Tennis Ball Detection and Tracking

Since tennis is one of the most popular sports, tennis analysis receives much attention. This interest is further motivated by the possible applications over a wide range of topics, such as tactic analysis, indexing, retrieval, and summarization [Miy2003, SuLJ1998, Mili2000]. Since the “ball” is the most important object in tennis, detecting and tracking the ball became crucial in tennis video analysis. However, so far there is no algorithm able to obtain satisfactory results in locating the ball in BTV due to the following challenges:

- The ball has serious deformation due to the reflection
- The ball is very small, especially at the far side of the camera. For some frames, balls are so small that human eyes are unable to see them
- The ball is often occluded by the players and the net. It may mix with the audience and complicated background
- The appearance of the ball varies irregularly over frames. Its size, shape, and speed change irregularly over frames

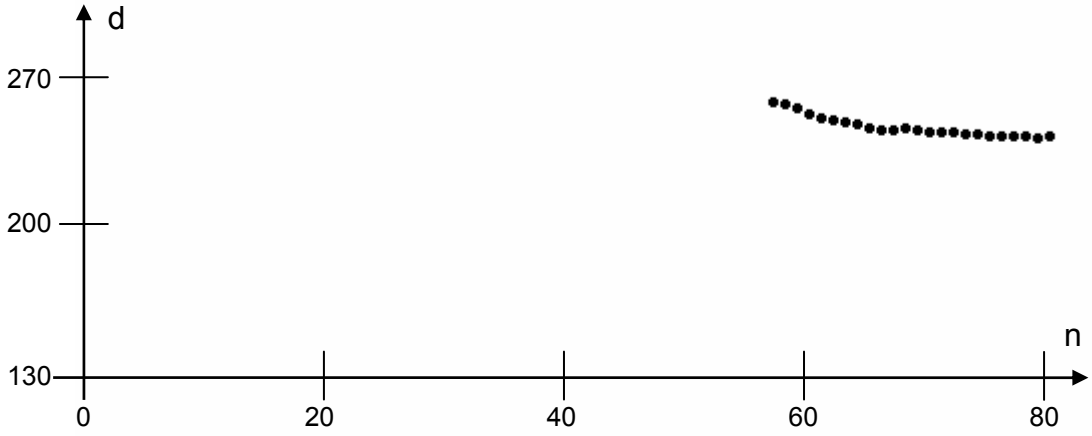


(a) The candidates with penalty mark filter.

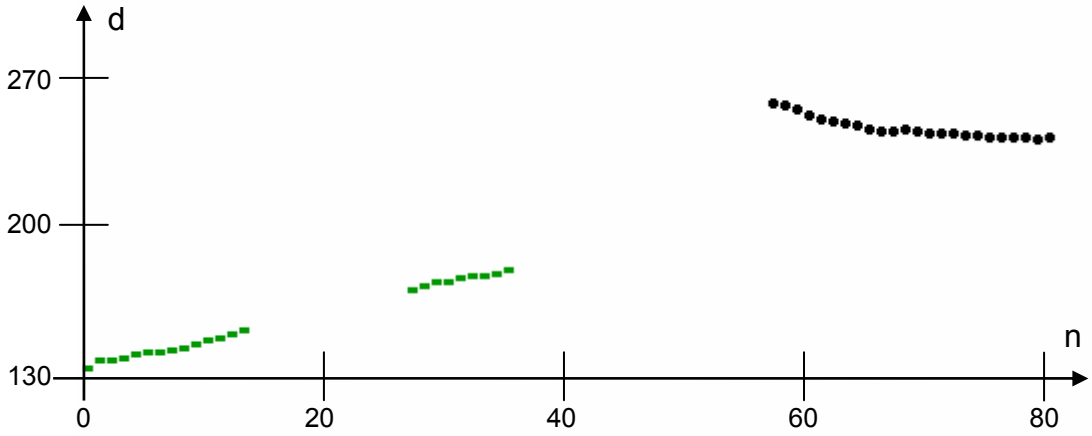


(b) The candidates without penalty mark filter.

Figure 3.16 Two DISTANCE-images of a sequence showing the effect of the penalty marker filter. The sequence of frames is from 36890 to 36970 of FIFA 2002 final. In the figure, n is the serial number of the frame in the sequence and d is the distance between the candidate and the top left corner of the original frame. The black dots, the green rectangles, and red crosses are the candidates in Category 1 to 3 respectively.



(a) The mined ball trajectories with penalty marker filter. The produced is the true ball trajectory.



(b) The mined ball trajectories without penalty marker filter. These two trajectories of green rectangles are the trajectories of the penalty marker.

Figure 3.17 Mined trajectories with and without the penalty marker filter (on the sequence of frames from 36890 to 36970 of FIFA 2002 final). In the figure, n is the serial number of the frame in the sequence and d is the distance between the candidate and the top left corner of the original frame.

This section proposes a trajectory-based algorithm for ball detection and tracking in BTV. The proposed algorithm can be viewed as another instance of the trajectory-based approaches [YXLT2003, YuTW2003, YXTL2003, YSWC2004]. At the same time, the proposed algorithm has its own new elements. First, it uses locations of the players at hitting points (a point when the racket hits the ball) to infer the locations of the ball. Second, it uses these hitting points to infer the turning points of the ball route. Last, it infers which player hits the ball from the data of player locations, hitting points, and ball candidate locations.

The proposed algorithm is for locating the ball in each frame of the sequence, which has five components as depicted in Figure 3.18: player detection and tracking, candidate detection, hitting point detection, candidate trajectory generation, and trajectory processing. The components in the dotted rectangles are the ones that do not appear in the algorithm for locating the ball in BSV.

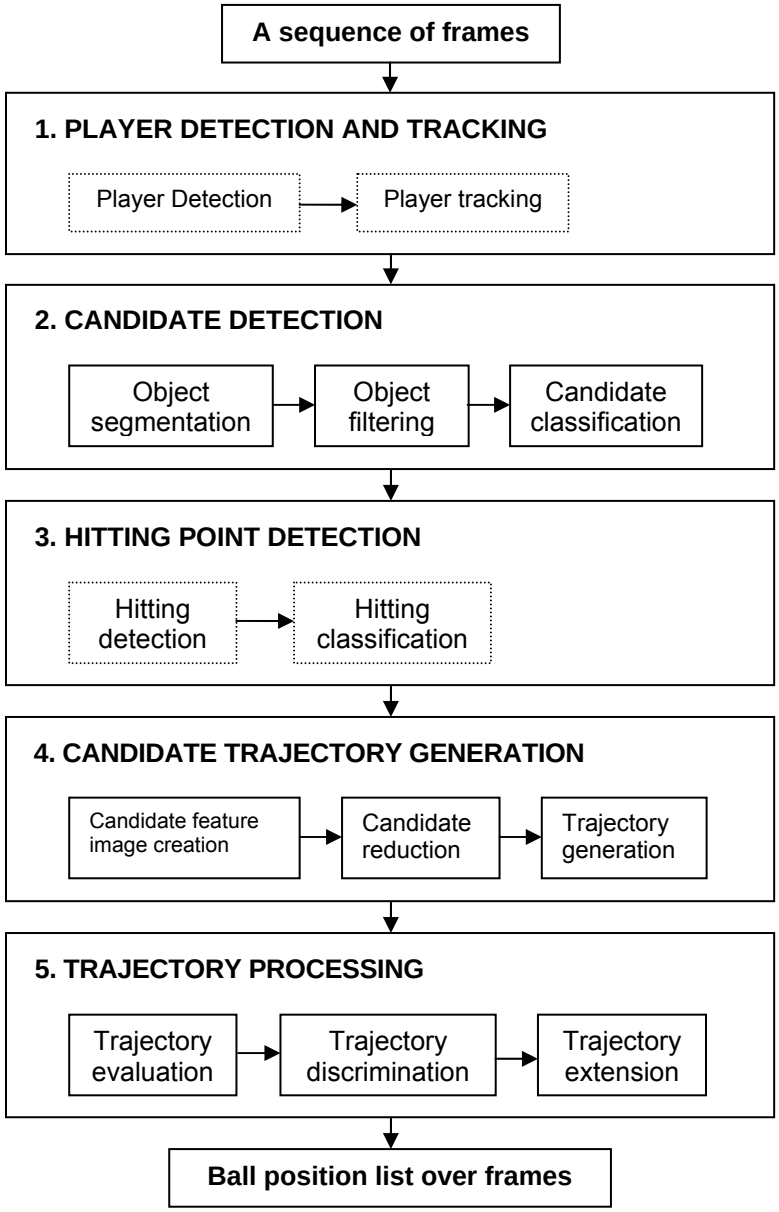


Figure 3.18 Block diagram of the algorithm for locating the ball in broadcast tennis video.

In the *Player Detection and Tracking* component, we find the locations of far-player and near-player in the game for each frame using mainly the color of players, the far-player (near-player) being the one who is far from (near) the camera. In the *Candidate Detection* component, we produce the ball candidates for each frame by removing the identified non-ball objects. Then we classify the candidates into two categories. In the *Hitting Point Detection* component, we use the algorithm presented in [XMXK2003] to detect the hitting points in each sequence from audio. Then we differentiate the hittings into far-player hittings or near-player hittings. In the *Candidate Trajectory Generation* component, we first create the various candidate feature images (CFI). Then we use a Kalman filter-based procedure to produce the candidate trajectories from each CFI. In the *Trajectory Processing* component, we take three steps to obtain the ball trajectories. Firstly, we evaluate each candidate trajectory. Then we identify the ball trajectories through a selection procedure. Finally, we extend the ball trajectories according to the hitting points and player locations.

The rest of the section is organized as follows: Section 3.7.2 presents the proposed algorithm. Section 3.7.3 gives the various experimental results.

3.7.2 Algorithm for Locating the Ball in BTV

Since the proposed algorithm depicted in Figure 3.18 has a similar structure to the algorithm for locating the ball in BSV presented in the preceding sections, we will not explain the algorithm in detail. We spend more efforts to describe the procedures that are designed only for this algorithm or the

procedures that have much difference from the similar procedures in the algorithm for locating the ball in BSV.

Player detection and tracking: For each frame, we first find the boundary of the court and remove all long lines and the net. We then use the pixel growing procedure to segment the objects of the frame. Knowing that every frame has two-side players (far-player and near-player), we conclude that two/four relatively large objects should be the players. In addition, the criteria below are also considered: (a) the near-player should be at the lower part of the frame and the far-player should be at the upper part of the frame; (b) there should be one or two players on one side of the net; (c) two sides of the net should have the same number of players. Without loss of generality, the following discussion will consider the singles match only, where there are two players in total.

Ball candidate generation: To produce the ball candidates for each frame, the below sieves are built to remove the non-ball objects.

Ball Size Sieve Θ_1: We filter out the objects out of the ball size range, which are estimated from the detected net.

Line Sieve Θ_2**:** We filter out all long lines, including straight lines and curves as the ball cannot be deformed into a long line.

Ball Color Sieve Θ_3: We filter out objects without ball color pixels.

Shape Sieve Θ_4: The ball image can have a shape quite different from a circle, but in most frames its width-to-height and height-to-width ratios are less than 2.5 according to the results of our statistical analysis.

Ball Location Sieve Θ_5: At the instance of hitting, we remove the objects that are far from the hitting player. We will explain how to find the player hitting the ball in the latter part of this section.

Each sieve Θ_i is a Boolean function on domain $\mathrm{O(F)} = \{\, \mathrm{o} : \mathrm{o}\ \textit{is an object in frame}\ \mathrm{F}\}$. After sieving, the remaining objects of $\mathrm{O(F)}$ form the ball candidate set $\mathrm{O(F)}$ of frame F.

$$\mathrm{C(F)} = \{\mathrm{o} :\ \mathrm{o} \in \mathrm{O(F)},\quad \Theta_i(o) = 1 \text{ for } i = 1 \text{ to } 5\} \tag{3.14}$$

Ball candidate classification: Like computing the probability of soccer ball candidate, we compute the probability $P(o)$ that the candidate is the tennis ball. According to the probability $P(o)$, the candidates in $C(\mathrm{F})$ are divided into two categories. Category 1 contains the objects with high probability and Category 2 contains the objects with low probability.

Hitting detection: The sound emitted by the racket hitting the ball is distinct. In this chapter, we use the algorithm by Xu et al [XMXK2003] to obtain frames where hittings occur.

Hitting classification: The detected hittings are divided into two categories: near-player hittings and far-player hittings, which are hits by the near-player and that by the far-player, respectively. This classification is achieved based on two cues: (a) in frames around instances of near-player hittings, we can obtain good and clear ball candidates; (b) the ball must be hit alternatively by the two players in a tennis game.

Ball location inference: The above hitting classification can tell us which player hits the ball at each hitting point. Thus, we can estimate the ball location according to the player location as the ball is in the vicinity of the player when he/she is hitting the ball. The detected hittings will also serve as

the start and end points of the ball trajectory. In addition, the hitting can help in removing non-ball objects as the ball must be near the hitting player.

Trajectory processing: We use a procedure similar to the one to process the soccer ball trajectory. The difference is that we can compute the ball locations as the camera seldom moves. In Figures 3.19, 3.20, and 3.21, black vertical

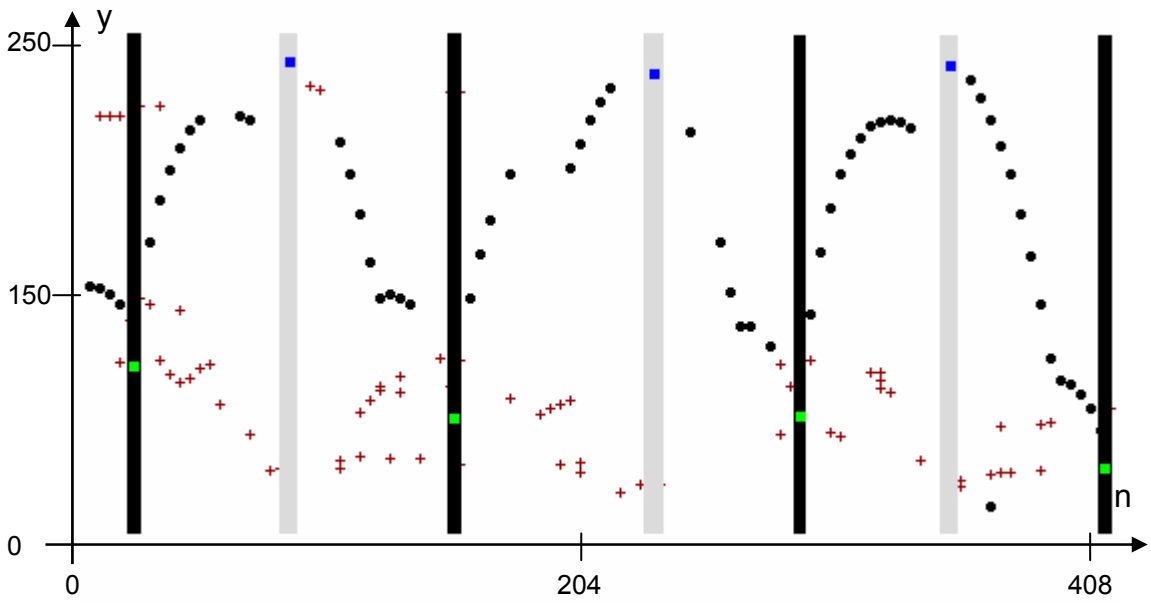


Figure 3.19 Obtained ball candidates. The black dots and red crosses stand for categories 1 and 2 respectively. The blue and green rectangles are the hitting points by the far- and near- players respectively.

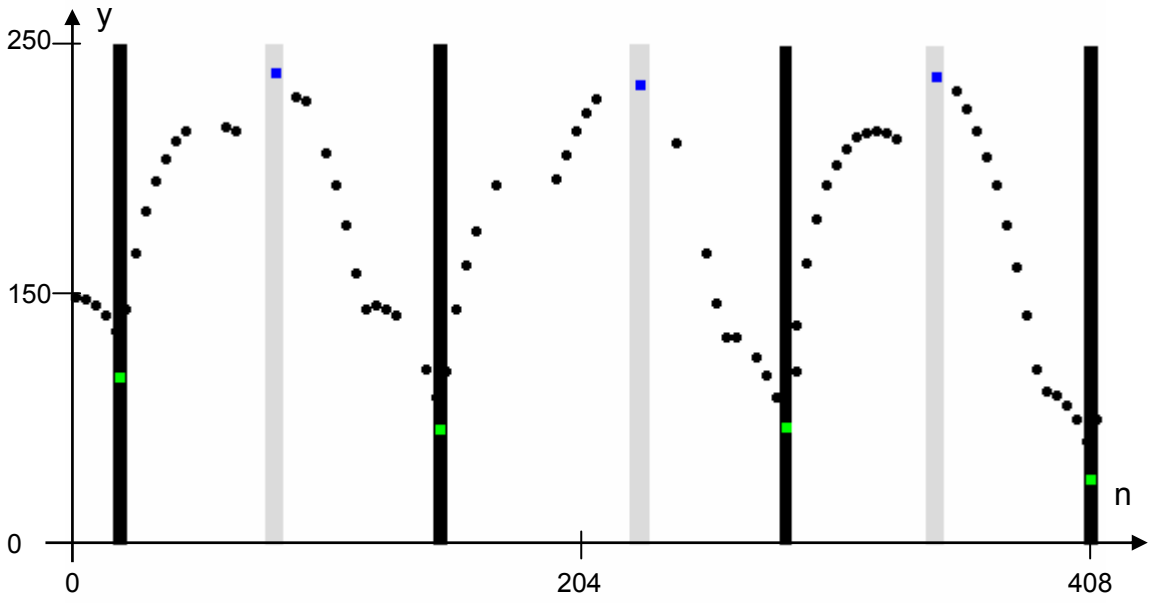


Figure 3.20 Mined ball trajectories. They are mined from all the candidates of the sequence. The blue and green squares are the hitting points by the far- and near- players respectively.

lines and grey vertical lines denote frames where the near-player hittings and the far-player hittings occur respectively. Figure 3.20 shows the identified ball trajectories; and Figure 3.21 shows the final ball trajectories.

Ball location computation: When the ball is near the far-player, it may be occluded by the player, too small to be seen, or be mixed with the audience. Thus, it is useless to track the ball. So, we compute the ball location by extending the trajectory to the far-player's hitting location. This location is inferred from the obtained player's location and the obtained hittings as the ball is near the player who hits it. The ball location extended in the above-described way is not exact, but it can greatly facilitate content analysis. For example, this location is enough to identify the winning-pattern of the game.

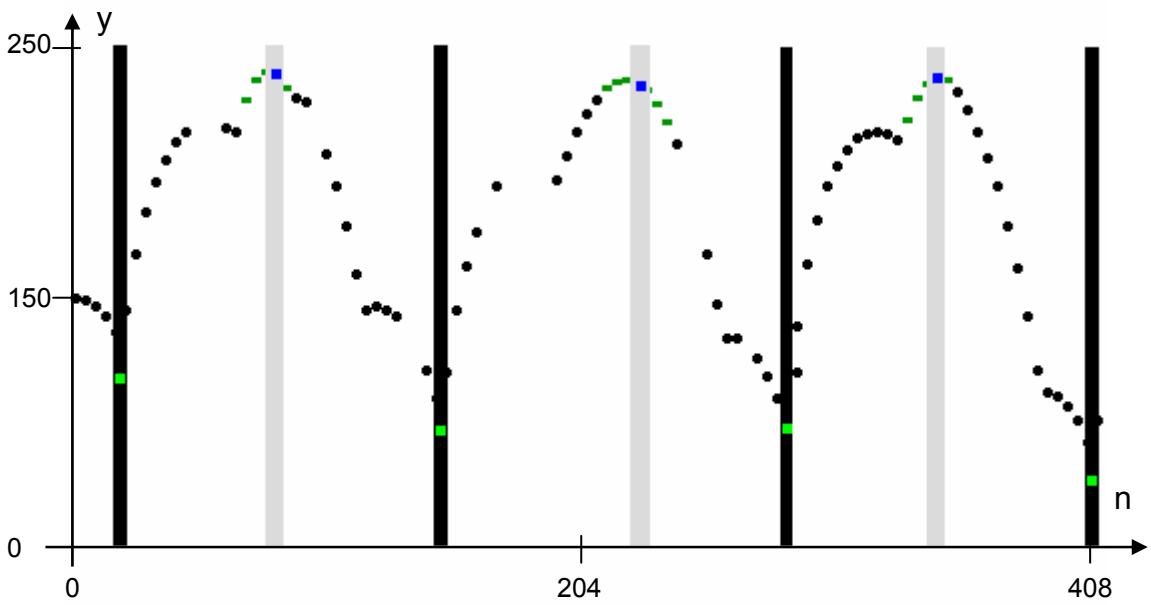


Figure 3.21 Obtained final ball trajectories. The green rectangles are the ball locations computed by quadric curve fitting. The blue and green squares are the hitting points by the far- and near-players respectively.

3.7.3 Experimental Results of Locating the Ball in BTV

The proposed algorithm has been tested on 7 sequences (a total of 120 seconds), which are from MPEG-1 video recorded from TV signal. The

content of the video is the Men's Final of FRENCH OPEN 2003, which is the game between Juan Carlos Ferrero of Spain and Martin Verkerk of Holland, held on June 8, 2003. The frames were grabbed by DirectX 9.1. Note that, each frame must have two players. Hence, their ground truths are the number of frames in the considered sequence. The experimental results of player and hitting detection are shown in Table 3.6, in which column “player” records the number of frames each player is detected within a sequence; column “hitting” shows the ratio of hitting frames to the total number of successfully detected hitting frames associated with that player within the sequence.

Table 3.6 Results of Player Detection and Tracking.

Sequences	# frame	Near-player		Far-player	
		Player	Hitting	Player	Hitting
05355-06132	778	767	11/11	732	10/10
08905-09334	430	430	6/6	358	4/5
14526-14866	341	341	3/3	341	2/2
19025-19274	250	232	3/3	250	2/2
24492-24981	490	449	6/6	476	6/7
27520-27891	372	372	2/2	372	2/2
36960-37310	351	349	4/4	351	3/3

The experimental results of identifying the ball locations are shown in Table 3.7, in which column “balls/frames” gives the ratio of total frames to frames containing the ball; column “d+t” gives the numbers of detected balls and tracked balls separately; column “(d+t)%” gives the percentage of “d+t” to the number of frames containing the ball; column “final” shows the number of the obtained balls by detection, tracking, and computation; column “final%” shows the percentage of “final” to the number of frames containing the ball. Notice that the percentage of the ball detection and tracking are not very high. This is because the ball cannot be seen when it is close to the far-player. The promising indication is that the proposed algorithm computes the ball

locations from the obtained player's hitting locations. Table 3.7 also shows the results of detecting the tennis by the algorithm presented in [Miy2003] for comparing our algorithm with it.

Table 3.7 Results of Ball Detection and Tracking.

Sequences	balls/ frames	detected by Miy2003		detected+tracked		Computed	
		d	**d%**	d + t	(d+t)%	final	final%
05355-06132	738/778	336	45.5%	336+121	62.0%	738	100%
08905-09334	376/430	256	68.1%	265+17	75.0%	361	97.0%
14526-14866	294/341	239	81.3%	250+11	88.8%	294	100%
19025-19274	171/250	92	53.8%	100+38	80.7%	171	100%
24492-24981	441/490	293	66.4%	300+20	72.6%	426	97.6%
27520-27891	275/372	141	51.3%	146+174	80.0%	275	100%
36960-37310	349/351	254	72.8%	269 +10	79.9%	349	100%

3.8 Summary

This chapter has presented a trajectory-based algorithm for locating the ball in broadcast soccer video (BSV), which used a series of adapted and proposed techniques to overcome the multiple challenges of the problem. This algorithm contains four contributions.

- The first contribution is a ball size estimation method that estimates the ball sizes from the sizes and locations of salient objects. This estimation is based on the image generation principle of pin-hole camera and it overcomes the challenge that the ball size changes over frames irregularly.
- The second contribution is the ball candidate generation technique. This technique used anti-model approach to produce the ball candidates for each frame. This anti-model approach avoids the challenge to build the ball representation. At the same time, it achieved a high recall for obtaining the true ball candidates.

- The third contribution is the candidate feature images (CFIs) that present the spatial and temporal data within an image. These images enable the temporal filters and trajectory-based analyses to be applied to them. In another word, these CFIs are not only the visual presentation media, but also the objects to process on.
- The fourth contribution is trajectory analysis. The trajectory evaluation and production procedures successfully produce the ball trajectories through processing candidate trajectories. Based on the reliable ball trajectories, the ball trajectory refinement procedure extends the ball trajectories to obtain the extended ball trajectories.

Section 3.7 has presented another trajectory-based ball detection and tracking algorithm, which locates the ball in broadcast tennis video (BTV). Besides making use of the techniques used in the algorithm of locating the ball in BSV, this algorithm has made its own contributions.

- It used locations of the players at hitting points (a point when the racket hits the ball) to infer the locations of the ball.
- It used these hitting points to infer the turning points of the ball route.

Lastly, it inferred which player hits the ball from the data of player locations, hitting points, and ball candidate locations.

Chapter 4

Detection of Ball-Related Event in Broadcast Soccer Video

This chapter presents two applications of ball detection and tracking. The first application is to use the ball locations to detect ball-related events in broadcast soccer video. The ball locations used are computed by the ball detection and tracking algorithm for BSV presented in the preceding chapter. The other application is to apply the information derived from ball location and the results of detecting ball-related events to enrich the enhanced soccer videos, generated in computer graphics technology.

