Faculdade de Engenharia da Universidade do Porto

FEUP

Intelligent Tracking of Handball Players at F.C.PORTO and FADEUP

Marco André Ferreira e Silva

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Supervisor: Luís Paulo Reis

July 2011

Resumo

A utilização de novas tecnologias no desporto tem crescido muito nos últimos anos. Os vídeos de Desportos, como importantes conteúdos de multimédia que são, têm recebido cada vez mais atenção devido ao seu potencial comercial e de entretenimento.

Com a introdução de técnicas de análise automática baseada em processamento de imagem, torna-se possível a recolha de dados de jogo de uma forma mais fácil, mais rápida e com um maior rigor, ao invés da obtenção de dados estatísticos de forma esforço manual, que implica muito tempo despendido.

Com a obtenção de dados de jogo, pode-se elaborar um estudo minucioso das equipas, obtendo-se um modelo de jogo característico, sendo depois possível implementar tácticas de forma a contrariar os pontos fortes e explorar os pontos fracos das equipas adversárias, bem como melhorar os aspectos técnicos e tácticos da própria equipa.

Algum trabalho já vem sendo desenvolvido no tracking de jogadores, contudo uma das componentes que ainda está pouco desenvolvida e que merece a nossa atenção é o seguimento da bola. O seguimento da bola permite que na análise ao jogo se saiba todos os percursos da bola, e portanto inferir quais as principais zonas de ataque e posse de bola da equipa adversária; por outro lado podemos ajudar a equipa de arbitragem no esclarecimento de situações de difícil análise, entre as quais, quando a bola sai do terreno de jogo ou quando esta entra na baliza.

Neste Tese, serão utilizados diversos vídeos para o desenvolvimento de 4 métodos. Serão utilizados vídeos obtidos do pavilhão da FADEUP, bem como vídeos de um jogo real, nomeadamente a final da Supertaça Portuguesa 2011.

Abstract

The use of new technologies in sport has grown tremendously in recent years. Sports Videos, as important multimedia content, have received increasing attention because of its commercial and entertainment potential.

With the introduction of automated analysis techniques based on image processing, it becomes possible to collect data for a game easier, faster and with greater accuracy, instead of collecting statistics on a manual effort, which involves much time spent.

With the acquisition of game data, you can prepare a detailed study of the teams, resulting in a characteristic form of the game, after being possible to implement tactics to counter the strengths and exploit the weaknesses of opposing teams, as well as improve the technical and tactical aspects of your own team.

Some work is already being developed in the tracking of players, however one aspect that is still underdeveloped and that deserves our attention is the following of the ball. The tracking of the ball allows the analysis of the game, being possible to extract all the routes of the ball, and therefore infer what the main areas of attack and possession of the opposing team, and in the other hand it can help the referee team to clarify difficult situations analysis, for example when the ball leaves the field of play or when it enters the goal.

In this thesis, several videos will be used for the development of the four methods. The videos were obtained from the FADEUP pavilion, and from the final of the Portuguese Supercup 2011.

Agradecimentos

À minha família e namorada por todo o apoio emocional dado.

À Q-Free empresa que soube compreender o meu estatuto de trabalhador-estudante.

Ao professor Doutor Luís Paulo Reis e a Catarina Santiago por toda a ajuda prestada.

List of Contents

List of Figures

List of Tables

Abbreviations and Symbols

List of abbreviation

AIS	Adjacent Images Subtraction
BLOB	Binary Large Object
CHT	Circle Hough Transform
DEEC	Departamento de Engenharia Electrotécnica e de Computadores
DFT	*Discret Fourier Transform*
DST	Destination
FEUP	Faculdade de Engenharia da Universidade do Porto
FPS	Frames per Second
H	Height
HSV	Hue Saturation Value
IP	Internet Protocol
NAT	Network Address Translation
RGB	Red Green Blue
ROI	Region of Interest
SBS	Static Background Subtraction
SRC	Source
SVM	Support Vector Machine
W	Width

List of symbols

cm	Centimeter
m	Meter
mm	Millimeter
ω	Angular Frequency
α	Angle

Chapter 1

Introduction

Image processing is being widely studied and developed throughout the world in various areas of research. The main areas of development are in the medical field, in surveillance and sports. The convergence of computer vision and multimedia technologies has led to the desire to develop applications for the automatic analysis in sports. The searching for a more competitive team in sport is the main objective for coaches. Digital image processing is becoming an important ally to understand and study a high variety of Sport.

Sports Image processing can be applied to identify players, balls, and to construct game models of the teams. In this particular project, the main objective is to follow the ball and to extract the 3 dimension coordinates of it.

In this chapter I present the reason and purpose of this Thesis. It will be presented the objectives and the results expected for this project.

1.1 - Objectives

The main goal of this Thesis is to detect and follow a ball in a Handball game. The challenge is to implement a new algorithm capable of detect and follow the ball in a very feasible way. I will develop some methods for this purpose and make a comparison between them.

This project can be seen as a tool for sport coaching. The video sequences are potentially of great interest for coaches and fans. The results could be used to analyze game tactics and to generate game statistics. Also, this project could be great helpful for the referees, because with the following of the ball, is possible to detect get off the field, or even better, it could detect if the ball over passed the goal line.

An example of the image that is going to be used is given in Figure 1.

Figure 1 - Picture of the field t maximum resolution

It would be very grateful, if this project become a basis to a more robust tracking system. If that happens, a more commercial system should be implemented and sold to clubs, referees associations and championship leagues associations. That would be a good investment for potential developers of the system.

1.2 - Description of the Problem

The position of the camera is good for tracking the players, because it gets almost the entire field, but when it comes to detect the ball a lot of problems rise from that type of image.

The first one is the distance to the playing action, making the ball very small, also it has a high barrel effect introduced by the radial distortion of the lens.

In a Handball game match the ball is invariably the focus of attention. Although players can be successfully detected and tracked on the basis of colour and shape, similar methods cannot be extended to ball detection for several reasons:

- The ball is very small and moves fast, and consequently exhibits it usually has irregular shape, various size and unstable colour when moving in different velocities.

- The ball is occluded almost all the time by players, so it is hard to identify it.

- Other objects on the pitch may look like the ball (e.g. players' heads or other body parts), I will call them "false alarms".

- The ball could get velocities in the order of 28 m/s, so with a rate of 25 FPS the ball could travel 1.2 meters between 2 frames. So if the ball were not detected in one frame, the distance between two positions of the detected ball could reach 2.4 meters.

Generally, in TV broadcast streams the ball is mostly of good resolution in the image center. But in this work, the cameras are fixed in the top of the pavilion, so the ball will not be always in the center of the image, causing some difficulties in its identification, besides that, the ball travel at high speeds, making its shape change a lot, like is possible to see in figure 2.

There have been successful attempts such as a football ball tracking system based on modified Hough Transform technique, and a baseball tracking system using color-based region detection technology. But in the Handball case this kind of approach is now in a beginning phase. Tracking the ball in a Handball game poses many challenges due to the ball small size and high speed. Tracking a Handball ball is challenging due to the small size of the ball, the relatively large size of the field, the high speed it travels and the high congestion of players in a small area. Some detection algorithms were discarded from the beginning, like the Hough Transform, because it usually works with objects that move much slower and are bigger than the Handball ball.

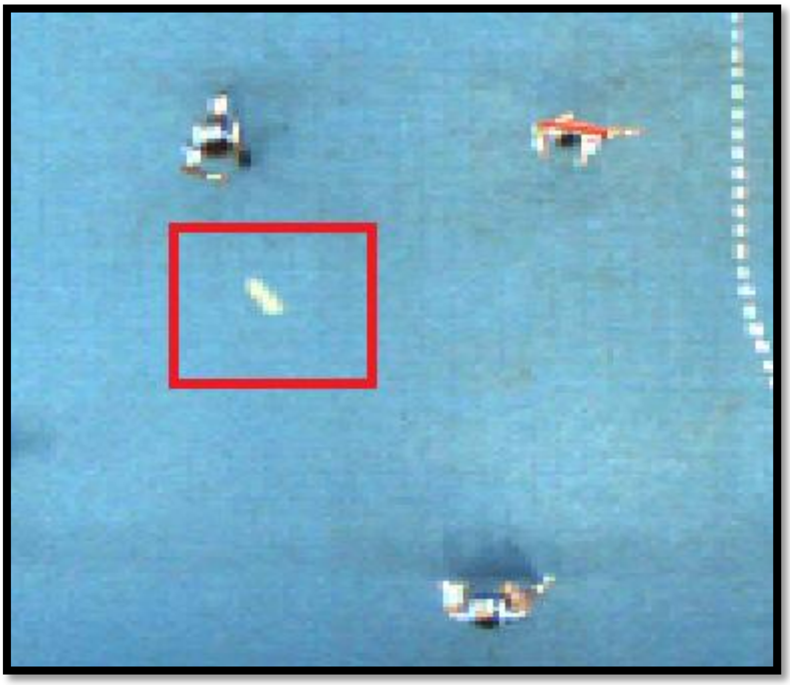


Figure 2 - Example of the ball in the image

To solve this problem we have decided to use two architectures. One is based on the acquisition and processing of images from a single camera located in the center of the field. It will be developed two possible approaches to this architecture. Some problems due to this architecture must be overcome like the difficulty in extraction of the height of the ball (z coordinate).

The other architecture is the use of a stereo vision system. It will be composed of two or more cameras getting the images from different perspectives. With this information it will be shown a possible way to obtain the 3 Dimensions coordinates (x,y,z).

It will be made a comparison between these two architectures in 3 parameters: Time consuming (FPS), percentage of correct ball detections, false positives and reliability of the obtained position. The comparison of the ball position will be made with a system of sensors installed in the field.

We will proceed to the background subtraction from 2 possible ways. The first will be the subtraction of gray-scale images; the other way will be the subtraction of edges images.

The accuracy of the proposed method will be tested with a feasible system of sensors.

Another concern is the amount of data that needs to be processed to have a real-time analysis. In our example images we will have a resolution of 1024x768 with a 15 FPS rate, so the amount of data that needs to be processed is about (1024x768x8x15)/8 = 11,8 MB/sec.

1.3 - My Solution

To solve this difficult task, I have proposed some solutions. I have outlined some objectives to achieve:

- Clean image from noise
- Identify moving features and agglomerate them
- Develop a technique for the determination of the (x, y, z) coordinates
- Apply physics equations to the movement of the ball
- Apply techniques to optimize the detection and tracking of the ball

Some know processing image techniques will be applied like:

- Image Difference
- Threshold of the image
- Elimination the noise with a Smoothing Filter and Erosion operation.
- Dilation to agglomerate the pixels into objects(Players)
- Detect the Ball and extract the (x ,y) position.
- Image colour segmentation
- Filter object by Size and Colour

In this project I developed 2 main methods, all of them are based in the previous techniques. The first One is based in Background Subtraction with image color segmentation by the color of the ball. Then, a filter is applied to detect the ball by size. The second one is based in an Adjacent Image Subtraction with image color segmentation by color of the ball, and again a filter for the size of the ball is applied.

But this methods, as the reader will see, have some problems, and my challenge is to develop other methods to overcome this. So 2 new methods will be developed based on the first 2 ones. Those methods, make also the detection of players, and makes estimations about the position of the ball in the next frames.

1.5- Structure of the Thesis

This Thesis is organized in 6 chapters. It starts to give an explanation about the game and image processing techniques. Then, the developed methods are presented and explained. Finally the results are shown with a comparison being between them.

Chapter 1, Introduction, explain the purpose of this project, given a small introduction to the objectives and developed methods to achieve them. The next Chapter, State of Art, give the reader an overview about the Handball game, with some rules and objects necessary to their practice. In that chapter, is also possible to see some basic introduction to image processing techniques, like smoothing, image subtraction, Hough transforms among others. Also a presentation of others work will be made. In Chapter 3, a system overview will be presented. In this chapter, is possible to see the basis for the development of the methods, with the presentation of the High and Low level approaches. The OpenCV is also described in this chapter, the different used videos are also shown. In Chapter 4 is made an explanation about the main techniques, like the Static Background Subtraction, and the Adjacent Image Subtraction. It is also explained, how the distorted image correction due to the radial distortion of the lens is made. After that an explain of the developed methods is made. Those methods are, Static Background Subtraction with Balls color segmentation, Adjacent Image Subtraction with Balls color segmentation, the third method, combines the first two ones in order to reduce the false detections of lookalike balls objects. The final developed method, also combine the 2 first ones, but it also predict the ball location in the next frame, detect the ball when it is stopped or in the possession of a player. The next Chapter, it is presented the results for each developed methods, and an evaluation of them is made.

Chapter 2

State of Art

In this chapter a theoretical explanation will be given about the Handball game. With this, is expected to explain to everyone the basics of this Sport. It will be given an explanation about the rules of the game. A short presentation will be made about the equipment and methods used in this project.

2.1 - Sport Analysis

A short introduction about Handball is made in this point.

2.1.1 – What is Handball

Handball is one of the fastest indoor sports. It is an Olympic sport played in over 150 countries. It is a sport where players are encouraged to be athletic, be flamboyant and inventive, and above all you work together as a team. It helps you keep fit and healthy.

With physical contact, unlimited substitutions and shots at goal that can reach over 100 km/hour, there are always something exciting happening on the field of play. This game is especially appreciated in the north Europe.

2.1.2 – Basic Rules

Each team has 12 players. However, only 7 players take the court at any one time and one of these must be the goal keeper. The reminding players are substitutions during the game. They enter and leave from the substitution area of the court. Players alternate between an attacking position and a defending position, depending on who has possession of the ball. The goal keeper of each team wears a different colour from the rest of the team.

For adults, the game lasts for two 30 minute halves. For children of different age groups, the halves are of shorter lengths. The emphasis of children's games is not winning or tactics, but rather to establish friendships, develop skills and confidence.

Generally there are 4 officials: 2 referees, 1 time keeper and 1 score keeper. Two referees are required because of the fast nature of the game. One referee is on the outer goal line and watches for any goal are violations and verifies if the ball completely crossed the goal line for a goal. The other referee is behind the attacking team, watching for any defensive or offensive fouls by the court players or any possession violations. The time keeper looks after the time and notes the time when a player may renter the court after a suspension or disqualification. The score keeper keeps the score and statistics of the match. The time keeper and score keeper also assist the referees if any illegal substitutions occur

2.1.3 – Ball characteristics

The rules are virtually the same for men's and women's handball. Only the size of the ball is different. For children the ball is smaller again. As a guide the ball sizes are as follows:

- for under 8 year olds the circumference of the ball is 48 cm and the weight is at least 290 grams (size 0)
- for male youth (8-12 years) and female youth (8-14 years) the circumference of the ball is 50-52 cm and the weight is at least 315 grams (size 1)
- for women, male youth (12-16 years) and female youth (over 14 years) the circumference of the ball is 54-56 cm and the weight is at least 325-400 grams (size 2)
- for men or male youth (over 16 years) the circumference of the ball is 58-60 cm and the weight is at least 425-475 grams (size 3)

The ball is the most important object in the game. The team who scores more goals win the game, so the possession and control of the ball is the main concern of the players. The ball has a small size and could reach a high speed.

The ball could vary in size, according to the category of the game:

Table 1 - Dimensions of the ball

SIZE	USED BY	CIRCUMFERENCE (IN CM)
III	Men and male youth older than 16	58–60
II	Women, male youth older than 12 and female youth older than 14	54–56
I	Youth older than 8	50–52

An image of the ball can be seen in the next figure:

Figure 3 - Typical ball of the handball game

2.1.4 – Field characteristics

The field in which this game is played have a rectangular form. The field has 40m*20m. The most important line in the field is the 6 m line that corresponds to the zone of the keeper and no one else could violate that area. Other important mark is the 7m line. When one player that suffer a fault in the moment of the shoot, a 7m free shot are given to his team, is like a penalty shootout in the football

Usually Handball is played in pavilions like the one in the figure 4.

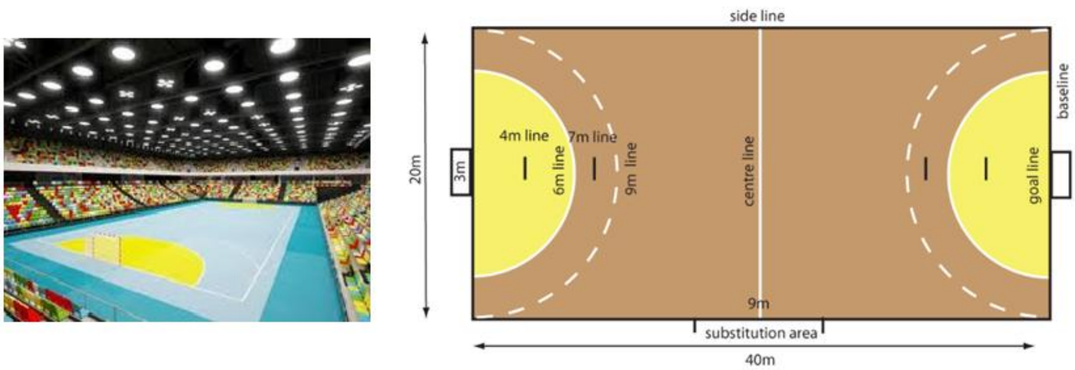


Figure 4 - typical pavilion and field measurements

2.2 -Cameras

An essential equipment in this project is the camera. The camera chosen was a digital camera. A digital camera is a camera that takes video or still photographs, or both, digitally by recording images via an electronic image sensor.

Figure 5 - Digital camera

Digital cameras can display images on a screen immediately after they are recorded, storing thousands of images on a single small memory device, and deleting images to free storage space.

2.2.1 -Digital Single-lens

Digital single-lens reflex cameras (DSLRs) are digital cameras based on film single-lens reflex cameras (SLRs). They take their name from their unique viewing system, in which a mirror reflects light from the lens through a separate optical viewfinder. In order to capture an image the mirror is flipped out of the way, allowing light to fall on the imager. Since no light reaches the imager during framing, autofocus is accomplished using specialized sensors in the mirror box itself.

2.2.2 -Methods of Image Capture

At the heart of a digital camera is a CCD or a CMOS image sensor. In a CCD for capturing images, there is a photoactive region (an epitaxial layer of silicon), and a transmission region made out of a shift register (the CCD, properly speaking). An image is projected through a lens onto the capacitor array (the photoactive region), causing each capacitor to accumulate an electric charge proportional to the light intensity at that location. A one-dimensional array, used in line-scan cameras, captures a single slice of the image, while a two-dimensional array, used in video and still cameras, captures a two-dimensional picture corresponding to the scene projected onto the focal plane of the sensor. Once the array has been exposed to the image, a control circuit causes each capacitor to transfer its contents to its neighbor (operating as a shift register). The last capacitor in the array dumps its charge into a charge amplifier, which converts the charge into a voltage. By repeating this process, the controlling circuit converts the entire contents of the array in the semiconductor to a sequence of voltages. In a digital device, these voltages are then sampled, digitized, and usually stored in memory; in an analog device (such as an analog video camera), they are processed into a continuous analog signal (e.g. by feeding the output of the charge amplifier into a low-pass filter) which is then processed and fed out to other circuits for transmission, recording, or other processing.

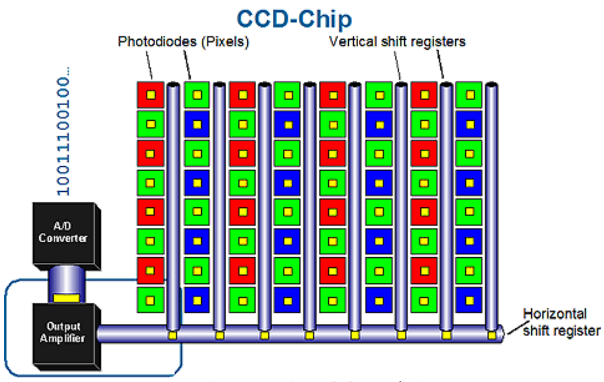


Figure 6 - CCD chip

2.3 -Basics of Image Processing

To realize this project is very important to have a good understanding on Image Processing techniques. Image processing can be used in many areas, namely in the medicine, surveillance, industries, face recognition and many more.

One of the most important things to know is the colour representation model. The most important are RGB and HSV colour models.

2.3.1 – RGB Color model

The RGB model is represented by a Cartesian coordinate system. The colour space can be represented by a cube as the one in figure 7. As seen in the 7 figure, the values of the colors are normalized between 0 and 1. The images have always the 3 components of this space, combined together to represent some colour. The number of bits used to represent each pixel in RGB space is called the pixel depth. The Red, Green and Blue images are an 8-bit image in the RGB images. So each RGB colour pixel has a depth of 24 bits. Usually the values used in the representation of each channel is in the range [0,255], instead of the normalized values [0,1].

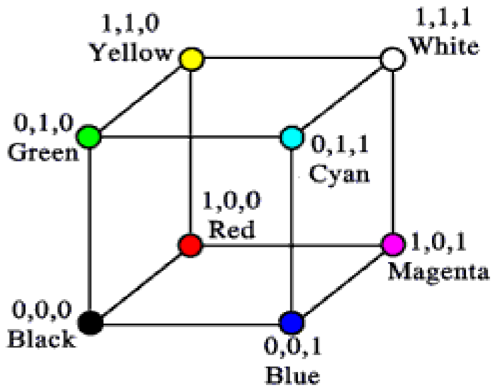


Figure 7 - RGB cube
[1]

2.3.2 -HSV Color Model

In my thesis I will use another colour space, the HSV space. This colour space is more approximated to human eye representation of the colour. When humans view a colour object, we describe it by its Hue, Saturation and Intensity. The Hue is the colour attribute that describes the true colour, Saturation gives the degree of dilution of the colour with white colour and Intensity gives the brightness of the colour. Because this colour space is very similar to the perception of the human vision, this become the most used colour space in the description of some colour object.

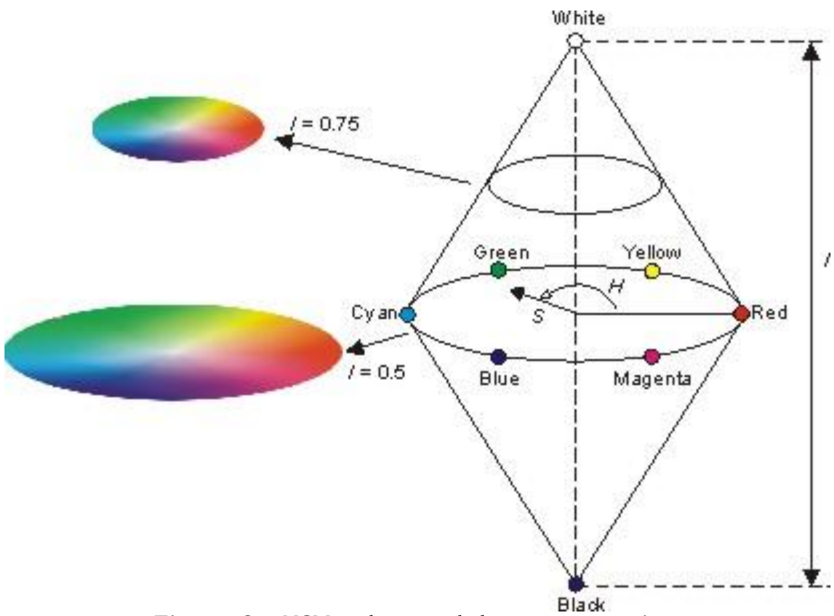


Figure 8 - HSV color model representation

In this project I will use the HSV colour model representation because of its invariance to light changes. Is very important to have a robust system capable of bee has many as possible independent from external factors.

[1] Image taken from http://viz.aset.psu.edu/gho/sem_notes/color_2d/html/primary_systems.html

Some other techniques will be presented, like smoothing and sharpening of an image.

2.4 – Image Transforms

In image processing a variety of techniques can be applied to remove noise, isolating individual elements and joining separate elements in the image. These techniques could be used before or after the processing of the image. Next I will present the techniques most used in my work, namely: Smoothing, and the two image morphological operations of Dilation and Erosion. I will present either some technique to find contours with edge operators, detect circles with Circle Hough Transform a algorithm to predict the route of the object using Kalman Filter.

2.4.1 - Smoothing

Smoothing is a very useful image processing technique. The pixels are modified based on the characteristics of the surrounding pixels.

