2017

SISTEMA DE VISIÓN POR COMPUTADOR PARA DETECCIÓN Y SEGUIMIENTO DE BALÓN DE FUTBOL EN CASOS COMPLEJOS UTILIZANDO CÁMARAS ESTÁTICAS

CÁRDENAS NAHUEL, DANIEL GERMÁN

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

Sistema de visión por computador para detección y seguimiento de balón de futbol en casos complejos utilizando cámaras estáticas

Tesis de Grado presentada por

Daniel Germán Cárdenas Náhuel

como requisito parcial para optar al título de

Ingeniero Civil Electrónico

y al grado de

Magíster en Ciencias de la Ingeniería Electrónica

Profesor Guía
Dr. Marcos Zuñiga Barraza

Valparaíso, 2016.

TÍTULO DE LA TESIS:

SISTEMA DE VISIÓN POR COMPUTADOR PARA DETECCIÓN Y SEGUIMIENTO DE BALON DE FÚTBOL EN CASOS COMPLEJOS UTILIZANDO CÁMARAS ESTÁTICAS

AUTOR:

Daniel Germán Cárdenas Náhuel

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el título de Ingeniero Civil Electrónico y el grado de Magíster en Ingeniería Electrónica de la Universidad Técnica Federico Santa María.

Dr. Marcos Zuñiga Barraza _____

Valparaíso, Agosto de 2016.

AGRADECIMIENTOS

A mis padres y hermana,
por su incondicional apoyo.

También a mi familia
y amigos.

RESUMEN

Este trabajo forma parte de un proyecto el cual busca implementar un sistema de video-análisis en partidos competitivos, con el propósito de entregar información estadística a cuerpos técnicos para dar soporte a la toma de decisiones. Estos sistemas utilizan cámaras estáticas alrededor del estadio para extraer información relacionada a el desempeño de los jugadores y sus interacciones.

En particular, esta tesis se centra en el caso del balón de fútbol, el cual es un elemento crucial para el desarrollo del juego, ya que éste permite obtener información relevante para el entendimiento de los distintos acontecimientos del partido. El seguimiento automatico de balón presenta varios problemas principalmente debido a su pequeño tamaño en el video, el cual dificulta la extracción de sus caracteristicas y lo hace mas sensible a distorcionarse cuando presenta una alta velocidad. Además, existen problemas cuando el balón interactua con los jugadores, porque éste presenta ocultaciones totales o parciales, o su dinámica se vuelve impredecible.

En este trabajo incorpora soluciones del estado del arte, agregando a estos una representación elíptica del balón, la cual hace mas rapido y certero su reconocimiento. Por otro lado, también incorpora un robusto sistema de seguimiento de balón para casos de ocultación.

Esta investigación se enmarca dentro del proyecto FONDECYT 11121383: *"Methodology and applications for incremental Behaviour Learning in Video guided by information Reliability"*.

Palabras Clave:

visión por computador, extracción de características, detección de objetos, seguimiento de objetos.

ABSTRACT

This work is part of a project which implements a video-analysis system, with the purpose of bringing statistical information to support technical staff in decision making for competitive matches in collective sports. These systems use static cameras installed around the stadium and utilise their captures to extract relevant information relative to the performance of players and their interactions.

In particular, this thesis focuses on the soccer ball, a crucial element in the game, because it presents a relevant information for the understanding of the dynamics of gameplay. Automatic ball tracking presents several challenges, mainly because of the small size of the ball in the video, which difficults the extraction of its features and makes it more sensitive to distortion when the ball is in high speed. Moreover, there are problems when the ball interacts with players, because it presents partial or total occlusion, or its dynamics become unpredictable.

This work utilises state-of-the-art algorithms, incorporating an elliptical representation into the ball model, which makes its detection more reliable and fast. Besides, a robust tracking system is incorporated to deal with occlusion situations.

This research has been supported, in part, by Fondecyt Project 11121383: *"Methodology and applications for incremental Behaviour Learning in Video guided by information Reliability"*.

Keywords:

computer vision, feature extraction, object detection, object tracking.

CONTENIDO

List of Figures

Chapter 1

INTRODUCTION

Nowadays, the high competitiveness in collective sports has lead to an increment in the complexity of team analysis to improve the experience of sports broadcasting and support technical staff in decision making. In particular, the technical staff requires accurate and reliable information to determine right decisions, such as number of passes, interceptions and ball recoveries, or time in possessions of the ball [2].

Generally, the data gathering process is performed manually or with semi-automatic procedures. However, these methods are often expensive and time consuming. For this reason, the development and investigation of technological systems which can extract information about the game have been increasing more and more.

In soccer, one of the most competitive sports around the world, the information of players and ball are relevant to take good decisions. While the tracking of players is related to individual performance analysis, the ball tracking provides relevant information on team strategies, because the ball is the most relevant element in the game and any player decision is relative to its position and state (possession or free) [3].

This thesis is part of the project **FONDECYT No. 11121383** *Methodology and Applications for Incremental Behaviour Learning in Video guided by Information Reliability*, which one of its goals is the analysis of team behaviour using automatic or semi-automatic procedures in order to analyse the individual and collective performance of a team.

1.1 Context

Currently, sports have become a relevant key for the humanity, taking an important role in different areas such as social, economic, physical and physiological. Throughout our history, sports have been important in the life of people, independent of their culture. An important aspect of sports is the revenue, rising $71.3 billion USD equivalent to $48 trillion CLP in 2015 [1].

There is no doubt that competitive sports have been incorporating technology. For example, you might be familiar when sport broadcasters analyse the team performance after game, showing a high detail summary which probably was produced with technology, or, in some repetition shown each player position and ball direction, or maybe, in tennis when a virtual replay is generated showing exactly where the ball hits the ground and even predicts its future trajectory (to determine whether it would have hit the wicket, for example). Besides, technical staff often uses technology to observe and manage relevant information as they wish, in order to support their decisions.

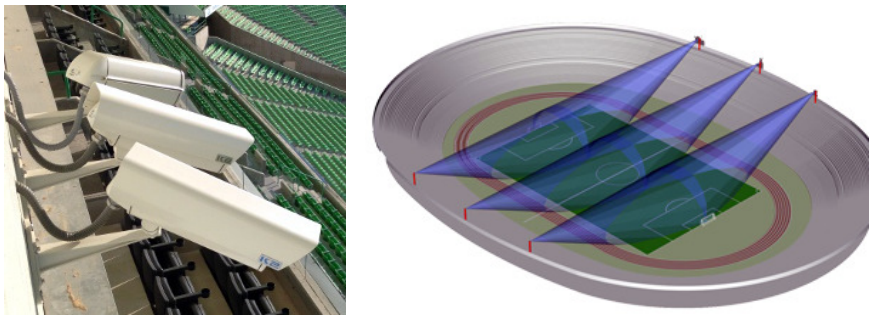
There are several methodologies to acquire player and ball information, but mainly, the most common are GPS and visual systems. GPS (global position system) is a satellite-based navigation system, which has to be incorporated into the player equipment to obtain their position. Unfortunately, this sensor presents several problems (see chapter 2).

In the other hand, visual systems are more viable to use, because they do not incorporate invasive technology. Several video analysis tools use broadcast videos to extract the information relative to the game. However, broadcast videos can only submit information relative to a specific part of the field, losing relevant information related to other sections of the field, as for example, when a player loses his marks in a key position far to the focus of attention.

For this reason, actually, the most competitive teams have been incorporating cameras around their stadium (as depicted in Fig. 1.1) in order to cover all the field and, this way, gather all possible information relative to the match.

To extract the information from cameras, current systems tend to use operators which examine the videos and then manually note each aspect of the game. The problem with this process is that requires a high cost in human and time resources. In consequence, with the purpose of reducing this cost,

[1] Global sports market - total revenue from 2005 to 2017
https://www.statista.com/statistics/370560/worldwide-sports-market-revenue/



(a) Match analysis: 3 cameras covering all the field.

(b) ISSIA-CNR: Panoramic view using 6 cameras.

Figura 1.1. Examples of video systems.

the data gathering process can be supported with machines, which compute algorithms that help in the player and ball annotations. However, these computational systems use to present several problems (depicted in Section 1.3), especially in detecting complex situations of the match, and also, comparing with real-time applications, they present a high delay.

Figura 1.2. Optasports: Operator gathering information from videos.

Regarding FONDECYT project aforementioned, the ball tracking approach is incorporated to the framework VAT (Video Analysis Tool), a platform which the main goal is to obtain a general behaviour analysis approach capable of

extract and learn valuable information from noisy video scenes for real-time applications. This way, the approach will be validated utilising videos from three different types of applications, where one of them is collective sports. The learning process of these behaviours can be guided to obtain meaningful information by estimating the reliability of the attribute information associated with the object models and processes. The estimation of reliability measures allows to properly model the uncertainty of data due to noisy videos other difficulties (e.g. illumination changes, object occlusion, high volume of data).

In this tool, a ball model is incorporated to a multi-hypothesis tracker based in [4]. This multi-model approach will incorporate different object representations dynamically calculated according to the context of the scene. This way, for instance, if two objects are about to occlude each other, for example, in our case a ball and a player, a finer model can be calculated to better understand the separation between both objects.

1.2 Aims and objectives

The main goal of this thesis is to **develop and implement a novel methodology for detecting and tracking the soccer ball, using static cameras installed around the stadium**.

The main goal subsequently defines several specific goals such as:

- To implement and evaluate the state-of-the-art of solutions related to ball tracking.
- To integrate the most performing state-of-the-art solutions in the ball approach.
- To test the performance of the integarted approach in real soccer videos, using qualitative and quantitative metrics to evaluate the system capabilities.
- To integrate this methodology in the framework VAT, inside the multi-modelling tracking system.

1.3 Problems

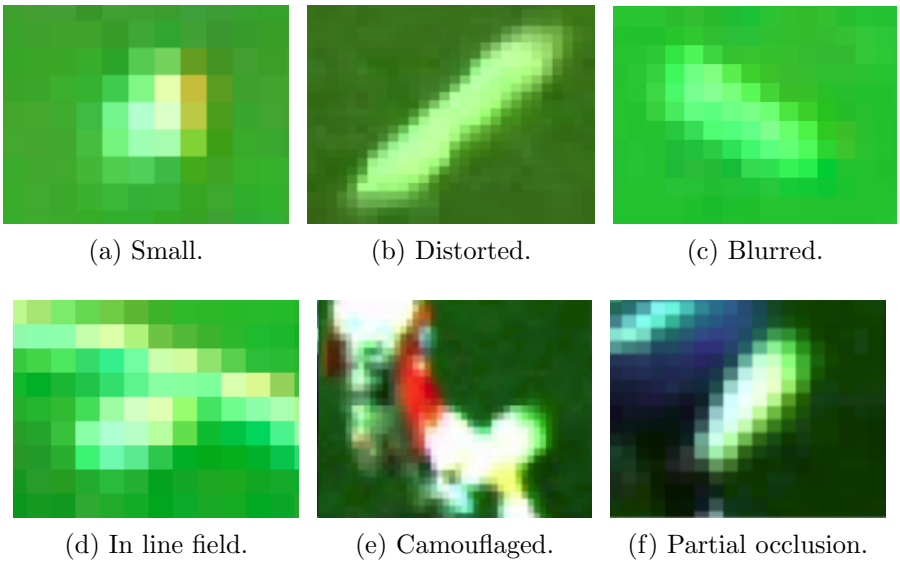
In particular, detecting and tracking the ball can be a difficult task for several reasons:

- The ball features become unreliable when the distance to the camera is high, as the ball is detected in a low number of pixels (Fig. 1.3 (a)).
- In high speed, the ball presents significant distortion that compromises the stability of the shape and size features, and presents motion-blurred shape and appearance (Fig. 1.3 (b,c)).
- The ball merges with player equipment, when they use similar colours, or with field lines and socks, when the ball is white (Fig. 1.3 (d,e)).
- The ball is frequently occluded by players (Fig. 1.3 (f)).
- There are many false alarms in the scene, such as small marks of the field or partial regions of players, like socks and shoes with similar colour.
- The ball suddenly changes its trajectory when interacting with players, complicating the prediction of the future position using dynamic models.
- The illumination changes modify the appearance of the ball.

In addition, multiple fixed cameras allow to locate the ball inside the field in an easier and more robust way, improving the resolution of dynamic occlusion situations. The effectiveness of a multi-camera system also relies on a robust single-camera tracking process.

For these reasons, existing methods for direct ball recognition present problems when trying to apply them in this context. For example, several works base their detection in the circularity of the ball [5–8], not considering its frequent distortion, or propose learning methods [9–11] not considering cases of small size when the ball patterns are not recognisable.

Also, another problem is the computational cost of the solution. This thesis proposal has to be a robust solution, but also has to require low computation cost in order to reach or get close to real-time applications.



(a) Small. (b) Distorted. (c) Blurred.

(d) In line field. (e) Camouflaged. (f) Partial occlusion.

Figura 1.3. Complex situations on ball detection.

1.4 Thesis roadmap

In **Chapter 2**, the state-of-the-art is presented. First, we describe a brief overview of some commercial match analysis tools. Then, the state-of-the-art of ball tracking is presented, in relation to the key ideas of each studied work. In **Chapter 3**, the main ideas behind our approach are described, presenting the different stages of the whole system. In **Chapter 4**, different state-of-the-art criteria for classifying the ball are described and evaluated. Then, an elliptical representation of the ball is presented and analysed to extract different characteristics of the ball. In **Chapter 5**, a well known ball tracking approach [1] which our thesis is based, is described. Then, our improvements to this approach are presented. In **Chapter 6**, a novel ball occlusion management system is presented. In **Chapter 7**, the experimental results of our algorithm are presented, contrasting our approach with an state-of-the-art ball tracking system. Finally, in **Chapter 8**, the main conclusions and future work are presented.

Chapter 2

STATE OF THE ART

Over the last decades, the attraction in professional soccer and the management for forming successful teams has increased the motivation of researchers and engineers for proposing new technological tools able to properly assist the technical staff on decision making and training methodologies [12]. One of these utilised technologies is video analysis, but the analysis has been mostly conducted on a limited time of the video data where the automatic analysis remains quite brief. In order to get an overview of soccer analysis tools, we present a survey of the commercial systems and the research works that has been conducted in the following sections.

2.1 Commercial Soccer Match Analysis Tools

The utilisation of video has started to be employed in the 1980s as a means of studying motion analysis in professional soccer. These match analysis studies simply involved the observation of performance using a video cassette recorder and a television monitor to control playback and visualise footage.

Over the past 20 or so years, various computerised match analysis systems have appeared on the market, each offering different and often improved levels of usability and data. The majority of these systems have been developed as various notational research projects in universities, and it is only in recent years that their usage has been truly extended to clubs or national federations [12]. Nowadays, the development of technology for sports analysis has widespread into different solutions, which can be classified into invasive sensors or computer vision technologies [4].

2.1.1 Invasive sensors

These type of sensors require that the ball and players utilise a device which sends signals to a receiver. One of the most popular invasive sensors are GPS which utilise satellites to get the position of each element. Nevertheless, due to the distance of satellites to the devices, they cannot determine a position with less than three meters of precision [13]. To solve this problem, these systems consider the installation of their own signal receivers in order to get an accurate precision of their positions. The advantage of these systems is that they provide a low computational cost, but it is not preferred by most of the football federations since it is regarded as interfering within the game [1], which is not allowed in official matches. Additionally, the use of active sensors makes the players feel uncomfortable during training.

2.1.2 Computer vision technologies

The video is the richest source of information so far in the soccer game analysis. Video analysis tools require (to a greater or lower extent) the manual interaction of operators and the implementation of machine vision methodologies. Arriaza and Zuñiga show a survey of the most relevant match analysis system in soccer [14]. Currently, some of the most popular computer vision systems are STATS SportVU™ and TRACAB™

STATS SportVU™ is one of the leading player tracking systems and provides a physical, technical and tactical coaching tool using multiple HD cameras installed within a stadium. It is mostly intended for coaches and management staff to analyse physical, technical and tactical performance information through an interactive and engaging coaching tool. Their standard solution utilise three HD cameras in a single location for covering all the field. This solution tracks the 2D position of players and the ball. Besides, they offer an alternative solution which incorporates a six HD camera setup in a single location adding another 3 camera setup in another location for 3D tracking. Recently, they acquired the technology of two popular tracking systems (Prozone™ [15] and Amisco™ [16]). Their data delivery time depends on each team's requirement, reaching real-time performance in basketball. Nevertheless, no

[1] Law number 4 of the FIFA - the players' equipment,
`www.fifa.com/mm/document/afdeveloping/refereeing/law\_4\_the\_players\_equipment\_en\_47415.pdf`

scientific reports have been found documenting the utilisation of SportVU™ system in soccer.

textbfTRACAB™ is presented as a system able to identify the position of every moving object in the field, in real-time. The acquisition system is composed by two sets of stereo HD cameras placed on the top of the stadium in order to capture all the pitch. This system can extract the 3D position of all players and the ball under different environmental conditions. However, in the work of Duque [17], an improvement of ball tracking is presented, showing that this system is less reliable when it is applied to track the soccer ball.

2.2 Computer vision methodologies

Several studies have been conducted so far in the vision-based analysis of soccer games [9, 18, 19]. Most of them have been tested in ideal conditions. However, when they were tested in difficult situations, even the most relevant state-of-the-art algorithms, present complications when the ball is occluded or merged with other agents in the field.

2.2.1 Ball appearance approaches

Many ball approaches are based on circular shape models [5–8, 20, 21]. The most utilised model is the circular Hough transform (CHT). In [8], a modified version of CHT is applied in order to detect the possible positions of the ball. Then, a neural classifier is applied to confirm if the ball has been properly detected. CHT performs matching between a circular vector space and the x-y gradients of the image, making difficult the detection of the ball when is highly distorted. Moreover, these works mainly focus on broadcast videos, where the ball presents a good resolution and a circular shape. These conditions are not applicable in our context when the camera takes a far plane of the field, or the ball presents too fast movement because its shape is distorted.

Other proposals also use machine learning tools to detect the ball [9]. For instance, in [10, 11], variants of Viola-Jones algorithm are utilised in robotics applications, like RoboCup, to get a colourless representation of the ball. These approaches present some limitations to perform an appropriate analysis in more real situation because they are implemented for recognising soccer ball patterns with textures which are only visible in zoomed images, and where

there are not large variations in the appearance of the ball.

2.2.2 Ball detection in occlusion situations

Several ball tracking methods deal with occlusions by applying statistical filters [5,18,19,22–25]. For example in [22] a ball tracking approach is proposed based in Kalman filters, incorporating dynamic conditions to control the velocity state vector. In [23], a distributed state estimation based in particle filters is proposed for 3D ball tracking. This approach can operate stand-alone, as well as in network mode by modelling the dynamics with ballistic motion. Few of these works are deeply focused on instant ball detection in occlusion situations, and those that deal with these situations, assume that the ball shape is circular. In [6] the ball location is estimated analysing the frames between the disappearance and the re-appearance of the ball. Different ball paths are generated, and a circular filter is utilised to determine the best ball trajectory in the occlusion sequence. The authors conclude that the method tends to fail after a player kicking or heading the ball, which is attributable to the failure in the detection of the ball-like regions. In [26] the authors propose a soccer ball detection approach which is applicable to occlusion cases in low resolution images. In this approach, a circle detection method is applied, generating a set of candidates, which are evaluated with Freeman chain code [27] to study their contour. Moreover, the ball could only be detected when the ball did not appear as part of the region corresponding to another object (e.g. partial occlusion with a player).

2.2.3 Ball detection system approaches

In order to detect the ball with static cameras around the stadium, several ball approaches use two separate processing phases [1,23,28,29]: In the first stage, a set of ball candidates are selected using different ball features approach. Then, in the second stage, the candidates are validated in order to recognise which of them really contains the ball using a ball dynamics approach.

In [1], different candidate trajectories are studied, using the coherence and dynamics of the ball size, colour, and shape. They select the trajectory that better represents the ball using a classification score for each candidate. Then, they connect the fragmented trajectories, using a post-processing interpolation of the ball locations using the previous and posterior trajectories to an occlu-

sion situation.

2.2.4 Spatial-temporal context approaches

Other works utilise the correlated information in space and/or in time neighbourhoods to detect the ball and players [30–33]. These models require a target to be interacting with other moving objects (players). The appearance of these contextual objects should be sufficiently distinct so that they themselves can be reliably detected in novel video frames. To locate the target, each near player is employed as a predictor for the ball centre.

In [33] player positions are utilised in order to decide ball possession. This research is based in that the ball trajectory is intimately linked to that of the players who pass it to each other or steal it from one another, allowing disambiguation and improved performance.

Unfortunately, the utilisation of a player model is beyond the scope of this thesis. However, future works will focus on combining the player and ball models, in order to improve the accuracy of the results.

2.3 Discussion

A new ball detection and tracking approach is presented, able to better resolve occlusion situations in a single camera view, which is also robust on fast movement situations.

The key idea of this work is first, detecting the ball in the scene and then use its dynamics to properly track the ball in any situation. We integrate different conventional methods based on ball trajectory analysis using a new ball model. Then, when a ball occlusion situation occurs, a new robust tracking algorithm is activated.

In order to deal with the distortion of the ball in case of fast movement, our work utilises an elliptical model to describe the ball. We utilise the minimum radius of the ellipse to represent the real size of the ball, and also utilise the distortion direction in order analyse its coherence with ball dynamics.

While several existing methods based in trajectory analysis [1, 34, 35] have to restart the detection of the ball in the whole scene after the ball is occluded, our proposal focuses more on these complex cases, implementing mechanisms which cope with both partial and total occlusion situations. For this purpose,

we integrate the previous information of the ball (dynamics and appearance) with the current evidence, in order to manage several plausible ball positioning hypotheses. Therefore, the previous dynamics of the ball let us analyse the distortion case even in occlusion situations. Also, we add some constraints to our dynamics model in order to manage unpredictable change of the ball movement, validating the dynamics model only when we have certainty that the ball is moving according to the previous information.

Chapter 3

OVERVIEW OF OUR APPROACH

In this chapter we elaborate the main ideas behind our approach, schematised in Fig. 3.1, which can be summarised in three main stages. The first stage focuses on identifying different objects of interest in the scene, generating a set of ball candidates. The second stage analyses the trajectory of each ball candidate in order to identify which of them is the real ball. Finally, the third stage is performed only for occlusion situations using a new system methodology.

3.1 First stage - Object extraction

In the first stage, a set of objects is extracted from the video sequence in relation with their similarity to the ball. In our approach, the ball colour is assumed as white, because is the most frequent colour in official matches. To do this, the video sequence is segmented, identifying the objects of interest. Then, simple features are obtained and then utilised to filter them in order to get the most similar objects to the ball. Finally, elliptical features are extracted, improving the accuracy of the representation of the ball in case of distortion.

3.1.1 Segmentation

In computer vision, segmentation is the process of partitioning an image or video into multiple segments, in order to simplify the representation into something that is more meaningful and easier to analyse. In our thesis, the segmentation process separates each frame into foreground segments which contain

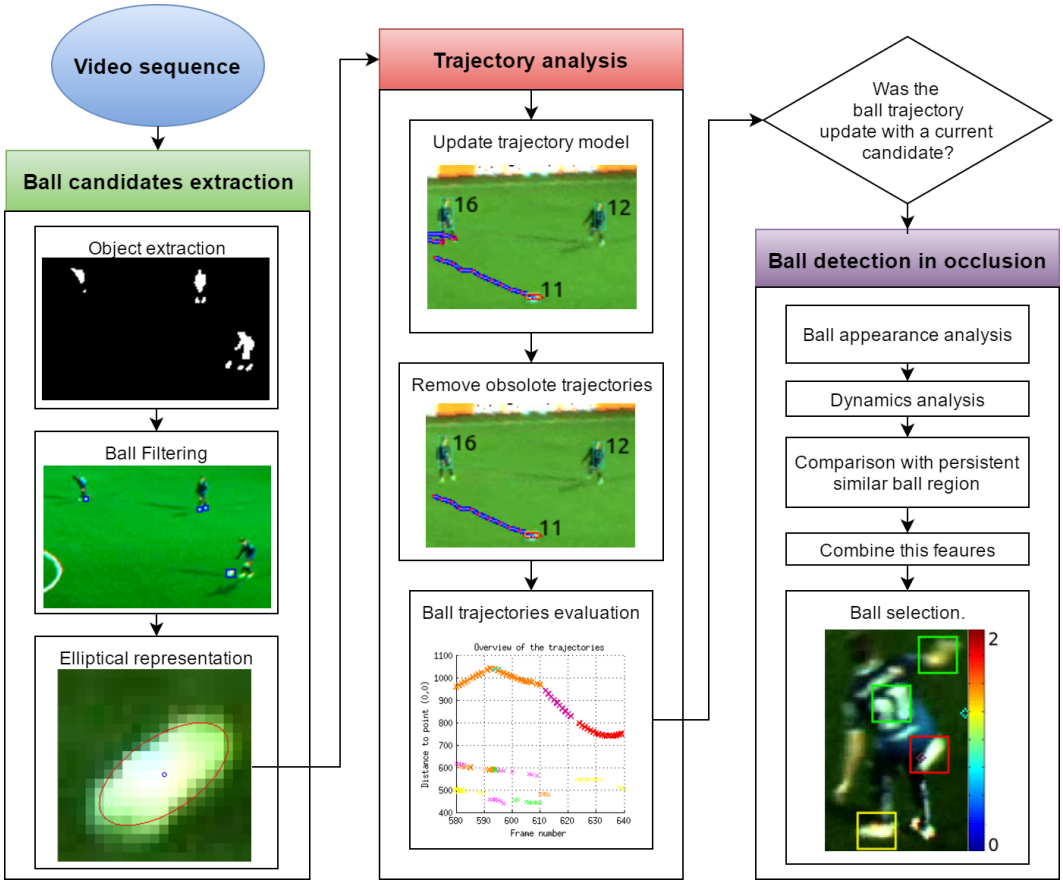


Figura 3.1. Block diagram of the proposed ball tracking algorithm.

each element of interest (represented as white pixels), and background segments which contain the other elements (represented as black pixels).

In our work, a binary segmentation process based in colour is applied [36]. This method is based on the idea that a soccer field has one distinct dominant colour (a tone of green) that may vary from stadium to stadium, and also due to weather and lighting conditions within the same stadium. Therefore, this methodology identifies as foreground each object that is not grass (player, ball and line fields) and, in consequence classifies the grass as background.

The dominant field colour is described by the mean value of each HSI (Hue, saturation, intensity) colour component, computed around its respective histogram peak. The computation involves the determination of the peak index, i_{peak} for each histogram. Then, an interval $[i_{min}, i_{max}]$ about each peak

is calculated, where i_{min} and i_{max} refers to the minimum and maximum of the interval, respectively, satisfying the conditions:

$$i_{min} \leq ipeak \text{ and } i_{max} \geq i_{peak} \tag{3.1.1}$$

$$H[i_{min}] \geq K \cdot H[i_{peak}] \text{ and } H[i_{min}-1] < K \cdot H[i_{peak}] \tag{3.1.2}$$

$$H[i_{max}] \geq K \cdot H[i_{peak}] \text{ and } H[i_{max}+1] < K \cdot H[i_{peak}], \tag{3.1.3}$$

where H refers to the colour histogram and K is a constant value, referring to the minimum percent of the peak value. In our implementation $K = 0.2$. Finally, the mean colour i_{mean} in the detected interval is computed for each color component, described as:

$$i_{mean} = \frac{\sum_{i=i_{min}}^{i_{max}} i * H[i]}{\sum_{i=i_{min}}^{i_{max}} H[i]} \tag{3.1.4}$$

In Fig. 3.2 the histogram of each colour component is illustrated. In red the peak point of the histogram (i_{peak},$H[i_{peak}]$). In green, the intervals points (i_{min},$H[i_{min}]$) and (i_{max},$H[i_{max}]$). In cyan the mean colour (i_{mean}). Besides, the hue colour histogram confirm the existence of one dominant color in the scene

Field coloured pixels in each frame are detected in relation to the distance of each pixel (x, y) to the mean colour. We utilise a cylindrical metric, defined as:

$$d_{intensity}(x,y) = |I_{x,y} - I_{mean}|, \tag{3.1.5}$$

$$d_{chromaticity}(x,y) = \sqrt{S_{x,y}^2 + S_{mean}^2 - 2 \cdot S_{x,y} \cdot S_{mean} \cdot \cos(\theta)}, \tag{3.1.6}$$

$$d_{cylindrical}(x,y) = \sqrt{d_{intensity}(x,y)^2 + d_{chromaticity}(x,y)^2}, \tag{3.1.7}$$

where H, S, and I refer to hue, saturation, and intensity, respectively, and θ is the Hue value in radians.

Finally, the field region is defined as those pixels having $d_{cylindrical} < T_{colour}$, where T_{colour} is a pre-defined threshold value.

In Fig. 3.3 (a) a frame image is illustrated evidencing that green is the dominant colour. In Fig. 3.3 (b) the cylindrical distance in thermal colour is

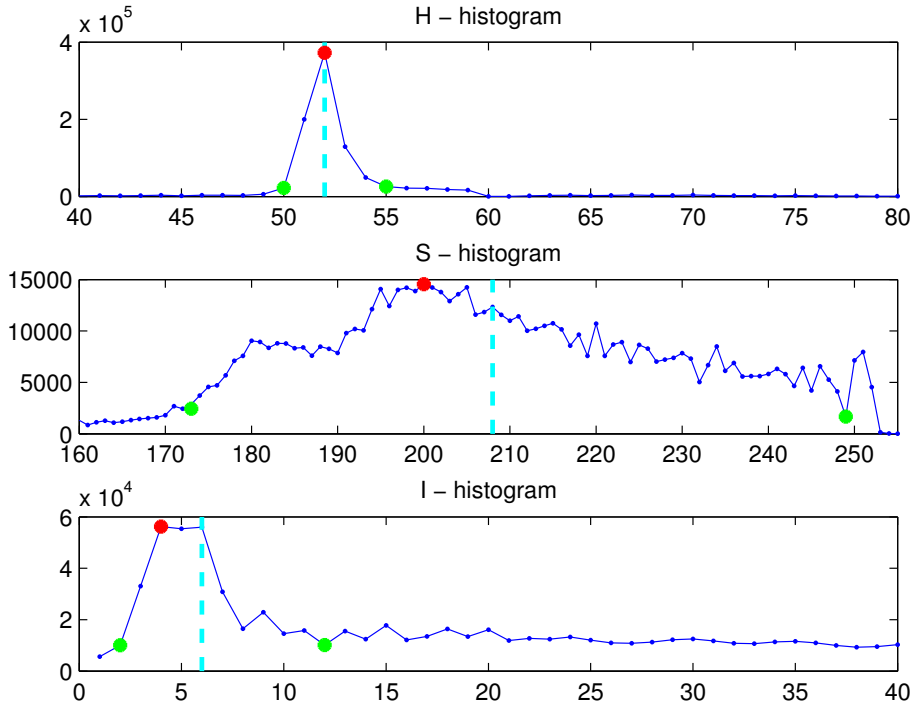
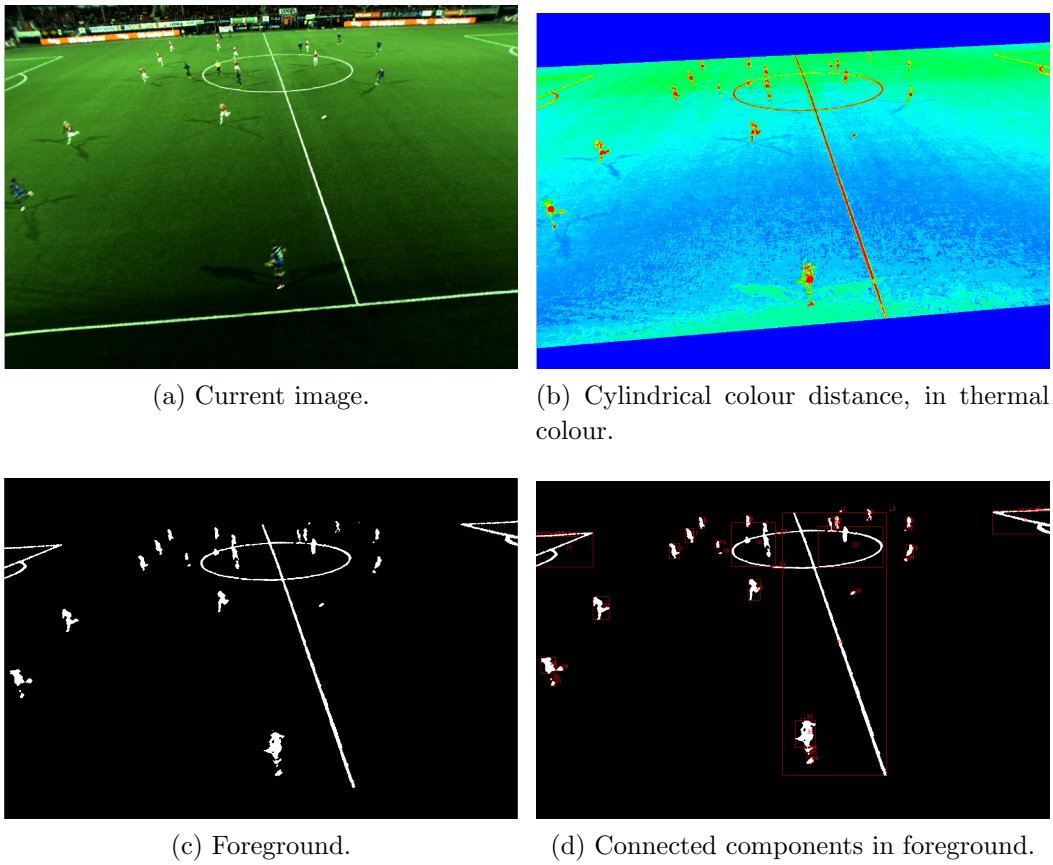


Figura 3.2. Histogram of each HSI component.

illustrated. Besides, a binary mask is applied to ignore other objects outside the field. This is possible because, in our context, the cameras are static. In Fig. 3.3 (c) ball, players, field lines, or a mixture of these objects are segmented as foreground. Finally, in Fig. 3.3 (d) an algorithm for detecting connected regions is applied [37], labelling each object as a blob and representing them in a red bounding box.

3.1.2 Object Filtering

Once every region of interest is identified in the scene, different feature extraction approaches based on colour, shape, and context are utilised in order to filter these regions, obtaining a reduced number of regions. As described in Section 1.3, due to the constant distortions and problems of the ball detection and with the purpose of not implementing a high computational cost algorithm, the filtering process only discards objects which are very different to



(a) Current image.

(b) Cylindrical colour distance, in thermal colour.

(c) Foreground.

(d) Connected components in foreground.

Figura 3.3. Segmentation process.

the ball (e.g. field lines or players), keeping only the most similar ones. Nevertheless, some regions are misclassified, as socks and shoes. From now on, we will refer to the final objects resulting from this process, as *ball candidates*. The details of the appearance criteria are described in Chapter 4.

3.1.3 Elliptical Features

An ellipse representation is associated with each ball candidate, in order to get more specific features of the ball. The elliptical feature is utilised to estimate the real size of the ball when is highly distorted (e.g. by high speed) in order to improve the ball filtering results. Also, the analysis of ball trajectories is improved by combining the deformation orientation with the velocity vector. Also, these features let us estimate the shape of the ball in occlusion situations.

The elliptical approach is described in Chapter 4.

3.2 Second stage - Trajectory analysis

In the second stage, the candidate trajectories are analysed. The features obtained in the previous stage for each candidate are combined with their dynamics model, in order to rank each trajectory determining the real ball as the trajectory with higher rank. The details in the trajectory analysis system are described in Chapter 3.

3.2.1 Trajectory model

For each ball candidate, the trajectory is analysed using a trajectory model which combines the appearance of the candidate with its dynamics model. This way, we can improve the accuracy in locating the real ball position in the scene.

3.2.2 Trajectory processing

The trajectory system ranks each ball-trajectory by a *confidence index*. This index corresponds to a heuristic measure that indicates the likelihood that the trajectory belongs to the real ball. A confidence threshold is then used for selecting the real ball trajectory.

3.3 Third stage - Occlusion detection system

The third stage is used for occlusion situations. An occlusion is detected when the ball candidates are not coherent with the real ball trajectory detected in the second stage. This stage combines the appearance information of the ball to generate possible regions where the ball can be. Then, each possible region is validated with respect to the similarity to no-ball objects in the previous frame, discarding shoes, socks, or any other similar region. The dynamics of the ball are constantly evaluated in order to use them only when we have certainty that the ball does not changes its dynamics.

3.4 Software Tools

As previously mentioned, we utilise the framework VAT, which is developed in **Qt Creator** [1], a cross-platform IDE (integrated development environment) for C++ programming, including libraries for graphic interface design and debugging tools.

Some computer vision algorithms were applied using **Open CV** [2], an open source computer vision and machine learning software library, which provides a common infrastructure for computer vision applications. In this thesis, OpenCV is utilised to:

- Open, save and visualise images.
- Process regions of interest in the image.
- Use simple filters as gradient and gaussian filters.
- Obtain maximal and minimum values of an image.
- Use basic thresholding operations.
- Simplify the drawing of rectangles and circles, improving the representation of our results.
- Generate a thermal image, to represent the confidence of each pixel in an image.
- Use a dynamics model (Kalman Filter).
- Matrix operations.

Finally, **MATLAB** tool [3], a high-performance language for technical computing, was utilised to preliminarily evaluate alternatives of solution, and to graphically visualise and evaluate final results.

[1] Qt Company, web page,
`https://www.qt.io/about-us/`

[2] OpenCV, web page,
`http://opencv.org/`

[3] MathWorks, web page - MATLAB and Simulink for Technical Computing,
`https://www.mathworks.com/products/matlab.html`

Chapter 4

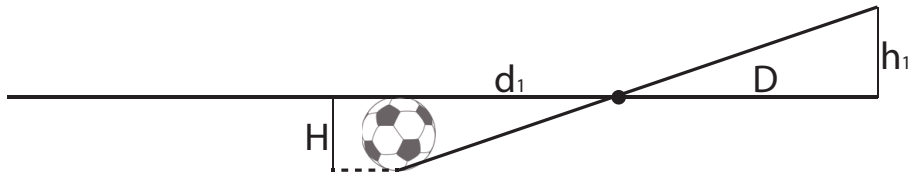
APPEARANCE CRITERIA

In this chapter, we describe, discuss and evaluate different state-of-the-art appearance criteria for classifying the ball. These ball approaches can be classified by size, colour, shape, and context.

4.1 Ball size estimation principle

The size of an object is a common feature utilised to separate the ball from other elements in the field. An accurate size estimation for each frame is important because the ball size varies greatly from frame to frame, due to the distance of the ball to the camera.

The size of the ball in the video depends on the distance to the camera due to the pinhole principle. The effect of the size variation is illustrated in Fig. 4.1.



(a) Ball close to camera.

(b) Ball far to camera.

Figura 4.1. Pinhole principle.

Let H be the real diameter of the ball and h_i the diameter of the ball on

the image. Similarly, let D and d_i be the distance from the pinhole to the ball and to the image, respectively. Then, we have the relation:

$$\frac{h_i}{H} = \frac{d}{D_i}, \tag{4.1.1}$$

Due that H and D do not vary in different ball positions, we can notice that the size h_i is inversely proportional to the distance d_i, because $D_i \cdot h_i = H \cdot d$.

Based on this principle, in [1] the size of the ball is estimated using a procedure to select a number of frames where the ball is located in various positions, presenting different sizes. Using this information and adding a linear interpolation, the ball size is represented in a variation matrix related to the position of the ball in the field. However, this approach is considered in the context of broadcasting, when the camera is moving.

In our context, the estimation of the ball size can be simplified, using a common assumption utilised on static camera systems, as in [29]: *"the projected 2D image height of an object varies linearly with its vertical position in the image, being zero in the vanishing point"*. This assumption enables the use of a simple but highly discriminatory criterion, which indirectly uses the depth of the object to estimate the size of the ball.

Accordingly, we utilise y-axis location of the ball centre (y) to represent the projected diameter of ball $h(y)$, using a simple linear representation:

$$h(y) = a \cdot y + b, \tag{4.1.2}$$

where a represents the size variation per pixel, and b is the distance from the origin point to the vanishing point. These parameters are previously obtained, fitting a linear regression from some frames, similarly to [1].

4.2 Ball colour estimation principle

Due to the very small size of the ball, and to motion, the ball commonly presents a single blurred colour (the ball colours are mixed). For this reason, we utilise a main colour to represent the ball, using a large range, considering changes in the illumination conditions. Sometimes, the ball may have a shade of green due to the reflection of light from the play field. For this reason, we extract the colour from those pixels near the centre of the ball.

In [24], only the blue and red channels are utilised to represent the ball colour into a single value. This representation is based on the assumption that the colour of the ball is commonly mixed with the grass.

Similar to the size estimation, we utilise some samples to learn the mean colour of the ball. Based on this principle, the distance of a pixel to the ball colour is represented as:

$$c_d = (R(i,j) - \bar{R}_b)^2 + (B(i,j) - \bar{R_B})^2, \tag{4.2.1}$$

where c_d is the colour distance. $\bar{R_B}$ and $\bar{R_B}$ are the mean red and blue colour channels of the ball, respectively. $R(i,j)$ and $B(i,j)$ are the red and blue channel colours of the pixel in (i,j). To determine if a pixel has a colour similar to the ball, a simple threshold operation is applied:

$$c_d < T_c, \tag{4.2.2}$$

where T_c is calculated in relation to the colour distance error of some ball samples.

4.3 Appearance criteria

In this Section, the state-of-the-art criteria to represent the appearance of the ball are described. Each criterion is separated in relation with its main feature type (size, colour, shape, context). For each criterion we utilise a simple threshold to determine if an object is the ball.

4.3.1 Dimension appearance criteria

The dimension of the ball is utilised to represent the scale of ball. On these approaches the dimension of the ball is represented using the minimum bounding box obtained from the segmentation process. The bounding box is a rectangle that encloses an object of interest in the image.

Bounding box size criterion

We utilise the height (H) and width (W) of the bounding box in order to discard big objects, mainly field lines. We utilise a simple threshold criterion to discard no-ball objects:

$$H > T_H, \tag{4.3.1}$$

$$W > T_W, \tag{4.3.2}$$

where the size thresholds T_H and T_W are obtained in relation to the error between the bounding box size and the predicted size of the ball. The predicted size is based in the principle explained in Section 4.1. This criterion is not too accurate, because this ball representation is very sensitive to distortions, representing the ball as a big object when it is extremely distorted.

Eccentricity

The eccentricity descriptor of the ball is utilised in [1,5,25,38]. Eccentricity is the ratio between the longest diameter and the shortest one of a region, and is calculated as:

$$Ecc = \frac{\max(W, H)}{\min(W, H)}, \tag{4.3.3}$$

where W and H are the width and height of the bounding box, respectively. In case of perfect circularity, this value is $Ecc = 1$. However, due to distortions, this value tends to increase. We use a simple threshold $Ecc < T_{ecc}$ to determine if and object is the ball. Similarly to [1] we set $T_{ecc} = 3$.

4.3.2 Color appearance criteria

The colour descriptors of the ball are based in the colour principle explained in Section 4.2.

Tong - Central colour representation

This descriptor is based on the assumption that the ball presents one single colour, and the centre colours are stronger than those in the edge, due to the blurring of the ball with the background. For these reasons, a few pixels of the centre are enough to represent the ball colour. In this criterion, five pixels of the centre are utilised (a single morphological cross of 3x3). We utilise the colour distance explained in Section 4.2.

Yu - colour representation

This descriptor is presented in [1] and focuses on counting the number of pixels of the ball which their colour is close to the colour model. Yu defines this parameter as:

$$\frac{pxb}{A} > T_c, \tag{4.3.4}$$

where A is the total number of pixels of the ball, and pxb is the number of pixels that are inside the colour range, defined in Eq. 4.2.2.

4.3.3 Shape appearance criteria

Many shape descriptors have already been studied in the literature and applied in practice. In this section we list just a few known methods for measuring the shape of the ball. This methods analyse the shape of the binary image obtained in the segmentation process as depicted in Fig. 3.3 (c).

Form factor

The form factor descriptor is presented by Tong in [5], but also it is utilised in [25, 38] with other names. The form factor is a parameter, referred as F, derived from the relation between the shape area and shape perimeter. F is calculated as:

$$F = \frac{P^2}{4\pi A}, \tag{4.3.5}$$

where P is the perimeter of the shape, calculated as the number of pixels in the contour of the foreground shape. A is the area of the shape, calculated as the number of pixels in the shape. The more circular the shape is, the more the form factor is closer to one, because in perfect ball shape $P^2 = 4\pi A$. We utilise a simple threshold to determine if an object is the ball.

Circular variance

The circular variance is a shape descriptor utilised in [5, 24, 39]. This parameter can describe the resemblance to a circle of a region. Circular variance parameter (CV_1) is defined as:

$$CV_1 = \frac{\mu_R}{\sigma_R}, \tag{4.3.6}$$

where μ_R and σ_R are the mean and variance of the distance to its centroid to all points on the contour, respectively. These parameters are calculated as:

$$\mu_R = \frac{1}{K}\sum_{i=0}^{K-1}\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2} \tag{4.3.7}$$

$$\sigma_R = \frac{1}{K}\sum_{i=0}^{K-1}(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2}-\mu_R)^2, \tag{4.3.8}$$

where K is the number of pixels on the contour. ($\bar{x}$,$\bar{y}$) is the centre of the region. This parameter varies between $[0, \text{inf}]$. In order to set this range to $[0, 1]$ to contrast to other shape descriptors, we modified this parameter to:

$$CV_2 = 1 - e^{-\frac{\mu_R}{\sigma_R}}, \tag{4.3.9}$$

This adaption just changes the parameter scale, but not its performance.

Circularity

Yu *et al.* present this parameter in [1]. Using the relation between the area and radius of a circle, this parameter is calculated as:

$$C = \frac{A}{\pi r^2} \tag{4.3.10}$$

where A is the number of pixels of the ball, and r is the estimated radio, calculated as the mean distance from the centre to the contour of the ball. The more circular the shape is, the more this parameter is closer to one.

Hu moment

This descriptor utilises the well known Hu moment invariants to analyse the circularity of the shape. In [21], this parameter is compared to the most standard circularity measures, performing better in the case of shapes with boundary defects. To calculate this parameter, first, we have to define the $p-q$ moment $m_{pq}(S)$ of a planar shape S as:

$$m_{pq}(S) = \int\int_S x^p y^q dxdy \tag{4.3.11}$$

Basic shape features are computed from the $p-q$ moments. E.g $m_{00}(S)$ is equal to the area of S, while ($m_{10}(S)/m_{00}(S)$,$m_{01}(S)/m_{00}(S)$) is the centroid

of S. The centralised $p-q$ moment $\mu_{pq}(S)$ is calculated as:

$$\mu_{pq}(S) = \int\int_S (x - \frac{m_{10}(S)}{m_{00}(S)})^p (y - \frac{m_{01}(S)}{m_{00}(S)})^q dxdy \tag{4.3.12}$$

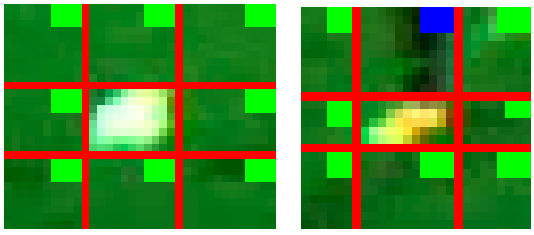
Finally, the Hu moment parameter $\Phi(S)$ is defined as:

$$\Phi(S) = \frac{1}{2\pi} \frac{\mu_{00}(S)^2}{\mu_{00}(S) + \mu_{02}(S)} \tag{4.3.13}$$

The more circular the shape is, the more this parameter is closer to one.

4.3.4 Context criterion

This criterion assumes that the ball exists in the field and is surrounded by grass. Eight neighbouring regions of the candidate of ball are evaluated, as depicted in Fig. 4.2. Only the object where all the neighbours are dominant grass are classified as ball. This descriptor leads to discard bad segmented parts of players. The grass colour is obtained in the segmentation process as the dominan colour, it calculation is described in Section 3.1.1.



(a) Ball region and eight grass regions.

(b) No-ball region and seven grass regions.

Figura 4.2. Examples of grass context.

4.4 Coarse-to-fine object filtering

Coarse-to-fine is a simple strategy, used for detecting objects in the image using different feature criteria, avoiding an expensive and unnecessary search for classifying the ball candidates. At the coarse step, the objects that obviously are not resembling the ball are eliminated using simple features. At the

fine step, ball candidates are further examined using a more complex feature criteria.

Each criterion is utilised to filter no-ball regions, in order to decrease the number of ball candidates. Based on the coarse-to-fine principle, we filter each object in the following order: size, colour, shape, and context (as depicted in Fig. 4.3).

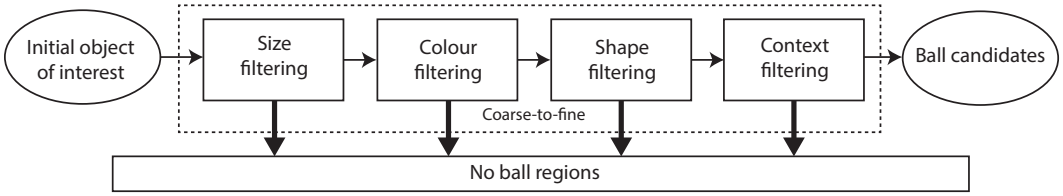


Figura 4.3. Coarse-to-fine structure for detecting ball candidates.

This order is previously defined because:

1. The size of the ball can be calculated using the dimension parameters previously obtained in the segmentation process, when the connected component of each object is obtained, without requiring to analyse the whole image.

2. Colour analysis requires processing only the direct channel values; we utilise the RGB colour space.

3. Shape features require deeper processing of the image, resulting in a higher computational load in comparison to colour features.

4. The context parameter requires processing the candidate region and its surroundings.

In Table 4.1 the processing time of each criterion is presented (Tested in Matlab). These times are coherent with the conventional structure of the coarse-to-fine-filter applied to the ball, excepting Yu-colour, because this feature requires the shape information to compute the colour of the ball. However, in Section 4.5.2 preliminary results shown that Yu-colour is not performing well in comparison with the other colour criteria, maintaining the structural criteria of the filter.

Criteria	**Mean[ms]**	**Std. dev [ms]]**
Dimensional descriptors		
Bounding box size	0.05	0.03
Eccentricity	0.14	0.10
Colour descriptors		
Tong - Central colour	0.43	0.03
Yu - Colour representation	4.30	3.64
Shape descriptors		
Form factor	2.13	1.5
Circular variance	1.22	0.94
Circularity	0.85	0.66
Hu moment	1.26	0.98
Context descriptor		
Context criterion	14.41	11.1

Table 4.1. Processing time of each criteria.

4.5 Preliminary results

We need to first test the performance of each descriptor as a classifier in order to build a coarse-to-fine filter which could present a higher global time performance.

Following the coarse-to-fine strategy explained in Section 4.4, we elaborate a testing methodology depicted in Fig. 4.4. First, we extract a set of images from the segmentation process. These images contain the real ball, players, parts of players, field lines, or a mix of these objects. We manually create a dataset which contains 95 ball images and 3534 no-ball images. Among the ball images, some of them present high distortion. This set of images is analysed with respect to each stage in order to determine the most relevant descriptors. Then, we utilise these most relevant descriptors to filter the extracted images, generating a sub-dataset for the next stage. For example, shape descriptors

were tested without considering field line regions or other objects, because they should have been discarded in a previous stage (with size and colour descriptors).

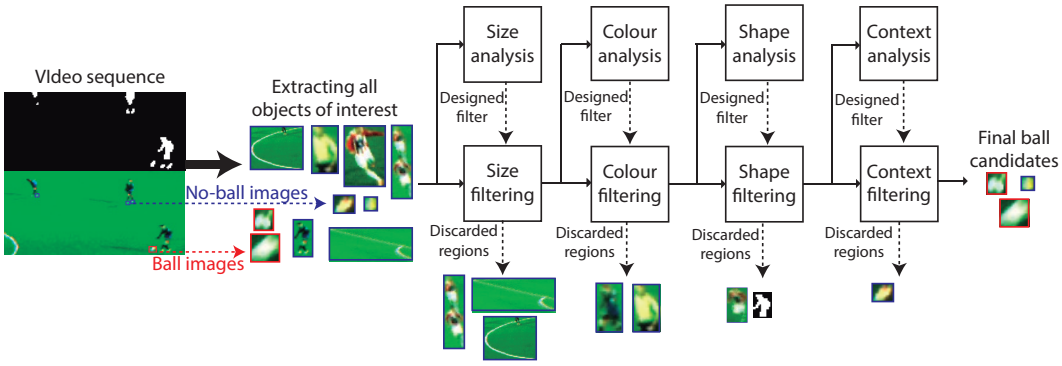


Figura 4.4. Preliminary tests to design coarse-to-fine structure.

In order to decide which criterion is going to be finally utilised, we first evaluate the performance of all the criteria of the same type. If the best-performed criterion is able to filter a significant number of images, we select it for our coarse-to-fine structure as the first criterion of its stage. Besides, we re-evaluate the performance of the other criteria, but previously filtering the initial dataset with the selected criterion. With the new dataset, we re-evaluate if it is necessary to use another descriptor in the same stage, according to its performance.

To measure the performance of each feature as a classifier, we plot a ROC curve. The ROC curve is defined as *"a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings"* [1]. Due to the coarse-to-fine structure, we are interested in filtering as many regions as possible, but maintaining the real ball regions.

[1]Receiver operating characteristic - Wikipedia, The Free Encyclopedia, `https://en.wikipedia.org/w/index.php?title=Receiver_operating_characteristic`

4.5.1 Preliminary results of dimensional descriptors

We evaluate the performance of each dimensional descriptor. The results of this analysis are presented in Fig. 4.5. We recognise that the *bounding box dimension* presents a notoriously better performance than the *eccentricity*.

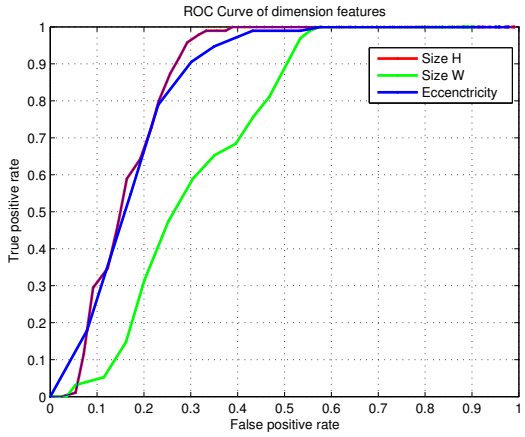


Figura 4.5. ROC curve of size features.

However, when we filter each region by *bounding box dimension* (discarding 710 no-ball images and generating a sub-dataset with 95 ball images and 2815 no-ball images), we notice that the eccentricity can filter different no-ball regions presenting a good performance, as depicted in Fig. 4.6.

4.5.2 Preliminary results of colour descriptors

Both colour features were tested after filtering the initial regions by dimensional descriptors (discarding 317 no-ball images and generating a sub-dataset with 95 ball images and 2438 no-ball images). In Fig. 4.7, we can appreciate that both colour representations discard a considerable number of no-ball images. This is because the colour is a good feature to differentiate the ball from players (when they have a different colour in their clothes) and the majority of regions in this stage are players. We notice that *Yu-colour* initially gets a better performance, but the *central colour* descriptor can reject more images maintaining the number of real ball images. The performance deterioration of *Yu-colour* is because it cannot identify the ball when it is blurred due that the colours in the contour are mixed with the grass colour.

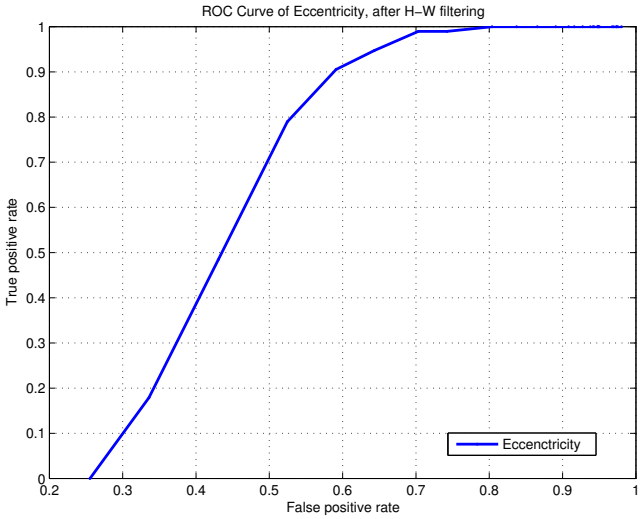


Figura 4.6. ROC curve of size features, after filter by boounding box.

4.5.3 Preliminary results of shape descriptors

We test the performance of each shape descriptor over the real ball and no-ball images, previously obtained filtering by size and colour (discarding 1078 no-ball images and generating a sub-dataset with 95 ball images and 1360 no-ball images). In Fig. 4.8, we can appreciate that the *form factor* presents a significantly better performance to classify ball and no-ball regions.

After applying *form factor* filtering (discarding 415 no-ball images and generating a sub-dataset with 95 ball images and 945 no-ball images), we test the performance of the other descriptors. In Fig. 4.9, we can appreciate that the other descriptors do not present a good performance.

4.5.4 Preliminary results of context descriptor

At this stage, we have 95 ball images and 945 no-ball (we only filter with *form factor* in the shape stage). In Fig. 4.10 we can appreciate that the **context** descriptor can discard a significant number of no-ball regions, maintaining all the real ball images. After filtering by context, we discard 298 no-ball images, obtaining 95 ball images and 647 no-ball images.

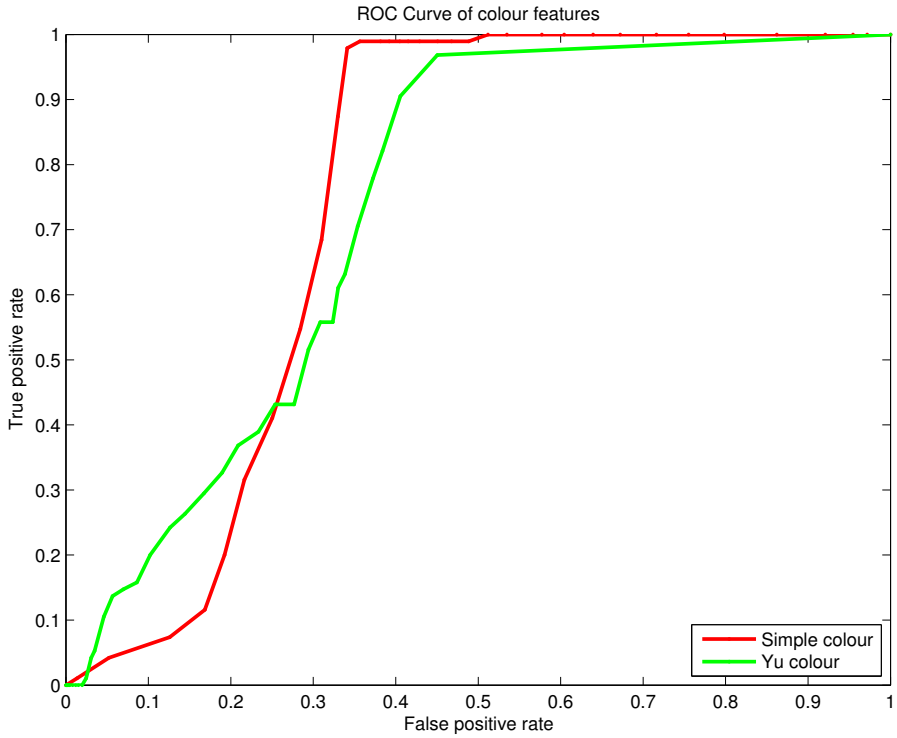


Figura 4.7. ROC curve of colour features.

4.6 Building coarse-to-fine filter

Based on the preliminary results, we build the course to fine structure, which tries to filter no-ball regions, but maintaining the number of real ball images. For this reason, we select threshold values T_i, for each applied descriptor i, which provide a true positive rate over 0.98 in their respective ROC curve. However, in order to guarantee that each filter can maintain more distorted ball region than those present in the evaluation dataset, we increase the threshold value in relation with the standard deviation σ_i, obtained for each selected descriptor. With this consideration, we set the final threshold value as:

$$T_i^* = T_i + K \cdot \sigma_i \tag{4.6.1}$$

We have empirically determined a value of $K = 0.2$.

We finally build the coarse-to-fine filter, following the conclusions for each stage, described below:

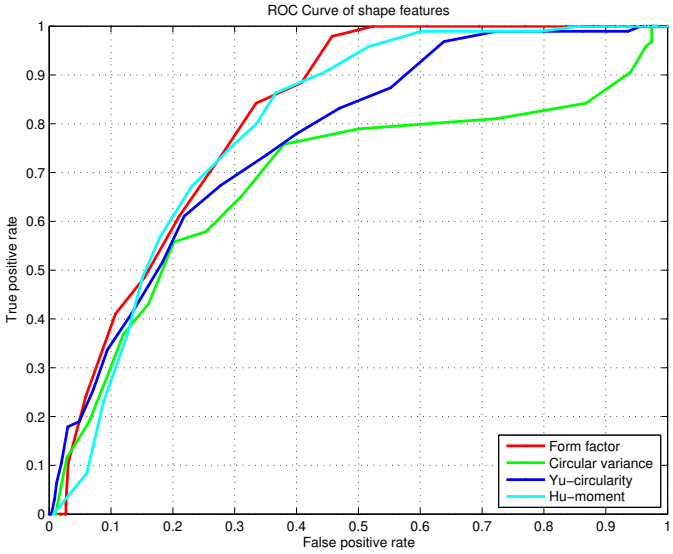


Figura 4.8. ROC curve of shape features.

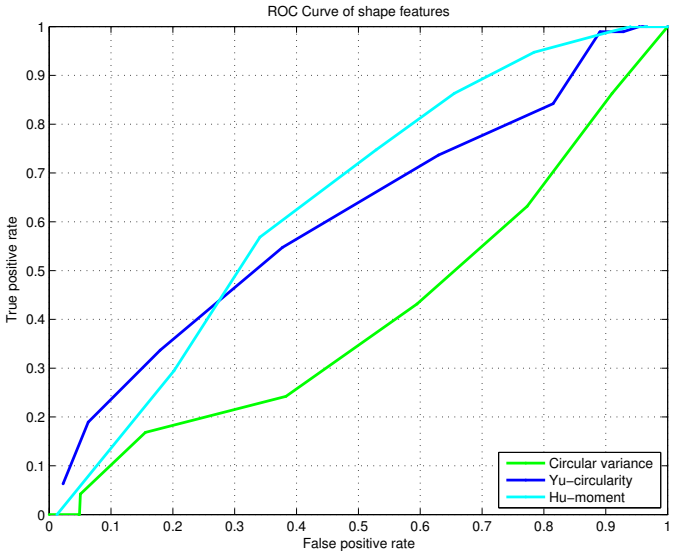


Figura 4.9. ROC curve of shape features, after filtering by form factor.

Dimensional stage

The ROC curve of dimensional descriptors shows that the *bounding box dimension* descriptor presents a significantly better performance. However, the *eccentricity* descriptor is capable of discarding other no-ball regions that *bound-*

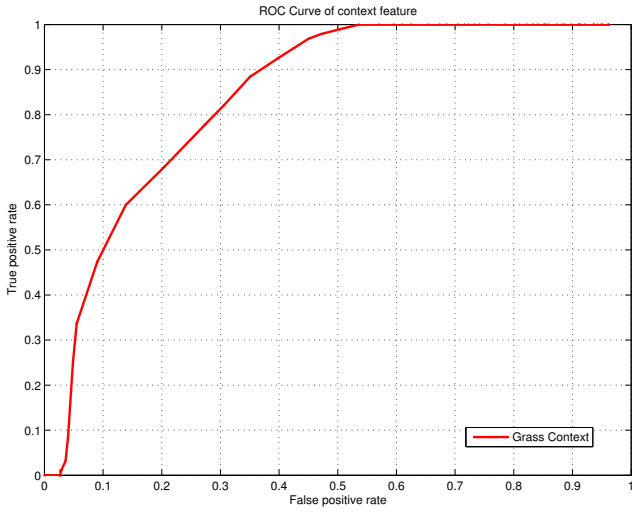


Figura 4.10. ROC curve of context feature.

ing box dimension cannot. For these reasons, we implement both features in the dimensional stage.

Colour stage

The ROC curve of colour descriptors shows that both perform well. However, although *Yu-colour* presents a similar performance to *central colour*, it presents complications to recognise a little number of ball images. Besides, as shown in Table 4.1, *Yu-colour* demands high processing time. For these reasons, we decide to incorporate only the *central colour* descriptor in the colour stage.

Shape stage

The ROC curve of shape descriptors shows that the *form factor* presents a better performance. Besides, after filtering by *form factor*, the other descriptors decrease significantly their performance, because they are very sensitive to ball distortion and they are strongly correlated to the *form factor*. For these reasons, we decide to only incorporate the *form factor* descriptor in the shape stage.

Context stage

The ROC curve of the *context* descriptor shows that this criterion able to properly filtering a significant number of no-ball images. For this reason, we decide to incorporate it in the context stage.

Defining the structure of coarse-to-fine filter

The structure and parameters of the coarse-to-fine filter are presented in Table 4.2.

Criteria	**Threshold**	**Mean[ms]**	**std dev**	**#Initial images**	**#Discarded images**	**%Filtered images**
Dimensional stage						
Bounding box size	30	0.05	0.03	3534	719	20.34%
Eccentricity	2.34	0.14	0.10	2815	377	13.39%
Colour stage						
Central colour	90	0.43	0.03	2438	1078	44.21%
Shape stage						
Form factor	1.4	2.13	1.5	1360	415	30.51%
Context stage						
Context criterion	15.8	14.41	11.1	945	298	31.53%

Table 4.2. Survey of configurations and preliminary results of coarse-to-fine filter.

4.7 The effect of high velocity on the ball

The dimension of a the ball may be over-estimated, specially when it is moving. This is because, firstly, there is a motion blur effect caused by high speed. Second, if the ball is on the ground, its size is linearly related to the Y-axis position (explained in Section 4.3.3). Nevertheless, this relation will be over-estimated when the ball is located in a significant elevation from the ground, making necessary a 3D representation using multi-camera. Due to the scope of this thesis, we deal only with the first case. However, preliminary results show that the second case does not over-estimate the ball size significantly.

When a camera captures a frame, it does not represent a single instant of time. Due to the camera constraints (shutter speed [2]), the image may represent

[2]Shutter speed - Wikipedia, The Free Encyclopedia,

the scene over a period of time. Most often, this exposure time is brief enough for considering that the captured frame represents an instantaneous moment. However, this is not always so, and a fast moving object or a longer exposure time may result in blurring artifacts which make the ball look like an elliptical shape. This effect is illustrated in Fig. 4.11, we can observe that the distortion is a representation of the change of the ball position over a period of time.

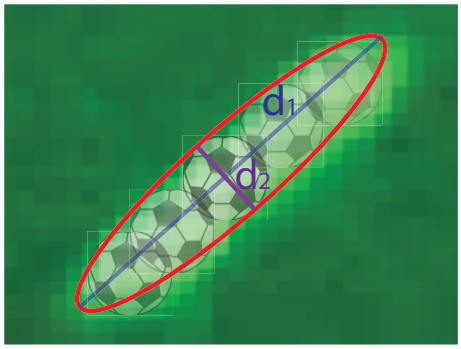


Figura 4.11. Motion blur representation.

In this representation the ball displacement can be represented using the maximum distortion (d_1) and the ball maintain its size in the orthogonally direction (d_2.

4.7.1 Elliptical representation of the ball

In contrast to other approaches, we propose an elliptical ball model which is depicted in Fig. 4.12. We use an elliptical representation because it is a simple way to fit the visual deformation of the ball in these cases. In high speed, the shape of the ball is elongated in the direction of its velocity vector (maximum radius of the ellipse $\vec{A}$), maintaining the original size of the ball orthogonally (minimum radius of the ellipse $\vec{B}$).

In order to calculate the elliptical parameters of the ball, we utilise principal component analysis (PCA) in the ball foreground image assuming that each pixel in the foreground is part of the input dataset. With PCA, we can find the orientation of the ball deformation using the eigenvectors. Besides, the eigenvalues lead to determine the maximum and minimum radius of the ellipse. It is then easy to determine the orientation of the ellipse ϕ, by using the

`https://en.wikipedia.org/w/index.php?title=Shutter_speed`

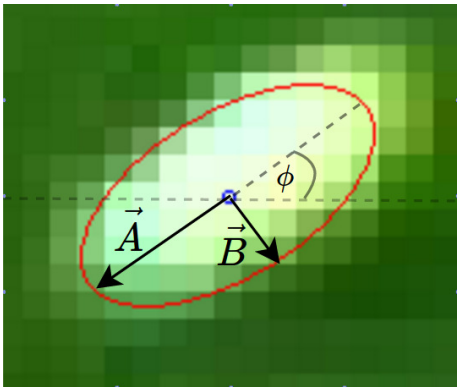


Figura 4.12. Elliptical representation of the ball. ϕ is the orientation angle. $\vec{A}$ and $\vec{B}$ are the maximum and minimum radius vector of the ellipse, respectively.

angle between the maximum radius eigenvector $\vec{A}$ and the x-axis of the image referential.

As in PCA the eigenvalues do not variate if the analysed object translates or rotates, we can generalize the ball model representation in a new coordinate system, where a non-rotated ellipse is centred in the origin. This projection is useful to simplify the calculation, because in the new system there is no correlation between the new axis dimensions. This way, the covariance matrix is diagonal, and the eigenvalues are equal to the variance of the dimensions in this space ($\sigma^2_{uu} = \lambda$).

We will name the dimension of the first principal component as $\vec{u}$ and the dimension of the second one as $\vec{v}$. This change of coordinates is illustrated in Fig. 4.13

In order to prove the relation between the elliptical parameters with the shape of the ball, we initially consider its shape as a perfect ellipse, in a continuous space. Reformulating the variance as a continuous equivalent, we get:

$$\sigma^2_{uu} = \frac{\sum_{i=0}^{N}(u[i] - \mu_u)^2}{N} \rightarrow \hat{\sigma}^2_{uu} = \frac{\int\int_D u^2 dudv}{Area_D}. \tag{4.7.1}$$

The continuous equivalent of the discrete position of the pixel $u[i]$ is the continuous position u. N is the number of pixels in the ellipse D, and its continuous equivalent is the area of the ellipse $Area(D)$. Finally, the continuous

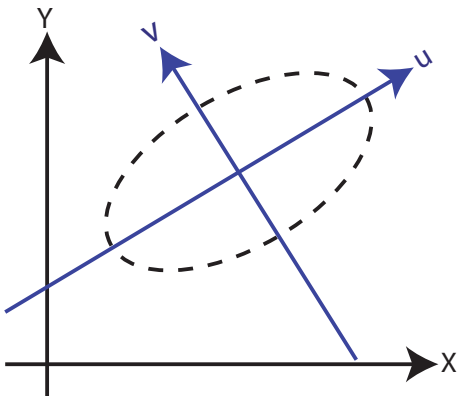


Figura 4.13. PCA space dimension of the ball.

equivalent of the sum of all pixels in the foreground is the surface integral of the ellipse.

With this continuous expression, we can prove that the maximum radius $|\vec{A}|$ of a continuous ellipse can be calculated using the principal eigenvalue λ_1, as:

$$|\vec{A}| = 2 \cdot \sqrt{\lambda_1} \tag{4.7.2}$$

Proof

To solve the surface integral, we need firstly to define the elliptical surface in the continuous domain as:

$$\frac{u^2}{|\vec{A}|^2} + \frac{v^2}{|\vec{B}|^2} < 1 \tag{4.7.3}$$

Then, in order to simplify the calculation of the surface integral, we apply a change of variable defined as:

$$u(r,\theta) = |\vec{A}| \cdot r \cdot \cos(\theta) \tag{4.7.4}$$
$$v(r,\theta) = |\vec{B}| \cdot r \cdot \sin(\theta) \tag{4.7.5}$$

With this parametrization, the surface integral is defined in the domain:

$$-\pi < \theta < \pi \tag{4.7.6}$$
$$0 < r < 1, \tag{4.7.7}$$

The Jacobian of this parameter transformation is:

$$\begin{aligned} J(r,\theta) &= \frac{du}{dr}\frac{dv}{d\theta} - \frac{dv}{dr}\frac{du}{d\theta} && (4.7.8)\\ &= |\vec{A}| \cdot |\vec{B}| \cdot r \cdot (\cos(\theta)^2 + \sin(\theta)^2 && (4.7.9)\\ &= |\vec{A}| \cdot |\vec{B}| \cdot r, && (4.7.10) \end{aligned}$$

We re-define the surface integral as:

$$\begin{aligned} \int\int_D u^2 du dv &= \int_{-pi}^{pi}\int_0^1 J(r,\theta) \cdot u(r,\theta)^2 dr d\theta && (4.7.11)\\ &= \int_{-pi}^{pi}\int_0^1 (|\vec{A}| \cdot |\vec{B}| \cdot r) \cdot (|\vec{A}| \cdot r \cdot cos(\theta))^2 dr d\theta && (4.7.12)\\ &= |\vec{A}|^3 \cdot |\vec{B}| \cdot \int_0^1 r^3 dr \cdot \int_{-pi}^{pi} \cos(\theta)^2 d\theta && (4.7.13)\\ &= \vec{A}|^3 \cdot |\vec{B}| \cdot (\frac{1}{4}) \cdot (\pi) && (4.7.14)\\ &= \frac{|\vec{A}|^3 \cdot |\vec{B}| \cdot \pi}{4} && (4.7.15) \end{aligned}$$

It is not necessary to demonstrate that the area of an ellipse is $Area(D) = \pi \cdot |\vec{A}| \cdot |\vec{B}|$. Then, solving the expression of Eq. 4.7.16, we get:

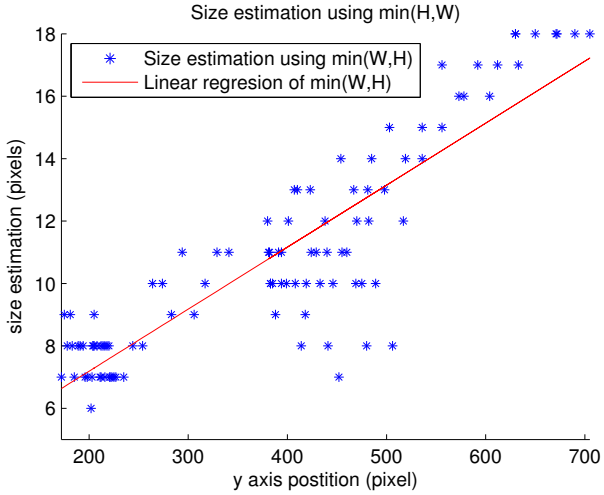
$$\lambda_1 = \frac{\int\int_D u^2 du dv}{Area_D} = \frac{|\vec{A}|^3 \cdot |\vec{B}| \cdot \pi/4}{\pi \cdot |\vec{A}| \cdot |\vec{B}|} = \frac{|\vec{A}|^2}{4} \tag{4.7.16}$$

Finally, solving for $\vec{A}$, we get $|\vec{A}| = 2\sqrt{\lambda_1}$. Similarly, we get $|\vec{B}| = 2\sqrt{\lambda_2}$, with $\vec{A}$ in the direction of $\vec{u}$ and $\vec{B}$ in the direction of $\vec{v}$.

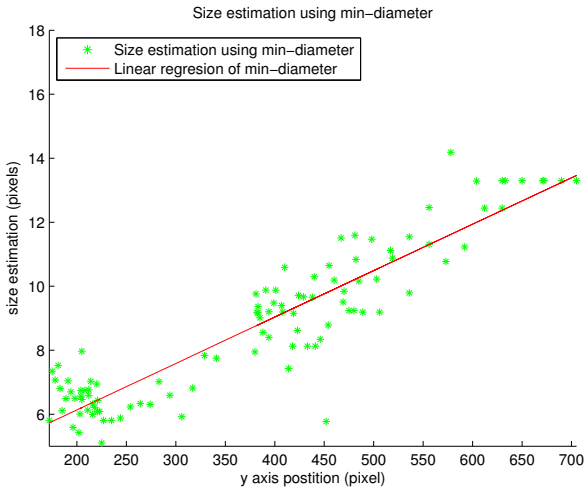
4.7.2 Size prediction using elliptical features

Using the elliptical representation, we can notice that the minimum radius is representative of the real ball size in the image. According to the ball size

principle (Section 4.1), we compare the relation between the position of the ball and its size in the image, using the minimum bounding box dimension ($min(W, H)$) and the minimum diameter of the ball (*min-diameter*) criteria. In Fig. 4.14 we can appreciate that the *min-diameter* approach presents a better fitting of a linear regression, obtaining a mean error of 0.865 pixels in comparison with the $min(W, H)$ which presents a mean error of 2.519 pixels.



(a) Using $\min(H, W)$ criterion to estimate the size.



(b) Using min-diameter criterion to estimate the size.

Figura 4.14. Linear fitting of the ball size using different dimensional criteria.

This way, we set the predicted size of the ball $h_B(y)$, using the bounding

box dimension as:

$$h_B(y) = a_B \cdot y + b_B \tag{4.7.17}$$

where $a_B = 0.0199$ and $b_B = 3.152$. Similarly, we set the predicted size of the ball $h_E(y)$, using the dimension of the ellipse as:

$$h_E(y) = a_E \cdot y + b_E \tag{4.7.18}$$

where $a_E = 0.0145$ and $b_E = 3.2320$. In our approach, we utilise the elliptical dimension because it is robust to the ball distortion. In the other hand, the predicted size of the ball using the bounding box is utilised to contrast our approach to conventional methods [1].

4.7.3 Relation between elliptical shape and ball velocity

To model the motion blur effect, the expected dimension of the ball is adapted to be a function of the velocity. In order to study this relation, we use a dataset of real ball images extracted from a sequence, which contains different distortions, velocities and positions in the field.

In particular, we studied the relation between the elliptical parameters and the velocity norm of the ball. According to the motion blur effect, we utilise the distance from the focus of the ellipse to its centre (focal distance) to represent the effect of the distortion.

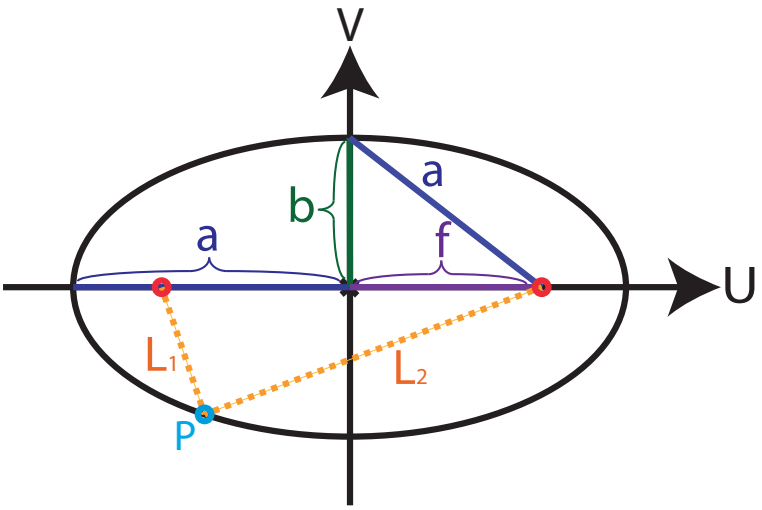


Figura 4.15. Focal distance f in an ellipse.

The elliptical parameters are illustrated in Fig. 4.15. In this figure, we can appreciate the focal distance (f), the maximum and minimum radius of the ellipse a and b, respectively, and the distance (L_1 and L_2) from any location in the contour of the ellipse (P) to each focus. The focuses or foci are defined as two points inside the ellipse, where $L_1 + L_2 = 2a$. Therefore, when the point P crosses the vertical axis, we get the relation $L_1 = L_2 = a$. With this relation, we can determine the focal distance f as:

$$f = \sqrt{a^2 - b^2} \tag{4.7.19}$$

We analyse the relation between the ball distortion (focal distance) and its speed. This relation is illustrated in Fig. 4.16. We can appreciate that the relation can be fitted with a linear regression through the origin (zero distortion when the ball does not move), represented as:

$$f = K_f * |\vec{v}|, \tag{4.7.20}$$

where K_F is a constant that depends of the shutter speed of the camera. In our case, we set $K_f = 0.216255$.

This way, we can estimate the focal distance of the ball when it is distorted in function of its speed. Finally, the largest radius of the ellipse can be calculated solving a in Eq. 4.7.19:

$$a = \sqrt{b^2 + f^2}, \tag{4.7.21}$$

where b is estimated using the $h_E(y)$, presented in Fig. 4.14.

4.7.4 Relation between distortion and velocity angle

The distortion direction is the orientation which the ball is more elongated. It is represented by the angle of the maximum radius of the ellipse ($\angle\vec{A}$). When the ball presents the motion blur effect, it is elongated in the same direction of the velocity vector. The angle difference between the elongation angle ($\angle\vec{A}$) and velocity angle ($\angle\vec{V}$) is denoted by $\angle\psi$ and calculated as:

$$\angle\psi = |\angle\vec{A} - \angle\vec{V}| \tag{4.7.22}$$

Although the angle difference has to be low when the ball is moving, this is not true when the ball moves at low speed. In Fig. 4.17 we can appreciate

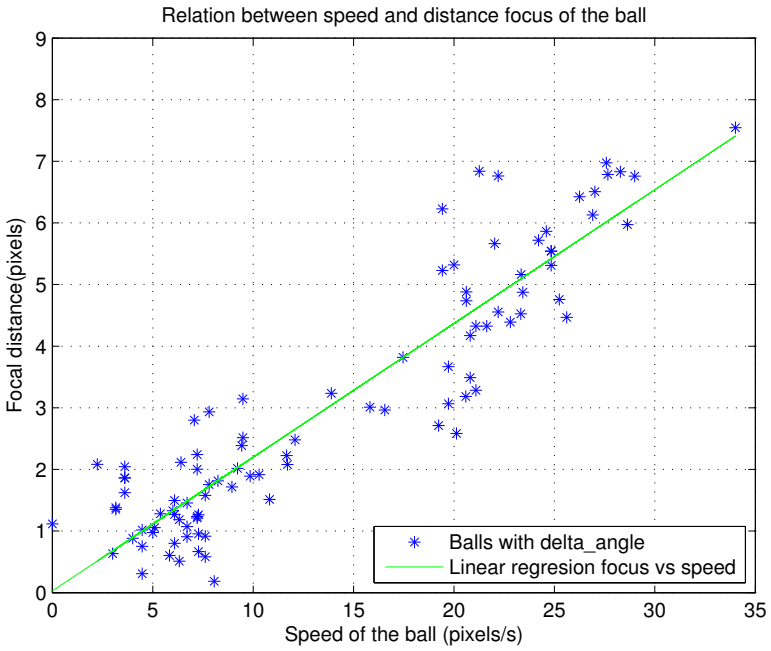


Figura 4.16. Focal distance vs speed.

that the angle difference in low-speed cases is much sparse than in high-speed cases.

This way, we use angle difference to increase the confidence associated to a ball candidate object, by combining this ball appearance criterion with its dynamic model in the trajectory analysis, in Section 5.2. For this purpose we use the relation:

$$\angle\psi < T_\psi \tag{4.7.23}$$

where T_ψ is a threshold value, that we set as $T_\psi = \pi/12$. However, we use this relation when the speed is in an interval of confidence $|\vec{V}| > 12[pixel/s]$.

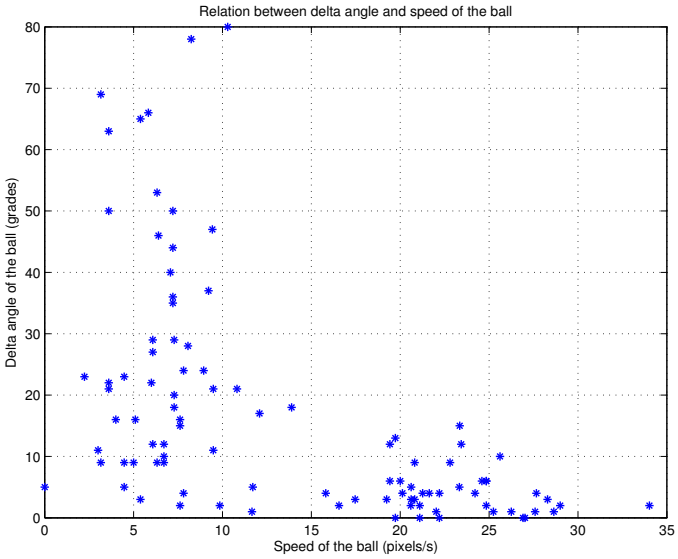


Figura 4.17. Relation between distortion and velocity angle vs speed.

Chapter 5

TRAJECTORY ANALYSIS

The second stage analyses the trajectory of each ball candidate in order to determine where the real ball is. This stage is mainly based in a trajectory-based system proposed by Yu *et al.* in [1]. In this chapter, we first explain the proposed system by Yu *et al.* Then, we describe our modification to this system, with the purpose of dealing with distortions.

5.1 Yu - Trajectory analysis system

Yu *et al.* created a trajectory analysis system, which combines the appearance and dynamics to study the trajectory generated by each ball candidate. We base our work on this approach because it has been proved that this methodology outperforms other state-of-the-art approaches in complex situations [18].

In Fig. 5.1, the block diagram of this methodology is illustrated. There are four main steps: Ball size estimation, ball candidate detection, candidate trajectory detection and trajectory processing. Our method is basically based on the candidate trajectory detection and trajectory processing steps. However, we also describe the first two steps (ball size estimation and candidate detection) to understand in a better way how they utilise their trajectory analysis.

5.1.1 Ball size estimation

This step estimates the size of the ball using the perspective of the camera, in the same way as described in Section 4.1, but using a variation matrix to contrast the size of the ball from other salient objects, such as the centre circle of the field, the goalmouth, or the players. However, this approach is considered for broadcasting cases, when the camera constantly changes its

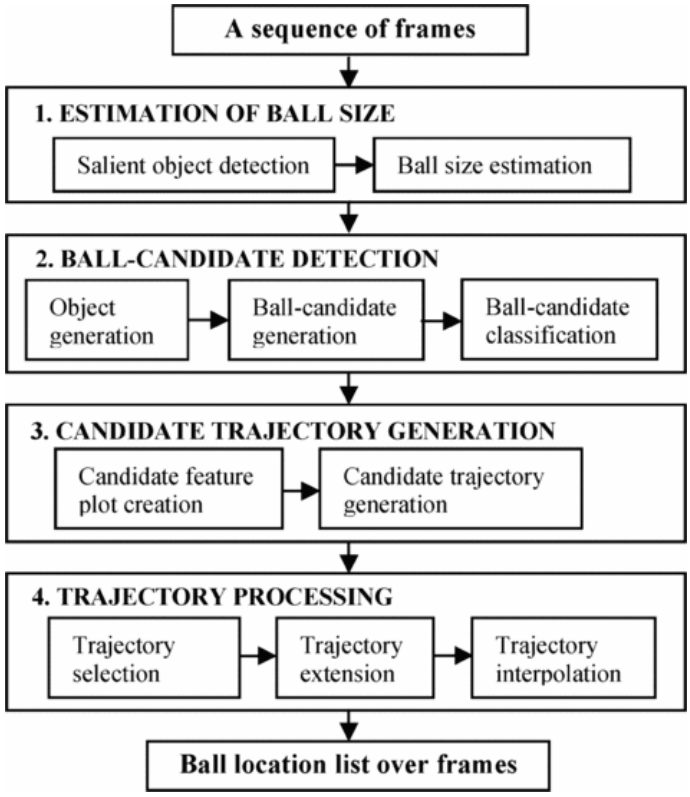


Figura 5.1. Block diagram of the trajectory-based ball system [1].

perspective, not considering a static camera as in our context. For this reason, our proposal is based on the same principle to estimate the ball size, but we implement a different size approach, as is described in Section 4.7.2.

5.1.2 Ball candidate detection

The ball candidate detection of this approach is similar to our solution. They use a set of features to classify ball candidates:

1. Bounding Box size estimation- Described in Section 4.3.1.
2. Eccentricity. Described in Section 4.3.1.
3. Yu - colour representation. Described in Section 4.3.2.
4. Yu - circularity. Described in Section 4.3.3.

They do not specify the strategy utilised to classify the ball. In Section 4.3.3, we analyse these features with other state-of-the-art approaches to get a better classifier in complex situations.

5.1.3 Candidate trajectory detection and update

This phase focuses on the trajectory behaviour of each candidate, managing the information of the ball candidates over a sequence of frames, with the purpose of finding a relation between these objects in different frames, grouping them according to a dynamics model which represents their movement. The trajectory generation approach is described in detail in [40]. A diagram of trajectory generation is schematised in Fig. 5.2. For each ball candidate, the

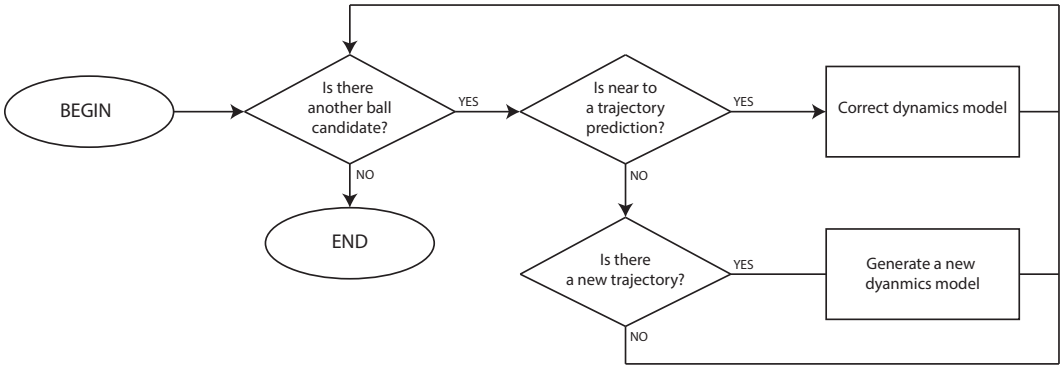


Figura 5.2. Block diagram of the trajectory generation process.

system first determines if the ball candidate location is near to a predicted point of the dynamics model of a previously generated trajectory. In the case that the ball candidate is near to the position predicted by one of the previously calculated dynamics models, the ball candidate is associated with the model which generated the nearest prediction, and then, the candidate location is utilised to update the dynamics model. In the other hand, if there is no near predicted position, the system checks if the candidate location is near to another ball candidate found in the previous frame. In this case, the system creates a new trajectory and initialises a new dynamics model, using the position of the candidate in the previous frame and the current position of the ball candidate.

After analyse each ball candidate, the system has to check the case that a trajectory cannot be associated with a ball candidate. In this case its dynamics

model just updates the position using the predicted location and increases a counter Z_{TET} in order to determine how many consecutive predictions are used by the dynamics model. If this counter is higher than a threshold $N_{CUP} = 6$, the trajectory is maintained but not analysed in posterior times (set as sleep).

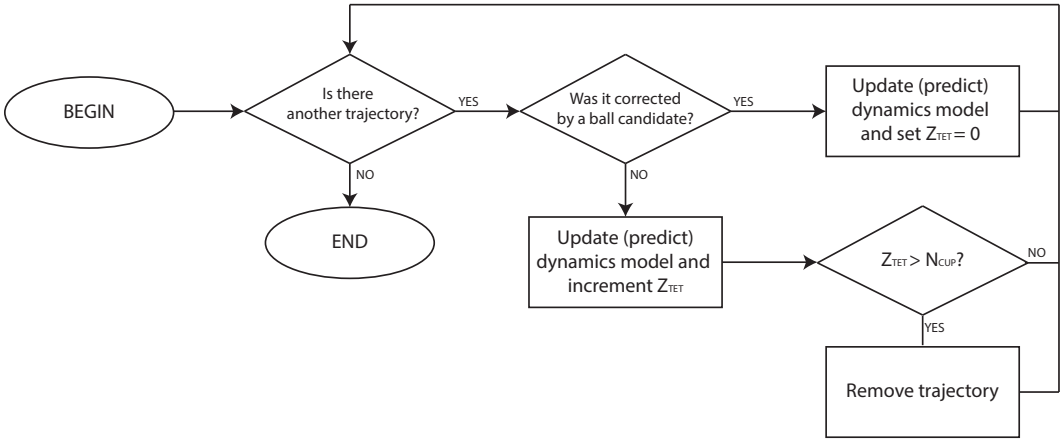


Figura 5.3. Flowchart of the trajectory update process.

Kalman filter formulation

The dynamics of each ball candidate can be modelled using the basic Kalman filter formulation. This approach requires to track the ball when it is not interacting with players or other objects. Therefore, Kalman filter is easily modelled for a ball moving free of interactions, as the ball presents a continuous trajectory without abrupt changes.

In Kalman filter, the state is described by the following linear equation:

$$\mathsf{X}_{\mathsf{k+1}} = \mathsf{A} \cdot \mathsf{X}_{\mathsf{k}} + \mathsf{w}_{\mathsf{k}}, \tag{5.1.1}$$

where X_{k} is the state vector at time k, w_{k} is the system noise and A is the state transition matrix. We define the state vector X_{k} as:

$$\mathsf{X}_{\mathsf{k}} = \begin{bmatrix} x_k \\ y_k \\ v_k^{(x)} \\ v_k^{(y)} \end{bmatrix}, \tag{5.1.2}$$

where (x_k, y_k) is the center point, and $(v_k^{(x)}, v_k^{(y)})$ is the velocity of the ball. The state transition matrix is defined as:

$$\mathsf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1.3}$$

This means that the dynamics model of the ball is assuming that the ball maintains its velocity when it is moving. This is enough to represent the ball displacement when it is free of interaction, because in our context, the friction of the ball with the grass and the wind is not significant, allowing Kalman filter to correct this error. The system noise matrix is initialised as a diagonal matrix with a expected variance of 1 pixel:

$$\mathsf{w} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1.4}$$

In the same way, the observation (or measurement) is described by the following linear equation.

$$\mathsf{Z_{k+1}} = \mathsf{H} \cdot \mathsf{X_{k+1}} + \mathsf{v_{k+1}}, \tag{5.1.5}$$

where $\mathsf{Z_k}$ is the observation vector at time k, $\mathsf{v_k}$ is the observation noise, and H is the observation matrix. We define the observation vector $\mathsf{Z_k}$ as:

$$\mathsf{Z_k} = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix}, \tag{5.1.6}$$

where $(\hat{x}_k, \hat{y}_k)$ is the measurement of the center of the ball. The observation matrix is set as an identity matrix:

$$\mathsf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0, \end{bmatrix} \tag{5.1.7}$$

The observation noise matrix is initialised as a diagonal matrix with a expected variance of 3 pixels.

$$\mathsf{v} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \tag{5.1.8}$$

Classification of the ball candidates

In this step, each ball candidate is analysed more critically using different properties in order to categorise the reliability of them. These properties are:

1. **Size:** Relation between the size of the ball and the predicted size of the ball. This property is defined as:

$$1 - \beta_{SP} \leq \min(H, W)/h_B(y) \leq 1 + \beta_{SP}, \tag{5.1.9}$$

   where $h_B(y)$ is the predicted size of the ball defined in Section 4.7.2 and β_{SP} is a constant related to the size prediction error.

2. **Min-separation:** The *min-separation* is the minimal distance of a ball candidate from other object (including objects extracted in the ball candidate generation step). This way, the min-separation has to be higher than a threshold T_{ms}.

3. **Yu colour:** Referenced in Section 5.1.2, but utilising a more strict threshold.

4. **Yu circularity:** Referenced in Section 5.1.2, but utilising a more strict threshold.

A ball candidate is said to be in Category 1 if it has all the properties defined above. Those that violated one of these properties are said to be in Category 2. All the other ball-candidates are classified in Category 3.

Candidate feature image

The candidate feature image is defined as *"an image that depicts a combination of candidate features over the frames in the given sequence"*. It is utilised just for visualising the candidate behaviour in a better way. An example of this representation is illustrated in Fig. 5.4. In this image, the x-axis represents the frame number in the video-sequence and the y-axis represents the distance of the centre of the candidate to the origin of the image (located in the top left corner of the image). This way, is easy to appreciate the movement of the candidates because near locations present near distance to the origin. In this representation, each point represents a candidate location in a given frame. Besides, each point is represented in different ways depending on the category of the candidate.

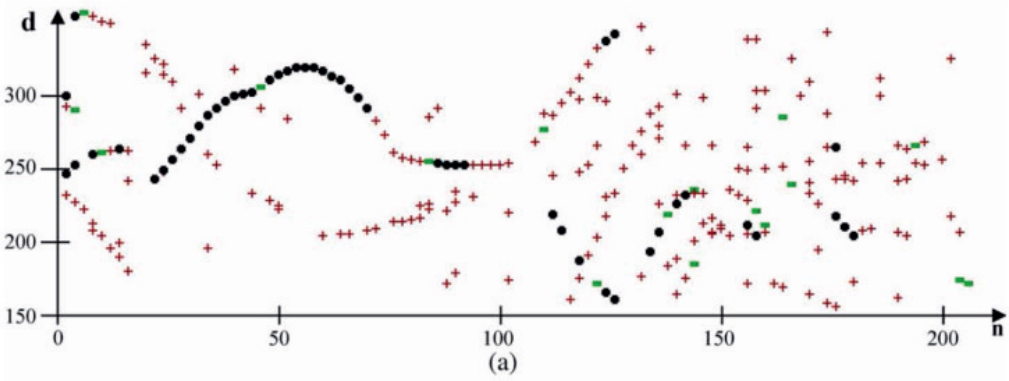


Figura 5.4. Example of a candidate feature image presented in [1]. In red crosses, the ball candidates of category 3. In green circles, the ball candidates of category 2. In black circles, the ball candidates of category 1.

5.1.4 Trajectory processing

Given a set of trajectories $\Gamma(S_k) = T_1, T_2, ..., T_N$ from the video-sequence S_k, where T_i is the trajectory with index i. The next step is to select which is the real ball trajectory. This step requires a certain period of time for analysing the different trajectories in different times in sequence S_k.

Confidence index of a trajectory

Yu *et al.* [1] define a *confidence-index*, denoted by $\Omega(T)$, which measures the confidence associated with a ball trajectory. This confidence index depends on the categorisation of each ball candidate in T_i. Let N_c be the number of candidates in the trajectory T_i, with n_1, n_2 and n_3 the number of candidates of category one, two, and three, respectively. Let $r_1 = n_1/N_c$ and $r_2 = n_2/N_c$ denote the ratio of the candidates in T_i of category one and two, respectively. Then, the confidence index is defined as:

$$\Omega(T) = \sum_{i=1}^{3} \lambda_i n_i + \sum_{j=1}^{2} \mu_j r_j, \tag{5.1.10}$$

where the parameters λ_n and μ_n are utilised to adjust the relative importance of the components.

Trajectory selection and extension

When the *confidence index* of each trajectory is calculated, only the trajectories with the highest values are analysed. Besides, the system identifies the trajectory with the highest value as the main trajectory. Then, other trajectories which are temporally close to the main trajectory are analysed. Then, the best trajectories that continue or precede to the main trajectory are selected, in relation with their confidence index. This process is repeated until all the sequence using the selected trajectories are covered, as depicted in Fig. 5.5. Finally, when each ball trajectory is identified, the algorithm proceeds to

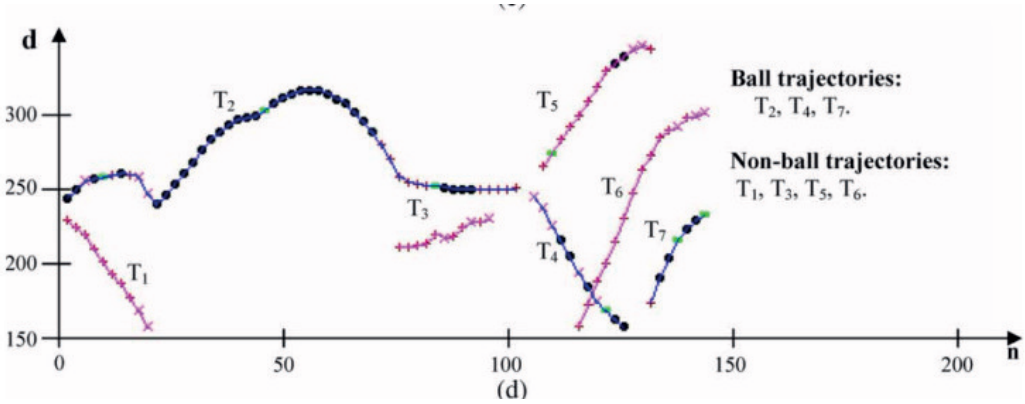


Figura 5.5. Final trajectories, after selecting ball trajectories, presented in [1]. In blue the ball trajectory. In red, the rejected trajectories.

extend each trajectory using its dynamics model, searching for the "missing ball locations" when the ball could not be detected as candidate.

5.2 Our improvement to the trajectory analysis system

We add some improvements to handle the deformation of the ball when high speed is present. These improvements can be summarised as:

- *Elliptical ball representation*, utilised to better represent the distortion of the ball in high speed situations. In [1] they utilised a circular shape representation to extract a circularity measure C (Yu -circulatiry explained in Section 4.3.3), which assumes that the ball will always maintain a

circular shape. This is insufficient to properly characterise the different appearances of the ball, and then it is not appropriate to differentiate a ball object from no-ball objects.

- *Ball pixel size*, calculated by replacing the bounding box size representation with the minimum radius in our ellipse representation. This allows us to improve the appearance categorisation of the ball performed in [1], reducing the error between the calculated and estimated ball size. This representation is described in Section 4.7.2.

- *New confidence index*, which proposes a modification by adding an elliptical confidence parameter $\Phi_e(T)$ to the *confidence index* calculation 5.1.10. This allows us to detect a reliable ball trajectory with a lower number of frames, compared to the method proposed in [1].

We define the elliptical confidence Φ_e for a candidate trajectory T, as defined in Eq. (5.2.1).

$$\Phi_e(T) = \frac{1}{N} \sum_{i=0}^{N} \frac{\gamma_i}{1 + k\alpha}, \tag{5.2.1}$$

where N is the number of ball candidates in the trajectory, $\angle\psi$ is the absolute angle difference between the direction of the distortion $\angle\vec{A}$ and the velocity vector $\angle\vec{V}$, described in Section 4.7.4, k is a weighting factor for the angle difference, and γ_i is defined as:

$$\gamma_i = \begin{cases} 0 & i \notin L \\ 1 & i \in L \end{cases}, \tag{5.2.2}$$

with L the set of ball candidates in T who satisfy the relation presented in Eq. 4.7.23. The reliability flag γ is relevant to determine if the ball presents slow speed or low resolution, as in these cases the deformation will not be prominent, making the determination of the principal component unreliable.

Our system candidate feature image

In Fig. 5.6 we show our visual representation the candidate feature image with some modification. As we observe there are few possible ball trajectories (represented in different colour). However, the more reliable categorisation

of ball candidates (showed as circled points) is more present in the real ball trajectories, in comparison to the other trajectories.

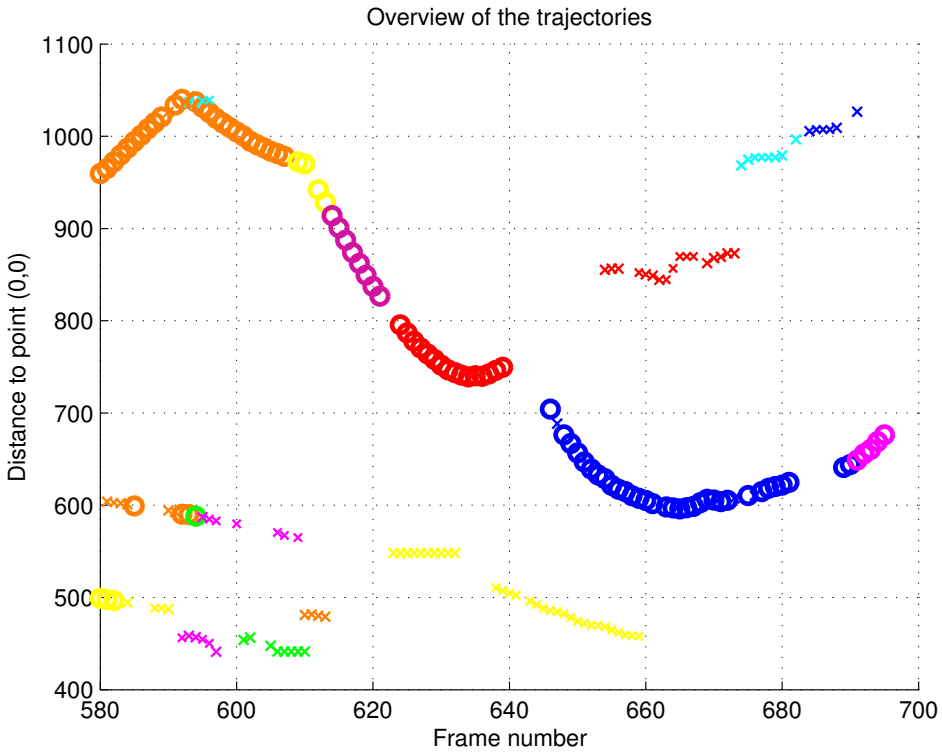


Figura 5.6. Overview of the trajectory selection phase of our method.

Chapter 6

BALL TRACKING IN OCCLUSION SITUATIONS

In several previous proposals, issues related to ball occlusion were commonly treated as a single image problem, without considering the dynamics of the ball or the distortion due to its movement. Moreover, these proposals are not applicable when the ball overlaps with players or field lines. For this reason, it is necessary to add a more robust method to detect the ball in all these situations.

(a) Ball small, distorted and blurred. (b) Ball partially occluded. (c) Ball merged with player feet. (d) Ball merged with shadow.

Figura 6.1. Examples of ball detection issues.

In contrast to other proposals, which focus on ball detection in difficult cases, we observe that the ball does not present a perfect circular shape, because the ball is normally distorted when a player kicks it, or when it is partially occluded, merged, or shadows are present (see Fig. 6.1). Besides, it is difficult to get a reliable shape parametrisation when the ball is observed in a low number of pixels.

For these reasons, it is important to consider the previous information related to the ball in these cases, for predicting the current state of the ball when the current information is ambiguous. Therefore, it is possible to utilise the *appearance information* and the *dynamics information* of the ball, and its *similarity* to other no-ball objects in the previous frame, to estimate the most reliable location of the ball.

In our occlusion management system, this information is combined in three main phases, as is schematised in Fig 6.2, generating different reliability maps (RM), which represent the confidence of the ball position for each criterion. Then, these reliability maps are joined into a single ball reliability map, where the points with more reliability represent the best possible positions of the ball.

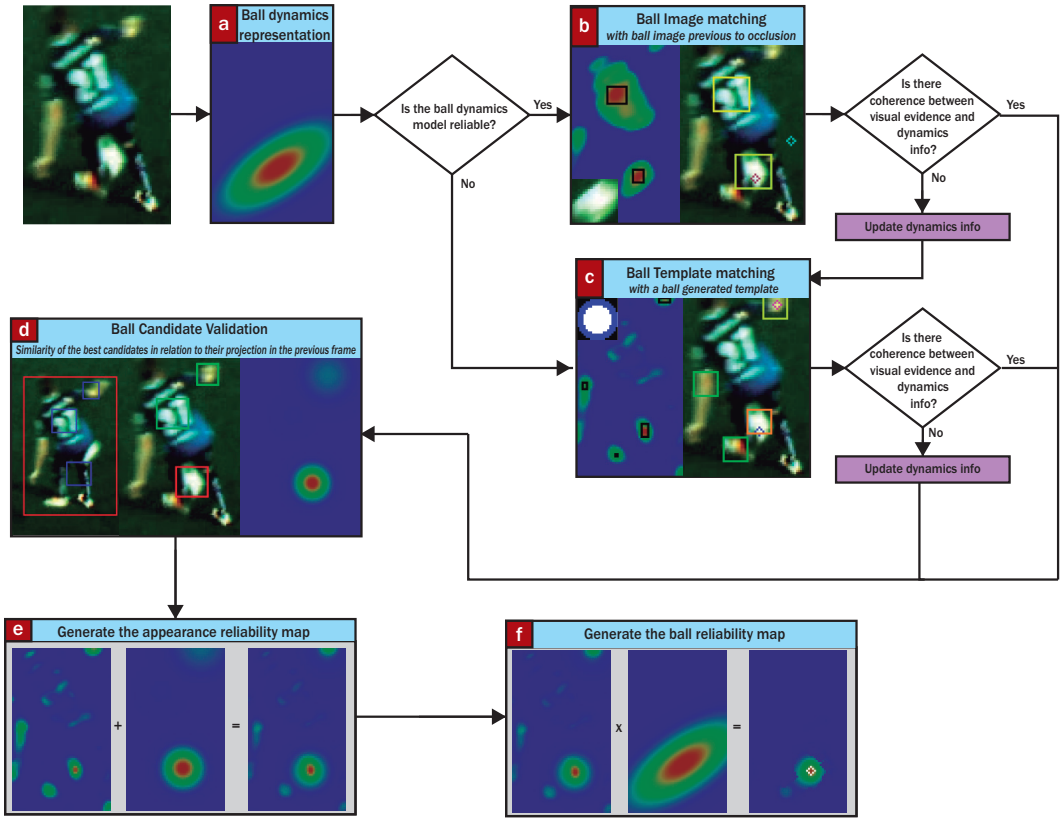


Figura 6.2. Diagram of the proposed occlusion management system. Processes **(a)**,**(b)**, and **(c)** are related to phase one. Process **(d)** is related to phase two. Processes **(e)** and **(f)** are related to phase three.

In phase one, *the dynamics RM* (a) is generated using the dynamics model and, depending on its behaviour, one of two possible templates is utilised to generate a new *Matching RM* (b or c). In phase two, a new *validation RM* (d) is generated in relation with a validation process, which compares the most reliable regions obtained from the *appearance and dynamics RMs*, with other regions previously detected as no-ball objects. The *validation RM* is utilised to decrease the confidence of locations that are similar to no-ball regions. In phase 3, the *appearance RM*(e) is generated, joining the *matching and validation RMs*. Finally, the *ball RM* (f) is generated joining the *appearance and dynamics RMs*.

6.1 First phase

6.1.1 Dynamic reliability map generation

In the first phase, the most likely position of the ball in the current frame is estimated using the previously acquired knowledge of the ball movement obtained by the dynamics model.

We represent the predicted position of the ball using a Gaussian distribution (as shown in process *a* in Fig. 6.2), where the Gaussian mean is located using the predicted position X_{k+1} and its covariance is obtained from the parameters of the *posteriori error covariance matrix* P_k of the Kalman model. With this information, we can find the current occlusion region which involves all objects which are close to the predicted position of the ball using the dynamics model.

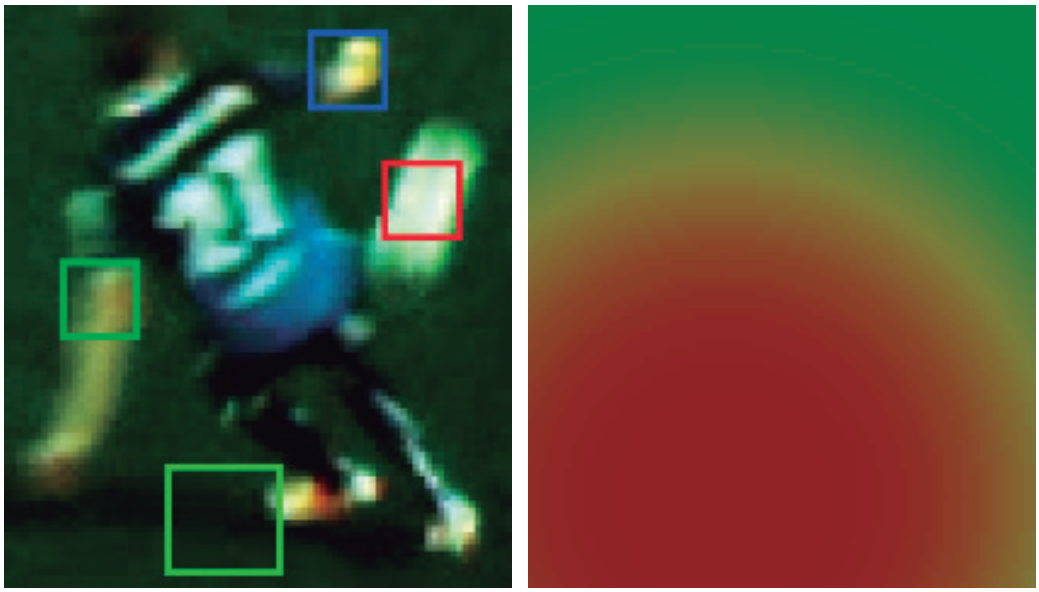
Nevertheless, the ball could present abrupt changes in its movement when it is interacting with other moving objects. For this reason, it is necessary that the process provides a mechanism for coping with these abrupt changes in the dynamics.

To solve this problem, the dynamics model is constantly re-evaluated in order to update the covariance utilised in the reliability map, increasing its parameters in case that the visual evidence supports a change in the ball movement. An initial evaluation is performed utilising the visual evidence obtained in the segmentation phase (Section 3.1.1).

If the predicted position of the ball is not inside the foreground area, we increase the covariance of the dynamics model in order to decrease the impact

of the ball movement in the decision of the real ball position.

This effect is illustrated in Fig. 6.1 (b), showing a frame where the player kicks the ball, changing its normal trajectory.



(a) Next frame after the player kicks the ball. The bottom rectangle is the predicted ball location. The red rectangle is the real ball location.

(b) Dynamics RM updated after using the foreground image.

Figura 6.3. Example of the case where the ball dynamics is unreliable.

6.1.2 Matching reliability map generation

Once the dynamics RM is generated, depending on its behaviour (if it was updated or not), one of two different matching methodologies are activated, as depicted in the process (b) or (c) of the diagram in Fig. 6.2.

In the case when the predicted ball position is coherent with the foreground result, the matching phase is performed utilising the ball image previous to the occlusion, as a template (process (b)). This way, the shape of the ball would not be compromised as long as it does not have interaction with other moving objects. On the other hand, when the dynamics model is not reliable, the matching phase is performed utilising a generated template, consisting in the ball representation without distortion ((process (c)). Preliminary tests have

shown a better performance when this template is utilised in cases when the distortion of the ball cannot be predicted.

Image matching

In the image matching phase (process (b)), the matching score R is calculated using a normalised matching difference relation, defined as:

$$R(x,y) = 1 - \frac{\sum_{i,j}(T(i,j) - I(x+i, y+j))}{\sqrt{\sum_{i,j}(T(i,j)^2 \cdot I(x+i, y+j)^2)}}, \tag{6.1.1}$$

where $R(x,y)$ is the matching coefficient in the pixel (x,y) between the ball image template T, and the current occlusion region I. Using this coefficient, a matching reliability map is generated.

Detecting most reliable regions

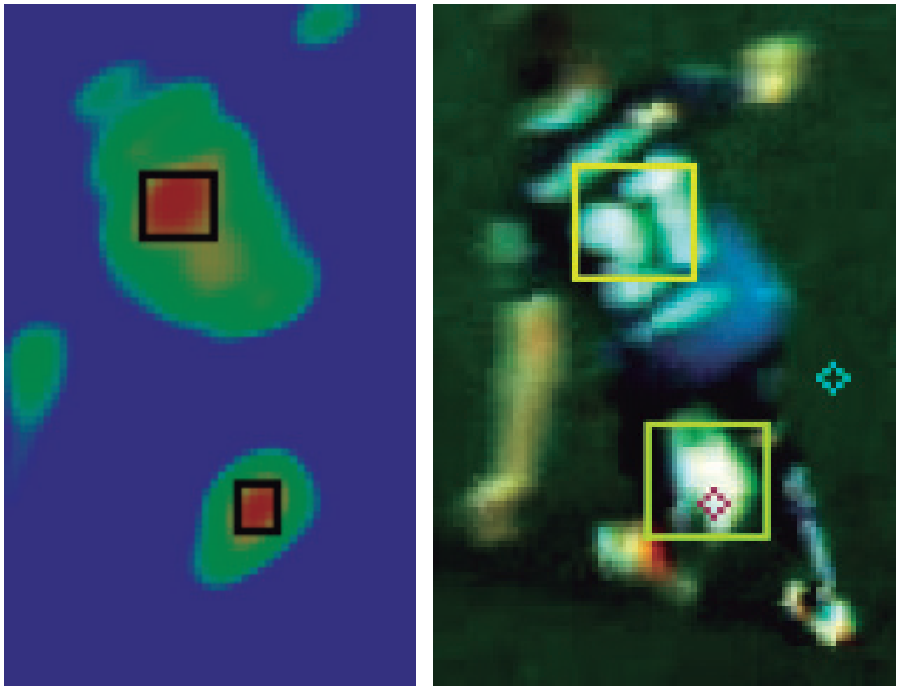
Once the matching RM is obtained, a search for the max-matching regions is performed, generating candidates in the most reliable zones, an example is depicted in Figure 6.4.

The max-matching regions are located using a threshold value determined in relation with the distribution of the value of the pixels in the reliability map. Using the cumulative histogram of the reliability map, the threshold value is determined at least selecting the 20% of the highest values. This way, this system can detect reliable regions in any situation, considering cases when the matching scores are low because the region which contains the ball is significantly different to the ball template (due to partial occlusion or merging).

Re-evaluation of the dynamic reliability map

The coherence between the image matching phase result and the predicted position of the ball is then evaluated. If there is no a reliable candidate near the predicted point, the ball dynamics covariance is increased. Then, the algorithm forces to set the predicted position of the dynamic model as a candidate to the ball location. This incorporation is utilise to considering the case of partial occlusion, as the appearance reliability could increase in a post validation of the ball in the next phase of the process, explained in Section 6.2.

The dynamics RM covariance is updated in relation with the candidate behaviour. For example, in the case of a candidate with high matching coeffi-



(a) Ball matching reliability map. In black rectangle. Most reliable regions.

(b) Most reliable regions of reliability Map

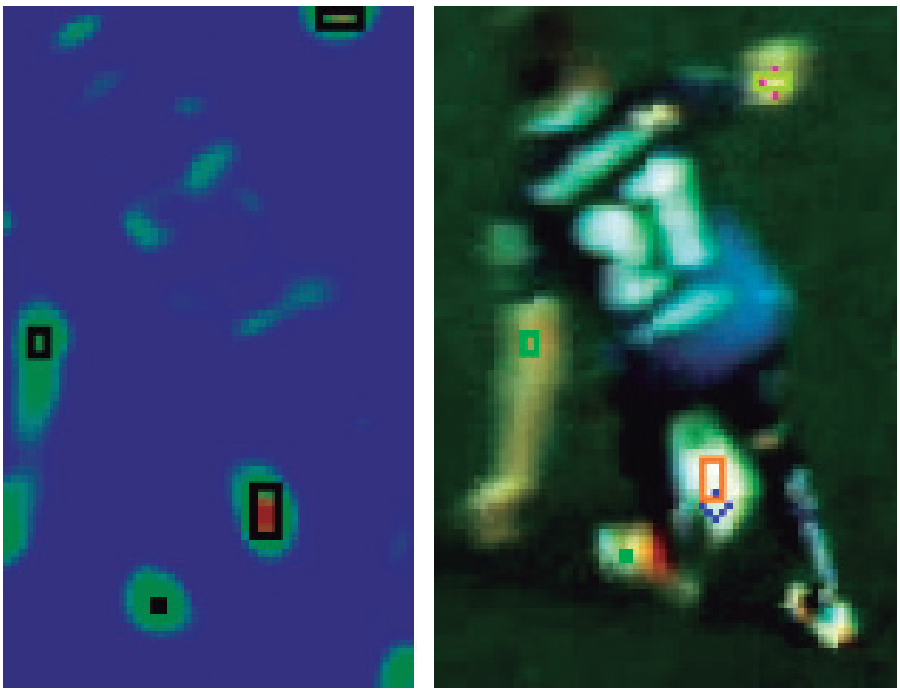
Figura 6.4. Matching RM using ball template with more reliable regions.

cient is far to the predicted point, a high increment of the dynamics covariance is applied, because this could evidence a change of the ball velocity. In the other hand, if there is no reliable and near candidate to the prediction, a low increment of the covariance is applied, because it could mean that the ball is occluded and not really changing its dynamics.

Template matching

Then, in the template matching phase (process (c)), a template of the ball without distortion is generated, in order to detect the most significant candidate ball position. The main difference of this phase with respect to the image matching phase is that the matching results get a higher matching score when the ball presents an unpredicted distortion. For this purpose, a matching coefficient is calculated over the occlusion region using a ball template generated without distortion, based on the "separability filter" proposed in [6]. An

example is depicted in Fig. 6.5.



(a) Generate template matching RM. In black rectangle the most reliable regions.

(b) Most reliable regions extended to the current image.

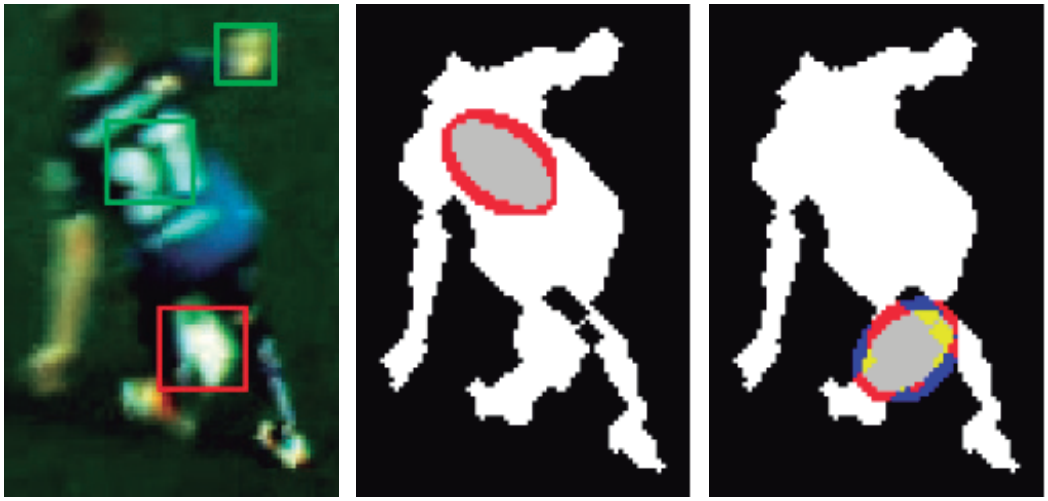
Figura 6.5. Matching RM using generated template with more reliable regions.

Then, the most significant candidate positions are selected, generating for each of them a distorted template considering the potential distortion occurring if the ball is located in the corresponding candidate.

Similarly to the image matching phase, the most reliable candidates are utilised to update the dynamics model, if a no reliable candidate is found near the predicted position.

6.1.3 Interpretation of the matching results

It is possible to find a relation between the state of the ball and the matching coefficient. For instance, when the ball is over a player and not visually merged with a similar object, the ball can be found in the max-matching position;



(a) Current image with most reliable regions.

(b) Generated template in the player shirt.

(c) Generated template in correct ball location.

Figura 6.6. Example of template generation.

when the ball changes drastically in appearance or it is partially occluded, the ball is not necessarily in the max-matching region, but it should be present in some regions with significant reliability; in other case, when the ball is totally occluded it should be not present in a reliable zone.

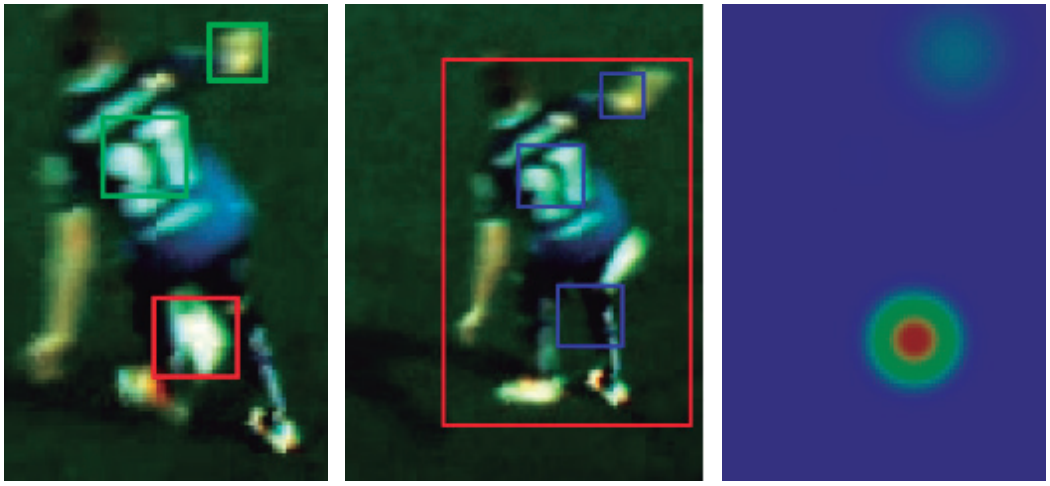
6.2 Second phase

Once the phase one select the most reliable zones as ball candidates, each candidate is analysed in a validation phase (process (d)), differentiating no-ball candidates from the previous frame which are similar to the ball, as socks and shoes with similar color.

6.2.1 Validation reliability map generation

In this phase, the current occlusion region is utilised to match over a region of the previous frame with the same center position, but considering a bigger area. This consideration is based on the assumption that the ball is moving faster that the player, and the player maintains a similar posture frame to frame. Then, the most similar region from the previous frame to the current occlusion region is obtained. The occlusion projection is defined as the result of this

matching, and it is represented with the red rectangle in Fig. 6.7 (b). Next, the projection of each ball candidate is obtained, using the same relative position with respect to the occlusion projection, represented with blue rectangles in Fig. 6.7 (b). Finally, each ball candidate and its projection are compared to generate a validation reliability map as is showed in Fig 6.7 (c).



(a) Most reliable region in the current frame.

(b) Projection of each region in the previous frame.

(c) Validation reliability map.

Figura 6.7. Validation process images.

Similarly, we utilise a matching score equivalent to the score presented in Equation (6.1.1) to get the similarity between the candidate ball region and its projection to the previous frame. This similarity score for each candidate c is referred as S_c.

The validation reliability map is generated using the similarity score, representing each candidate as a gaussian distribution with its mean value in the center position of the candidate and its covariance matrix is considered as a diagonal matrix with variance proportional to the size of the candidate. The peak value of each gaussian R_c depends on the similarity score and in the ball state assumption presented in Table 6.1.

6.2.2 Interpretation of the validation results

This additional ball state assumption is added in order to deal with the case when the ball is not moving. It is considered that the ball is not moving

Max-Gaussian Value	Condition	Candidate assumed state
$R_c = 1$ - S_c	The projection is not near to previous ball position	The ball is moving or changes its velocity direction
S_c	The projection is near to previous ball position	The player stopped the ball

Table 6.1. Gaussian peak values, assumptions and interpretations.

if the position of the projection of the candidate is located in the previous ball position. This way, each no ball candidate will have a low score if its appearance is highly similar to its projection, except for the case when the ball is not moving.

For instance, in Fig. 6.7 (a), the real ball, an arm of the player and the number in his uniform are detected as similar ball regions in the matching phase. Assuming that the ball is moving, the no-ball candidates look similar to their projection in 6.7 (b). This means that these candidates do not really move in relation to the player, and with the previous ball information it can determined that its projection is not near to the previous ball location, so it cannot be considered that the ball is not moving. In the other hand, the real ball candidate projection looks different to its current region image, so we can assume that the ball is moving.

6.3 Third phase

In this phase, each reliability map generated in previous phases is combined in order to generate the final reliability map which represents the ball location.

6.3.1 Appearance reliability map generation

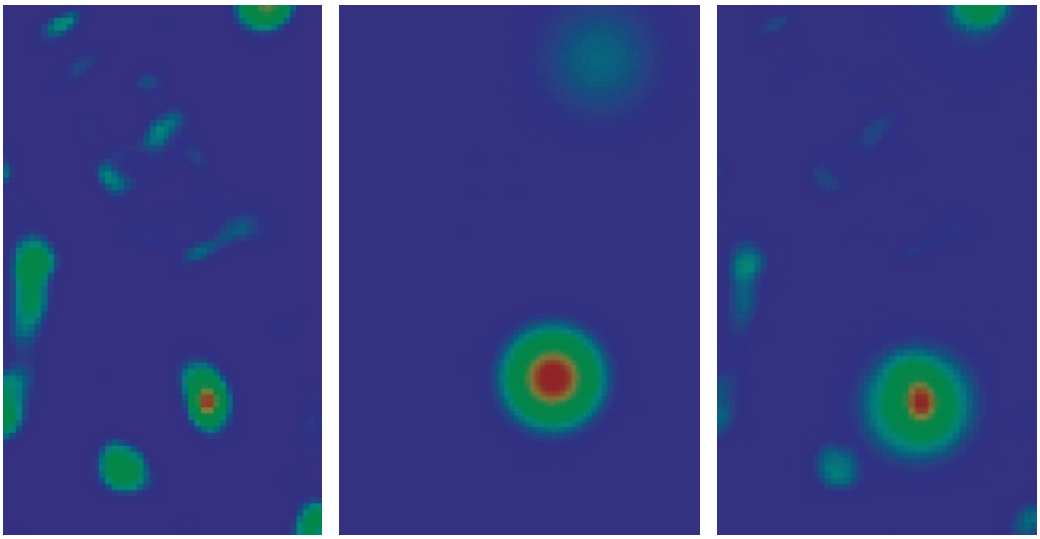
Finally, the matching and validation reliability maps are combined in order to get a more accurate representation of where the ball could be present, based mainly on the visual information.

Each pixel of the *matching RM* ($M_{RM}(x, y)$) and the *validation RM* ($V_{RM}(x, y)$) are averaged to generate the *appearance RM* ($A_{RM}(x, y)$). This operation is

performed as:

$$A_{RM}(x,y) = \frac{1}{2}M_{RM}(x,y) + \frac{1}{2}V_{RM}(x,y) \tag{6.3.1}$$

The average is performed to maintain the scale of the *appearance RM* between 0 and 1, as both utilise the same score described in Eq. (6.1.1) to calculate the pixel values. An example of A_{RM} is shown in Fig. 6.8.



(a) Matching reliability map. (b) Validation reliability map (c) Appearance reliability map

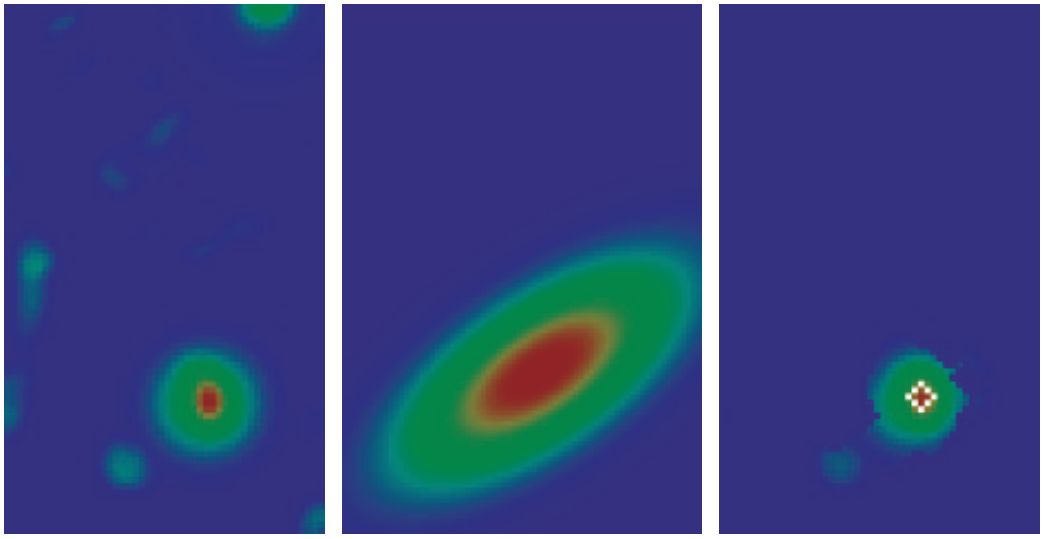
Figura 6.8. Appearance reliability map generation.

6.3.2 Ball reliability map generation

Finally, the *appearance RM* (A_{RM}) is combined with the dynamics model, represented by the *dynamics RM* ($D_{RM}(x,y)$). Both reliability maps are multiplied in order to generate the final reliability of the ball position, denoted as *ball RM* (B_{RM}). Additionally, we multiply the ball RM with the foreground image ($FG(x,y)$) with the purpose of considering the position of the foreground pixels, because the ball cannot be totally occluded by the background (grass). This operation is performed as:

$$B_RM(x,y) = A_{RM}(x,y) \cdot D_{RM}(x,y) \cdot FG_{RM}(x,y) \tag{6.3.2}$$

Then, the most reliable point is selected as the best position of the ball, and its value in the ball reliability map represents the certainty of its position.



(a) Appearance reliability maps.

(b) Dynamics reliability maps.

(c) Final ball reliability map. In a white circle, the most reliable ball location.

Figura 6.9. Ball reliability map generation.

Chapter 7

EXPERIMENTAL RESULTS

This chapter describes the final tests to our ball tracking methodology.

7.1 Dataset

To test the performance of our approach, we utilise a data-set captured at Alfheim Stadium in Norway, presented in [41]. This data-set was obtained using three static cameras (Basler acA1300-30gc) to cover all the field. An example of the resulting images of each camera is depicted in Fig. 7.1.

(a) CAM0: Left camera. (b) CAM1: Central camera. (c) CAM2: Right camera.

Figura 7.1. The views from the wide-angle cameras in Alfheim Stadium.

All the videos are H.264 encoded using the libx264 software. The generated files encode a sequence number and also incorporate a timestamp (in nanoseconds) of the first frame in the sequence, allowing the synchronisation between the cameras.

These cameras were previously calibrated for correcting the distortion created by the wide-angle lens. More characteristics of these cameras are described in Table 7.1.

Camera	Basler acA1300-30gc
Resolution	1280x960
MP Resolution	1.3 MP
Frame rate	30 fps
Lens model	3.5 mm Kowa-LM4NCL
Sensor model	Sony ICX445
Shutter mechanism	global shutter

Table 7.1. Properties of the cameras utilised for obtaining the Alfheim dataset [41].

7.2 Performance of our implementation

The accuracy of our algorithm compared to the trajectory-based system (TBS) [1], described in Section 5.1, is presented in Table 7.2. Both, our approach and TBS were implemented in C++.

Sequence	Ground-truth		Accuracy of TBS approach			Accuracy of our approach					
	#frm	#oc-frm	accuracy	free_acc	occ_acc	d_{coef}	#correct	#false	accuracy	free_acc	occ_acc
CAM1-28-61	34	9	23.53%	20.00%	33.33%	88.85%	31	3	91.18%	92.00%	88.89%
CAM1-570-714	145	38	78.62%	98.13%	23.68%	80.16%	142	3	97.93%	100%	92.11%
CAM1-1976-2028	53	7	85.47%	92.60%	48.57%	81.21%	53	0	100%	100%	100%

Table 7.2. Accuracy of our algorithm using different sequences.

7.2.1 Performance parameters

We utilise the dice coefficient (d_{coef}) proposed in [42] to evaluate the overlapping area between the ground-truth and the resulting regions obtained by our algorithm. This coefficient is calculated as:

$$d_{coef} = \frac{2 \cdot |GT \cap REG|}{|GT| + |REG|}, \tag{7.2.1}$$

where $|GT|$ and $|REG|$ are the area of the ground-truth and of the obtained region, respectively, and $|GT \cap REG|$ is the area of the intersection of both regions.

We set a successful detection of the ball when d_{coef} is over 60%. We utilise a low value of this coefficient because it tends to be low when the object uses

few pixels in its representation, as one pixel becomes significant in such a little area.

We also evaluate the number of frames where the ball was correctly tracked (#correct), and the false detections (#false).

We obtain the accuracy on ball tracking considering three measurements:

- The performance in the complete sequence (*accuracy*), where #frm is the number of frames in the sequence.
- The performance in occlusion situations (*oc_accuracy*), where #oc-frm is the number of frames where the ball is occluded, as referred in Figure 7.2.
- The performance in isolated ball situations ($free\_accuracy$).

7.3 Results in difficult situations

In Figure 7.2 we can observe the candidate result score in different cases utilising the occlusion management system proposed in this thesis.

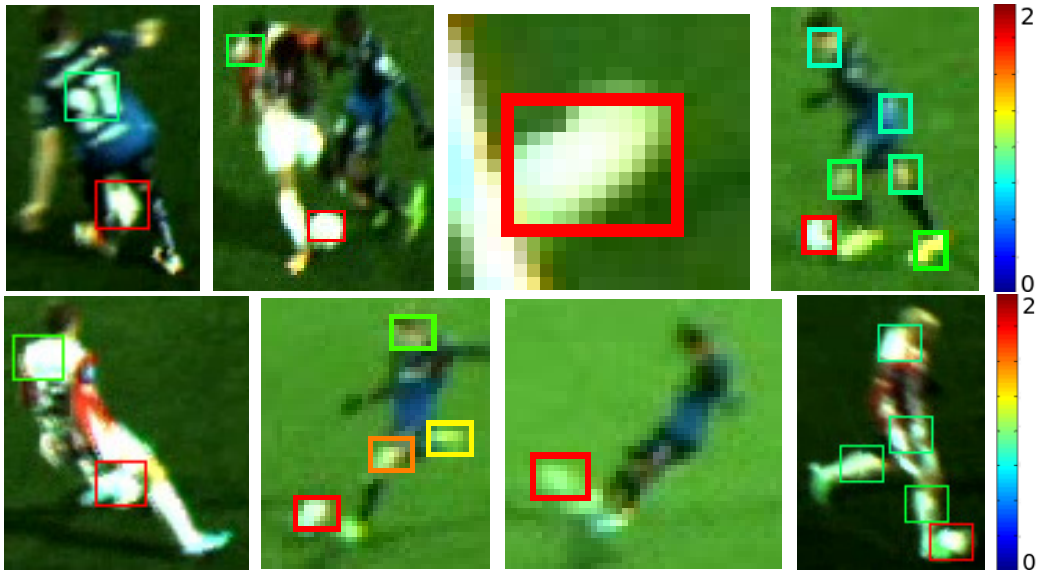


Figura 7.2. Algorithm results for several difficult cases, including occlusion, camouflage, low resolution, deformation due to speed, and illumination problems.

7.4 Reliability of the generated trajectories

In Fig. 7.3, a comparison between the reliability score of the different trajectories (represented in different colours), using elliptical features (our approach) and without them (TBS approach) can be observed. The real ball candidates are represented in squares and no-ball trajectories in crosses. For a fair comparison, in this analysis, we disable our occlusion management system and our coarse-to-fine filter for ball candidates detection. The reliability (Rel) of a trajectory T is calculated as:

$$Rel(T) = \frac{\Omega(T)}{|T|}, \tag{7.4.1}$$

where $\Omega(T)$ is the confidence index of the trajectory and —T— is the number of frames of the trajectory.

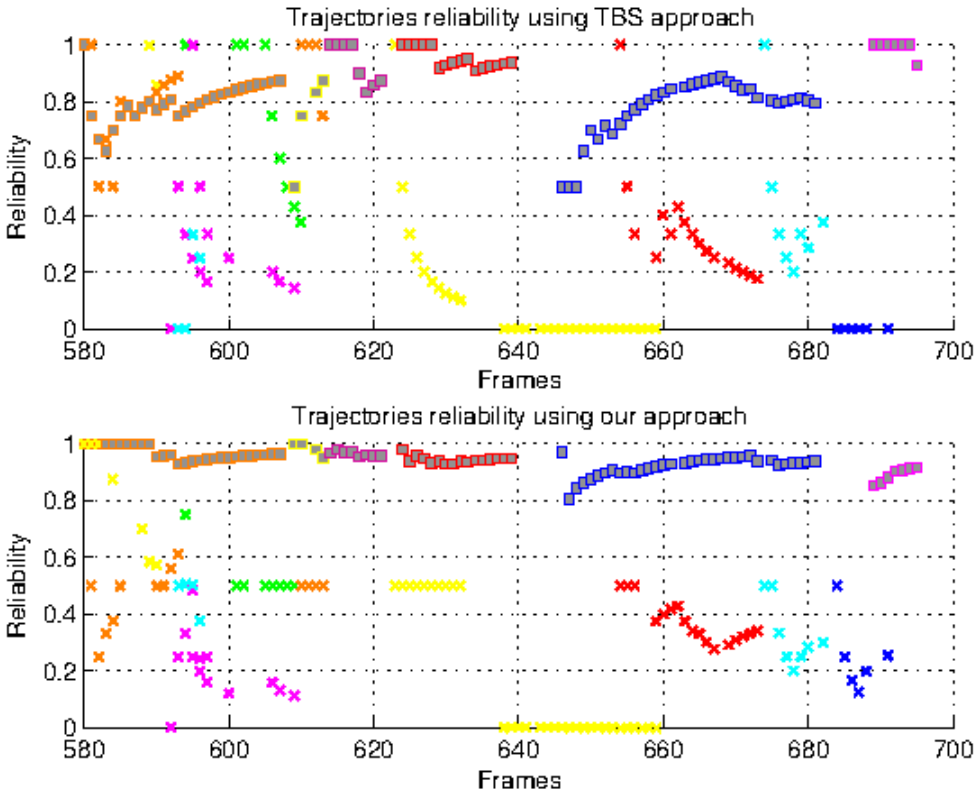


Figura 7.3. Trajectory reliability in sequence CAM1-570-714. In squares the real ball trajectories.

7.5 Computing time performance

The Table 7.3 summaries the computing time performance of our implementation.

Module	Mean[s]	sigma[s]	min[s]	max[s]
Acquisition	0.0234	0.007	0.0150	0.0790
Segmentation	0.1095	0.0122	0.2050	4.3790
Connected components	0.0077	0.0008	0.0060	0.0100
Blob filtering	0.0005	0.0.0005	0.0000	0.0200
Tracking	0.0135	0.0118	0.0050	0.5400
Trajectory updating	0.0008	0.0005	0.0000	0.0020

Table 7.3. Properties of the cameras.

We implement six module to completely track the ball. These modules are:

1. **Acquisition module:** It performs the acquisition of image frames from a video sequence.

2. **Segmentation module:** It performs the foreground segmentation of the image. This process is detailed in Section 3.1.1.

3. **Connected components module:** It connects each region of the foreground using run-length encoding methodology [37].

4. **Blob filtering module:** It discards extremely large and small regions of the image.

5. **Ball tracking module:** It performs the proposed ball tracking approach.

6. **Tracking module:** It performs the reliability tracking approach presented in [4], which utilises our ball approach.

7. **Trajectory update module:** It updates each trajectory information with the current properties of each candidate.

7.6 Discussion

7.6.1 General performance comparison

In comparison to TBS approach, our method obtains a substantially better performance, mainly in occlusion situations. The sequence CAM1-28-61 presents extreme distortion of the ball due to speed. This causes that TBS is not able to reach a reliable score in the ball trajectory, because in this sequence the real ball trajectories are not able to attain the minimum number of frames required to validate a reliable trajectory. In sequence CAM1-570-714 the ball presents several occlusion cases and abrupt changes in its displacement. Also, the isolated ball trajectories require a longer number of frames. For this reason, the difference in accuracy is noticeable in occlusion cases.

The main difference in isolated cases, is that our algorithm required a lower number of frames to detect the ball in any case, and also our approach allows to detect the ball immediately after the occlusion, because it does not lose the ball in these cases. For this reason we do not need to validate a post-occlusion trajectory to detect the ball. In occlusion cases, TBS can detect the ball at the beginning of the occlusion situation, but when the ball velocity vector changes drastically, the location of the ball fails. Our method focus on these cases in deep, getting a better performance.

7.6.2 Ball detection comparison

Using our modification in the trajectory analysis system it is possible to obtain a better reliability of the real ball trajectories. In the presented results, we can observe that in the ball trajectories between frames 580 and 615 (orange rectangles), and between 643 and 681 (blue rectangles), our approach is capable of initially getting a higher reliability of the ball trajectory. This is because our modification considers the cases when the ball is distorted, improving the categorisation of the ball, and in consequence the confidence index. This way, our approach requires a lower number of frames to validate the real ball trajectory, in comparison to TBS. Besides, our ball candidate generation approach is capable of filtering 74.32% more candidates than TBS in sequence CAM1-570-714, and of reducing the mean square error of the ball size prediction from 2.873 pixels to 0.9354 pixels.

7.6.3 Time performance

The time performance of the complete system is slow in relation with real-time performance, presenting an average time processing of 0.2118[s] per frame, equivalent to 4,7214[Hz]. However, if we only consider our ball tracking approach, discarding the acquisition, segmentation and connected component modules, the time performance increases drastically to 0.022[s], equivalent to 45.4545[Hz]. The segmentation process is the bottleneck in performance.

Chapter 8

CONCLUSIONS

Prior work on tracking systems to record the soccer ball position on the field, can be found in the literature. Most of these works rely on two different approaches. The first approach is oriented to detect the ball in the current frame using different appearance criteria, and the second one focuses on predicting the position of the ball in the future. Yu *et al.* [1], for example, report that the ball detection is improved studying the behaviour of ball candidates instead of a direct selection, and utilising their trajectories to assist the detection of the ball in occlusion situations.

However, these approaches do not focus on bad ball quality cases and tend to fail when the ball interacts with players, because it presents unpredictable movement.

In this thesis, we have presented a novel ball tracking algorithm able to cope with difficult situations, which in several cases are still open issues in computer vision. To the best of our knowledge, we have evaluated and selected the best ball feature approaches from the state-of-the-art to improve the ball candidate generation, considering complex situations, as distortions and small ball size. The main technical contributions of our approach are: the elliptical representation of the ball that allows us to improve the detection of the ball in cases of distortion due to high speed; and a new occlusion management approach, able to cope with severe partial occlusion situations.

We have tested our algorithm in real soccer competition sequences including complex object interaction situations, comparing our algorithm with a state-of-the-art approach (TBS [1]). We have outperformed this method in both free displacement and occlusion situations with a notorious difference in the later. We prove that, the more complex a situation is, the higher the accuracy is, in comparison with TBS. Besides, we have also demonstrated that

it is possible to detect the ball using a lower number of frame. We have also demonstrated that, in case of occlusion, it is also possible to detect the ball in several complex situations, such as when the ball is totally or partially occluded, when the ball merges with a field line or with players, even with similar colour, or when the ball is simultaneously occluded and distorted.

These modifications enable operators to reduce their manual corrections when they are recording the ball position, reducing in this way the time of the gathering process. Therefore, technical staff is capable of utilising the statistical information and take decisions faster.

The proposed ball detection and tracking approach performs with a low computational cost, not requiring an expensive hardware to achieve real-time. In contrast to other vision systems, our solution was validated not considering an ideal context (high quality cameras in a stadium with good illumination), reducing the technology cost.

This thesis therefore indicates that our system takes advantage of some ball characteristics that generate problems to conventional methods (described in Section 1.3). For example, the small size of the ball that complicates the detection of patterns in the ball may also unify the colour of the ball into one, simplifying its colour representation. Another example is the distortion of the ball, which decreases the accuracy of shape features but also supports the reliability of a trajectory containing the real ball, because it is possible to associate the ball distortion with its movement direction.

Most notably, to our knowledge, this is the first work to investigate and represent the motion blur effect of the ball in real sequences of soccer.

Our results provide evidence of potentially higher performance, in case of implementing this solution using low quality cameras and mid-cost hardware, reducing the technical requirements of a vision system for data gathering.

Nevertheless, this approach can still be improved, especially in detecting the ball when it is camouflaged by other object. Also, the algorithm is still significantly dependant on the correctness of the dynamics model, mostly on situations where a bad estimation can lead to an unrecoverable trajectory. This implies that this approach is still semi-automatic, requiring in this cases one manual correction by an operator, for stopping error propagation. Our complete solution cannot perform in real-time due that the segmentation process is computationally expensive. However, the segmentation process is not in the scope of this thesis and it can be also easily accelerated using GPU

computing for parallel computation.

Future work will focus on testing our approach thoroughly, in order to properly characterise its performance in different situations. It is difficult to track the ball in long occlusion sequences and drastic changes in ball movement. This way, a player model should therefore be incorporated, which could complement contextual information to facilitate the ball detection.

The utilisation of a more complex dynamics model should be studied in more detail, specially in occlusion situations, as in this cases the dynamics of the ball cannot be predict using an unimodal model as the Kalman filter. Alternative tracking solutions should be able to represent multi-modal distributions, as particle filters of MHT. Currently, when the ball drastically changes its dynamics, the covariance matrix parameters of the gaussian model is increased, making the dynamics model to decrease its role in the estimation of the ball location. In the other hand, using multi-modal tracking models the generation of the dynamics RM can be improved, as it is possible to represent both cases (when the ball movement changes or not) in the same distribution with different peak values.

REFERENCIAS

[1] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "Trajectory-based ball detection and tracking in broadcast soccer video," *Multimedia, IEEE Transactions on*, vol. 8, no. 6, pp. 1164–1178, 2006.

[2] M. Hughes and I. M. Franks, *Notational analysis of sport: Systems for better coaching and performance in sport*. Psychology Press, 2004.

[3] S. M. Shukri and M. M. Shaukhi, "A study on multi-agent behavior in a soccer game domain," *World Academy of Science, Engineering and Technology*, vol. 14, pp. 308–312, 2008.

[4] M. D. Zúñiga, F. Brémond, and M. Thonnat, "Real-time reliability measure-driven multi-hypothesis tracking using 2d and 3d features," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 1–21, 2011.

[5] X.-F. Tong, H.-Q. Lu, and Q.-S. Liu, "An effective and fast soccer ball detection and tracking method," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 795–798.

[6] J. Miura, T. Shimawaki, T. Sakiyama, and Y. Shirai, "Ball route estimation under heavy occlusion in broadcast soccer video," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 653–662, 2009.

[7] T. D'Orazio, M. Leo, P. Spagnolo, M. Nitti, N. Mosca, and A. Distante, "A visual system for real time detection of goal events during soccer matches," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 622–632, 2009.

[8] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante, "A new algorithm for ball recognition using circle hough transform and neural classifier," *Pattern recognition*, vol. 37, no. 3, pp. 393–408, 2004.

[9] A. Rehman and T. Saba, "Features extraction for soccer video semantic analysis: current achievements and remaining issues," *Artificial Intelligence Review*, vol. 41, no. 3, pp. 451–461, 2014.

[10] A. Treptow and A. Zell, "Real-time object tracking for soccer-robots without color information," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 41–48, 2004.

[11] S. Mitri, S. Frintrop, K. Pervolz, H. Surmann, and A. Nuchter, "Robust object detection at regions of interest with an application in ball recognition," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.* IEEE, 2005, pp. 125–130.

[12] C. Carling, A. M. Williams, and T. Reilly, *Handbook of soccer match analysis: A systematic approach to improving performance.* Psychology Press, 2005.

[13] A. J. Coutts and R. Duffield, "Validity and reliability of gps devices for measuring movement demands of team sports," *Journal of science and Medicine in Sport*, vol. 13, no. 1, pp. 133–135, 2010.

[14] E. J. Arriaza and M. D. Zuniga, "Soccer as a study case for analytic trends in collective sports training: A survey," *International Journal of Performance Analysis in Sport*, vol. 16, no. 1, pp. 171–190, 2016.

[15] D. S. Valter, C. Adam, M. Barry, and C. Marco, "Validation of prozone®: A new video-based performance analysis system," *International Journal of Performance Analysis in Sport*, vol. 6, no. 1, pp. 108–119, 2006.

[16] P. Brulé, C. Carling, A. David, F. Da Graça, J. Giorgi, D. Villet, and J. Garberino, "Amisco: the development of a computerised match analysis system to automatically track the movements of soccer players," in *Proceedings of the IV World Congress of Notational Analysis of Sport, University of Porto*, 1998, pp. 22–25.

[17] C. Duque, *Priors for the Ball Position in Football Match using Contextual Information*. Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan, 2010.

[18] T. D'Orazio and M. Leo, "A review of vision-based systems for soccer video analysis," *Pattern recognition*, vol. 43, no. 8, pp. 2911–2926, 2010.

[19] J. Ren, J. Orwell, G. A. Jones, and M. Xu, "Tracking the soccer ball using multiple fixed cameras," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 633–642, 2009.

[20] V. Pallavi, J. Mukherjee, A. K. Majumdar, and S. Sural, "Ball detection from broadcast soccer videos using static and dynamic features," *Journal of Visual Communication and Image Representation*, vol. 19, no. 7, pp. 426–436, 2008.

[21] C. Lyu, Y. Liu, B. Li, and H. Chen, "Multi-feature based high-speed ball shape target tracking," in *Information and Automation, 2015 IEEE International Conference on*. IEEE, 2015, pp. 67–72.

[22] J.-Y. Kim and T.-Y. Kim, "Soccer ball tracking using dynamic kalman filter with velocity control," in *Computer Graphics, Imaging and Visualization, 2009. CGIV'09. Sixth International Conference on*. IEEE, 2009, pp. 367–374.

[23] T. Misu, A. Matsui, M. Naemura, M. Fujii, and N. Yagi, "Distributed particle filtering for multiocular soccer-ball tracking," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3. IEEE, 2007, pp. III–937.

[24] D. Liang, Y. Liu, Q. Huang, and W. Gao, "A scheme for ball detection and tracking in broadcast soccer video," in *Pacific-Rim Conference on Multimedia*. Springer, 2005, pp. 864–875.

[25] M. Kia, "Ball automatic detection and tracking in long shot views," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 6, p. 1, 2016.

[26] J. Halbinger and J. Metzler, "Video-based soccer ball detection in difficult situations," in *Sports Science Research and Technology Support*. Springer, 2013, pp. 17–24.

[27] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. 33, no. 2, pp. 260–268, 1961.

[28] K. Choi and Y. Seo, "Probabilistic tracking of the soccer ball," in *Statistical Methods in Video Processing*. Springer, 2004, pp. 50–60.

[29] J. Renno, J. Orwell, and G. A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," in *BMVC*, 2002, pp. 1–10.

[30] J. Liu, P. Carr, R. T. Collins, and Y. Liu, "Tracking sports players with context-conditioned motion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1830–1837.

[31] K. Teachabarikiti, T. H. Chalidabhongse, and A. Thammano, "Players tracking and ball detection for an automatic tennis video annotation," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE, 2010, pp. 2461–2494.

[32] F. Poiesi, F. Daniyal, and A. Cavallaro, "Detector-less ball localization using context and motion flow analysis," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 3913–3916.

[33] X. Wang, V. Ablavsky, H. B. Shitrit, and P. Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play," *Computer Vision and Image Understanding*, vol. 119, pp. 102–115, 2014.

[34] F. Yan, W. Christmas, and J. Kittler, "A maximum a posteriori probability viterbi data association algorithm for ball tracking in sports video," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 279–282.

[35] J. Ren, J. Orwell, G. Jones, M. Xu *et al.*, "Real-time modeling of 3-d soccer ball trajectories from multiple fixed cameras," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 350–362, 2008.

[36] A. Ekin, R. Mehrotra *et al.*, "Automatic soccer video analysis and summarization," *Image Processing, IEEE Transactions on*, vol. 12, no. 7, pp. 796–807, 2003.

[37] B. M. Cahill III and J. R. Hedden, "Manipulation of run-length encoded images," Dec. 20 1988, uS Patent 4,792,981.

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

[39] Y. Liu, D. Liang, Q. Huang, and W. Gao, "Extracting 3d information from broadcast soccer video," *Image and Vision Computing*, vol. 24, no. 10, pp. 1146–1162, 2006.

[40] X. Yu, C. Xu, Q. Tian, and H. W. Leong, "A ball tracking framework for broadcast soccer video," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 2. IEEE, 2003, pp. II–273.

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

[42] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "Etiseo, performance evaluation for video surveillance systems," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 2007, pp. 476–481.