4.1 Event and Ball-Related Event

In sports video, an event is an interesting occurrence. The events are defined by the particular sports game and are usually well-known to both players and viewers of the sport. For example, in soccer video the interesting events are goals, corner kicks, free kicks, penalty shots, etc. In tennis, the interesting events are scoring, serving and play/break.

In soccer videos, the events can be divided into ball-related and non-ball-related events. An event is called *ball-related* if its occurrence involves the

ball location in the soccer field or ball status (moving, stopping, holding, etc). In a soccer game, some of the events are non-ball-related. For example, kicking an opponent, red card, and stumbling are non-ball-related. Nevertheless, most of the events in a soccer game are ball-related. For example, *touch*, *passing*, *goal*, *team possession*, *play/break*, *corner kicking*, and *hand ball* are ball-related events. Table 4.1 shows examples of ball-related events, with names and definitions extracted from "FIFA Law of the Game" [FIFAlaw] and the "Soccer Dictionary" [SoTe2002].

Table 4.1 Definitions of the Selected Ball-Related Events of Soccer.

Name	Definition
Break	The ball is out of the field or the play has been stopped by the referee
Clear	To kick the ball away from one's goal
Corner Kick	A kick made from corner arc
Dribbling	Run with the ball at your feet, playing it on every step or every other step
Driving	Playing the ball well forward and running after it
Free Kick	The player kicks a stationary ball without any opposing players within 10 feet of him
Goal Kick	A type of restart where the ball is kicked
Hand Ball	A deliberate handling of a ball by a player other than the goalkeeper in the penalty area
Near Missing	A ball that crosses the goal line out of but very close to the goalmouth
Passing	A player kicks the ball to his teammate from inside the goal area away from the goal
Penalty Kick	Penalty kicks are taken from the penalty mark. All players (of both teams) except the kicker and opposite goalkeeper must remain on the field of play outside the penalty area and penalty arc
Play	A generic term as in "play the ball"
Possession	A player or a team in control of the ball
Shooting	A player kicks the ball at the opponent's net in an attempt to score a goal

In this chapter, we propose a structured method for detecting a number of ball-related events in broadcast soccer video, including *touch, passing, goal, team possession, and play/break*.

4.2 Related Work in Event Detection in Soccer Video

We first give a quick review of previous work on event detection in soccer video. The content of a soccer video is intrinsically multimodal since a video author uses visual, auditory, and textural channels together to convey meaning. In detecting event, text was used as an aid for various sports videos such as football [MHBK2002], baseball [ZhCh2002], soccer [SnWo2003]. In detecting the events of soccer, many algorithms used mainly visual and auditory low-level features. Some efforts have started to use the object-related features, derived from the detected objects such as ball, goalmouth, ellipse, etc., to improve the event detection performance [ABCB2003(a-c), ACN2002, DGLD2004, GLCZ1995, YXLT2003, WLXY2003]. There was an increasing interest in detecting soccer events with the aid of the ball location [GLCZ1995, CHHG2003, ToQi2001] or based on the ball trajectory [YXLT2003, YLLT2003] because many events closely correlate with the ball. According to the used features, the event detection algorithms can be divided into the following six categories:

- Visual low-level feature-based
- Low-level auditory feature-based
- Visual and auditory low-level feature-based
- Shape-based
- Ball location-aided
- Ball trajectory-based

The first three categories are based on the *low-level features*; the last three categories are based on the *object-related features*. In the following sections, we examine the algorithms in these six categories.

4.2.1 Visual Low-Level Feature-Based Methods

There were some event detection algorithms using only visual low-level features. Kawashima et al [KaYA1994] proposed a qualitative analysis of team behavior based on multi-scale region analysis for soccer video. They used the color backprojection to identify the player and used the color histogram analysis to discriminate the team of players. Then they evaluated the team behavior through analyzing the relative places among the players.

Taki et al [TaHF1996] employed several cameras placed along the touchline on the soccer field, covering the whole soccer field. They extracted the field and the players from fixed-camera video (FCV). Then they evaluated the team performance through analyzing the movement of players.

Xie et al [XiCD2002] proposed an algorithm for parsing the structure of soccer video. They used hidden Markov models (HMM) to model the various states of soccer game. Standard dynamic programming techniques were used to obtain the maximum likelihood segmentation of the game into two states: play and break. The features used in HMM are dominant-color ratio (the ratio of the number of the pixels in grass color to the number of all the pixels in a frame) and motion intensity (the average magnitude of effective motion vectors in a frame).

Xu et al [XXCD2001] classified frames into three kinds of views (global, zoom-in and close-up). This classification is based the observed fact that the grass areas of the global, zoom-in and closed-up frames are in a decreasing order. The paper then used heuristic rules to segment the input video play and break segments. The algorithms in the paper are very fast because it developed the effective rules based on few features.

Misu et al [MNZI2002] used a technique of integrating multiple visual features, including color statistics, texture of whole object, texture of head, and local motion vector, to robustly track players from soccer video. The paper made different templates for different features and used pattern-matching unit to evaluate the matching cost functions. Simultaneously, it generated an observation covariance matrix. The final evaluation value was computed based on the matching cost functions and the covariance matrix. The strength of the paper is that it used the covariance matrix, which makes the algorithm relies on the right features in the corresponding status because the matrix varies with the occlusion status.

Assflag et al [ABBN2002] used two approaches to detect events such as penalty kick, free kick and corner kick from BSV. The first approach uses camera motion only, whereas the second also included information regarding the location of players on the soccer field.

Instead of frame-level analysis, several shot-level sports video parsing techniques were proposed in [NgPZ2001, DuXT2003]. Ngo et al [NgPZ2001] used histograms to represent motion and color features from shots and exploited a hierarchical clustering approach to aggregate shots with similar visual low-level features. Through manual investigation of the clustering results, they tried to explain semantic meanings for each cluster.

Ekin [Eki2003] in his PhD thesis used a rule-based algorithm to detect goal. Their rules were built on the shot level and the shots were classified by a Bayesian classifier based on the visual features [EkTe2003a, EkTe2003b, EkTe2003c]. In his thesis, Ekin detected events using the auditory features for tennis and baseball video but not for soccer video.

4.2.2 Auditory Low-Level Feature-Based Methods

The auditory channel also provides strong clues for the presence of semantic events in video documents. Xu et al [XDXT2003, XMXK2003] employed representations of the audio signal in terms of time-domain measurements and frequency-domain measurements to train game-specific sound recognizers (e.g. “Applause”, “Whistling”, “Excited/plain commentator speech”, “Silence”, and “Ball hitting”, etc.) by hierarchical Support Vector Machines to detect events such as serve, reserve, ace, return, and score in tennis video. Clearly, there are strong relationships between those significant game-specific sounds have and the action of players, referees, commentators, and audience in broadcast sports videos. These actions can be heuristically mapped to interesting events according to specific sports game rules.

4.2.3 Visual and Auditory Low-Level Feature-Based Methods

The integrated use of low-level features from visual and auditory channels is a natural extension of single modality low-level feature approaches in video indexing research. Because there is more information available, the results of event detection improve when the low-level features from both video and audio are employed. In using visual and auditory low-level features to detect events, many algorithms have been developed for other sports video such as football video [CZKA1996, MHBK2002], baseball video [HHXG2002, RuGA200, ZhCh2002], tennis video [Ekin2003, XDXT2003], and multiple-game video [Ekin2003, NaHu2001], but no such algorithms were developed specially for soccer video before the work of [YXLT2003].

Woudstra et al [WVPM1998] modeled auditory and visual information of soccer video. Their objective is to develop a soccer video retrieval system, which semantically retrieves soccer video based on the auditory and visual features. They discussed the possibility to detect events using the extracted features, but they give neither more details nor the implementation.

Snoek and Worring [SnWo2004, SnWo2002] gave a review on detecting event using auditory, visual and textual information. They presented a framework for multimodal video indexing. Snoek and Worring [SnWo2003] proposed an algorithm to detect event from soccer video using auditory, visual, and textual information. They addressed the synchronization problem of the heterogeneous information sources.

Recently, Duan et al [DXTX2003, DXTX2004] developed a unified framework for semantic shot classification of sports video including soccer, tennis, basketball, etc. The paper used the domain knowledge of each game to perform a top-down video shot classification, including identification of video shot categories for a specific sports game, visual and auditory feature representation, and supervised learning. Based on the results of semantic shot classification, events are detected using shot change templates. A successful application of their semantic shot classification is an audio-visual integration scheme for detecting events in tennis videos [XDXT2003].

4.2.4 Shape-Based Methods

An algorithm is called *shape-based* if it detects event mainly using the detected shapes to infer events. For soccer video, the detected shapes facilitate the event detection. For example, the scope to detect goal can be reduced through

detecting goalmouth because a goal only happens in a goalmouth scene. Many algorithms made use of detected shapes as one kind of cues to detect event.

Gong et al [GSCZ1995] proposed an automatic soccer parsing system to classify a sequence of frames into various play categories (e.g. shot at left goal, top-left corner kick, etc.) based on a priori model mainly comprising line mark recognition, motion detection, ball detection, etc.

Wan et al [WLXY2003] developed an automatic algorithm to track the camera field-view in real time. They detected straight lines and ellipse and used lines, ellipse, motion vector, etc to infer the camera field-view.

Assfalg et al [ABCB2003(a-c)] developed an algorithm to extract soccer highlights, including forward launch, shooting, turnover, placed kick. Their algorithm first detected the shapes such as lines, corner, etc. Then it classified the soccer field zones. Last, it used the model based on the zone classification and camera motion to extract the highlights.

Although the above-mentioned algorithms used other features, the detected shapes played a major role in detecting events.

4.2.5 Ball Location-Aided Methods

An algorithm is called *ball location-aided* if it detects events mainly using low-level feature and using the ball location as an aid to improve the accuracy of event detection. An increasing number of algorithms used the ball locations in event detection and video enhancement for soccer games. They mainly used visual and auditory low-level features and the information derived from ball location was used as an aid to improve the accuracy of event detection [YYYL1995, ABCB2003(a-c), CHHG2003, GSCZ1995, Mili2000, YaSM2002].

Yow et al [YYYL1995] created a trace of all ball positions at a single frame using the ball locations in a sequence of frames to produce interesting panoramic views, which can be considered as a kind of video enhancement.

Gong et al [GSCZ1995] classified a sequence of soccer frames into various play categories, such as shot at left goal, top-left corner kick, play in right penalty area, in midfield, etc, based on a priori model comprising four major components: a soccer court, a ball, the players, and the motion vectors.

Tovinkere and Qian [ToQi2001] proposed a method for detecting semantic events which may happen in a soccer game. Their method uses a set of heuristic rules which are derived from a hierarchical entity-relationship model representing the prior knowledge of soccer events. Their method requires the 3D position information of the ball as input. However, their paper does not give a method to obtain the 3D positions of the ball.

Haas et al [HMSP2002] developed a system to decide whether there is a goal by analyzing the relative position between the ball and the goalmouth. Chen et al [CHHG2003] used the ball locations to improve the event detection. The events in their experiments were *shooting*, *goal*, *corner kick*, and *free kick*.

Assfalg et al [ABCB2003(a-c)] developed an automatic annotation system to identify the various highlights based on the various video cues including the ball motion. Since they think that the ball locations cannot be detected robustly, they did not develop an algorithm for locating the ball.

4.2.6 Ball Trajectory-Based Methods

An algorithm is termed as *ball trajectory-based* if processing ball trajectory is one of the key steps in event detection. Some algorithms make use of the ball

location but not the ball trajectory to improve the event detection. Here, we propose to detect event based on ball trajectory, in which the primary results were presented in [YXLT2003]. Generally speaking, events cause the changes of ball motion because most of the events are the results of the interactions between players and the ball. These changes form the pivots in the ball trajectory. Hence, we can obtain the event candidates by finding the pivots of ball trajectory. The ball speed also gives us some cues for event analysis. In addition, the relation between the ball trajectory and the soccer field also helped us in event detection. This relation analysis can give us better performance because the algorithms using it can closely follow the rules of soccer game. The algorithms in trajectory-based event detection not only improve play/break analysis and high-level semantic event detection, but also detect the basic actions and analyze team ball possession, which may not be achieved based only on the low-level feature [YXLT2003, YLLY2003].

4.2.7 Low-Level Feature and Object-Related Feature Approaches

Both the low-level feature-based and the object-related feature approaches have their virtues and demerits. There are three virtues in the former approach. First, the features are relatively easier to be extracted. Second, the algorithms are fast because they did not need to do time-consuming object detection. Last, the algorithms of the former approach can sometimes be generalized to detect events for different sports video. The demerit of this approach is that the results may not be very accurate because the low-level features do not correlate exactly with the events. The virtue of the latter approach is that it can achieve relatively more accurate results than the low-level feature approach. But the

object-related features are very difficult to be obtained because it is difficult to detect objects, especially the ball in ball-game video.

4.3 Our Proposed Event Detection Algorithms

We first give the key idea behind our approach. In soccer game, most events are caused by interactions. There are two important types of interactions: player-ball and player-player. Player-ball interactions relate to the ball motion, whereas player-player interactions relate to the team tactics. In our approach to event detection, we take advantage of our accurate ball detection and tracking results to help in event detection and we target *ball-related events* in broadcast soccer video, including *touch*, *passing*, *goal*, *team possession*, and *play/break*.

Our method decomposes each ball-related event into (i) a number of simpler *basic actions* involving the ball, and (ii) other salient information relating to surrounding objects (players, referees, etc). We use the ball location and trajectory information to help in detecting the basic actions and these are then combined with other information to detect the more complex ball-related event.

4.3.1 Detection of Basic Actions

The *touch* (of the ball) is the *basic action* because many events are defined according to how a player touches the ball. Thus, touch detection is a chief component in many other ball-related events, as listed in Table 4.1.

Relation between Pivot and Touch: One of the main objectives of the soccer players is to control the motion of the ball. To achieve this objective, players attempt to *touch* the ball with the right force and direction. Each touch of the ball

will alter the speed and direction of the ball. These touches translate to motion change points in the trajectory of the ball. We call such motion change points *pivot points* of the ball motion, or *pivots* in short. However, pivots may also be caused by a number of other factors, such as camera motion, ball bounce, and so on. Therefore, pivot is a necessary (but not sufficient condition) for touches of the ball.

Detection of Pivot: Now we first detect the pivots. Let V = {*p* | *p* is a local speed minimum point or an acceleration start point}. Let $f_1(p) = r$ and $f_2(p) = c$ be the functions, representing Y curve and X curve of the ball position over frames, where *r* and *c* are the row and the column of the ball center in the frame *p* respectively. Let S_1 = {*p* | *p* is a local trajectory minimum of f_1}, and S_2 = {*p* | *p* is a local trajectory maximum or minimum of f_2}. Then, we consider $S=V+S_1+S_2$ to be the pivot set of the segment. V, S_1, and S_2 are the sets of the points that the ball has the significant changes in (*apparent*) speed, row and column respectively. The *apparent speed* is the speed of the ball in the image coordinate system. Figure 4.1 shows the pivot detection result for a sample segment in which a vertical bar indicates a pivot point. In Figures 4.1 to 4.3, *v* stands for the ball speed and *n* stands for the serial number of the frame.

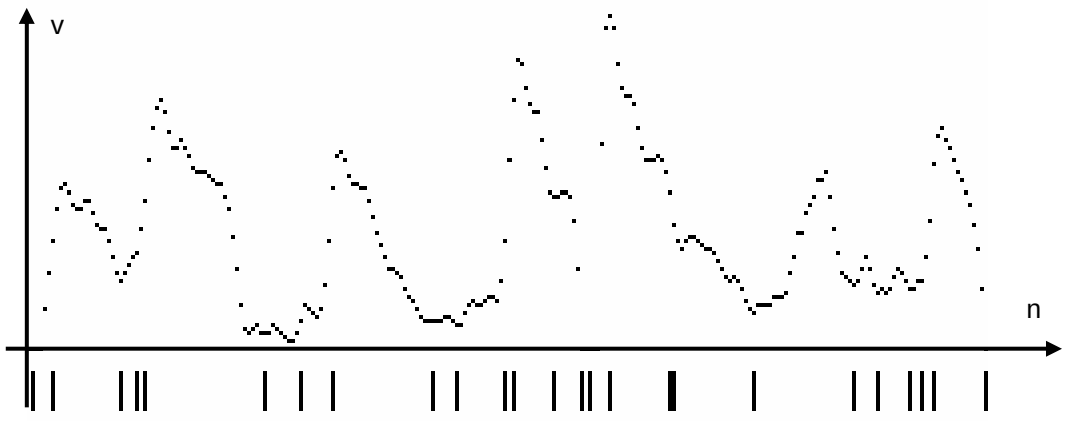


Figure 4.1 Pivots from ball trajectory (vertical bars).

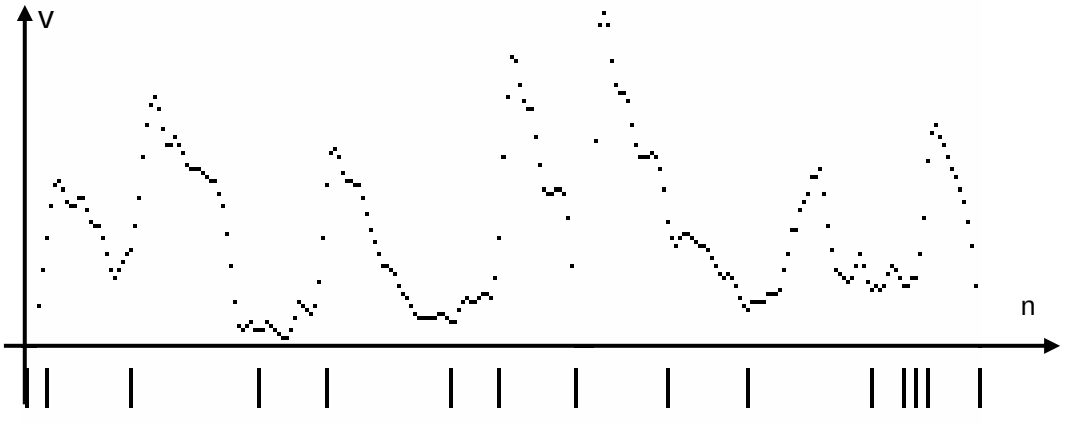


Figure 4.2 Touch points (vertical bars).

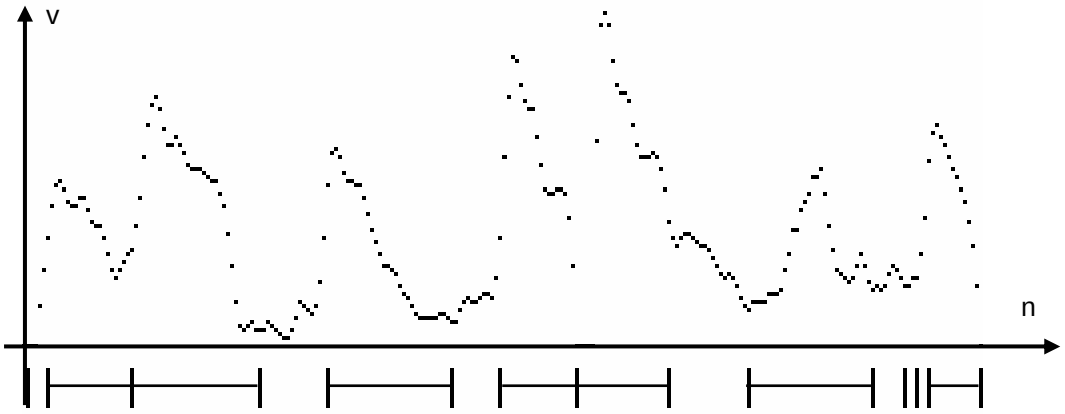


Figure 4.3 Passings (line segments between two bars).

Touch Detection: A *touch point* is defined to be a frame where a person touches the ball. The touch includes ball kick (by foot), and other categories of touches. When a person touches the ball, the ball trajectory should form a pivot point. For each pivot point, we also check whether a person touches the ball. The pivots where nobody touches the ball are removed, and the rest form a set of touch points. Figure 4.2 shows the curve of the ball speed over frames in which the black vertical bars indicate the touch points.

Detection of Passing: A passing is an action that a player passes the ball to his teammate. Normally, such an action produces a significant ball trajectory. Passings are vital to the understanding the game as they comprise a large portion of the game. So we designed a scheme to detect passings. Besides the

passing, a soccer game consists of mainly fighting, dribbling, possession transit, and shooting. These actions comprise a smaller portion of the game. Figure 4.3 shows the obtained passings for a sample video segment, in which a horizontal line segment with two vertical bars at its two ends indicates a passing.

4.3.2 Detection of Complex Events

Complex events are detected based on the detection result of the basic actions, the ball trajectory, and the result of mark detection. We choose *goal* as an example to show how complex events are detected in our proposed approach. The steps involved in goal detection are shown in Figure 4.4.

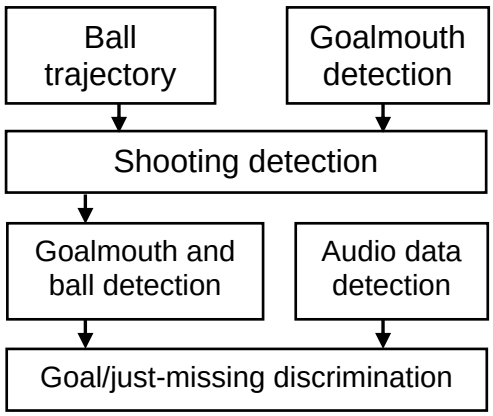


Figure 4.4 Architecture of goal detection.

In the first phase, we first detect the goalmouth, and then we detect the shooting against the goalmouth. In the second phase, we further find the ball that is close to the goalmouth for a number of frames. However, the fact that the ball is in the goalmouth in a frame can not infer that there is a goal since we are detecting the event that happened in 3D space through processing 2D frames. Hence, we decide whether there is a goal by considering the ball trajectory and goalmouth position relation. Notice that in a near-missing

shooting, the ball might also be in the goalmouth for a few frames. Indeed, the challenge we are facing is how to avoid classifying the near-missing into the goal. We can further verify the goal with whistling and audience shout.

4.4 Team Possession Analysis

We analyze the team ball possession based on the result of touch detection as described above. To credit the ball possession, we must determine which team touches the ball. For each frame where a touch is encountered, the objects in the ball neighborhood are extracted first. Then, each object is evaluated to check whether it is a player (person). Finally, a Support Vector Machine (SVM) is used to recognize the team of each person found [ChLi2002]. In addition, when a team consecutively touches the ball, we consider that the team possesses the ball. The flowchart of team ball possession analysis is depicted in Figure 4.5.

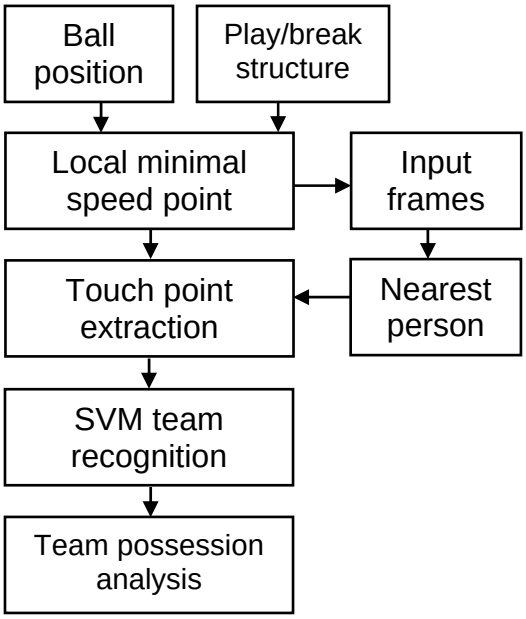


Figure 4.5. Flowchart of team ball possession analysis for broadcast soccer video.

4.4.1 Color histogram

In a soccer game, the people in the soccer field are in five categories: the players in Team A and B, goalkeepers in Team A and B, and the referee. The jerseys of these five categories of people are in five different colors. Hence, the color histogram of a person can determine which team he belongs to. For each type of people, we manually identify a color to differentiate them from other people in advance. For each such color, we build several color bins which span a range of color around it. Then for each person we calculate his color distribution on the pre-built bins, which is used to evaluate his team by a SVM developed by C.-C. Chang and C.-J. Lin [ChLi2002].

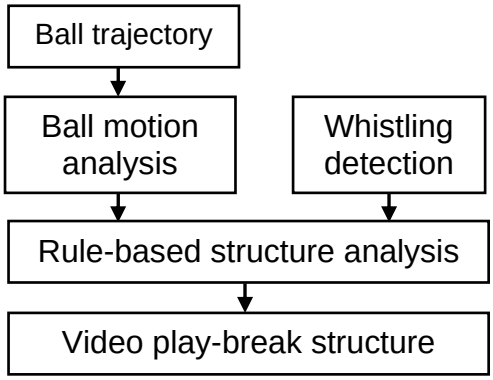


Figure 4.6 Architecture of play-break analysis.

4.5 Play/break Structure Analysis

In a soccer game, the ball is either in play or break [XCDS2002, XXCD2001]. There is a break when the ball is out of the field or the referee stops the game. In the soccer video, almost all play frames show the ball, but not all the frames with the ball are play frames. Thus, the ball trajectory provides the solid basis for the play/break structure analysis. A series of connected trajectories form a ball curve, whose two ends may not be in play. Thus, our aim is to decide how

long of two ends are not part the play portion. Besides the motion speed of the ball, whistling is another reference for determining play/break cutting point. Hence, we analyze the play/break structure based on the ball trajectory with the aid of whistling. Figure 4.6 depicts the architecture of play/break analysis.

4.5.1 Whistling Detection

In soccer game, there are three types of whistling: long, double, and multiple. All the whistlings indicate game resume and stop or separating points of play/break. Since Xu et al [XMXK2003] have developed a whistling detection algorithm for BSV, we consider the result of whistling detection as one of the inputs.

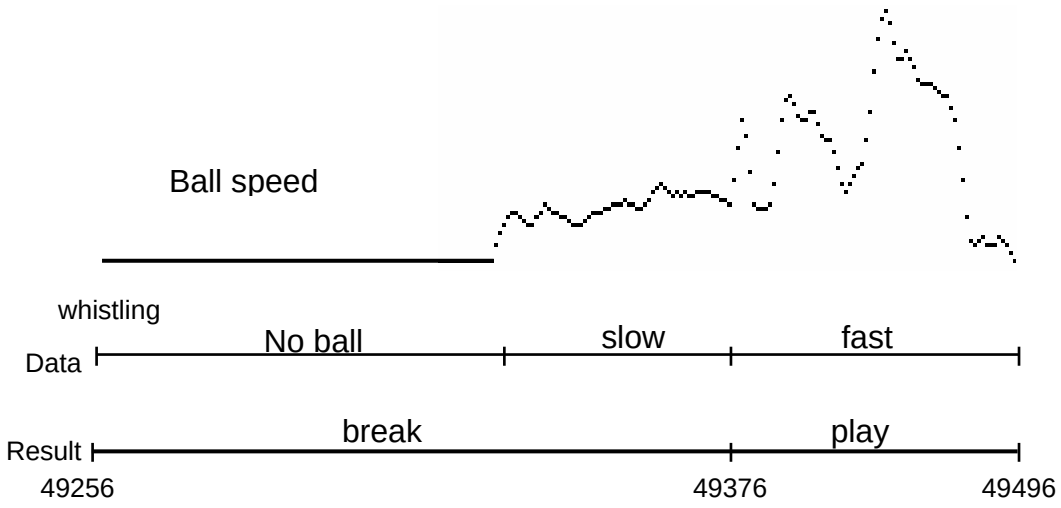


Figure 4.7 A sample of play/break separation. The result for FIFA 2002 final (frames from 49256 to 49496).

4.5.2 Structure Analysis

Now we describe the functions of the system of play/break analysis shown in Figure 4.6. The inputs are the ball trajectory and the whistling over frames (time). We also use the apparent ball speed to indicate whether the frame

belongs to the play portion too. The primary goal of the first phase of the system is to compute the derived information such as the ball speed and the grass ratio in each frame. A rule-based procedure decides the separation points by fusing the derived information from the ball trajectory and the whistling. Figure 4.7 shows the result for finding one play/break separation point.

4.6 Experimental Results of Event Detection

Here we present the experimental results of touch, passing, goal, near-missing detection, team ball possession analysis, and play/break analysis.

4.6.1 Results of Event Detection

For semantic analysis to BSV, the inputs to our detection modules are the ball trajectory and the video. Various analysis results are shown in Tables 4.2-4.4 respectively. We can detect almost all the touches visible in the frames. Unfortunately, not all the touches are shown in the frames. Furthermore, when a touch is not shown in the frame we do not know whether the corresponding trajectory is a pass or otherwise. This is the main reason why we are unable to detect some passings. Initially, we have tested with 6 segments that contain 27 touches and 18 passings for touch and passing detection. The total time length of these 6 segments is 55 seconds. For goal detection, we have tested with 2 games each that took place during day and night. Two games contain 5 goals and 16 near-missings. For goal and near-missing detection, we differentiate them by checking the relative position between the ball trajectory and the goalmouth. However, the issue for this discrimination is that for some near-

missings, the ball is very close to the goalmouth in a number of frames. Thus, the algorithm considers them as goals.

Table 4.2 Event detection performance.

Event	Precision	Recall	# events
Touch	95.6%	81.5%	27
Passing	100%	76.8%	18
Goal	100%	100%	5
Near-missing	81.2%	100%	16

There are no results of detecting touches and passings in existing papers. For goal detection, previous results are at shot level, i.e. the algorithm detected the goal shots [ChSZ2004] and the reported recall and precision were 90-94%. Here the goal detection is at frame-level, i.e. the algorithm reports the frames in which the ball are inside the goalmouth.

4.6.2 Results of Team Ball Possession Analysis

We cut the play portion of the soccer video at its touch points. Hence, the play portion is divided into a set of touch segments. A touch segment is called a Team A (B) possession segment if its two touches are from Team A (B). For the team possession analysis, the player team discrimination is reliable and the error is mostly inherited from the error of touch point detection. In the expression “v/w” in Table 4.3, v is the correctly detected number and w is the ground truth of team ball possession.

Table 4.3 Team possession analysis performance**.**

Sequence	# frames	Brazil	Germany	Accuracy
002900-003001	102	100/80	0/0	78.4%
003240-003308	169	0/0	169/169	100%
005368-005503	136	0/0	0/0	100%
008026-008296	271	183/271	0/0	66.5%
048957-049102	146	0/0	100/146	68.5%
049415-049974	560	0/0	481/560	85.9%

4.6.3 Results of Play/Break Analysis

Table 4.4 shows play/break analysis results using only ball speed, only grass ratio, and both the ball speed and the grass ratio. The results show that the (*apparent*) ball speed derived from the ball locations in frame can improve the performance when they are used together with the grass ratio.

Table 4.4 Play/break analysis performance.

sequences	# fram.	ball motion		grass ratio		ball and ratio	
		# corr	%	# corr	%	# corr	%
1006300-1006723	424	348	82	330	77	338	79
1007203-1007317	115	48	41	45	39	47	40
1007880-1008181	302	172	56	225	74	237	78
1008347-1008557	211	163	77	158	74	178	84
1008792-1008924	133	86	64	63	47	83	62
1008974-1009684	711	461	64	424	59	421	59
1009819-1010399	581	454	78	552	95	581	100
1010490-1011433	944	503	53	675	71	660	69
1011457-1011964	508	424	83	423	83	443	87
1012403-1012622	220	103	46	143	65	143	65
1012650-1013377	728	578	79	526	72	534	73
1013588-1013943	356	317	89	292	82	312	87
1014556-1017351	2796	1425	50	1728	61	2277	81
1017353-1017546	194	97	50	127	65	135	69
1017876-1018646	771	441	57	674	87	683	88
1018764-1019217	454	390	85	291	64	311	68
1019340-1019857	518	235	45	490	94	503	97
1019859-1020200	342	151	44	221	70	234	68
1020227-1023272	3046	1992	65	2051	67	2533	83
1023643-1024083	441	402	91	181	41	187	42
TOTAL	**13795**	**8790**	**64%**	**9619**	**70%**	**10840**	**79%**

In Table 4.4, “ball motion”, “grass ratio”, and “ball and ratio” mean that we analyze the play/break using ball motion only, grass ratio only, and both ball

motion and grass; “# corr” and “%” mean the numbers of the frames the algorithms correctly tell their play/break classes and the corresponding accuracy.

4.7 Enhancement and Enrichment of Broadcast Soccer Video

In this section, we present a video generating and enrichment system to enhance and enrich broadcast soccer video. We generate the video based on the results of camera calibration and the generated video is enriched with music and illustrations. The illustrations are figures and icons which illustrate the video contents and help viewers to understand the video better.

4.7.1 Overview of the Proposed System

Figure 4.8 gives the overview of the proposed video generating and enrichment system. It performs four main steps. In the first step, the feature points of an image are extracted based on our straight line detection algorithm and ellipse detection algorithm. They are used to compute *eye-position, look-at,* and *fovy*. In the second step, the 3D positions of the ball and players are estimated from their locations in the images. Next, a video is generated using computer graphics. Finally, the system creates figures and icons to illustrate the video contents. The music suitable for the video contents is also added.

The input video of the system, which is recorded from TV signal using a WinTV^{TM} card, is in MPEG-1 format (*the resolution is 352X288*). For each frame, we construct a corresponding frame using Direct3D^{TM} and compress all reconstructed frames in the original order into a new video clip.

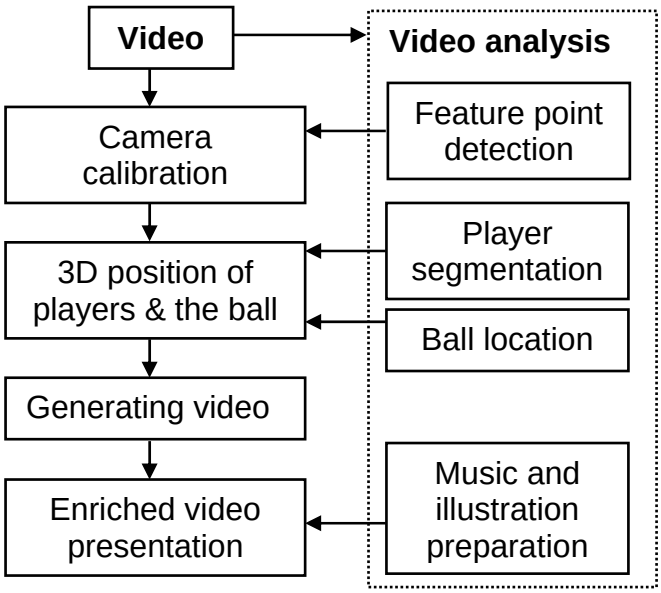


Figure 4.8 Overview of the enhancement and enrichment system of broadcast soccer video.

4.7.2 Camera Calibration

To generate a frame, four groups of parameters of camera must be known [BeBi2000, HaZi2003]:

1. Position of the camera (*eye-position*): (eye*x,* eye*y,* eye*z*).
2. Direction of viewing (*look-at*): (center*x,* center*y,* center*z*).
3. Orientation of the camera (*up-vector*): (up*x,* up*y,* up*z*).
4. Field of view: (*fovy, aspect*).

Among these parameters, the value of the aspect ratio *aspect* is 352:288. The eye position (eye*x,* eye*y,* eye*z*) is *video-defined,* i.e. each video has only one eye position, which can be computed through camera calibration using six 2D to 3D point correspondences [HaZi2003, LiHF1990]. To accurately find the eye position, this chapter selects several pairs of the left and right goalmouth frames to compute the eye position. We use goalmouth frames because in such frames the points on the goal-bar are not on the playground and enough

crossings of lines can be located. As [BeBi2000] has stated, the orientation of camera (up*x,* up*y,* up*z*) can be safely assumed to be $(0, 1, 0)$. Hence, the *frame-varying* parameters, which have different values for each frame, are *look-at*, i.e. (center*x,* center*y,* center*z*) and *fovy*. To compute the *frame-varying* parameters through 2D homography transform, at least four feature points in the soccer field are needed for each frame [HaZi2003]. In this chapter, two types of scenes are reconstructed: goalmouth scene and midfield scene. Sections 5.1 and 5.2 describe how to find feature points for a frame of the goalmouth or midfield scene respectively.

When four feature points are found on the ground, we can determine the 2D homography transform given in the following formulae (5.1) (see page 87-91 in [HaZi2003] for more details), which transforms an image point $X' = (x_1', x_z', x_3')$ to a world point $X = (x_1, x_2, x_3)$. Thus we can determine *look-at* and *fovy* easily because determining *look-at* and *fovy* are equivalent to finding the real-world coordinates of three points (176, 144), (176, 0), and (176, 288) in the frame.

$$X = HX' \qquad (4.1)$$

where H is the homography transform matrix.

Feature Point Detection in Goalmouth Scene

The straight-line Hough transform is used to detect the straight lines in the goalmouth scene [KHXK1995]. Once the lines are detected in a goalmouth scene, the crossing points of these lines can be found. These crossing points are the desired feature points for camera calibration as their coordinates in the real world are known.

Feature Point Detection in a Midfield Scene

Here we find four feature points based on the ellipse, which is detected in a robust and fast manner using our algorithm that will be presented in Chapter 5, to compute *look-at* and *fovy*.

In Figure 4.9, A is the position of the camera, B and C are the crossing points between the central line (CL) and the circle, D is the crossing between the upper touch line and the CL, and X is any point on the CL in the real world; b, c, d, and x are the image points of B, C, D, and X respectively. Then they satisfy the following equation as the *cross-ratio* is invariant, which is the most fundamental projective invariant (see page 42 in [HaZi2003] for more details).

$$\frac{\overline{XC}}{\overline{XD}} : \frac{\overline{BC}}{\overline{BD}} = \frac{\overline{xc}}{\overline{xd}} : \frac{\overline{bc}}{\overline{bd}} \qquad (4.2)$$

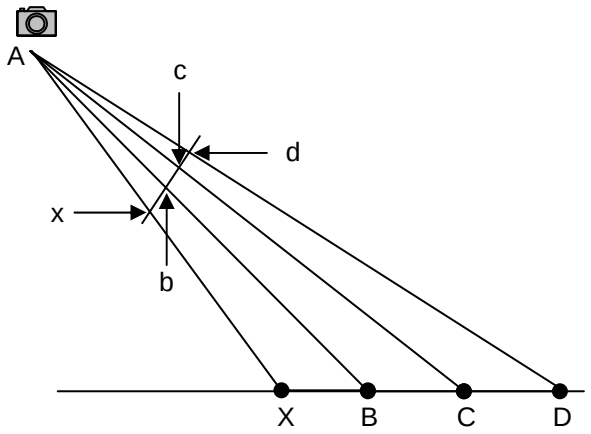


Figure 4.9 The projective transformation of the central line in the soccer field.

If we have detected b, c, and d, we can obtain X for any x on the CL in the image according to the equation (4.2); vise versa we can obtain x for any X on the CL in the real world.

In Figure 4.10, let O be the center of the circle; let U and V be the middle points of BO and OC. In the frame, o, u, and v are not the midpoints of the line segments bc, bo, and oc because of the perspective projection. Hence, we

need the equation (4.2) to compute u and v in the image. Let p (q) be the crossing point between the ellipse and the line that passes u (v) and be perpendicular to the CL in the image. Then b, c, p, and q are four feature points because we know their world coordinates.

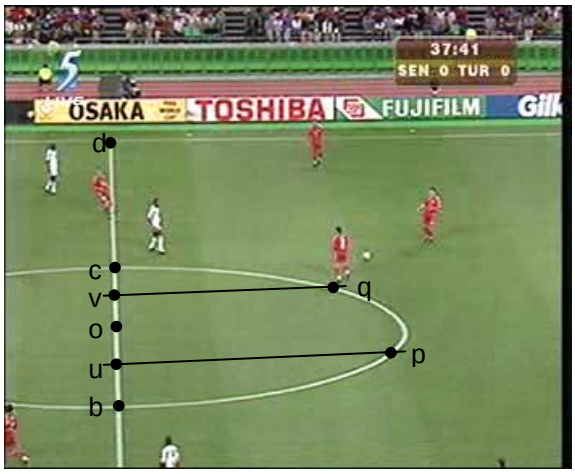


Figure 4.10 A frame with the ellipse and the points involved. Each black dot represents a point involved in the procedure of finding the feature points.

4.7.3 Results of Enhancement and Enrichment

A graphical user interface (GUI) is developed to display the generated video and the illustrations because general media players do not have such function. Figure 4.11 shows two generated and enriched frames of the goalmouth and midfield scenes respectively. In each frame, there are four small info icons around the main viewing window. From top to down, they illustrate the apparent ball speed, the apparent ball direction, the team possession, and the aim of the camera (*or lookat*) respectively. A sequence of generated frames is encoded into video. The generated video then further is decorated with matching music.

Figure 4.11 Two enhanced and enriched frames: one frame of the goalmouth scene and one frame of the midfield scene.

4.8 Summary

By using the computed ball locations, this chapter detected basic actions that may not be detected by the existing approaches. Furthermore, it analyzed the team ball possession, which is another analysis not easily done by the existing approaches. In addition, we improved the play/break analysis and goal detection. In event detection, for each event we built the rule-based model which is derived from the soccer game rules rather than heuristics. The ball trajectory plays an important role in building models as the ball closely correlates relation with game events. Our primary experimental results are very promising. In addition, we developed an enhancement and enrichment system for broadcast soccer video.

Chapter 5

A Robust Ellipse Hough Transform

In this chapter, we present a *robust ellipse Hough transform* (RobustEHT) that is capable of detecting small and partial ellipses. This work was motivated by our need for a robust ellipse detection algorithm as part of the ball-size estimation routines described in Chapter 3. Our new algorithms are based on two main ideas: (1) a new notion of *unbiased measure function* for partial ellipses and small ellipses that make our RobustEHT more robust, and is *normalized* which greatly simplifies the peak detection procedure, and (2) a new *accumulator-free* computation scheme for finding the top *k* peaks of the measure function, without the need of a complex peak detection procedure.

5.1 Introduction

In computer vision and image processing, detecting ellipses in 2D digital images has been widely studied. Ellipses in 2D digital images arise either from actual elliptic objects or from circular objects projected onto 2D images. Ellipse detection has a wide spectrum of applications such as locating an object, computing 3D objects, and recognizing certain manufactured parts.

The Hough transform (HT), first introduced in 1962 by P.V.C. Hough [Hou1962], is a widely used technique for detecting ellipses in digital images [Bal1981, BeBS1999, ChLi2003, GuZa1997, HsHu1990, IlKi1988, ImHT2002,

KHXO1995, Lea1993, Ols1998, PrIK1994, SiDH1984, YoSe1993, YuIK1989]. A Hough transform is called an ellipse Hough transform (EHT) when it is applied to detect ellipse. Like the general Hough transforms, the basic idea of EHT is to gather the evidences of the ellipse occurrences in the Hough space through various voting procedures. The Hough space consists of cells and each cell defines a unique ellipse in the image space. The procedure of gathering evidence transforms the problem of ellipse detection into the problem of detecting the peaks of a *measure function*. In the literature, this function appeared as a *voted accumulator* that has a counter for each cell in the Hough space.

In the past several decades, a large number of EHTs were proposed. These EHTs shared two common key steps: (a) the *voting* procedure used to compute the accumulator, and (b) a *peak detection* procedure for finding local maxima (*corresponding to potential ellipses*) of the voted accumulator. We term the voted accumulator as a *measure function* defined on the Hough space. We further term the measure functions used by most of the existing EHTs as the *absolute measure functions* (AMFs) because they measure a cell by the absolute vote count of the cell. An AMF is biased against the *small* ellipses and *partial* ellipses as these ellipses possess fewer sample points than large perfect ellipses, respectively. The bias against small ellipses is eliminated if the AMF is *normalized* by the visible length of the cell considered as observed in [Kul1979, Dav1992b]. However, the bias against the partial ellipses is *not* eliminated by this normalization.

This chapter first gives an introduction to the standard EHT (SEHT) and the combinatorial EHT (CEHT) as well as their improvements (*or variants*).

Then it presents a robust EHT (RobustEHT), which is robust for detecting the partial ellipses and the small ellipses in the presence of different sizes of ellipses. To correctly handle partial ellipses, we define the *unbiased measure function* (UMF) by considering all "significant arcs" of the partial ellipse in the original image. Thus, the RobustEHT is more robust when it is applied to the detection of partial ellipses in noisy images. In addition, the UMF is a normalized measure function and so it greatly simplifies the peak detection procedures. The slight penalty with the use of the UMF is an increase in the computational complexity associated with the need to find the significant arcs of each cell.

We remark here, that there are several ways to control this increase in the computational complexity by first using a procedure to produce a tighter Hough space. However, our focus in this research work is on developing a robust ellipse Hough transform and thus we pay less effort on computational time in this chapter.

In computing the UMF, our algorithm considers each cell in ellipse Hough space one by one. Suppose that we are given an estimate (say k) of the number of ellipses to be detected. Since the measure function is normalized, the RobustEHT algorithm only needs to keep a small number of "peak-values" cells (cells with maximal measures). Our accumulator-free algorithm is better than that of [ImHT2002] since the computational complexity for computing the UMF is the same as that for computing the measure function for the SEHT.

The RobustEHT has been tested on the synthesized frames and the real data (broadcast soccer video) and compared with the SEHT. Experiment

results strongly demonstrate that the RobustEHT is better than the SEHT as well as other existing EHTs in robustness.

5.2 An Introduction to Ellipse Hough Transforms

This section describes the existing EHTs organized according to their voting ways. Section 5.2.1 first gives the formal definition of the EHT. Then, we describe the standard EHT and the combinatorial EHT respectively. Lastly, we comment on the existing measure functions and EHTs.

5.2.1 Definition of Ellipse Hough Transform

The general procedure of ellipse detection is as follows. Let F be the given digital 2D image, or image in short. Assume that a procedure has already been chosen to produce the sample points, which are edge points in most of the literature. The first step in the algorithm of ellipse detection is to obtain sample points using the chosen procedure. Then, it detects the presence of ellipses and finds the parameters of ellipses in F from the sample points computationally. Particularly, an EHT transforms a given frame into a *measure function* defined on the Hough space. Thus, the problem of ellipse detection is changed into the problem of function analysis. Here, we give a formal definition of the EHT.

Definition 5.1 Let F be an image and P_1, P_2, and P_3 be three procedures. Then $(M(\bullet); P_1, P_2, P_3)$ is called as an ellipse Hough transform (EHT) if

a. P_1 can form the Hough space H of any given frame F;

b. $M(c) \geq 0$, for any $c \in H$;

c. P_2 can compute all values of $M(\bullet)$ on H;

d. P_3 can find some cells in H according to $M(\bullet)$. The found cells are considered as the cells corresponding to the wanted ellipses in F.

In most of the papers on the EHTs, the measure functions were computed by various voting procedures. In a hidden way, each voting procedure actually defines a measure function. There are two types of voting procedures:

- *one-to-many*: In a procedure of this type, each sample point votes for all the cells that might have produced the point. The standard EHT (SEHT), described by D. H. Ballard in 1981, and its improvements, use this type of voting procedures [Bal1981, BeBS1999, ChLi2003, GuZa1997, Hou1962, HsHu1990, IlKi1988, ImHT2002, KHXO1995, Lea1993, Ols1998, PrIK1994, SiDH1984, YuIK1989].
- *five-to-one*: In a procedure of this type, each combination of 5 sample points votes for the cells determined by the 5 points. The combinatorial EHT (CEHT), proposed by D. Ben-Tzvi and M. B. Sandler in 1990, and its variants, use this type of voting procedures [XuOK1990, XuOj1993, Lea1993, KHXO1995].

5.2.2 Standard Ellipse Hough Transform

The representative of the one-to-many type of the EHTs is the SEHT. Hence, we first examine how the SEHT detect ellipses. In the Cartesian coordinate

system, each ellipse in an image can be represented by a 5-parameter cell (x, y, a, b, θ), where (x, y) is the center, a and b are the major and minor axes respectively, θ is the tilt degree, of the ellipse. All the ellipses in the image consist of a 5-parameter family. The SEHT needs to determine the Hough space that comprises all the ellipses in the image in its initialization. This determined Hough space H can be expressed as follows.

$$H = \bigcup_{i=1}^{k} S_i. \qquad (5.1)$$

where $S_i = \{(x, y, a, b, \theta): x^i_{min} < x < x^i_{max},\ y^i_{min} < y < y^i_{max}, a^i_{min} < a < a^i_{max},\ b^i_{min} < b < b^i_{max},\ \theta^i_{min} < \theta < \theta^i_{max}\}$ for $i = 1$ to k.

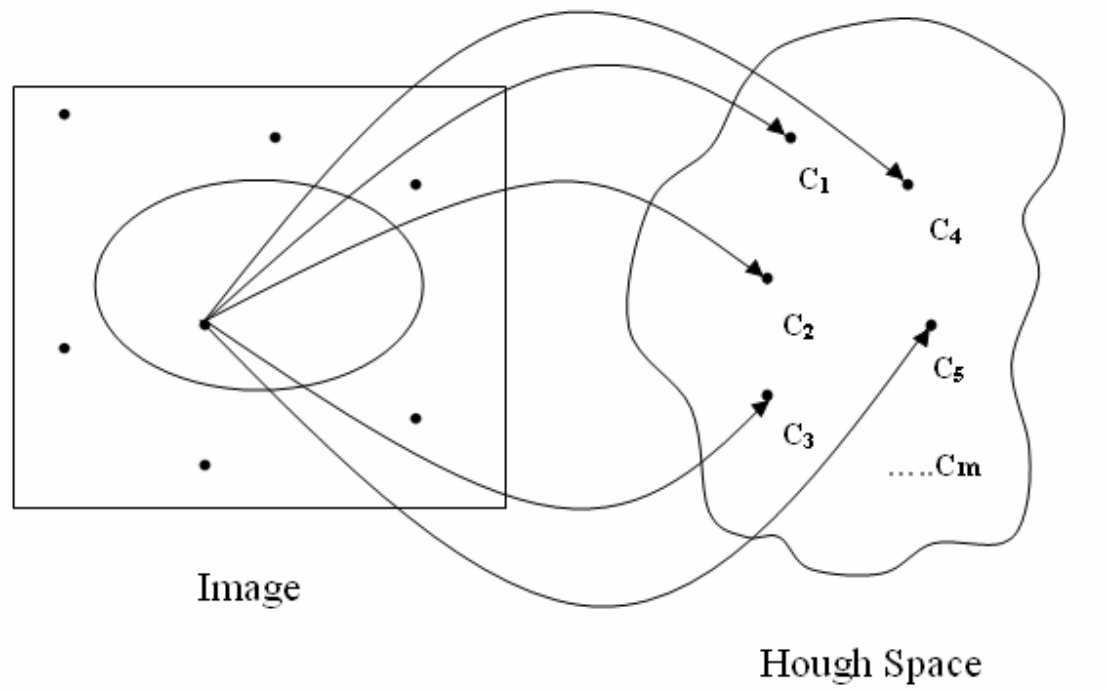


Figure 5.1 Illustration of voting way of the standard ellipse Hough transform.

The main part of a SEHT is the procedure of gathering the evidence of ellipse. Every sample point votes for the cells that might have produced the sample point illustrated in Figure 5.1. Let F be the considered image and H be its Hough space. Let $P = \{p_1, p_2, p_3, ..., p_n\}$ be all sample points in F. Let

$p = (u, v)$ be a sample point and $c = (x, y, a, b, \theta) \in \mathrm{H}$ be a cell, we say that *p* votes for *c* if *p* meets the following equation.

$$\frac{(u \cdot \cos\theta + v \cdot \sin\theta - x)^2}{a^2} + \frac{(-u \cdot \sin\theta + v \cdot \cos\theta - y)^2}{b^2} = 1. \qquad (5.2)$$

Then, the voting function $V(p, c)$ on $\mathrm{P} \times \mathrm{H}$, where $p \in \mathrm{P}$ and $c = (x, y, a, b, \theta) \in \mathrm{H}$, can be defined as follows

$$V(p, c) = \begin{cases} 1, & \text{if } p \text{ votes for } c; \\ 0, & \text{otherwise.} \end{cases} \qquad (5.3)$$

Voting function determines the particular SEHT being implemented. In general, this function is defined implicitly through the choice of a parameterization and a quantization of the Hough space [PrIK1994]. Let $M_s(c)$ be the final count of cell *c* after voting procedure has terminated. We name $M_s(\bullet)$ as *the standard measure function* (SMF) defined on the Hough space H, which maps each cell of the Hough space into the final count of the cell in the accumulator array. Then, we have the following formula

$$M_s(c) = \sum_{i=1}^{n} V(p_i, c) \qquad (5.4)$$

Once the voting procedure has finished, the SEHT estimates the presence and location of the local peaks of the SMF. Thus, the SEHT obtains the cells corresponding to the potential ellipses in the given image. Many ways of speeding up the voting procedure were studied since the voting procedure of the SEHT is very time-consuming.

- **Multistage Hough Transform:** Multistage EHTs decompose the problem of detecting five parameters of ellipse into several stages. Each stage detects some of the five parameters. Hence, the total number of voting is reduced as finding fewer parameters can be done by voting on the lower

dimension Hough space [MuNi1991, GuZa1997, Dav1989, LeWo1999, XiJi2002, KaOh2002].

- **Voting Group by Group**: C. F. Olson [Ols1998, Ols1999] speeded up the voting by grouping the edge points. This method votes group by group instead of one by one so the total number of voting is reduced.
- **Probabilistic Voting**: Kiryati et al [KiEB1991] proposed the Probabilistic HT (PHT) to speed up the SEHT. The PHT uses a small, randomly selected subset of the edge points in the image to do voting. Because a small subset can perform a good voting, the voting time can be reduced considerably.
- **Coarse-to-Fine Voting**: The fast HT by Li et al [LiLL1986], the adaptive HT by Illingworth and Kittler [IlKi1987], the multi-resolution HT by M. Atiquzzaman [Ati1992] all used coarse-to-fine voting. Their method detects shape including ellipse in a rough to fine resolutions. In both rough and fine resolutions, the Hough spaces are small so that the time for voting is reduced significantly. Note that coarse-to-fine voting can be involved in both image and Hough spaces.
- **Parallel Implementation**: Ben-Tzvi et al [BeNS1989] speeded up the voting procedure with parallel implementation.

5.2.3 Combinatorial Ellipse Hough Transform

The simple form of the EHTs in the five-to-one class is the combinatorial Hough transform proposed by V. F. Leavers et al [LeBS1989] and Ben-Tzvi and Sandler [BeSa1990], which is based on the fact that a single cell that has

n parameters can be determined uniquely with n feature points from the image. Particularly, in ellipse detection each 5-point votes the cell determined by them, which is illustrated in Figure 5.2.

We concisely describe how the combinatorial EHT (CEHT) works. Let F be the given image. Let $P = \{p_1, p_2, p_3, ... p_n\}$ be all sample points in F. Let $R = \{r_1, r_2, r_3, ... r_m\}$ be all 5-point combinations that can determine a cell (*ellipse*). Then, the voting function $V(r, c)$ on $R \times H$, where $r \in R$ and $c = (x, y, a, b, \theta) \in H$, can be defined as follows

$$V(r, c) = \begin{cases} 1, & \text{if } r \text{ determines } c; \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

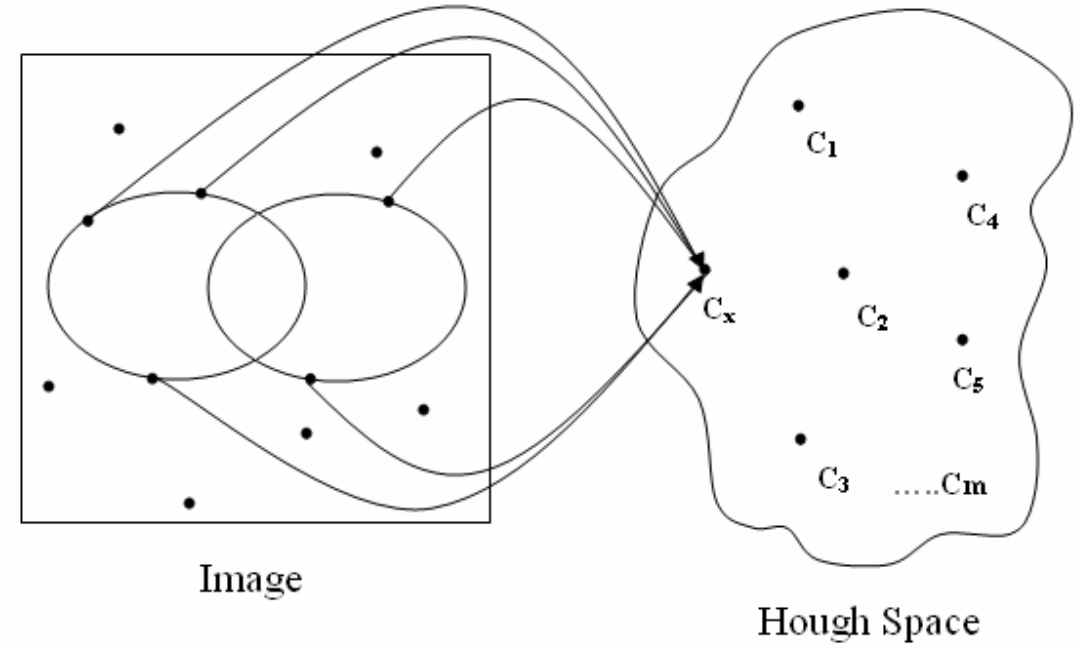


Figure 5.2 Illustration of voting way of the combinatorial ellipse Hough transform.

Let $M_c(c)$ be the final count of cell c after the voting procedure has terminated. We name $M_c(\bullet)$ as *the combinatorial measure function* (CMF) defined on the Hough space H, which maps each cell of the Hough space into

the final count of the cell in the accumulator array. Then, we have the following formula

$$M_c(c) = \sum_{i=1}^{n} V(r_i, c) \qquad (5.6)$$

Once the voting procedure has finished, the CEHT analyzes the CMF to estimate the presence and location of the local peaks of the CMF.

Like the SEHT, the CEHT is very slow in voting because normally the number of 5-point combinations is very large. To reduce the computational time, a lot of improved CEHTs were proposed. For example, Xu et al [XuOK1990] proposed the Random HT. The REHT is run long enough to detect a global maximum of the measure function *dynamically*. The REHT does not select the subset before voting. Instead, it selects and votes in the same time and stop the voting when one cell meets the predefined criteria. The REHT actually computes an approximate proportional function of $M_c(\bullet)$. In recent years, many improvements have been done to the CEHT and the REHT. Some of them are the Dynamic HT [Lea1992], which combines the technique of randomly selecting edges and the connectivity detection of edges, the improved implementations of the RHT [XuOj1993], the Connective HT [Yue1991], which reduces the computational complexity of the Dynamic Combinatorial HT [LeBS1989].

5.2.4 Comments on the Existing Hough Transforms

As we have seen above, the existing ellipse Hough transforms defined two types of measure functions: $M_s(\bullet)$ as well as its variants and $M_c(\bullet)$ as well as its variants. Generally speaking, the approximate proportional measure

functions used by probabilistic ellipse Hough transforms are not as robust as the SMF and the CMF. Hence, the SEHT and the CEHT probably are two of the most robust EHTs in the literature.

Another key factor to determine the performance of an EHT is how to compute the measure function. Most of the existing variants of the EHTs employ the voting procedures to compute the measure functions. Differences among the voting procedures are in two aspects. One difference is whether they are one-to-many voting or many-to-one voting, which are determined by the used measure function. The other difference lies in the data structures for recording the voting results, among others, bin-structure [LiLa1986], fixed accumulator array [Bal1881], dynamic link [Lea1992a], etc. All the voting procedures share a common voting direction which is from sample points in image to the Hough space, illustrated in Figure 5.1 and 5.2.

In general, there are three issues for the existing EHTs.

- Firstly, they are *not robust* in detecting either the small ellipses or the partial ellipses since the AMF is biased against both the small ellipses and the partial ellipses. An AMF is fair only when all the target ellipses are nearly complete and there is not any big difference in their perimeters.
- Another issue is that the existing EHTs require *a large amount of memory* for the accumulator. Imiya et al [ImHT2002] proposed an EHT without accumulator. Their basic idea is that a target shape must be made of a subset of sample points. The simple implementation of this idea needs to consider all the subsets of the sample points. They partitioned the image and consider the subsets of the sample points in a sub-image to reduce the number of the subsets. Their method overcomes the memory

requirement issue, but at the expense of greatly increasing the computational complexity.

- Thirdly, peak detection remains problematic. The peaks vary in magnitudes so a uniform criterion cannot be used for peak detection.

5.3 Our Proposed Robust Ellipse Hough Transform

5.3.1 Definitions and Notations

We define EHTs using set-theoretic notations and our definition can be considered to be an extension of the sets defined in [HaAn1997] for the straight line HT. We believe that this gives a more rigorous treatment to the EHT than the voting function and the accumulator.

Let H be the Hough space. For any $c \in H$, we can draw an ellipse defined by *c* on the image F illustrated in Figure 5.3. Note that the part of the ellipse defined by *c* may lie outside the given image as illustrated in Figure 5.3.

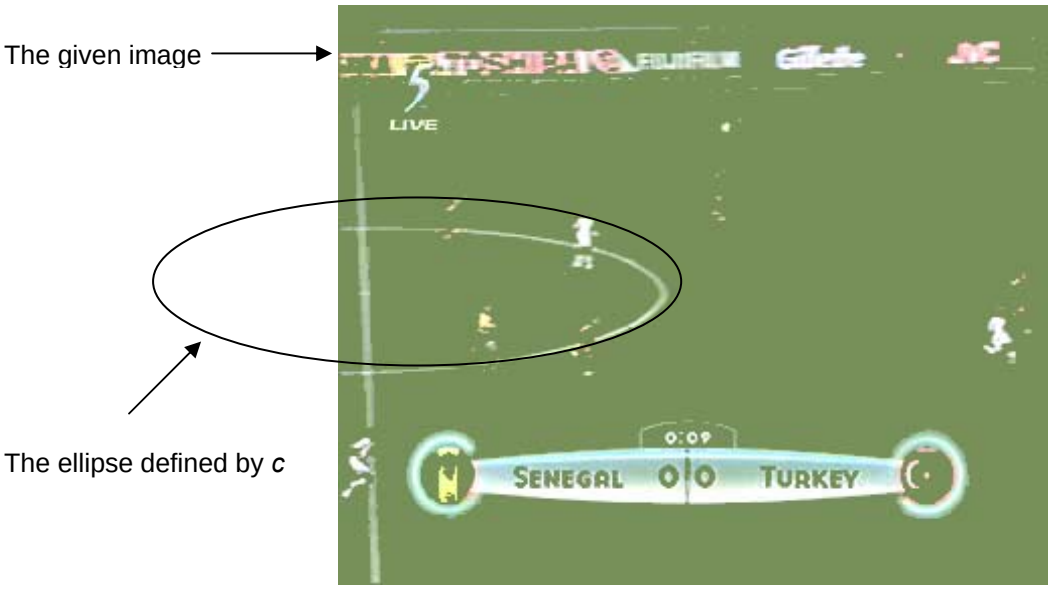
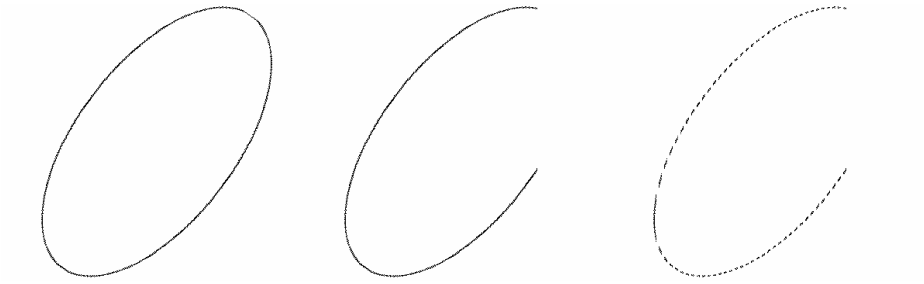


Figure 5.3 A sample image of broadcast soccer video and an ellipse defined.

Definition 5.2 Let F be the given image and $c \in H$. A point *p* in the image space is called *the base point* of *c* if *p* is on the ellipse defined by *c* regardless of whether *p* is in the given image. We define the following three notations.



a. The ideal support of sample cell *c*.

b. The support of sample cell *c*.

c. The voting support of sample cell *c*.

Figure 5.4 A cell *c* of the Hough space, its ideal support $\Theta(c)$, support $\Re(c)$ and voting support $\Omega(c)$.

a. The set $\Theta(c)$, called *the ideal support,* comprises all the base points of *c*;

b. The set $\Re(c)$, termed *the support*, is given as follows.

$$\Re(c) = F \bigcap \Theta(c) \tag{5.7}$$

c. The set $\Omega(c)$, termed *the voting support*, is given as follows.

$$\Omega(c) = \{p : p \in P \ \& \ V(p, c) = 1\} \tag{5.8}$$

The sample $\Theta(c)$, $\Re(c)$, and $\Omega(c)$ are showed in Figure 5.4. The support and the voting support defined for ellipse in this chapter are equivalent to the support and the voting support defined for straight line in [HaAn1997]. However, we have also introduced the concept of the *ideal support* which is valid for closed shapes such as ellipses.

We also note that $M_s(c) = |\Omega(c)|$ for each cell $c \in H$. Thus, for each cell in the Hough space, we explicitly define the evidence set $\Omega(c)$, while the voting procedure only computes its size.

5.3.2 Measure Function Normalization

In the SEHT, the values of $M_s(\bullet)$ differ greatly between the small ellipse and the large ellipses. Let c_1 and c_2 be two cells in the Hough space H. Assume that they correspond to two complete ellipses in the given image F. However, $M_s(c_1) << M_s(c_2)$ if $|\Theta(c_1)| << |\Theta(c_2)|$. This shows that $M_s(c)$ cannot fairly measure the small and large ellipses. In other words, it cannot indicate how likely cell c corresponds to an ellipse. To fairly measure the ellipse with the different sizes, we define *the normalized measure function* $N(\bullet)$ to replace *the standard measure function* $M_s(\bullet)$.

Definition 5.3 (**NMF**) The following $N(c)$ defined on H is called *the normalized measure function* (NMF).

$$N(c) = \frac{M_s(c)}{|\Theta(c)|} \tag{5.9}$$

For the small shapes defined by c, $N(c)$ may be sensitive to noise. To handle this problem, one way is to use the threshold for $N(c)$ that varies with the size of the measured shape and the noise level in the input image.

Proposition 5.2 $N(c) = 1$ if cell c corresponds to a complete ellipse in the given image.

According to the above discussion, we know that $N(\bullet)$ can measure cells better than $M_s(\bullet)$. With $N(\bullet)$ we need to find only the global peaks instead of the local peaks of $M_s(\bullet)$. Thus, we form a *normalized EHT* (NEHT) for detecting ellipses from a given image as follows.

Algorithm 5.1 (**Normalized Ellipse Hough Transform**)

The input is an image and the output is a set of ellipses.

1. (*Initialization*)

   Find all sample points $\mathrm{P} = \{p_1, p_2, p_3, \ldots p_n\}$;

   Determine the Hough space H.

2. (*Voting*)

   For each $p \in \mathrm{P}$ and for each $c \in \mathrm{H}$, $\mathrm{M}_s(c) = \mathrm{M}_s(c) + \mathrm{V}(p_i, c)$.

3. (*Normalizing*)

   For each $c \in \mathrm{H}$, $\mathrm{N}(c) = \dfrac{\mathrm{M}_s(c)}{|\Theta(c)|}$.

4. (*Peak Detecting*)

   Find the global peaks of $\mathrm{N}(c)$ to obtain the parameters of the ellipses occurring in the given image.

5.3.3 Accumulator-Free Computation Scheme

Unlike the conventional voting procedures, we use a procedure to compute N(*c*) in the direction from the Hough space to the image. This procedure computes cells in the Hough space one by one and more details of this procedure is described below. Let *c* be the considered cell in the Hough space. We first compute the ideal ellipse in the image defined by *c*. Then, we find all feature points in the computed ideal ellipse and count the number of the found feature points. In another word, we produce $\mathrm{M}_s(c)$. Finally, we convert $\mathrm{M}_s(c)$ into N(*c*) by dividing $\mathrm{M}_s(c)$ by the cardinal number of the *ideal support*. Based on the above discussion, we have the following the Accumulator-Free EHT (AFEHT).

Algorithm 5.2 (Accumulator-Free Ellipse Hough Transform)

The input is an image and the output is a set of ellipses.

1. (*Initialization*)

   Find all sample points $\mathrm{P} = \{p_1, p_2, p_3, \ldots p_n\}$;

   Determine H, the Hough space of the given image.

   Determine *k*, the upper-bound of the number of ellipses in F;

   Create and initialize a list L with *k* elements;

2. (*Measuring each cell*)

   For each $c \in \mathrm{H}$, do:

   Compute $\mathrm{M}_s(c)$ on $\Re(c)$ for the current cell c.

   Convert $\mathrm{M}_s(c)$ into $\mathrm{N}(c)$.

   Replace the element of the smallest value in L with $\mathrm{N}(c)$ if it is larger than the smallest value in L and the distance from *c* is to the other cells in L is larger than a threshold.

3. (*Selection*)

   Select the elements with the largest values from L according to the given threshold.

5.3.4 Unbiased Measure Function for Partial Ellipses

To the best of our knowledge, no work has been done to modify the SMF to facilitate detection of partial ellipses. To improve the robustness of detecting partial ellipses, we modify the SMF and the NMF to benefit detecting partial ellipses. As a result, we form a *robust accumulator-free EHT* (RobustEHT). It can robustly detect partial ellipses from the given image simultaneously.

To measure partial ellipses better, we identify the significant arcs of the ellipse defined by the considered cell. In other words, we search along the ellipse to find the arcs. To do so, we define *an unbiased measure function* (UMF). To define the UMF, we prepare several notations. Let F be the given image. For any $c \in \mathrm{H}$, let E(*c*) be the ellipse defined by *c*. Let $\overrightarrow{OX}$ be the ray from the center to its farthest right point on E(*c*). Let $\overrightarrow{Op}$ be the ray from the

center to p on E(c). We use $\angle(p,\ c)$ to represent the angle from $\overrightarrow{OX}$ to $\overrightarrow{Op}$ in the considered ellipse, which $\angle(p,\ c)$ is illustrated in Figure 5.5.

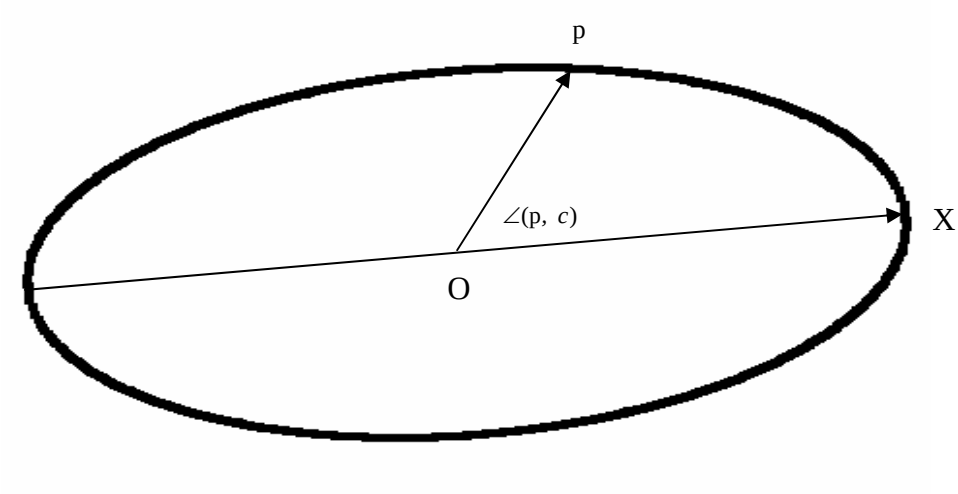


Figure 5.5 The ellipse defined by c and a sample angle $\angle(p,\ c)$ on it.

Definition 5.4 Assume that the set of arcs $\partial = \bigcup_{i=1}^{k}(\alpha_i, \beta_i)$ such that $0 \le \alpha_1 < \beta_1 < \alpha_2 < \beta_2 < .... < \alpha_k < \beta_k \le 2\pi$ and $\beta_i - \alpha_i > \delta$ for $i = 1, 2, ..., k$, where δ is a predefined threshold. $\overleftrightarrow{\Re}[\partial, c]$, called *the arc potential set,* is given as follows.

$$\overleftrightarrow{\Re}[\partial, c] = \{p:\ \ p \in \Re(c)\ \&\ \angle(p, c) \in \partial\} \tag{5.10}$$

Definition 5.5 For each cell $c \in H$, its value of U(c), termed as *the unbiased measure function* (UMF), is given as follows

$$U(c) = \max_{\partial}\left\{\frac{1}{|\Theta(c) \cap \partial|}\left(\sum_{i=1 \text{to} n}^{p_i \in \overleftrightarrow{\Re}[\partial, c]} V(p_i,\ c)\right)\right\} \tag{5.11}$$

The UMF, the NMF, and the SMF all have different values when we measure a partial ellipse. Figure 5.6 illustrates these different values.

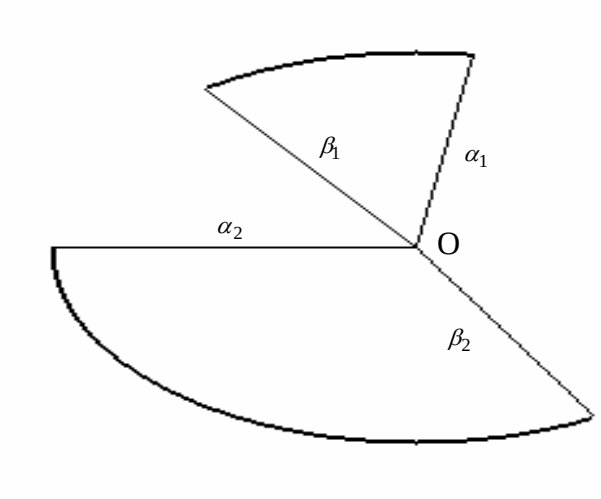


Figure 5.6 A sample partial ellipse. Its center is at O. Its two arcs are (α_1, β_1) and (α_2, β_2). Its values of the SMF, the NMF, and the UMF are 683, 0.495, and 1 respectively.

Algorithm 5.3 (Robust Accumulator-Free Ellipse Hough Transform)

The input is an image and the output is a set of ellipses.

1. (*Initialization*)

   Find all sample points $\mathrm{P} = \{p_1, p_2, p_3, \ldots p_n\}$;

   Determine H, the Hough space of the given image.

   Determine *k*, the upper-bound of the number of ellipses in F;

   Create and initialize a list L with *k* elements;

2. (*Measuring each cell*)

   For each $c \in \mathrm{H}$, do:

   Compute $\mathrm{I}(c)$ on the base set of the current cell c.

   Replace the element of the smallest value in L with $\mathrm{I}(c)$ if it is larger than the smallest value of L and the distance from *c* to the other cells in L is larger than a threshold.

3. (*Selection*)

   Select the elements with the largest values from L according to the threshold.

Comment 5.1 We should set proper thresholds for δ and $|\partial|$ in Definitions 5.4 and 5.5 in implementing the above algorithm. Otherwise, the unbiased measure function $U(c)$ would be very sensitive to noises.

In practice, we compute $U(c)$ as follows. First we arrange all points in $\Theta(c)$ in a list L in ascending order, according to $\angle(p, c)$ and this creates a corresponding binary list R. For ith point of L, R[i] = 1 if this point is a sample point; R[i]=0 otherwise. Then, for list R we search for all significant ellipse arcs. Last, we compute $U(c)$ on these found segments.

5.4 Samples and Experiment Results

Firstly we use synthesized samples to explain the merits of the NMF over the SMF and the UMF over both the SMF and the NMF. Then, we compare the robustness of the RobustEHT with the robustness of EHTs using the SMF or the NMF including the SEHT (the algorithm in [Bal1881]), the NEHT (Algorithm 5.1), and the AFEHT (Algorithm 5.2) in detecting partial ellipses from broadcast soccer video (BSV). Before comparing the robustness of the algorithms, we will describe the framework for detecting ellipses from BSV. This framework mainly has two components in detecting the ellipse from each frame of BSV. One is the estimation component, which estimates the locations and sizes of the target ellipses using image processing techniques and domain knowledge of soccer video. The other is the search component, which uses the various EHTs (SEHT, NEHT, AFEHT, and RobustEHT) to search the best ellipse around the estimated ellipses.

5.4.1 Synthesized Samples

Synthesized sample one: We synthesize an image called F_1 with the resolution of 400X300 shown in Figure 5.7 to show the merits of the NMF over the SMF in measuring cells. Assume that all black points are sample points.

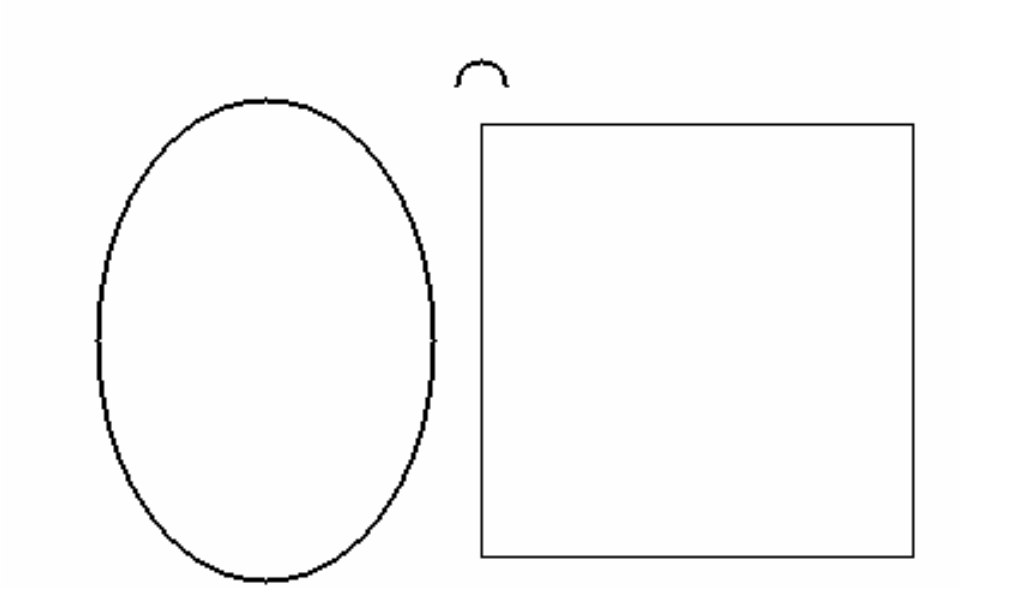


Figure 5.7 A synthesized binary image of an ellipse, a half circle, and a square.

Let $\{(x, y, a, b, \theta) : 0 < x, y < 300; 8 < b \le a < 200; 0 < \theta < 2\pi\}$ be the determined Hough space H. We consider three cells in H, $c_1 = (38, 144, 100, 70, \pi/2)$, $c_2 = (80, 152, 10, 10, 0)$, $c_3 = (278, 144, 90, 90, 0)$. F_1 comprises three objects O_1, O_2, and O_3. O_1 is a complete ellipse defined by c_1. O_2 is semicircle corresponding to c_2. O_3 is a square, which works as noise, and which c_3 corresponds to the circle that covers most of its points of this square. Let H be its Hough space for detecting ellipses. The values of c_1, c_2, and c_3 in $M_s(\bullet)$ and $N(\bullet)$ are shown in Table 5.1.

Table 5.1 Values of $M_s(\bullet)$ and $N(\bullet)$ on $c_1, c_2,$ and c_3 for F_1.

	c_1	c_2	c_3
# of Base Set	980	116	1020
Values of $M_s(\bullet)$	980	60	108
Values of $N(\bullet)$	1.00	0.517	0.106

In Table 5.1, $M_s(c_3)$ is larger than $M_s(c_2)$ but c_2 corresponds to a small half circle while c_3 does not correspond to any ellipse. Hence, $M_s(\bullet)$ cannot obtain perfect candidates if the ellipse candidates are selected simply by choosing those cells being larger than a global threshold against the SMF. In contrast, perfect candidates can be obtained by setting a threshold with value 0.20 against the NMF.

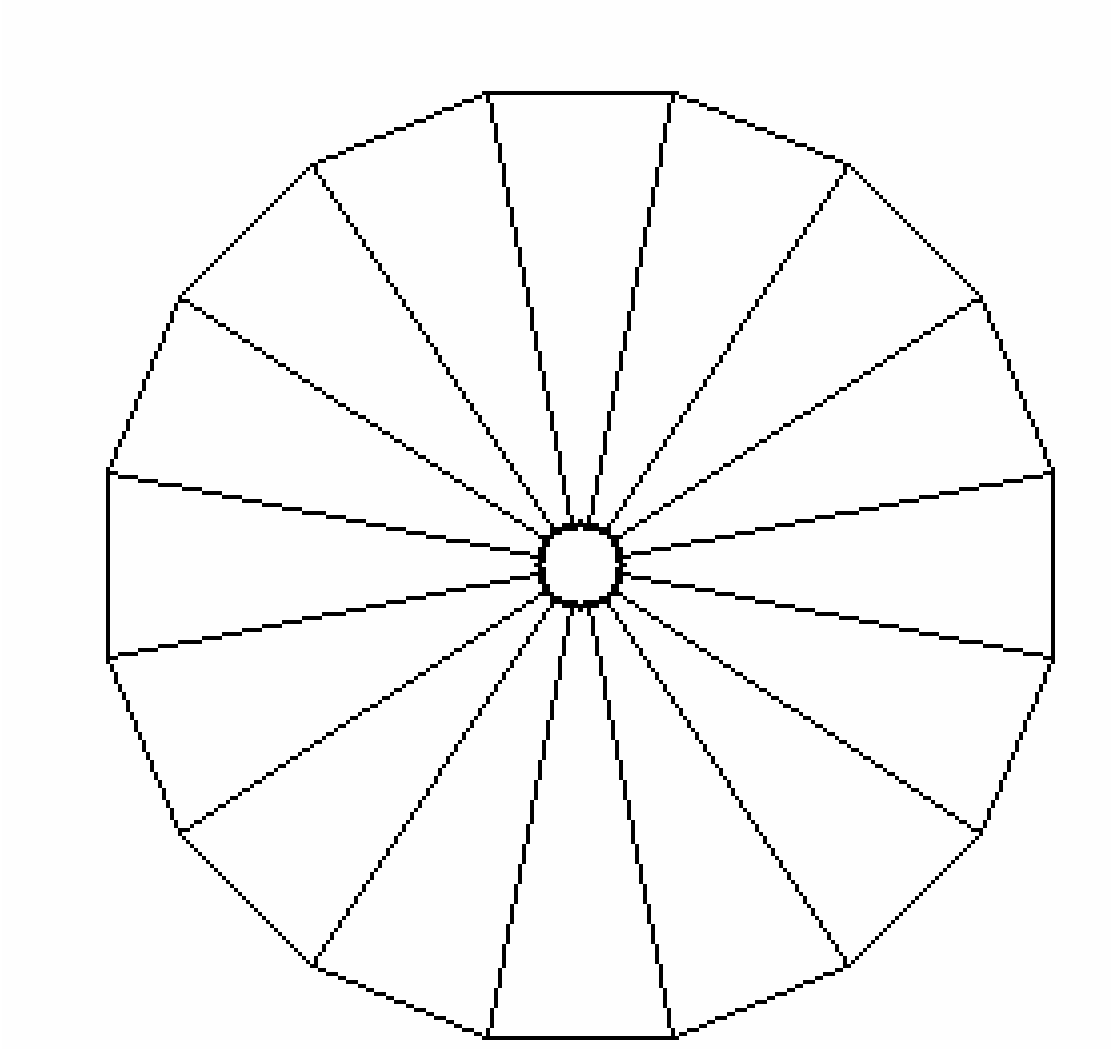


Figure 5.8 A circle and a hexadecagon centered at (144, 144) with 16 line segments linking them.

Synthesized sample two: We synthesize another image called F_2 shown in Figure 5.8 to show merits of the NMF over the SMF and the UMF over both the NMF and the SMF. The problem is to detect circles from F_2.

Let $\{(x, y, r) : x = 144;\ y = 144;\ 8 < r < 122\}$ be the determined Hough space H. In F_2, a circle centers at (144, 144) with radius = 8; a hexadecagon centers at (144, 144) too and the length between one of its vertical points and its center is 121; 16 evenly-distributed line segments link the vertical points of the hexadecogon and the circle. Assume that all black points in F_2 are sample points. All points except the points on the circle work as noise. All values of the SMF $M_s(\bullet)$, the NMF $N(\bullet)$ and the UMF $U(\bullet)$ for the image are given in Table 5.2. Table 5.2 shows that by setting a global threshold $M_s(\bullet)$ cannot generate perfect circle candidates, but both $N(\bullet)$ and $U(\bullet)$ can. However, $U(\bullet)$ has a better value pattern to identify peaks, i.e. the peaks in $U(\bullet)$ are sharper than the ones in $N(\bullet)$.

Table 5.2 Partial values of $M_s(\bullet)$, $N(\bullet)$, and $U(\bullet)$ for F_2.

The values of the different measure functions on the hypothesized circles centered at (144, 144) with the various radii															
R	8	9	10	11	12	13	14	…	116	117	118	119	120	121	122
$M_s(\bullet)$	0	44	116	76	32	32	32	…	32	76	444	504	216	32	0
$N(\bullet)$	0.0	0.4	1.0	0.6	0.2	0.2	0.2	…	0.0	0.1	0.3	0.4	0.2	0.0	0.0
$U(\bullet)$	0.0	0.0	1.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Synthesized sample three: We synthesize another image called F_3 with the size of 640X300 shown in Figure 5.9 to show merits of the UMF over both the SMF and the NMF. The problem is to detect circles from F_3.

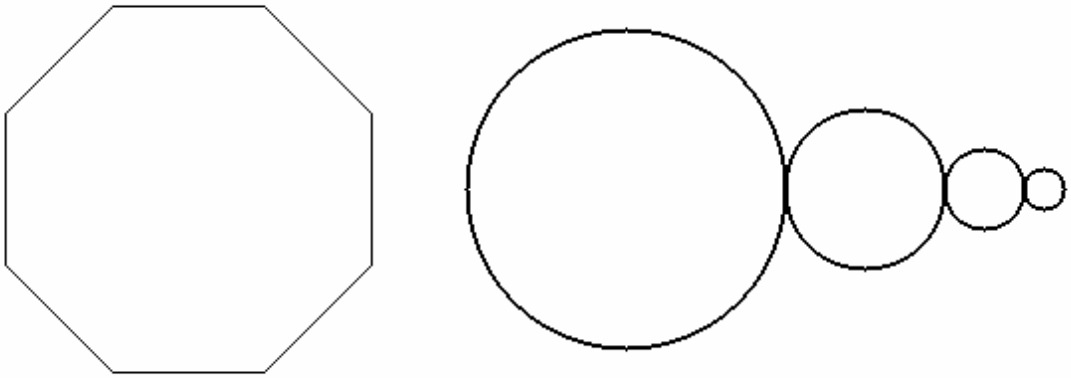


Figure 5.9 A hexagon and four circles with the various radii.

Let us consider the following seven cells in the Hough space H, which have the large values when they are measured by the SMF and the NMF. The seven cells are: c_1 = (120, 150, 92), c_2 = (120, 150, 93), c_3 = (120, 150, 94), c_4 = (340, 150, 80), c_5 = (460, 150, 40), c_6 = (520, 150, 20), c_7 = (550, 150, 10).

Table 5.3 Partial values of $M_s(\bullet)$, $N(\bullet)$, and $U(\bullet)$ for F_3.

Cells	c_1	c_2	c_3	c_4	c_5	c_6	c_7
# of base set	1044	1052	1060	908	452	228	116
$M_s(\bullet)$	176	216	128	908	452	228	116
$N(\bullet)$	0.17	0.21	0.12	1.00	1.00	1.00	1.00
$U(\bullet)$	0.00	0.00	0.00	1.00	1.00	1.00	1.00

In F_3, a hexagon centers at (120, 150); four circles are defined by c_4, c_5, c_6, c_7 respectively. The hexagon works as noise. All values of the SMF $M_s(\bullet)$, the NMF $N(\bullet)$ and the UMF $U(\bullet)$ for the image are given in Table 5.3. Table 5.3 shows that the values of $M_s(\bullet)$ for different size perfect circles have big differences. However, both $N(\bullet)$ and $U(\bullet)$ have the same values for different size perfect circles. Hence, we can identify the perfect circles using either $N(\bullet)$ or $U(\bullet)$.

Synthesized sample four: We synthesize another image called F_4 with the size of 640X300 shown in Figure 5.10 to show merits of the UMF over both the SMF and the NMF in detecting the partial circles. The problem is to detect partial circles from F_4.

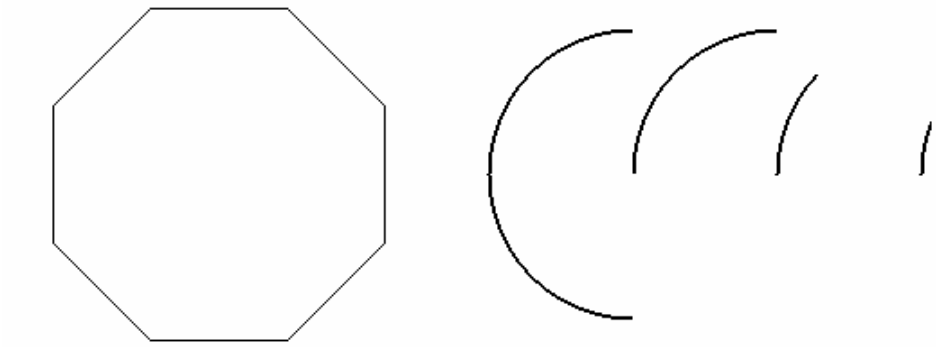


Figure 5.10 A hexagon and four arcs of circles with the same radius and various lengths of arcs.

Let us consider the following seven cells in the Hough space H, which have the large values when they are measured by the SMF and the NMF. The seven cells are: c_1 = (120, 150, 92), c_2 = (120, 150, 93), c_3 = (120, 150, 94), c_4 = (350, 150, 80), c_5 = (430, 150, 80), c_6 = (510, 150, 80), c_7 = (590, 150, 80). In F_4, a hexagon centers at (120, 150); four arcs of the different circles are defined by c_4, c_5, c_6 and c_7 respectively. The hexagon works as noise. All the values of the SMF $M_s(\bullet)$, the NMF $N(\bullet)$ and the UMF $U(\bullet)$ for the image are given in Table 5.4. Table 5.4 shows that the values of $M_s(\bullet)$ and $N(\bullet)$ are different for four arcs. However, $U(\bullet)$ have the same values for four arcs. Hence, we can easily identify the partial circles using $U(\bullet)$.

Table 5.4 Partial values of $M_s(\bullet)$, $N(\bullet)$, and $U(\bullet)$ for F_4.

Cells	c_1	c_2	c_3	c_4	c_5	c_6	c_7
# of base set	1044	1052	1060	908	908	908	908
$M_s(\bullet)$	176	216	128	450	225	112	58
$N(\bullet)$	0.17	0.21	0.12	0.45	0.25	0.12	0.06
$U(\bullet)$	0.00	0.00	0.00	1.00	1.00	1.00	1.00

Synthesized sample five: We synthesize another image called F_5 with the size of 800X600 shown in Figure 5.11 to show merits of the UMF over both the SMF and the NMF in detecting ellipses in a complex image. F_5 comprises nine full ellipses and four partial ellipses and they overlap one another.

Let us consider the following twenty four cells in the Hough space H, which have the large values measured by the SMF. The twenty four cells are: nine cells corresponding to full ellipses c_1 = (200, 400, 150, 50, 0), c_2 = (350, 400, 150, 50, 0), c_3 = (275, 300, 100, 100, 0), c_4 = (530, 400, 100, 100, 0), c_5 = (500, 170, 120, 40, 0.75π), c_6 = (150, 200, 80, 80, 0), c_7 = (270, 200, 40, 40, 0), c_8 = (330, 200, 20, 20, 0), c_9 = (360, 200, 10, 10, 0); three cells corresponding to partial ellipses c_{10} = (550, 200, 80, 80, 0), c_{11} = (630, 200, 80, 80, 0), c_{12} = (710, 200, 80, 80, 0); thirteen cells not corresponding to any ellipse but they have large values of the SMF c_{13} = (126, 405, 76, 45, 0), c_{14} = (217, 370, 37, 20, 0), c_{15} = (264, 226, 39, 25, 0), c_{16} = (270, 187, 36, 25, 0), c_{17} = (275, 304, 100, 65, 0), c_{18} = (390, 406, 55, 41, 0.5π), c_{19} = (564, 407, 66, 94, 0.5π), c_{20} = (465, 407, 37, 35, 0.5π), c_{21} = (275, 284, 94, 84, 0)), c_{22} = (275, 388, 62, 27, 0), c_{23} = (273, 429, 64, 17, 0), c_{24} = (430, 321, 57, 37, 0).

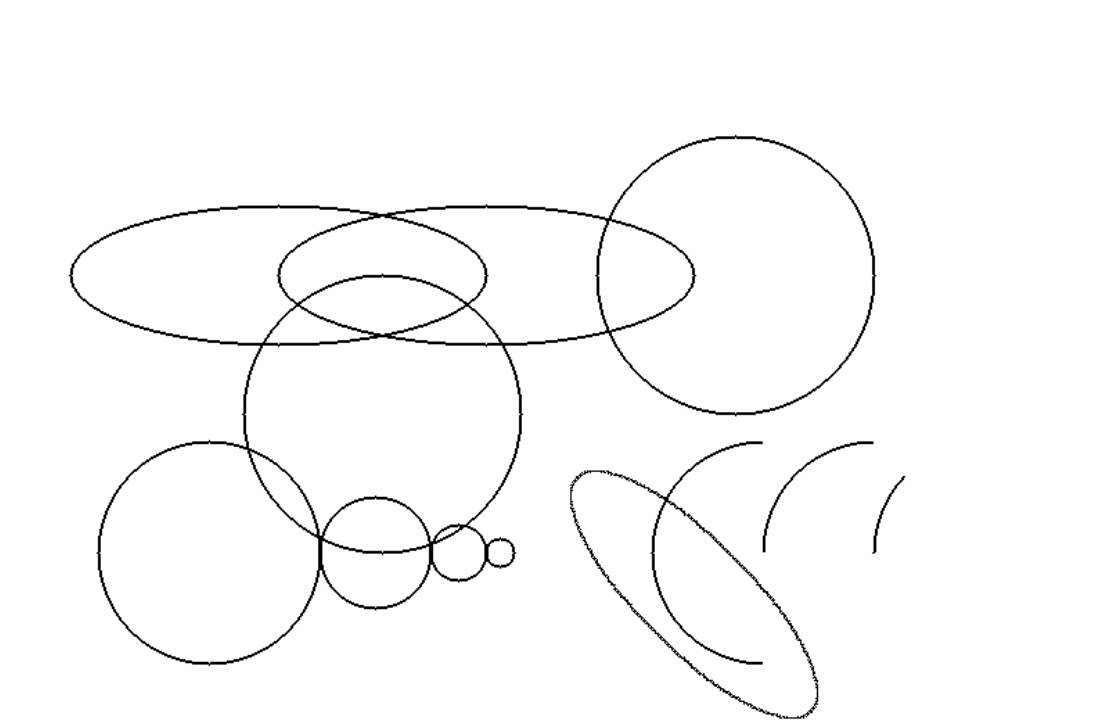


Figure 5.11 A complex synthesized image: nine full ellipses and four partial ellipses overlapped one another.

In F_5, the real ellipses intersect with one another. As a result, some cells that do not correspond to any ellipse have large values of the SMF. All the values of the SMF $M_s(\bullet)$, the NMF $N(\bullet)$ and the UMF $U(\bullet)$ for the image are given in Table 5.5, in which # means the number of the points in the base set. In calculating $U(\bullet)$, an arc is valid to be measured if its length is longer than 7% of the perimeter of the considered ellipse. Furthermore, we calculate $U(c)$ if the sum of lengths of all arcs on an ideal ellipse of cell c is longer than 12% of the perimeter of the ideal ellipse.

Table 5.5 Partial values of $M_s(\bullet)$, $N(\bullet)$, and $U(\bullet)$ for F_5.

Cells	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
#	1268	1268	1132	1132	1012	908	452	228	116	908	908	908
$M_s(\bullet)$	1268	1268	1132	1132	1012	908	452	228	116	452	225	112
$N(\bullet)$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.45	0.25	0.12
$U(\bullet)$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Cells	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}	c_{21}	c_{22}	c_{23}	c_{24}
#	756	357	400	381	1031	600	1001	450	1112	556	505	587
$M_s(\bullet)$	145	78	90	97	188	122	202	116	363	134	58	70
$N(\bullet)$	0.19	0.22	0.22	0.25	0.18	0.20	0.22	0.26	0.33	0.24	0.11	0.12
$U(\bullet)$	0.91	0.00	0.00	1.00	0.00	0.89	0.00	0.60	0.90	0.00	0.00	0.00

Table 5.5 shows that identifying the ellipses from the values of U(•) is much easier than from the values of M_s(•) and N(•), though we still have problem in identifying all ellipses in one round. For such complex image, we need two rounds. The first round is to identify the almost full ellipses and remove them. The second is to identify the partial ellipses. After all almost full ellipses are removed the values of I(•) on cells c_{12} to c_{24} would be 0.00. Thus, we can identify the partial ellipses easily. When we use just a single round, there is *the tangent ellipse problem*, which a cell has a high measured value if it is tangent to a real ellipse. By avoiding *the tangent ellipse problem,* some random Hough transforms (RHT) were implemented to detect the ellipse one by one, i.e. they remove the shape that they have detected and then they detect another shape [XuOK1990, XuOj1993, KiKA2000].

5.4.2 Framework for Detecting Ellipse from BSV

Here we describe the framework for detecting ellipse from broadcast soccer video (BSV), which is depicted in Framework 1. This framework targets to detect partial ellipses that have less than half of their areas out of the frame. The center line of the soccer field appears in frame when more than half of the ellipse appears in the same frame. Hence we first detect the center line and rotate the frame to make the center line vertical. As a result, the ellipse becomes horizontal since it is perpendicular to the center line. Then, we estimate the horizontal ellipse in the rotated frame through sample point statistics and template matching. Note that before the rotation we may shift the frame a little, so the transformed frame can contain most sample points.

Hence, in general we do a transform to the frame including shift and rotation. In the transformed frame, we estimate the locations and sizes of the target ellipses by detecting characteristic points or by making sample point statistics according to the symmetry of ellipse. Once we obtain the estimated ellipses in the transformed frame, we convert them into estimated ellipses in the original frame. Lastly, we search for the best ellipse around the estimated ellipses using various EHTs, including the SEHT, the NEHT, the AFEHT, and the RobustEHT. In this search procedure, we modify the measure function by considering the obliqueness of the ellipse. The concrete measure function was presented in [YLXT2004a].

Framework 5.1

Input is a frame and a prespecific threshold.

Component 1. Estimation of the target ellipse.

1.1. Detection of the center line.

1.2. Transform of the frame.

1.3. Estimation of the target ellipse.

1.4. Conversion of the estimated ellipses.

Component 2. Use various EHTs to detect the target ellipse around the estimated ellipses.

2.1. Determine the set of the sample points and the Hough space H.

2.2. Find the measure value of each cell in H.

2.3. Find the cell with the maximum value in H. Output is zero if the maximum value is less than the predefined threshold; otherwise is the cell with the maximum value.

In the next two subsections, we will compare the performances of different EHTs based on the above framework. The test has been conducted on a whole MPEG1 video of a game, which is the quarter final of 2002 FIFA World Cup (Senegal vs Turkey). This MPEG1 video is recorded from TV signal using WinTV card. The framework gets its frames from the video using Microsoft DirectX^{TM} 9.0.

5.4.3 Comparison on Robustness

Although the NEHT has eliminated the bias to measure small ellipses, it still has a bias against the partial ellipses. We give the detection results of the NEHT on several given global thresholds to show the dilemma of the NEHT between the recall and the precision when it is applied to detect partial ellipses in the frames of BSV. In contrast, the RobustEHT achieves a high recall and 100% precision with the properly-set global threshold. All these results are shown in Table 5.6.

Table 5.6 Comparison on the robustness of RobustEHT and NEHT.

EHT	thresh.	false alarm	recall	precision
RobustEHT	0.650	0	91.3%	100%
NEHT	0.600	0	50.3%	100%
NEHT	0.500	6	68.0%	99.9%
NEHT	0.400	60	88.4%	99.3%
NEHT	0.300	179	92.0%	98.0%

We give the detection results of the SEHT on several given global thresholds to show the dilemma of the SEHT between the recall and the precision when it is applied to detect partial ellipses in the frames of BSV, which are shown in Table 5.6. From Table 5.6 and 5.7, we can see that the NEHT is slightly better than the SEHT and the RobustEHT is much better than the NEHT and the SEHT.

Table 5.7 Comparison on the robustness of RobustEHT and SEHT.

EHT	thresh.	false alarm	recall	precision
RobustEHT	0.650	0	91.3%	100%
SEHT	350	0	68.3%	100%
SEHT	300	93	81.3%	98.8%
SEHT	250	163	88.9%	98.1%
SEHT	200	366	89.9%	95.9%
SEHT	150	485	91.4%	94.7%

5.5 Conclusions

This chapter has presented a new robust ellipse Hough transform (RobustEHT). Since it has eliminated the biases against both the small and partial ellipses, RobustEHT can robustly detect both the small and partial ellipses. Furthermore, it also eliminated the requirement for large amount of memory as it uses an accumulator-free computation scheme. This scheme also eliminated the complex peak detection of the SMF. The experimental results showed that the RobustEHT is more robust than all the existing EHTs.

Chapter 6

Summary and Future Work

The work in this thesis was motivated by a desire to develop a better automatic indexing and retrieval system for broadcast sports video. With the fast-paced advances in computer and video recording, consumers of broadcast sports video have set higher requirements for their video consumption. Consumers are no longer satisfied with just passively viewing the entire sports video; instead they also want to have freedom to choose the video they are interested in and the choice of only receiving the video segments that they deem to be interesting. Therefore, *event detection* is a vital requirement of any modern software application for management of broadcast sports video to address this user requirement. Another important requirement is that of providing enriched broadcast sports video – namely, interesting video segments that have been enriched with graphically generated contents (such as advertisements).

This thesis addressed three closely-related problems that arise in building automatic indexing and retrieval system for broadcast sports video to help in fulfilling these user requirements. In this chapter, we summarize the key contributions in these areas and discuss some avenues for improvements and extensions.

6.1 Summary

This thesis has addressed the following three problems:

- the ball detection and tracking problem in broadcast sports video
- the applications of the ball locations computed in ball-related event detection and the enrichment of broadcast soccer video
- the ellipse detection problem in broadcast soccer video

In Chapter 3, we presented a trajectory-based ball detection and tracking algorithm for BSV for the first problem that the thesis addressed. In this algorithm, a batch of techniques was used to overcome the challenges of the problem. A ball-size estimation method, which estimated the ball sizes from the sizes of the detected salient objects, overcame the challenge that the ball sizes change over frames. Then we built six sieves (*anti-models*) to identify the non-ball objects, which avoided the difficult task to build the representation of the ball. By using six sieves to remove the non-ball objects, we obtained the ball candidates, which have high false positives and low false negatives. The candidate feature image (CFI) was introduced to visually present the space-temporal relation among the candidates of a sequence of frames. The CFIs gave us a great intuition in designing our algorithms and a great convenience for visually verifying the results of our algorithms. The Kalman filter was successfully used to generate the candidate trajectories from various CFIs. To avoid being fooled by the non-ball objects that look like the ball more than the actual ball itself, we did not decide whether an object is a ball. Instead, we determined whether a trajectory was a ball trajectory, termed as ball trajectory mining. We recovered from high false positives

through trajectory mining. Based on the mined ball trajectories, we further refined the ball trajectories. The proposed ball detection and tracking algorithm achieved a very good result in terms of the accuracy of locating the ball and the false alarms. In addition, the false alarms are very close to the real ball locations.

We did various experiments relating to the above-mentioned algorithm. First, we evaluated the detection and tracking performances of the algorithm. Second, we tested the techniques of ball size estimation and ball trajectory mining. Experimental results showed that ball trajectory mining technique is very robust, i.e. it can tolerate a certain percentage of missing real ball candidates. Thus, the estimated ball sizes were not very accurate, but they helped us to obtain the candidates with good quality. The experiments also proved that the penalty mark filter was indispensable because without it the algorithm might consider that a trajectory of a penalty mark was a ball trajectory. Last, we compared our algorithm with the algorithm presented in [DACN2002, DGLD2004] by D'Ozao et al. The experimental results showed our algorithm much outperformed theirs.

We also applied the trajectory-based approach to develop a ball detection and tracking algorithm for broadcast tennis video. This algorithm adopted many principles and techniques used in the algorithm for locating the ball from BSV. On the other hand, we proposed several techniques to tackle the unique challenges of locating the ball from BTV. For example, we used the hitting point to infer the pivot (*direction turn point*) of ball trajectory. We also used the position of player and hittings to infer the approximate ball position.

In Chapter 4, we proposed a *trajectory-based event detection* approach for the detection of ball-related events in broadcast soccer video. Our approach also makes use of the ball locations obtained in Chapter 3. This approach not only improved play-break analysis and high-level semantic event detection, but also detected the basic actions and analyzed team ball possession, which might not be analyzed based on only the low-level features. Following this approach, we designed various algorithms to detect touching, passing, goal, team possession, and play/break. In addition, we developed an enhancement and enrichment system. This system first automatically generated the video of the goalmouth scene as the existing systems did and the midfield scene, i.e. our system extended the generation range of BSV. Then the system enriched the generated video using the icons that illustrate the information derived from ball locations and the results of event detection to enhance the viewers' viewing experience.

In Chapter 5, we presented a robust and accumulator-free ellipse detection algorithm. This algorithm was motivated by our need for an accurate and robust ellipse detection algorithm that is capable of detecting the partial ellipses commonly found in BSV. This algorithm is robust because we proposed an unbiased measure function that could fairly measure the ellipses in different sizes and/or partial ellipses. This algorithm is accumulator-free because it could select the limited number of the cells that have a higher probability to correspond to real ellipses among the measured cells. Experimental results showed that this algorithm is much more robust than the existing ellipse Hough transforms that used the absolute measure functions.

6.2 Future Work

This thesis also leaves a number of topics unexplored. Here, we highlight some possible extensions and new research directions.

- The algorithms in Chapter 3 were designed to overcome the key challenges in ball detection and tracking but some challenges remain. One such challenge is the presence of shadow – and extending the algorithm to account for shadow will enlarge the application range of the algorithms.
- The algorithms in Chapter 3 require the manual settings of some parameters such as the ball color, line color, and player color as their inputs. Automatically obtaining these parameters should be a good extension of the algorithms.
- The trajectory-based approach proposed in the thesis has been applied to two sports: soccer and tennis video with promising results. We aim to extend the use of this approach to other sports. For example, we can develop trajectory-based algorithms for locating the ball in badminton video, basketball video, golf video, etc.
- In Chapter 3, trajectory analysis including trajectory generation and trajectory processing showed its power and robustness. Along this direction, it is possible to build a trajectory process machine, which produces the target trajectory from the given candidates and does trajectory criteria evaluation.
- The trajectory-based event detection is another new approach. In Chapter 4, several algorithms were designed to detect various events. These algorithms have shown the advantage of trajectory-based event detection approach. Following this approach, a further study can afford us to detect

more events and more details of the events. It is possible that this approach can improve the performance of event detection for other ball games.

- The ball trajectory has some relations to high-level events. Thus, it might be good to index the video using different types of trajectory.
- Based on the ball trajectory and detected trajectory-based events, an improved annotation system may be a promising research area.
- It is a possible interesting direction to apply the techniques of enhancement and enrichment of sports video to the game industry. Currently, making games requires a large amount of manual work. The automatic generating video has the possibility to be adopted to reduce this manual work.
- The presented ellipse detection algorithm offers many opportunities for improvement. For example, it is possible to speed up the algorithm. When our ellipse detection algorithm was applied to detect the ellipse in BSV, we made use of the domain knowledge to tighten the Hough space. For the normal images, it is possible that the algorithm can integrate the existing speed-up techniques such as probabilistic sampling to form even better ellipse detection algorithms.
- Chapter 5 developed some techniques to make ellipse detection algorithms robust and accumulator-free. These techniques have the potential to be applied to other shape detection algorithms.

References

ABBN2002 J. Assfalg, M. Bertini, A. del Bimbo, W. Nunziati, and P. Pala. Soccer highlights detection and recognition using HMMs, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2002), pp. 825-828, 2002.

ABCB2001 J. Assfalg, M. Bertini, C. Colombo, and A. del Bimbo. Extracting semantic information from news and sport video, *Proc. Int'l Sym. on Image and Signal Processing and Analysis* (ISPA 2001), pp. 4-11, 19-21 June 2001.

ABCB2002 J. Assfalg, M. Bertini, C. Colombo, and A. del Bimbo. Semantic annotation of sports videos, *IEEE Multimedia*, vol. 9, no. 2, pp. 52-60, Apr.-June 2002.

ABCB2003a J. Assfalg, M. Bertini, C. Colombo, D. del Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification**,** *Computer Vision and Image Understanding*, vol. 92, pp. 285-305, 2003.

ABCB2003b J. Assfalg, M. Bertini, C. Colombo, D. del Bimbo, and W. Nunziati. Automatic extraction and annotation of soccer video highlights**,** *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2003), vol. 3, pp. 527-30, 14-17 Sept. 2003.

ABCB2003c J. Assfalg, M. Bertini, C. Colombo, D. del Bimbo, and W. Nunziati. Highlight extraction in soccer videos**,** *Proc. Int'l Conf. on Image Analysis and Processing* (ICIAP 2003), pp. 498-503, 17-19 Sept. 2003.

ACCE1996 S. Adali, K. S. Candan, S. -S. Chen, K. Erol, and V. S. Subrahmanian. The advanced video information system: data structures and query processing, *Multimedia Systems*, vol. 4, no. 4, pp. 172-186, 1996.

ACSS2003 K. O. Arras, J. A. Castellanos, M. Schilt, and R. Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints, *Robotics and autonomous systems*, vol. 44, pp. 41-53, 2003.

ADHC1994 F. Arman, R. Depommier, A. Hsu, and M. Y. Chiu. Content-based browsing of video sequences, *Proc. ACM Multimedia* (ACM MM 1994), pp. 97-103, 1994.

Ati1992 M. Atiquzzaman. Multiresolution Hough transform-An efficient method of detecting patterns in images, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 14, no. 11, pp. 1090-1095, 1992.

ATOH1992 A. Akutsu, Y. Tonomura, Y. Ohba, and H. Hashimoto. Video indexing using motion vectors, *Proc. of SPIE Visual Comm and Image Processing* (SPIE 1992), pp. 343-350, 1992.

BaKK2002 N. Babaguchi, Y. Kawai, and T. Kitashi. Event based indexing of broadcasted sports video by intermodal collaboration, *IEEE Trans. on Multimedia*, vol. 4, pp. 68-75, March, 2002.

Bal1981 D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

BeBC1999 A. B. Benitez, M. Begi, and S. F. Chang. Using relevance feedback in content-based image metadata search, *IEEE Internet Computing*, pp. 59-69, July-Aug. 1999.

BeBS1999 N. Bennett, R. Burridge, and N. Saito. A method to detect and characterize ellipses using the Hough transform, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 21, no. 7, pp. 652-657, Jul 1999

BeBi1998 T. Bebie and H. Bieri.SoccerMan--Reconstructing soccer games from video sequences, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP1998), pp. 898-902, 1998.

BeBi2000 T. Bebie and H. Bieri.A video-based 3D reconstruction of soccer games, *Eurographics,* vol. 19, no. 3, 2000.

BeNS1989 D. Ben-Tzvi, A. A. Naqvi, and M. B. Sandler. Efficient parallel implementation of the Hough transform on a distributed memory system, *Image and Vision Computing*, vol. 7, no. 3, pp. 167-172, August 1989.

BeSa1990 D. Ben-Tzvi and M. B. Sandler. A combinatorial Hough transform, *Pattern Recognition Letters*, pp. 167-174, vol. 11, no. 3, 1990.

BeSh1991 J. R. Bergen and H. Shvayster. A probabilistic algorithm for computing Hough transforms, *J. of Algoritms*, pp. 639-656, 1991.

Bim2002 A. D. Bimbo, Multimedia computing and systems, *IEEE Multimedia*, vol. 7, no. 1, pp. 18-21, Jan. March, 2000.

BoFr1993 P. Bouthemy and E. Francois. Motion segmentation and qualitative scene analysis from an image sequence. *Int'l J. of Computer Vision*, vol. 10, pp.157-182, 1993.

BoLi1991 C. Bouman and B. Liu. Multiple resolution segmentation of textured images, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 13, no. 2, pp. 99-113, 1991.

BoMe2003 J. E. Boyd and J. Meloche. Evaluation of statistics and multiple-hypothesis tracking for video traffic surveillance, *Machine Vision and Applications*, vol. 13, pp.344-351, 2003.

BrMM1999 R. Brunelli, O. Mich, and C. Modena. A survey on the automatic indexing of video data, *J. of Visual Communication and Image Representation*, vol. 10, no 2, pp. 78-112, 1999.

Bro1983 C. M. Brown. Inherent bias and noise in the Hough transform, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 5, pp. 493-505, 1983.

BSAD2001 A. Branca, E. Stella, N. ancona, and A. Distante. Goal distance estimation in soccer game, *Proc. IEEE Int'l Conf. on Image Analysis and Proc* (ICIAP 2001), pp. 565-569, 2001.

CaDi2002 R. Cabasson and A. Divakaran. Automatic extraction of soccer video highlights using a combination of motion and audio features, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, vol. 5021, pp. 272-276, Jan. 2002.

CCMS1998 S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. A fully automated content-based video search engine supporting spatiotemporal queries, *IEEE Trans. on Circuits and Sys. for Video Tech.* vol. 8, no. 5, pp. 602-615, Sept. 1998.

Cha2002 S. F. Chang. The holy grail of content-based media analysis, *IEEE Multimedia*, vol. 9, pp. 6-10, April-June 2002.

ChLi2002 C.-C. Chang and C.-J. Lin. LIBSVM---A Library for Support Vector Machines, *http://www.csie.ntu.edu.tw/~cjlin/libsvm/* (as of 2002).

Che1976 P. P. S. Chen. The entity-relationship model--Toward a unified view of data, *ACM Databases*, vol. 1, no. 1, pp. 9-36, March.1976.

Che1995 Y. Cheng. Mean shift, mode seeking, and clustering, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 17, no. 8, pp. 790-799, 1995.

ChHG2002 P. Chang, M. Han, and Y. Gong. Extract highlights from baseball game video with hidden Markov models, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2002), 2002.

CHHG2003 T. Chen, M. Han, W. Hua, Y. Gong, and T. S. Huang. A new tracking technique: object tracking and identification from motion, *Proc. Int'l Conf. on Computer Analysis of Images and Patterns* (CAIP 2003), Aug. 25-27, Groningen, The Netherlands, 2003.

ChLi2003 Y. C. Cheng and Y. S. Liu. Polling an image for circles by random lines, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 25 no. 1, pp. 125- 130, 2003.

ChSL1999 H. S. Chang, S. Sull, and S. U. Lee. Efficient video indexing scheme for content-based retrieval, *IEEE Trans. on Circuits and Sys. for Video Tech*. vol. 9, no. 8, pp. 1269-1279, Dec. 1999.

ChSL2004 K. Choi, Y. Seo, and S. W. Lee. Probabilistic tracking of soccer ball player and ball, *Proc. European Conf. on Comp. Vision* (ECCV 2004), pp. 112-119, 2004.

CoBP1999 C. Colombo, A. Del Bimbo, and P. Pala. Semantics in visual information retrieval, *IEEE Multimedia*, vol.6, no.3, pp. 38-53, July-Sept. 1999.

CoHi1994 I. J. Cox and S. L. Hingorani. An efficient implementation of and evaluation of Reid's multiple hypothesis tracking algorithm for visual tracking, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 1994), pp. 437-442, 1994.

CoHi1996 I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol.18. no.2, pp.138-150, 1996.

CoLe1991 I. J. Cox and J. J. Leonard. Probabilistic data association for dynamic world modeling: A multiple hypothesis approach, *Proc. Int'l Conf. Advanced Robotics,* Pisa, Italy, 1991.

CoMe2002 D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 24, no. 5, pp. 1-18, 2002.

CoRM2000 D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2000), vol. 2, pp. 142-149, 2000.

Cou1997 J. D. Courtney. Automatic video indexing via object motion analysis, *Pattern Recognition*, vol. 30, no. 4, pp. 607-626, Apr. 1997.

CoVa1995 C. Cortes and V. Vapnik. Support-vector networks, *Machine Learning*, vol. 20, pp. 273-297, 1995.

Cox1993 I. J. Cox. A review of statistical data association techniques for motion correspondence, *Int'l J. of Computer Vision*, vol. 10, no 1, pp.53-66, 1993.

CSCZ2004 S.-C Chen, M.-L. Shyu, M Chen, and C. Zhang. A decision tree-based multimodal data mining framework for soccer goal detection, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2004), pp. 265-268, Taibei, Taiwan, June, 2004.

CSKH1997 S. Choi, Y. Seo, H. Kim, and K. S. Hong. Where are the ball and players?: Soccer game analysis with color-based tracking and image mosaick, *Proc. of Int'l Conf Image Analysis and Processing*, (ICIAP 1997), 1997

CZGS1995 C. H. Chuan, H. Zhang, M. Sakauchi, Y. Gong, and L. T. Sin. Automatic parsing of TV soccer programs, Proc*. of Int'l Conf. on Multimedia Computing and Systems* (ICMCS 1995), pp. 167-174, 1995.

CZKA1996 Y. L. Chang, W. Zeng, I. Kamel, and R. Alonso. Integrated image and speech analysis for content-based video indexing, *Proc. of Int'l Conf. on Multimedia Computing and Systems* (ICMCS 2002), pp. 306-313, 1996.

DACN2002 T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences, *Proc. of Int'l Conf on Pattern Recognition* (ICPR 2002), 2002.

DAGK2000 S. Dagtas, W. Al-Khatip, A. Ghafoor, and R. L. Kashyap. Models for motion-based video indexing and retrieval, *IEEE Trans. on Image Processing*, vol. 9, no. 1, pp. 88-101, Jan. 2000.

DaHD2003 H. Dang, C. Han, and Z. Duan. A new data association approach for automatic radar tacking, *Fusion 2003*, pp. 1384-1388.

Dav1989 E. R. Davies. Finding ellipse using the generalised Hough transform, *Pattern Recognition Letters*, vol. 9, pp. 87-96, 1989.

Dav1992a E. R. Davies. Modelling peak shapes obtained by Hough transform, *IEE Proc. E: Computers and Digital Techniques*, vol. 139, no 1, pp. 9-12, 1992.

Dav1992b E. R. Davies. Simple two-stage method for the accurate location of Hough transform peaks, *IEE Proc. E: Computers and Digital Techniques*, vol. 139, no 3, pp. 242-248, 1992.

Dav1995 M. E. Davis. Media streams: Representing video for retrieval and repurposing, *PhD Dissertation*, MIT, 1995.

DGLD2004 T. D'Orazio, C. Guarangnella, M. Leo, and A. Distante. A new algorithm for ball recognition using circle Hough transform and neural classifier, *Pattern Recognition*, vol. 37, pp. 393-408, 2004.

DKRD2003 R. Dahyot, A. C. Kokaram, N. Rea, and H. Denman. Joint audio-visual retrieval for tennis broadcasts, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2002), Apr. 2003.

DLNC2002 T. D'Orazio, M. Leo, M. Nitti, and G. Cicirelli. A real time ball recognition system for sequences of soccer images, *Proc. of IASTED SPPRA Conf.*, Crete, Greece, June 2002.

DLND2002 T. D'Orazio, M. Leo, M. Nitti, and A. Distante. Ball recognition in real sequence of soccer images with different light conditions, *Proc. of IEEE Conf. on Systems, Cybernetics and Informatics*, Orlando, 2002.

DMNR2002 C. Dorai, A. Mauthe, F. Nack, L. Rutledge, T. Sikora, and H. Zettl. Media semantics: Who needs it and why? *Proc. of ACM Multimedia* (ACM MM 2002), pp. 580-583, 2002.

DrCi2001 T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements, *Proc. of Int'l conf. on Computer Vision* (ICCV 2001), 2001.

DuHa1972 R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures, *Comm. Assoc. Comput. Mach.,* vol. 15, pp. 11-15, 1972.

DuXT2003 L. Duan, M. Xu, and Q. Tian. Semantic shot classification in sports video, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, vol. 5021, pp. 300-313, Jan. 2003.

DXCT2003 L. Duan, M. Xu, T. S. Chua, Q. Tian, and C. Xu. A mid-level representation framework for semantic sports video analysis, *Proc. of ACM Multimedia* (ACM MM 2003), pp. 33-44, 2003.

DXTX2003 L. Duan, M. Xu, Q. Tian, and C. Xu. Nonparametric color characterization using mean shift, *Proc. of ACM Multimedia 2003* (ACM MM 2003), pp. 243-246, 2003.

DXTX2004 L. Duan, M. Xu, Q. Tian, and C. Xu. Mean shift based video segment representation and applications to replay detection, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2004), 2004.

EiMu1999 S. Eickeler and S. Muller. Content-based video indexing of TV broadcast news using HMMs, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 1999), pp. 2997-3000, 1999

Eki2003 A. Ekin. Sports video processing for description, summarization, and search, *PhD Dissertation*, School of Engineering and Applied Science, University of Rochester, Rochester, New York, USA, Jan. 2003.

EkTe2002 A. Ekin and A. M. Tekalp. A framework for analysis and tracking of soccer video, *Symp. Electronic Imaging: Science and Technology: Visual Com. and Image Processing* (VCIP 2002), Jan. 2002.

EkTe2003a A. Ekin and A. M. Tekalp. Robust dominant color region detection and color-based applications to sports video, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2003), 2003.

EkTe2003b A. Ekin and A. M. Tekalp. Generic play-break event detection for summarization and hierarchical sports video analysis, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2003), July 2003.

EkTe2003c A. Ekin and A. M. Tekalp. Automatic soccer video analysis and summarization, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases IV*, Jan. 2003.

EkTe2003d A. Ekin and A. M. Tekalp. Generic Event Detection in Sports Video using Cinematic Features, *Computer Vision and Pattern Recognition Workshop*, Vol 4, Madison, Wisconsin, USA, June 16 - 22, 2003.

EkTM2003 A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization, *IEEE Trans. Image Processing*, vol. 12, no. 7, pp. 796-807, 2003.

ErTS2003 C. E. Erdem, A. M. Tekalp, and B. Sankur. Video object tracking with feedback of performance measures, *IEEE Trans. on Circuits and Sys. for Video Tech.*, vol. 13, no. 4, pp. 310-324, Apr. 2003.

FeTe1998 A. M. Ferman and A. M. Tekalp. Efficient filtering and clustering for temporal video segmentation and visual summarization, J. Visual Com. and Image Representation, vol. 9, 1998.

FeTe2001 A. M. Ferman and A. M. Tekalp. A fuzzy framework for unsupervised video content characterization and shot classification, J. of Electronic Imaging, vol. 10, pp. 917- 929, Oct. 2001.

FeTM2002 A. M. Ferman, A. M. Tekalp, and R. Mehrotra. Robust color histogram descriptors for video segment retrieval and identification, *IEEE Trans. on Image Processing*, vol. 11, pp. 497-508, May 2002.

FETM2002 Y. Fu, A. Ekin, A. M. Tekalp, and R. Mehrotra. Temporal segmentation of video objects for hierarchical object-based motion description, *IEEE Trans. Image Processing*, vol. 11, pp. 135-145, Feb. 2002.

FIFAlaw International Football Association Board. Laws of the games, Federation Internationale de Football Association, 11 Hitzigweg, 8030 Zurich, Switzerland, July 2002.

FiHi1989 A. L. Fisher and P. T. Highnam. Computing the Hough transform on a scan line processor (image processing), *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 11, no. 3, pp. 262-265, 1989.

FoMR2002 G. L. Foresti, L. Marcenaro, and C. S. Regazzoni. Automatic detection and indexing of video-event shots for surveillance applications, *IEEE Trans. on Multimedia*, vol. 4, no. 4, pp. 459-471, Dec. 2002.

GaKA1998 U. Gargi, R. Kasturi, and S. Antani. Performance characterization and comparison of video indexing algorithms, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2001), 1998.

GBSG2001 J. Geusebroek, R. van den Boomgaard, A. Smeulders, and H. Geerts. Color invariance, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 23, no. 12, 2001.

GeSm1999 T. Gevers and A. W. M. Smeulders. Color based object recognition, *Pattern Recognition*, vol. 32, pp. 453-464, May 1999.

GLCZ1995 Y. Gong, T. S. Lim, H. C. Chua, H. J. Zhang, and M. Sakauchi. Automatic parsing of TV soccer programs, *2nd Int'l C. on Multimedia Comp. and sys.,* pp. 167-174, 1995.

GoWo1992 R. C. Gonzalez and R. E. Woods. Digital image processing, *Addison-Wesley*, Boston, MA, 1992.

GrHu1990 W. E. L. Grimson and D. P. Huttenlochner. On the sensitivity of the Hough transform for object recognition, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 10, no. 3, pp. 255-274, 1990.

Gro1997 W. I. Grosky. Managing multimedia information database systems, *Communications of the ACM*, vol.40, no.12, pp. 72-80, 1997.

GuFT1998 B. Gunsel, A. M. Ferman, and A. M. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking, *J. Electronic Imaging*, vol. 7, pp. 592-604, 1998.

GuLe1998 C. Gu and M. -C. Lee. Semiautomatic segmentation and tracking of semantic video objects, *IEEE Trans. Circ. Syst. Video Tech.* vol. 8, pp. 572-584, Sept. 1998.

GuNe1993 J. Gu and E. J. Neuhold. A data model for multimedia information retrieval, *Proc. of 1st Int'l Conf on Multimedia Modelling*, Singapore, pp. 113-127, 1993.

GuRV1996 V. N. Gudivada, V. V. Raghavan, and K. Vanapipat. A unified approach to data modelling and retrieval for a class of image database applications, *In Multimedia Database Systems, Issues and Research Directions, Subrahmanian*, V.S., S.Jajodia (Eds), Springer Verlag, pp.37-78, 1996.

GuZa1997 N. Guil and E. L. Zapata. Lower order circle and ellipse Hough transform, *Pattern Recognition*, vol. 30, pp. 1729-1744, 1997.

HaAn1997 K. Hansen and J. D. Anderson. Understanding the Hough transform: Hough cell support and its utilization, *Image and Vision Computing,* vol. 15, pp. 20-218, 1997.

HaDK2000 M. -S. Hacid, C. Decleir, and J. Kouloumdjian. A database approach for modeling and querying video data, *IEEE Trans. on Knowledge and Data Eng*., vol. 12, no. 5, pp. 729-750, 2000.

Ham1999 A. Hampapur. Semantic video indexing: Approach and issue, *SIGMOD Record, Special section semantic interoperability in global information systems*, vol.28, March 1999.

HaZi2003 R. Hartley and A. Zisserman, Multiple view geometry in computer vision, *Cambridge University Press* (2rd edition), UK, 2003.

HSGG1999 L. He, E. Sanocki, A. Gupta, J. Grudin. Auto-summarization of audio-video presentations, *Proc. ACM Multimedia* (ACM MM 1999), Nov.1999.

HeMa1995 R. S. Heller and C. D. Martin. A media taxonomy, *IEEE Multimedia*, vol.2 no.4, pp. 36-45, winter 1995.

HHXG2002 M. Han, W. Hua, W. Xu, and Y.H. Gong. An integrated baseball digest system using maximum entropy method, *Proc. of ACM Multimedia* (ACM MM 1999), pp. 347-350, 2002.

HLWC1999 J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. Wong. Integration of multimodal features for video scene classification based on HMM, *IEEE Workshop on Multimedia Signal Processing*, 1999.

HMSP2002 A. Haas, R. Maierhofer, R. Sendlhofer, and M. Polzleitner. Real-time scene analysis of the location of the ball position in soccer games, *The World Multiconference on Systemics, Cybernetics and Informatics,* Orlando, Florida, July 14-18, 2002.

HNRR1990 D. J. Hunt, L. W. Nolte, A. R. Reibman, and W. H. Ruedger. Hough transform and signal detection theory performance for images with addictive noise, *Comput. Vision Graphics Image Process.* vol. 52, no. 3, pp. 386-401, 1990.

Hou1962 P. V. C. Hough. Method and means for recognising complex patterns, U.S. Patent no. 3069654, 1962.

HuLW2000 J. Huang, Z. Liu, and Y. Wang. Joint video scene segmentation and classification based on hidden Markov model, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2000), vol.3, pp. 1551-1554, New York, 2000.

IlKi1987 J. Illingworth and J. Kittler. The adaptive Hough transform, *IEEE Trans. on Pattern Anal. Machine Intel.,* vol. 9, no. 5, 1987.

IlKi1988 J. Illingworth and J. Kittler. A survey of the Hough transform, *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 87-116, 1988.

ImHT2002 A. Imiya, T. Hada, and K. Tatara. The Hough transform without the accumulators, *SSPR & SPR 2002, INCS,* pp. 823-832, 2002.

InSa2002 N. Inamoto and H. Saito. Intermediate view generation of soccer scene from multiple videos, *Proc. of Int'l Conf on Pattern Recognition* (ICPR 2002), August 2002.

IrAn1998 M. Irani and P. Anandan. Video indexing based on mosaic representation, *IEEE Trans. on Pattern Anal. Machine Intel.* vol. 86, no. 5, pp. 905-921, May. 1998.

IsBl1996 M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density, *Proc. European Conf. on Comp. Vision* (ECCV 1996), pp. 343-356, 1996.

IsBl1998a M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking, *J. of Computer Vision*, vol.29, no.1, pp. 5-28, August 1998.

IsBl1998b M. Isard and A. Blake. ICondensation: Unifying low level and high-level tracking in a stochastic framework, *Proc. European Conf. on Comp. Vision* (ECCV 1998), pp. 767-781, 1998.

JaFa1991 A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters, *Pattern Recognition*, vol. 23, no. 12, pp.1167-1186, Dec. 1991.

JaKi1994 A. Yla-Jaaski and N. Kiryati. Adaptive termination of voting in the probabilistic circular Hough transform, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 16, pp. 911-915, 1994.

JaSL1998 A. K. Jain, G. Sudhir, and J. C. M. Lee. Automatic classification of tennis video for high-level content-based retrieval, *Proc of the Int'l Workshop on Content-based Access of Image and Video Databases* (CAIVID 1998), 1998.

KaBG1990 K. P. Karmann, A. V. Brandt, and R. Gerl. Moving object segmentation based on adaptive reference images, *Proc. Conf. Eusipco*, Barcelona, pp. 951-954, Sept. 1990.

KaHi1997 H. Kälviäinen and P. Hirvonen. An extension to the randomized Hough transform exploiting connectivity, *Pattern Recognition Letters*, vol. 18, pp. 77-85, 1997.

KaKA1999 H. Kälviäinen, N. Kiryati, and S. Alaoutinen. Randomized or probabilistic Hough transform: Unified performance evaluation, *Proc. IAPR SCIA*, Kangerlussuaq, Greenland, pp. 256-266, 1999.

KaOh2001 K. Kanatani and Naoya Ohta. Automatic detection of circular objects by ellipse growing, *Memoirs of the Faculty of Engineering, Okayama University*, Japan, vol. 36. no. 1pp.107-116, Dec, 2001.

KaOh2002 K. Kanatani and Naoya Ohta. Automatic detection of circular objects by ellipse growing, *The 9th Syposium on Sensing via Image Information* (SSII2002), pp. 355-360, Yokohama, Japan, 2002

KaYA1994 T. Kawashiima, K. Yoshino, and Y. Aok. Qualitative image analysis of group behavior, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1994), June. 1994.

KeOG2001 D. K. Keren, M. Osadchy, and C. Gotsman. Antifaces: A novel, fast method for image detection, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 23, no. 7, pp.747-761, July. 2001.

KHCS1997 H. Kim, K. S. Hong, S. Choi, and Y. Seo. Where are the ball and players?: Soccer game analysis with color-based tracking and image mosaick, *Proc. Int'l Conf. on Image Analysis and Processing* (ICIAP 1997), 1997.

KHXK1995 H. Kälviäinen, P. Hirvonen, L. Xu, and O. Erkki. Probabilistic and non-probabilistic Hough transforms: Overview and comparisons, *Image and Vision Computing*, vol. 13, pp. 239-252, 1995.

KiBr1991 N. Kiryati and A. M. Bruckstein. Antialiasing the Hough transform, *CVGIP*: *Graphical Models Image Processing*. vol. 53, no. 3, pp. 213-222, 1991.

KiEB1991 N. Kiryati, Y. Eldar, and A. Bruckstein. A probabilistic Hough transform, *Pattern Recognition*, vol. 24, no. 4, pp. 303-316, 1991.

KiHo2001 H. Kim and K. S. Hong. Soccer video mosaicing using self-calibration and line tracking, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 2000), pp. 592- 595, 2000.

KiKa1992 T. Kikukawa and S. Kawafuchi. Development of an automatic summary editing system for audio-visual resources, *Trans. Electron. Inform*, pp. 204-212, 1992.

KiKA2000 N. Kiryati, H. Kälviäinen, and S. Alaoutinen. Randomized or probabilistic Hough transform: Unified performance evaluation, *Pattern Recognition Letters*, vol. 20, pp. 1157-1164, 1999.

KiLB1991 N. Kiryati, M. Lindenbaum, and A. M. Bruckstein. Digital or analog Hough transform, *Pattern Recognition Letters*, vol.12, pp. 291-297, 1991.

KiOG2003 E. Kijak, L. Oisel, and P. Gros. Temporal structure analysis of broadcast tennis video using Hidden Markov Models, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, vol. 5021, pp. 277-288, Jan. 2003.

KiSH1998 H. Kim, Y. Seo, and K. S. Hong. Physics-based 3D position analysis of a soccer ball from monocular image sequences, *Proc. Int'l Conf. on Computer Vision* (ICCV 1998), pp. 721-726, 1998.

KoDD1999 V. Kobla, D. DeMenthon, and D. Doermann. Detection of slow-motion replay sequences for identifying sports videos, *Proc. 3rd Workshop on Mult. Signal Processing* (MMSP 1999)pp. 135-140, 1999.

KoDD2000 V. Kobla, D. DeMenthon, and D. Doermann. Identifying sports videos using replay, text, and camera motion features, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, vol. 3972, pp. 332-343, Jan. 2000.

KoKO2003 T. Koyama, I. Kitahara, and Y. Ohta. Live mixed-reality 3D video in soccer stadium, *Proc. IEEE and ACM Int'l Symp. On mixed and augmented reality* (ISMAR 2003), pp. 178-186, Tokyo, Japan, October 07-10, 2003.

KoDN1993 D. Koller, K. Danilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes, *J. of Computer Vision*, vol.10, no. 3, pp. 257-281, 1993.

KRST1997 S. R. Kulkami, P. J. Ramadge, D. D. Saur, and Y. P. Tan. Automatic analysis and annotation of basketball video, *Storage and Retrieval for Image and Video Databases V*, pp. 176-187, Feb.1997.

KTLA1998 T. Kawashima, K. Tateyama, T. Iijima, and Y. Aoki. Indexing of baseball telecast for content-based video retrieval, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 1998), pp. 871-874, 1998.

Kul1979 Z. Kulpa. On the properties of discrete circles, rings, and disks, *Computer Graphics and Image Processing*, vol. 10, no. 4, pp.348-365, August 1979.

KuRB2001 G. Kuhne, S. Richter, and M. Berer. Motion-based segmentation and contour-based classification of video objects, *Proc. ACM Multimedia* (ACM MM 2001), 2001.

LaVW2002 M. Lazarescu, S. Venkatesh, and G. West. On the automatic indexing of cricket using camera motion parameters, *Proc. Int'l Conf. on Multimedia and Expo* (ICME 2002), pp. 809-812, 2002.

LeBS1989 V.F. Leavers, D. Ben-Tzvi, and M.B. Sandler, A dynamic combinatorial Hough transform for straight lines and circles, *Proc. 5th Alvey Vision Conf.*, pp. 163-168, UK, Sept. 1989.

Lea1990 V. F. Leavers. The dynamic generalized Hough transform, *Proc. European Conf. on Comp. Vision* (ECCV 1990), Antibes, France, April.1990.

Lea1992a V. F. Leavers. It's probably a Hough: The dynamic generalized Hough transform, and an application to the concurrent detection of circles and ellipses, *Comput. Vision Graphics Image Processing*, vol.56, no. 3, pp. 381-398, 1992.

Lea1992b V. F. Leavers. The dynamic generalized Hough transform: Its relationship to the probabilistic Hough tranforms and an application to the concurrent detection of circles and ellipses, CVGIP: Image Understanding, vol. 56, pp. 381-398, 1992.

Lea1993 V. F. Leavers. Which Hough transform? *CVGIP: Image understanding*, vol. 58, pp. 250-264, 1993.

LeDD2003 M. Leo, T. D'Orazio, and A. Distante. Feature extraction for automatic ball recognition: comparison between wavelet and ICA preprocessing, *Proc. of the 3rd Int'l Sym. on Image and Signal Processing and Analysis*, vol. 2, pp. 587-592, 2003.

LeHM1994 C. H. C. Leung, J. Hibler, and N. Mwara. Content-based retrieval in multimedia databases, *Computer Graphics*, vol. 28, no. 1, pp.24-28, Feb. 1994.

LeMi2002 R. Leonardi, and P. Migliorati. Semantic indexing of multimedia documents, *IEEE Multimedia*, vol. 9, pp. 44-51, Apr.-June 2002.

LEPS2003 B. Li, J. Errico, H. Pan, and M. I. Sezan. Bridging the semantic gap in sports, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases vol. 5021*, pp. 314-326, Jan. 2003. (Awarded Best Paper Prize for Contribution with Most Industrial Potential).

LeSF2003 V. Lepetit, A. Shahrokni, and P. Fua, Robust data association for online applications, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2003), vol. 1, pp. 281-288, 2003.

LeWo1999 Y. Lei and K. C. Wong. Ellipse detection based on symmetry, *Pattern Recognition Letters,* vol. 20, no. 1, pp. 41-47, 1999.

LiDK2000 H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video, *IEEE Trans. on Image Processing*, vol. 9, no. 1, pp. 147-156, 2000.

LiDo1998 H. Li and D. Doermann. Automatic identification of text in digital video key frames, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 1998), pp. 129-132, 1998.

LiEf2000 R. Lienhart, and W. Effelsberg. Automatic text segmentation and text recognition for video indexing, *Multimedia Systems Journal*, vol. 8, no. 1, pp. 69-81, 2000.

LiHF1990 Y. Liu, S. Huang, and O. D. Faugeras. Determination of camera location from 2D to 3D line and point correspondences. *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 12, no. 1, pp. 28-37, 1990.

LiLa1986 H. Li and M. A. Lavin. Fast Hough transform based on the bintree data structure, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1986), pp. 640-642, 1986.

LiLL1986 H. Li, M. A. Lavin, and R. J. LeMaster. Fast Hough transform: A hierarchial approach, CVGIP, vol. 36, no. 2/3, pp. 139-161, 1986.

LiOS1996 J. Z. Li, M. T. Ozsu, and D. Szafron. Modelling of video spatial relationships in an object database management system, *Proc. Int'l. Workshop on Mult. Dat. Man. Sys*, pp. 124-132, 1996.

LiOS1997 J. Z. Li, M. T. Ozsu, and D. Szafron. Modelling of moving objects in a video database, *Proc. IEEE Int'l Conf. on Mult. Comp. and Systems*, pp. 336-343, June 1997.

LiSe2001 B. Li and M. I. Sezan. Event detection and summarization in sports video, *IEEE Workshop on Content-Based Access of Image and Video Libraries* (CBAIVL 2001), pp. 132-138, 2001.

LiSe2002 B. Li and M. I. Sezan. Event detection and summarization in American football broadcast video, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases vol. 4676*, pp. 202-213, Jan. 2002.

LiWC1998 Liu Z. , Y. Wang, and T. Chen. Audio feature extraction and analysis for scene segmentation and classification, *J. of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 20, pp. 61-80, 1998.

LiWe2002 R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos, *IEEE Trans. on Circuits and Sys. for Video Tech.*, vol. 12, no. 4, pp. 256-268, Apr. 2002.

LiYT1993 Z. N. Li, B. Yao, and F. Tong. Linear generalized Hough transform and its parallelization, *Image Vision Computing*, vol. no. 1, pp. 11-24, 1993.

Low1992 D. Lowe. Robust model based motion tracking through the integration of search and estimation, *J of Computer Vision*, vol. 8, pp. 113-122, 1992.

LSDM2001 D. Li, I. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval, *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533-544, 2001.

LuJZ2001 L. Lu, H. Jiang, and H. J. Zhang. A robust audio classification and segmentation method, *Proc. ACM Multimedia* (ACM MM 2001), pp. 203-211, Oct.2001.

LVWC1998 M. Lazarescu, S. Venkatesh, G. West, and T. Caelli. Combining NL processing and video data to query American football, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 1998), pp. 16-20, Aug. 1998.

MaBl1999 J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects, *Proc. Int'l Conf on Computer Vision* (ICCV 1999), pp. 572-578, 1999.

MBCM1999 E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach, *Proc. Int'l Conf on Computer Vision* (ICCV 1999), pp. 262-268, 1999.

McHo1993 F. R. McFadden and J. A. Hoffer. Modern database management, *The Benjamin/Cummings Pub. Co.*, Inc., Redwood City, CA, 1993.

MeNg1999 T. Meier and K. N. Ngan. Video segmentation for content-based coding, *IEEE Trans. Circ. Syst. Video Tech.* vol. 9, pp. 1190-1201, Dec. 1999.

MHBK2002 S. Miyauchi, A. Hirano, N. Babaguchi, and T. Kitahashi. Collaborative multimedia analysis for detecting semantical events from broadcasted sports video, *Proc. of Int'l Conf on Pattern Recognition* (ICPR 2002), vol. 2, pp.1009-1012, 11-15 Aug. 2002.

MIAT1998 K. Matsui, M. Iwase, M. Agata, T. Tanaka, and N. Ohnishi. Soccer image sequence computed by a virtual camera, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1998), 1998.

MiCL2001 A. Mittal, L. F. Cheong, and T. S. Leung. Dynamic Bayesian framework for extracting temporal structure in video, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2001), pp.110-115, 2001.

MiIi2000 H. Miyamori and S. L. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge, *Proc Int'l Conf. Automatic Face and Gesture Recognition 2000, IEEE CS Press*, Los Alamitos, Calif, 2000.

Miy2002 H. Miyamori. Improving accuracy in behaviour identification for content-based retrieval by using audio and video information**,** *Proc. of Int'l Conf on Pattern Recognition* (ICPR 2002), vol. 2, pp. 826 - 830, 11-15 Aug. 2002.

Miy2003 H. Miyamori. Automatic Annotation of Tennis Action for Content-based Retrieval by Collaborating Audio and Visual Information, *IEICE Trans. D-II*, vol. J86-D-II, no.4, pp.511-524, 2003.

MNZI2002 T. Misu, M. Naemura, W. Zheng, Y. Izumi, and K. Fukui. Robust tracking of soccer players based on data fusion, *Proc. of Int'l Conf on Pattern Recognition* (ICPR 2002), vol. 1, pp. 10556-10561, Quebec City, QC, Canada, August 11 - 15, 2002.

MSSO2000 K. Matsumoto, S. Sudo, H. Saito, and S. Ozawa. Optimized camera view point determination system for soccer game broadcasting, *IAPR Workshop on Machine Vision Application*, pp. 3-22, 2000.

MuNi1991 H. K. Muammar and M. Nixon. Tristage Hough transform for multiple ellipse extraction, *IEE Proc. Comput. Digital Tech.* vol. 138, no. 1, pp. 27-35, 1991.

NaHu2001 M. R. Naphadem and T. S. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval, *IEEE Trans. on Multimedia*, vol. 3, no. 1, pp. 141-151, 2001.

NaKa1997 Y. Nakamura and T. Kanade. Semantic analysis for video contents extraction - spotting by association in news video, *Proc. ACM Multimedia* (ACM MM 1997), pp. 393-401, 1997.

NCRT1998 A. Neri, S. Colonnese, G. Russo, and P. Talone. Automatic moving object and background separation, *Signal Proc*., vol. 66, pp. 219-232, Apr. 1998.

Nee2003 C. J. Needham. Tracking and modelling of team game interactions, *PhD Dissertation*, School of Computing, The University of Leeds, UK, 2003.

NeSR2001 S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos, *Proc. ACM Multimedia* (ACM MM 2001), pp. 261-269, 2001.

NgPZ2001 C. W. Ngo, T. C. Pong, and H. J. Zhang. On clustering and retrieval of video shots, *Proc. ACM Multimedia* (ACM MM 2001), pp. 51-60, 2001.

NgWB2001 H. T. Nguyen, M. Worring, and R. Boomgaard. Occlusion robust adaptive template tracking, *Proc. Int'l Conf on Computer Vision* (ICCV 2001), July 7-14, 2001.

NgZP2001 C. W. Ngo, H. J. Zhang, and T. C. Pong. Recent advances in content-based video analysis, *J. of Image and Graphics*, vol. 3, no. 1, pp.445-468, 2001.

NiCo1993 L. Nigay and J. Coutaz. A design space for multimodal systems: Concurrent processing and data fusion, *INTERCHI'93 Proc*. 1993.

NiPe1988 W. Niblack and D. Petkovic. On improving the accuracy of the Hough transform: Theory, simulations and experiments, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1998), pp. 574-579, June. 1988.

OdBo1995 J. M. Odobez and P. Bouthemy. Multiresolution estimation of parametric motion models, *J. of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348-365, 1995.

OhMS1999 Y. Ohno, J. Miura, and Y. Shirai. Tracking players and a ball in soccer games, *Int'l Conf. On Multisensor Fusion and Integration for Intelligent Sys*., Taipei, Taiwan, Aug. 1999.

OhMS2000 Y. Ohno, J. Miura, and Y. Shirai. Tracking players and estimation of 3D position of a ball in soccer games, *Proc. Int'l Conf. on Pattern Recognition* (ICPR 1999), vol. 1, pp.145-148, Sept. 3-7, 2000.

Ols1998 C. F. Olson. Improving the generalized Hough transform through imperfect grouping, *Image and Vision Computing*, vol. 16, pp. 627-634, 1998.

Ols1999 C. F. Olson. Constrained Hough transform for curve detection, *Computer Vision Image Understanding*, vol. 58, pp.329-345, 1999.

Ozy1999 E. Ozyildiz. Adaptive texture and color segmentation for tracking moving objects, *Master thesis*, Pennsylvania State University, 1999.

PaBS2001 H. Pan, P. van Beek, and M. I. Sezan. Detection of slow-motion segments in sports video for highlights generation, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2001), 2001.

PaLS2002 H. Pan, B. Li, and M. I. Sezan. Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2002), 2002.

Par1997 J. R. Parker. Algorithms for image processing and computer vision, *John Wiley & Sons Inc.* 1997.

PfLE2001 S. Pfeiffer, R. Lienhart, and W. Effelsberg. Scene determination based on video and audio features, Multimedia Tools and Applications, vol. 15, no. 1, pp. 59-81, 2001.

PiJC1998 G. S. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1998), pp. 260-265, 1998.

PiMi1995 R. W. Picard and T. P. Minka. Visual texture for annotation, *Multimedia Systems*, Springer Verlag, vol.3, no. 1 pp. 3-13, 1995.

PMJD2002 M. Petkovic, V. Mihajlovic, W. Jonker, and S. Djordjevis-Kajan. Multi-modal extraction of highlights from TV Formula 1 programs, *Proc. IEEE Int'l Conf. on Multimedia and Expo*, (ICME 2002), pp. 817-820, 2002.

POJC2002 G. S. Pingali, A. Opalach, Y. D. Jean, and I. B. Carlbom. Instantly indexed multimedia databases of real world events, *IEEE Trans. Multimedia,* vol. 4, pp. 269-282, June 2002.

PoKV1998 M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, *Proc. Int'l Conf on Computer Vision* (ICCV 1998), 1998.

PrIK1994 J. Princen, J. Illingworth and J. Kittler. Hypothesis testing: a framework for analyzing and optimizing Hough transform performance, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 16, no. 4, pp. 329-341, Apr 1994.

PYIK1989a J. Princen, H. K. Yuen, J. Illingworth and J. Kittler. A comparison of Hough transform methods, *Proc. IEE 3rd Int'l Conf. on Image Processing and Its Application*, University of Warwick, July 1989.

QiTo2001 R. J. Qian and V. Tovinkere. Detecting semantic events in soccer games: Towards a complete solution, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2001), pp.1040-1043, 2001.

RaHa1998 C. Rasmussen and G. Hager. Joint probabilistic techniques for tracking multi-part objects, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 1998), pp. 16-21, 1998.

RaHa2001 C. Rasmussen and G. Hager. Probabilistic data association methods for tracking complex visual objects, *IEEE Trans. on Pattern Anal. Machine Intel.*, pp. 560-576, 2001.

Rei1979 D. B. Reid. An algorithm for tracking multiple targets, *IEEE Trans. on Automatic Control*, Dec. 1979.

ReNS2001 G. Reynolds, S. Nepal, and U. Srinivasan. Automatic detection of 'goal' segments in basketball videos, *Proc of ACM Multimedia* (ACM MM 2001), pp. 261-269, 2001.

ReZi1996 I. Reid and A. Zisserman. Goal-directed video metrology, *European Conf. on Comp. Vision* (ECCV 1996), pp. 647-658, 1996.

RPTB2001 Y. Rubner, J. Pusicha, C. Tomasi, and J. M. Buhmann. Emprical evaluation of dissimilarity measures for color and texture, *Computer Vision and Image Understanding,* vol. 84, pp. 25-43, 2001.

RuAn2000 Rui Y. and P. Anandan. Segmenting visual action units based on spatial-temporal motion patterns, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2000), 2000.

RuGA2000 Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for TV baseball programs, *Proc. of ACM Multimedia* (ACM MM 2000), pp. 105-115, 2000.

RuTG1998 Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases, *Proc. of International Conference on Computer Vision* (ICCV 1998), pp. 59-66, 1998.

SaAy1996 H. Sawhney and S. Ayer. Compact representation of videos through dominant and multiple motion estimation, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 18 pp. 814-830, Aug. 1996.

SaGe2000 L. A. E. Al Safadi and J. R. Getta. Semantic modelling for video content-based retrieval systems, *Austral Asian Comp. Science Conf.* (ACSC 2000), pp. 2-9, 2000.

SCKH1997 Y. Seo, S. Choi, H. Kim, and K. Hong. Where are the ball and players? Soccer game analysis with color based tracking and image mosaic, *Proc. Int'l Conf. on Image Analysis and Processing*, Florence, Italy, pp. 196-203, Sept. 17-19, 1997.

Sha1975 S. D. Shapiro. Transformations for the computer detection of curves in noisy pictures, *Comput. Graphics Image Process*, vol. 4, 1975.

Sha1995 B. Shahraray. Scene change detection and content-based sampling of video sequences, *Symp. Electronic Imaging: Science and Technology: Digital Video Compression, Algorithms and Technologies*, pp. 2-13, Feb. 1995.

ShDY1996 D. Shaked, O. Yaron, and N. Kiryati. Deriving stopping rules for the probabilistic Hough transform by sequential analysis, *Computer Vision Image Understanding*, vol. 63, pp. 512-526, 1996.

ShFo1988 Y. Bar-Shalom and T. Fortmann. Tracking and data association, *Academic Press*, 1988.

Shi1989 M. Shiono. Comparison experiments of three kinds of table look-up methods for Hough transform computation, *Trans. Inst. Electron. Inform. Commun. Eng. D-11*, Japan 72(6), pp. 963-966, 1989.

ShIa1979 S. D. Shapiro and A. Iannino. Geometric constructions for the predicting Hough transform performance, *IEEE Pattern Anal. Mach. Intell.*, 1979.

ShKo2004 H. Shum and T. Komura. A spatiotemporal approach to extract the 3D trajectory of the baseball from a single view video sequence, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2004), Taibei, Taiwan, 2004.

ShWe1990 C. J. Sheng and H. T. Wen. Fast generalized Hough transform, *Pattern Recognition Letters*, vol.11, no. 11, pp. 725-733, 1990.

SiDH1984 T. M. Silberberg, L. Davist, and D. Harwood. An iterative Hough procedure for three-dimensional object recognition, *Pattern Recognition*, vol. 17 (6), pp. 621-629, 1984.

Sil1986 B. W. Silverman. Density estimation for statistics and data analysis, *Chapman & Hall*, 1986.

SmBe2000 J. R. Smith and A. B. Benitez. Conceptual modeling of audio-visual content, *Proc. of IEEE Conf. on Multimedia & Expo* (ICME2000), vol. 2, pp. 915-918, 2000.

SmBu1975 P. Smith and G. Buechler. A branching algorithm for discriminating and tracking multiple objects, *IEEE Trans. Autom. Contr.* Vol. 20, pp. 101-104, 1975.

SmZh1994 S. W. Smoliar and H. J. Zhang. Content-based video indexing and retrieval, *IEEE Multimedia*, vol. 1, pp. 62-74, Summer 1994.

SnWo2002 C.G.M. Snoek and M. Worring, A review on multimodal video indexing, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2002), Lausanne, Switzerland, August 2002.

SnWo2003 C.G.M. Snoek and M. Worring, Time Interval Maximum Entropy based Event Indexing in Soccer Video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2003), vol. 3, pp. 481-484, Baltimore, USA, July 2003.

SnWo2004 C.G.M. Snoek and M. Worring, Multimodal Video Indexing: A Review of the State-of-the-art, *Multimedia Tools and Applications*, In Press, 2004.

SoDe2002 http://www.decatursports.com/articles/soc/soccerterms.htm, Soccer Dictionary, 2002.

Sri1995 R. K. Srihari. Automatic indexing and content-based retrieval of captioned images, *Int'l Conf on Document Analysis and Recognition*, vol. 2, pp. 1165-1168, August 14 - 15, 1995.

SrPP1999 Srinivasan, S. D. Petkovic, and D. Ponceleon. Towards robust features for classifying audio in the cuevideo system, *Proc. ACM Multimedia* (ACM MM 1999), 1999.

Ste1991 R. S. Stephens. Probabilistic approach to the Hough transform, *Image and Vision Computing*, vol. 9, no. 1, 1991.

STKR1997 D. D. Saur, Y-P. Tan, S. R. Kulkarni, and P. J. Ramadge. Automated analysis and annotation of basketball video, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases, vol. 3022,* pp. 176-187, Jan. 1997.

SuCh2001 H. Sundram and S. F. Chang. Constrained utility maximization for generating visual skims, *IEEE Workshop on content based Access of image and video libraries* (CBAIVL-2001), Kauai, HI USA, Dec. 2001.

SuLJ1998 F. Sudhir, J. C. M. Lee, and A. K. Jain. Automatic classification of tennis video for high-level content-based retrieval, *Proc. Int'l workshop on Content-based Access of Image and Video Databases* (CAIVD 1998), pp. 81-90, 1998.

Sun2002 H. Sundram. Segmentation, structure detection and summarization of multimedia sequences, *PhD Dissertation*, Dept of Electrical Engineering, Columbia University NY, Aug.2002.

SwBa1991 M. J. Swain and D. H. Ballard. Color indexing, *J. of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

TaHF1996 T. Taki, J. Hasegawa, and T. Fukumura. Development of motion analysis system for quantitative evaluation of teamwork in soccer games, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 1996), pp. 815-818, 1996.

Tho1992 A. D. H. Thomas. Compressing the parameter space of the generalized Hough transform, *Pattern Recognition Letters*, vol.13, no. 2, pp.107-112, 1992.

ToQi2001 V. Tovinkere and R. J. Qian. Detecting semantic events in soccer games: Towards a complete solution, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2001), pp.1040-1043, 2001.

TsMa1979 S. Tsuji and F. Matsumoto. Detection of ellipse by a modified Hough transformation, IEEE Trans. on Computer, pp. 777-781, 1979.

UMIS2002 O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka. An object detection method for describing soccer games from video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2002), pp. 45-48, 2002.

VeBE1996 D. D. Velthausz, C. M. R. Bal, and E. H. Eertink. A multimedia information object model for information disclosure, *Proc Int'l Conf on Multimedia Modeling* (MMM 1996), Toulouse, France, pp. 289-304, 12-15, Nov. 1996.

WLXY2003 K. W. Wan, J. H. Lim, C. Xu, and X. Yu. Real-time camera field-view tracking in soccer video, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2003), 2003.

WVPM1998 A. Woudstra, D. D. Velthausz, H. J. G. de Poot, F. Moelaert El-Hadidy, Willem Jonker, Maurice A. W. Houtsma, R. G. Heller, and J. N. H. Heemskerk: Modeling and retrieving audiovisual information: A soccer video retrieval system, *Multimedia Information Systems*, pp. 161-173, 1998.

WuHu2001 Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking, *Proc. of Int'l conf. on Computer Vision*, (ICCV 2001), pp. 26-33, 2001

WYYX2003a K. W. Wan, X. Yan, X. Yu, and C. Xu. Real-time goalmouth detection in mpeg soccer video, *Proc. of ACM Multimedia* (ACM MM 2003), pp.311-314, 2003.

WYYX2003b K. Wan, X. Yan, X. Yu, and C. Xu. Robust goalmouth detection for virtual content insertion, *Proc. of ACM Multimedia* (ACM MM 2003), pp468-469, 2003.

WXCY2004 J. Wang, C. Xu, E. Chng, X. Yu and Q. Tian. Event detection based on non-broadcast sports video, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2004), 2004.

WaRe1990 H. L. Wang and A. P. Reeves. Three-dimensional generalized Hough transform for object identification, *J. Soc. Photo-Opt. Instrum. Eng.* 1192(1), pp. 363-374, 1990.

XCDS2002 L. Xie, S. F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden Markov models, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing*, (ICASSP 2002), pp. 4096-4099, 2002.

XDXT2003 M. Xu, L. Y. Duan, C. Xu, and Q. Tian. A fusion scheme of visual and auditory modalities for event detection in sports video, *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 2003), pp.189-192, 2003.

XiJi2002 Y. Xie and Q. Ji. A new efficient ellipse detection method, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 2002), vol. 2, pp. 957-960, 2002.

XiRD2003 Z. Xiong, R. Radhakrishnan, and A. Divakaran. Generation of sports highlights using motion activity in combination with a common audio feature extraction framework, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2003), vol. 1, pp. 5-8, 2003.

XMXK2003 M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian. Creating audio keywords for event detection in soccer video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2003), vol. II, pp.281-284, 2003.

XuOj1993 L. Xu and E. Oja. Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities, *CVGIP: Image Understanding*, vol. 57, no. 2, pp. 131-154, 1993.

XuOj2001 L. Xu and E. Oja. Further developments on RHT: Basic mechanisms, algorithms and computational complexities, *Proc. IEEE Int'l Conf. on Pattern Recognition* (ICPR 2001), vol. 1, pp. 125-128, 2001.

XuOK1990 L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT), *Pattern Recognition Letters*, vol. 11, pp. 331-338, 1990.

XXCD2001 P. Xu, L. Xie, S. F. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and system for segmentation and structure analysis in soccer video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2001), Aug 22-25, 2001.

YaKA2002 M. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey, *IEEE Trans. on Pattern Anal. Machine Intel.*, vol. 24, pp. 34-58, 2002.

YaSM2002 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, *Proc. of Int'l Conf. on Pattern Recognition* (ICPR 2002), 2002.

YaLC2004 Y.-Q. Yang; Y-D Lu, and W. Chen. A framework for automatic detection of soccer goal event based on cinematic template, *Proc. of Int'l Conf. on Machine Learning and Cybernetics,* vol 6, pp. 3759–764, 26-29 Aug. 2004.

YaYH2004 X. Yan, X. Yu, and T. S. Hay. A 3D reconstruction and enrichment system for broadcast soccer video, *Proc. ACM Multimedia* (ACM MM 2004), New York, USA.

YIIM1989 J. Yamoto, K. Irisawa, I. Ishii, and H.Makino. Algorithm for extracting ellipses using weighted center points map, *Trans Inst. Electron. Inform. Commum. Eng. D-11*, Japan, vol. 72, no. 7, pp. 1009-1016, 1989.

YiTL1992 R. K. K. Yip, P. K. S. Tam, and D. N. K. Leung. Modification of Hough transform for circle and ellipse detection using a two-dimensional array, *Pattern Recognition*, vol. 25, no. 9, pp. 1007-1022, 1992.

YLLT2003 X. Yu, H. W. Leong, J. H. Lim, Q. Tian, and Z. Jiang. Team possession analysis for broadcast soccer video based on ball trajectory, *Proc. IEEE Pacific-rim Conference on Multimedia* (PCM 2003), pp. 1811-1815, Singapore, December 15-18, 2003.

YLXT2004a X. Yu, H. W. Leong, C. Xu, and Q. Tian. A robust Hough-based algorithm for partial ellipse detection in broadcast soccer video. *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2004), Taibei, Taiwan, 2004.

YLXT2004b X. Yu, H. W. Leong, C. Xu, and Q. Tian. A robust and accumulator-free ellipse Hough transform, *Proc. ACM Multimedia* (ACM MM2004), New York, USA.

YSWC2004 X. Yu, C. H. Sim, J. R. Wang, and L. F. Cheong. A trajectory-based ball detection and tracking algorithm in broadcast tennis video, *Proc. IEEE Int'l Conf. on Image Processing* (ICIP 2004), 2004.

Yue1991 S. Y. K. Yuen. Connective Hough transform, *Proc. British Machine Vision Conference*, Glasgow, 1991.

YuIK1989 H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the Hough transform, *Image and Vision Computing*, vol. 7, pp. 31-37, 1989.

YuTW2003 X. Yu, Q. Tian and K. W. Wan. A novel ball detection framework for real soccer video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2003), vol. II, pp. 265-268, 2003.

YWXT2003 X. Yu, K. W. Wan, C. Xu, Q. Tian, and H. W. Leong. An accurate ball detection and tracking system for broadcast soccer video, *Technical Demo Description of ICME 2003* (ICME 2003), 2003.

YXLT2003 X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, *Proc. ACM Multimedia* (ACM MM 2003), pp. 11-20, 2003.

YXTL2003 X. Yu, C. Xu, Q. Tian, and H. W. Leong. A ball tracking framework for broadcast soccer video, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2003), vol. II, pp. 273-276, 2003.

YXTY2003 X. Yu, C. Xu, Q. Tian, X. Yan, K. W. Wan, and Z. Jiang. Estimation of the ball size in broadcast soccer video, *Proc. IEEE Pacific-rim Conference on Multimedia* (PCM 2003), pp. 929-934, Singapore, Dec. 15-18, 2003.

YYHL2004 X. Yu, X. Yan, T. Z. Hay, and H. W. Leong. 3D reconstruction and enrichment of broadcast soccer video, *Proc. ACM Multimedia* (ACM MM2004), New York, USA.

YYWL1997 M. M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu. Video browsing and scene transitions on compressed sequences, *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases IV.* 1997, IS&T/SPI97.

YYYL1995 D. Yow, B. L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlight from digital video, *Proc. Asian Conf on Computer Vision* (ACCV 1995), pp.499-503, 1995.

ZhCh1997 D. Zhong and S. F. Chang. Video object model and segmentation for content-based video indexing, *IEEE Int'l Sym. on Circuits and Systems* (ISCAS 1997), Hong Kong, June 1997, Special Session on Networked Multimedia Technology and Application.

ZhCh1999 D. Zhong and S. F. Chang. An integrated approach for content-based video object segmentation and retrieval, *IEEE Trans. on Circuits and Sys. for Video Tech.* vol. 9, no. 8, pp. 1259-1268, Dec. 1999.

ZhCh2001 D. Zhong and S. F. Chang. Structure analysis of sports video using domain models, *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME 2001), pp. 920-923, 2001.

ZhCh2002 D. Q. Zhang and S. F. Chang. Event detection in baseball video using superimposed caption recognition, *Proc. of ACM Multimedia* (ACM MM 2002), pp. 315-318, 2002.

ZhFa1992 Z. Zhang and O. D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames, *J. of Computer Vision*, vol. 7, no. 3, pp. 211-241, 1992.

ZhNe2001 T. Zhao and R. Nevatia. Car detection in low resolution aerial image, *Proc. Int'l Conf. on Computer Vision* (ICCV 2001), July 7-14, 2001.

Zho2001 D. Zhong. Segmentation, indexing and summarization of digital video content, *PhD Dissertation*, Dept of Electrical Engineering Columbia University, NY, Jan. 2001.

ZhVK2000 W. Zhou, A. Vellaikal, and C. C. J. Kuo. Rule-based video classification system for basketball video indexing, *Proc ACM Multimedia 2000 Workshops*, ACM Press, New York, pp. 213-216, 2000.

ZLSW1995 H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution, *Proc. ACM Multimedia* (ACM MM 1995), pp.15-24, 1995.

Related Published Papers

I First-Author Papers

1. Xinguo Yu, Changsheng Xu, Hon Wai Leong, Qi Tian, Qing Tang, and Kong Wah Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, *Proc. of ACM MM 2003*, pp.11-20.

2. Xinguo Yu, Qi Tian and Kong Wah Wan. A novel ball detection framework for real soccer video, *Proc. ICME 2003*, Vol II, pp. 265-268.

3. Xinguo Yu, Changsheng Xu, Qi Tian, and Hon Wai Leong. A ball tracking framework for broadcast soccer video, *Proc. ICME 2003*, Vol II, pp. 273-276.

4. Xinguo Yu, Kong Wah Wan, Changsheng Xu, Qi Tian, and Hon Wai Leong. An accurate ball detection and tracking system for broadcast soccer video, *Technical Demo Description of ICME03*, Baltimore, US, 2003.

5. Xinguo Yu, Changsheng Xu, Qi Tian, Xin Yan, Kong Wah Wan and Zhenyan Jiang. Estimation of the ball size in broadcast soccer video using salient objects, *Proc. of PCM 2003*. pp. 929-934.

6. Xinguo Yu, Hon Wai Leong, Joo Hwee Lim, Qi Tian, and Zhenyan Jiang. Team possession analysis for broadcast soccer video based on ball trajectory, *Proc. of PCM 2003*, pp. 1811-1815.

7. Xinguo Yu, Hon Wai Leong, Changsheng Xu, and Qi Tian. A robust Hough-based algorithm for partial ellipse detection in broadcast soccer video, *Proc. of ICME04*, vol.3, pp. 1555-1558, Taibei, Taiwan, June, 27-30 June 2004.

8. Xinguo Yu, Chern-Horng Sim, Jenny Ran Wang, and Loong Fah Cheong. A trajectory-based ball detection and tracking algorithm in broadcast tennis video. *Proc. of ICIP04, pp1049-1052, 24-27 October, Singapore*.

9. Xinguo Yu, Hon Wai Leong, Changsheng Xu, and Qi Tian. A robust and accumulator-free ellipse Hough transform, *Proc. ACM MM04, pp256-259, Columbia U, New York, USA*.

10. Xinguo Yu, Xin Yan, Tze Sen Hay, Hon Wai Leong. 3D reconstruction and enrichment of broadcast soccer video, *Proc. ACM MM04 pp260-263, Columbia U, New York, USA*.

II. Co-author Papers

1. Kong Wah Wan, Joo Hwee Lim, Changsheng Xu, Xinguo Yu. Real-time camera field view tracking in soccer video, *Proc. of ICASSP 2003*, pp. 185-188, 2003.

2. Kong Wah Wan and Xinguo Yu. An efficient annotation system for soccer video, *Technical demo of ICME 2003*, Baltimore, USA, 2003.

3. Kong Wah Wan, Xin Yan, Xinguo Yu, and Changsheng Xu. Real-time goalmouth detection in mpeg soccer video, *Proc. of ACM MM 2003*, pp. 311-314.

4. Kong Wah Wan, Xin Yan, Xinguo Yu, and Changsheng Xu. Robust goalmouth detection for virtual content insertion, *Proc. of ACM MM 2003*, pp. 468-469.

5. Jin Jun Wang, Changsheng Xu, Eng Siong Chng, Xinguo Yu and Qi Tian. Event Detection based on non-broadcast sports video, *Proc. ICIP04, pp1637-1640, 24-27 October, Singapore*.

6. Xin Yan, Xinguo Yu, Tze Sen Hay. A 3D reconstruction and enrichment system for broadcast soccer video, *Proc. ACM MM04 pp746-747, Columbia U, New York, USA*.

Appendix A

Use of Kalman Filter

The state equation is described by the following linear equation:

$$X_{k+1} = A_k X_k + W_k \qquad \text{(A.1)}$$

where X_k is the *state vector* at time k, W_k is the *system noise* and A_k is the *state transition matrix*. The *measure vector* Z_k is related to the state vector via the measure equation:

$$Z_k = I_k X_k + V_k \qquad \text{(A.2)}$$

where I_k is the *measurement matrix* and V_k is the *noise measure matrix*.

In the ball motion, the state will include the x and y coordinates of the ball, the velocity components of the ball v_x and v_y, and the acceleration components of the ball a_x and a_y. So the state at any point in time can be represented with the vector $(x, y, v_x, v_y, a_x, a_y)^T$. The state transition matrix is derived from the theory of motion under constant acceleration which can be expressed with the equations.

$$\begin{cases} x(k+1) = x(k) + v_x(k) + \frac{1}{2}a_x(k), \\ y(k+1) = y(k) + v_y(k) + \frac{1}{2}a_y(k), \\ v_x(k+1) = v_x(k) + a_x(k), \\ v_y(k+1) = v_y(k) + a_y(k), \\ a_x(k+1) = a_x(k), \\ a_y(k+1) = a_y(k). \end{cases} \qquad \text{(A.3)}$$

Therefore, in equation A.1, the system noise matrix is initialized to 0, and the state transition matrix, A_0, is initialized as follows:

$$A_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A.4)}$$

The system noise covariance matrix (used in updating W_k) is initialized as follows:

$$Q_0 = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{5} \end{bmatrix} \quad \text{(A.5)}$$

The measure matrix, I_k, we use is independent of k and is given by:

$$I_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{(A.6)}$$

The initial value for the noise measure matrix, V_0, is given by:

$$V_0 = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \text{(A.7)}$$

To compute the initial state vector, X_0, we look for two ball candidates that are "close by" in two consecutive frames. Let (x_0,y_0) and (x_1,y_1) be the positions of the ball candidates in the two frames, respectively. We use these to compute the initial velocity for the ball candidate. We initialize the acceleration to 0. Then, the initial state vector, X_0, is given by:

$$X_0 = (x_1, y_1, x_1 - x_0, y_1 - x_0, 0, 0)^{\mathrm{T}} \quad \text{(A.8)}$$

The author wishes to thank his colleague, Mr. Yang Minjiang, for help with the use of Kalman filter in OpenCV.

Appendix B

Sequences and Symbols of the Test Video

The test video is a whole video of the first half of the game between Senegal and Turkey (FIFA2002), in which the game starts at the frame 06310. We first segment the video into the sequences with the soccer field and we obtain 139 sequences with soccer field. At the same time, we obtain 138 sequences without the soccer field because there is a sequence without the soccer field between each pair of adjacent sequences with the soccer field. Among the sequences with soccer field, there are 15 replay sequences that are detected by detecting the moving replay sign, which is a large object, using algorithm presented in [PaLS2002]. In addition, there are 56 ball-less sequences. A sequence is called a ball-less sequence if the whole sequence does not contain any ball at all. The remaining 68 sequences are classified into three classes according to the number of frames in sequences: S-length (21-300 frames), M-length (301-1000 frames), and L-length (longer than 1000 frames). All the sequences with the soccer field and their symbols are tabulated in Table B.1. In the table, “~ball” means “ball-less”; the numbers after a symbol indicate the start and end frames of the corresponding sequence.

Table B.1 Sequences with the soccer field and their symbols of the test video (FIFA2002 quarter-final Senegal vs Turkey).

Type	Symbols and Their Frames
Replay	R01: 07478-07608, R02: 14002-14143, R03: 23374-23635, R04: 31506-31679, R05: 34646-34929, R06: 36139-36608, R07: 37855-38058, R08: 40289-40746, R09: 43317-43482, R10: 47400-47642, R11: 48987-49306, R12: 63519-63723, R13: 63728-63772, R14: 71321-71508, R15: 71529-71577
~ball	X01: 06816-07011, X02: 07028-07073, X03: 07727-07874, X04: 08189-08267, X05: 10405-10485, X06: 13385-13557, X07: 14348-14548, X08: 17557-17630, X09: 17639-17671, X10: 23325-23366, X11: 24323-24393, X12: 24944-25038, X13: 25075-25098, X14: 25829-25983, X15: 27546-27593, X16: 27606-27678, X17: 27689-27747, X18: 29340-29365, X19: 29371-29397, X20: 29429-29455, X21: 33444-33474, X22: 34186-34207, X23: 34938-34971, X24: 36026-36091, X25: 36096-36129, X26: 38865-39027, X27: 39035-39068, X28: 40225-40279, X29: 42456-42717, X30: 42876-42943, X31: 43162-43307, X32: 47012-47239, X33: 48938-48980, X34: 49826-49860, X35: 49945-50016, X36: 50146-50243, X37: 50313-50445, X38: 50448-50484, X39: 50494-50701, X40: 52121-52196, X41: 57441-57639, X42: 57979-58143, X43: 58178-58251, X44: 58322-58521, X45: 58533-58844, X46: 58856-58917, X47: 66867-67489, X48: 67701-67743, X49: 71257-71314, X50: 71630-71714, X51: 72960-73226, X52: 74183-74206, X53: 74233-74419, X54: 74554-74673, X55: 77064-77471, X56: 77473-77507
S-length	S01: 07203-07317, S02: 08347-08557, S03: 08792-08924, S04: 12403-12622, S05: 17353-17546, S06: 31148-31386, S07: 33479-33610, S08: 35667-35954, S09: 39073-39185, S10: 42996-43160, S11: 52379-52571, S12: 52573-52635, S13: 72799-72951, S14: 73877-74169
M-length	M01: 06300-06723, M02: 07880-08181, M03: 08974-09684, M04: 09819-10399, M05: 10490-11433, M06: 11457-11964, M07: 12650-13377, M08: 13588-13943, M09: 17876-18646, M10: 18764-19217, M11: 19340-19857, M12: 19859-20200, M13: 23643-24083, M14: 24413-24899, M15: 25104-25793, M16: 26276-26695, M17: 26698-27493, M18: 29706-30271, M19: 30386-31110, M20: 31689-32248, M21: 33676-34080, M22: 34216-34549, M23: 35197-35614, M24: 36615-37408, M25: 37432-37842, M26: 38066-38857, M27: 39230-40213, M28: 40783-41158, M29: 43733-44316, M30: 47909-48859, M31: 49320-49719, M32: 55540-56359, M33: 56412-57385, M34: 59204-59916, M35: 70780-71211, M36: 71821-72791, M37: 73232-73834, M38: 74830-75523, M39: 75552-76334, M40: 76356-77057
L-length	L01: 14556-17351, L02: 20227-23272, L03: 27785-29256, L04: 32257-33400, L05: 41174-42450, L06: 44346-45525, L07: 45533-47004, L08: 50944-52024, L09: 52652-55391, L10: 60078-61650, L11: 61766-63511, L12: 63784-65202, L13: 65204-66720, L14: 67748-70744

For the whole test video, the number of the sequences of each type and their total frames are tabulated in Table B.2, in which “~field” means “without the soccer field”.

Table B.2 Distribution of various types of the sequences in the test video (FiFA 2002 quarter-final Senegal vs Turkey).

Type	# of seg.	# of frames	Symbols of seg.
~Field	138	8585	Not Available
Replay	15	3341	R01 – R15
~Ball	56	6385	X01-X56
S-length	14	2512	S01-S14
M-length	40	24465	M01-M40
S-length	14	25460	L01-L14
Total	177	70748	Not Applicable