The smoothing of an image is a very simple process. First a mask must be defined. That mask has all coefficients at 1's. The objective is to slide the mask over the image and each pixel is replaced by the average of the pixels in the neighborhood defined by the mask. This process is a linear operation based on a convolution between the mask and the image. The operator applied is:

$$h = \frac{1}{L^2}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & 1 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2.1}$$

Where L is the "LxL" dimension of the mask.

There are some morphological operators that can be used in the treatment of the images. The most important are the Dilation and Erosion. It is also possible to combine both to obtain some particular results in the image, the two possible combinations are opening and closing.

2.4.2 - Dilation

Dilation is a morphological operation that gradually enlarges the boundaries of regions of foreground pixels. With greyscale images, dilation increases the brightness of objects by taking the neighborhood maximum when passing the structuring element over the image. With binary images, dilation connects areas that are separated by spaces smaller than the structuring element and adds pixels to the perimeter of each image object.

The algorithm works as follows: it takes two pieces of data as inputs. The first is the image which is to be dilated. The second is as set structuring element also known as kernel.

The Kernel determines the effect of the dilation in the input image. Most often the kernel is a small solid square with an anchor point in the center. The kernel is scanned over the image, and the maximal value of the overlap is replaced in the anchor pixel. It could be seen in the next example with a 3x3 kernel:

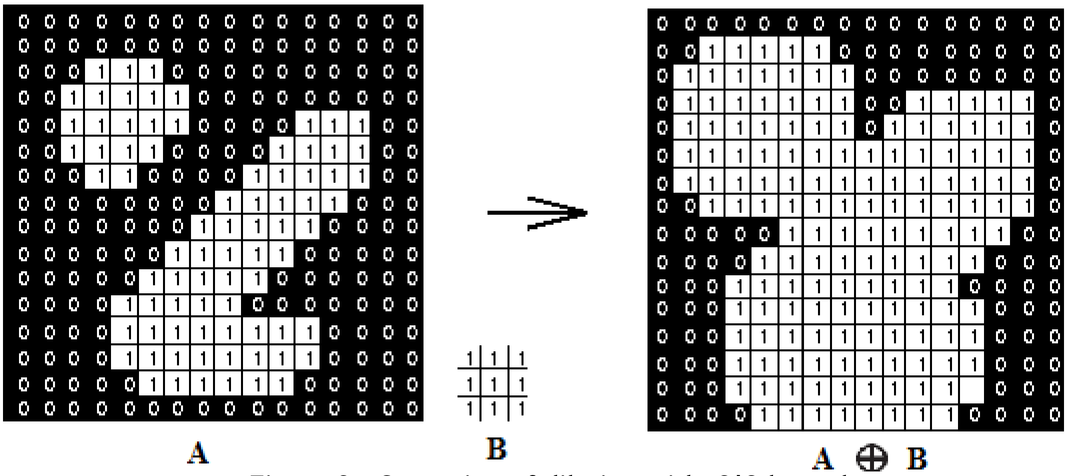


Figure 9 - Operation of dilation with 3*3 kernel

2.4.3 Erosion

Erosion generally decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element. With grayscale images, erosion reduces the brightness (and therefore the size) of bright objects on a dark background by taking the neighborhood minimum when passing the structuring element over the image. With binary images, erosion completely removes objects smaller than the structuring element and removes perimeter pixels from larger image objects.

The process is almost the same as the dilation process, but instead of replacing the overlap minimum, the pixel value is replaced by the minimum value of the overlap between the kernel and the image pixel. An example of operation could be seen in the next example:

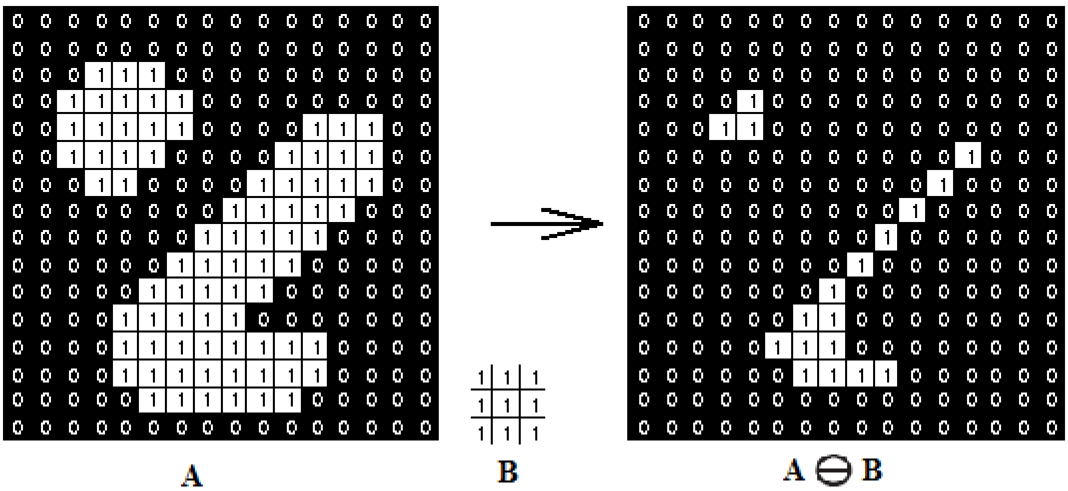


Figure 10 - Operation of Erosion with a 3*3 kernel

2.4.4 - Opening and Closing

It is possible to combine the two morphological operators as we which, to obtain the desired results.

Opening and closing are combinations of the erosion and dilation operators. In the Opening case the erosion operator is first applied and after that it is applied the dilation operator. Opening is often used to count regions in a binary image. For example, if we have thresholded an image of cells on a microscope slide, we might use opening to separate out cells that are near each other before counting the regions. The effect of the operator is to preserve *background* regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

In the case of Closing, we dilate first and then erode. Closing is used in most of the more sophisticated connected-component algorithms to reduce unwanted or noise-driven segments. The effect of the operator is to preserve *foreground* regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

2.4.5 - Median Filter

This type of filter is used for reduce the small noise in an image. Normally small noise are some pixels that have a grey value different from its neighbor pixels. This technique consist in a definition of a window around that noise pixel. The idea, is to compute the median of the grey value of all the pixels in the window, and put that value in the middle of the window, in the position known as anchor.

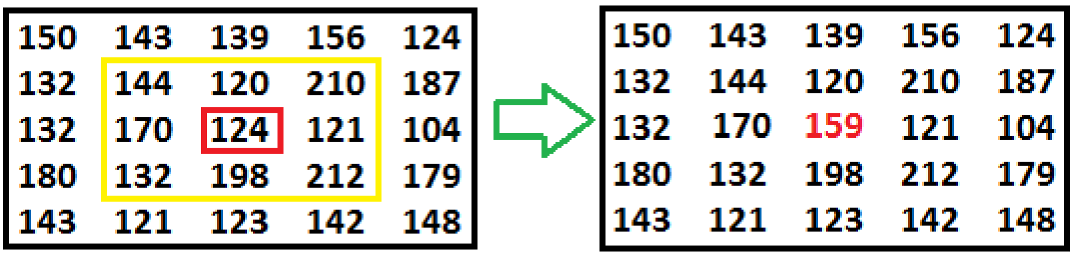


Figure 11 - Smoothing filter operation

Is possible to see in the figure 11 that, a 3x3 window was defined in a portion of the image centered in a pixel with a grey value of 124. Then the median value of all pixels was calculated: 144, 120, 210, 170, 121 132, 198 212 e 124. That calculation gives a value of 159. That value was inserted in the middle of the window, as is possible to see the image of the right. This process is performed all over the image.

This fact becomes clear if detection of contours is applied. In my example, I use a 3x3 window, but is normal to see median filter based on a 5x5, 7x7 and 9x9 windows.

Is important to refer, that, with the application of this method, some characteristics of the image could be lost in the area of the edges of the objects. This will be seen in the next point, when the reader could see the how to detect the boundaries of the contours.

2.4.6 - Canny Edge Detector

Edges in images are areas with strong intensity contrasts, i.e., a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. The Canny edge detection algorithm is known to many as the optimal edge detector[Canny_86]. image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (no maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a no edge). If the magnitude is above the high threshold, it is made an edge.

An example of an edge image operation can be seen in the next figure.

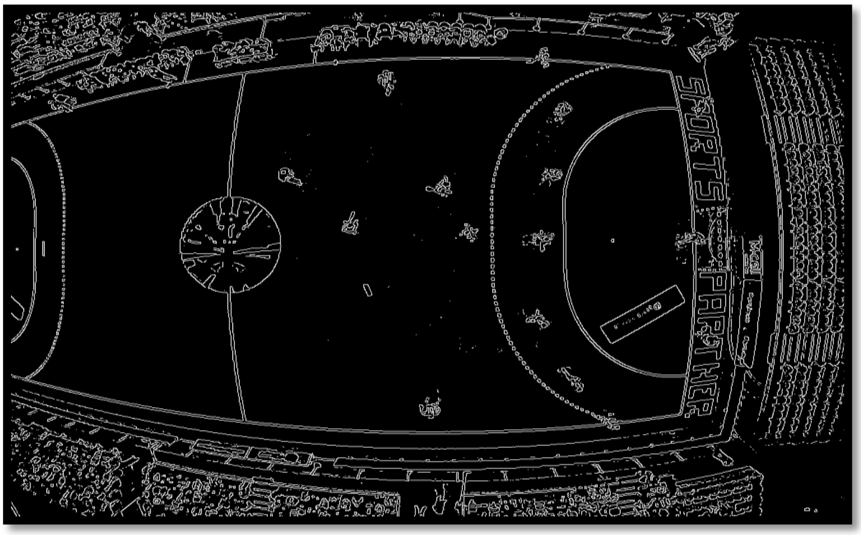


Figure 12 - Result Image from the application of the canny edge operator

2.4.6.1 - Threshold

In this project, all the developed methods require the application of a threshold mechanism as we shall see later in the presentation of them. There are 2 main types of threshold in image segmentation:

- Simple Thresholding
- Adaptive Thresholding

Thresholding is usually the first step in image segmentation and can be defined by the following mathematical equation:

$$g(x,y) = \begin{cases} 1 & if \quad f(x,y) > T \\ 0 & if \quad f(x,y) \le T \end{cases} \qquad (2.2)$$

Where T is the Threshold value. The threshold value is very important. A wrong threshold could provoke a disaster in the detection system.

The Simple Thresholding algorithm is based on the histogram of the image. The T value is calculated as follows:

Table 2 - Threshold algorithm

1. Start to select a value for T, based on the average values of the grey image.
2. Segment the image using T to produce two groups of pixels: G1 consisting of pixels with grey levels higher than T, and G2 with grey levels lower than T.
3. Than is computed the average of the grey values from the pixels of G1 to give $\mu 1$ and from G2 to give $\mu 2$.
4. A new Threshold is than computed:

$$T = \frac{\mu 1 + \mu 2}{2} \qquad (2.3)$$

5. The steps 2-4 are repeated a predefined number of times.

The biggest problem of this Basic Single Value Threshold technique, is that it not respond very well for bimodal histograms. If there is an histogram with several local maximums, the extraction of a particular object would be impossible using a single threshold value. The solution comes with the application of an Adaptive Threshold technique. The algorithm starts to divide the image into sub images and threshold these individually. Since the threshold for each pixel depends on its location within an image this technique is said to adaptive.

2.5 – Circle Hough Transform

Hough Transform is an algorithm to find lines and circles in an image. The line Transform is the simplest Transform, but for this work the Circle Hough Transform is the one that is more important. Basically it has an accumulator space with 3 dimensions.

One for *x*, one for *y* and another for the radius *r*. The first process is to determine the edge of the image.

To find a circle in the image by this algorithm is necessary to consider that any point in a binary image could be part of some set of possible circles. A circle with radius R and center (a, b) can be described with the parametric equations.

$$x = a + R\cos(\theta) \quad (2.4)$$
$$y = b + R\mathrm{sen}(\theta) \quad (2.5)$$

If we want to detect some circle in the image, it is good to know the possible range of the Radius R. If the value of the circle is defined, the search resumes to only 2 dimensions.

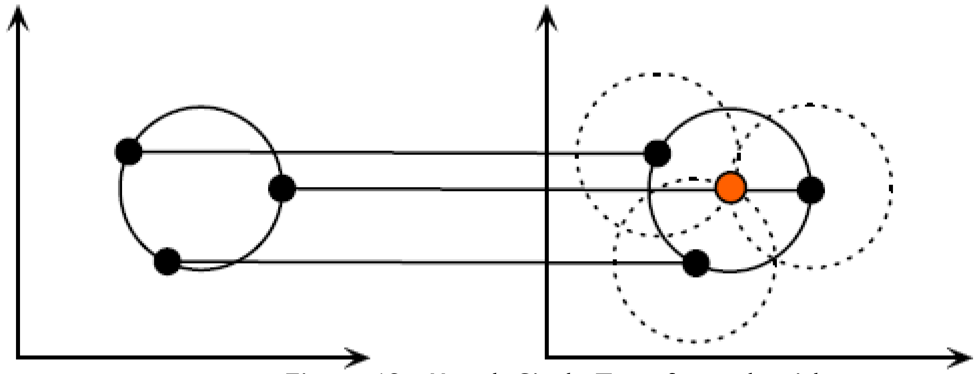


Figure 13 - Hough Circle Transform algorithm

The algorithm works as follow: First it is defined a radius R and a θ angle with a range [0,360]. Each point in the left image generates a circle with radius R in the parameter space (right image).

The main disadvantage of HCT is the heavy processing time that it implies a slower speed. The intersection of the circles in the right image from figure 13 gives place to an accumulator. The point with the highest value is defined as a center of a circle in the left image.

Multiple Circles with Known R

We can find multiple circles using the same method.

2.6 – Motion Based Algorithms

Motion Based algorithms are used to construct the motion information of the objects. The base to create the model, are the currently available measurements and an estimation mechanism. These algorithms are also able to detect objects candidates.

Motion information demonstrated during a Handball game is very obvious. The ball when is not in the possession of any player moves across the field following physical patterns.

2.6.1 – Kalman Filter

There are some studies that use the Kalman Filter for tracking objects. In my project the Kalman Filter could be used to predict movement and position of the ball.

The Kalman filter is a state estimation algorithm based on a feedback control mechanism: this filter predicts the process and then obtains the feedback from the measurements. We can construct a sequence of estimated locations where the objects might appear in images. Dynamic information such as the speed and the acceleration can also be estimated.

I could apply this algorithm to my project, in order to predict the position of the Ball and the Players in the image. The application of this algorithm also helps in the detection accuracy by eliminating objects that are not located at or near the predicted location.

The negative aspect of this algorithm is the processing cost. Is a very heavy algorithm, not very compatible for a real-time requirement.

2.6.2 – Object Tracking Based on Trajectory Properties

A simpler algorithm could be applied based on the trajectory of the ball and includes 4 components: ball size estimation, candidate estimation, candidate trajectory generation and trajectory processing. This algorithm was presented by Guezic, for detecting and tracking a football ball in broadcast videos. In the ball size estimation component, the ball size is estimated from that of salient objects such as the goalmouth, the ellipse and the people. The size of the ball varies from frame to frame. Its size is estimated based on these reference objects. In the candidate detection component, some non-ball objects are removed from the image. In the candidate trajectory generation component, a set of candidate feature images are created from all candidates in the sequence of frames. Each candidate feature image produces a candidate trajectory set. In the trajectory processing component, the trajectory of the ball is detected from the set of candidates. Other trajectories overlapped with that of the ball are removed. The ball trajectory and the remaining trajectories are extended using a model matching procedure. Most of the objects contained in the ball trajectory are the ball.

2.7 – Moments

To obtain more information from the Contours, it is important to obtain some characteristics from them. To calculate the moment of the Contour, is performed an integration over all the pixels.

$$m_{p,q} = \sum_{i=1}^{n} I(x,y)\, x^p y^q \qquad (2.6)$$

The sum is over all the pixels of the Contour boundary denoted by *n* in the equation. One useful thing that we can get from the moments is the *center of mass*. We that we can search for the central of mass of any object, by computing his 1st moment. So the x and y coordinates of that object center will be given by the following equations:

$$x = \frac{\sum_x \sum_y x I(x,y)}{\sum_x \sum_y I(x,y)} \qquad (2.7)$$

$$y = \frac{\sum_x \sum_y y I(x,y)}{\sum_x \sum_y I(x,y)} \qquad (2.8)$$

2.8 – Related Work

In this point I will show some theoretical studies and developments done in this area. I will also show some related commercial products available in the market.

2.8.1 – Theoretical Studies

Now I'm going to examine methods developed by others. The using of images in the detection of objects is very useful in many areas. Several research and development were made in such areas like Medicine, Quality Control in industries, in self-navigation of vehicles, in sports and many others.

An area in great expansion is the detection of human actions with the identification and interpretation of some events. Some authors like G. Herzog, C.-K. Sung [HerSun_89] have dedicated some studies to the interpretation of the player's events in a Football game. Some particular techniques were developed to do that, like the Hidden Markov Models applied in the methods of Y. Wu and T. S. Huang [WuHuan_99]. The goal of the work developed by R. Mann and A. Jepson [ManJep_02] was to find and analyze motion events and make causal inferences for enhanced extraction of scene dynamics.

After my research, I conclude that most of the research in object detection was made in face and texture classification. Few studies were found on the detection of the ball, I will present the most important ones.

Most of the research in ball detection is based in color segmentation. That is always made by applying an edge detection algorithm first. Hongyng Zhang, present a ball detection algorithm based on colour information and Hough Transform. His main objective was to detect the colour balls under different illumination conditions. To do that, he first applies a multiscale Retinex algorithm to enhance the image. Then an adaptive Hough transform was introduced to locate the position of the ball.

A particular technique is quite often used to detect circular object in images, which is the Circle Hough Transform (CHT) used by D'Orazio and Guaragnella [OraGua_03]. The principle of operation of this Transform has already been explained above. Despite the good result of CHT in static images with big circles, the performance isn't so good when the ball get high velocity, that' because of the blurred effect of the ball. One of the most disadvantages of this method is the high amount of computation needed to perform the detection of the circles. That task is even more difficult if radius is not known.

Another technique was developed by Ville Lumikero, the main process consist in threshold a greyscale image into a binary form [Lum_04]. After that he applied morphological operations such as erosion and dilation to "clean" the images from noise and to fill up holes in objects. Then the ball candidates are thresholded by size and colour.

Dawei Liang, provides a method for ball detection and tracking based on broadcast football video. It extracts from several frames the balls, based on their colour, shape and size. In ball tracking procedure, Kalman filter based template matching is utilized to track the ball in subsequent frames. Kalman filter and the template are initialized using detection results. In each tracking step, ball location is verified to update the template and to guide possible ball re-detection. To solve the false positive detection Dawei has created a graph.

Each graph node represents a ball candidate. Since the ball's locations in two adjacent frames are close to each other, only those candidate pairs (between adjacent frames) whose Euclidean distance is smaller than the threshold dmax contribute to the graph edge set.

B.L. Velammal created an anti-model approach to eliminate non-ball objects and remaining objects are identified as ball-objects. Region growing segmentation was chosen for segmentation. After segmentation, the ball and non-ball objects were classified using the shape properties. The non-ball objects were eliminated, and the resulting objects consist of only ball objects or ball candidates.

The method to search and detect the ball was based on segmentation of the frame.

He created a sequential model in order to extract the ball position:

Cricket video -> Conversion into frames -> Region Growing Segmentation -> Ball-Candidate Generation and Non-ball candidate elimination.

First Velammal pre-processed the video to remove noise and to enhance contrast. After that, a segmentation process was performed using Region Growing Algorithm. The elimination of objects was based on several sieves, as colour, shape and size. A median filter was used on the pre-processing in order to eliminate the noise. The he applied a Seeded Region-Growing Segmentation algorithm. To solve the problem of false positives, Velammal instead of concentrate the efforts in the identification of the ball in each frame, he applied an approach of analyze the candidates motion over consecutive frames. The general idea was to search for the ball on the ball candidates instead of in the entire frame.

Another way to detect objects in an image is by using a Boosted Cascade of Classifiers. Viola and Jones have developed this idea with a Boosted Cascade of Simple Features [VioJon_01]. They have divided their work in 3 stages. The first is the application of an "Integral Image", which allows the features used by their detector to be computed very quickly. The second is a learning algorithm, based on Adaboost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest.

Other techniques were developed algorithms capable of construct 3D trajectories. Hua-Tsung Chen have reconstructed 3D information from single view 2D video sequences based on the integration of multimedia features, basketball domain knowledge and the physical characteristics of ball motion. First he extracted the 2D trajectory and then with physical characteristics of the ball motion he reconstructed the 3D trajectories. He did this on broadcast basketball videos. Is very difficult in a single frame distinguish the ball. So he gathered motion information over successive frames in order to discriminate the real ball. To obtain 2D ball candidates over frames, he detects ball candidates by visual features and explores potential trajectories among the ball candidates using velocity constraint. To reconstruct 3D trajectories from 2D ones, he set up the motion equations with the parameters: velocities and initial positions, to define the 3D trajectories based on physical characteristics. The 3D ball positions over frames can be represented by equations.

The method used to detect the ball was analyzing the moving pixels and the ones in the basketball colour. To detect the moving pixels, a frame difference was applied.

To track the ball, Hua-Tsung Chen integrated the physical characteristic of the ball motion into a dynamic programming-based route detection mechanism to track the ball candidates, generating potential trajectories and identifying the true ball trajectory.

2.8.2 - Commercial Products

In terms of commercial products I found three companies with very good results, the Hawk-Eye Innovations, the TRACAB and the PROZONE Sports.

Hawk-Eye Innovations Ltd. has used the expertise gleaned from five years of broadcasting to remain at the cutting edge of sports technology and broadcasting. The company continues to grow and develop and Hawk-Eye recently created a system for the BBC's snooker coverage and is currently developing a goal-line football system for the English FA. It is based on the principle of triangulation using the visual images and timing data provided by high-speed video cameras at different locations around the area of play. The proposal involves placing six cameras in the stands at each end of the field. The images are processed from a bank of computers in real time. This data is then sent to a central computer, which combines all the information to determine whether or not the ball has crossed the line. As soon as a ball has been tracked across the goal line, the central computer will transmit an automatic signal directly to the referee to inform him whether or not a goal has been scored. This information can be communicated to a watch or an ear piece as required[1].

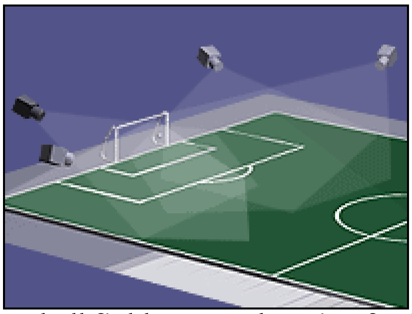


Figure 14 - Football field camera location for Hawk-Eye

In tennis **Hawk-Eye Innovations** uses Video Images is used to identify the center of the ball within each frame of each camera. Camera movement is compensated for by also tracking the lines of the court.

[1] Information taken at http://www.hawkeyeinnovations.co.uk

The system triangulates the information from each calibrated camera to provide the 3D position of the ball.

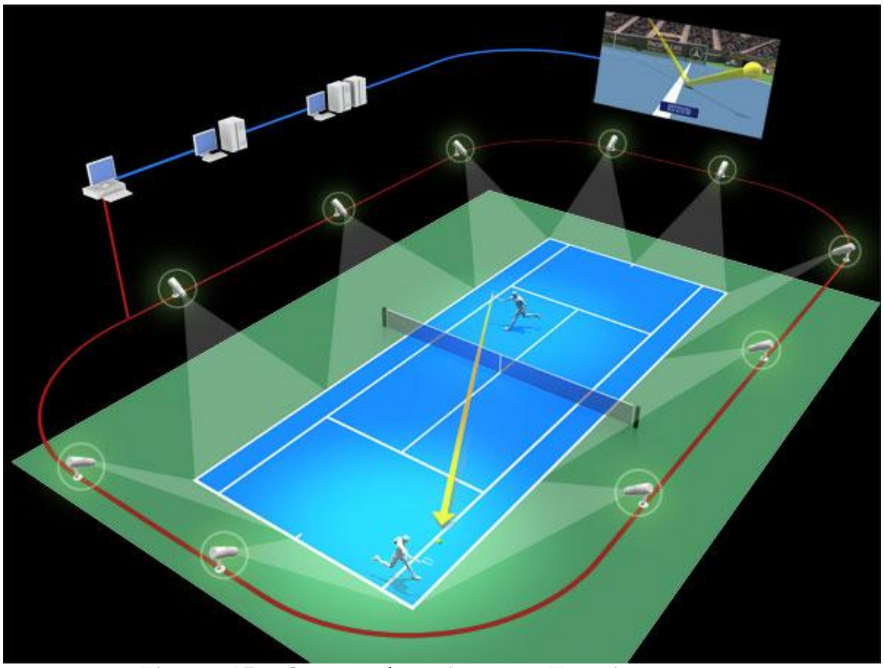


Figure 15 - Camera location at a Tennis court

Figure 16 - Calculation of the exact bounce area that the ball made with the court

Another company with good proven results is the **TRACAB**. The system consists of two multi-camera units which contains eight small cameras each. By using more than one camera the system gets less sensitive to a camera failure, even if one camera fails the unit will still be operational. It also provides the camera units with the feature of dynamic resolution, meaning that the individual cameras can be angled so that the camera unit delivers a higher resolution in the parts of the pitch where there are most crowded and occluded situations, such as the penalty boxes. The Tracab system is based on SAAB patented stereo vision technology. This basically means that every inch of the pitch is constantly covered by at least two cameras. The x, y, z measurements for the objects on the pitch are made through analyzing both these images and, most importantly, the difference between them. This method results in true three dimensional tracking[1].

[1] Information taken from the Website http://www.tracab.com

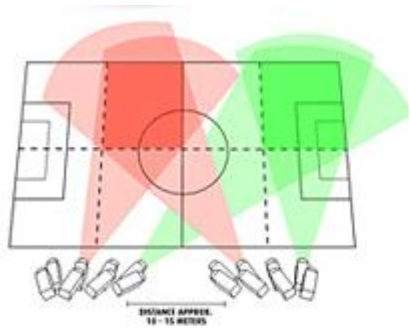


Figure 17 - Camera disposal for the TRACAB system

ProZone Sports is a company that has focus on the creation of models of the teams, and provides management and coaching staff with a multi-layered, interactive coaching tool via a rich and engaging interface. It allows immediate video access to every single 'event' in the game (e.g. shots, passes, crosses, set pieces) and also allows you to look at performance data and visual representations (e.g. pass maps) for and against individual players, units within the team or the team as a whole. To maximize interactivity and effective usage, all data and maps are linked to the video so that the coaching team can view the analysis in the context of the game.

2.9 – Conclusions

In this chapter, I have shown various techniques for object detection and tracking. Those techniques will be the base for my work. Despite the referred techniques were quite good for the detection and tracking of the ball, it must be considered that one single technique will not perform the entire intended project. So I must combine more than one technique to obtain the desired results. I must have in mind, that several difficulties must be overcome:

- Computation cost of each technique.
- Those techniques were not developed to track object that travel at such a higher speed.
- Generally the objects that those techniques follow are much bigger than a Handball ball, particularly when the cameras are so distant from the field.

I will combine essentially the following techniques:

- Background Subtraction.
- Edge Determination.
- Colour Determination.
- Contour Determination
- Motion and Dynamic Information

These techniques combined together will for sure result in an excellent method for tracking the Ball and the Players as well.

I will experiment 3 different methods of Background Subtraction, the Static, the Dynamic and the Edge.

An example for the use of Motion and Dynamic information could be the distance between the positions that the ball appears in two consecutives frames, could not be higher than a determined value.

In the next chapter, I will present some possible solutions based on the various ideas learned from this chapter.

Chapter 3

General Approach

In this chapter I will be show some solutions for the project, the used techniques and the developed algorithms. An explanation about the algorithms will be made, with some diagrams to help their understanding.

The main idea behind my solutions is to put the image in a situation that a contour algorithm will be applied to detect the objects. The key factor and the main goal are to achieve high accuracy and fast processing speed. To obtain that, I will perform some operations based on background subtraction and color segmentation/shape recognition. First I will apply different Background Subtractions, and then I will apply some morphological operations in order to obtain an output of agglomerated objects to be analyzed.

In a real game situation, if the algorithm finds more than one possible ball close to each other it is needed to decide which one is the real ball and which ones is noise (shadow or players body members). To solve that, one important technique must be taken in account, which is the determination of the colour of the ball.

To face the distance of the cameras to the field, and the high wide-angle which cause a high barrel effect in the images, another technique must be applied. To solve that, I decided to divide the image and apply different morphological operations to each part, taking in account the size and distortion of the objects, and the amount of noise in the image.

3.1 -Software Analysis

3.1.1 - High Level Approach

In a higher level the system presents the following architecture:

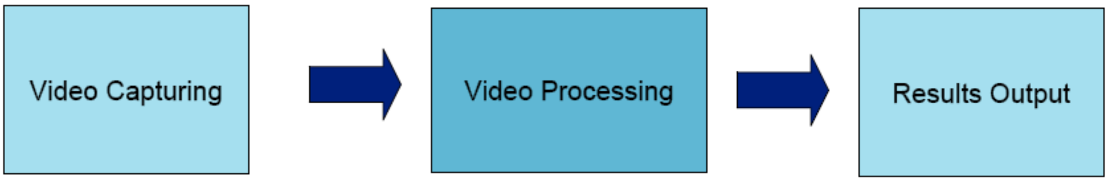


Figure 18 - High level system

The system computes a list of image regions where it assumes human faces. It is based in four stages:

1. Video capturing
2. Video processing
3. Results Output

The high level system could be described has a three major topics. The first one is the capturing of the images that would be made by some cameras placed in the top of a pavilion. Then a video processing is performed based on several image processing techniques described in the previous chapter. Finally the results are outputted to a file.

3.1.2 – Low Level Approach

The second main step in the system is the video processing. It consists in the analysis of the images, with the application of multiple image processing techniques.

The third and last step of the system is the way how the results will be shown to the user. A GUI interface will be developed for that.

In a low level system we got the following sequence:

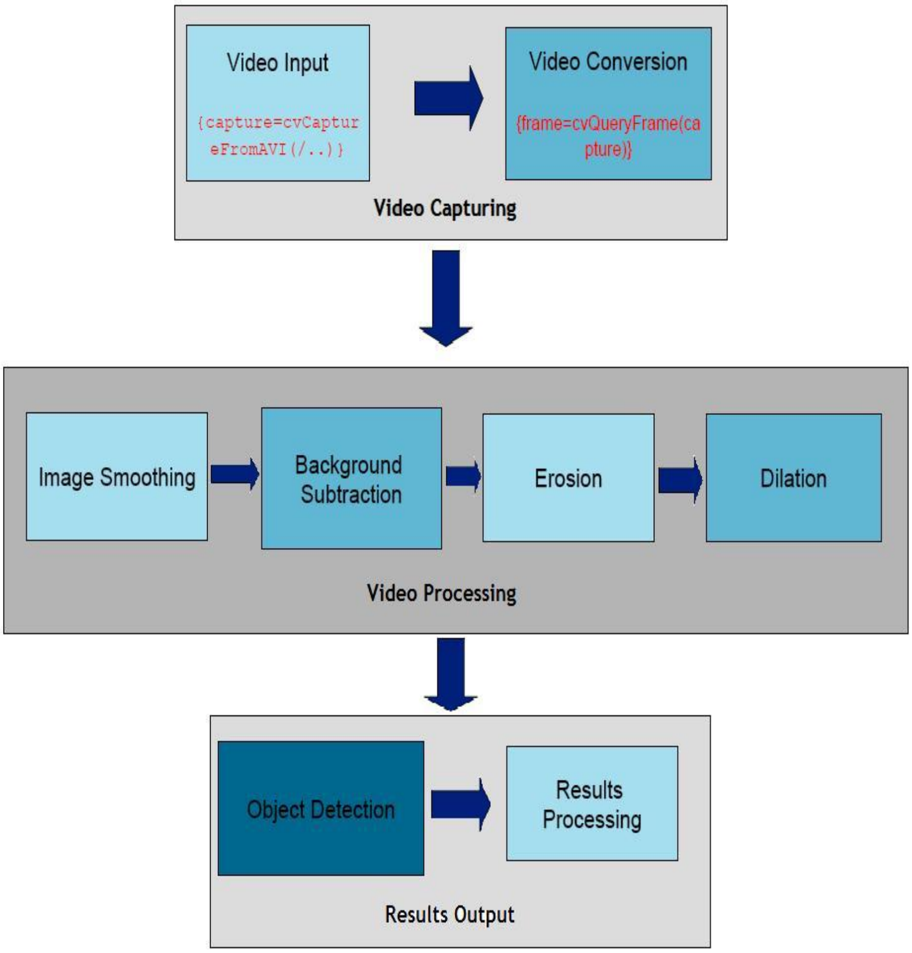


Figure 19 - Low level system

First the Images are captured and converted to frames. Then several images techniques were applied to the images. One common image processing technique is the Image Smoothing. This is used to reduce the noise by the application of low pass filters. After that an Image Subtraction is performed to identify the foreground objects. In the case of this Thesis two possible subtraction methods are applied, namely the Static Background Subtraction and the adjacent Images Subtraction. Some methods are going to be developed based on this two methods. After that some morphological operations are done. Usually the Erosion operation is used to remove the noise of the image, and the dilation operation is used to fill holes and to join some close foreground objects. Then an object filtering and detection algorithm is applied to identify the wanted object. Finally an output of the results is performed.

3.1.3 – Use Cases

In this point the interaction between the user and the system is presented.

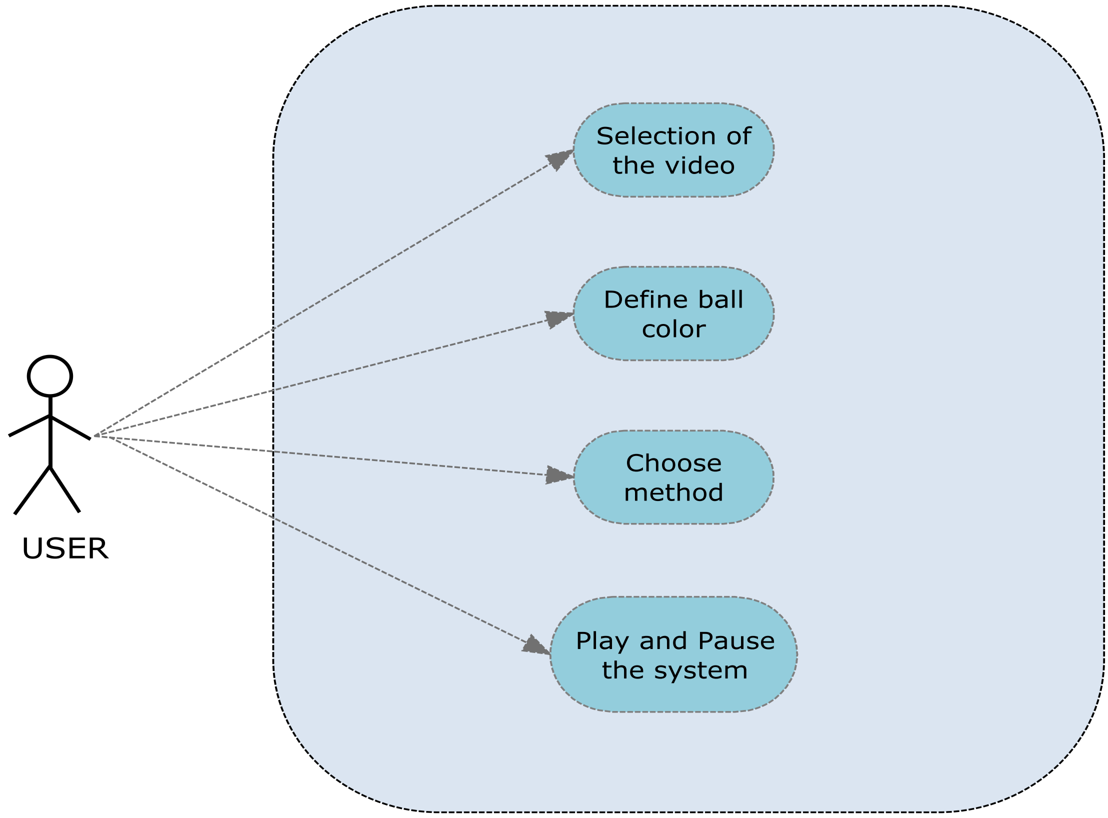


Figure 20 - Use cases of the system

As is possible to see in the figure 20, there are some interaction between the user and the system. In order to the system work, the user must perform almost all the actions presented in the last figure. The **Selection of the video** is a mandatory action to be done, because the system needs a video file as input. Because all the developed methods makes a segmentation of the image by the **colour of the ball** is necessary that the user select the colour of the ball. Another important use case is the **choose method** where the user must choose one the 4 implemented methods. One of the important action from the user is the possibility to **Start and Pause** the processing whenever he wants.

3.1.4 – Class Diagram

In this point, is done the definition of the classes, and how they are of the relation between them.

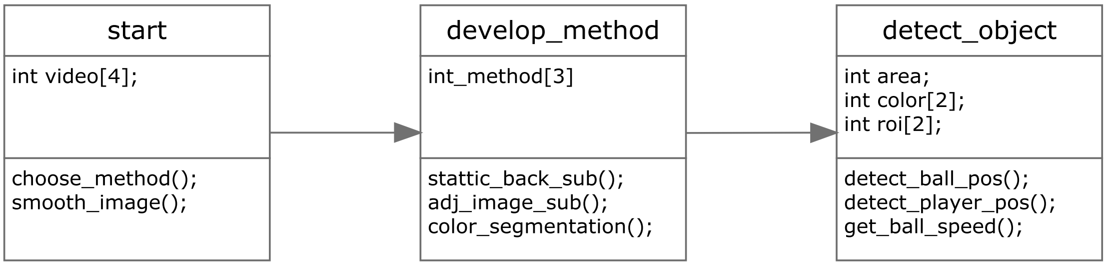


Figure 21 - Class diagram

The first class **START**, is the responsible for the choice of the video, and for the application of several operations to remove the noise of the images. The next class is the **DEVELPO_METHOD**, where the some operations will be made, like the **Static Background Subtraction**(static_bak_sub), and the Adjacent Image Subtraction (adj_image_sub). In this class there's another method responsible for the segmentation of the image by the colour of the ball, the **Color Segmentation.**

The last class is the **DETECT_OBJECT**, this one is responsible for the real **detection** of the **ball** and the **players**. Another important method in this class is the calculation of the ball's speed, responsible for the estimation of the ball's position.

3.1.5 – Open CV

The application was done in C + + using the OpenCV library. OpenCV is an Open Source Computer Vision Library broadly structured into five main components, four of which are shown in the next figure. The CV component contains the basic image processing and higher-level computer vision algorithms; ML is the machine learning library, which includes many statistical classifiers and clustering tools. HighGUI contains I/O routines and functions for storing and loading video and images, and CXCore contains the basic data structures and content.

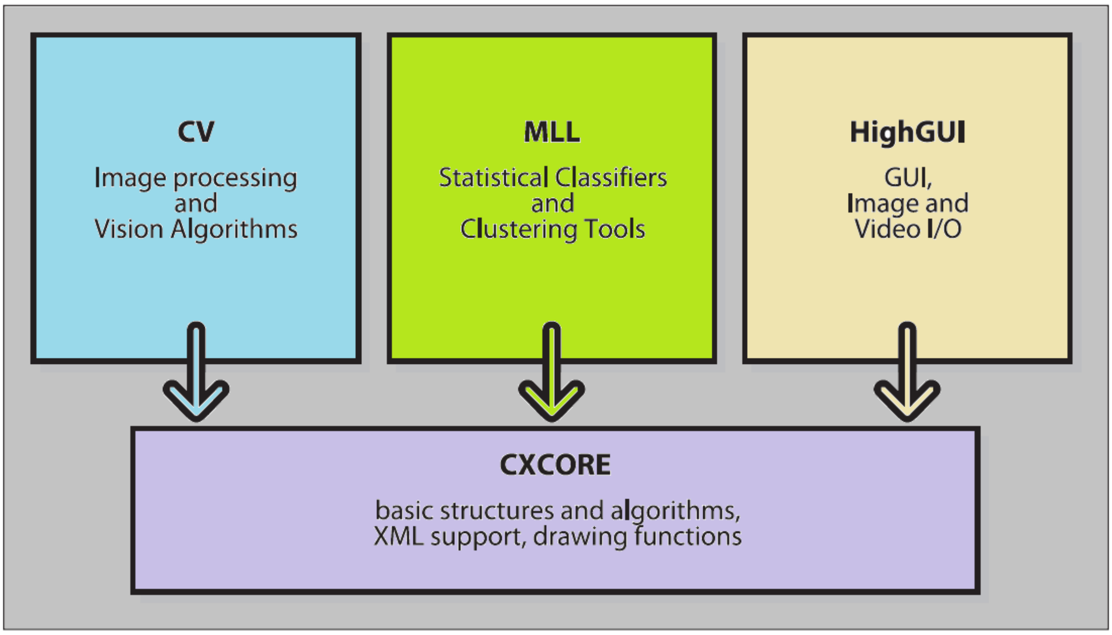


Figure 22 - The basic structure of OpenCV

When using OpenCV, an important data type must be understood, is the IplImage data type. IplImage is the basic structure used to encode what we generally call “images”. CvMat is the OpenCV matrix structure. Though OpenCV is implemented entirely in C, the relationship between CvMat and IplImage done in C++. For all intents and purposes, an IplImage can be thought of as being derived from CvMat. Therefore, it is best to understand the (would-be) base class before attempting to understand the added complexities of the derived class. A third class, called CvArr, can be thought of as an abstract base class from which CvMat is itself derived.

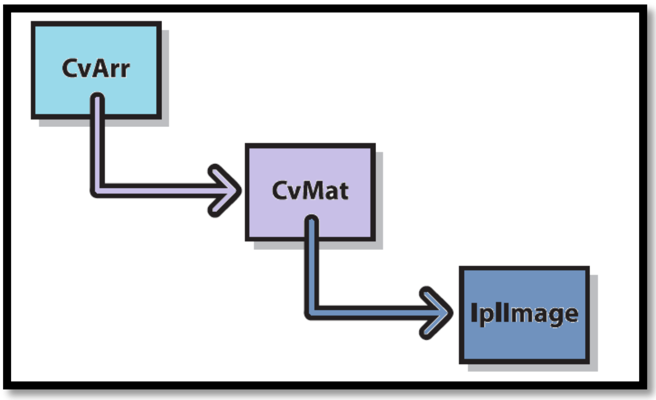


Figure 23 - Structure used in OpenCV

3.2 -Physical Architecture

In order to be possible to detect the ball and the players, several possible architectures could be implemented. In order to obtain a perfect detection of the ball location the positioning of the cameras is extremely important. My proposed architecture should be composed of 6 cameras, 3 in each part of the field. With that, a triangulation could be made, extracting in an easy way the 3 coordinates of the ball, and overcome the issue of occlusion caused by the players.

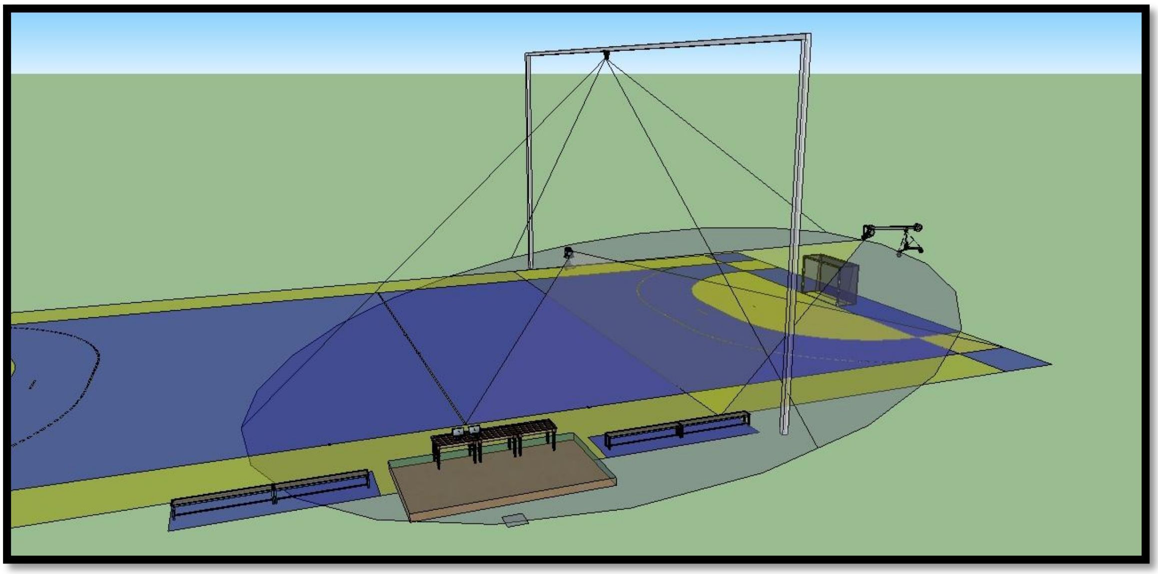


Figure 24 - Ideal physical location of the cameras

As is possible to see from the figure 24, the 3 cameras are in a strategic position, and when combined together make the extraction of the 3 coordinates far more easy than with a single camera.

Of course it must be taken in consideration the total cost of the system, so a smart choice must be done. Because we only have 2 cameras available, the solution found was to put each one over each part of the field. With the disposal of cameras seen in figure 25 , it will be very difficult to detect the ball, essentially due to distance from the camera to the field. In addition to the difficulty in detection of the ball, another even greater challenge is posed to determine the height of the ball to the ground. To do this, propose a solution, as will the next chapter.

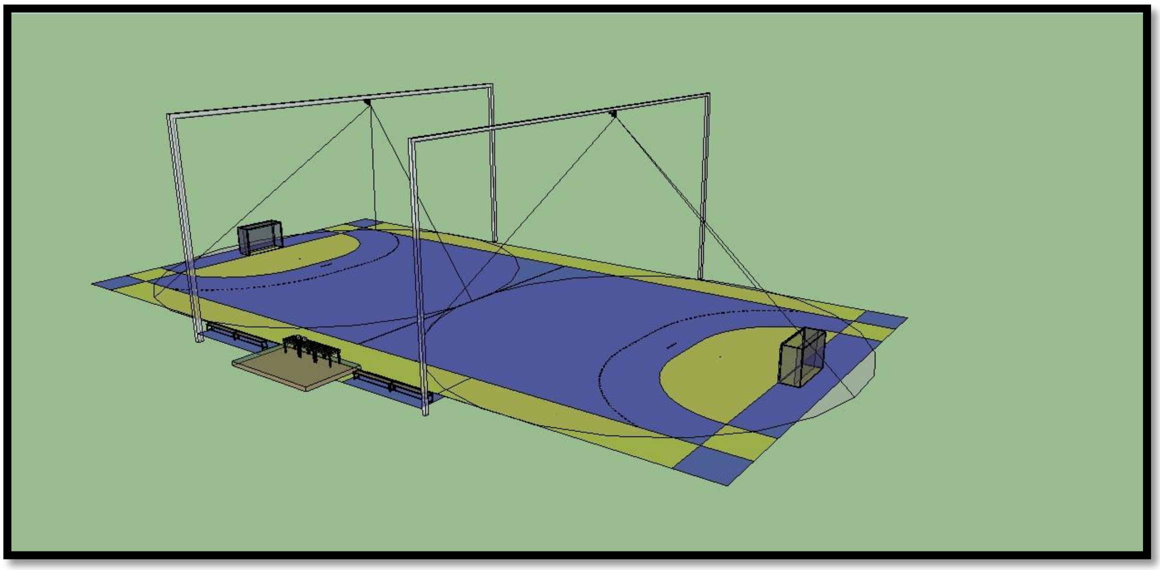


Figure 25 - Real location of the cameras

3.2.1 -Real-Time architecture

The main objective of this project is that the system works in real-time. So the big picture of the physical connections is presented in the next figure.

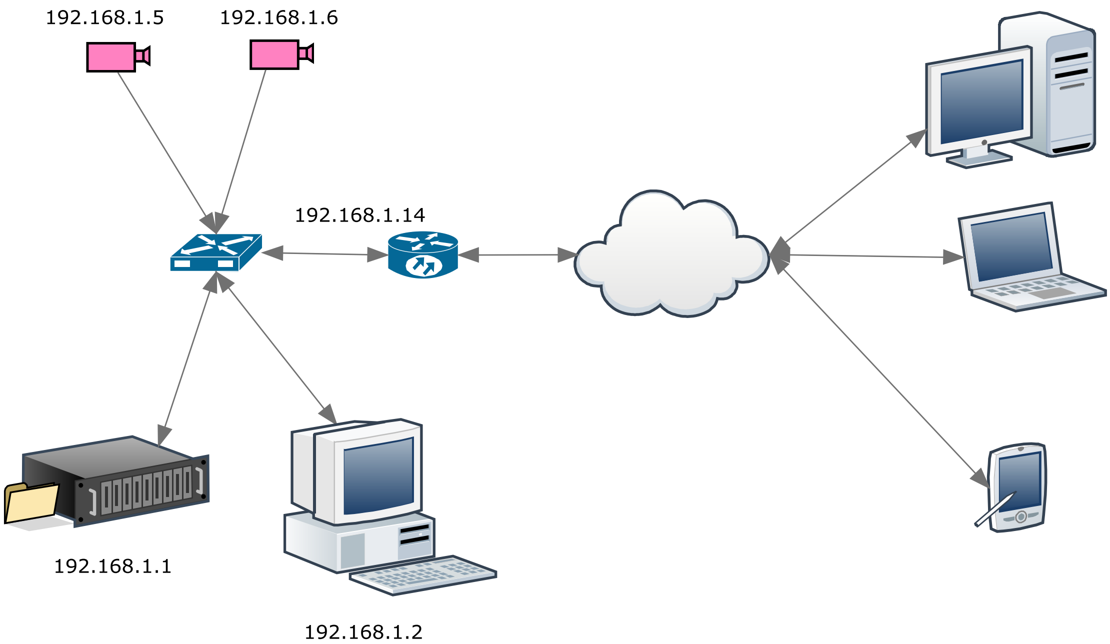


Figure 26 - Private network of the system

To make a good system, I think that a network with 16 hosts is good enough. That addresses already include a possible change in the topology of the network caused by an increase in the number of equipment in the system.

The private network is 192.168.1.0 with a sub mask of 255.255.255.240. I give the **file server** the IP 192.168.1.1. **Workstation** have the IP 192.168.1.2 and for an external user to have access to it I defined the Port 8001 to be used for NAT purposes. The two **cameras** of my actual system have the following IP addresses: 192.168.1.5 and 192.168.1.6. I dedicate some IP addresses to a possible increase of equipment's in the system. So the some IPs were reserved for a possible implementation of two more workstations(192.168.1.3 and 192.168.1.4). The remaining free IP addresses are reserved for an increasing in the number of cameras(192.168.1.7-192.168.1.13). The **default gateway** of this network will have the IP address 192.168.1.14. The Router will allow outside devices to access the server, thus having the ability to view the file results. The public IP of this network should be given by the ISP. The next figure shows a sequence diagram of the system.

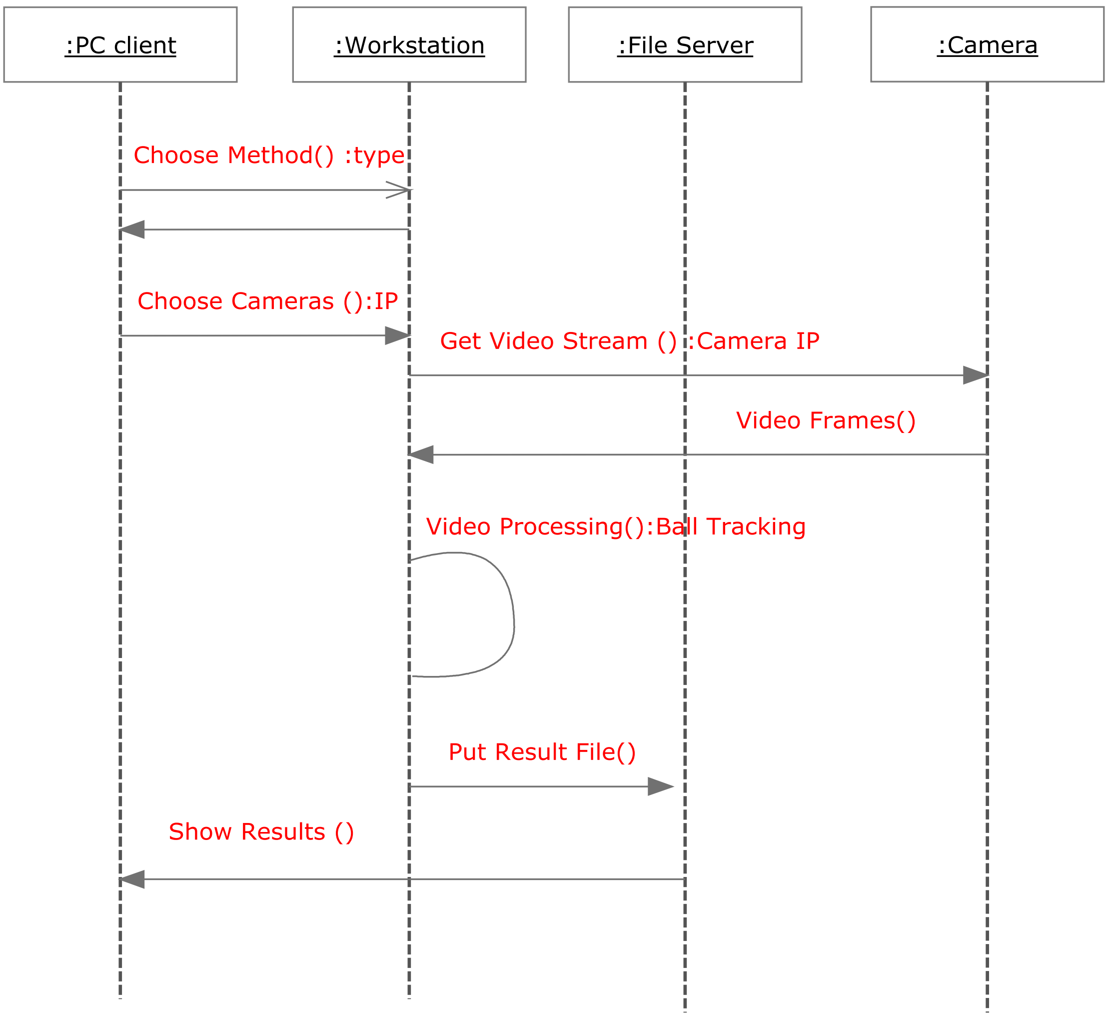


Figure 27 - Sequence diagram

The sequence diagram works as follows: When the user wants to automatic track the ball of the game he must enter the public IP address of the router, which is given by the ISP, and the port associated to the workstation which is 8001. For example, if the public address of the router is for instance 62.48.121.2, the client should type in the browser the following:

http://62.48.121.2:8001

Then the NAT protocol would create a table to allow the access to the workstation. An example of a packet from the source to the destination can be seen in table. Then a web page should appear asking the client to insert the method and camera IP for detecting the ball. Then the workstation starts the streaming of video frames from the camera. The next step is the application of the chosen method in the frame, in order to detect the position of the ball. From that moment, the workstation starts to send a file with the position of the ball to the file server. Finally the File server send the data to the Client, in order to be seen by him.

Table 3 - Conversation between the client and the workstation using NAT protocol

ACTION	PACKET	
	SRC IP	**DST IP**
PC-> Router	Client IP	62.48.121.2:8001
NAT	62.48.121.2:8001	192.168.1.14:3030
Router -> Workstation	192.168.1.14:3030	192.168.1.2:3030
Workstation -> Router	192.168.1.2:3030	192.168.1.14:3030
NAT	192.168.1.14:3030	62.48.121.2:8001
Router -> PC	62.48.121.2:8001	Client IP

3.3 - Used Videos

Were used some videos acquired from two different pavilions. The first type of videos were obtained from the pavilion of FADEUP. The second type of videos were obtained in the pavilion of Algarve, in a real Game between two professional teams.

In the first type I only have video from on part of the field obtained by a single camera. In the second type I have videos obtained from two cameras corresponding to the two sides of the field. In this type, which correspond to a real game, I decided to divide the video based on attack rolls.

This gave origin to short excerpts of video with a few hundred frames each. I also used some video obtained from my Webcam to carry out some tests.

The description of the videos could be seen in the following table.

Table 4 - Videos used in this project

VIDEO	# FRAMES	DESCRIPTION	IMAGE

FADEUP		Video from the FADEUP pavilion with some passes and shots.	
A		RIGHT field side of Super Cup	
B		LEFT field side of Super Cup	

3.4 -Real world Transformation

An important step must be implemented in order the data makes sense. This step is the conversion of the measurements in the image to the real world values. The conversion isn't so easy as it looks.

A very important property of the usual pinhole model for camera projection is that 3D lines in the scene are projected to 2D lines. Unfortunately, wide-angle lenses (specially low-cost lenses) may introduce a strong barrel distortion, which makes the usual pinhole model fail. Lens distortion models try to correct such distortion**[AlvGom_09]**.

In the images obtained by this cameras, as we can see in the images along this Thesis, is possible to see that its always present some distortion in it, caused essentially by the barrel distortion induced by the lens. So the important thing here is the selection of reference points inside the central image, measure the distance between these two points and then transform those values to Real World meters values.

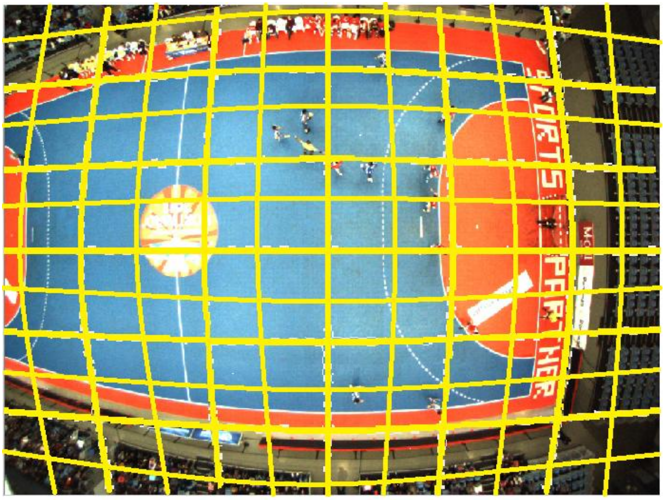


Figure 28 - Barrel distortion in the frame

3.5 -Color Segmentation

In the developed methods, an important feature will always be implemented, is the color segmentation of the image. I decided to detect Make the segmentation of the image in the HSV color Space, so that the colors are easier to separate.

The algorithm works as follows:

Table 5 - Color segmentation algorithm

1. Convert the image to HSV color Space
2. Split the image into 3 channels: Hue, Saturation and Value
3. With the mouse click times on the desired object in 20 frames
4. Three pixels values will be collected from the channels Hue and Saturation and Value
5. The average and standard deviation will be calculated
6. A Threshold will be applied in the original HSV image with the following ranges

$$HSVth = HSV(H \pm 2\sigma, S \pm 2\sigma, V \pm 2\sigma) \qquad (3.1)$$

An example of the three channel image can be seen in the following image, where a filter of the blue color is applied:

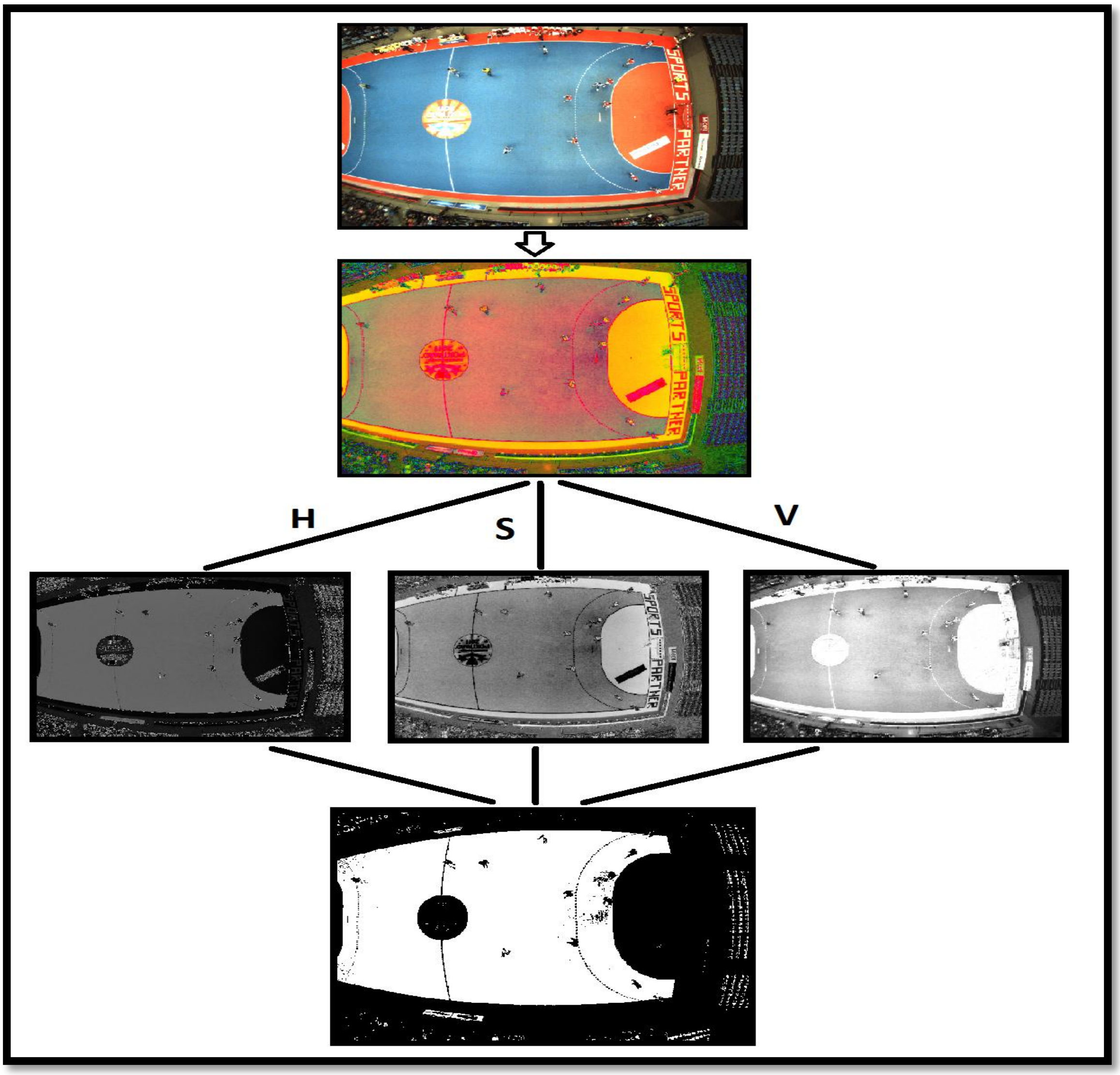


Figure 29 - Color segmentation based on HSV color space

From the figure 29 is possible to see the whole process of threshold the image with the color of the field. This procedure, will be used to segment the image based on the color of the ball.

Chapter 4

Applied Techniques and Developed Methods

To accomplish this job I decided to create 4 mainly methods. All methods are based on some of the following procedures: Static Background Subtraction, Adjacent Images Subtraction, colour segmentation, Smoothing Image filtering, morphological operations of Erosion and Dilation, Logical operations of AND and OR among others. The 4 created methods combine all some or all of those procedures. The objective is to choose the best combination of all. The first method is called **Static Background Subtraction with Colour Segmentation**, the second one is **Adjacent Images Subtraction with Colour Segmentation**, the third one is a combination of these two methods, in order to eliminate the false positives corresponding to the players members. The fourth method is also based in these the first two ones, nut its main objective is to overcome the issue of the missing ball obtained in the Adjacent Images Subtraction.

After the presentation of the techniques, it will be presented the develop methods.

4.1 – Difference between images

The idea to use this method is to subtract two images. The differences between them represent foreground objects. The area of the image plane where there is a significant difference between the observed and the background images indicates the location of the moving objects.

4.1.1 – Static Background Subtraction

The Static Background Subtraction model is created as follows: first, an empty field image is captured. Then that image is converted to grey scale image. The resulting image is smoothed to remove small noise. The same process is applied to every current image. Then is performed the subtraction between the frames. The result will be an image with the foreground corresponding to the objects.

```
void static_background()
{
        static_image="image from an empty field";
        current_frame=frame;
        obj_Image= current_frame-static_image;
}
```

The next figure show an image sequence of a background subtraction.

Figure 30 - Static Background Subtraction

Then I choose a value to make the binarization of the image. After some experience values, I choose the threshold value of 60, i.e., every pixel with a grey value higher than 60 will become 255 (WHITE), the others pixels whose values are below 60 they become 0 (BLACK).

When the subtraction is made the result image from the difference background method is obtained and it is shown in the next figure:

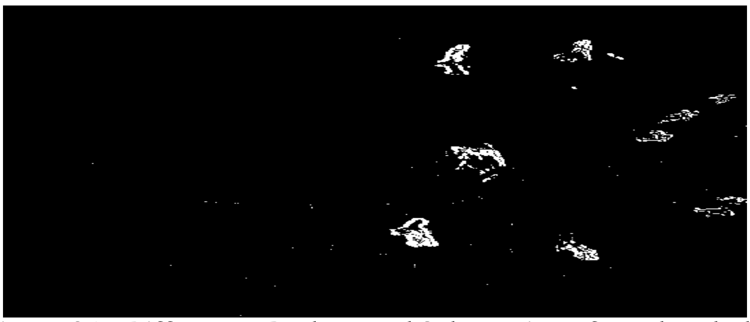


Figure 31 - Difference Background Subtraction after Threshold

4.1.2 – Adjacent Images Subtraction

This method consist in subtract adjacent images, and detect the objects that moved. Compared to the static Background Subtraction is has some vantages and some disadvantages. One advantage is that this method is more robust to changes in the environment, because the consecutive images will have a similar environment and the only detected difference will be the moving objects. The detected disadvantage is that if the target object remains stopped or moving with a low speed the subtraction will not show the object.

The subtraction method consists in the following algorithm:

```
if (frame_number>=1)
{
        current_frame=frame;
        diff (current_frame, last_frame); // a subtraction is made between the actual and the previous frame
        last_frame=current_frame;
}
```

The next image shows a sequence of an adjacent image subtraction.

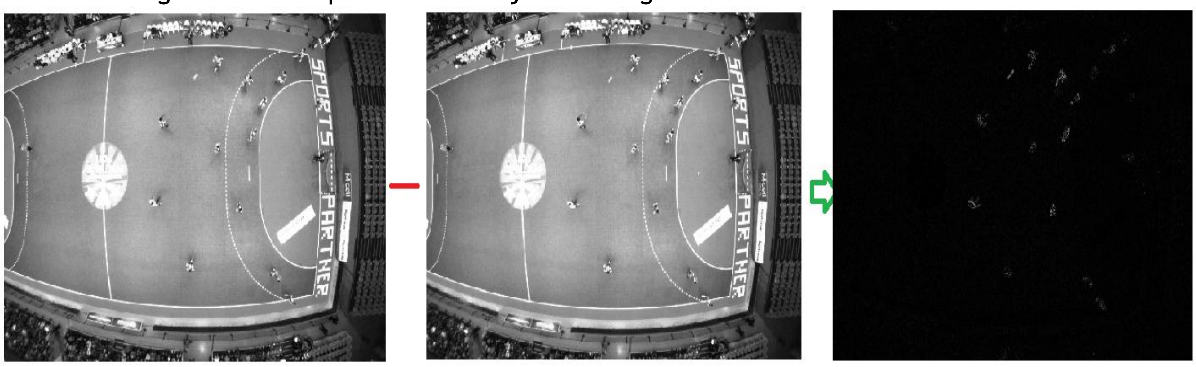


Figure 32 - Adjacent image subtraction

The fact that the subtraction only begin at the second frame is due to the fact that in the first frame the last_frame is empty.

The result of the subtraction will be only the objects that move between the frames.

4.1.3 – Edge Images Subtraction

From my research I found a work done by Xie [Xie_03], and I decided to test his method. The intention of this method is the decrease of the noise in the image, especially due to the changing in the illumination conditions. Taken in account that the ball has a few pixels, the noise is a big issue.

The technique works as follow: It uses a background image that corresponds to the field with no objects in it. The Static Background Subtraction model is created as follows: first, the background image is converted to the grey scale image. The resulting image is smoothed to remove small noise. An edge image is then generated by the Canny operator. The same process is applied to every current image. Then is performed the subtraction between the frames. The result will be an image with the foreground corresponding to the objects.

The threshold value for the edge will vary accordingly to the illumination conditions. So, for each video I will test some values for the threshold. I have made some experiences to see the best one for these particular images.

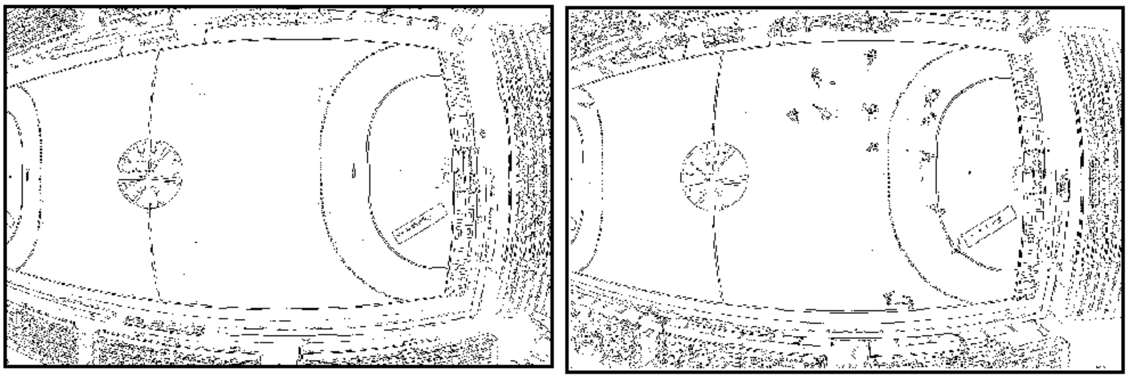


Figure 33 - Edge images

4.2 - Find Ball Candidates

The main idea is to find the ball in each image by the application of two filter parameters based on size and color of the ball.

4.2.1 - Size of the contours

Owning to motion effect, a moving ball usually appears larger than a static one.

By size, is easy to detect the possible players and ball candidates. I decided to make two ranges of values for those objects. Any other object with a different size will be discarded.

I begin this analysis for each particular object, and from the collected values I have defined the range of values for each one of them.

4.2.2 - Filtering by colour

The image is first converted to a HSV color space because the RGB color space is more susceptible to mistakes due to changes in the illumination. To perform the color segmentation by the colour of the ball is necessary to make a manually work. So the task consist in get some pixels of the ball by clicking over it in a frozen image. After that it is calculated the average of the extracted values, and a range is defined based on the median and the standard deviation. The range will be vary between [average – 2σ , average + 2σ], because it contains 95,4% of the obtained values.

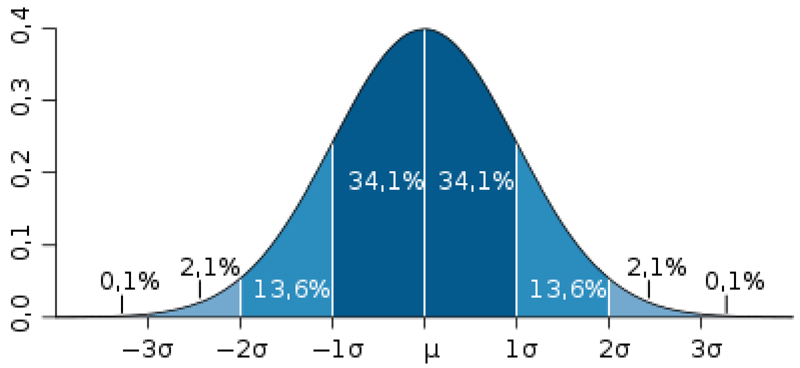


Figure 34 - Standard deviation diagram

Then an segmentation operation is performed based on the range of values manually obtained.

4.3 - Reconstruct 3D trajectories from 2D

Some works of 3D trajectory reconstruction are built based on multiple cameras located on specific positions. All the works done in the determination of the height of the ball consist in the triangulation of a set of cameras strategically placed in the field. In my work, due to the disposal of the cameras, the approach must be done in a different manner.

In this chapter the calculation of the ball height will be done.

I have developed a method consisting in the using the area of the ball. Physical equation will also be used. It is important to obtain a good estimation of the ball area.

This chapter presents a physics-based algorithm for 3D reconstruction. The 2D-to-3D inference is intrinsically a challenging problem due to the loss of 3D information in projection to the 2D frames. With the 2D trajectory extracted, physical characteristics of the ball motion are involved to reconstruct the 3D trajectories.

4.3.1 –Determining Ball Height

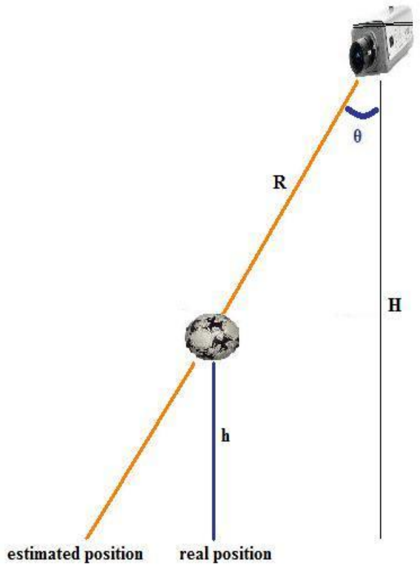


Figure 35 - Relation between the observed ball position and the real position

An initialization of some variables must be done. To do that, it has been done the collection of the ball area in particular positions. I decided to extract the ball area of the ball just in the center of the image, i.e., in a vertically position just under the camera.

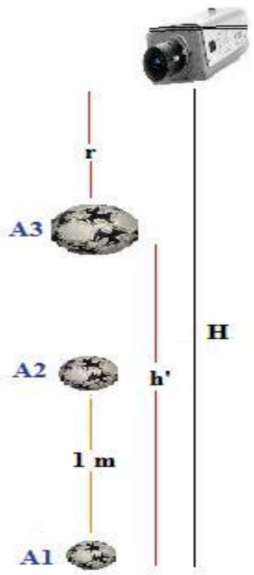


Figure 36 - Relation between the balls area and their distance to the camera

Taken in account that it is a linear relation between the area of the ball and the distance to the camera, I can make the follow relation:

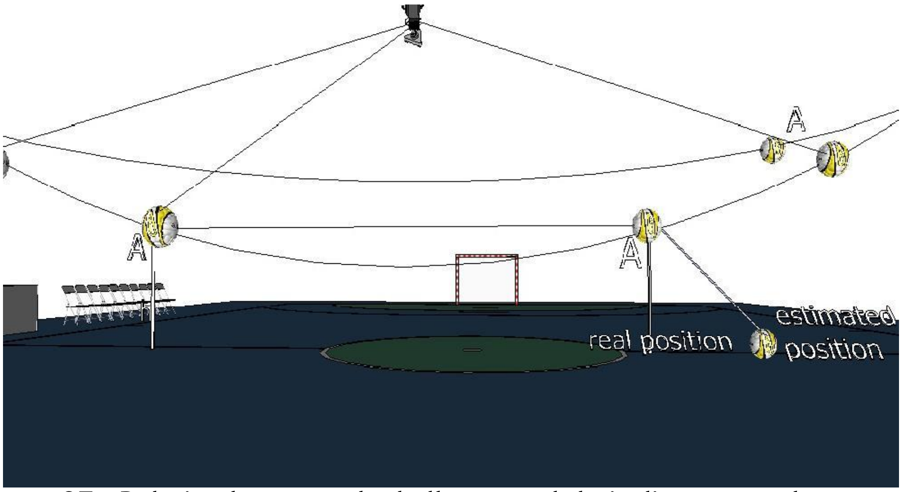


Figure 37 - Relation between the ball area and their distance to the camera

From figure 37 is possible to see that the ball's area is equal if the distance, let's say R, to the camera is the same for every position describe by a sphere with radius R.

4.3.2 -Radial Distortion

I found some studies and methods to solve this issue. All the methods prove that Software can correct those distortions by warping the image with a reverse distortion. Radial distortion can be corrected using Brown's distortion model [Bro_66]. Brown's model caters for both radial distortion and for tangential distortion caused by physical elements in a lens not being perfectly aligned. The idea is that straight lines in real life should remain as straight lines in an image. This is because the image just shows a projection of the scene onto the image plane, and a projection matrix will always preserve straight lines. Radial distortion will tend to curve straight lines. So if we can detect lines that should be straight in the images, then we can use an optimization routine to try to find the distortion coefficients that will make these lines straight.

$$Xu = Xd + (Xd - Xc)(K1r^2 + K2r^4 + \cdots) \qquad (4.1)$$

$$Yu = Yd + (Yd - Yc)(K1r^2 + K2r^4 + \cdots) \qquad (4.2)$$

Simplifying we obtain:

$$\begin{bmatrix} Xu \\ Yu \end{bmatrix} = \begin{bmatrix} Xd \\ Yd \end{bmatrix} + L(r) \begin{bmatrix} Xd - Xc \\ Yd - Yc \end{bmatrix} \qquad (4.3)$$

where: (Xu, Yu) denotes the undistorted image point, (Xd, Yd) denotes the distorted image point and (Xc, Yc) is the center of distortion. L(r) is the function defining the shape of the distortion model, which is usually approximated by a Taylor expansion as:

$$L(r) = K1r^2 + K2r^4 \ldots, \qquad (4.4)$$

$$r = \sqrt{(Xd - Xc)^2 + (Yd - Yc)^2}, \qquad (4.5)$$

where the set

$$K = (K0, K1, \ldots, Kn)^T \qquad (4.6)$$

gathers the distortion parameters. The complexity of the model is given by the number of terms of the Taylor expansion we use to approximate L(r), but according to Tsai [Tsai_87],for practical purposes, we can safely approximate the radial distortion equation by using only the first term of the infinite series:

$$\mathrm{L(r) = 1 + Kr^2}. \qquad (4.7)$$

We finally have:

$$\begin{bmatrix} Xu \\ Yu \end{bmatrix} = \begin{bmatrix} Xd \\ Yd \end{bmatrix} + (1 + Kr^2)\begin{bmatrix} Xd - Xc \\ Yd - Yc \end{bmatrix} \qquad (4.8)$$

To undistorted the image is necessary to choose some object in the image that we already know its characteristics in the real world. I decided to choose an line of the field to obtain the K value.

The k value will be calculated on the fly. I decided to capture 7 points on the image to better determine the k value.

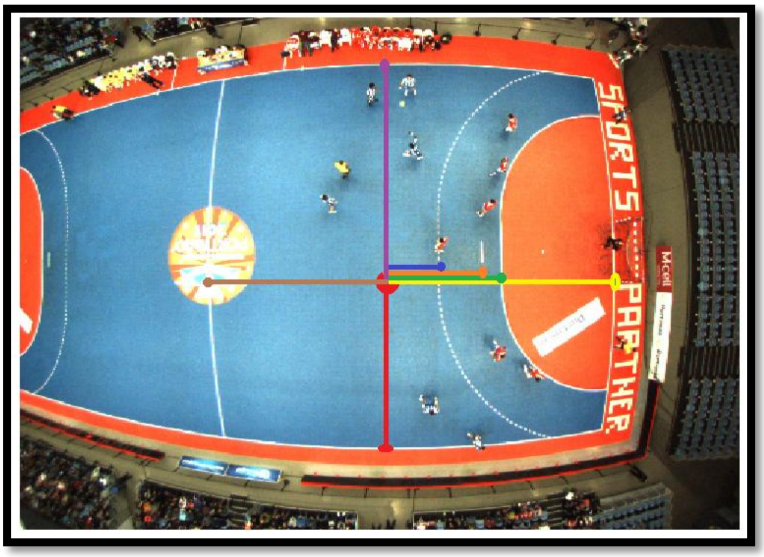


Figure 38 - Reference points used to determine the coefficients of radial distortion

I drew some lines in the image with different colors. They starts from the center of the image until one of the points to be measured.

(Xc,Yc)=(512,384); Here is the calculation for the yellow point.

Table 6 - Conversion from distorted to undistorted image

POINT	XU	XD	YD	r^2	K
yellow	825	835	396	32976	$2{,}978^{-5}$

After the calculation for all the values an approximation to the k values was made:

$$K \equiv 3{,}0^{-5}$$

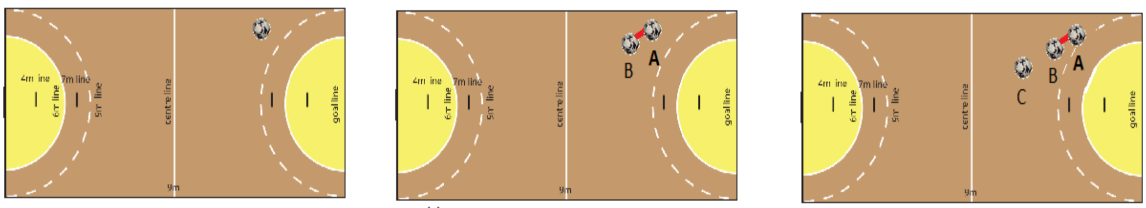
The final equation to undistorted image is:

$$\begin{bmatrix} Xu \\ Yu \end{bmatrix} = \begin{bmatrix} Xd \\ Yd \end{bmatrix} + 136 \begin{bmatrix} Xd - 512 \\ Yd - 384 \end{bmatrix} \qquad (4.9)$$

4.4 – Estimate ball position

With this technique the future ball location will be estimated. Predicting the ball trajectory, will improve the processing speed.

If the ball are detected in two consecutive frames, is easy to estimate a possible position for the ball in the next frame. First an Adjacent Image Subtraction is performed until the ball is detected.



a) **INITIAL POSITION** b) BALL SPEED DETERMINATION c) ESTIMATION OF BALL POSITION

Figure 39 - Ball position estimation

Is possible to see from the figure 39, that the ball is first located in the left frame. That will be the position A. In the next frame, the ball is again located at position B. With those two positions, it will be possible to estimate the position of the ball in the next frame.

The estimation algorithm will work like that:

First a Window will be defined by the user in the area of the ball.

Figure 40 - Possible ball position in the next frame

It will be then applied the AIS algorithm with color segmentation only in that window to search for the ball. Once the ball is detected, and because that would be the first time the ball is detected there wouldn't be any reference velocity value. Because of that, the region of analysis in the next frame, should be a big region based on the "maximum" velocity of the ball and all possible directions.

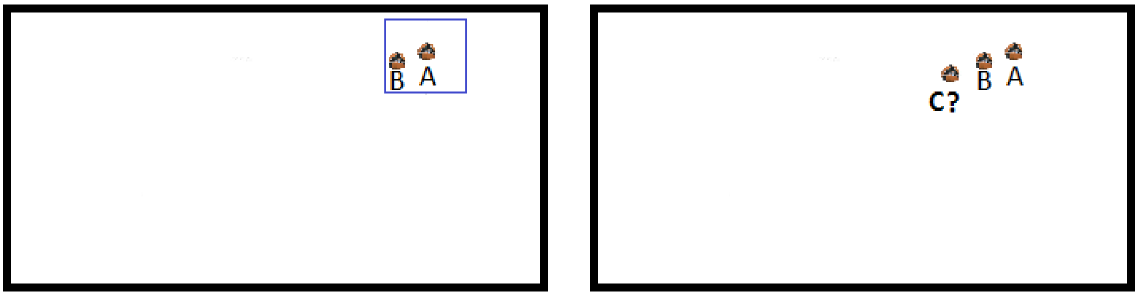


Figure 41 - Ball position estimation

Looking into the left image in the figure 41, is possible to see that the ball in the next frame, represented by B, is actually inside the window. Is then gathered the position of the ball by the application of the AIS with ball color segmentation. At this stage we have two points for analysis, the point A(Ax, Ay) and B(Bx, By). To estimate the next ball position is necessary to make some mathematical and physical analysis. The first thing to do is to convert this coordinates that correspond to an distorted image to coordinates of an undistorted image, then the distance is calculated, and the coordinate of that estimation point is then converted again to distorted environment to be able to define the window around it. That process can be seen in the figure 42. All this process starts by undistorted the points coordinates. This is done by the application of the equation obtained in the XYZ chapter.

$$\begin{bmatrix} Xu \\ Yu \end{bmatrix} = \begin{bmatrix} Xd \\ Yd \end{bmatrix} + 136 \begin{bmatrix} Xd - 655 \\ Yd - 345 \end{bmatrix} \qquad (4.10)$$

Then the next step is to determine the distance between this two points:

$$d = \sqrt{(Bx - Ax)^2 + (By - Ay)^2} \qquad (4.11)$$

At this point I only now the average speed of the velocity of the ball, so I decided to consider that the ball will travel the same distance and the same direction through the next frame, like we see in the right image of the figure 41. So the coordinates of the point C will be:

$$Cx = 2Bx - Ax \qquad (4.12)$$

$$Cy = 2By - Ay \qquad (4.13)$$

This coordinates take in account that the velocity will be constant between the points A and C. The velocity of the ball will be :

$$v = \frac{d}{\Delta t} \qquad (4.14)$$

Determined by the distance and the elapsed time between the two frames. Because this video has a rate of 30 FPS the elapsed time between two frames will be 33,3 ms.

The next step is to perform the estimation of the ball position in the next frame by the application of the equation 7 and 8. After the estimation is done, is necessary to “distort” again the points coordinates of the estimation ball position. This is accomplished by the following equations:

$$Xd = \frac{Xu+89080}{137} \qquad (4.15)$$

$$Yd = \frac{Yu+46920}{137} \qquad (4.16)$$

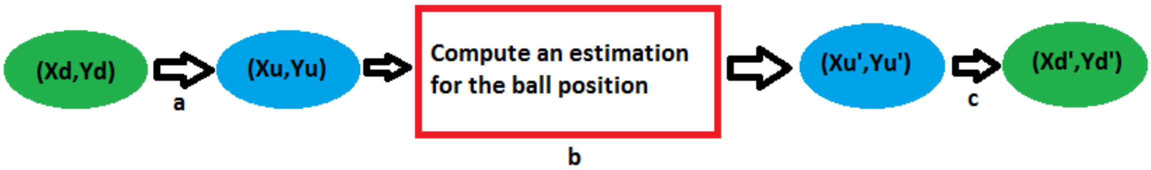


Figure 42 - Process coordinates estimation

4.5 - Static Background Subtraction with Analysis by Size and Color

From my research this method is by far the most used one. Is an easy and efficient way to identify objects in an image. To perform this method, is necessary to grab an image of the empty field. If that in not possible, like perhaps in an highway, is necessary to grab several images, and construct on possible empty image background, using combined information of all images. In my case, I could grab an image of the empty field, so it was a good start. After that, is performed a subtraction between the current and that image. Both images are previously smoothed to reduce the noise. The next step is to collect the colour of the ball and perform a segmentation of the image by that colour. Finally, an Logic AND operation is made between those two images, and the final result should be the ball and some possible ball candidates. The algorithm can be seen in figure 44.

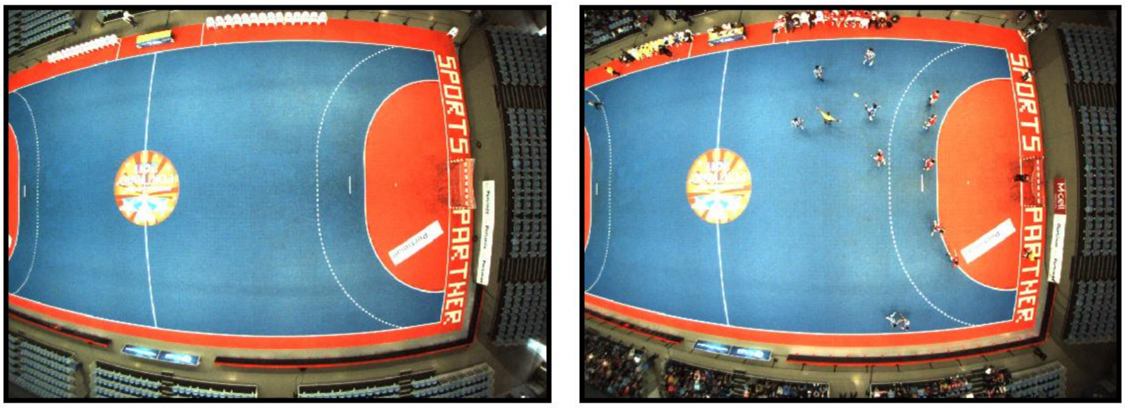


Figure 43 - In the left image is possible to see the empty field of the pavilion; in the right image is possible to see an real image from the game

Is possible to see in the figure 43, an empty image of the field. That image will be used as background for the subtraction. Because the developed algorithms only works in a grey level, the images must first be converted to grey.

Figure 44 - Grey images of the last figure

After the acquisition of the images shown in the last figure 44, the next step is to subtract then. After the subtraction is applied a threshold to the image.

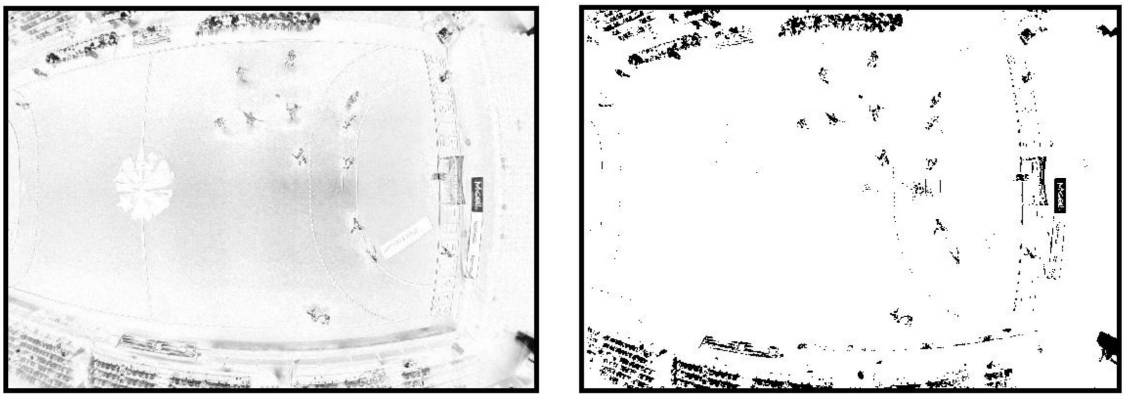


Figure 45 - The left image is the result of the subtrction; The right image is the Thresholded left image

Is possible to see in the right image from the figure 45 the result of an application of a segmentation to the image with a threshold value of 60. As explained in chapter 2, the threshold value is very important to the detection works. In this project I decided to choose an Threshold value manually. I tested several different values, and then I choose the one I thought was the most appropriate. In the next figure is possible to see, several images resulting from 4 different Threshold values.

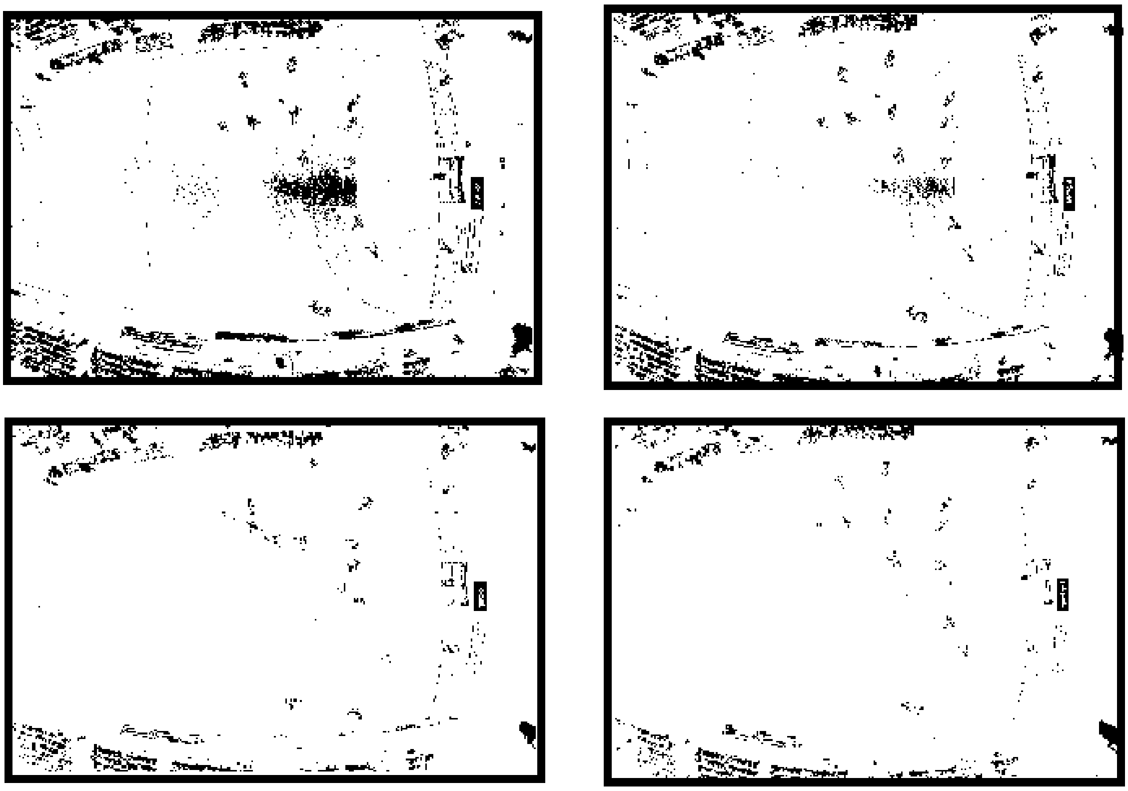


Figure 46 - Different Threshold values of 40, 50, 70 and 80 applied to the image

From the figure 46 is possible to see that depending of the applied Threshold value, we can obtain very different image segmentation. From a 40 threshold value, it is possible to see that there is too much noise in the image, so the solution is to increase the threshold value.

As is seen in the figure 46, when increasing the value of threshold, some important information could be lost.

The next step is to eliminate the region around the field and some noise. The elimination of the surroundings, is extremely important, because that would interfere in the application of the algorithm.

After that is applied an dilation operation to join the objects into BLOBS.

Figure 47 - Cleaning the surroundings of the field

After that, some morphological operations are applied to clean the noise and to join the players members.

Until here, only a part of the process was done, parallel to that is necessary to segment the image by the color of the ball. The process to get the color of the ball was explained in the point 4.2.1.

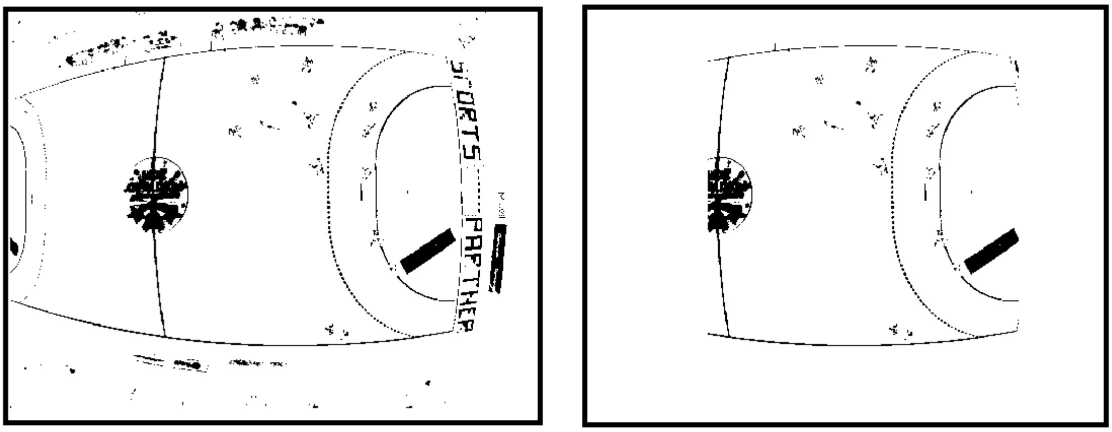


Figure 48 - Color segmentation and surroundings cleaning

The next step in this method is the application of the logical operation AND between these right images of the last two figures. After that, it will be filtered objects that have the size of the ball. The algorithm of this method can be seen in the next figure.

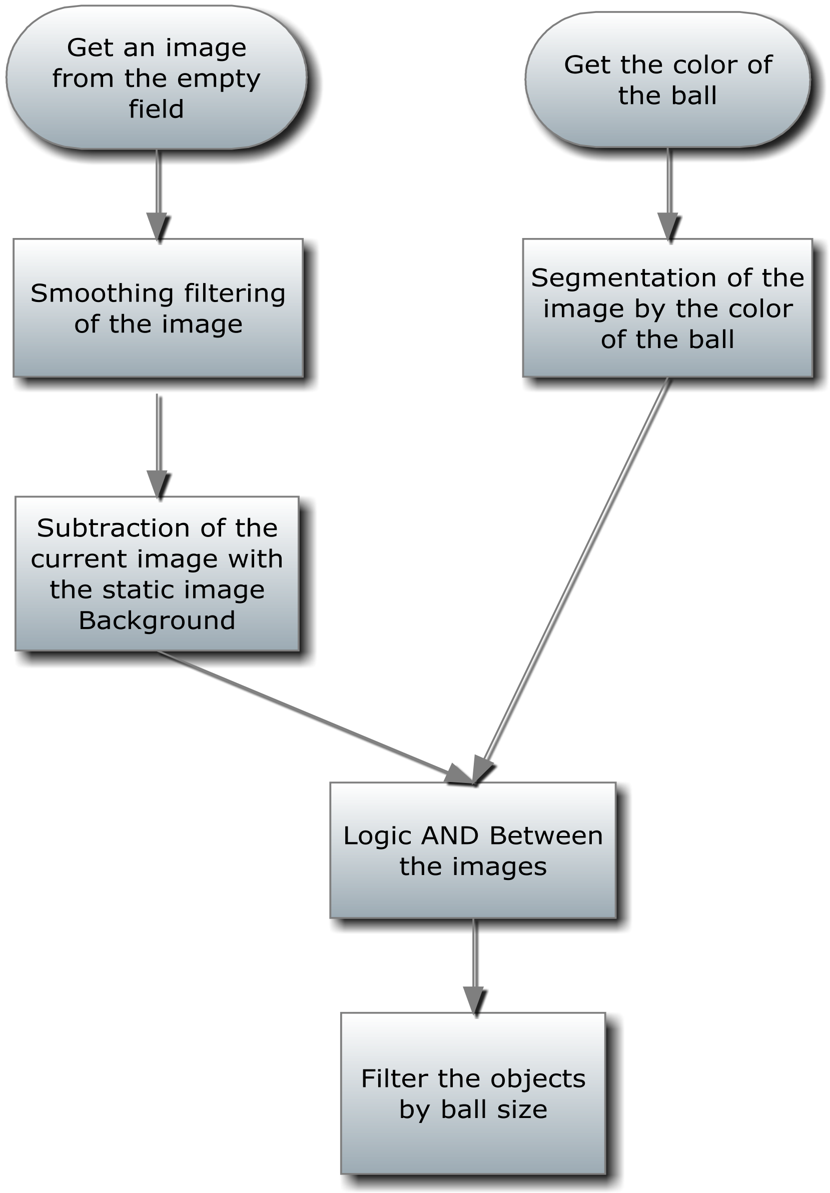


Figure 49 - Static background subtraction with color segmentation

4.6 - Adjacent Images Subtraction and Analysis by Size and Color

The last system is tremendously dependent on the illumination conditions. Even worst, if an some object like a piece of paper is in the field and it have the same area and color of the ball, this method will always “think” it is a ball.

In the method developed in this point, an adjacent image subtraction is performed. The main idea is to overcome some of the problems of the Static Background Subtraction, originated by the changes in the illumination. When two adjacent images are subtracted, the changes in the illumination aren´t so noticed. Especially with this video, which have 25 FPS, which gives a 40 mili seconds between frames. The idea here, is also to use the color segmentation of the image, and combine that with the result from adjacent images subtraction.

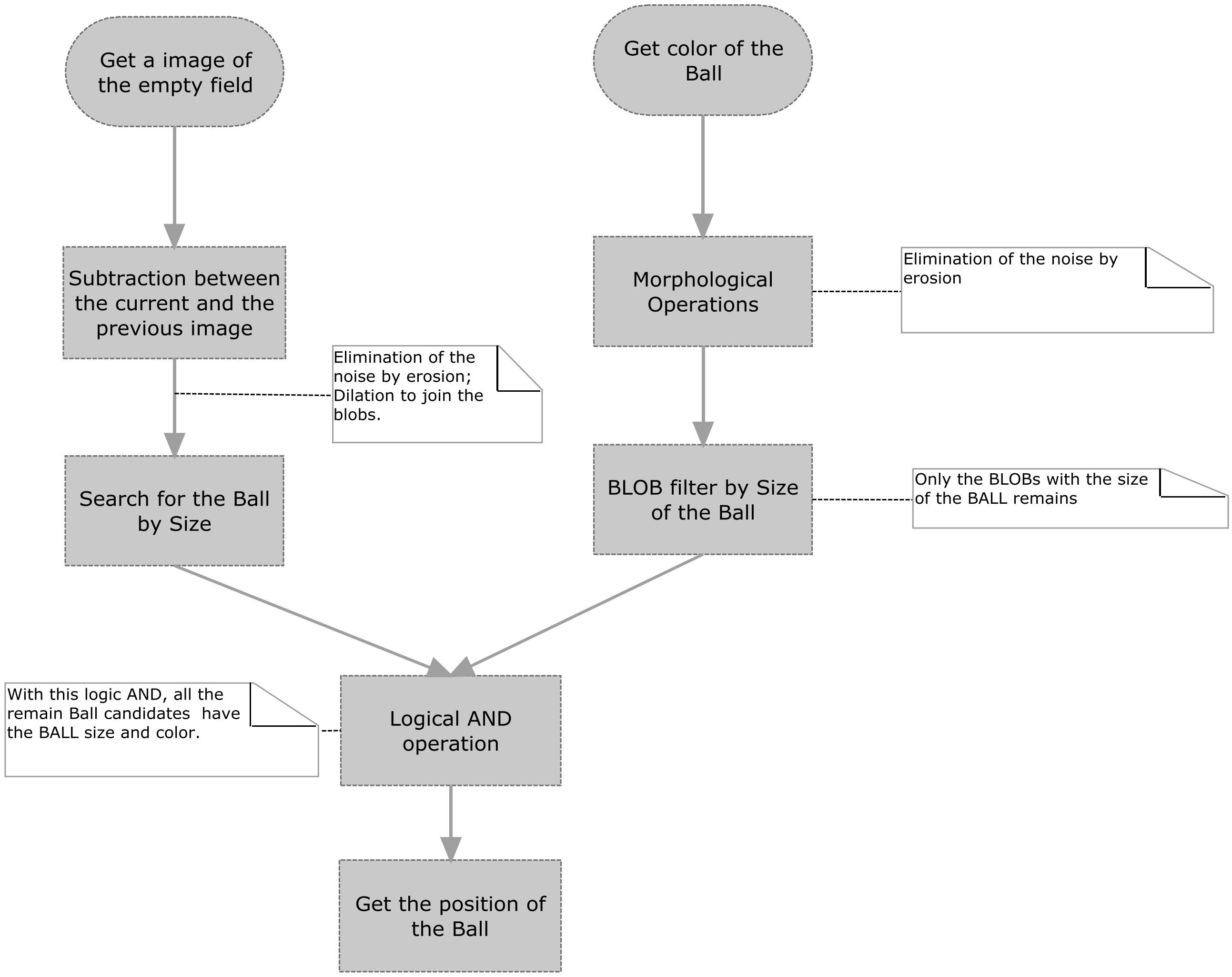


Figure 50 - Adjacent images subtraction with color segmentation

In the next figure is possible to see two adjacent images. Current frame of the video is subtracted by the previous one. First they are converted in grey.

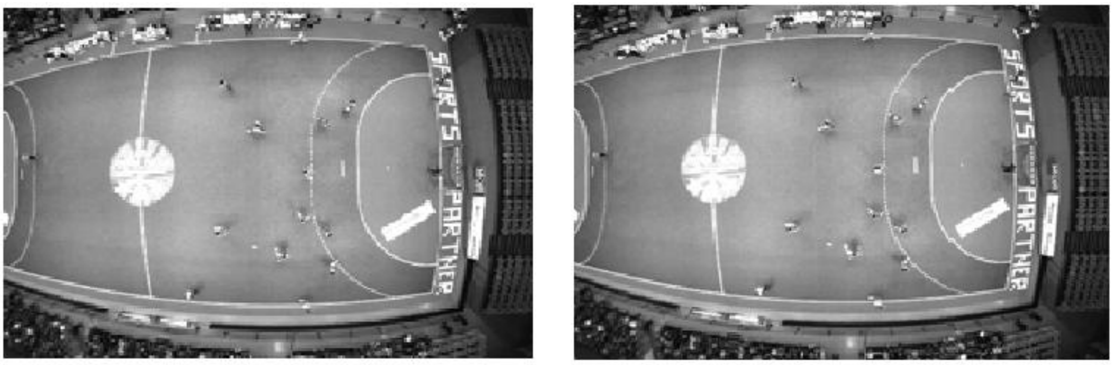


Figure 51 - Adjacent images

After the subtraction, a threshold to the image must be done. That procedure can be seen in the figure 53.

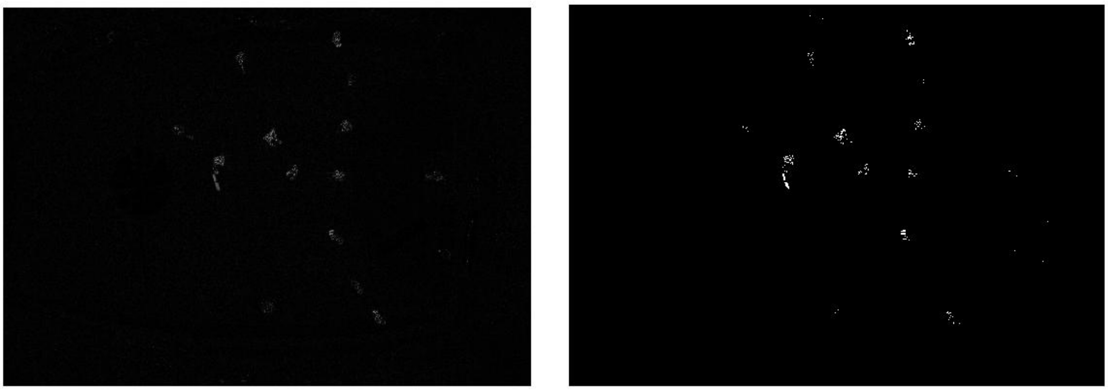


Figure 52 - Threshold of the image after the subtraction

After the Threshold of the subtracted image is necessary to segment the original image by the color of the ball. That procedure can be seen in the next figure.

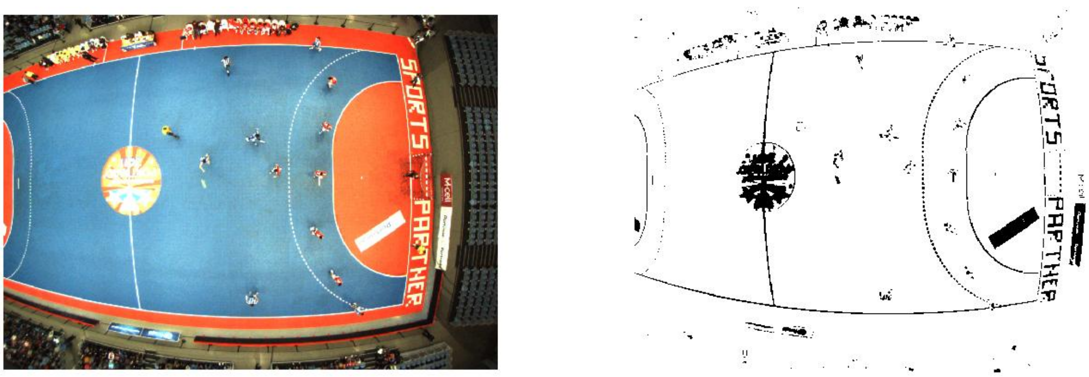


Figure 53 - Binarization of the image based on the color of the ball

I will invert the color of the pixel to adequate to the color segmentation. Then is performed an logic AND operation between the images. That should be enough to identify the ball and reduce highly the false positives.

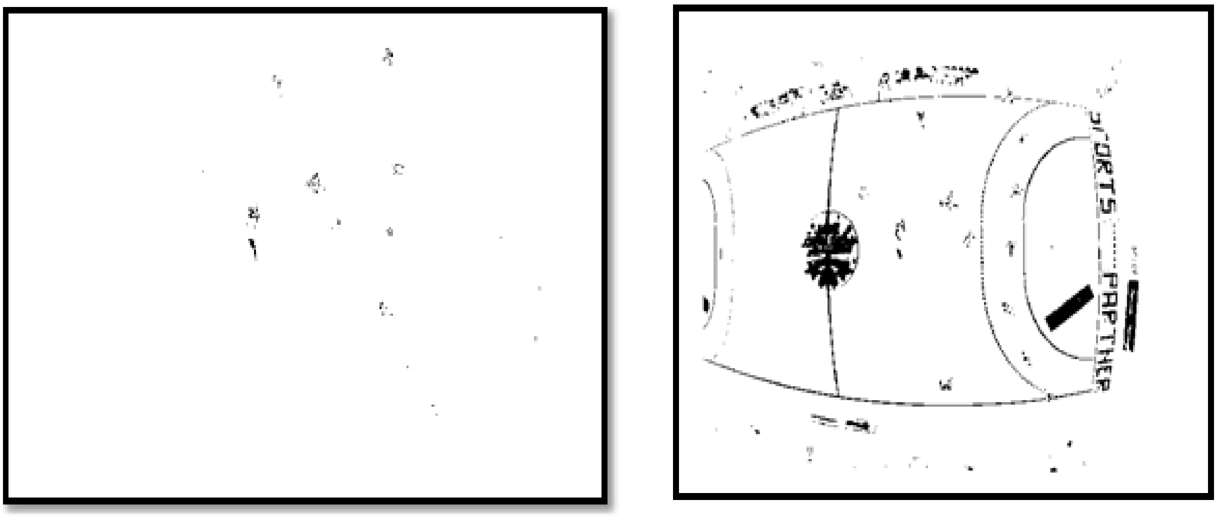


Figure 54 - Images to be combined: the left one of the result from the subtraction between two adjacent frames; the other in the right is the result of a color segmentation

With this method some issues of the last method are overcome, like the change in the illumination conditions, and is no longer necessary to eliminate the region around the field.

4.7 – Eliminating False Positives

Due to the need to reduce false positives, I decided to implement a new method. The idea was to use the first and the second method together, in order to eliminate the false positives. So, to identify the ball it will be used again an Adjacent Image Subtraction with color segmentation, and to eliminate the false positives an algorithm was used based on the detection players by a static Background Subtraction.

Because of the detection of some False Positives in method 2 due to members of the players, I decided to make a combination of those false positives with BLOBS Players resulting from Static Background Subtraction applied in the first method. The idea is to eliminate the false positives that corresponds essentially to the body members of the players.

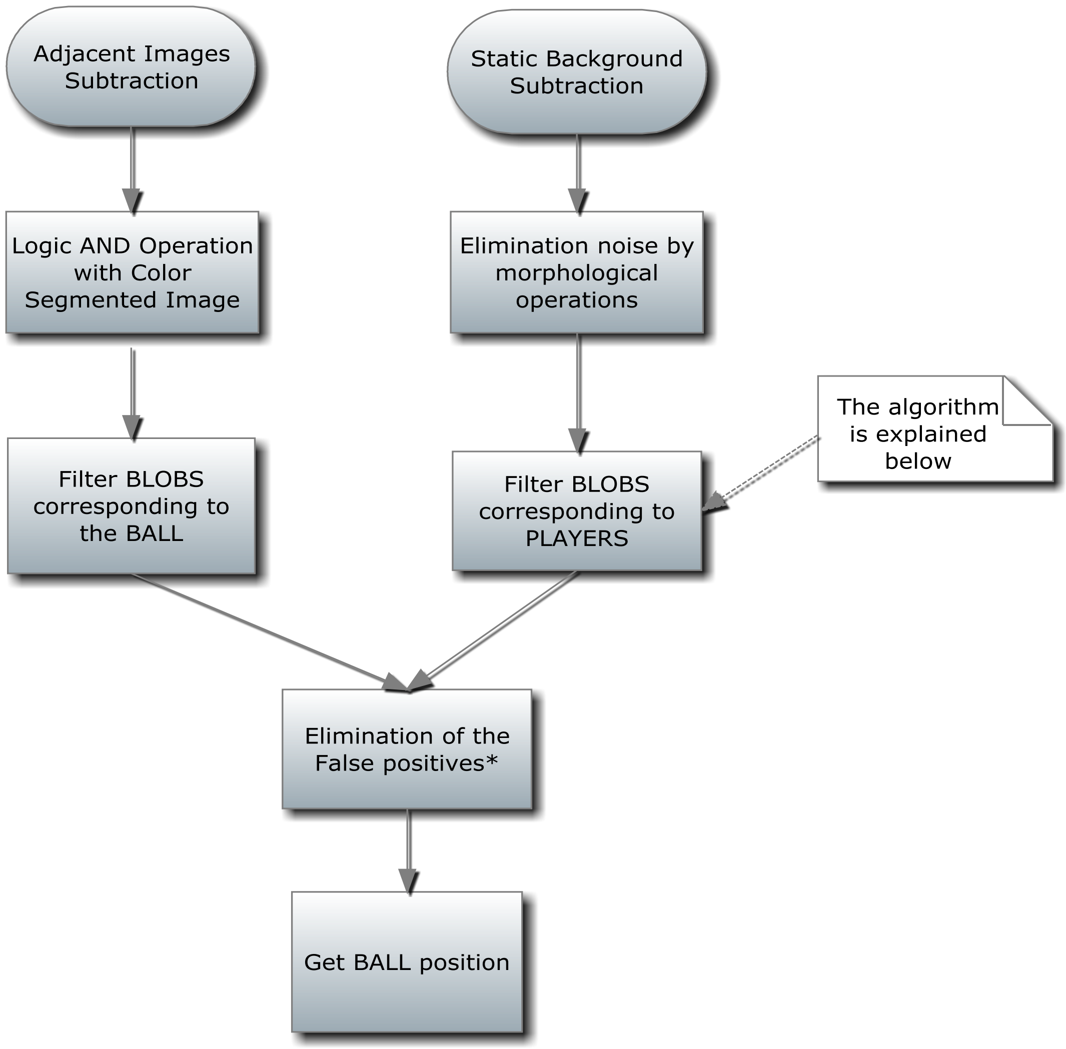


Figure 55 - Eliminating false positives algorithm

It is very important to detect the players in the field. So a Static Background Subtraction was performed. The idea was to after the subtraction, it was done a cleaning in the image to remove the noise. After that, a dilation process was performed. At this point the only points in the image should be the players and the ball, so it was needed to eliminate the ball from the image in order to perform another algorithm.

The combination of the images to eliminate the false positives should be analyzed more carefully. So I implemented an algorithm to perform that. After the identification of the ball candidates from the second method, is performed an comparison with the players BLOBS obtained from the Static Background Subtraction. The idea is to eliminate all the ball candidates that match’s with the players BLOBS.

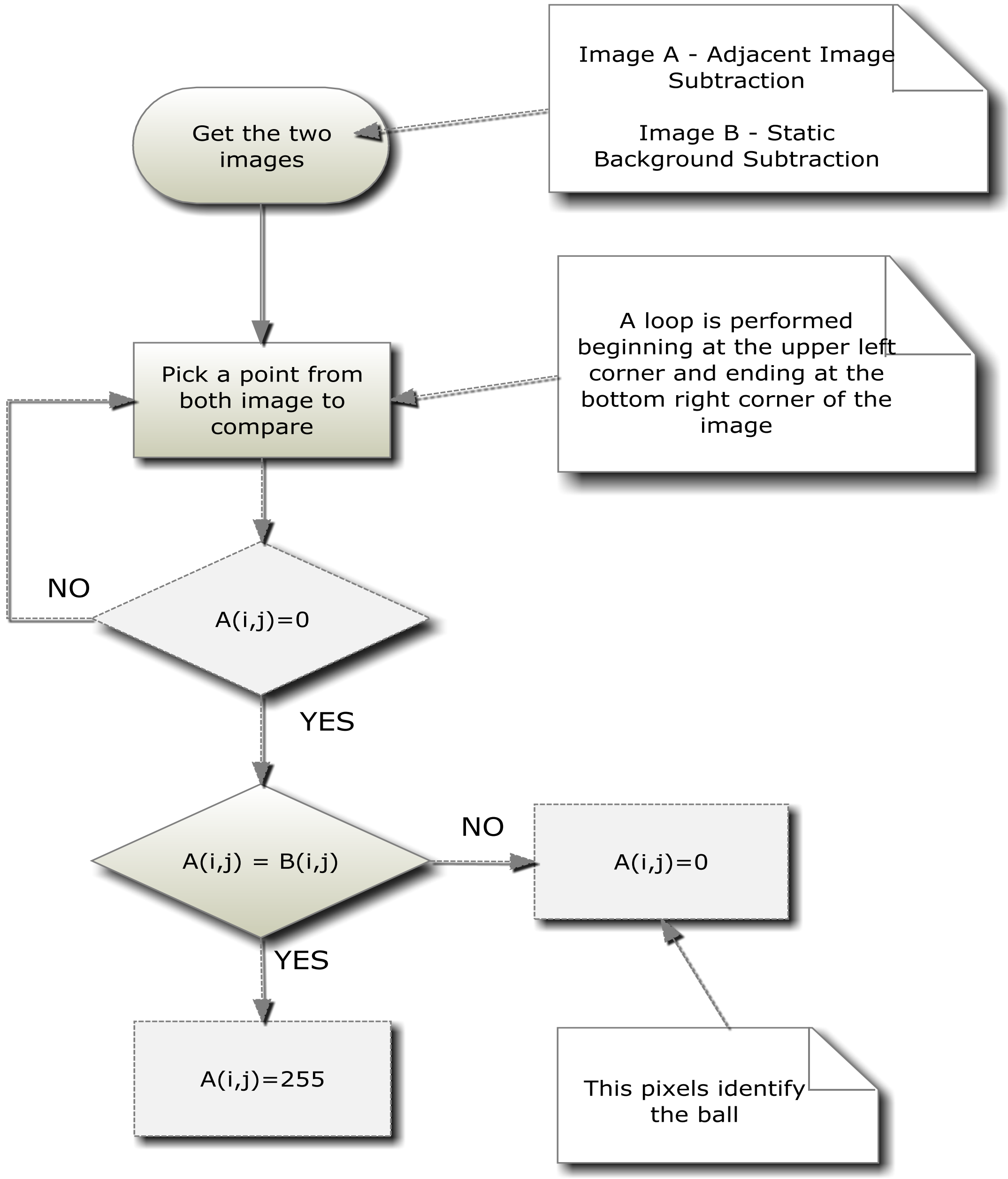


Figure 56 - Algorithm that eliminate the pixels corresponding to the players

The algorithm start with the application of the Ball's color segmentation obtaining the following image:

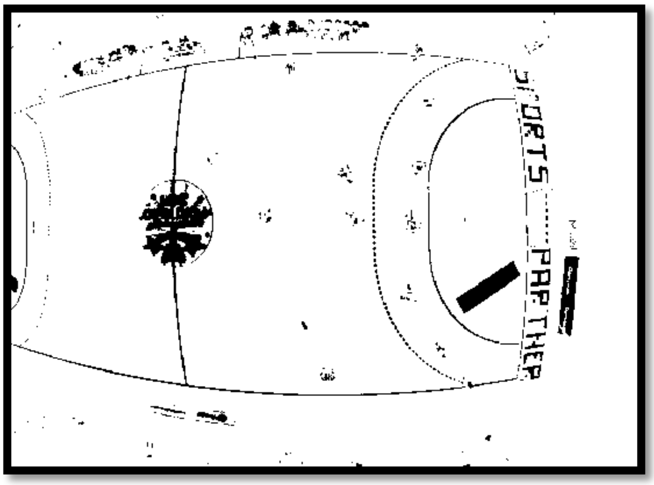


Figure 57 - Ball's color segmentation

Parallel to that is performed an Adjacent Image Subtraction and an Static Background Subtraction. The objective of the AIS is to identify the ball, while the objective of the SBS is to identify only the players.

Is important to note that the SBS applied in this method is quite different from the one applied in the first method. That is because in this method, the objective of the SBS is to identify the players instead of the ball. So the main differences will be the morphological operations and the filtering parameters.

Figure 58 - Static background subtraction

As is possible to see in the figure 59, there are several regions that can be ignored. So I decided to cut them. The result could be seen in the next figure:

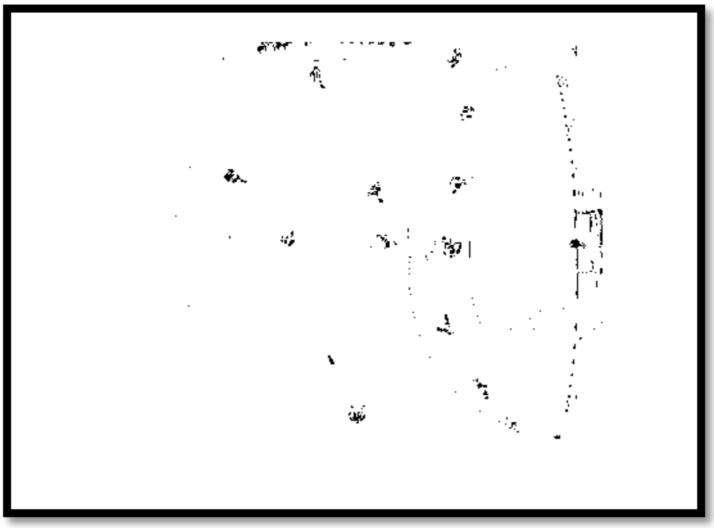


Figure 59 - Elimination of non-important regions of the image

After this, is necessary to eliminate all the noise of the image. A morphological operation of Erosion will be applied to eliminate noise and dilation will be performed to join players members. Then a filter by size is applied to obtain the players shape. The final result of this process can be seen in figure 61.

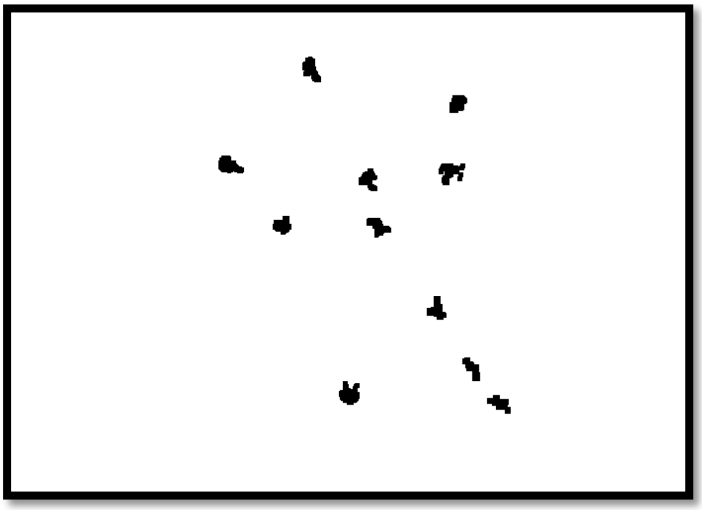


Figure 60 - Identifying players in the image

At this point we have an image with the players detected and an image with the moving objects with the colour of the ball. This two images could be seen in the next figure:

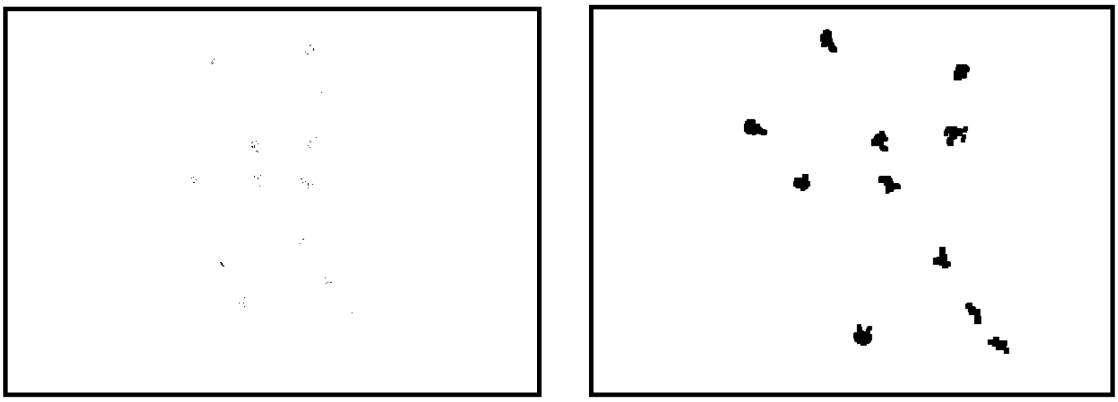


Figure 61 - - In the left image we have the adjacent images subtraction with color Ball segmentation; in the right image we have an identification of players by Static Background Subtraction

This method has great results regarding to the detection of the ball and the players. The objective to eliminate the false positives was accomplished. Because this method, relies on the Adjacent Image Subtraction, the ball will not be detected if it have a low velocity. The next method, I intend to overcome this issue.

4.8 – Region of Interest and Ball position Estimation

To overcome the problems from the previous method, I decided to developed a new method based in a switching between the methods Adjacent Image Subtraction and Static Background Subtraction with Ball color segmentation. The first one, will be used to detect the ball when its speed is higher than a threshold. The second one is used to detect the players when they are in possession of the ball, or to detect the ball when it has a velocity above a determined threshold. Furthermore is also necessary to define manually a region around the ball manually to start all the process. That window will have a fixed dimension, based on the maximum speed of the ball. The basic idea is to alternate every time the ball is identified or not. So I created a method capable of follow the ball until it disappears. In that moment the Static Background Subtraction is applied. Two possible things could have happened, the first one is that the ball run out of velocity and it is almost stopped; the second possibility is that the ball is in possession of a Player, so what the algorithm must do is to search for the closest player to the last position of the ball and start to track him. In this moment the size of the window should increase to 3 times the standard dimension. That's because the region of that window must be able to identify the player and some region around it. With that size, it must be possible to see the ball if that player decides to pass or shoot it. If the ball is identified again, is applied again the Adjacent Images Subtraction, and the window gets his default dimension.

In this method I decided to use other different approach to estimate the ball position in the frame. That is based in the definition of an window around the ball. Those window will be called Region of Interest, and is made by the User at the beginning of the application. After the window is defined, the method 3 will be applied inside. If the ball is detected with this mode, the following algorithm will be applied:

Table 7 - Moving window algorithm

1. A window is define with the mouse. The Left Button will define the Upper Left (Xa,Ya) coordinate, and the Right Button will define the Bottom Right (Xb, Yb) coordinates.
2. The 3rd method will be applied to search for the ball.
3. If the ball is detected, the coordinate of the ball inside the Window will be extracted.
4. The position of the ball (bp_x,bp_y) in the frame will be compared with the Width (Xb-Xa) and Height (Yb-Ya) of the Window, and the coordinate of the window will be Updated:

```
If(bp_x > width/2)
{
    Xa=Xa+width;
    Xb=Xb+width;
}
else
{
    Xa=Xa-width;
    Xb=Xb-width;
}
If(bp_y > height/2)
{
    Ya=Ya+height;
    Yb=Yb+height;
}
else
{
    Ya=Ya-height;
    Yb=Yb-height;
}
```

5. If the ball is not detected in the ROI, then the size of the ROI will double.
6. The Player will be the tracked object now, using a Static Background Subtraction
7. If the ball is detected inside the Roi, the it returns to its original size, and the 3rd method is applied again.

An example of this algorithm can be seen in the next figure.

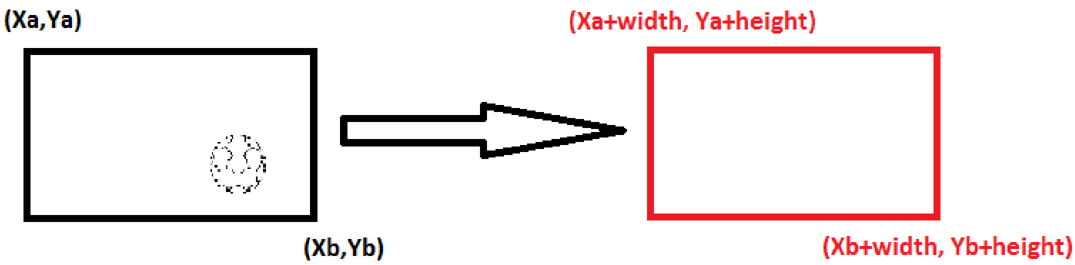


Figure 62 - Ball position estimation based on the ball position in the ROI

The algorithm can be seen in the next figure.

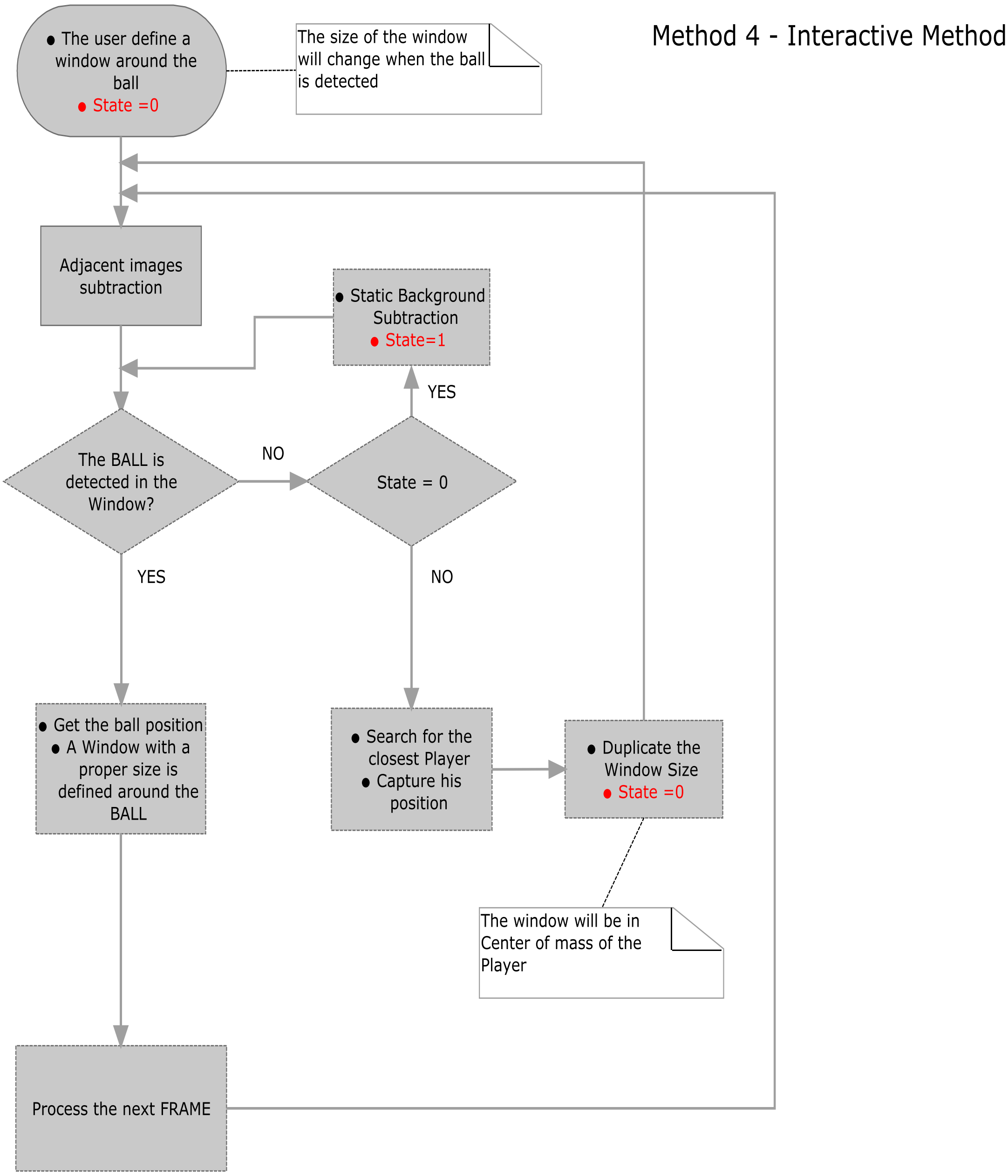


Figure 63 - Method Region of Interest and Ball position Estimation algorithm

I will show images obtained by the FADEUP videos. I start to define a window enclosing the ball. Then I apply the same approach as discussed earlier, that is a shape analysis and color analysis.

If the ball does not appear to increase the size of the window and applies STATIC subtraction. It is expected that the next position of the ball, and so put the window there. The ball should appear near the center. The position of the ball is always found, so when you stop seeing the ball we have to see which player is closest to the position of the ball. If the ball is not detected two cases can happen: Either the ball is stopped or it is in possession of any player. As such research is the first by a ball near the center of the window, if not detected any ball research is being done to find a player.

If the ball is found, is extracted from the position of the ball. In the next frame should be applied again to the same process with the large window, if the ball is detected can then calculate the speed of the ball. If this speed exceeds a certain value, the method reduces the size of the window and puts the window in the position expected in the next location of the ball, and the method is applied to the subtraction of adjacent images. If the ball is not detected, this indicates that this is in possession of any player. Then the search is carried out to be a player. If detected more than one player, then the search is the player's choice falls on one who is closest to the center of the window.

In the next frame will be an analysis of AIS, if the ball is detected is stored the position of the ball. In the next frame is performed the same method, and seeks to be detected if the ball is reduced and the window size determines the speed. begins to tud "as the initiator. If the ball is not detected, it backs to SBS and apply the method. Is then detected the position of the ball, and the player position as well. The forecasting system will work best if you get at least three consecutive positions of the ball.

Chapter 5

Result Analysis And Methods Evaluation

In this chapter it will be made a description of the results obtained by the application of the previous techniques.

All the methods will be analyzed according to some parameters like Ball Detection Rate, False Alarm Rate, Clean Detection Rate and Processing time. To the ball detection rate

I only consider the ball detection when the ball is visible, i.e., not partial or total occluded by the player's. The False Alarm Rate will indicate the rate that false positives appear in the frame. The other parameter related to false positives is the Clean Detection parameter, which indicates the rate of time the system only detect the ball. The last analyzed parameter is the Processing time for each frame.

I decided to show the results for each particular method, and for each kind of video. I decided to make two groups of videos, one are the videos from the Super Cup, and the others are the videos from the FADEUP.

The base for my evaluation is based in the following table:

Table 8 - Analysis criteria

SITUATION	THE BALL IS DETECTED	THE BALL ISN'T DETECTED
The Ball is present	A	B
The Ball isn't present	C	D

The results will be generated according to the following equations:

$$Ball\ Detection\ Rate = \frac{\sum A}{\sum A + \sum B} \times 100 \qquad (5.1)$$

$$False\ Alarm\ Rate = \frac{\sum C}{\sum C + \sum D} \times 100 \qquad (5.2)$$

There's another important parameter that can be determined in order to get the rate of false positives, i.e., the number of objects similar to the ball, that the system interpret as being the ball.

Table 9 - Clean detection parameters

SITUATION	ONLY THE BALL IS DETECTED	DETECTED BALL AND MORE OBJECTS
Ball is present and is detected	E	F

With this results is possible to construct another bases for analysis, which is the relation between those one, which I will call the **clean detection rate**.

$$Clean\ Detection\ Rate = \frac{\sum E}{\sum E + \sum F} \times 100 \qquad (5.3)$$

5.1 -Results from Static Background Subtraction with Analysis by Size and Color

This method is very effective in the search for the ball, because all objects presented in the current image that doesn't appear in the background image are very well identified. The worst part of it is that is very sensitive to errors with changes in illumination. Therefore, in our particular case, due to some changes in the illumination during the game, there are a lot of noise in the image. That originate some errors in the classification of the objects, and the number of false positives is also big.

Regarding from the last chapter where this method was introduced, here we will see the results of that method.

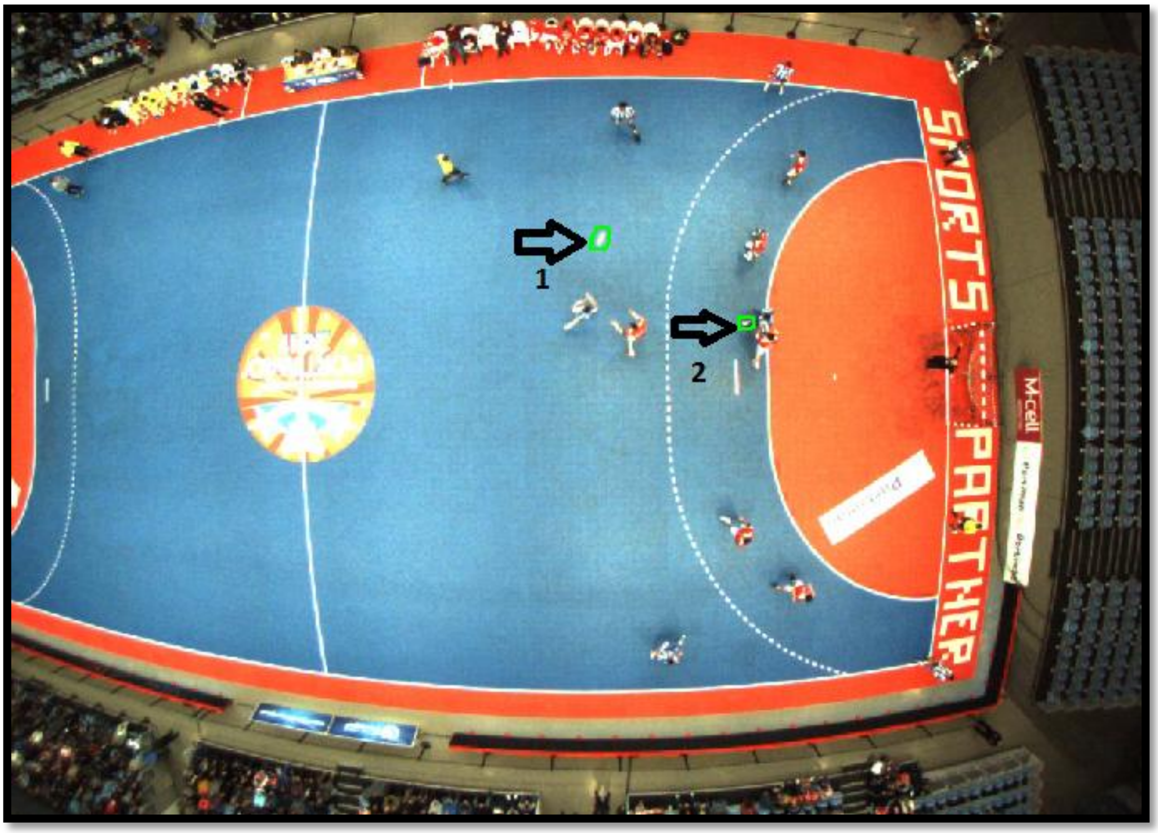


Figure 64 - Example of the final image with an False positive, generated by the players shoes.

Regarding to the figure 65 is possible to see in that the ball was detected (1) among one false positive, which correspond to an player shoe(2).

The videos from the FADEUP, were also tested with this method. There are some images from it:

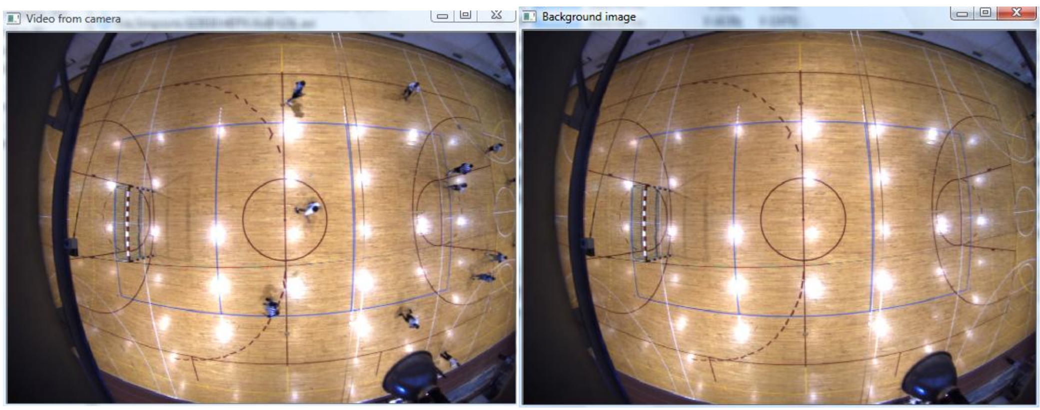


Figure 65 - Images from the FADEUP, ready to be subtracted

As is possible to see in the figure 66, the illumination of this pavilion was not the perfect one. It is possible to see some the reflex of the bulbs on the field.

Figure 66 - Subtraction of the image

Then the image is binarized by the a Threshold value:

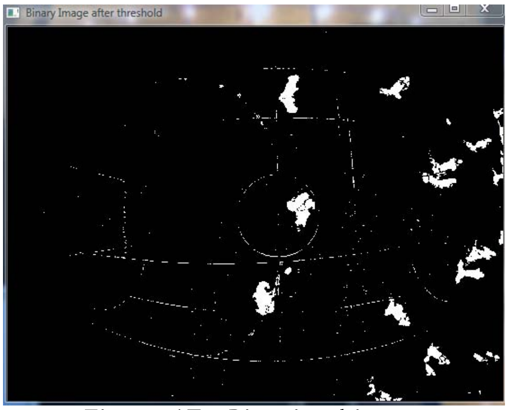


Figure 67 - Binarized image

The lines of the field that already exists are due to the variation in the illumination of the pavilion. I could vary the threshold value, in order to eliminate the darker field lines. Or I could apply an erosion operation to this image.

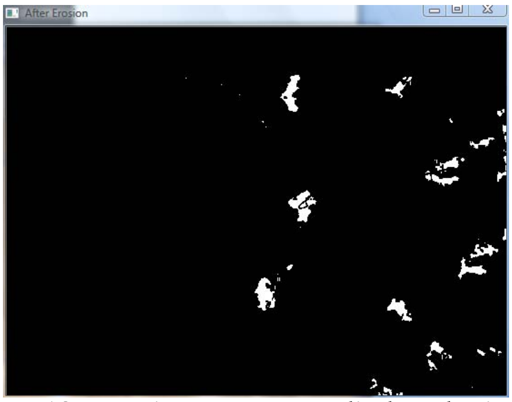


Figure 68 - Erosion process applied to the image

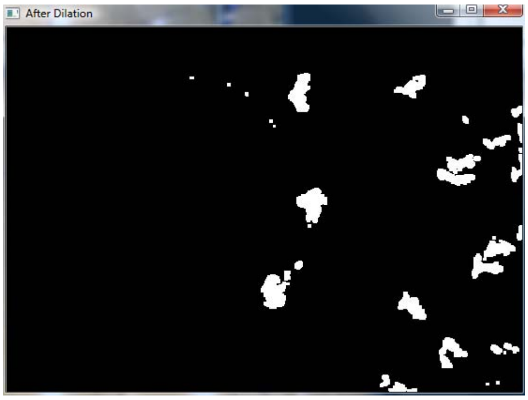


Figure 69 - Dilation process applied to the image

With this operation I could take the risk of eliminate some possible ball candidates.

It is possible to see in the image some noise, due to some objects in the field and also due to the shadow of the players.

This method present the following results for the two types of videos:

Table 10 - Static Background Subtraction with Analysis by Size and Color

VIDEO	BALL DETECTION RATE	FALSE ALARM RATE	CLEAN DETECTION RATE	PROCESSING TIME PER FRAME
SUPER CUP	78 %	30,3 %	67,8 %	41,6 ms
FADEUP	72,3 %	36,5%	62,3	43,4 ms

From the table is possible to see that for the SUPER CUP videos, the percentage of ball detection is 78 %, the percentage for the FADEUP video is a little worst. That is due to the bad illumination conditions and also to some object that appear during the real game and weren't present in the background image.

Is possible to see the video of this tracking in the following link:

http://www.youtube.com/watch?v=_JSb4H3EJj4

5.2 -Results from Adjacent Images Subtraction and Analysis by Size and Color

The idea of this method was to overcome the problems of the errors obtained by the first method. This method present better results than the ones obtained by the first method. So this will be the basis to more complex methods developed in this Thesis. However, some problems are created in the implementation of this method, like the disappearance of the objects that moves slowly between two frames. To overcome that, I propose a method to face that, as we will see ahead.

It is possible to see in the image from the left of the figure 70 that the ball as a duplicate representation. Let's see what happens when the AND operator is applied.

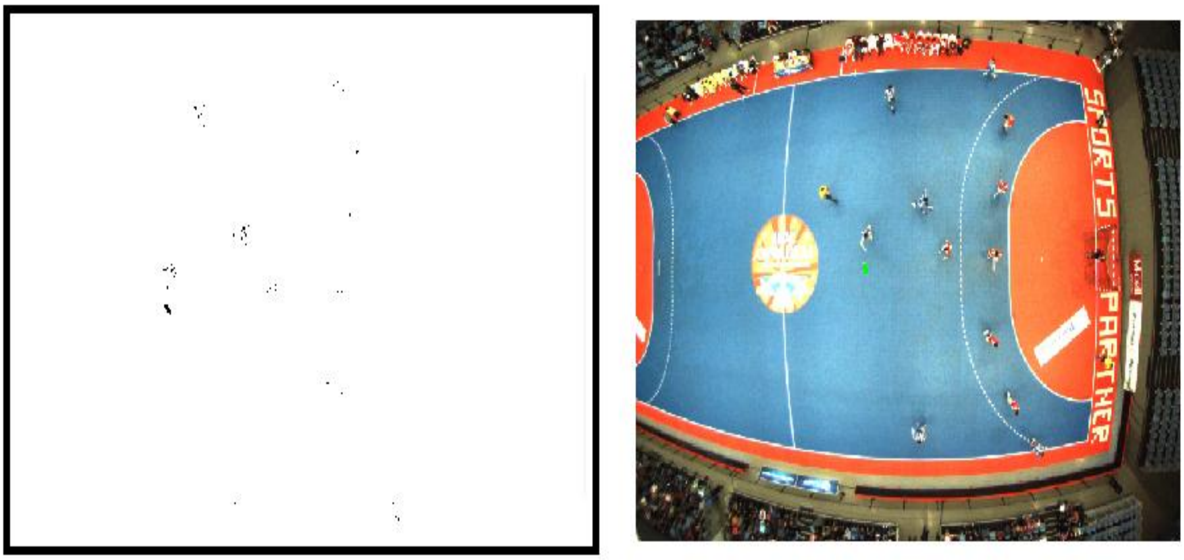


Figure 70 - The image in the left is result from the AND operation. The image in the right show the identification of the ball in the real image.

Table 11 - Adjacent Images Subtraction and Analysis by Size and Color

VIDEO	BALL DETECTION RATE	FALSE ALARM RATE	CLEAN DETECTION RATE	PROCESSING TIME PER FRAME
SUPER CUP	96,2 %	62,5 %	42,64 %	45,45 ms
FADEUP	92,4 %	64,7 %	38,43 %	47,6 ms

Is possible to see from the table 11 that the percentage of ball detection is very high 96,2 %, much more better than the one of the first method.

Another problem of this method is that it is no capable of detect a ball that is stopped in the field, or moving with a low speed.

The video from this method is available in the following link:

http://www.youtube.com/watch?v=yTSn8P4Z39M&feature=BFa&list=UL6esmffB4Ex0&index=3

5.3 - Results from the Elimination of False Positives

The objective of this method is to make an analysis on the previous image, an eliminate all the pixels on the left image that exist in the right image. The only remaining object should be the ball.

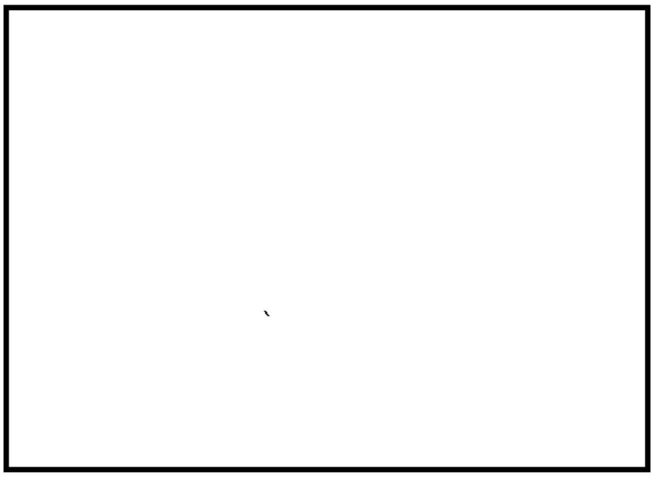


Figure 71 - From the image is possible to see that the remaining pixels correspond only to the ball.

The algorithm of this procedure could be seen in the next table.

Table 12 - Results from the Elimination of False Positives

VIDEO	BALL DETECTION RATE	FALSE ALARM RATE	CLEAN DETECTION RATE	PROCESSING TIME PER FRAME
SUPER CUP	95,8 %	1,3 %	99,3 %	66,6 ms
FADEUP	92,3 %	3,5 %	93,5 %	71,4 ms

The resulting image will only have the pixels corresponding to the ball.

This method was perfect to our propose to eliminate the false positives. The false alarm is very low. The only identified object, actually correspond to the ball. But the problem of the disappearance of the ball when it have a low speed is still there. So I decided to implement a new method, capable of detect the ball when it move slowly or even when it is stopped.

Two videos from this method are available on the following links:

http://www.youtube.com/watch?v=RrJigcyflyk

On the second video one I decided to print in the image the path of the ball:

http://www.youtube.com/watch?v=ZUPMJvdEeno

5.4 - Results from Region of Interest and Ball position Estimation

I will show here the debug images to help in the understanding of what I have done.

Figure 72 - Region of interest in the image

Inside the window is possible to see the contour of the ball.

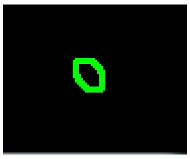


Figure 73 - Region of interest in the frame

Is possible to see the small highlighted square in the left image from the figure 73, which is the processing window in the whole frame. In figure 74, is possible to see how the system sees the ball.

Inside the window is possible to see the ball. This window already have the standard dimension, and it has been already updated to the put its center in the position of the ball in the last frame.

This graceful method works in peace until the ball disappears from the ROI window. In that moment in necessary to perform an Static Background Subtraction and search for players. The next two figures explain that.

Figure 74- Window approaching a player

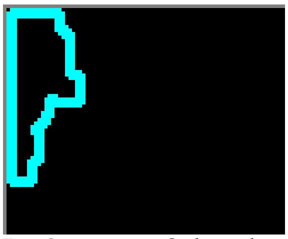


Figure 75 - Contour of the player

It is possible to see in figure 75 that the ball is in possession of the player. So the next step is to search is to track that player in the field.

We can see also the application of this method on the videos from the Super Cup.

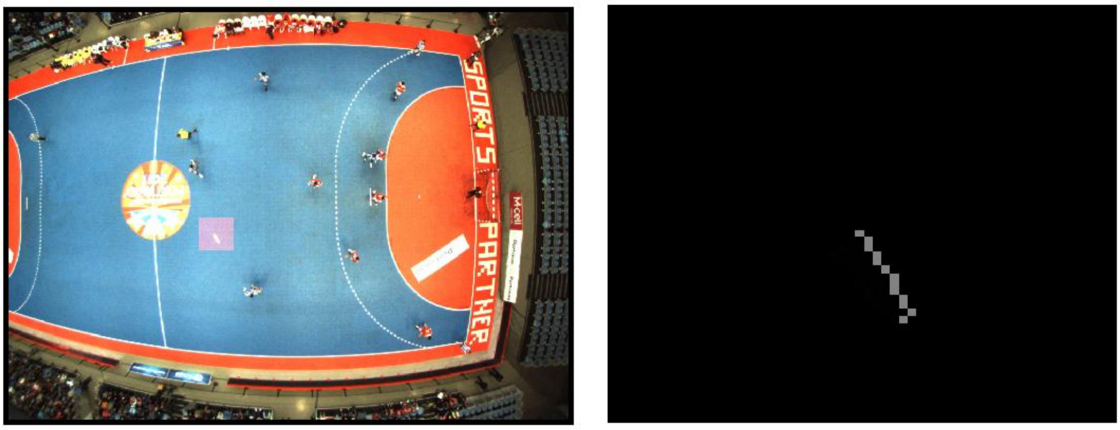


Figure 76 - Method applied to Super Cup videos

The negative thing about this method is that it consumes a great amount of memory, and the way I developed the application it crashes after some minutes.

Table 13 - Results from Region of Interest and Ball position Estimation

VIDEO	BALL DETECTION RATE	FALSE ALARM RATE	CLEAN DETECTION RATE	PROCESSING TIME PER FRAME
SUPER CUP	97,3 %	0,6 %	99,6 %	62,5 ms
FADEUP	95,2 %	1,4 %	94,5 %	64,4 ms

The video from this method can be seen in the following link.

http://www.youtube.com/watch?v=Qzq3qteFZIE

5.5 - Methods Evaluation

The performance of the developed methods will be measured by the percentage of the Ball detection based on my criteria, the percentage of false positives, and the processing time.

To achieve this values, I run and observe frame by frame. On important thing that must be known, is that I don't impose that the system detect the ball when it is in the possession of the player. The Ball detection, is the percentage of times that the algorithm detect correctly the ball. The False Positives, are the number of times that the system incorrectly detect an object as being the ball. The objective is that the percentage of Ball Detection be high, and the False Positives be minimize. The third criteria is the processing speed, that will be measured in frames per second.

The evaluation was performed in three stages: first the videos were gathered from the pavilion of FADEUP and from the Portimão pavilion during the Super Cup. The Super Cup videos, actually are a real game videos. Then the videos were processed using each one of the 4 methods I developed. Finally the results were analyzed.

The results could be seen on the followings charts. The Ball detection rate has the following average results:

The results have the following correspondence:

Method 1: Results from Static Background Subtraction and Analysis by Size and Color

Method 2: Results from Adjacent Images Subtraction and Analysis by Size and Color

Method 3: Eliminating false positives

Method 4: Region of Interest and Ball position Estimation

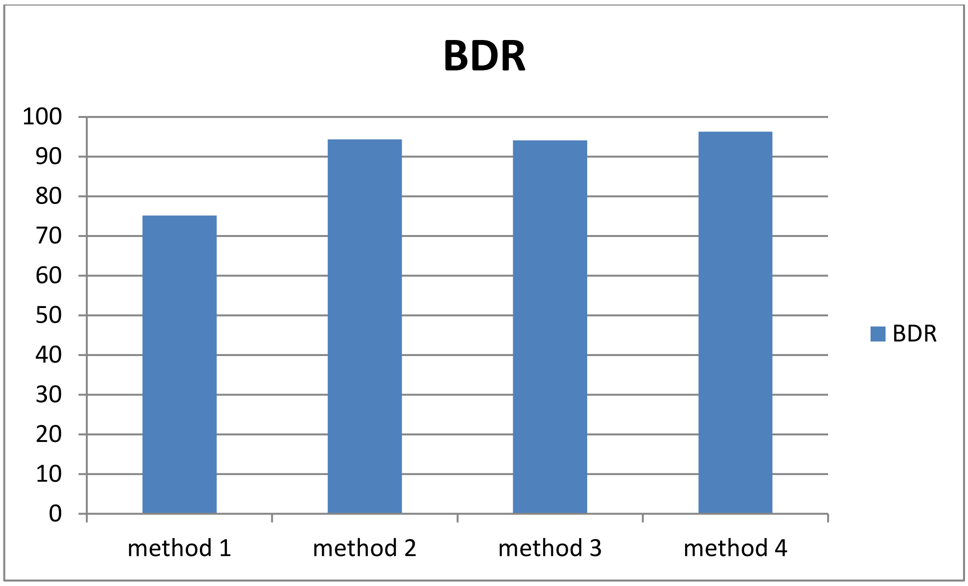


Figure 77 - Ball detection rate

From the figure 77 is possible to see, that the best Ball detection method corresponding to the “Region of Interest and Ball position Estimation”. The False Alarm Rate can be seen in the next figure:

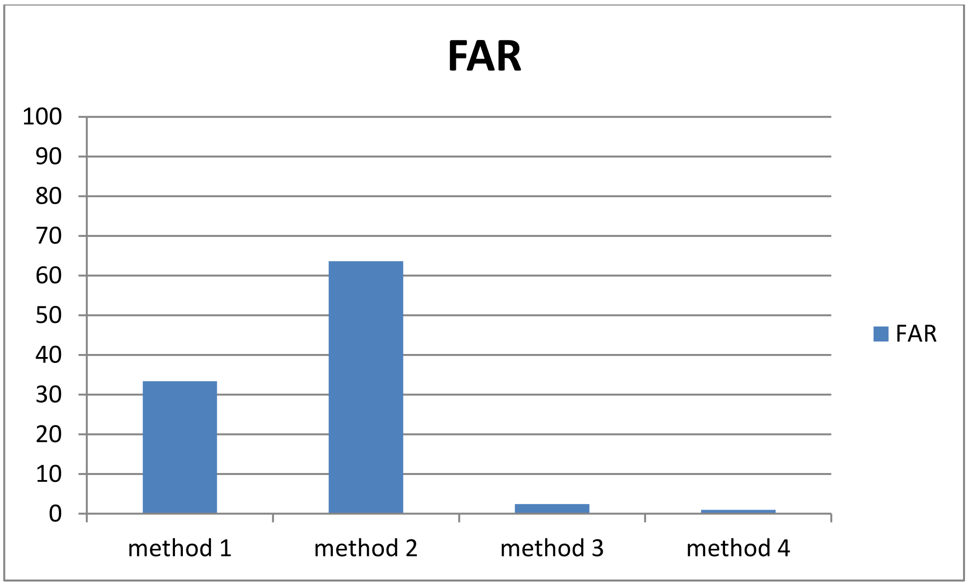


Figure 78 - False alarm rate

From the figure 78, is possible to see, that the worst method with a higher False Alarm Rate is the method “Adjacent Images Subtraction and Analysis by Size and Color”. The Clean Detection Rate could be analyzed by the graph of the following figure:

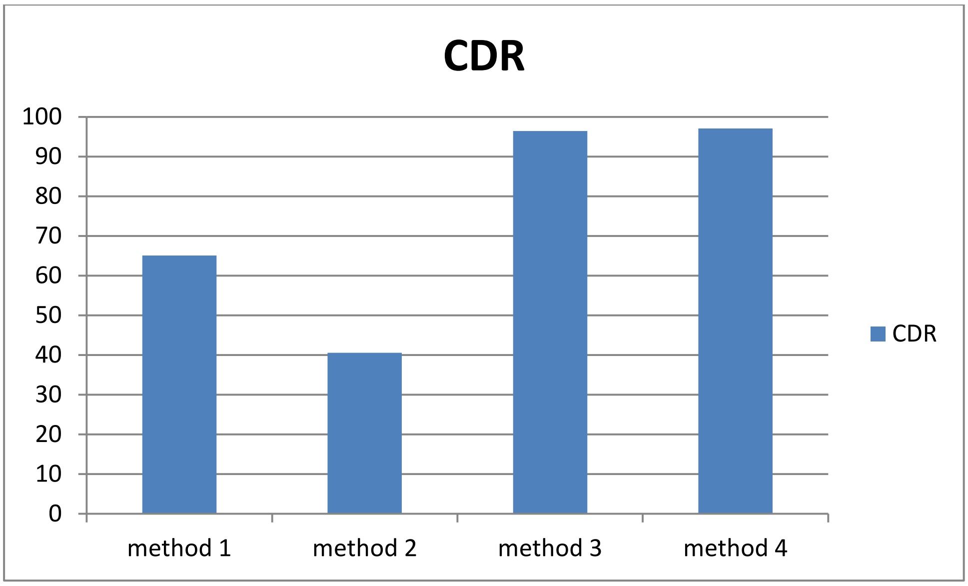


Figure 79 - Clean detection rate

From the figure 79 is possible to see that the best methods in terms of Clean Detection are the " Eliminating false positives" and "Region of Interest and Ball position Estimation". Finally, the Processing Time spent in each frame, can be seen in the following figure:

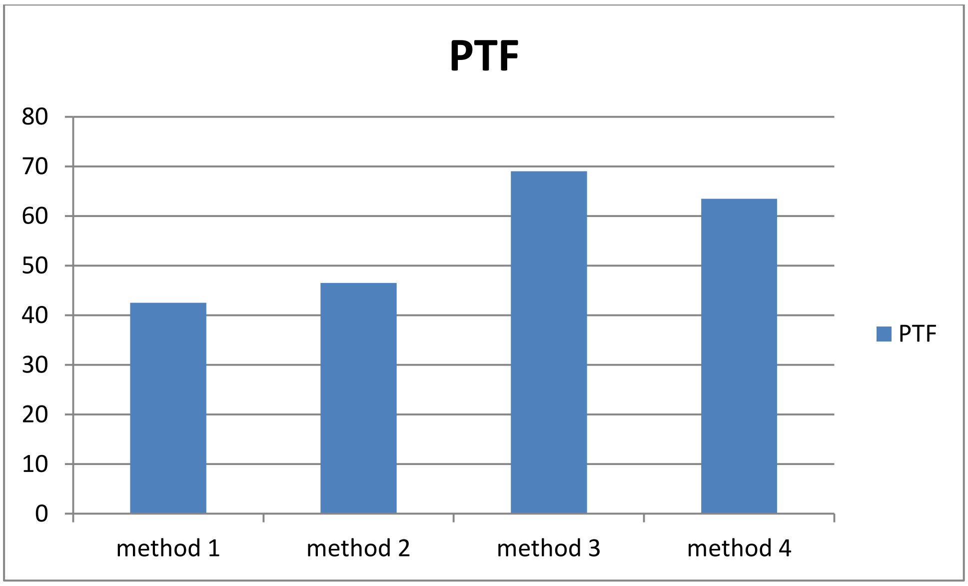


Figure 80 - Processing time per frame

From the last figure is possible to see that the method with the faster method is the method 1, while the slowest one is the method "Region of Interest and Ball position Estimation".

The overall results can be seen in the next table.

Table 14 - Overall results for the applied methods

METHOD	BALL DETECTION RATE	FALSE ALARM RATE	CLEAN DETECTION RATE	PROCESSING TIME PER FRAME
Results from Static Background Subtraction and Analysis by Size and Color	75,15 %	33,4 %	65,05 %	42,5 ms
Results from Adjacent Images Subtraction and Analysis by Size and Color	94,3 %	63,6 %	40,54 %	46,53 ms
Eliminating false positives	94, 05 %	2,4 %	96,4 %	69 ms
Region of Interest and Ball position Estimation	96,25 %	1 %	97,05 %	63,45 ms

From table 14 is possible to see that the results from the 4 methods are quite different. The method 1 is the one with the worst results in the detection rate. The better methods to a good detection are the methods “Eliminating false positives” and “Region of Interest and Ball position Estimation”. The second and third parameter to analyze is the False Alarm Rate and Clean Detection Rate which indicate the amount of bad detections, caused by similar objects to the ball. Those results, show us that the method “Results from Adjacent Images Subtraction and Analysis by Size and Color” is the one with worst results. Again, the methods “Eliminating false positives” and “Region of Interest and Ball position Estimation” have the best results.

The final parameter to analyze is the processing time, which tell us, that the fastest one is the method “Results from Static Background Subtraction and Analysis by Size and Color”.

5.6 - Conclusions

All the proposed methods have some advantages and disadvantages. Some of them are best to search for the tracking of players, and some to search for the ball. To search for the players the best technique is the Background Subtraction with a Static Image Subtraction. But to this project the most important thing is to identify the ball. So when it comes to choose the best method some considerations must be done. The first one, is that the purpose of this work was to developed a method to detect and track the ball in REAL-TIME. So the processing time of each frame should be an important thing to have in consideration. But, despite the processing time is a very important parameter, the percentage of ball detection isn’t less. So the user must reach to a compromise. Looking to or results, If the intention is to extract the ball coordinates in real time, the Results from Adjacent Images “Subtraction and Analysis by Size and Color” method should be used. But if the intention is to obtain the results a few seconds later, the best method is with no doubts the “Region of Interest and Ball position Estimation”.

Chapter 6

Conclusions and Future Work

In this chapter, the last one, some conclusions are made regarding to this project, and some considerations about future work will be done as well.

6.1 - Conclusions

This document presents an approach to digital image processing in sport environment. The main objectives were to study new methodologies for a better tracking of moving objects, like player and ball in the game. This particular document, focused on the detection of the ball. To this end, we collected images from cameras strategically placed at the top of the pavilions. The main objective was the use of image processing techniques, to determine the three coordinates of the ball. To this end, we developed four methods. The first was based on Static Background Subtraction, the second in Adjacent Subtraction image. These two methods are the most commonly used methods with regard to the identification of objects. However the results were not very satisfactory, mainly due to their high error rates caused by false positives, for example, the method based on Adjacent image processing has presented a value of 62 % of False alarm rate, which indicates that in each 10 frames the algorithm wrong detect in 6 frames similar look alike objects as being the ball. Those false positives were due essentially to members of the players body's. However the second method showed the best results in terms of detection of the ball, so I decided to develop another method based on it. So, to decrease the False Alarm Rate, the solution was to create another method where it made a combination of the output of the method "Adjacent Images Subtraction and Analysis by Size and Color" with the blob of the players. Here an algorithm would be applied to negate all pixels in common of the two frames. As can be seen on the results obtained in Chapter 5, the False Alarm Rate fell to 2,4 % , so that this method was a success.

The method "Eliminating false positives" shows good results without a doubt, however, there were still two parameters that I saw that could be improved, the Ball Detection Rate, and Processing Time Frame. So I decided to create another method. This new method is based on the definition of a window around the ball, where the processing will be performed only within that window. Detection are then performed either for ball or the players.

This method could improve the results of the third, and the Ball Detection Rate increased to 94.05 and 96.2% processing time of each frame dropped from 69 ms to 63.45 ms.

A particular objective was not achieved, which was to detect the height of the ball relative to the field. With studies accomplishments, was in fact possible to determine the height of the ball with only one camera, however the area of the ball would be extremely important. Although you can collect all the methods in the area of the ball, it has never had consistent results, it is possible to see the output file of the methods in Appendix A. Poor detection area of the ball due to several factors, including the great distance of the ball the camera, the high speed of the ball, and also the lighting conditions.

In conclusion, I can say that this project was very rewarding on a personal level because I had my first contact with image processing techniques.

6.2 - Future Work

To accomplish this job in a proper way many things could be done. A multi camera approach could be made, so the ball Height will be obtained by the combination of images from different perspectives. It would be good to use also the information of this analysis to help the referee team. That could be done for example by identifying the lines of the field and to detect when the ball pass through it.

It would be nice to use two computers, working in parallel, thereby reducing the processing time.

To obtain even better results, it would be nice if the color of the ball was totally different from the color field and the players' equipment.

References

[AlvaGome_09]Luis Alvarez, Luis Gomez and Rafael Sendra, An Algebraic Approach to Lens Distortion by Line Rectification,

[AlpYil_04] Apler Yilmaz. Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras, 2004

[AncBra_00] N. Ancona, A. Branca. Example based object detection in cluttered background with Support Vector Machines. Instituto Elaborazione Segnali ed Immagini.
Bari, Italy 2000.

[Brow_66] