Context-Aware Framework for Analysis of Horse-Race Videos

A thesis

submitted in fulfilment

of the requirements for the degree

of

Doctor of Philosophy in Electronic

at

The University of Waikato

by

Mohammad Hedayati

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

October 2018

Abstract

Today sports analysis systems draw the attention of many commercial entities and are providing many opportunities for computer vision researchers to study and develop automated sports analysis systems. The primary objective of a sports video analysis is to interpret low-level visual features to high-level semantics that can be understood by the end user. Although all sports video analysis systems are based on similar principles at the low-level visual extraction, interpreting this visual information to high-level human understanding is extremely specific to the sport in question.

This project is the very first attempt to evaluate the jockeys' performance from horse-race TV broadcasts. The aim of this thesis is to extract high-level information from the horse race by automatically detecting and tracking jockeys, specifically at turning points of the race. However, detecting and tracking of jockeys is extremely difficult. This is because in most horse-races there are more than six jockeys and they closely follow each other to gain a leading position, consequently locating jockeys and maintaining their identities through a video sequence is highly challenging due to the frequent occlusion. In addition, the background in horse-race videos is continually changing, thus there is not only background clutter but the jockeys themselves may be obscured by obstacles such as trees, towers and bars.

To tackle these challenges, a context-aware analysis system is proposed. The proposed system is developed based on deterministic reasoning which was obtained by observing various horse races. One important property of horse-races is the group dynamic behaviour of jockeys in the race. The jockeys race around a circular track and the camera typically follows the jockeys, thus the jockeys rel-

ative to each other tend to move as a slowly changing group. This homogeneous characteristic of jockeys in the race is very useful, especially when local information of jockeys is poor or abrupt due to occlusion or background clutter.

The proposed system combines multiple cues, such as local jockeys' information, background characteristics and group dynamic properties, to detect and track jockeys around a turning segment of a horse-race. We demonstrate that our proposed system can be generalized to work on other domains. The outline of the proposed model contains three modules; (1) scene analysis (2) jockey detection and, (3) tracking and data association. The main objective of scene analysis is to extract the turning segment in the race. The detection of jockeys is accomplished by determining the location of each jockey's cap. For the tracking of the jockeys, we implement a robust hierarchical tracking scheme which iteratively adapts and updates itself by gathering the jockeys' properties at each of the point, object and group levels. To boost the tracking accuracy, data association is applied as a multi-target management system to maintain multiple jockeys' identities over time, to initialise the tracking and to terminate trajectories.

Acknowledgements

I wish to express my appreciation to the many people who have helped and supported me throughout this research. Firstly, I would like to convey my sincere gratitude to my chief supervisor Dr. Michael J. Cree for his continued guidance, support and advice through the duration of this project. Thank you for the ideas and critiques along the way. My sincere appreciation and gratitude to my mentor and co-supervisor Professor Jonathan B. Scott. Thank you for your supervision and encouragement and patience. I would like to express my sincere thanks to Dr. Peter Blockley for his support from the very beginning of the research. I would like to thank the MomuTech company for providing the sample videos. Finally, I wish to express my deepest appreciation to my parents for all their encouragement and support. I would never have been able to accomplish my Ph.D. without your unconditional support.

Publication

1. Hedayati, M., Cree, M. J., & Scott, J. (2017). Effect of Contextual Information on Object Tracking. International Conference on Image and Vision Computing New Zealand (IVCNZ). Conference held in Christchurch, New Zealand.

2. Hedayati, M., Cree, M. J., & Scott, J. (2016). Scene structure analysis for sprint sports. International Conference on Image and Vision Computing New Zealand (IVCNZ). Conference held in Palmerston North, New Zealand.

3. Hedayati., Cree, M. J., & Scott, J. (2015). Combination of mean shift of colour signature and optical flow for tracking during foreground and background occlusion. In Image and Video Technology. Conference held in Auckland, New Zealand: Springer.

4. Hedayati, M., Cree, M. J., & Scott, J. (2014). Network structure for tracking of jockeys in horse races. In IVCNZ '14 Proceedings of the 29th International Conference on Image and Vision Computing New Zealand (pp. 13-18). Conference held in Hamilton, New Zealand.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

The advances of image sensors have enabled the deployment of video cameras on various platforms such as surveillance systems, unmanned aerial vehicles and mobile agents. The amount of visual information generated by an integrated image sensor is huge and demands advanced computational methods to extract the important information from the massive video databases. Automated video analysis is a set of computer vision and machine learning techniques to automatically extract information of interesting event from videos [Chung, 2010]. Traditionally, video surveillance was the researcher's focus since the beginning of video analysis study; however, with the improvement of processing power, the application of automated video analysis has been deployed to many other areas, including sports analysis [Barris and Button, 2008].

A comprehensive sports analysis system must solve three challenging computer vision tasks: scene analysis, object detection, and tracking. The scene analysis is responsible for extracting important segments from the videos. Object detection is necessary to automatically detect the object of interest in each frame, and object tracking is important to maintain the identities of detected objects over sequences of frames. It is worth noting that the application of scene analysis, object detection, and tracking algorithms are not limited to sports analysis sys-

tems. These methods have been intensely investigated over past decades, nevertheless, due to the various difficulties, they are not complete solution. These problems are mainly caused by noise in images, illumination changes, background clutter, low contrast between the object of interest and the background (camouflage), complex object motion, unpredicted camera motion and, most importantly, occlusions [Maggio and Cavallaro, 2011; Needham and Boyle, 2001; Chandel and Vatta, 2015].

The primary objective of sports video analysis is to interpret low-level visual features to generate high-level semantics that can be understood by the end user. The complexity of a sports video analysis system depends on the semantic level of interpretation which categorises into (1) event detection and (2) high-level analysis [D'Orazio and Leo, 2010; Thomas, 2011]. Event detection algorithms are designed to extract a particular event from the sport video, such as a penalty or goals. A high-level analysis system usually deals with extracting high-level semantics, such as contenders' movement, player skill and team strategy. The high-level semantic analysis should be able to solve two challenging tasks, object detection and object tracking. One issue facing sport analysis system is the interpretation of low-level visual futures to high-level human understanding [Li et al., 2004]. Although, at low-level visual extraction, all sport video analysis systems follow the same principle, interpreting this visual information to high-level human understanding is extremely specific to the sport in question.

To interpret low-level visual features, the use of *prior knowledge* is needed. Sport events have a well-defined content structure with predefined regulations which are known to the audience in advance [Kompatsiaris et al., 2012]. The prior knowledge can be grouped into two categories: production knowledge and domain knowledge [Li et al., 2004]. Production knowledge refers to the information gained from the video production, such as the camera angle and shot type (e.g. close view shot, far view shot). In contrast, the domain knowledge refers to the

structure of the game or the property of the key event in the particular sport (e.g. turning point in horse racing; penalty or corner in soccer).

What should be pointed out here is that most sports analysis systems are rule based. These rules are obtain based on prior knowledge and they are specific to the sport in question. Thus, it is necessary to investigate the structure and regulation of the sport in question before developing any analysis system.

1.2 Aim and Contribution

Most sport analysis research is focussed on team sports such as soccer [Breitenstein et al., 2009; Liu et al., 2009], hockey [Cai et al., 2006] and baseball [Lien et al., 2007]. These sports have a lot in common with surveillance systems, which is a well investigated problem. However, the challenge here is that this thesis addresses a sports structure that has not been studied before. In addition, the level of difficulty for automated analysis of horse races is much greater in comparison to studied sports. By consulting with professional horse race analysers, it was found that the approximate distance of the jockeys [1]from the fence is one of the key factors in evaluating contenders' performance. This information is more valuable at the turning points of the race, thus, the first task here is to obtain the turning segments from the race, and then to extract the trajectory of contenders from the turning segments.

To obtain the turning event, we propose a statistical motion analysis framework to break down the video into shots and then identifies the turning segments within each shot.

To extract the trajectory of contenders, we should address two challenging computer vision tasks; (1) object detection and (2) object tracking. These tasks are extremely difficult, especially in horse racing due to the variations in jockey

[1] Jockeys are the contenders in horse races.

appearance, dynamic motion of camera and jockeys, changes of camera viewpoints and the frequent occlusion of jockeys.

This project brings in high level contextual information to improve object detection and tracking under highly dynamic environment. This high level information is exploiting particular property of the horse races which helps to crack the most challenging sports analysis systems that have been developed so far. It is demonstrated in chapter 6 that the proposed system can be generalized to work on other domains, such as tracking group of people.

1.3 Thesis Structure

The outline of our proposed model contains three subsystems, namely (1) scene analysis (2) and contender detection (3) tracking and data association. In regard to this outline, the rest of the chapters are organised as follows:

Chapter 2 presents a general overview of the proposed horse race analysis system. This chapter discusses the challenges and characteristics of the horse race videos and outlines the main components of our proposed model.

Chapter 3 reviews key scene analysis techniques and then introduces a novel statistical motion analysis framework to retrieve the turning segments from the horse races.

Chapter 4 studies object detection algorithms for the sport analysis system and then discusses our framework for contender localisation.

Chapter 5 first reviews the tracking and data association techniques which prove their reliability in various tracking application, and then proposes the hierarchical multi tracking system to tackle the challenges of contenders tracking in the horse racing.

Chapter 6 examines the robustness of the proposed tracking model and then analyses the overall performance of developed horse race system.

Chapter 7 has closing remarks and provides recommendations for further work.

CHAPTER 2

System Overview

The majority of sport analysis research is focussed on team sports, especially on soccer [Breitenstein et al., 2009; Liu et al., 2009], hockey [Cai et al., 2006], baseball [Lien et al., 2007] and American football [Atmosukarto et al., 2013]. Regardless of the differences in the rules and regulations of these sports, all of them have relatively similar structures and consequently follow similar capturing techniques. For instance, the players in these games are moving on the pitch with uniform colour; this characteristic solves many issues regarding player detection. The opposite team have a contrasting shirt colour which greatly helps to handle the occlusion between opposition team members (although the overlapping of players in the same team is a challenging task). Furthermore, good video footage of wide and static views of the field that are suitable for both player detection and tracking can be extracted from any of these broadcast videos. Lastly, player detection can be considered a specific application of pedestrian detection, which is a well investigated application. The challenges for horse race analysis is that none of the above assumptions are valid; thus the level of difficulty in automated analysis of horse races should be considered to be much greater. The difficulties are as follows:

1- Motion complexity: In the race, the camera is generally moving constantly to keep contenders in the camera view. Therefore the motion complexity is not comparable to the static camera view.

Figure 2.1: Samples of two challenges that need to be dealt with in the horse race. The left image shows the occlusion between contenders and the right image illustrates the occlusion by other obstacles about the race track.

2- Frequent occlusion: In most horse races there are more than six contenders in the race and they closely follow other jockeys for a leading position. Therefore there is much more occlusion of contenders with each other than in team sports as can be seen in Figure 2.1.

3- Background clutter: The background is continually changing from one frame to another; thus there is not only background clutter but also contenders themselves may be obscured by obstacles such as trees, towers and bars (right image in Figure 2.1).

In addition to the above problems, some contenders may be wearing similar colours or have low contrast with the background.

2.1 Prior Knowledge

It is critical to know that sporting events have a well-defined content structure with predefined regulations which are known to the audience in advance [Kompatsiaris et al., 2012]. This prior knowledge is critical in designing a sports analysis system. In this section we outline the information that can be gained from production and domain knowledge of the horse race. This knowledge is repeat-

edly recalled in subsequent chapters; henceforth for simplicity, they are referred to by their index number.

K1: In sports which contenders race around a circular track, the camera typically follows the contenders. Therefore the motion of the background is much greater than that of the contenders in the camera view.

K2: The contenders in the race move in a row and behind one another. Therefore as the camera tries to keep them in the centre of the view, they usually spread vertically or horizontally in the middle of the frame where the motion and direction of their movement as well as the size of the contenders are relatively equivalent.

K3: Because it is a race along a track, the contenders relative to each other tend to move as a slowly changing group.

K4: The camera is positioned sufficiently far away from the race track, is fixed in position and is rotated to follow the contenders along the race track; thus the motion of the background in the camera view can be used as a reasonable approximation of the group flow with respect to the real world.

K5: As a consequence of properties K1 and K4, the trajectory of the group of contenders in the real world is the opposite of the background trajectory in the camera view.

K6: At some point in the race, the camera might stop moving (normally only at the end of the race). In this situation the actual group motion is used to estimate the trajectory.

K7: With regard to the most frequent view of the contenders with respect to the camera view, the frame structure can be categorised into side, three-quarter (3Q), and front views as illustrated in Figure 2.2. Furthermore, the

a) Sample side view frame structure

b) Sample 3Q view frame structure

c) Sample front view frame structure

Figure 2.2: The most frequent frame structures a)side b)3Q and c)front.

flow of the contenders (trajectory) over a short segment of the video can be classified into four motion models: side, 3Q, front, and turning, where the ideal trajectories of these motion patterns are shown in Figure 2.3.

2.2 Outline of Proposed System

The approximate distance of the jockeys from the fence at the turning point of the race track is one of the key factors in evaluating the performance of contenders in horse racing. To extract this information we need to extract the *frame sequences around the turning points* (turning segment), and then automatically detect and track the jockeys around turning segments as illustrated in Figure 2.4. The outline of our proposed model contains three subsystems: (1) shot analysis, shot is the continuous sequence of frames taken by a *single camera*, (2) contender de-

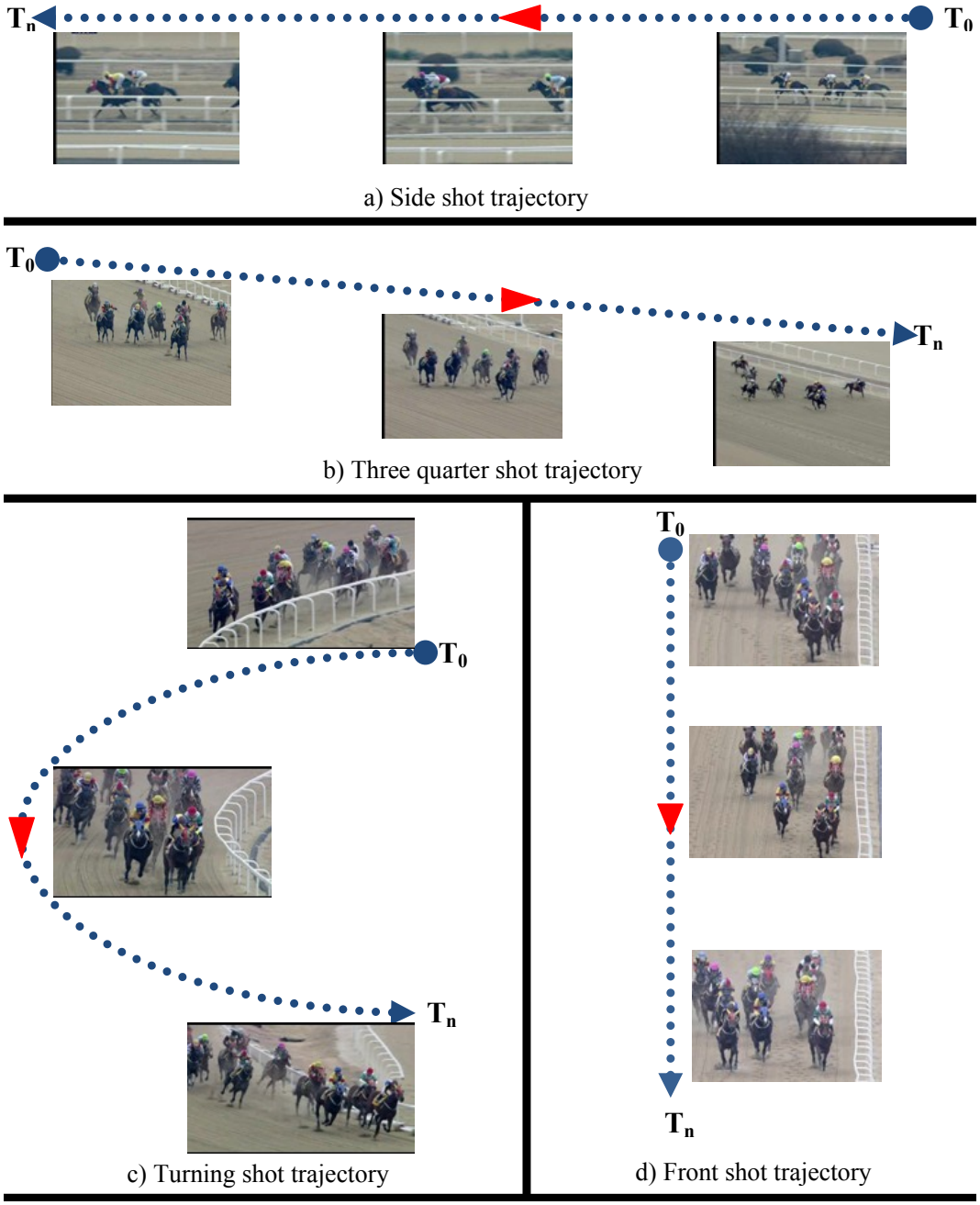

Figure 2.3: The ideal trajectories of contenders over short periods of time are illustrated by the dotted blue line, and the sample images show the frame structure at start, middle and end of the shot.

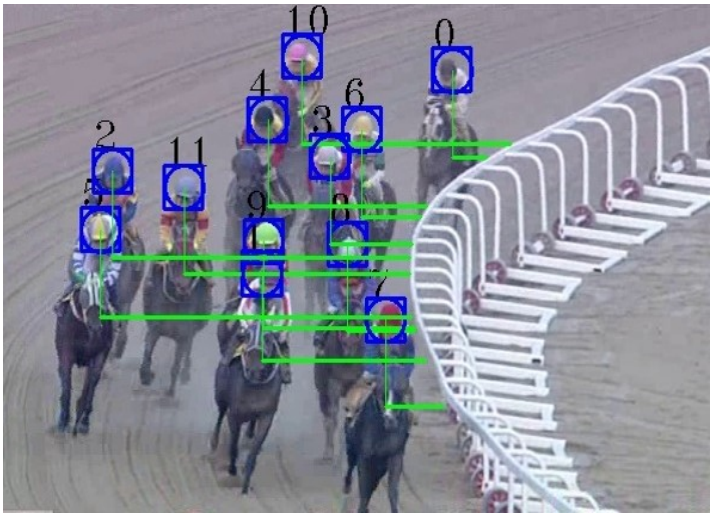

Figure 2.4: The main aim in this project is to extract the turning segment of the horse race video and then automatically detect and track the jockeys around the turning segment.

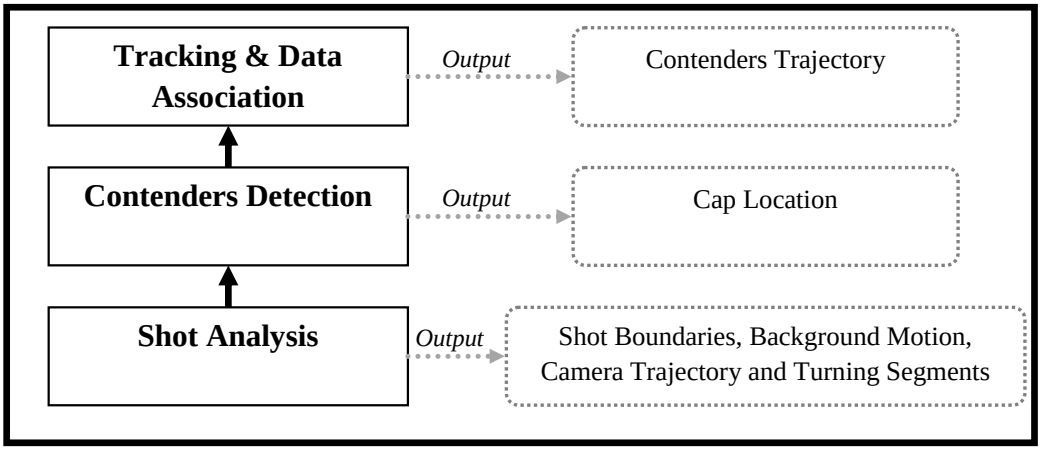

Figure 2.5: The proposed system is integrated from three subsystems and each one is responsible for retrieving different semantics.

tection and (3) tracking and data association, which is shown in Figure 2.5.

The main objective of shot analysis is to extract the turning segment of the race. For shot analysis we propose a statistical motion analysis framework to find the shot boundaries, to identify the shot trajectory and finally to extract the turning segments of the race. The detection of contenders is accomplished by determining the location of the jockey's cap in the frame using the Histogram of

Oriented Gradients (HOG) framework [Dalal and Triggs, 2005]. The cap shape is selected as the main feature of the detection algorithm due to three main reasons: firstly, they have a rigid and unique structure, secondly, the occlusion between the jockeys' caps is much less than the other parts of jockeys, and thirdly, the colours of the caps are usually different to the others, which reduces the uncertainty of tracking under partial and full occlusion. For the contender's trajectory retrieval, we propose a robust hierarchical tracking scheme which combines contender's local information at each of the point and object levels with the contextual information, such as the background properties and group movement characteristics of the contenders.

2.3 Evaluation Method

The proposed system contains three different algorithms namely scene analysis, cap detection and contender tracking. To analyse the accuracy of these algorithms, first the ground truth data (gold standard) must be created. The ground truth data is manually built according to the task of each algorithm. With the ground truth data and the output of the algorithm the following statistical measures are calculated:

- **True positive (TP) or hit:** An actual positive identified as positive.
- **False positive (FP) or false alarm:** An actual negative misidentified as positive.
- **False negative (FN) or missed:** An actual positive misidentified as negative.
- **True negative (TN) or correct rejection:** An actual negative identified as negative.

From TP, TN, FP, and FN, the recall and precision are estimated. The recall, also called sensitivity, is the fraction of positive samples that are correctly de-

tected as positive, namely

$$\textbf{Recall} = \frac{TP}{TP+FN} \,. \tag{2.1}$$

and precision, or the positive predictive value, is the fraction of positives that are relevant given by

$$\textbf{Precision} = \frac{TP}{TP+FP} \,. \tag{2.2}$$

Note that the above evaluation methodology is only used in shot analysis and cap detection. For analysing the performance of tracking of contenders a different approach is discussed in the final evaluation in Chapter 6. In addition, for shot analysis and cap detection our aim is to get maximum recall with best possible precision. This is because getting a false negative is very costly, specifically for shot analysis. This is because if we missed only one turning segment in the race videos, we can not evaluate the performance of all contenders in that segment. Considering that there is a maximum of four turning points in a race, missing one segment can have a huge effect on the final performance evaluation.

CHAPTER 3

Scene Analysis

In a sports broadcast, the cameras and their viewpoints are frequently changing. For example, in a short period, the camera might show a broad view of the field, followed by a close-up of a player, and then move to the crowd reaction [Kapela et al., 2015]. While a sports event can be a few hours in duration, it may nevertheless contain only a few interesting moments for a viewer or the professional sports analyst. The main aim of scene analysis is to automatically extract important moments from the video. Scene analysis systems usually start by breaking down the video into shots and then analysing the shot to retrieve essential video dynamic content [Bruno and Pellerin, 2002]. In the computer vision literature, the first task is often called *shot detection* [Hanjalic, 2002; Hu et al., 2007; Deokar and Kabra, 2014; Priya and Domnic, 2014], and the subsequent process is known as *event detection* [Zhong and Chang, 2004; Li et al., 2004; Xu et al., 2009; Kapela et al., 2015]. This chapter first reviews the significant approaches in shot detection, and event detection. Next the novel statistical motion analysis framework is introduced to retrieve critical events of the horse-race broadcast videos includes shot boundaries, background motion, and turning segment.

3.1 Literature Review

Although, there is no literature has been found related to horse-race analysis, this section reviews general principle of extracting information from the sport videos which is very useful for building horse-race analysis framework.

3.1.1 Shot Detection

Shots are the continuous sequence of frames taken by a *single camera*. The basis of any shot-detection method is the fact that frames surrounding a shot transition (shot boundary) display a significant change in their visual content [Hanjalic, 2002; Zabih et al., 1995; Asghar et al., 2014]. The shot boundaries usually are detected by finding the abrupt changes in the sequences of frames. These sudden changes are recognized by estimating the similarity of the frame sequences and if this similarity falls below a certain value a shot transition is taken to have occurred. The frame similarity can be estimated using different approaches, such as *pixel comparison, histogram difference, edge ratio*, and *motion based models*.

The pixel-based comparison methods [Zhang et al., 1993; Shahraray, 1995; Boreczky and Rowe, 1996] count the total number of pixels that change considerably from one frame to another, and the total number is compared with a predefined threshold to identify the shot change [Boreczky and Rowe, 1996]. The pixel by pixel comparison is too sensitive to light variation and camera or object movement. Moreover, it is not accurate at identifying gradual shot transitions.

In the histogram-based approach the grey or colour histograms of two consecutive frames are compared, and if they differ by more than a certain amount, then a change of shot has been identified. There are various histogram comparison methods reported in the literature such as bin to bin histogram differences [Mehtre et al., 1995], histogram intersection [Swain and Ballard, 1991], chi-square comparison [Nagasaka and Tanaka, 1991] and Bhattacharyya distance

[Bhattacharyya, 1946]. Comprehensive evaluation of these methods are reported by Dailianas et al. [1996] and Gargi et al. [2000]. The histogram-based methods are more reliable than pixel comparison and give better results under gradual shot transitions. However, they can miss the shot boundaries when two successive shots have similar colour distributions.

The edge based method [Zabih et al., 1995; Yoo et al., 2006] extracts the edges for two consecutive frames and the similarity between the extracted edges determine the shot boundaries. The Edge Change Ratio (ECR) proposed by Zabih et al. [1995] is a well-known edge-based algorithm which uses Canny edge detection to abstract the edges of two consecutive frames, then detect frame dissimilarity by looking at the maximum fraction of distinct edges in existing and incoming frames. Lienhart [1998] compared the performances of pixel, edge, and histogram approaches. He showed that all of these methods are very reliable for hard cut transitions, but, the detection of gradual transitions is not very precise.

Shot detection based on motion information [Bruno and Pellerin, 2002; Kamath et al., 2014] is based on the assumption that motion exhibit discontinuities across shot boundaries, thus the frame to frame motion can be used to detect the shot. In general motion information is very important in many computer vision application such as shot analysis [Bruno and Pellerin, 2002; Kamath et al., 2014], event detection [Lien et al., 2007] and object tracking. The motion information refers to the displacement of intensity patterns which are usually acquired by optical-flow techniques [Fortun et al., 2015]. The fundamental optical-flow equation is derived based on the assumption that the pixel intensity does not change with a small displacement [Horn and Schunck, 1981], and is given by,

$$f(x+dx; y+dy; t+dt) \approx f(x; y; t), \tag{3.1}$$

where $f(x; y; t)$ is the intensity of the image at position (x, y) at time t, dx and dy are x and y displacements and dt is the time difference. There are two ap-

proaches for optical-flow estimation: dense and sparse. In the dense model, the motion field is built based on the velocity of all pixels in the image. Popular dense optical-flow methods include those of Horn and Schunck [1981] and Szeliski and Coughlan [1997]. The dense optical-flow algorithms usually have difficulty in calculating flow in homogeneous regions or edges with orthogonal displacement. Sparse optical-flow solves these issues by only operating on the points that have strong gradients in both x and y direction. In the literature these points are called corners. The corner detection methods themselves are a broad topic but useful reviews are given by Tuytelaars et al. [2008] and Kerr et al. [2008]. The most commonly used sparse optical-flow, called Lucas-Kanade (LK), was proposed by Lucas et al. [1981].

Nguyen and Hwang [2002] used optical-flow to discover shot changes. They extracted the motion vectors between two consecutive frame then calculated the mean square prediction error (MSPE) and estimated how well the current frame could be predicted from the last frame using,

$$\text{MSPE} = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [O(i,j) - P(i,j)]^2 \tag{3.2}$$

where O and P are the input and predicted grey-scale frames with $N \times M$ dimensions.

Shot boundary detection using motion information is widely used in sport applications. This is because most sports analysis systems also use motion features to extract important event from a video. Motion information that has been extracted for shot-detection can be reused in the event-detection stage.

3.1.2 Event Detection

The primary task of event detection is to identify and extract the key events from a video. Event detection algorithms usually work by extracting key information

from the video and matching that information with an expected template of the sport event.

In earlier literature, multiple stage filtering was used to correlate the key information to specific events [Gong et al., 1995; Nepal et al., 2001; Ekin et al., 2003; Zhong and Chang, 2004]. Recently, researchers have applied machine learning methods such as neural network [Kapela et al., 2015; Montoliu et al., 2015], Hidden Markov Model (HMM) [Lien et al., 2007; Huang et al., 2009] and Support Vector Machine (SVM) [Ye et al., 2005; Sadlier and O'Connor, 2005] to classify the extracted information with respect to expected template of the sport events.

The three key features that are widely used to characterise sport events are, *colour, structural information*, and *motion features*.

The colour works because, the area surrounding the sports event usually has a distinct and uniform colour, therefore the colour distribution in the frame gives valuable information about the game (see Figure 3.1). Li et al. [2004] use the colour property of soccer games to distinguish close-up from far-field views. This model assumes the players are close to the camera when the frame is not dominated by the green soccer field. Thus, they classified the close-up from far views by counting the number of green pixels in the frame. In a similar approach Lien et al. [2007] used the proportion of soil or grass regions to the player skin colour to find the close-up shot in the baseball scenes.

The structural information is important because the playing field is usually enclosed by distinct boundaries and shapes. These outlines are unique and have specific meaning in that sport's context. The position and arrangement of the players gives important information about the game tactics (see Figure 3.2). Atmosukarto et al. [2013] identified the offensive team in American football using a presumption that the attacking team tends to be compactly located at the line of scrimmage [1] and there are at least five players near the ball in close proximity.

[1] A line of scrimmage is a line which can not be passed until the next play has begun.

Figure 3.1: Li et al. [2004] used the colour property of soccer games to classify different camera views. [Source: Li et al. [2004], 2004, "Bridging the semantic gap in sports video retrieval and summarization", Journal of Visual Communication and Image Representation, 15(3), 393–424. Underneath permission.]

Atmosukarto et al. [2013] used background subtraction to extract the players in the foreground and then line of scrimmage was identified in the area of the frame where the image gradient density of the foreground is at the highest. Finally they extract features from the offence regions to train a linear SVM and classify the different offensive formations.

Movement in the field betrays other important visual features that can be used to recognize action or tactical information. For instance, Lien et al. [2007] identified "running" in baseball by finding the global motion in the frames. Montoliu et al. [2015] used optical-flow to perform team activity recognition in Association Football. They manually divide the football pitch into ten cells and then extract the velocity vector for each cell over four consecutive frames. These velocity vectors are used to build a feature vector to analyse where and how players move on the pitch.

Sports videos may also contain auxiliary information such as overlay text or logos. These data can be used to detect a specific event or extract the game summary. For example, Pan et al. [2002] and Huang et al. [2009] used logo transition to spot replay or slow motion in the video. There are various papers [Zhang and Chang, 2002; Li et al., 2004; Kim et al., 2005] that used optical character recog-

Figure 3.2: Structure of serving in tennis games [Zhong and Chang, 2004]. Playing field is usually enclosed by distinct boundaries and shapes. In addition, the position and arrangement of the players gives important information about the game. [Source: Zhong and Chang [2004], "Real-time view recognition and event detection for sports video", Journal of Visual Communication and Image Representation, 15(3), 330–347. Underneath permission.]

nition (OCR) to extract the scoreboard or overlay data. Some approaches used audio cues [Nepal et al., 2001; Huang et al., 2009] to extract exciting events within sport. Nepal et al. [2001] used the crowd reaction to identify the occurrence of a goal in a basketball game. Huang et al. [2009] obtained the shot audio property to classify the video shot to speech, applause, ball hits, noise and music in tennis video using a hidden Markov model (HMM).

This literature review showed that the procedure of extracting information from the sport videos more or less follow a same principle, however interpreting that information to high level understanding is extremely specific to the sport in question. Therefore in designing scene analysis system the main issues that should be considered are (1) which visual features can best represent the characteristic of the event of interest and (2) how to interpret low-level visual information to give high-level semantic information.

3.2 Scene Analysis in Horse Races

The goal of this section is to extract the turning segments from within horse-race videos. To achieve this goal the video should be broken down into the shots. Then each shot is analysed separately to find turning points. Here a statistical motion analysis scheme is proposed to detect the *shots boundaries*, find the *background motion* and finally extract *the turning segments*.

3.2.1 Extracting Motion Information

As observed in section 2.1 properties K2 and K5, it can be concluded that accurate estimation of *background motion* is the backbone of the proposed event detection system in horse races. Note, however, background motion estimation is not an easy task, as there is other motion in the scene that is not background motion. At each instant, motion vectors can be due to any of three different sources,

1. The moving or panning camera which creates background motion.
2. Contenders which produce foreground motion.
3. Noise sources such as artefact and light variation, which generate outlier motion.

To get better estimates of background motion and to reduce the effect of foreground and outlier motion, we split the video frame into N independent and non-overlapping blocks (b), as shown in Figure 3.3. The frames are divided into three by four equal blocks thus $N = 12$. Next, the motion vectors for each block are obtained by using a pyramidal implementation of the Lucas-Kanade (LK) optical-flow technique [Bouguet, 2001]. The LK procedure starts by extracting corner points at frame n and then finds the corresponding location of these corners in the next frame, $n+1$. To detect corners, we used the model proposed by Rosten and Drummond [2006].

With the corner points q at frame n and their corresponding locations q' in frame $n+1$, the motion vector, $\mathbf{V}$, vector magnitudes, $|V|$, and the direction of movement, θ, are calculated for all corners per block. Furthermore, the error associated with each motion vector is approximated. The motion vector error, e, is estimated by obtaining intensity differences between the 21×21 pixel patch around the q and the q' and then dividing by the total pixels in the patch leading to,

$$e = \frac{1}{W} \sum_{i=0}^{W} (w_{q,i} - w_{q',i}). \tag{3.3}$$

where $w_{q,i}$ is the pixel intensity for the patch w, centred at corner q, and W is the patch size.

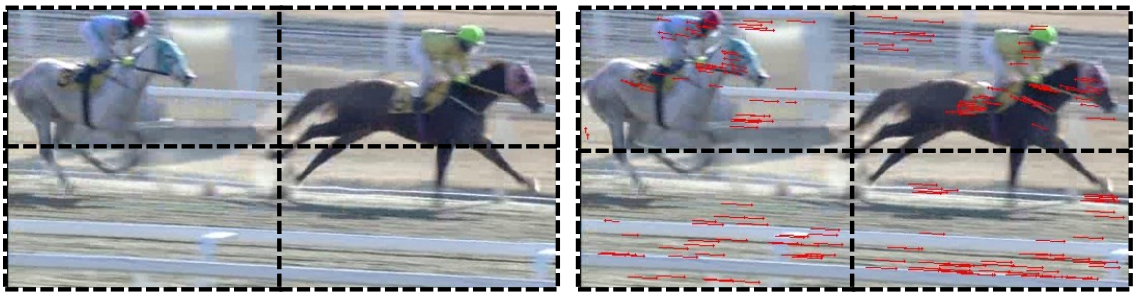

Figure 3.3: Motion feature extraction procedure starts by dividing the frames into blocks (left image) and then extracting the motion vectors in each block (red arrow on the right picture).

3.2.2 Detection of Shot Boundaries

The motion vector around shot boundaries exhibit discontinuity [Bruno and Pellerin, 2002; Nguyen and Hwang, 2002; Kamath et al., 2014]. This discontinuity can be estimated by finding the error associated with motion vectors per block using,

$$b_e = \frac{1}{n} \sum_{j=0}^{n} e_j, \tag{3.4}$$

where b_e is the block error, e is the vector error (see Equation 3.3) and n is the number of the motion vectors calculated in the block. If this error exceeds a certain threshold, continuity threshold, α, (see Section 3.3.1), it is assumed there is no flow continuity in the block and the block is marked as invalid. Finally the shot boundaries are identified when the total number of invalid blocks is larger than the number of valid blocks in a frame.

3.2.3 Background Motion Estimation

Background motion is estimated based on the distribution of velocity angles. Working with the angle distribution is more convenient than the magnitude or joint magnitude/angle distributions, mainly because the magnitude can have unbounded positive value, while the angles are limited in range. In addition, the detected corners that are closer to the camera move faster than the ones that are farther away. Therefore, while the background motion vectors tend to have similar angles, the magnitudes fluctuate widely.

To improve the distribution, the angle of vectors with small magnitude are excluded because they do not provide a reliable angle. Therefore the vectors with magnitude less than one pixel, referred to as static vectors, V_s, are filtered out. It should be noted, referring section 2.1, K6, that at some points of the race the camera may stop moving and as a result all the background vectors will be removed according to the above rule. This can lead to a large error in background motion estimation due to interpreting the contenders' motion as background motion. To avoid this miscalculation, the following condition is imposed on the system,

$$\textbf{Static Camera} = \frac{L_{V_s}}{L_V} > 0.5, \tag{3.5}$$

where "Static Camera" is the logical condition if the number of static vectors, L_{V_s}, are more than half of the total number of vectors, L_V.

The distribution of angles, from the remaining motion vectors, is estimated using a 72-bin histogram. The histogram bins are equally spaced at 5° across the interval from −180° to 180°. Furthermore, any bin value that falls below 50% of the histogram peak is set to zero. Let $H_{i,j}$ be the jth bin of histogram H in block i, then

$$H'_{i,j} = \begin{cases} \frac{H_{i,j}}{H_{i,max}} & \frac{H_{i,j}}{H_{i,max}} \geq .5 \\ 0 & \frac{H_{i,j}}{H_{i,max}} < .5 \end{cases} . \tag{3.6}$$

The angle distribution of the blocks with a greater number of background pixels have less angle variation. It can be seen in Figure 3.4 the angle histogram of the last two block, background blocks (bottem two subframes), only have one none-zero bin in their histogram. Thus, the histograms of each block are weighted by the contribution score (C), where the contribution score is the reciprocal of the total number of zero bins in the histogram (H'). This weighting procedure is reduced the contribution of the blocks that are affected by foreground and outlier motions. The pictorial example of the weighted histogram and their contribution to the final background motion estimation is shows in Figure 3.4.

Finally, by adding all weighted histograms, the optimal direction of the background motion vector, θ_{opt}, is estimated using,

$$\theta_{opt} = \text{Mode}\Big(\sum_{j=0}^{N} C_j H'_j\Big), \tag{3.7}$$

where N is the total number of blocks and C_j is the contribution score of the j^{th} block. The optimal magnitude of background velocity vector, $\mathbf{V}_{opt}$, is then estimated by taking the average of all motion vectors that have same angle of θ_{opt}.

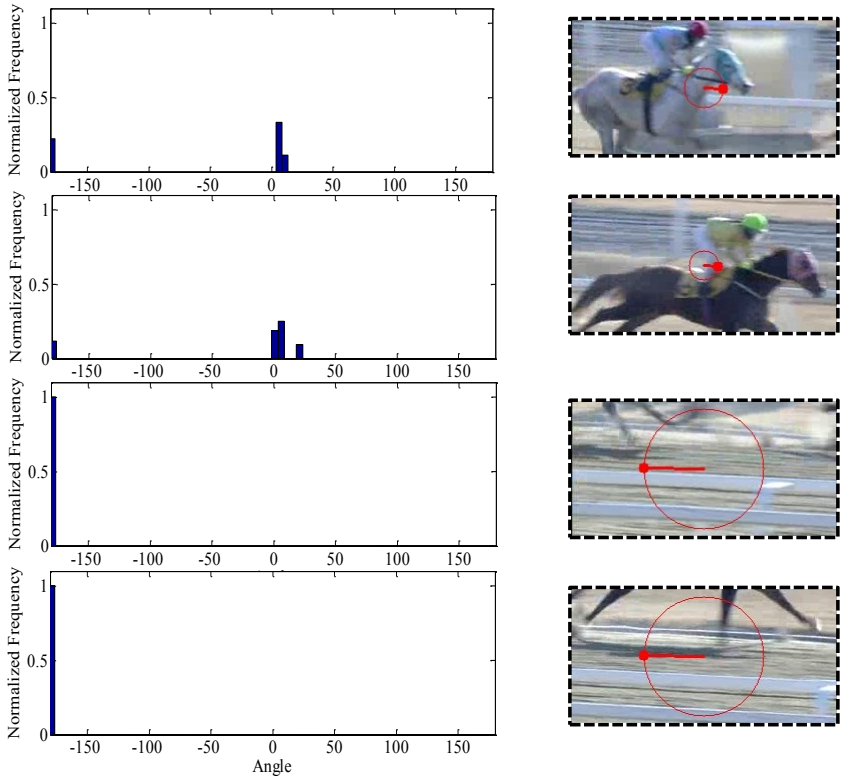

Figure 3.4: The main purpose of the weighting procedure is to reduce the contribution of blocks that are affected by foreground and outlier motion. The size of the red circles indicates the contribution of each block to the background motion estimation.

3.2.4 Turning Segments Extraction

It is known from section 2.1, the flow of the contenders over a short segment of the video is classified into four motion patterns: side, 3Q, front, and turning (See Figure 2.3). To create trajectories with respect to these motion patterns, the V_{opt} is accumulated from start to the end of each shot. The visual demonstration of the estimated motion pattern for one shot is shown in Figure 3.5[2].

Furthermore each motion pattern is broken down into multiple curves and

[2]Figure 3.5 contains embedded video.

Figure 3.5: The visual demonstration of the estimated motion pattern.

the motion direction of each curve segment is approximated using,

$$\tau = \arctan\left(\frac{y_{(t+\delta)} - y_t}{x_{(t+\delta)} - x_t}\right), \tag{3.8}$$

where δ is the step size, t is trajectory index. The trajectory index also indicates the frame number in the video. Thus, for each motion pattern there is one feature vector, τ, which contains all motion directions of the curve segment namely, $\tau = \{\tau_0, \tau_1, \ldots, \tau_n\}$. The process of building feature vector (τ) is shown in Figure 3.6.

Having vector τ, the shot classification and turning segment extraction is achieved by the building binary classifier tree as shown in Figure 3.7 and defining three logical conditions of C1, C2, and C3 namely,

C1: If vector τ contains at least one element with value of 90°, then the trajectory is created by the front view camera or it contains a turning segment.

This page contains embedded video.

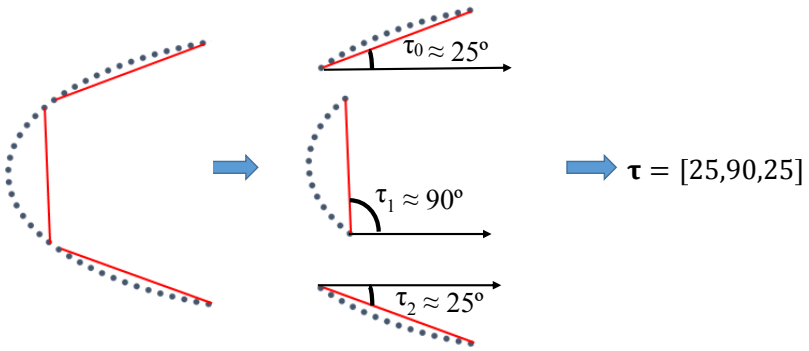

Figure 3.6: Feature extraction process for turning extraction. The camera trajectory is broken down into multiple curves and the tangent line to each curve is estimated and placed into vector τ.

Thus, the following condition is valid,

$$C1 = \begin{cases} 1 & \text{if} \quad (90-\epsilon) < |\tau_i| < (90+\epsilon) \\ 0 & \text{elsewhere,} \end{cases} \tag{3.9}$$

ϵ indicates the degree of freedom and it is set to 10°.

C2: If vector τ contains few elements with value of 90°, then the shot can not be a front view. This is because in the front motion pattern all directions of curve segments should be equal to 90°. Thus, the following condition valid:

$$C2 = \begin{cases} 0 & \text{if} \quad \kappa < \sum_{i=0}^{n} [(90-\epsilon) < |\tau_i| < (90+\epsilon)] \\ 1 & \text{elsewhere,} \end{cases} \tag{3.10}$$

where κ is a predefined threshold value.

C3: To distinguished the side view from 3–Q the following condition is applied:

$$C3 = \begin{cases} 1 & \text{if} \quad (180-\epsilon) < |\tau_{Mode}| \ \vee \ |\tau_{Mode}| < (0+\epsilon) \\ 0 & \text{elsewhere,} \end{cases} \tag{3.11}$$

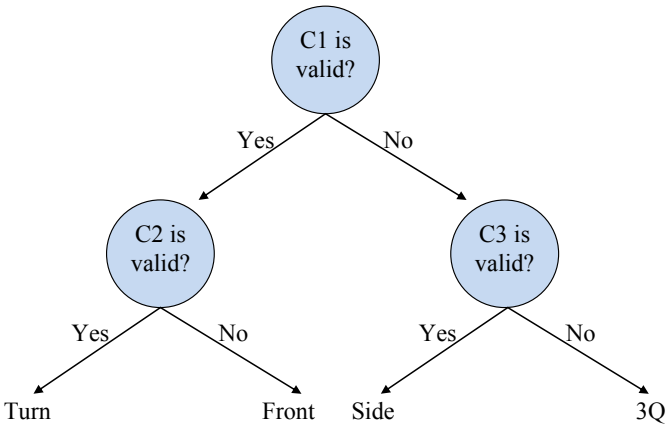

Figure 3.7: The structure of the binary tree classifier that is used for shot classification.

where τ_{Mode} is the mode of vector τ.

3.3 Preliminary Evaluation of Scene Analysis

3.3.1 Estimating Continuity Threshold

In order to find the proper value of the continuity threshold, α, for use with shot boundaries detection in horse races (Section 3.2.2). The precision and recall are examined for four different value of α, 6, 6.5, 7and 7.5, over twenty shots which manually identified from five videos. As demonstrated in Figure 3.8, the optimal range of α is between 6 and 6.5 and by setting α to 6.5 the shot detection algorithm achieves best performance with recall of 1 and precision of 0.98. In this evaluation the TP is the total number of correctly detected boundaries.

3.3.2 Turning Segment Extraction

The performance of the turning segment classification is affected by three parameters: step size, δ, angle range, ϵ, and κ. In this evaluation ϵ and κ is set to

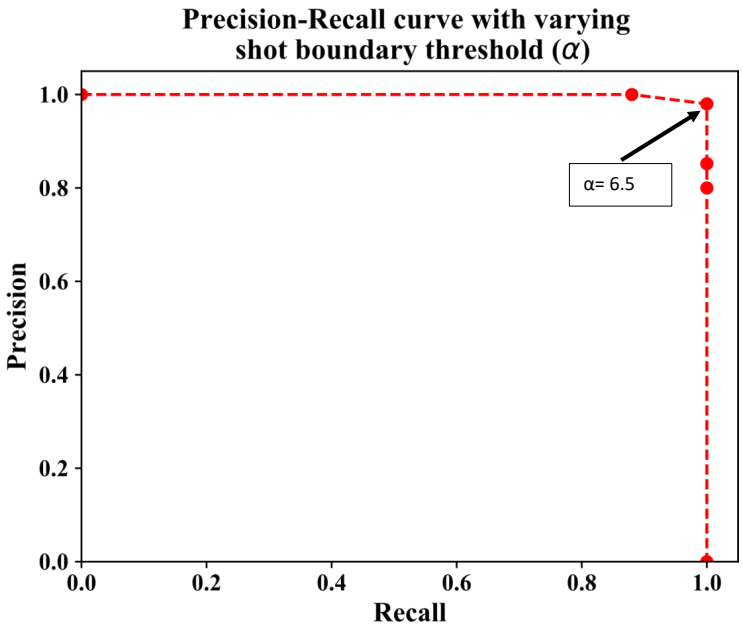

Figure 3.8: Precision–Recall versus threshold value (α) indicating the optimal range of α is between 6 and 6.5.

10 and 5 respectively. Next, the performance of the system is evaluated by using precision–recall graph with five different δ: 7, 11, 15, 19 and 23.

To build ground truth data turning segments of ten broadcast videos from three racing sites were manually extracted. The detail of statistical measures are tabulated in table 3.1. TP is defined by a detected turning point that falls within the manually selected turning segments. The sample results of our proposed turning point detection is illustrated in Figure 3.10.

It can be seen from of the precision–recall result in Figure 3.9, the classifier performs best when δ is between 11 to 15 giving a recall of 1 and precision of 0.87. It should be highlighted the algorithm in this range does not miss any turning segments which is the main priority here. This is because in some races there are only two turning points, thus missing one segment can have a huge effect on the final performance evaluation.

Table 3.1: The Statistical Measures for Ten Sample Videos with Five Different δ

		$\delta = 7$			$\delta = 11$			$\delta = 15$			$\delta = 19$			$\delta = 23$		
Id	**Total Turn**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**
H1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1
H2	3	3	1	0	3	0	0	3	0	0	3	0	0	3	0	0
H3	2	2	1	0	2	0	0	2	0	0	2	0	0	2	0	0
H4	2	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0
H5	3	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0
H6	3	3	2	0	3	2	0	3	2	0	3	2	0	3	2	0
H7	3	3	0	0	3	0	0	3	0	0	3	1	0	3	1	0
H8	3	3	1	0	3	0	0	3	0	0	3	0	0	3	0	0
H9	5	5	0	0	5	0	0	5	1	0	4	0	1	4	0	1
H10	3	3	2	0	3	1	0	3	0	0	3	1	0	3	0	0

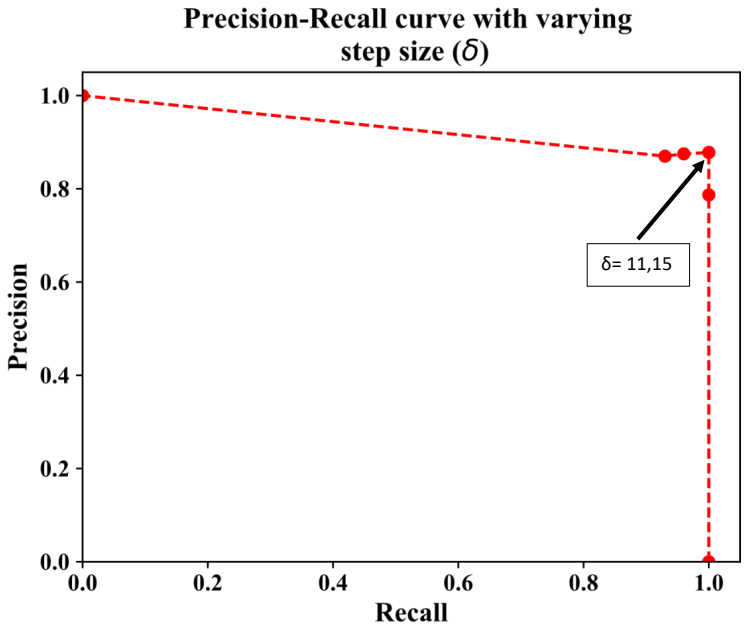

Figure 3.9: Precision–recall versus step size (δ) indicated the optimal range of step size is between 11 and 15.

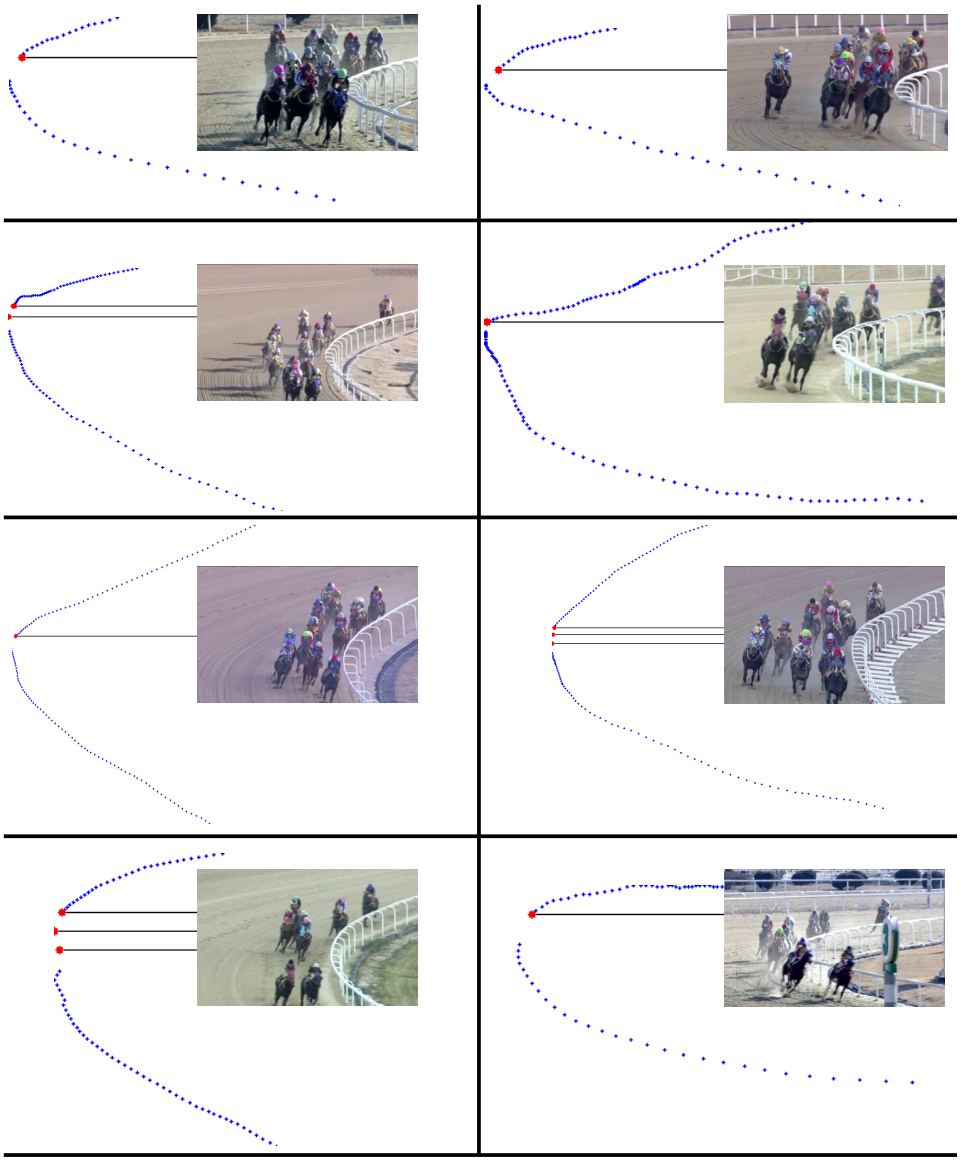

Figure 3.10: Sample results of turning point detection. The dotted blue line shows the camera trajectory and the red dot shows the beginning of curve segments that are perpendicular to their tangent line.

CHAPTER 4

Detection of Contenders

Object detection or object localisation is a fundamental problem in computer vision. The goal of object detection is to find the objects of interest in the image. Background subtraction and machine learning approaches are two common techniques used to extract object of interest from sport videos. Assuming the foreground pixels belong to object of interest, the aim of background subtraction is to identify a fragment of the frame that notably changed according to the background model [Piccardi, 2004]. The goal of machine learning approaches is to recognise the object of interest among predefined object classes (e.g. human, car, bicycle, ball). This task is achieved by learning the representation of objects from annotated sample images. It is worth noting that machine learning approaches are divided into classical model and deep learning approaches. Despite the recent amazing progress in object detection using deep learning framework, there is a paucity of literature on visual sport analysis using the deep learning concept. This is mainly because deep learning algorithms are very dependent on big image datasets [Li et al., 2015]. Considering that some sports can perform at any time under a different environment and lighting conditions (spotlight, day, night, sunny, cloudy, rainy) as well as pitch characteristic (clay, grass, carpet, etc.), collecting videos that contain enough sample of all of these cases is very hard, especially in pure research studies. Thus, learning information from different kinds of sports events is very challenging [Shih, 2017].

This chapter briefly reviews common object detection techniques that are employed in a sports analysis system, and then we will discuss our framework for jockeys localisation in horse racing.

4.1 Literature Review

Assuming the foreground pixels belong to object of interest, the aim of background subtraction is to identify a fragment of the frame that notably changed according to the background model [Piccardi, 2004]. The background model is often represented by the colour or intensity distribution of the pixels across the frame sequences [Piccardi, 2004]. Numerous background subtraction algorithms have been proposed over the past decades, including running Gaussian average [Wren et al., 1997], Gaussian mixture model (GMM) [Stauffer and Grimson, 1999], and temporal median filtering [Cucchiara et al., 2003]. According to review articles [Piccardi, 2004; Hedayati et al., 2010; Sobral and Vacavant, 2014; Higham et al., 2016] the GMM is one of the most reliable background subtraction methods. The GMM represents the colour or intensity distributions of pixels by a mixture of usually three or four Gaussian and any pixel that does not fall within a defined distance of one of the Gaussian is considered to be a foreground pixel. The GMM is used in various sports to extract the athlete from the background. Li and Lihong [2014] detected contenders in speed skating using the GMM from the three colour channels of R, G and B (see Figure 4.1). Gomez et al. [2014] extracted the player in beach volleyball by building the background model by applying the GMM on the grey scale images. In some particular environments when the object of interest are encompassed by the uniform colour of the background, the background modelling is achieved by finding dominant colours in the scene. Huang et al. [2007] detect soccer players by building the colour models for playfield pixels and non-playfield pixels. These approaches

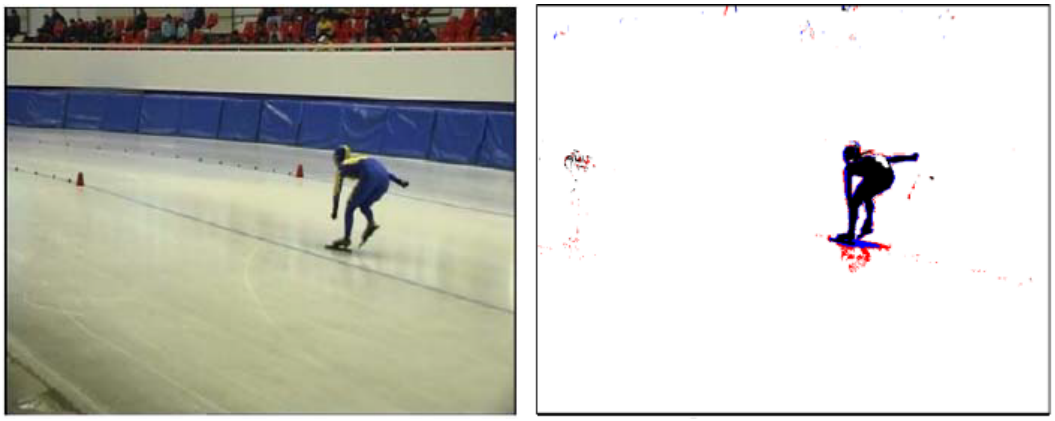

Figure 4.1: Li and Lihong [2014] detect contenders in speed skating using GMM and from three colour channels of R, G and B. [Source: Li and Lihong [2014], "Research of Background Segmentation Method in Sports Video", Indonesian Journal of Electrical Engineering and Computer Science, 12(6), 4274–4282. Underneath permission.]

usually used conventional segmentation algorithms, such as K-mean clustering, colour indexing, and pixel comparison to distinguish between foreground pixel and the background. Foreground detection using dominant colours is common for player detection in the sports that are played out on uniform playfield; for example, soccer [Huang et al., 2007; Hoernig et al., 2015], hockey [Higham et al., 2016], American football [Gu et al., 2004] and tennis [Jiang et al., 2009]. In general, background subtraction algorithms have decent detection accuracy in a simple environment, however, they can not be used in video with a moving background. In addition, the background model should be robust against illumination variation and small movements in the background (e.g. moving leaves, rain) [Sen-Ching and Kamath, 2004].

The machine learning approaches consider object detection as the classification problem in which the goal is to recognise objects from known classes [Amit and Felzenszwalb, 2014]. The general framework of machine learning-based object detection consists of of two stages namely, learning stage and detection [Li et al., 2015] as illustrated in Figure 4.2.

In the learning stage (left side of Figure 4.2), objects and non-object sample

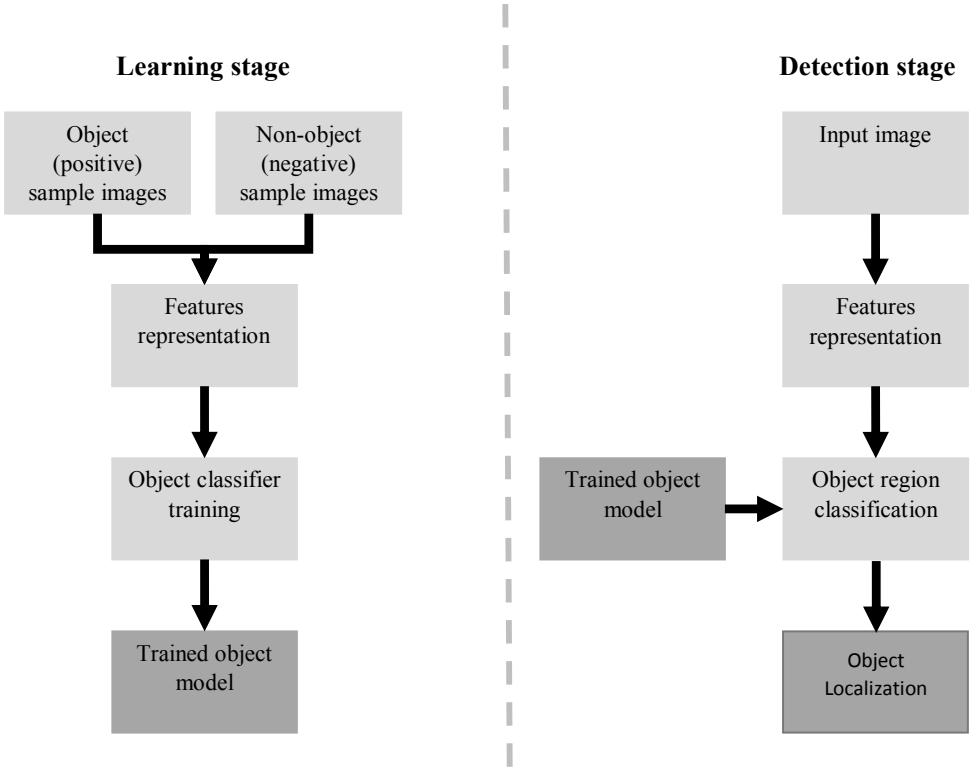

Figure 4.2: The object detection framework for machine learning approaches.

images are collected. Next, the feature extraction algorithm is applied to map image samples into feature space. Then, machine learning models are used to learn the object representation models to classify the objects [Li et al., 2015]. Let the object features be represented by vector $\mathbf{X}$, and the object classes by $c = \{1, \ldots, C\}$. Then a maximising posterior probability (MAP) concept can be used to classify the objects, so that the classification task is to estimate the posterior probability, $p(c|\mathbf{X})$ for all classes and then assign $\mathbf{X}$ to the class with maximum likelihood. The classification is achieved either by constructing a model for each class (generative) or by finding the decision boundary in the sample space that is between classes (discriminative) [Ulusoy and Bishop, 2006] (See Figure 4.3).

It has been shown by Jordan [2002] and Bouchard and Triggs [2004] with a sufficient number of positive and negative samples, discriminative detection accuracy is superior to generative models. Therefore if the aim is to find ob-

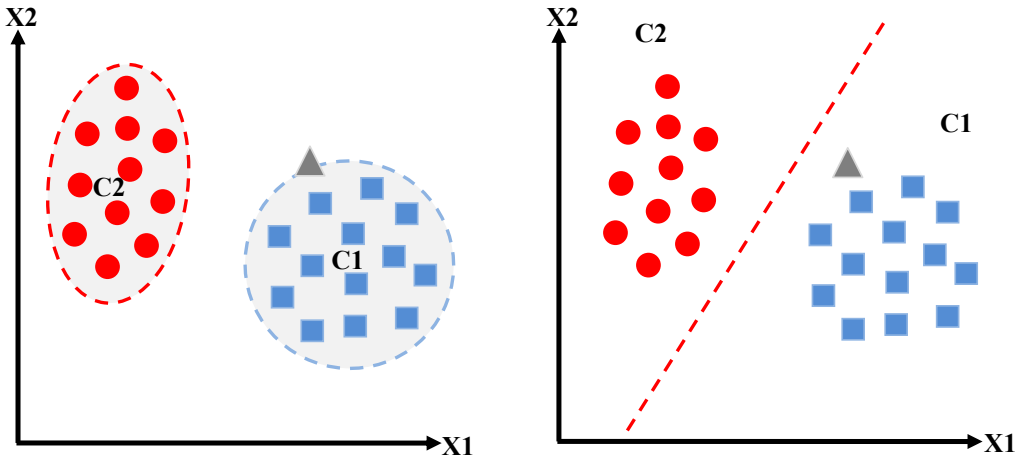

Figure 4.3: The generative approach, left image, assigns the new sample, grey triangle, to the closest class model, while the discriminative classification, right side, used direct mapping by finding the boundaries between the classes C1, and C2. The red line is classifying plane.

jects, such as a player or the ball, the discriminative approach is usually preferred. However, the generative approaches, such as Bayesian network and hidden Markov model (HMM), are broadly used in the sports event classification where the data labelling is a difficult and exhausting task.

In the detection stage (right side of Figure 4.2), input images are mapped into the same feature space and then the trained object model are used to find objects in the frame. The basic detection process consists of scanning the image using a predefined window size and then classifies each window as object or non-object. This is typically done at multiple resolutions of the image pyramid to detect objects at multiple scales [Amit and Felzenszwalb, 2014].

The list of studied papers for contender detection is tabulated in table 4.1. As can be seen from the table, 4.1 the adaptive boosting (AdaBoost) [Freund and Schapire, 1995] and support vector machines (SVM) [Boser et al., 1992] are widely used for contender detection in sport. AdaBoost is constructed by utilising many weak classifier algorithms. These weak classifiers usually are a cascade of the simple operators (e.g. binary classifier and thresholds) which eventually give a

powerful class discrimination result. AdaBoost is used extensively with Haar features to detect various objects, such as the face [Viola and Jones, 2001], hand [Ong and Bowden, 2004], ear [Islam et al., 2008] and pedestrians [Viola et al., 2005]. The SVM model tends to maximise the marginal distance between the classes to find an optimal hyperplane that separates the classes in the feature space. The margin is defined by estimating the distance between the closest points in each class to the hyperplane boundary. The combination of HOG features and SVM is very common in machine learning object detection approaches [Dalal and Triggs, 2005].

The majority of publication on player detection focus on team sports, especially on soccer [Breitenstein et al., 2009; Liu et al., 2009], hockey [Cai et al., 2006], baseball [Lien et al., 2007] and American football [Atmosukarto et al., 2013]. It is worth noting that, in all of these sports the players are encompassed by the uniform colour of the background pitch. Furthermore, good video footage of wide and static views of the field that are suitable for player detection can be extracted from any of these broadcast videos. Lastly, player detection can be considered as a specific application of pedestrian detection which is a well investigated topic. The challenges for horse race analysis is that none of the above assumptions are valid and the jockeys' appearance is quite different than the players in the studied papers.

Table 4.1: Significant literature On Object Detection in Sports

Year	Authors	Sport	Detection Approch	Feature	Classification Method
2001	Needham and Boyle	Indoor soccer	Dominant colour	Colour (HSI)	–
2005	Bertini et al.	Soccer	Machine learning	Haar-like	AdaBoost
2006	Zhu et al.	Soccer	Dominant colour & Machine learning	Colour (HSV)	SVM
2007	Lehuger et al.	Soccer	Machine learning	Haar-Like	AdaBoost
2007	Huang et al.	Soccer	Dominant colour	Colour (RGB)	–
2009	Jiang et al.	Tennis	Dominant colour	Colour (HSV)	–
2009	Lu et al.	Hockey	Machine learning	Haar-like	AdaBoost
2010	Maćkowiak et al.	Soccer	Dominant colour & Machine learning	Colour & HOG	SVM
2011	Salehifar et al.	Volleybal	Background subtraction	Intensity	–
2011	Tran et al.	Soccer	Dominant colour	Colour (HSI)	–
2013	Zhang et al.	Diving, Long jump, Speed skating	Background subtraction	Intensity	–
2014	Li and Lihong	Speed skating	Background subtraction	Colour (RGB)	–
2014	Gomez et al.	Beach volleyball	Background subtraction	Intensity	–
2014	Pettersen et al.	Soccer	Background subtraction	Intensity	–
2015	Reno et al.	Tennis	Background subtraction	Intensity	–
2015	Mahmood et al.	Baseball	Machine learning	Haar-like	AdaBoost
2016	Zhao and Lu	General	BGS (RGA)	Intensity	–

4.2 Localisation of Contenders

To locate the jockeys, in the turning segment of the race videos, the histogram of oriented gradients (HOG) framework proposed by Dalal and Triggs [2005] is used. Dalal and Triggs [2005] demonstrated that HOG feature representation of object, human, and a Linear SVM could give highly accurate object classifiers for human detection.

4.2.1 HOG Framework

The main idea behind HOG features is that local object appearance and shape can often be characterised rather well by the distribution of local intensity gradients or edge directions [Dalal and Triggs, 2005] as illustrated in Figure 4.4.

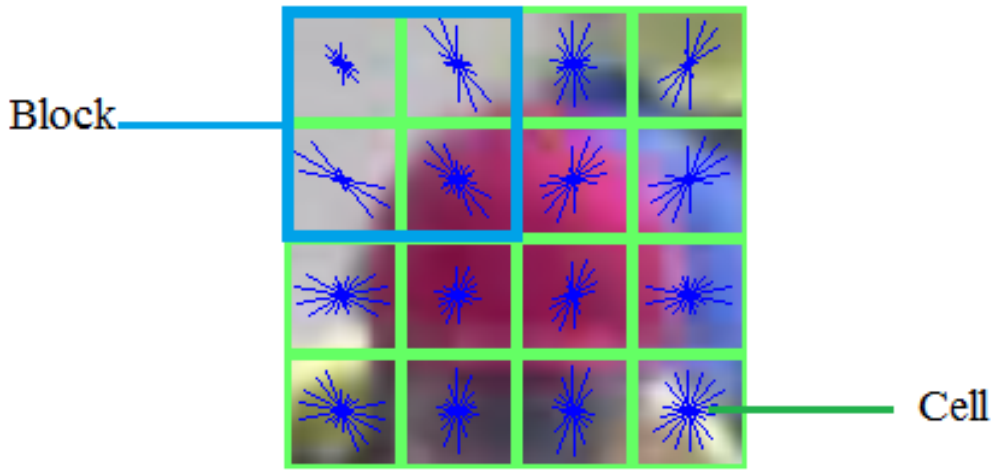

Figure 4.4: HOG characterise the object by the distribution of local intensity gradients or edge directions. The image illustrates the edge directions in each cell of a jockey's cap sample.

The first stage in HOG feature extraction is to divide the sample images into blocks, then each *block* is divided into smaller regions, called *cells*. The blocks

can overlap each other so that the same cell can contribute to several blocks. The amount of overlapping depends on *block stride*. Next, for each pixel within the cell, the vertical and horizontal gradients are obtained using Sobel operators. The gradient magnitude (g) and orientation (θ) are calculated using

$$g = \sqrt{G_x^2 + G_y^2}, \qquad \theta = \arctan\Big(\frac{Gy}{Gx}\Big). \tag{4.1}$$

where G_y is vertical and G_x is the horizontal gradient. Each point in the cell casts a vote into the gradient-orientation histogram bins with gradient magnitude as the weight. The orientation range of the histogram can be unsigned, from 0 to 180° or signed from 0 to 360°. To reduce the sensitivity the gradient energy inside the blocks are normalised. Two typical normalisation are L1-norm and L2-norm which define as

$$L1-\text{norm}: \frac{v}{\|v\|_1 + \epsilon} \rightarrow v, \qquad L2-\text{norm}: \frac{v}{\sqrt{\|v\|_2^2 + \epsilon}} \rightarrow v, \tag{4.2}$$

where $\|v\|_k$ is k-norm for k=1; 2 and v is the descriptor vector in block, and ϵ is a small positive number.

In the classification step normalised histograms are fed to linear SVM as shape descriptor to train the object model. Finally, in detection stage the scanning window is moved across the image to all positions and scales to locate the object of interest as shown in Figure 4.5.

4.2.2 Labelling

Clearly, the first stage of any classification model is to gather enough samples, so 8000 non-cap and 2000 cap templates were manually cropped from the turning segments of the race videos. Some of the positive and negative samples in the database are shown in Figure 4.6. Negative samples (non-cap) can be any tem-

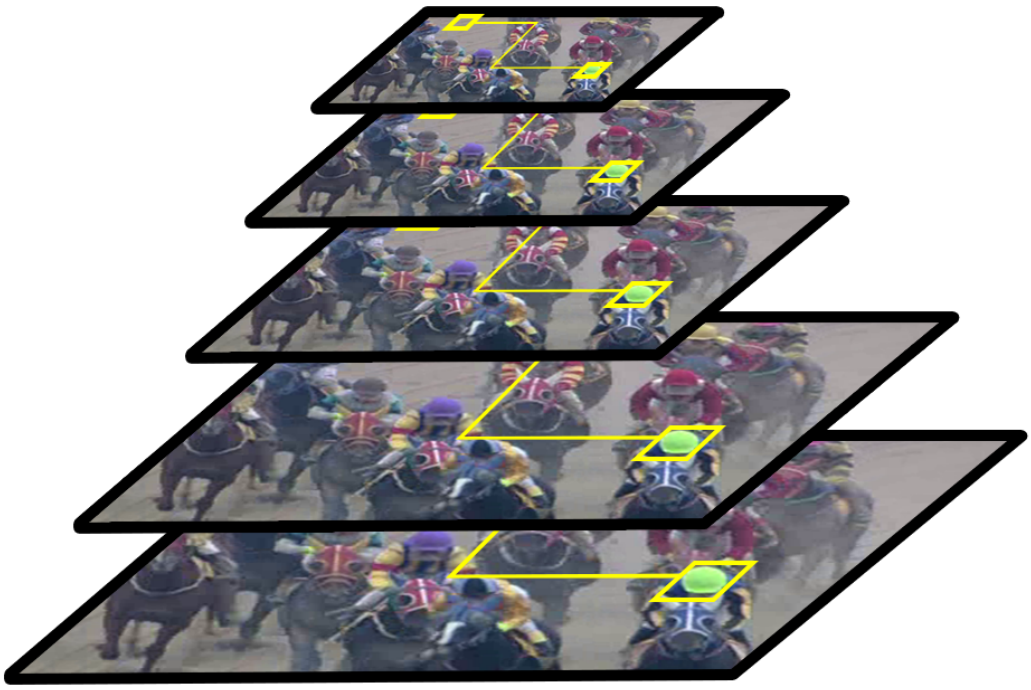

Figure 4.5: The detection process consists of scanning the image using a predefined window, yellow box, at all positions and different scales.

plate that does not contain a cap. The positive sample (cap) is represented by a square box, where the cap is located at the centre of the box with a margin of half a cap width around the caps as shown in Figure 4.7. It should be noted that this dataset is used to train the classifier (left side of Figure 4.2), to evaluate the accuracy of detection (right side of Figure 4.2) all cap locations inside 100 frames are manually marked for *testing*. The test dataset is used to optimize the accuracy of *cap detection* algorithm (see Section 4.2.3).

4.2.3 Finding Optimal Parameters

Parameters such as sample image size, block size, cell size and block stride were originally used to model human shape [Dalal and Triggs, 2005]. However, the intention here is to find the contenders by locating the shape of a jockey's cap,

Figure 4.6: The cap templates for the positive sample , left image, are represented by a square box. Negative samples (non-cap) can be any image that do not contain a cap.

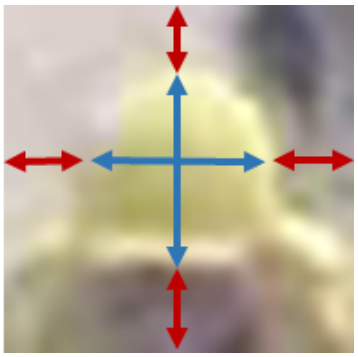

Figure 4.7: The cap is located at the centre of the box with margin of half a cap width around it.

thus new HOG parameters should be estimated.

The shape of the jockey's cap is selected as a feature due to three main reasons: firstly, they have rigid and unique structure. Secondly, occlusion between the jockeys caps is much less likely than the other parts of jockeys and thirdly, the colour of each cap is usually different to the other caps, which reduces the uncertainty of tracking under partial and full occlusion.

As mentioned in Section 2.1, production knowledge is critical in designing a sports analysis system. Production knowledge refers to production techniques

that are used for capturing videos, such as the camera angle and shot type (e.g. close view shot, far view shot). Specifically in our case, finding the most probable distance of the camera to the contenders at the turning points of the horse-race is important due to two reasons:

1. For training the SVM; all training samples should be of same size, therefore finding the proper sample size is important.

2. In the detection stage the predefined window is scanned across the image at multiple resolutions of the image pyramid to detect caps at multiple scales, so by knowing the approximate size of the cap in the frame the detection accuracy can be increased (see section 4.2.4).

It is impossible to find the actual distance of the camera to the contenders from broadcast videos. However, considering that the positive samples are built from 2000 cap images that were extracted from various races, the most frequent appearance of the cap sizes around turning segment can be estimated by finding distribution of cap sizes from the positive samples. This distribution shows the most probable cap sizes fall between 28×28 to 32×32 square pixels with the minimum cap size of 16×16 and maximum of 44×44 square pixels as illustrated in Figure 4.8.

In the learning stage and based on information in Figure 4.8, four different models are trained by SVM. The property of these model are tabulated in table 4.2.

To evaluate the accuracy of detection, the testing data set is built by manually marking all cap locations inside 100 sample frames. These test images are selected from 10 race videos with three different lighting conditions (cloudy, sunny and night). Next, two different scanning window with size of 32x32 square pixels (for Models 1 and 2) and 28x28 square pixels (for Models 3 and 4) are scanned across test images to locate the caps inside each image. The respond of detection

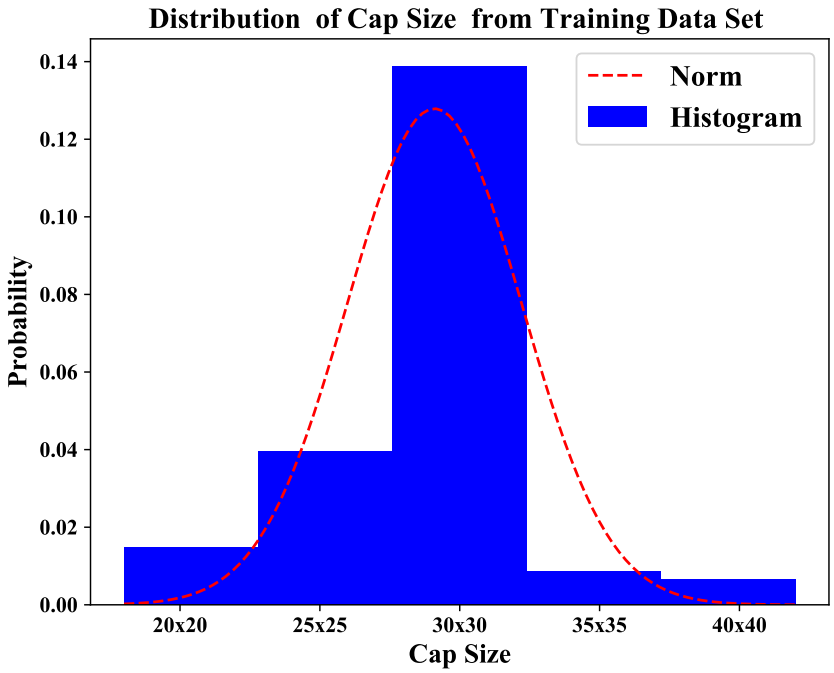

Figure 4.8: Distribution of the cap sizes around turning points in regard to the camera view. The most probable cap size falls between 28×28 to 32×32 square pixels.

is consider correct if the centre of detected bounding box lays inside the manually marked cap location (TP).

The precision-recall curve for each model with different hit threshold is illustrated in Figure 4.9. The hit threshold is the distance between features space and the SVM classifying plane. Increasing hit threshold moves the classifying plane towards the positive feature space. The hit threshold is varying from 0 to 2.

It can be seen from Figure 4.9 the cap detection is best with the Model 2 and 3. However, Model 3 is faster for detection in comparison to Model 2 due to smaller features vector (see last column of table 4.2). Thus the parameters listed in Model 3 are the optimal parameters for detecting of the cap of jockeys.

Table 4.2: The Properties of Four Trained models

Model	Parameter				
	Box Size	Block Size	Cell Size	Block Stride	Feature Vector Size
1	32x32	16x16	8x8	8	324
2	32x32	8x8	4x4	4	1764
3	28x28	8x8	4x4	4	1294
4	28x28	4x4	2x2	2	6084

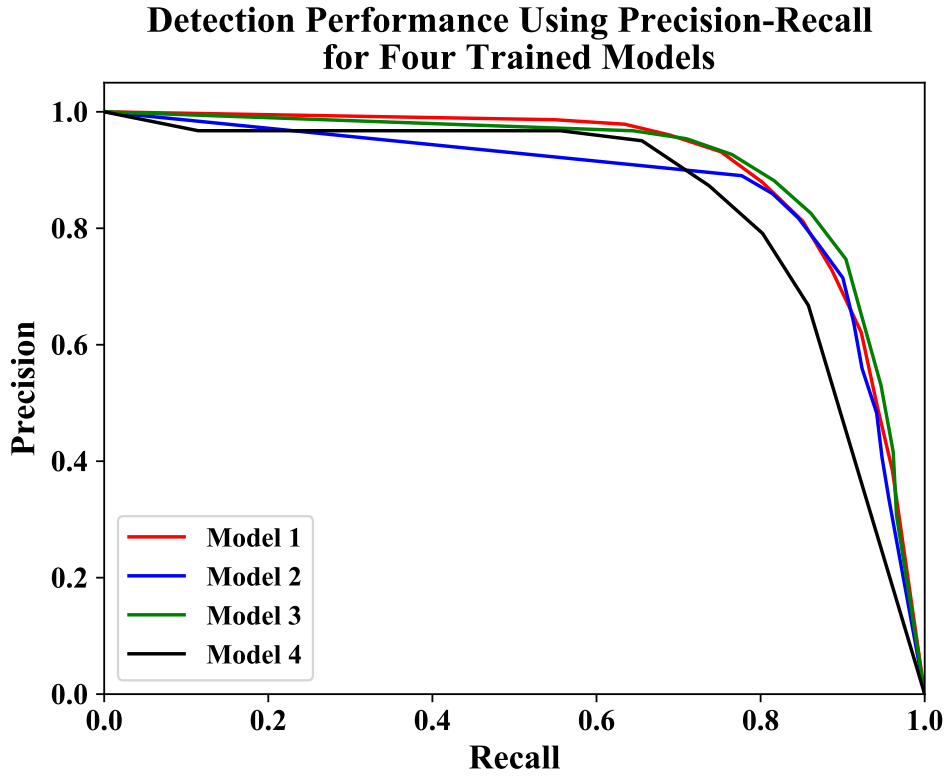

Figure 4.9: The detection performance of four trained models by varying hit threshold from 0 to 2. Models two and three (blue and red line) give the best performance.

4.2.4 Optimization

Objects are always embedded in a certain context. Contextual information plays an important role in any video analysis and image understanding applications. The cap detection algorithm discussed in Section 4.2.2 did not consider any contextual information. Therefore to improve the detection accuracy we merge two

contextual constraints as follows;

1 : It is shown in Figure 4.8 that the maximum cap size at turning points is 44×44 and the minimum is 16×16 square pixel. Any detection above or below these cap sizes are considered a false positive.

2 : property K2 of section 2.1: The contenders in the race move in a row and behind one another, so the motion and direction of their movement as well as *size* of the contenders are similar. Thus, in each turn, the cap size of each contenders should be similar with a little variation, so that any caps bigger or smaller than one-third of median width of detected bounding boxes is considered a false positive and filtered out.

The detection performance of Model 3 before and after contextual filtering is shown in Figure 4.10. As we expected the precision is improved after applying contextual filters by decreasing the false positive. The images in figure 4.10 visually demonstrate the effect of filtering before and after applying contextual constraints.

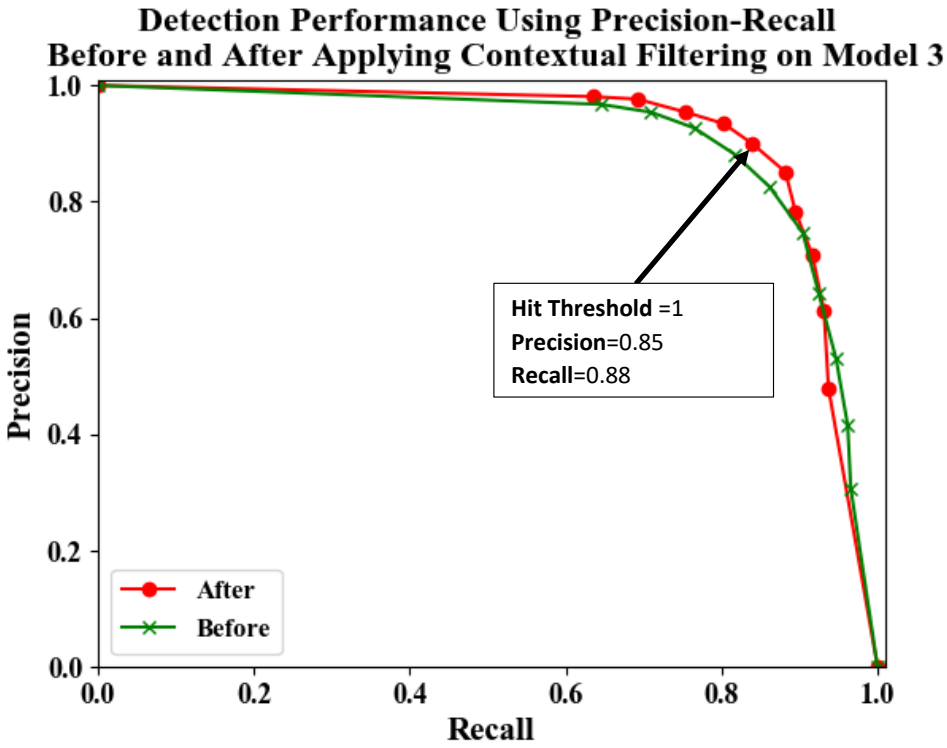

Figure 4.10: Detection performance using precision-recall before and after applying contextual filtering on Model 3. As It is expected the precision is improved after applying contextual filters.

Figure 4.11: The left image shows the detection result before contextual filtering and the right image illustrated the final result after applying contextual filtering. It is shown the false positive (red boxes) are removed by contextual filtering.

CHAPTER 5

Tracking and Data Association

Over the last few decades an enormous amount of study has been dedicated to object tracking [Yang et al., 2011]. Object tracking remains a challenging topic in computer vision due to problems caused by changes in size or pose of the object, noise produced by the image acquisition, variation of light, occlusion and background clutter [Yilmaz et al., 2006; Maggio and Cavallaro, 2011]. Moreover, the complexity of the tracking is increased if multiple moving objects are tracked. This is because locating targets and maintaining their identities through a video sequence is a highly challenging problem in crowded environments. The problems of multi-target tracking are normally solved by data association techniques [Yang et al., 2011; Betke and Wu, 2016]. Data association deals with the problems of selecting measurements that most probably originate from the object to be tracked [Bar-Shalom and Tse, 1975; Bar-Shalom et al., 2009]. The main challenge of data association approaches is the detection of unknown time-varying targets from a set of noisy and uncertain measurements [Maggio and Cavallaro, 2011].

This chapter first reviews the tracking algorithms and data association techniques which proves their reliability in various tracking applications and then we propose our hierarchical multi tracking system to tackle contenders tracking in horse races. To overcome the difficulties that exist in the horse race environment we propose a multiple object tracking system that gathers information from multiple cues such as the background characteristic, local contender infor-

mation (appearance and motion features) and the behaviour of contenders in the group.

5.1 Literature Review

5.1.1 Tracking

Primitive tracking systems use background subtraction approaches to separate the foreground from the background, and tracking was then performed by enforcing spatial continuity using Kalman filtering [Seo et al., 1997; Han et al., 2005]. Colour-based trackers, such as mean-shift [Comaniciu and Meer, 1999] and particle filtering [Blake and Isard, 1997], have achieved considerable success in many tracking applications. Particle tracking is a process of propagating the posteriori distribution of the reference target, according to a system dynamic model. Pérez et al. [2002] and Nummiaro et al. [2003] proposed two independent solutions that couple the colour information of objects with the dynamic model of the system. Particle tracking is successfully implemented in various sports, such as indoor soccer [Needham and Boyle, 2001; Morais et al., 2012], soccer [Zhu et al., 2006; Tran et al., 2011], beach volleyball [Gomez et al., 2014] and hockey [Okuma et al., 2004; Lu et al., 2009].

Mean-shift is a non-parametric technique for finding the mode of a probability density function by using gradient descent or ascent [Comaniciu and Meer, 1999] to find the local minima or maxima of a distribution by iteratively descending or climbing the density gradients until the point of convergence has been found. In statistics, mean-shift is considered a robust statistical technique. These models are labelled robust because they are less affected by outliers and small deviations from the model assumptions [Hampel et al., 2011]. In computer vision applications, the mean-shift was originally employed by Comaniciu and Meer

[1997] for segmentation purposes, and later Bradski [1998] utilised the mean-shift framework for tracking applications. The mean-shift tracker model calculates the centroid of the colour probability distribution within its 2D tracking window, then moves the window centre to the centroid of distribution.

Although the mean-shift tracker gives reasonable accuracy in a wide range of environments, it is prone to failure, when 1) the object and background have similar features causing gradient descent to get stuck in local minima, and when 2) the object is completely or partially occluded, the object likelihood is reduced leading to convergence to the wrong point. Various adaptations have been made on mean-shift to solve the above difficulties. Comaniciu et al. [2000] used the weighted probability distribution in order to assign higher weighting to pixels nearer to the centre of the window, based on the assumption that the foreground pixel is more likely selected near to the centre of the tracking window rather than its border. Allen et al. [2004] introduced a background–weighted histogram by assigning lower weight to colour features that belong to the background. The background weighted–histogram weights the probability distribution by considering the distribution ratio between background colour (pixels outside the tracking window) and the foreground colour (pixels inside the tracking window). Dixit and Venkatesh [2009] and She et al. [2004] handled partial occlusion by combining edge and colour features. Mazinan and Latifi [2012] solved the full occlusion problem when the object moves with constant speed by combining mean-shift with Kalman filter.

Optical flow-based trackers estimate the motion flow of moving objects to estimate the object displacement into the next frame. The median flow (MF) proposed by Kalal et al. [2010] is a good example of an optical flow-based tracker. MF trackers start by extracting sample points in the rectangular grid inside the bounding box of the object of interest. Next, these points are tracked by the LK tracker, and object movement is approximated by finding the median dis-

placement of these sample points. The optical flow based tracking algorithms are highly sensitive to occlusion. Therefore optical flow is usually combined with other tracking algorithms such as mean-shift and Tracking–By–Detection to handle occlusion. Hou and Li [2011] modelled the object and background motion using the LK sparse optical flow to track camouflaged objects. Oshima et al. [2006] proposed a single object tracking system for a static near-infrared camera. They embedded object flow information into the mean-shift tracker, where localisation is achieved by applying gradient ascent on the combination of magnitude, flow and colour features space.

Recently the tracking-by-detection algorithm has become popular for object tracking [Wu et al., 2015]. Tracking-by-detection has also been used in specific sports such as basketball [Lu et al., 2013], hockey [Okuma et al., 2004] and soccer [Breitenstein et al., 2009; Liu et al., 2009]. The methodology behind these models are similar to the discriminative object detection which consists of training a classifier to predict the presence or absence of the target in the frame. Given an initial object location, the goal of tracking-by-detection algorithm is to train online a classifier to distinguish the tracked object from the background. During tracking the initial sample space is updated and the classifier is retrained, so at each instant, the sampling space can be written as $\{x_0^+, x_1^+, \ldots, x_t^+, x_0^-, x_1^-, \ldots, x_t^-\}$, where the x_t^+ and x_t^- are the positive and negative samples at time t. There are various classifiers already integrated into the tracking-by-detection framework. Support vector tracking [Avidan, 2004] used the SVM classifier to distinguish foreground motion from the background. Kalal et al. [2012] proposed the long-term tracking task based on boosting classifier. The classifier is updated using all extracted appearances up to currant frame that passed the variance filter. Hare et al. [2016] employs the structured SVM to directly link the target's location space with the training samples to reduce the training time. Kernelised Correlation Filters (KCF) tracker proposed by Henriques et al. [2015] achieves the fastest and

highest performance among the recent top-performing tracking-by-detection algorithms [Li and Zhu, 2014]. The key to the KCF tracker is that the augmentation of negative samples are employed to enhance the discriminative ability of the track-by-detector scheme while exploring the structure of the circulant matrix [Henriques et al., 2012] for high efficiency.

Wu et al. [2015] performed a comprehensive evaluation of twenty five single tracking algorithms, and identified three important components that improve tracking performance. First, background information is necessary, mainly to separate background clutter from the object. Second, local models are particularly useful when the appearance of the target has partially changed, and third, the motion model is crucial for object tracking, especially when the motion of the target is abrupt.

5.1.2 Data Association

Data association models can be viewed as a multi target-management system that maintains the multiple target identities over the course of tracking. In general, data association is the process of matching information of newly observed objects (measurement) with previously observed information (state). This information can be object identities, appearance, positions, or trajectories [Betke and Wu, 2016]. This section reviews two main data association models that are broadly used in sport application, known as the classical model and the Markov Chain Monte Carlo (MCMC) approach. Main terminology used for the rest of this section:

1. *Target* is an object of interest (e.g. player, car, ball).

2. *State vector* or simply *state*, $\mathbf{x}$, is a vector containing the relevant dynamic information about the target (e.g. position, velocities, width, height).

3. *Measurements* or *observations*, **O**, are newly observed objects at frame t (i.e. result of object detection).

4. *Track* likewise *trajectory* is a sequence of observed measurement that has been decided or hypothesised by the tracker to come from a specific target.

5. *Association* likewise *association hypothesis* is the process of estimating how likely the measurements are generated by the targets.

6. *Assignment* is the procedure of assigning measurements to existing tracks or existing tracks to the measurements.

7. *Measurement validation* likewise *Gating* is a procedure to decrease the number of association hypotheses. This procedure estimates the area that the target will be seen at the time t with regards to the past states x_{t-1}. This region is called target $Gate$ and indicates the valid measurements to contribute to the association process (Figure 5.1).

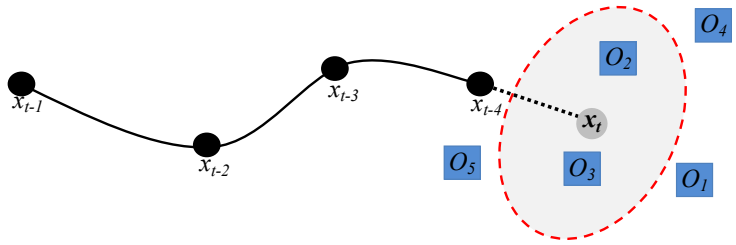

Figure 5.1: The observation (Blue boxes) within the red ellipse(gate) centred at the predicted target location $\mathbf{x_t}$ are the valid measurements to associate with the black trajectory.

8. *Birth* is the initialisation of a state upon arrival of a new target.

9. *Death* is the state termination when target leaves the scene.

Given the state of n targets up to time t, $\mathbf{X_t} = \{\mathbf{x_t^1}, \ldots, \mathbf{x_t^n}\}$, and m measurements at time t, $\mathbf{O_t} = \{\mathbf{o_t^1}, \ldots, \mathbf{o_t^m}\}$, data association algorithms first estimate how

likely each measurement $i = \{0,\ldots,m\}$ is generated from the targets $j = \{0,\ldots,n\}$ (association hypothesis) and then assign optimal measurements to existing tracks or existing tracks to the measurements. Therefore the data association task can be formulated as maximising a posterior (MAP),

$$\mathbf{X}^* = \underset{\mathrm{X}}{\operatorname{argmax}}\big(p(\mathbf{X}_t|\mathbf{O}_t)\big) \longleftrightarrow \underset{\mathrm{X}}{\operatorname{argmax}}\big(p(\mathbf{O}_t|\mathbf{X}_t)\,p(\mathbf{O}_t)\big) \tag{5.1}$$

To find the optimal solution, $\mathbf{X}^*$, it is common to treat the optimization procedure as an assignment problem. The assignment problem is the technique to exactly assign one measurement to one track in such a way the total cost of the assignment is minimised. With n targets and m observations the assignment problem is specified by,

$$\begin{aligned} &\text{Minimise} \quad \sum_{i=1}^{m} Z_{ij} A_{ij} \\ &\text{subject to} \quad \sum_{j=1}^{n} Z_{ij} = 1 \quad \forall\, i \quad \text{and} \quad \sum_{j=1}^{n} Z_{ij} \leq 1 \quad \forall\, j \end{aligned} \tag{5.2}$$

where Z is the auxiliary matrix where $Z_{ij} \in \{0,1\}$. The matrix A is called *assignment matrix*. Each element of the assignment matrix is equal to one association hypothesis so that $A \in R^{m\times n}$ and $n \geq m$. There are different approaches to solve the assignment problem of which the most well known ones are the JVC [1987], Munkres [1957] and Murty [1968] algorithms.

The classical data association algorithms were originally developed for radar tracking. The simplest data association approach is the global nearest neighbour (GNN) [Bar-Shalom, 1987] which directly assigns the closest measurements to the targets in the gate. To prevent allocating multiple measurements to a single target, GNN usually solves the assignment problem using bipartite matching (Equation 5.2). However, this model does not perform well in scenes with high clutter and fails to account for the possibility of false observation [McAnanama

and Kirubarajan, 2012]. GNN is mostly used in cases when the targets are well separated in the frame. For instance, the two approaches proposed by Dang et al. [2010] and Han et al. [2005] utilised GNN to track players in tennis videos. It is worth nothing that in tennis there is not much interaction between players and generally they move on separate paths. In a high interaction sports environment, such as hockey [Cai et al., 2006] and basketball [Lu et al., 2013], it is possible to combine GNN with robust tracking to increase the certainty in the association stage.

The Joint Probabilistic Data Association (JPDA) [Fortmann et al., 1983] solves the uncertainty problem by calculating a marginalised probability on the joint data association space. Unlike GNN, which updates the track based on the best association hypothesis, JPDA updates the target state based on the contribution of all the measurements in the gate region, where the measurement can be produced from the target or the background [Hamidreza Tofighi et al., 2015]. Both GNN and JPDA assume a fixed number of targets and cannot initialise new trajectories or terminate existing ones. Therefore they need an external mechanism to handle birth and death of targets.

Multiple Hypothesis Tracker (MHT) [Reid, 1979] is a complete data association model which is capable of initiating and terminating incoming and ongoing targets. The MHT is classified as a "deferred logic" model in which the decision about the birth of a new target and the death of an existing track is delayed until enough observations are collected from the association hypothesis. In theory, MHT requires keeping track of all associations' hypotheses between object tracks and incoming observations. As a result, if the number of observation grows, the association hypotheses can become unmanageable. In practice, the full method is computationally infeasible unless combined with pruning heuristics [Oh et al., 2004].

The Recursive Bayesian data association based on Markov chain Monte Carlo

sampling (MCMC) or Particle filtering are preferred for many vision-based multi target management systems. This is because they are computationally practical and target appearances such as shape, colour and texture can be easily incorporated into the sampling process [Yang et al., 2005]. Jaward et al. [2006] proposed a simultaneous detection and tracking based on the colour appearance of the targets. This approach estimates the association hypothesis by sequential Monte Carlo sampling, while using JPDA for data association. Khan et al. [2005] introduced multiple object tracking framework based on particle filter using Rao-Blackwellisation sampling schemes [Casella and Robert, 1996].

Due to occlusion the optimal targets sampling using MCMC is difficult, especially for long term object tracking [Ge and Collins, 2008]. The tracklet based sampling approach tackled the above sampling problem. The tracklets are the fragment of the trajectory which provides better sampling mechanism in comparison to the traditional one-to-one observations-targets mapping [Yu et al., 2007a]. The tracklets are less susceptible to drift and occlusion and they can be generated by simple tracking models. These models are known as Markov chain Monte Carlo Data Association (MCMCDA) [Oh et al., 2004; Yu et al., 2007b; Ge and Collins, 2008; Prokaj et al., 2011]. Ge and Collins [2008] used MCMCDA to produce the soccer player trajectory. This model groups tracklets that belong to the same player using properties such as colour appearance, object size, spatial proximity and velocity coherence.

The single object tracking approaches can be used to track any kinds of objects under different conditions. There is a general agreement in the evaluation of single object trackers performance [Wu et al., 2015]. However, there is not any standard method to evaluate multi-object tracking performance [Bernardin and Stiefelhagen, 2008]. One reason for this issue is that multi-object trackers are designed to track predefined objects (human, car and face) and in the certain environment. Furthermore, certain conditions can be assumed in each environ-

ment that are not valid for another application. For instance, Yang et al. [2005] assumed pedestrians can be occluded by one another in a short period of time, or Jiang et al. [2007] proposed a human tracking system by assuming the camera is static for the duration of tracking.

Apart from all the above difficulties, the most important issue directly related to the strategy of occlusion handling in multi-object trackers algorithms. All multi-object trackers use the data association technique to reacquire the lost objects after the occlusion. Obviously, the data association performance greatly depends on the quality of the detector [Milan et al., 2013]. Thus, to have a fair evaluation, a standard detection should be provided for various trackers. Bernardin and Stiefelhagen [2008] attempted to generate common metrics for measuring the performance of multiple object trackers that used the same object. However, the different tracking system requires different types of object as input and consequently, in practice these kinds of approaches are not feasible to establish standard metrics for all multiple object tracker under various conditions [Milan et al., 2013].

The literature studied show that the majority of tracking algorithms focused on local object information. During occlusion, however, local object information does not properly represent the true properties of objects, and without the data association, this issue leads to tracking failure. In the next section, we propose a framework that combines the group motion information with local object properties to continually track objects under occlusion without using any detection algorithms.

5.2 Proposed Tracking Framework

Local object information, such as appearance and motion features, are useful when the target is not occluded by other elements in the scene. However, when

the local information does not properly represent the object properties, due to occlusion and background clutter, the tracking often fails. To overcome the difficulties that exist in the horse racing environment, a multi–object tracking framework is proposed to combine multiple cues such as local object information, background characteristic and group motion dynamic to improve tracking under challenging environments. The background information is necessary for efficient tracking, mainly to separate background clutter from the contenders. Local appearance models are critical when the target appearance changes under partial occlusion or deformation. Motion models are relevant when the contenders are obscured by background elements such as trees, towers and bars.

The proposed tracking model contains three main modules, namely point level processing, contenders localisation and data association, as illustrated in Figure 5.2. The point processing block is based on the assumption that in reality the sample points are rarely independent and they are parts of bigger units, namely the objects that are being tracked [Kalal et al., 2010], and therefore the sample points should have same motion and similar colour distribution. Consequently, the location of the object can be estimated by tracking the points that sample from the same object. The point processing block aims to find the best points from noisy sample space by a series of filtering stages.

In contender localisation, two different strategies are used, namely object based and group based localisation. Object based localisation is applied when the sample points correctly represent the local object motion and appearance. The group based localisation is applied when the local information does not properly represent the object, mainly due to occlusion and background clutter.

The data association is applied as a multi target management system to maintain multiple jockeys' identities over time, to initialise the tracking, terminate trajectories and update each contender's appearance model.

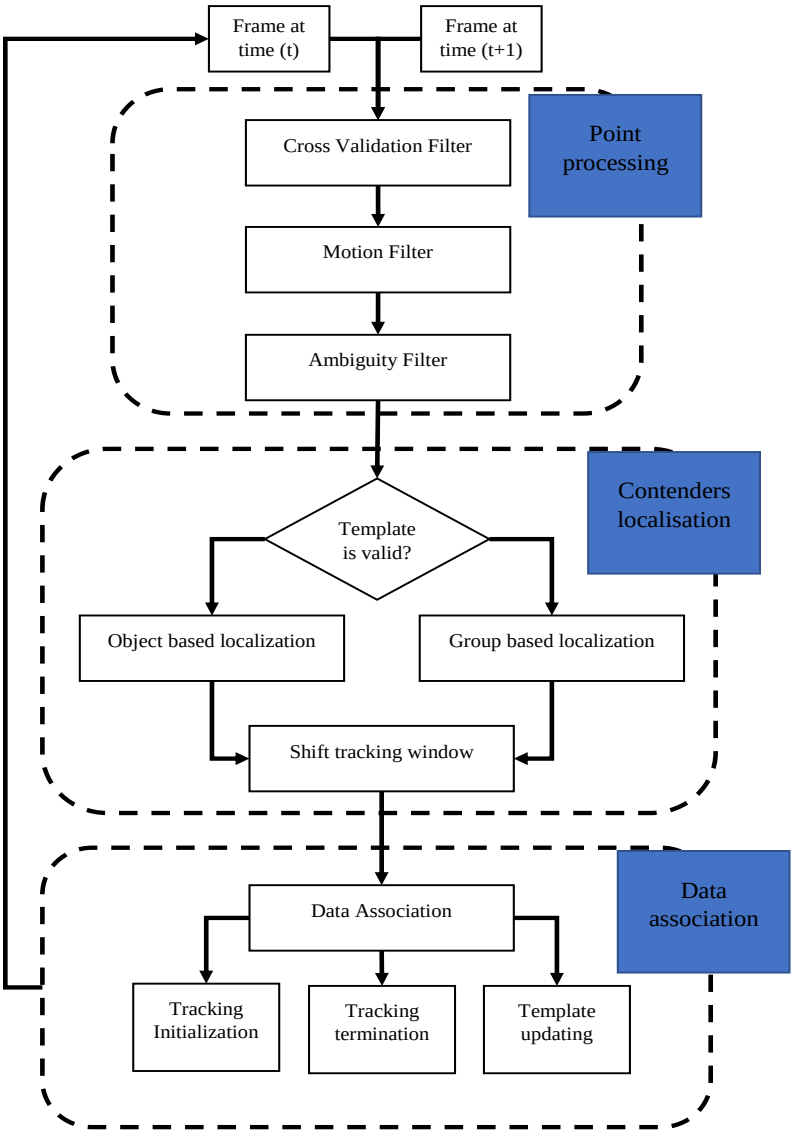

Figure 5.2: The block diagram of proposed tracking model. The point processing block aims to find the best points that represent the object property. For object localisation, two strategies are used: Object based localisation is applied when the sample points properly represent the object local properties and when the local information dos not properly represent the object property the localisation is switched to the group based model. The data association block maintains multiple jockeys' identities over time, initialise a tracking, terminates trajectory and finally updates the contender's appearance model.

5.2.1 Features Extraction

To overcome appearance ambiguities and to handle the occlusion, the object features are extracted from three sampling levels: point level, object level and group level. The definition of these features are as follows:

1. *Object template* (w) refers to the rectangular window around the object; it is also referred to as the tracking window.

2. *Point level motion cues* (U_p) are the flow of the sample points extracted from on the object template where $U_p = (u_{p,x}, u_{p,y})$ are the motion cues for a given point p. Particularly, given the point $p = (p_x, p_y)$ on the selected template at frame I, we estimate its corresponding location $p' = (p_x + u_{p,x}, p_y + u_{p,y})$ in the frame $I+1$ using the iterative pyramids Lucas-Kanade method [Bouguet, 2001]. The sample points are extracted using Shi and Tomasi corner detection [Shi and Tomasi, 1994].

3. *Point level colour cue* (H_p) refers to the colour distribution of 15×15 rectangular patches around sample points. Point level colour cues are calculated from the histogram of hue and saturation channels in HSV colour space.

4. *Object motion model*, $U_o = (u_{(o,x)}, u_{(o,y)})$, refers to the tracking window displacement. The object motion model is estimated by taking the average of all motion vector (J) at point level for the given object o by,

$$U_o = \frac{\sum_i^J U_{p,i}}{J} \tag{5.3}$$

5. *Object colour model* (H_o) refers to the colour distribution of the tracking window. The object colour model is calculated from the histogram of hue and saturation channels in HSV colour space.

6. *Group motion model,* $U_g = (u_{(g,x)}, u_{(g,y)})$, is estimated by taking the average motion models of all m objects using,

$$U_g = \frac{\sum_i^m U_{o,i}}{m} \tag{5.4}$$

7. *Object shape model* refers to cap shape structure. It is trained off-line with HOG features and linear SVM as discussed in Chapter 4. The object shape model is used with data association to initiate, terminate and update tracks.

8. *Object relative speed* $U_v = (u_{(v,x)}, u_{(v,y)})$, refers to the relative speed of the individual object with respect to the group motion model, viz

$$U_v = \frac{U_o}{U_g}. \tag{5.5}$$

9. *Background motion,* $U_b = (u_{(b,x)}, u_{(b,y)})$, refers to the dominant motion in the frame. The background motion is estimated using the algorithm described in Chapter 3.

5.2.2 Point Processing

The location of the object can be estimated by tracking the sample points that are distributed over the object surface. The point processing block aims to find the sample points that well represent the object using three stage filtering, namely the cross validation filter, the motion filter and the ambiguity filter.

5.2.2.1 Cross validation filter

In the cross validation filter, the forward-backwards error described by Kalal et al. [2010] is used to estimate the stability of motion cues at point level. With the sample point p at frame I and its corresponding location p' in the frame $I+1$, the

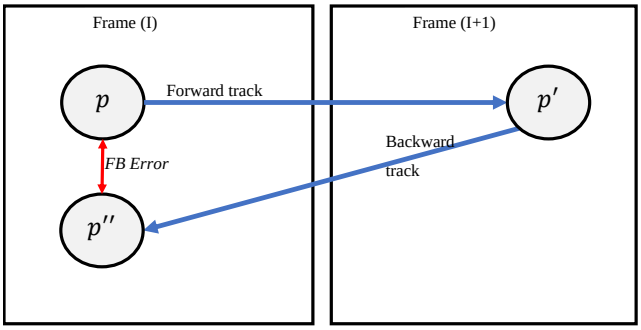

Figure 5.3: Given the sample point p at frame I and its corresponding location p' at frame $I+1$, we compute the backward flow of point p' to the frame I (p''). The forward–backward error is defined as the Euclidean distance between the original point and the forward-backward prediction (red arrow).

backwards flow of point p' to the frame I is computed. The forward-backwards error ε_{FB} of a point p is defined as the Euclidean distance between the original point and the forward-backward prediction. This process is illustrated in Figure 5.3. In the filtering stage the points are removed if their forward–backwards error is larger than 2 pixels [Kalal et al., 2010], that is

$$p = \begin{cases} 0 & \varepsilon_{\text{FB}} \geq 2 \text{ pixels} \\ 1 & \text{elsewhere.} \end{cases} \tag{5.6}$$

5.2.2.2 Motion filter

Knowing the background motion (U_b), object motion (U_o) and the motion of sample points (U_p), we estimate how likely the sample point is produced from the background by,

$$p' = \begin{cases} \text{Background} & d(U_p, U_b) < d(U_p, U_o) \\ \text{Foreground} & \text{elsewhere.} \end{cases} \tag{5.7}$$

Figure 5.4: The effect of the motion filter. The arrow in the left image indicates the notable area before occlusion. The image in the middle illustrates the result of tracking with motion filter, and the right image shows the result of tracking without motion filter.

Here d is the Euclidean distance function. The effect of the motion filter is illustrated in Figure 5.4.

5.2.2.3 *Ambiguity filter*

When the tracked object is occluded by another, some sample points that belong to one object might move to the other object which eventually causes tracking drift. For the occlusion problem of k objects, this task can be formulated as maximising a posterior by

$$\mathbf{k}^* = \underset{\mathbf{k}}{\operatorname{argmax}} S. \tag{5.8}$$

The vector $S = \{s_0, s_1, \ldots, s_m\}$ indicates how likely sample points p are generated from each object o. To measure the similarity, s, the histogram intersection, proposed by Swain and Ballard [1991], is used. It is especially suited to comparing histograms for recognition in our case because it does not require the accurate separation of the object from its background or occluding objects in the foreground. Having the object colour distribution, H_O, and point level colour distribution, H_p, the similarity score is found by intersection using

$$s = \sum_i \min(H_p(i), H_O(i)), \tag{5.9}$$

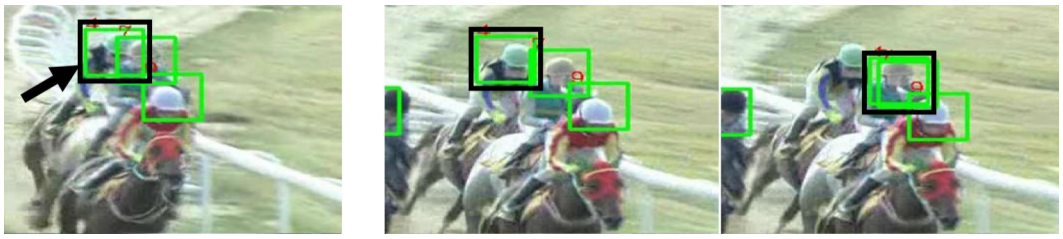

Figure 5.5: The effect of ambiguity filter. The arrow in the left image indicates the notable contenders before occlusion. The image in the middle illustrates the result of tracking using ambiguity filtering. The image in the right shows the result without using ambiguity filtering.

where i is the bin number of histogram. The effect of the ambiguity filter is illustrated in Figure 5.5.

5.2.3 Contender Localisation

We use two different strategies to locate the objects in the frame, namely object based localisation and group based localisation. Object based localisation predicts the new location of the tracking window by finding the mass centre of the weighted sample point. However, when the object is obscured by some other element in the video, or has an unpredictable motion, the sample points no longer represent the object template. In this case the group motion flow is used to estimate the location of objects. Which strategy is used is determined by estimating the quality of the object template inside the tracking window. If more than 60% of the samples points are filtered out in the filtering stages, the object template is not valid and the group based localisation is triggered.

5.2.3.1 Object based localisation

In object based localisation, the quality of the points (after point processing stages) are estimated by finding the colour similarity between each sample point and the object template s' using Equation 5.9 and 50% of the points with lowest similarity

matches are removed. From the remaining points the centre of mass for the new tracking window is calculated by

$$C_x = \frac{\sum_i^J s'_i p_{x,i}}{\sum_i^J s'_i}, \qquad C_y = \frac{\sum_i^J s'_i p_{y,i}}{\sum_i^J s'_i} \tag{5.10}$$

where J is the number of remaining sample points.

5.2.3.2 Group based localisation

To find the approximate location of object using group information, three values are estimated: *last valid object motion*, *group motion flow*, and *object relative speed to the group*. The last valid object motion model is referred to last estimated motion vector of the objects that do not suffer from occlusion or unpredicted motion as calculated in Equation 5.3. Group motion model (d_g) is estimated by taking the average motion models of all valid objects using Equation 5.4 and the object relative speed is estimated by mean of Equation 5.5. Using above value the new location of the object is approximated by moving the tracking window by the relative speed of objects to the group by

$$C'_x = C_x + u_{(v,x)} u_{(g,x)}, \qquad C'_y = C_y + u_{(v,y)} u_{(g,y)} \tag{5.11}$$

where C' is the new centre of the tracking window.

It is worth noting that with smooth camera movement, the last valid contender motion may be sufficient to predict the next location of the tracking window. However, as contender motion highly depends on the camera movement, which can be unpredictable, the last valid contender motion may not be reliable enough to predict the new tracking window. In contrast, the motion and direction of contenders are all similar. Thus the relative speed of contenders to the group is more informative because it is independent of camera motion. The ef-

fect of unpredicted camera movement for these two scenarios clearly can be seen from the video in Figure 5.6 when the camera start to zooming (timer≃ 0 : 12).

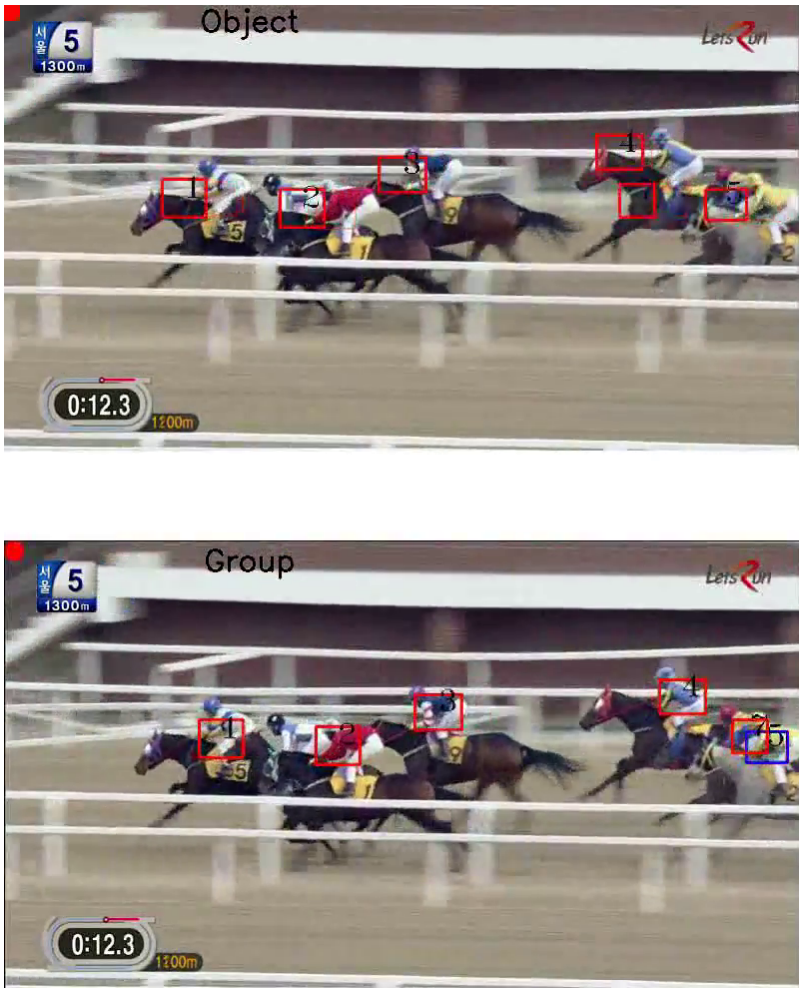

Figure 5.6: The result of tracking using last valid contenders motion is shown in top video, and the bottom video is the result of group based localisation. As can be seen from the sample videos, the relative speed of contenders to the group give better approximation of the tracking window.

Figure 5.6 contains embedded video.

5.2.4 Data Association

Initialising the tracker for contenders from the noisy and uncertain detection is very challenging. We used data association to initialise the trajectory and maintain multiple jockeys' identities over the course of tracking. Let the trajectories of n jockeys' caps at time t represent by the sequence of states, $\mathbf{X_t} = \{\mathbf{x_t^1}, \ldots, \mathbf{x_t^n}\}$, and the measurements O be the output of the the cap detection algorithm (see Chapter 4) at time t, $\mathbf{O_t} = \{\mathbf{o_t^1}, \ldots, \mathbf{o_t^m}\}$. Then the data association task is to assign n tracks to m new detected caps (object) with capability of initiating new contenders and terminating false trajectories. This problem can be simplified by building the assignment matrix, A, given by

$$A = \left[\begin{array}{c|c} C_{m,n} & B_{m,m} \\ \hline T_{n,n} & 0_{m,n} \end{array}\right] = \left[\begin{array}{cccc|cccc} c_{1,1} & c_{1,2} & \cdots & c_{1,n} & b_{1,(n+1)} & 0 & \cdots & 0 \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} & 0 & b_{2,(n+2)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} & 0 & 0 & \cdots & b_{m,(n+m)} \\ \hline t_{(m+1),1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ -\infty & t_{(m+2),2} & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -\infty & 0 & \cdots & t_{(m+n),n} & 0 & 0 & \cdots & 0 \end{array}\right] \tag{5.12}$$

where the columns of the matrix A correspond to the tracks, and the rows to the objects so that each element of the assignment matrix is equal to one association hypothesis. The sub matrices C , B and T are responsible to assign n tracks to m measurements, initiating new and terminating ongoing trajectories. Let i be the row and j be column of sub matrices in A then the association hypothesis of these matrices are defined by

1. Matrix, C, is the association hypothesis to assign objects to the track, namely

$$c_{i,j} = \begin{cases} 1 + \frac{1}{d_{i,j}} & \text{if} \quad o_i < r \\ 0 & \text{elsewhere,} \end{cases} \tag{5.13}$$

where d is the Euclidean distance between the centre of the detected cap and tracking window and r is the Gate radius (see Section 5.1.2).

2. Matrix, B, is the association hypothesis to initialised new potential tracks, given by

$$b_{i,(n+j)} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{elsewhere,} \end{cases} \tag{5.14}$$

3. Matrix, T, is the association hypothesis to terminate tracks, and is

$$t_{(m+i),j} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{elsewhere.} \end{cases} \tag{5.15}$$

Once the assignment matrix is built, data association is treated as an assignment problem. An assignment problem is a special case of the optimization problem that deals with assigning one measurement to one track in such a way the total cost of the assignment matrix is minimised (see Equation 5.2). There are many approaches to solve the assignment problem of which the best known ones are the JVC [1987], Munkres [1957] and Hungarian [1968] algorithms. With every newly observed cap, the tracks are updated according to three track characteristics, namely (1) potential track when there is not sufficient evidence to prove the track belongs to the true contender, (2) confirmed track is a track that belong to a valid contender, and (3) false track is a track that comes from false alarm and should be deleted.

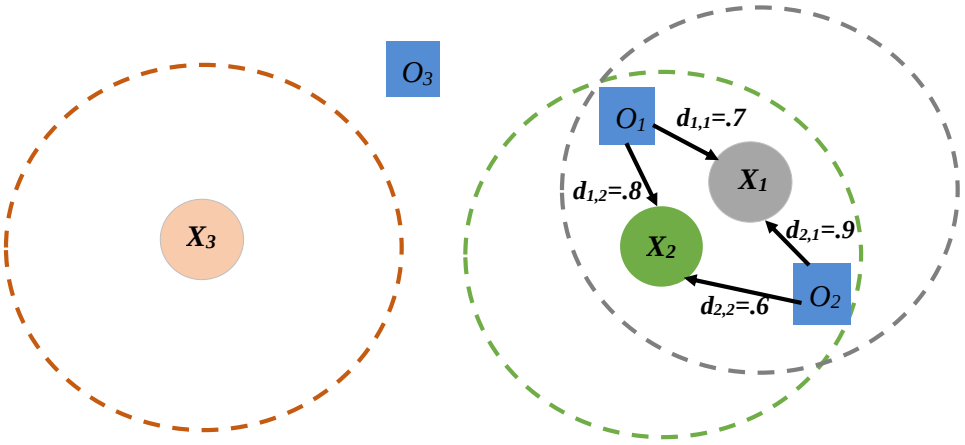

Figure 5.7: Example of the data association problem with three measurements and three states. The blue rectangles show the centre of the detected caps and the circles indicate the centre of the tracking window. Dash circles are the Gate corresponding to each state.

Let us assume there are three measurements and three states at frame I as shown in Figure 5.7. The rectangles indicate the centre of the detected caps, circles indicate the centre of the tracking window and the dashed circles are the Gates corresponding to each state. Based on the information in Figure 5.7, the association matrix A can be built as follows:

$$A = \left[\begin{array}{cccc|ccc} & \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & & & \\ \mathbf{o_1} & 1.7 & 1.8 & 0 & 1 & 0 & 0 \\ \mathbf{o_2} & 1.9 & 1.6 & 0 & 0 & 1 & 0 \\ \mathbf{o_3} & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 1 & & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right], \tag{5.16}$$

and by applying The Hungarian [1968] algorithm to the matrix A the optimal as-

signment matrix (A') is obtained as

$$A' = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right], \tag{5.17}$$

which is interpreted as follows:

1. From the top–left sub–matrix, cap O_2 is produced from track X_1 and its state should be updated.
2. likewise, cap O_1 is produced from track X_2 and its state should be updated.
3. From the top–right sub–matrix, cap O_3 has *potential* to be a new track.
4. From bottom -left sub–matrix, track X_3 has *potential* to be a false alarm.
5. The lower right sub–matrix should be ignored.

For cases *3* and *4*, we mention *potential* new track and false alarm because we delay the decision about the birth of a new target and death of the existing track until enough observations are collected from the association hypotheses. It is considered the new contender is arrived into scene, birth, if the new measurement is assigned into the same potential track three times over a five frame period, and eliminate the trajectory if the total false alarm in one track is more than half of the length of that track at the end of tracking course. The visual demonstration of the proposed tracking model in one turning segment is illustrated in Figure 5.8.

Figure 5.8 contains embedded video.

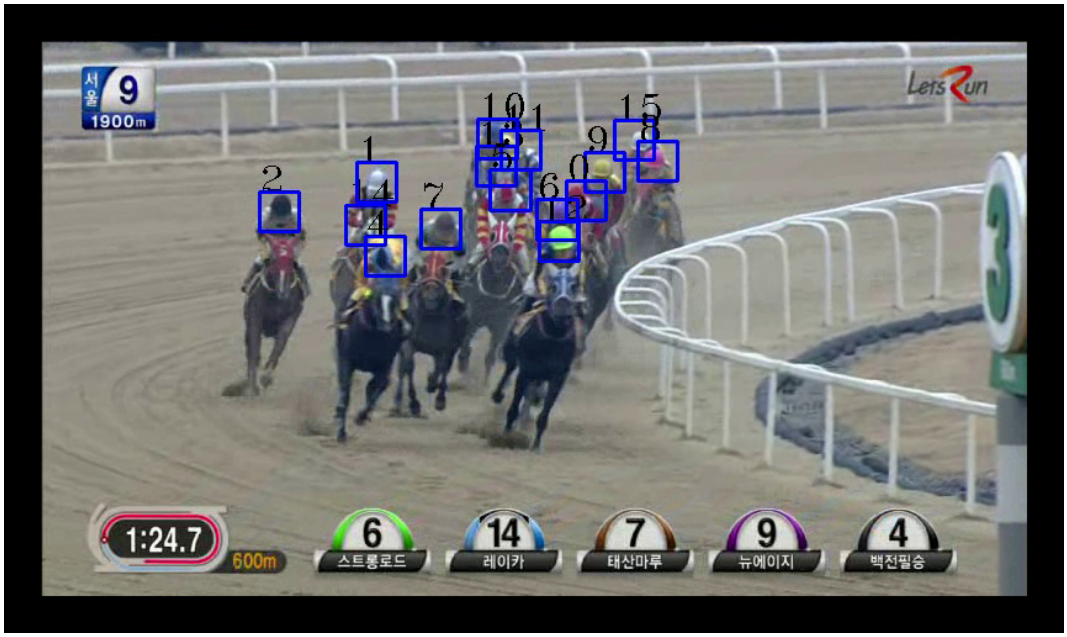

Figure 5.8: Visual demonstration of the proposed tracking model in one turning segment.

This page contains embedded video.

CHAPTER 6

Evaluation

According to section 1.2 of chapter 5, multi-object trackers handle the occlusion problem by reacquiring of lost objects after occlusion. However, the proposed tracking system uses a completely different approach. Our model does not employ any object detection algorithms for handling the occlusion. In fact, here we follow the methodology of single object tracker, adding group motion information to continually track objects under extreme occlusion.

To show how tracking performance can be improved by adding the group motion information, we examine the result of our tracking models with kernel correlation filter (KCF) in the first section of this chapter. We select KCF because of two main reasons. Firstly, KCF achieves the highest performance among the recent top-performing trackers [Henriques et al., 2015], and secondly, KCF is a single object tracker and standard evaluation methodology [Milan et al., 2013; Wu et al., 2015] can be used to examine the performances of both models. To test the robustness of the proposed model, we used two different types of objects, pedestrians and jockeys, under severe occlusion scenarios. In this experiment, we manually select the objects in each video to initialise the tracking, and we use no automatic initialisation or object reacquiring process during the tracking period. The second section of this chapter is analysing the overall performance of the proposed horse racing system with regards to two aspects: automatic contender initialisation and contenders tracking.

6.1 Tracking Evaluation

The main purpose of the tracking evaluation is to show how the group property and background motion information improve the tracking performance under occlusion and background clutter. The robustness of the proposed tracking model is examined with a state of art tracking algorithm, namely KCF, which achieves the highest performance among the recent top-performing trackers [Henriques et al., 2015]. To do this evaluation, three entities are defined: the tracker output, T, the correct result or the ground truth, GT, and distance function, d, which is a measure of the similarity between tracker output and the ground truth [Milan et al., 2013]. The tracker output and the ground truth are delimited by bounding boxes. The relative overlap of the ground truth and the tracker output determines the tracking accuracy (see Figure 6.1) according to

$$d(T,GT) = \frac{T \cap GT}{T \cup GT}. \tag{6.1}$$

When $d = 0$ there is no overlap between ground truth and tracking output bounding boxes, whereas $d = 1$ occurs when the two bounding boxes are identical. An object is considered correctly tracked if the tracking output is within a distance threshold of ground truth where the most commonly used threshold to consider correct tracking is 0.5 [Milan et al., 2013].

Eight challenging videos are used for this evaluation. Four videos are taken from horse race broadcast and four videos are of a group of five people walking together passing obstacles such as trees and other persons in the scene. The ground truth was built by manually extracting the bounding box for each object at every tenth frame of the video. It should be noted that, to have fair comparison, we removed the data association block from our tracking system. This is because, firstly, the aim of this section is to evaluate the performance of object

Figures 6.3, 6.5, 6.7, 6.9, 6.11, 6.13, 6.15, and 6.17 of this section contain embedded videos.

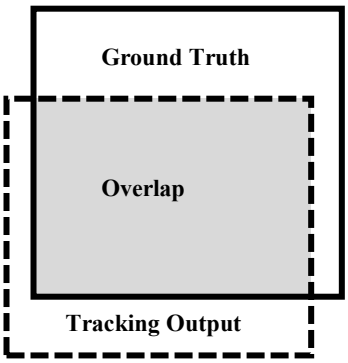

Figure 6.1: If both the tracker output and the ground truth are defined by bounding boxes. The intersection over union of the ground truth and the tracking output, determines the tracking accuracy.

tracking alone and not data association and secondly, the KCF is tracker system and did not use any data association mechanism. To initialise the tracking, the contenders at the beginning of each video are manually selected. The video properties, including video name, number of contenders and the length of each video is tabulated in Table 6.1.

The tracking performance for each individual contender per video is shown in Figures 6.2 to 6.16. The red lines in the graphs are the result of the proposed tracking model, the black lines are the result of the KCF tracker and the blue lines indicate the distance threshold. The distance threshold is set to 0.5.

As the graphs in Figure 6.2 and Figure 6.4 show, both models handled the contenders tracking in Race–1 and Race–2 well. This is because the camera movement for these two videos are smooth and there is only partial occlusion in the videos. However, the proposed model shows a significant drops in localisation accuracy for the contender *six* in the Race–1, at around frame 50. This is because, as it explain in Section 5.2.3, the proposed tracking model constantly checks the validity of the object template and if it is not valid the group based localisation will be used, thus when the contenders go under occlusion (full or partial), the

Table 6.1: The Properties of Test Videos

Video ID	**Number of objects**	**Number of frames**	**Significant characteristics**
Race–1	12	250	Moving camera, Partial occlusion
Race–2	11	350	Moving camera, Partial occlusion
Race–3	11	350	Moving camera, Partial & full occlusion
Race–4	12	250	Moving camera Partial & full occlusion
Waikato–1	5	290	Static camera Partial & full occlusion, Background clutter
Waikato–2	5	500	Moving camera Partial & full occlusion, Background clutter
Waikato–3	5	380	Static camera Partial & full occlusion
Waikato–4	5	400	Static camera Partial & full occlusion

tracking bounding box is swinging due to the switching between object based and group based localisation until objects recover from occlusion. The poor localisation performance of KCF tracker under occlusion slowly appears in Race–3 and Race–4, where contenders 2 and 9 in Race–3 and contender 6 in Race–4 are fully covered by other contenders in the race as shown in Figure 6.6 and Figure 6.8.

The benefit of switching localisation strategies is clear in videos Waikato–1 and Waikato –2. As shown in Figures 6.11 and 6.13, the performance of both models are identical until just before the objects walk behind the tree. It can be seen from Figures 6.10 and 6.12 the KCF tracker failed to track four of the five objects when they are occluded by the tree. This poor performance is due to two main reasons: first, the KCF algorithm did not encode the background mo-

tion information and therefore it does not distinguish between the background element (tree) and tracked object. This leads to the second and bigger problem which exists in almost all tracking-by-detection algorithms, as highlighted above in Section 5.1.1, the goal of the tracking-by-detection algorithm is to continually train the online classifier to distinguish the tracked object from the background, but each training update can introduce error. To be specific at the point of occlusion, the tree is considered a tracked object and the classifier is trained with the wrong features which leads to the tracking drift. The effect of this drawback is also seen in Waikato–3 and Waikato–4 when the tracked object occluded with other objects as shown in Figures 6.14 and 6.16.

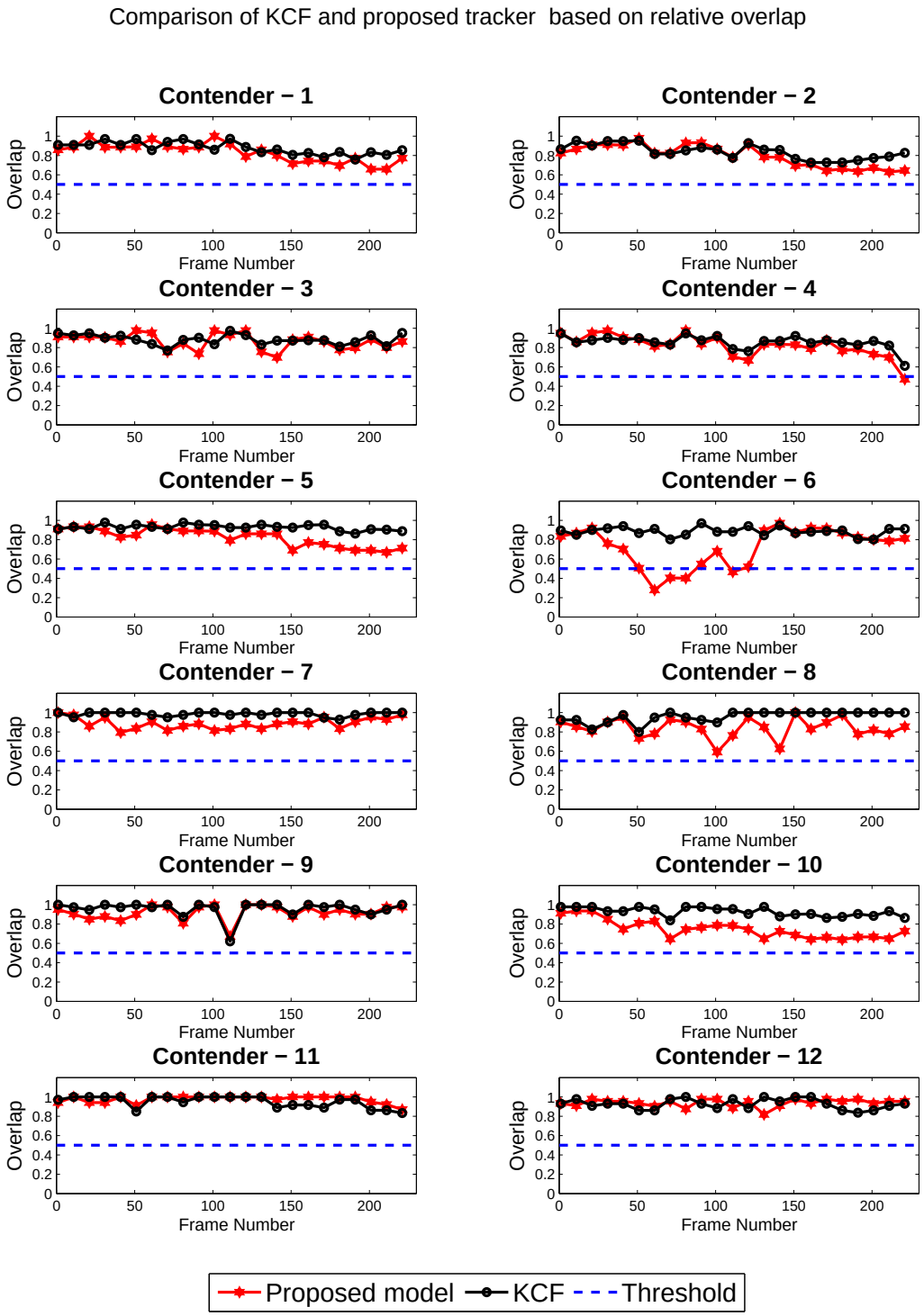

Figure 6.2: Comparison result between KCF and proposed model for Race–1. Both models, have essentially identical performance except for contender 6 where accuracy of the proposed model drops at around frame 50.

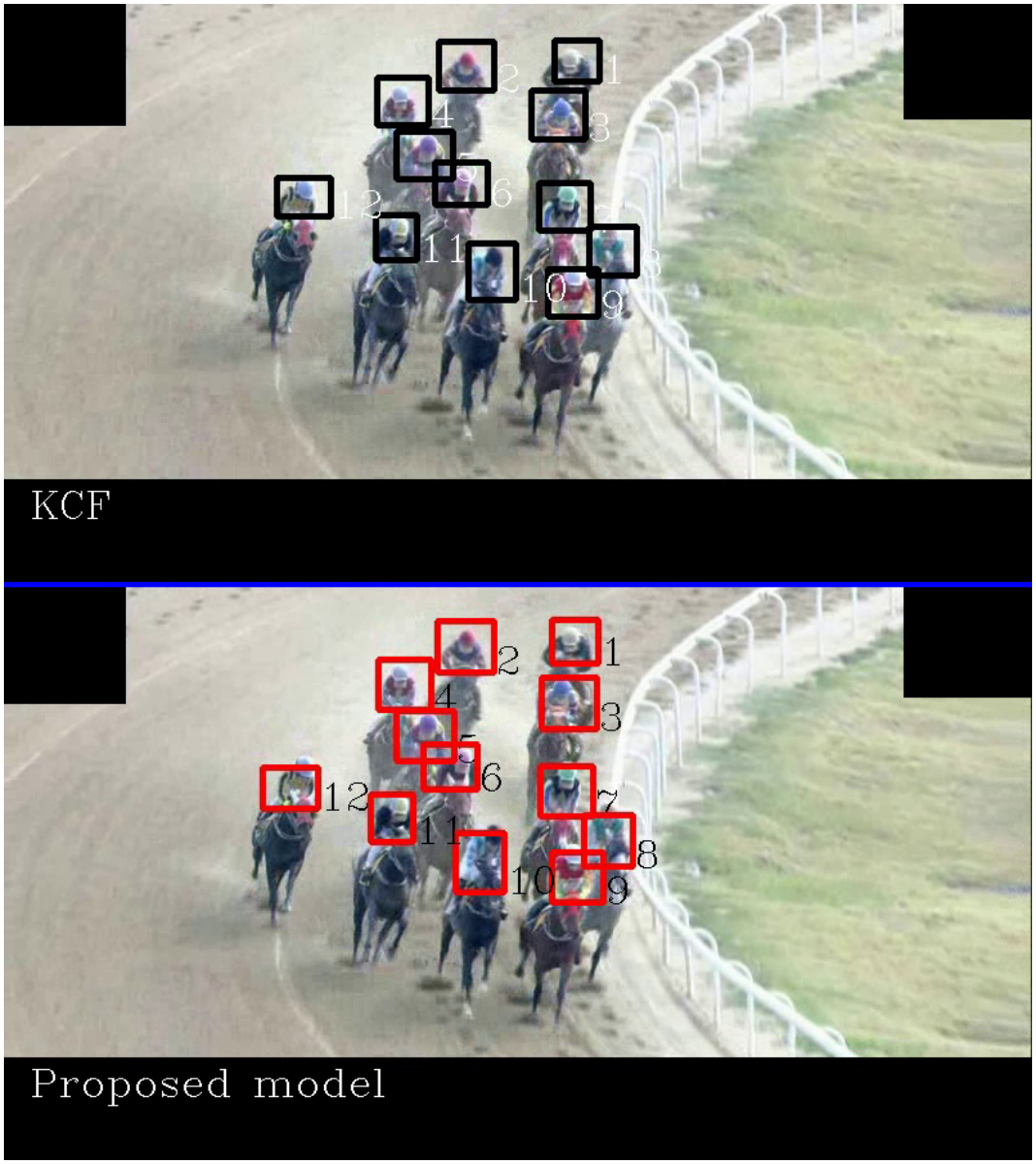

Figure 6.3: Visual tracking result for Race–1. When the contenders goes under occlusion (check contender number six) the tracking bounding box is swinging due to the switching between object based and group based localization until objects recover from occlusion.

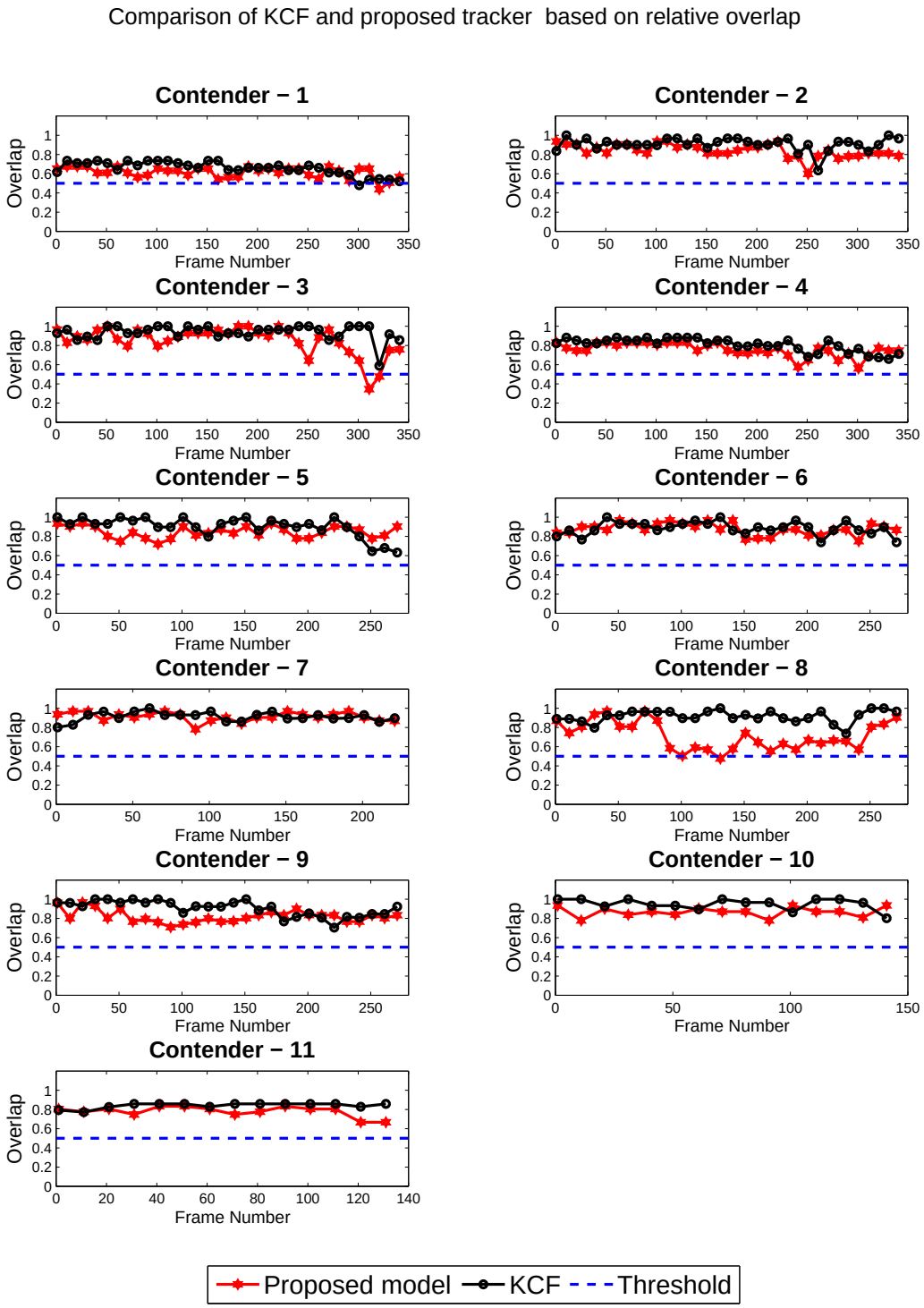

Figure 6.4: Comparison result between KCF and proposed model for Race–2. Both models, have essentially identical performance mainly due to smooth camera movement.

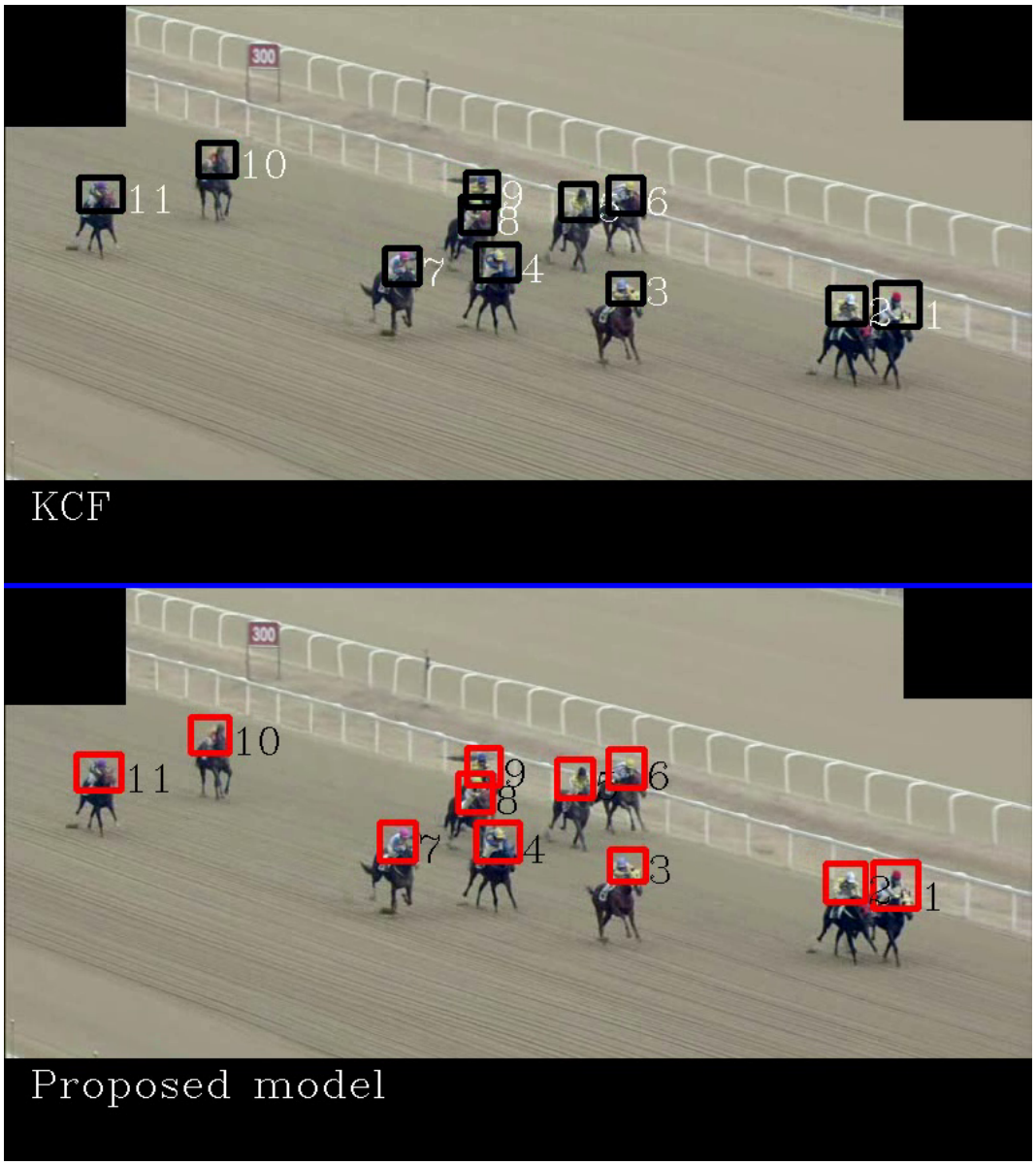

Figure 6.5: Visual tracking result for Race–2. As can be seen the tracking boxes of two models are visually identical to each other.

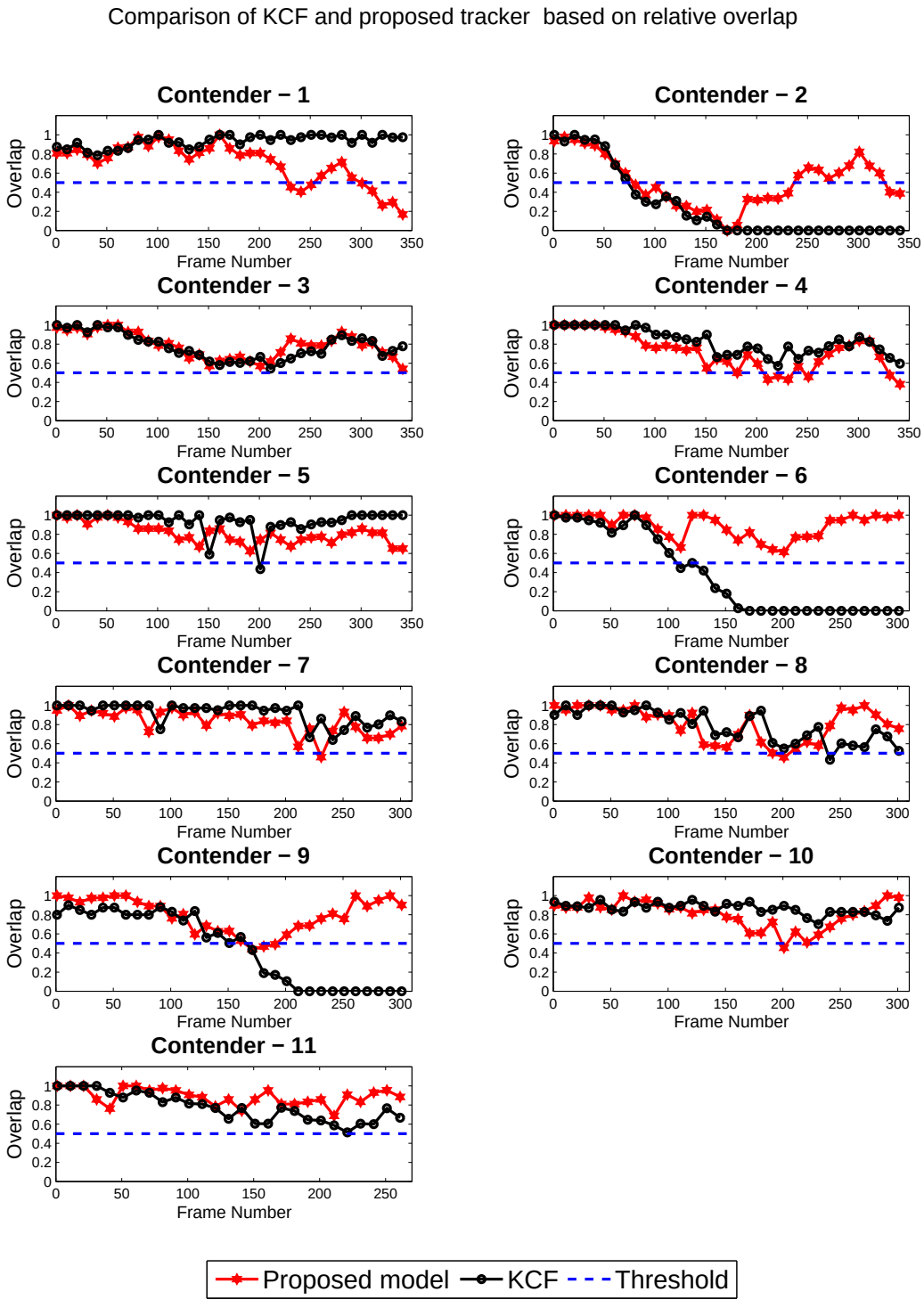

Figure 6.6: Comparison result between KCF and the proposed model for Race–3. The KCF tracker totally loses Contenders 2, 6 and 9 due to occlusion.

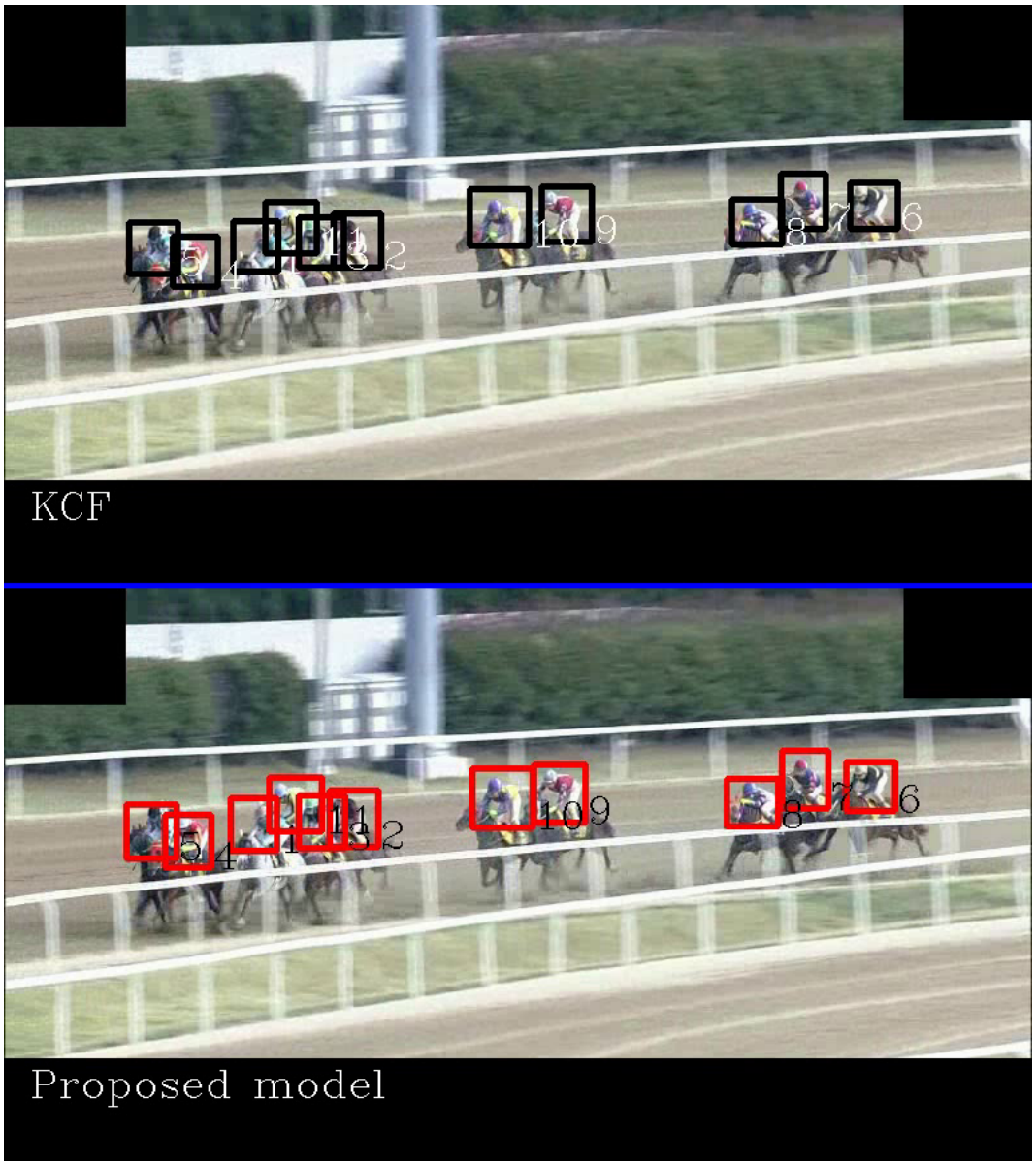

Figure 6.7: Visual tracking result for Race–3. The KCF tracker totally loses Contenders 2, 6 and 9 due to occlusion.

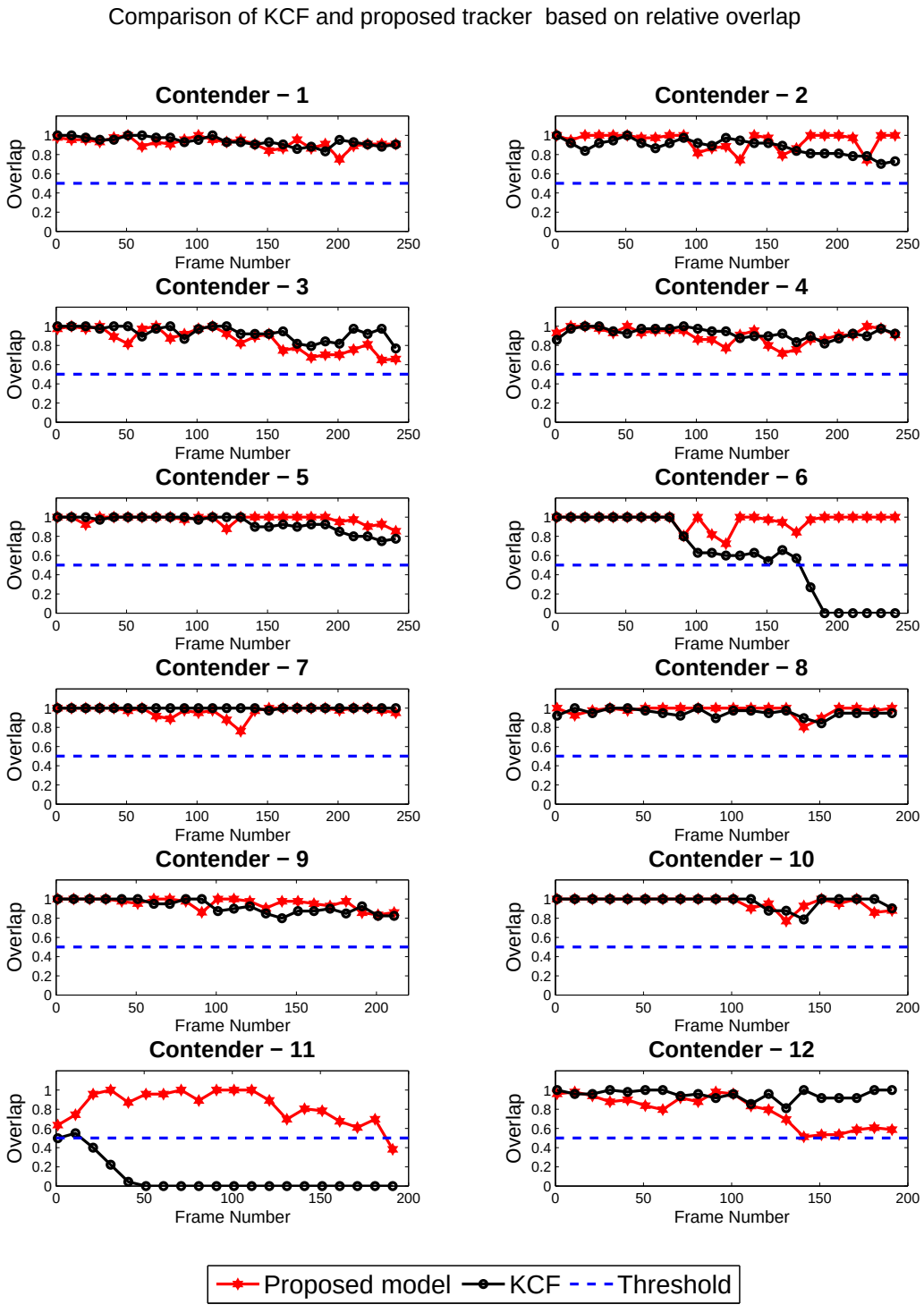

Figure 6.8: Comparison result between KCF and proposed model for Race–4. The KCF tracker totally loses Contenders 6 and 11 due to occlusion.

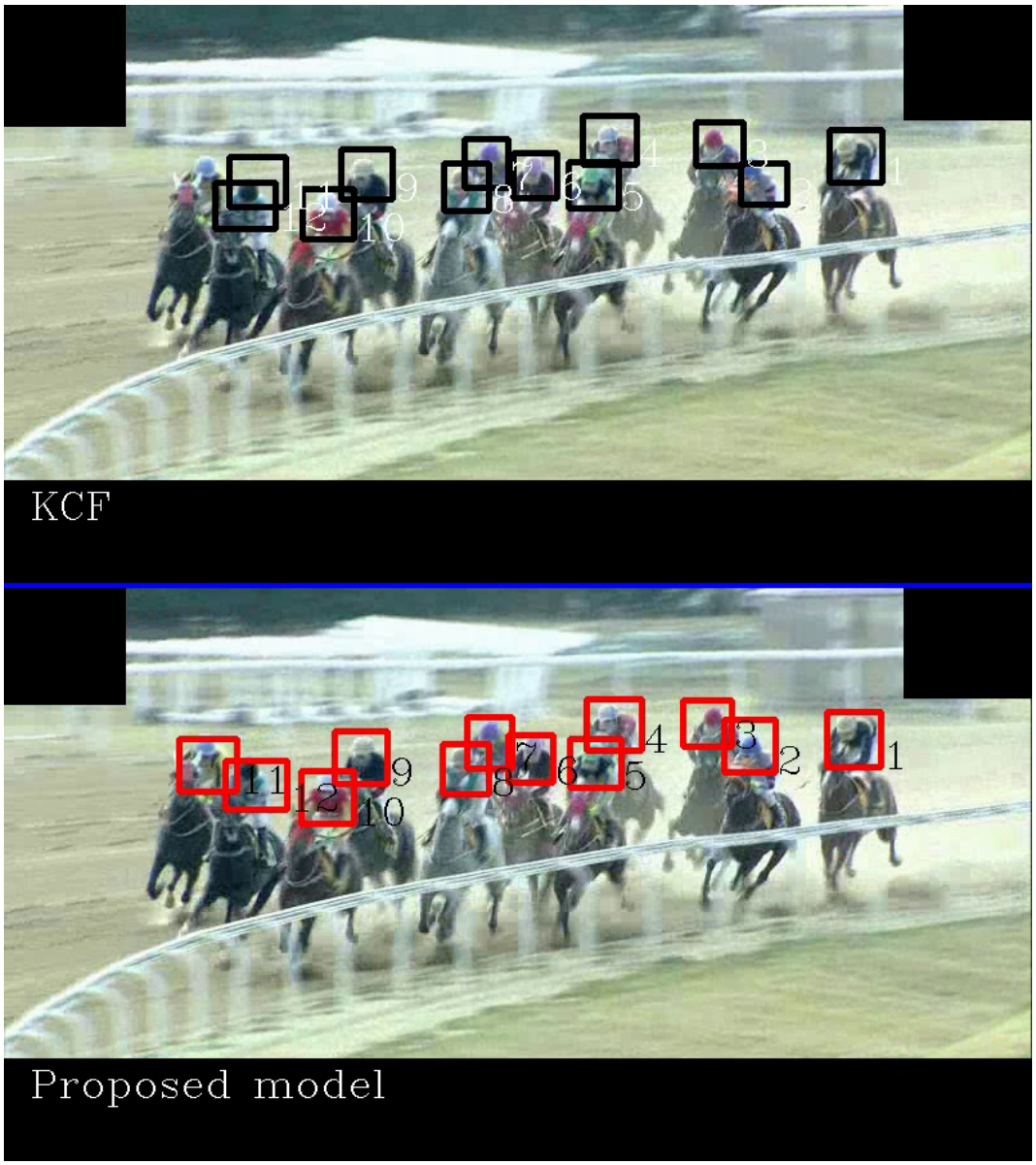

Figure 6.9: Visual tracking result for Race–4. The KCF tracker totally loses Contenders 6 and 11 due to occlusion.

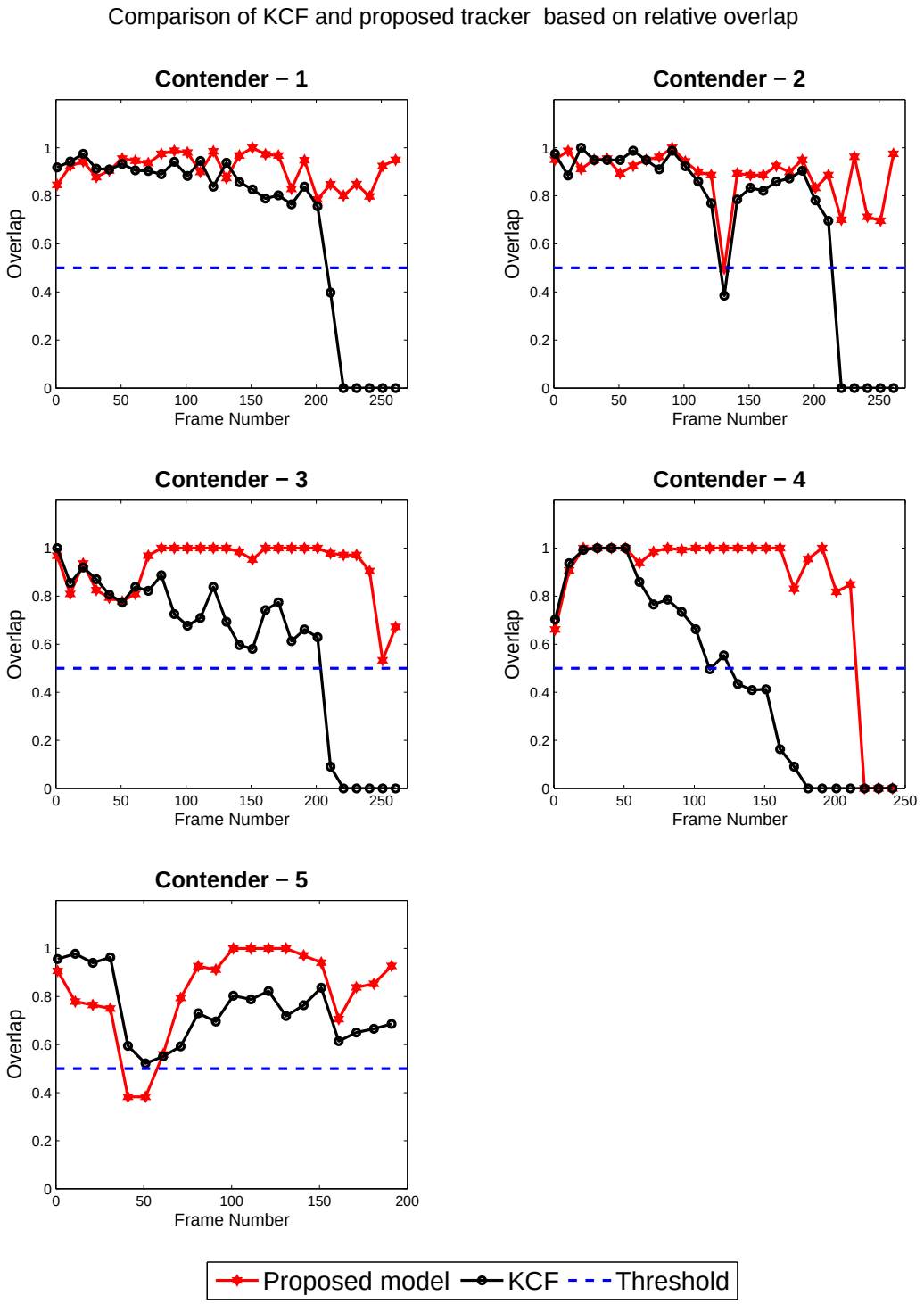

Figure 6.10: Comparison result between KCF and proposed model for Waikato–1. The KCF tracker failed to track four of the five objects when they are occluded by the tree.

Figure 6.11: Visual tracking result for Waikato–1. The KCF lost objects after they walk behind the tree. This is because the KCF algorithm did not encode the background motion information therefore the tree is considered as a tracked object and the classifier is trained with the wrong features which leads to the tracking drift.

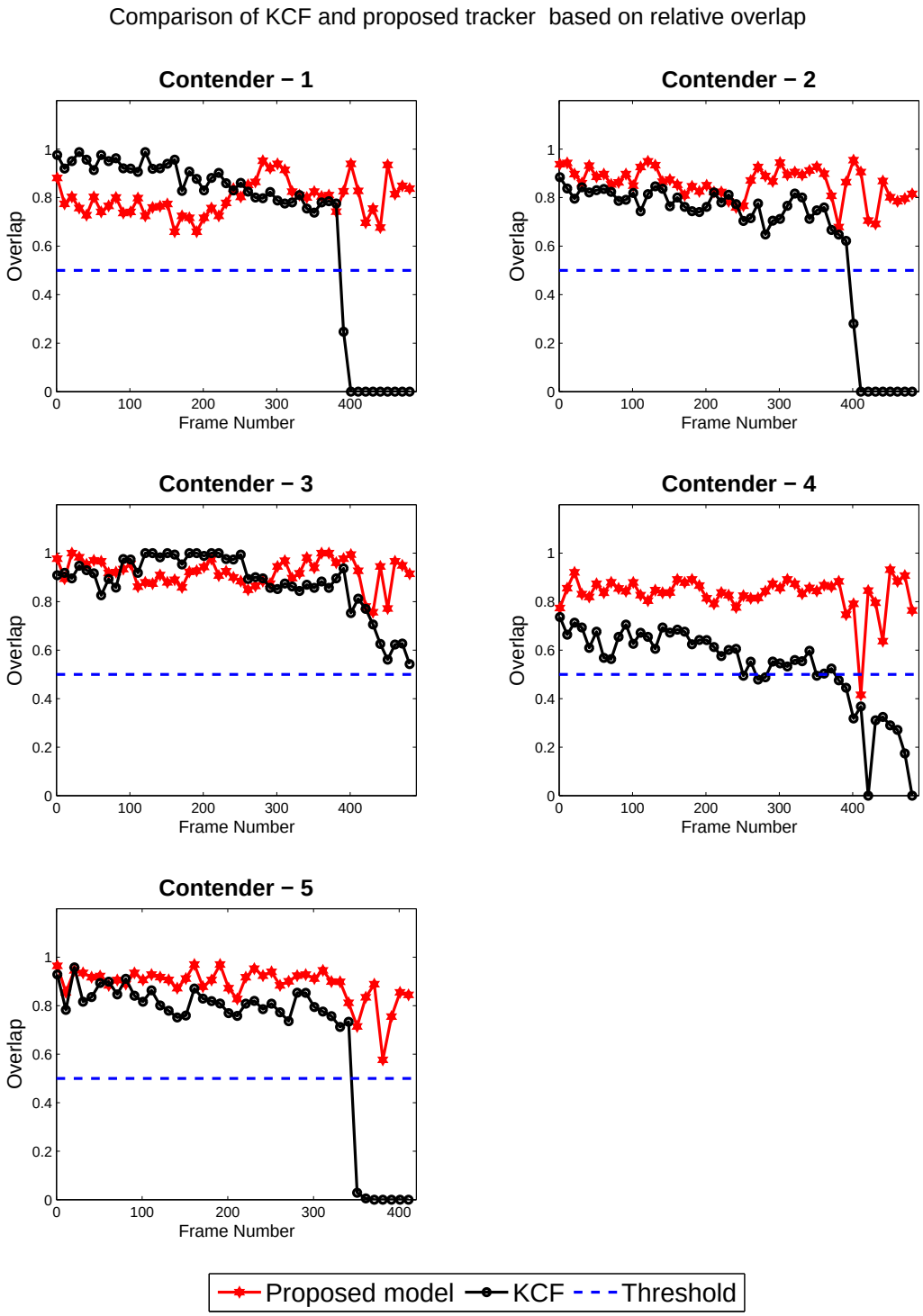

Figure 6.12: Comparison result between KCF and proposed model for Waikato–2. The KCF tracker failed to track four of the five objects when they are occluded by the tree.

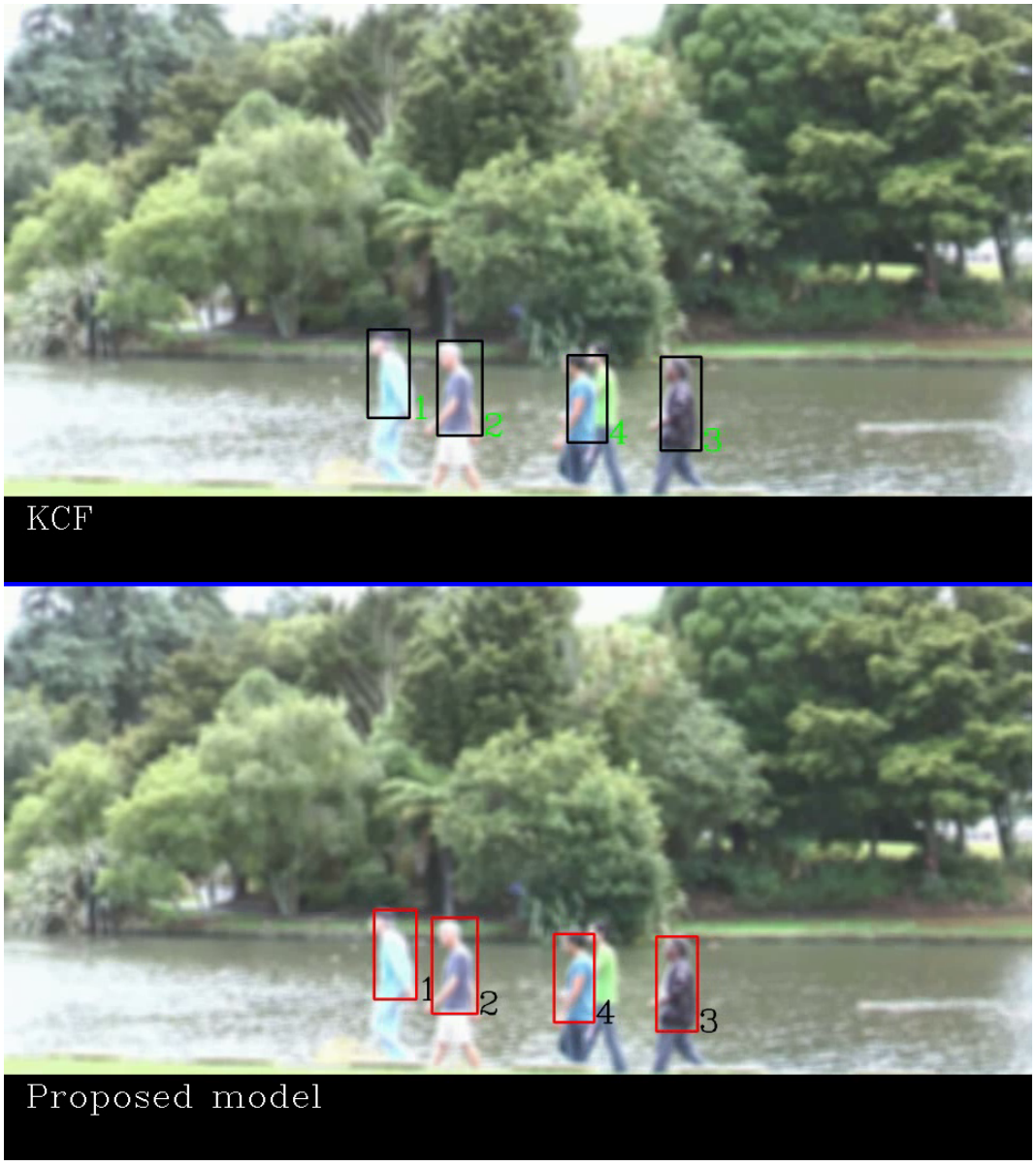

Figure 6.13: Visual tracking result for Waikato–2. The KCF tracker lost objects after they walk behind the tree. This is because the KCF algorithm did not encode the background motion information therefore the tree is considered as a tracked object and the classifier is trained with the wrong features which leads to the tracking drift. However, the proposed tracking model is shifting to group localisation and continues with tracking.

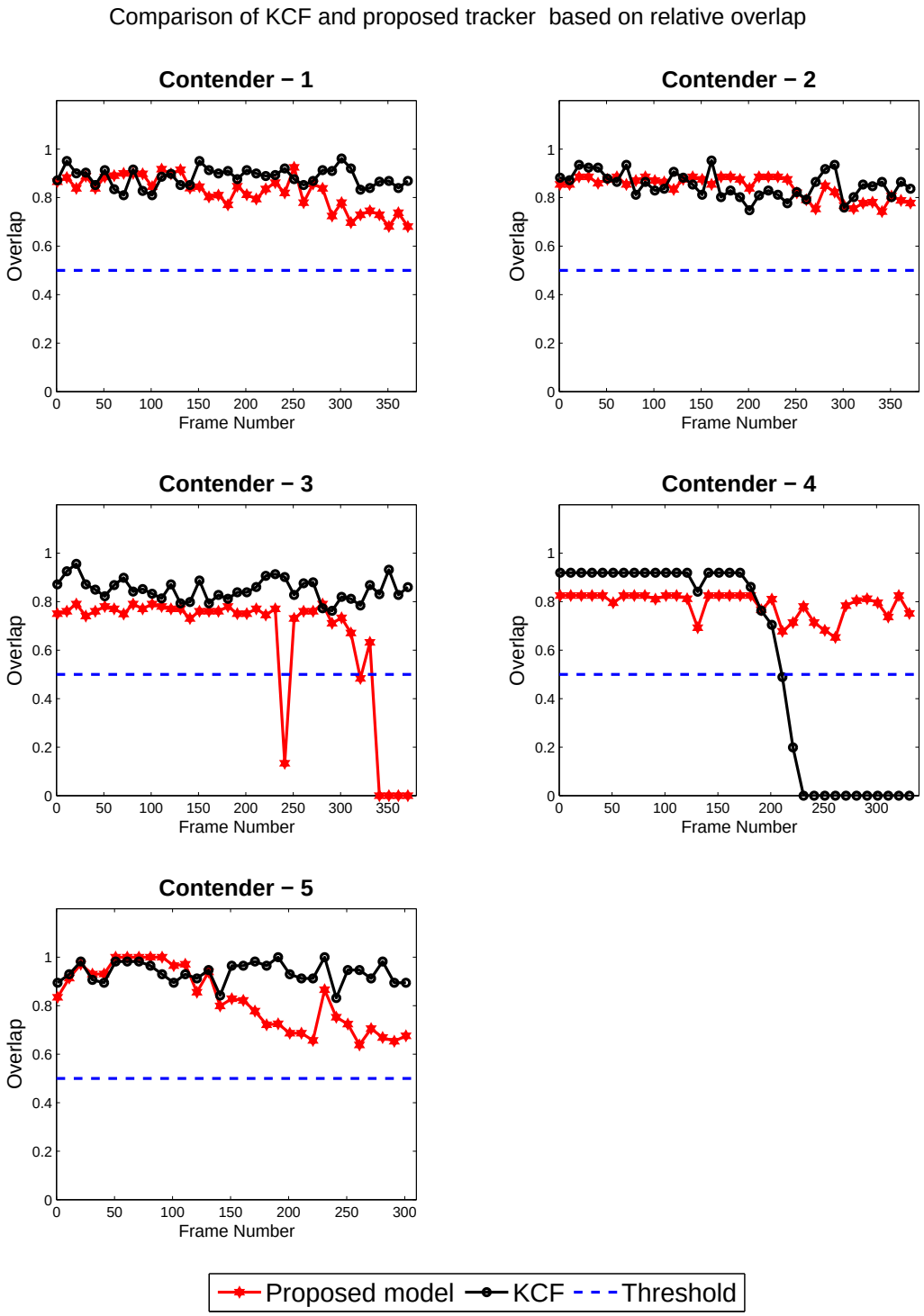

Figure 6.14: Comparison result between KCF and proposed model for Waikato–3. The proposed model fails to track objects 3 as both its motion and its appearance is similar to object 5.

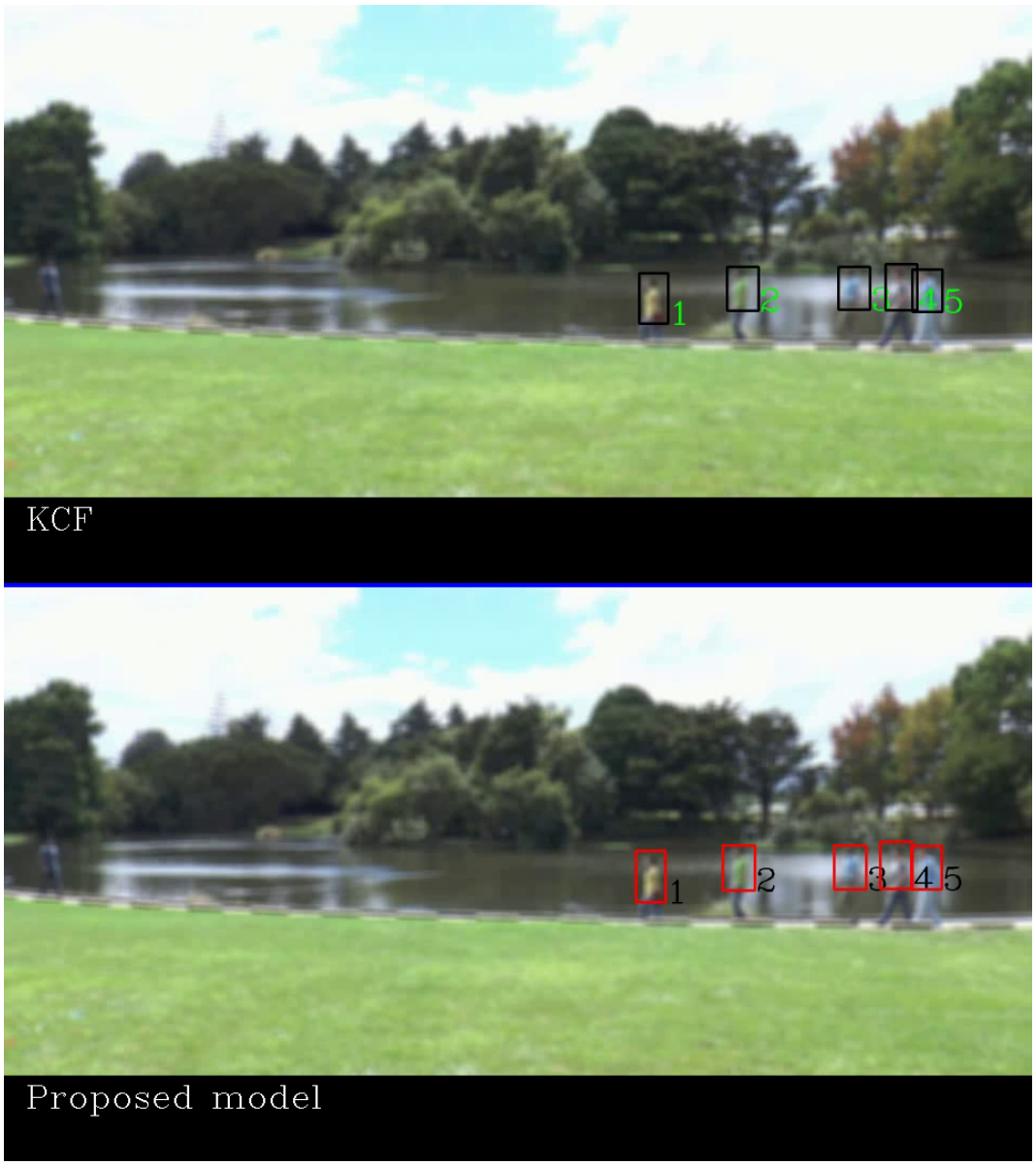

Figure 6.15: Visual tracking result for Waikato–3. The proposed model fails to track objects 3 as both its motion and appearance is similar to object 5.

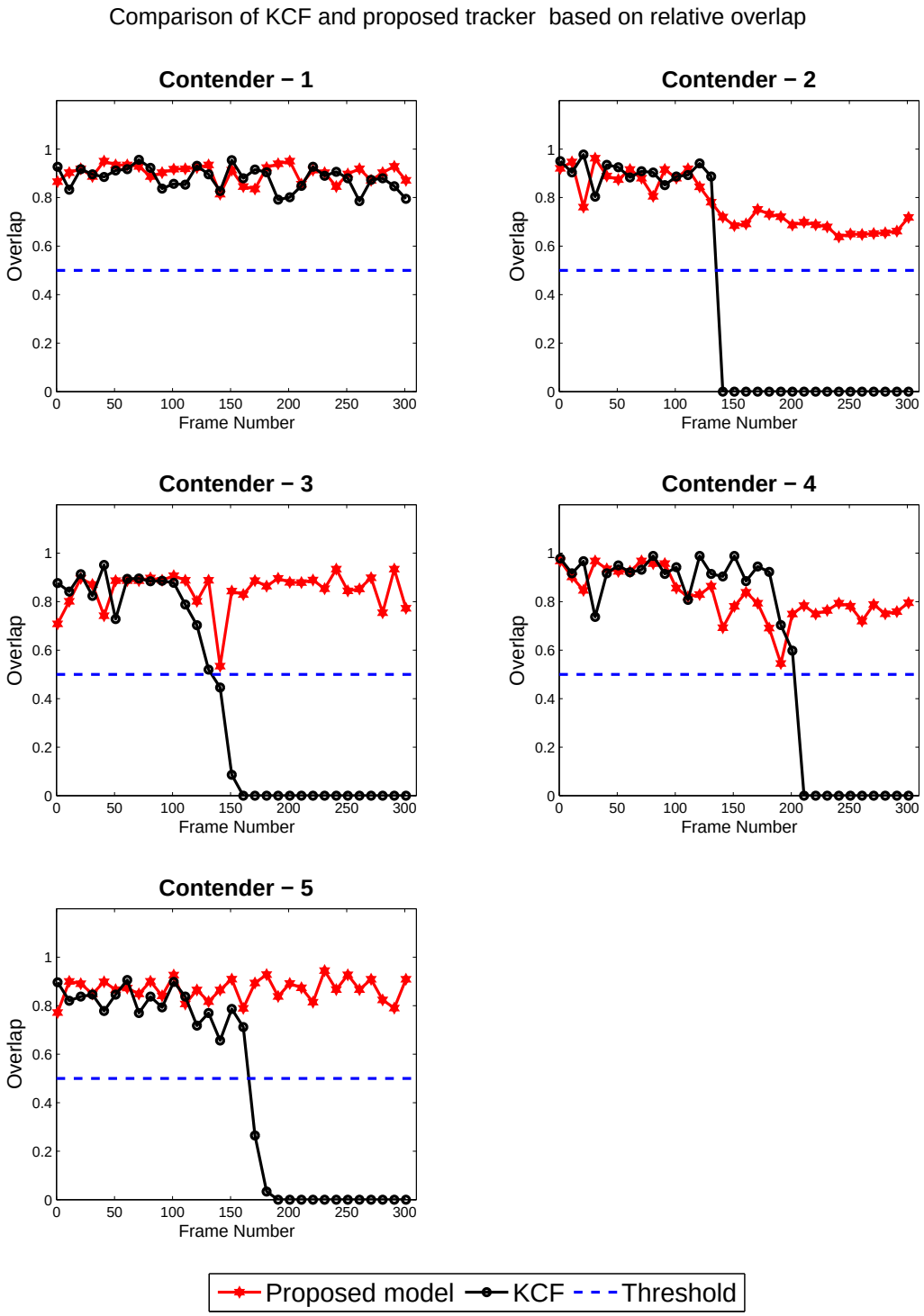

Figure 6.16: Comparison result between KCF and proposed model for Waikato–4. The KCF tracker failed to track four of the five objects when they are occluded by other objects.

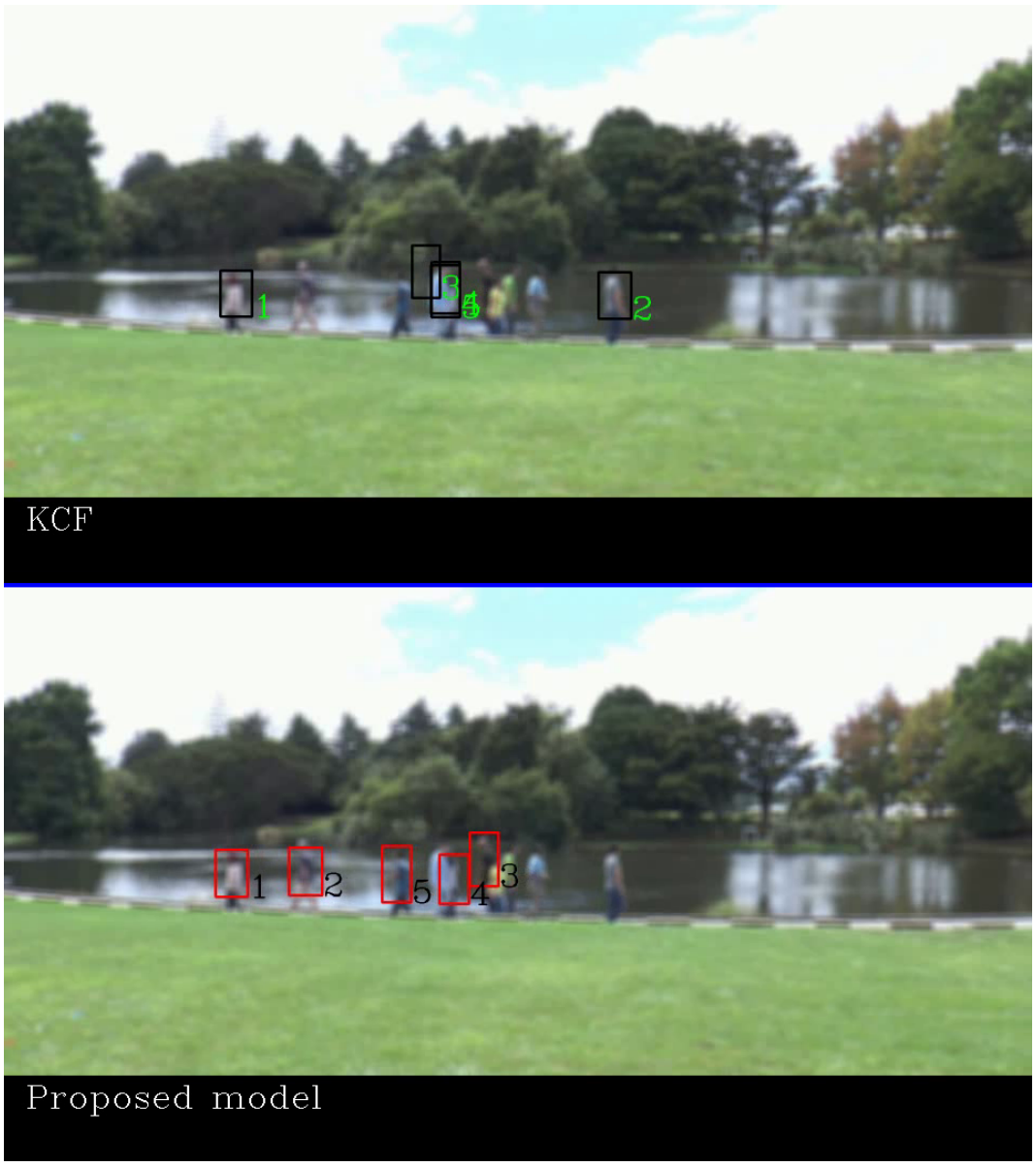

Figure 6.17: Visual tracking result for Waikato–4. Even without background clutter the KCF unable to tracked occluded objects. However, the proposed tracking model is shifting to group localisation and continues with tracking.

6.2 System evaluation

In Chapters 3 and 4 the turning segment and cap detection performance were analysed. In this section the overall performance of the proposed horse race system is evaluated with regards to two aspects: automatic contender initialisation and contender tracking. The difference of contender tracking evaluation in this section compared to last section is that here the final tracking result is based on both tracking and the data association strategy. Moreover, the tracking is consider successful if the centre of the tracking box, matches the ground truth only at the end of tracking.

To build the ground truth data ten turning shots from ten races are selected. The properties and sample images of these video are shown in Table 6.2 and Figure 6.18. Next the location of the jockeys' caps at the first and at the end of each turning shot are manually labelled. It should be pointed out that the videos which we are using in this evaluation is totally different from any videos that used before. These ten videos selected at beginning of this study to evaluate the performance of the proposed system. All of the selected videos are from Korea horse-racing site. The size of these videos are $800X600$ and are converted from analogue TV broadcast.

6.2.1 Initialisation of Contenders

In section 4.2.4 we evaluate the performance of cap detection, however, to initialise contenders, the detected cap should assigned into the same potential track three times over a five frame period. Therefore the initialisation process is directly related to object detection and data association algorithms. The accuracy of the contenders initialization is estimated using F1 score by

$$F1 = 2.\frac{\text{precision. recall}}{\text{precision} + \text{recal}} = \frac{2TP}{(2TP + FP + FN)}, \tag{6.2}$$

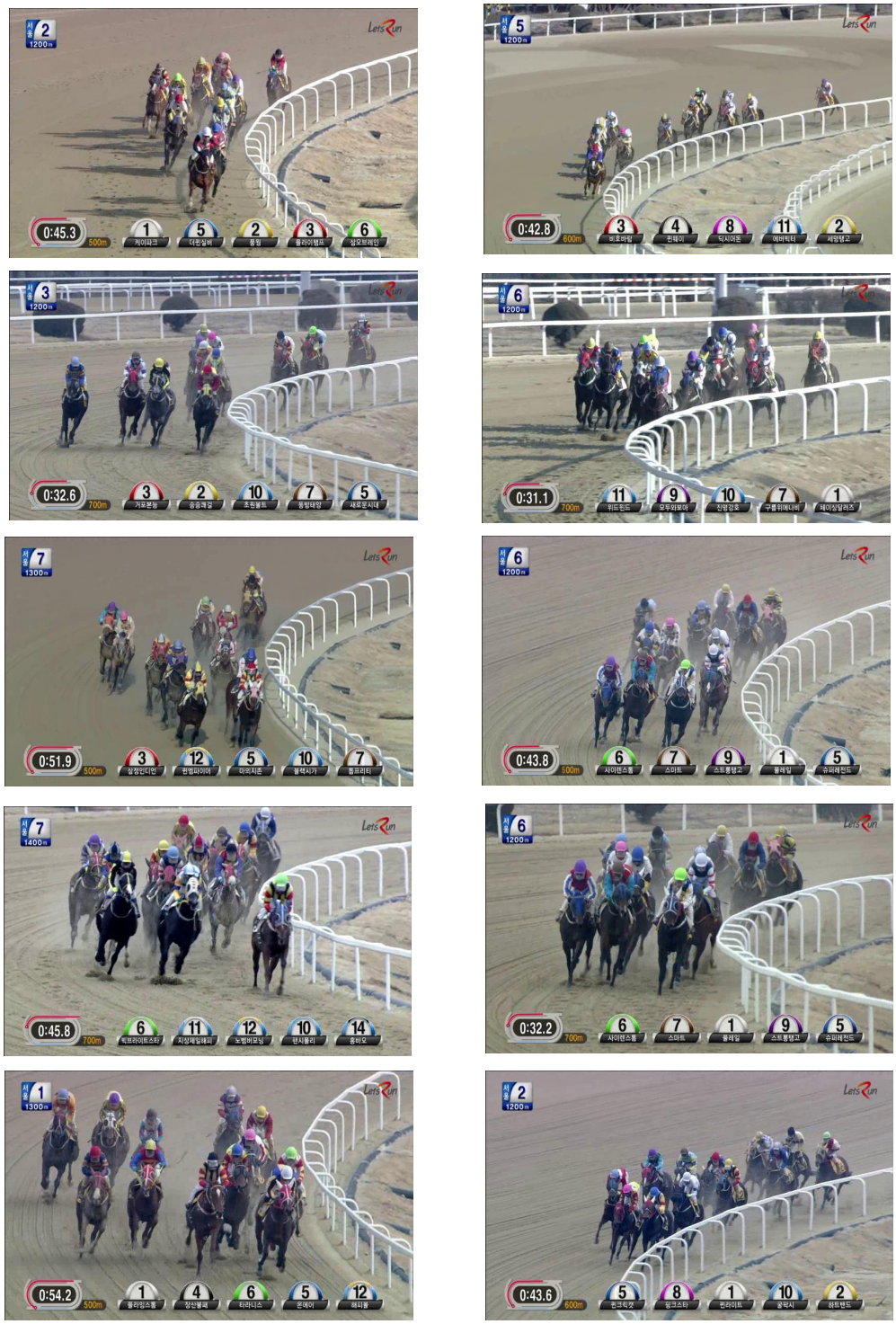

Figure 6.18: Sample images of final test videos

Table 6.2: The Properties of Final Test Videos

Video Name	Total Contenders	Duration of Turning Segment
v1	11	1 sec
v2	12	2 sec
v3	11	3 Sec
v4	12	2 Sec
v5	13	1 Sec
v6	12	1 Sec
v7	12	2 Sec
v8	12	1 Sec
v9	12	3 Sec
v10	12	1 Sec

Table 6.3: Evaluation Table for Cap Initialisation Module

Video Name	Total Contenders	TP	FP	FN	F1 Score
v1	11	10	1	1	.90
v2	12	12	2	0	.92
v3	11	10	2	1	.85
v4	12	12	0	0	1
v5	13	12	1	1	.96
v6	12	12	0	0	1
v7	12	12	2	0	.92
v8	12	10	2	2	.83
v9	12	12	2	0	.92
v10	12	12	1	0	0.96
Overall	117	113	13	4	0.93

The respond of detection is consider correct if the centre of detected bounding box lays inside the manually marked cap location (TP). It can be seen from Table 6.3, the overall F1 score for cap initialisation is 0.93, with lowest value of .83 in V8 and highest of .96 for V5. This result also proves the reliability of the contender detection module due to the performance similarity of cap initialisation and cap detection result in Chapter 4.

Table 6.4: Tracking Performance for Ten Selected Videos

Video Name	Total Contenders	Detected Cap	Hit	Miss	CTR
v1	11	10	10	0	1
v2	12	12	11	1	.91
v3	11	10	8	2	.8
v4	12	12	12	0	1
v5	13	12	12	0	1
v6	12	12	12	0	1
v7	12	12	11	1	.91
v8	12	10	9	1	.90
v9	12	12	12	0	1
v10	12	12	12	0	1

6.2.2 Contender Tracking

The performance of contender tracking for ten selected turning segment is tabulated in Table 6.4. The performance of a tracking algorithm is measured by calculating the ratio of successful tracked contenders to the total number of *detected contenders*. This measurement is called the correct tracking ratio (CTR). Here Hit indicates total number of correct tracked contenders and Miss is the number of unsuccessful tracking.

The result of Table 6.4 shows a promising result with average CTR of .94, where in 9 out of 10 cases the correct tracking ratio is above 0.90. The worst tracking performance belongs to V3 with CTR of .8. From analysis of V3, it is found that two failed-to-track contenders, number 5 and 8, are very close to each other when the tracks initiated. This probably caused imperfect sampling at the starting point and eventually led to tracking drift as shown in Figure 6.19.

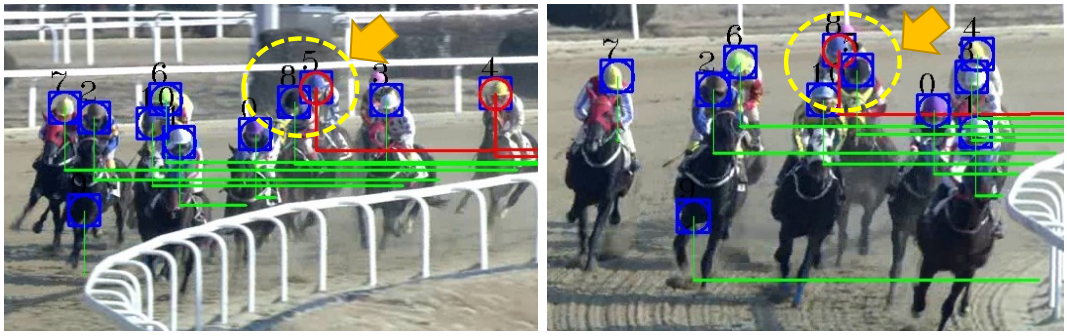

Figure 6.19: Contenders 5 and 6, inside yellow circle, are very close to each other at initial location which caused imperfect sampling and led to tracking drift.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, we proposed a framework for automated analysis of horse racing from TV broadcast videos. Within this framework we developed an automatic event detection algorithm which extracts the turning segments from the horse races. We implemented a contextual-aware analysis model to obtain the trajectory of contenders from the turning segments. The proposed system is developed based on deterministic reasoning driven by observing various horse races. We have faced many challenges both in designing and evaluating the proposed system.

Our first challenge was a paucity of literature in this topic. There are numerous successful sport analysis systems which have been reported in literature. These sport analysis systems are rules-based, and the rules have been obtained based on prior knowledge that is specific to the sport in question. The problem is that none of these rules can be applied to horse racing, due to the different structure and regulations of horse racing in comparison to studied sport. The second challenge was directly related to the nature of the horse racing. In horse racing contenders closely follow each other to gain a leading position and the camera view is continually changing. These two characteristics alone were the source of many problems such as the variations in jockey appearance, motion complex-

ity and the frequent occlusion of jockeys to name a few. These difficulties create a very challenging environment for detection and tracking of the contenders. However, despite all these challenges, we believe this project is a comprehensive framework that advances the field of research in video analysis.

By studying the rules and regulations of horse racing which were carried out in Chapter 2, we obtained a set of domain-specific rules. These rules helped greatly to crack the most challenging sports analysis systems that have been developed so far.

In short, the success of the proposed system is owed to three important properties of horse racing:

1. The camera typically follows the contenders; thus, motion of the background is much greater than contenders. By means of this characteristic, in Chapter 3 we developed a statistical motion analysis scheme to break the video into shots and then extract the turning segment within the shots.

2. Jockeys' caps have distinct colour with a rigid structure. In Chapter 4 we showed that the structure of the cap can be easily characterised by the distribution of local intensity gradients, which helped us to build a powerful contender detection model.

3. The contenders in the race follow each other's paths and move as a slowly changing group. This group dynamic often gives an important cue to approximate the location of obscured objects, especially when local information is poor or abrupt. In Chapter 5 we proposed a framework that combines this group dynamic with local object information to improve object tracking in challenging cluttered environments. The performance evaluation in Chapter 6 proved the robustness of the proposed systems, particularly in the presence of occlusion and background noise. Moreover, it is demonstrated in Section 6.1 that the proposed system can be generalized

to work on other domains.

This system addressed detection and tracking of contenders in the turning segment of the race. As the turning segment is the most difficult part of horse racing, it becomes clear that it could be done for the whole duration of the race, but this will require a great deal of work, especially in contender detection algorithms.

The structure of the jockey's cap is simple. This property allows us to get reasonable detection accuracy using traditional object detection algorithms. However, during the development of this project, several object detection algorithms based on convolutional neural networks (CNN) were proposed [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; Redmon et al., 2016]. CNN-based approaches are quite different from traditional detection algorithms. They deep neural architectures enable them to learn more complex features which show very promising result on the benchmark datasets. Thus, our future aim is to use deep convolutional neural networks to detect multiple parts of contenders, including the jockey's cap, the jockey's body, horse head and horse body.

However, the success of any CNN or deep learning architectures relies on the quality and the quantity of training data. For this project, we had very noisy, low-quality broadcast videos. Due to the high noise ratio and the sample size, the cap only spreads over a few pixels and its edges are often faded into the background. In addition, horse racing can be performed under different lighting conditions (e.g. spotlight, day, night, cloudy and rainy) and pitch characteristic (e.g. clay and grass). Therefore, collecting good quality video samples that contain all of the above variations will be very challenging, especially if we are not able to get them from the horse's mouth.

References

Allen, J. G., Xu, R. Y., Jin, J. S. (2004). Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pp. 3–7. Australian Computer Society, Inc.

Amit, Y., Felzenszwalb, P. (2014). Object detection. In *Katsushi I. (ed.), Computer Vision: A Reference Guide*, pp. 537–543. Springer.

Asghar, M. N., Hussain, F., Manton, R. (2014). Video indexing: a survey. *Framework*, *3*(01).

Atmosukarto, I., et al. (2013). Automatic recognition of offensive team formation in american football plays. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 991–998.

Avidan, S. (2004). Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, *26*(8), 1064–1072.

Bar-Shalom, Y. (1987). *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc.

Bar-Shalom, Y., Daum, F., Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems*, *29*(6), 82–100.

Bar-Shalom, Y., Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, *11*(5), 451–460.

Barris, S., Button, C. (2008). A review of vision-based motion analysis in sport. *Sports Medicine*, *38*(12), 1025–1043.

Berclaz, J., Fleuret, F., Turetken, E., Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, *33*(9), 1806–1819.

Bernardin, K., Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, *2008*, 1.

Bertini, M., Del Bimbo, A., Nunziati, W. (2005). Player identification in soccer videos. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 25–32. ACM.

Betke, M., Wu, Z. (2016). *Data Association for Multi-Object Visual Tracking*. Synthesis Lectures on Computer Vision. Morgan & Claypool.

Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, pp. 401–406.

Blake, A., Isard, M. (1997). The condensation algorithm-conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems*, pp. 361–367.

Boreczky, J. S., Rowe, L. A. (1996). Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, *5*(2), 122–128.

Boser, B. E., Guyon, I. M., Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152. ACM.

Bouchard, G., Triggs, B. (2004). The tradeoff between generative and discriminative classifiers. In *16th IASC International Symposium on Computational Statistics (COMPSTAT'04)*, pp. 721–728.

Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, *5*(1-10), 4.

Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. In *Intel Technology Journal*, pp. 214–219.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L. (2009). Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1515–1522. IEEE.

Bruno, E., Pellerin, D. (2002). Video shot detection based on linear prediction of motion. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, vol. 1, pp. 289–292. IEEE.

Cai, Y., de Freitas, N., Little, J. J. (2006). Robust visual tracking for multiple targets. In *European conference on computer vision*, pp. 107–118. Springer.

Casella, G., Robert, C. P. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, *83*(1), 81–94.

Chandel, H., Vatta, S. (2015). Occlusion detection and handling: a review. *International Journal of Computer Applications*, *120*(10).

Chung, Y.-C. (2010). *Automated video processing and scene understanding for intelligent video surveillance*. Ph.D. thesis, University of Missouri-Columbia.

Comaniciu, D., Meer, P. (1997). Robust analysis of feature spaces: color image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 750–755. IEEE.

Comaniciu, D., Meer, P. (1999). Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1197–1203. IEEE.

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

Cucchiara, R., , Grana, C., Piccardi, M., Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence*, *25*(10), 1337–1342.

Dailianas, A., et al. (1996). Comparison of automatic video segmentation algorithms. In *Photonics East'95*, pp. 2–16. International Society for Optics and Photonics.

Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893. IEEE.

Dang, B., Tran, A., Dinh, T., Dinh, T. (2010). A real time player tracking system for broadcast tennis video. In *Asian Conference on Intelligent Information and Database Systems*, pp. 105–113. Springer.

Deokar, M., Kabra, R. (2014). Video shot detection techniques brief overview. *International Journal of Engineering Research and General Science*, *2*(6), 817–820.

Dixit, M., Venkatesh, K. (2009). Combining edge and color features for tracking partially occluded humans. In *Asian Conference on Computer Vision*, pp. 140–149. Springer.

D'Orazio, T., Leo, M. (2010). A review of vision-based systems for soccer video analysis. *Pattern recognition*, *43*(8), 2911–2926.

Ekin, A., et al. (2003). Shot type classification by dominant color for sports video segmentation and summarization. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 3, pp. III–173. IEEE.

Fortmann, T., Bar-Shalom, Y., Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, *8*(3), 173–184.

Fortun, D., Bouthemy, P., Kervrann, C. (2015). Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, *134*, 1–21.

Freund, Y., Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pp. 23–37. Springer.

Gargi, U., et al. (2000). Performance characterization of video-shot-change detection methods. *IEEE transactions on circuits and systems for video technology*, *10*(1), 1–13.

Ge, W., Collins, R. T. (2008). Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC*, vol. 2, p. 5. Citeseer.

Girshick, R. (2015). Fast r–cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Gomez, G., López, P. H., Link, D., Eskofier, B. (2014). Tracking of ball and players in beach volleyball videos. *PloS one*, *9*(11), e111730.

Gong, Y., et al. (1995). Automatic parsing of tv soccer programs. In *Multimedia Computing and Systems, 1995., Proceedings of the International Conference on*, pp. 167–174. IEEE.

Gu, L., Ding, X., Hua, X.-S. (2004). Online play segmentation for broadcasted american football tv programs. In *Pacific-Rim Conference on Multimedia*, pp. 57–64. Springer.

Hamidreza Tofighi, S., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I. (2015). Joint probabilistic data association revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3047–3055.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., Stahel, W. A. (2011). *Robust statistics: the approach based on influence functions*, vol. 114. John Wiley & Sons.

Han, J., Farin, D., Lao, W., et al. (2005). Automatic tracking method for sports video analysis. In *Proc. Symposium on information theory in the Benelux, Brussels, Belgium*.

Hanjalic, A. (2002). Shot-boundary detection: unraveled and resolved? *IEEE transactions on circuits and systems for video technology*, *12*(2), 90–105.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M., Hicks, S. L., Torr, P. H. (2016). Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, *38*(10), 2096–2109.

Hedayati, M., Zaki, W. M. D. W., Hussain, A. (2010). Real-time background subtraction for video surveillance: From research to reality. In *Signal Processing*

and Its Applications (CSPA), 2010 6th International Colloquium on, pp. 1–6. IEEE.

Henriques, J. F., Caseiro, R., Martins, P., Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pp. 702–715. Springer.

Henriques, J. F., Caseiro, R., Martins, P., Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 583–596.

Higham, D., Kelley, J., Hudson, C., Goodwill, S. R. (2016). Finding the optimal background subtraction algorithm for eurohockey 2015 video. *Procedia Engineering*, *147*, 637–642.

Hoernig, M., Herrmann, M., Radig, B. (2015). Real-time segmentation methods for monocular soccer videos. *Pattern Recognition and Image Analysis*, *25*(2), 327–337.

Horn, B. K., Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, *17*(1-3), 185–203.

Hou, J. Y. Y. H. W., Li, J. (2011). Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*, *15*, 2201–2205.

Hu, Y., Han, B., Wang, G., Lin, X. (2007). Enhanced shot change detection using motion features for soccer video analysis. In *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 1555–1558. IEEE.

Huang, Y., Llach, J., Bhagavathy, S. (2007). Players and ball detection in soccer videos based on color segmentation and shape analysis. In *Multimedia Content Analysis and Mining*, pp. 416–425. Springer.

Huang, Y.-P., et al. (2009). An intelligent strategy for the automatic detection of highlights in tennis video recordings. *Expert Systems with Applications*, *36*(6), 9907–9918.

Islam, S. M., Bennamoun, M., Davies, R. (2008). Fast and fully automatic ear detection using cascaded adaboost. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, pp. 1–6. IEEE.

Jaward, M., Mihaylova, L., Canagarajah, N., Bull, D. (2006). Multiple object tracking using particle filters. In *2006 IEEE Aerospace Conference*, pp. 8–pp. IEEE.

Jiang, H., Fels, S., Little, J. J. (2007). A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE.

Jiang, Y.-C., Lai, K.-T., Hsieh, C.-H., Lai, M.-F. (2009). Player detection and tracking in broadcast tennis video. In *Pacific-Rim Symposium on Image and Video Technology*, pp. 759–770. Springer.

Jonker, R., Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, *38*(4), 325–340.

Jordan, A. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*, 841.

Kalal, Z., Mikolajczyk, K., Matas, J. (2010). Forward-backward error: Automatic detection of tracking failures. In *Pattern recognition (ICPR), 2010 20th international conference on*, pp. 2756–2759. IEEE.

Kalal, Z., Mikolajczyk, K., Matas, J. (2012). Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, *34*(7), 1409–1422.

Kamath, A., et al. (2014). Automatic video shot change classification using optic flow and robust pixel difference based method. *VLSI and Signal Processing*, *4*(6), 25–31.

Kapela, R., et al. (2015). Real-time event classification in field sport videos. *Signal Processing: Image Communication*, *35*, 35–45.

Kerr, D., Coleman, S., Scotney, B. (2008). Comparing cornerness measures for interest point detection. In *Machine Vision and Image Processing Conference, 2008. IMVIP'08. International*, pp. 105–110. IEEE.

Khan, Z., Balch, T., Dellaert, F. (2005). Multitarget tracking with split and merged measurements. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 605–610. IEEE.

Kim, E.-J., et al. (2005). A video summarization method for basketball game. In *Pacific-Rim Conference on Multimedia*, pp. 765–775. Springer.

Kompatsiaris, Y., Merialdo, B., Lian, S. (2012). *TV content analysis: Techniques and applications*. CRC Press.

Lehuger, A., Duffner, S., Garcia, C. (2007). A robust method for automatic player detection in sport videos. *Orange Labs*, *4*.

Li, B., et al. (2004). Bridging the semantic gap in sports video retrieval and summarization. *Journal of Visual Communication and Image Representation*, *15*(3), 393–424.

Li, S., Lihong, H. (2014). Research of background segmentation method in sports video. *Indonesian Journal of Electrical Engineering and Computer Science*, *12*(6), 4274–4282.

Li, Y., Wang, S., Tian, Q., Ding, X. (2015). Feature representation for statistical-learning-based object detection: A review. *Pattern Recognition*, *48*(11), 3542–3559.

Li, Y., Zhu, J. (2014). A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV Workshops (2)*, pp. 254–265.

Lien, C.-C., et al. (2007). Scene-based event detection for baseball videos. *Journal of Visual Communication and Image Representation*, *18*(1), 1–14.

Lienhart, R. W. (1998). Comparison of automatic shot boundary detection algorithms. In *Electronic Imaging'99*, pp. 290–301. International Society for Optics and Photonics.

Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H. (2009). Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, *30*(2), 103–113.

Lu, W.-L., Okuma, K., Little, J. J. (2009). Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, *27*(1), 189–205.

Lu, W.-L., Ting, J.-A., Little, J. J., Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, *35*(7), 1704–1716.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *7th international joint conference on Artificial intelligence*, vol. 2. Morgan Kaufmann.

Maćkowiak, S., Konieczny, J., Kurc, M., Maćkowiak, P. (2010). Football player detection in video broadcast. In *International Conference on Computer Vision and Graphics*, pp. 118–125. Springer.

Maggio, D. E., Cavallaro, D. A. (2011). *Video Tracking: Theory and Practice*. Wiley Publishing, 1st edn.

Mahmood, Z., Ali, T., Khattak, S., Hasan, L., Khan, S. U. (2015). Automatic player detection and identification for sports entertainment applications. *Pattern analysis and applications*, *18*(4), 971–982.

Mazinan, A. H., Latifi, A. A. (2012). Applying mean shift, motion information and kalman filtering approaches to object tracking. *ISA transactions*, *51*(3), 485–497.

McAnanama, J., Kirubarajan, T. (2012). A multiple hypothesis tracker with interacting feature extraction. *Signal Processing*, *92*(12), 2962–2974.

Mehtre, B. M., et al. (1995). Color matching for image retrieval. *Pattern Recognition Letters*, *16*(3), 325–331.

Milan, A., Schindler, K., Roth, S. (2013). Challenges of ground truth evaluation of multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 735–742.

Montoliu, R., et al. (2015). Team activity recognition in association football using a bag-of-words-based method. *Human movement science*, *41*, 165–178.

Morais, E., Goldenstein, S., Ferreira, A., Rocha, A. (2012). Automatic tracking of indoor soccer players using videos from multiple cameras. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 174–181. IEEE.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, *5*(1), 32–38.

Murty., K. G. (1968). An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, *16*(3), 682–687.

Nagasaka, A., Tanaka, Y. (1991). Automatic video indexing and full-video search for object appearances. In *Proc 2nd Working Conf Visual Database Systems*, pp. 119–133.

Needham, C. J., Boyle, R. D. (2001). Tracking multiple sports players through occlusion, congestion and scale. In *BMVC*, vol. 1, pp. 93–102.

Nepal, S., et al. (2001). Automatic detection of'goal'segments in basketball videos. In *Proceedings of the ninth ACM international conference on Multimedia*, pp. 261–269. ACM.

Nguyen, A. G., Hwang, J.-N. (2002). Scene context dependent key frame selection in streaming. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pp. 208–213. IEEE.

Nummiaro, K., Koller-Meier, E., Van Gool, L. (2003). An adaptive color-based particle filter. *Image and vision computing*, *21*(1), 99–110.

Oh, S., Russell, S., Sastry, S. (2004). Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, pp. 735–742. IEEE.

Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., Lowe, D. G. (2004). A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pp. 28–39. Springer.

Ong, E.-J., Bowden, R. (2004). A boosted classifier tree for hand shape detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 889–894. IEEE.

Oshima, N., Saitoh, T., Konishi, R. (2006). Real time mean shift tracking using optical flow distribution. In *2006 SICE-ICASE International Joint Conference*, pp. 4316–4320. IEEE.

Pan, H., et al. (2002). Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 4, pp. IV–3385. IEEE.

Pérez, P., Hue, C., Vermaak, J., Gangnet, M. (2002). Color-based probabilistic tracking. *Computer vision—ECCV 2002*, pp. 661–675.

Pettersen, S. A., Johansen, D., Johansen, H., Berg-Johansen, V., Gaddam, V. R., Mortensen, A., Langseth, R., Griwodz, C., Stensland, H. K., Halvorsen, P. (2014). Soccer video and player position dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pp. 18–23. ACM.

Piccardi, M. (2004). Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4, pp. 3099–3104. IEEE.

Priya, G. L., Domnic, S. (2014). Shot based keyframe extraction for ecological video indexing and retrieval. *Ecological Informatics*, *23*, 107–117.

Prokaj, J., Duchaineau, M., Medioni, G. (2011). Inferring tracklets for multi-object tracking. In *CVPR 2011 WORKSHOPS*, pp. 37–44. IEEE.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, *24*(6), 843–854.

Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99.

Reno, V., Mosca, N., Nitti, M., D'Orazio, T., Campagnoli, D., Prati, A., Stella, E. (2015). Tennis player segmentation for semantic behavior analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–8.

Rosten, E., Drummond, T. (2006). Machine learning for high-speed corner detection. In *European conference on computer vision*, pp. 430–443. Springer.

Sadlier, D. A., O'Connor, N. E. (2005). Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(10), 1225–1233.

Salehifar, H., Dehshibi, M. M., Bastanfard, A. (2011). A fast algorithm for detecting, labeling and tracking volleyball players in sport videos. In *2011 3rd international conference on signal Acquisition and processing (icasp 2011)*.

Sen-Ching, S. C., Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. In *Electronic Imaging 2004*, pp. 881–892. International Society for Optics and Photonics.

Seo, Y., Choi, S., Kim, H., Hong, K.-S. (1997). Where are the ball and players? soccer game analysis with color-based tracking and image mosaick. In *International Conference on Image Analysis and Processing*, pp. 196–203. Springer.

Shahraray, B. (1995). Scene change detection and content-based sampling of video sequences. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pp. 2–13. International Society for Optics and Photonics.

She, K., Bebis, G., Gu, H., Miller, R. (2004). Vehicle tracking using on-line fusion of color and shape features. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pp. 731–736. IEEE.

Shi, J., Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600. IEEE.

Shih, H.-C. (2017). A Survey on Content-Aware Video Analysis for Sports. *ArXiv e-prints*.

Sobral, A., Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding, 122*, 4–21.

Stauffer, C., Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE.

Swain, M. J., Ballard, D. H. (1991). Color indexing. *International journal of computer vision, 7*(1), 11–32.

Szeliski, R., Coughlan, J. (1997). Spline-based image registration. *International Journal of Computer Vision, 22*(3), 199–218.

Thomas, G. (2011). Sports tv applications of computer vision. In *Visual Analysis of Humans*, pp. 563–579. Springer.

Tran, Q., Vo, B., Dinh, T., Duong, D. (2011). Automatic player detection, tracking and mapping to field model for broadcast soccer videos. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, pp. 240–243. ACM.

Tuytelaars, T., Mikolajczyk, K., et al. (2008). Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision, 3*(3), 177–280.

Ulusoy, I., Bishop, C. M. (2006). Comparison of generative and discriminative techniques for object detection and classification. In *Toward Category-Level Object Recognition*, pp. 173–195. Springer.

Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511. IEEE.

Viola, P., Jones, M. J., Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, *63*(2), 153–161.

Wren, C. R., Azarbayejani, A., Darrell, T., Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, *19*(7), 780–785.

Wu, Y., Lim, J., Yang, M.-H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1834–1848.

Xu, C., Cheng, J., Zhang, Y., Zhang, Y., Lu, H., et al. (2009). Sports video analysis: Semantics extraction, editorial content creation and adaptation. *Journal of Multimedia*, *4*(2), 69–79.

Yang, C., Duraiswami, R., Davis, L. (2005). Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 212–219. IEEE.

Yang, L., Hanxuan Shao, Zheng, F., Wang, L., Song, Z. (2011). Recent advances and trends in visual tracking: A review. *Neurocomputing*, *74*(18), 3823–3831.

Ye, Q., et al. (2005). Exciting event detection in broadcast soccer video with mid-

level description and incremental learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 455–458. ACM.

Yilmaz, A., Javed, O., Shah, M. (2006). Object tracking: A survey. *Acm computing surveys (CSUR)*, *38*(4), 13.

Yoo, H.-W., et al. (2006). Gradual shot boundary detection using localized edge blocks. *Multimedia Tools and Applications*, *28*(3), 283–300.

Yu, Q., Medioni, G., Cohen, I. (2007a). Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE.

Yu, Q., Medioni, G., Cohen, I. (2007b). Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE.

Zabih, R., et al. (1995). Feature-based algorithms for detecting and classifying scene breaks. Tech. rep., Cornell University.

Zhang, D., Chang, S.-F. (2002). Event detection in baseball video using superimposed caption recognition. In *Proceedings of the tenth ACM international conference on Multimedia*, pp. 315–318. ACM.

Zhang, H., et al. (1993). Automatic partitioning of full-motion video. *Multimedia systems*, *1*(1), 10–28.

Zhang, J., Qiu, J., Wang, X., Wu, L. (2013). Representation of the player action in sport videos. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pp. 1–4. IEEE.

Zhao, Y., Lu, Z. (2016). Research on background segmentation method for sports video. *International Journal of Simulation – Systems, Science and Technology*, *17*(2), 11.1–11.5.

Zhong, D., Chang, S.-F. (2004). Real-time view recognition and event detection for sports video. *Journal of Visual Communication and Image Representation*, *15*(3), 330–347.

Zhu, G., Xu, C., Huang, Q., Gao, W. (2006). Automatic multi-player detection and tracking in broadcast sports video using support vector machine and particle filter. In *2006 IEEE International Conference on Multimedia and Expo*, pp. 1629–1632. IEEE.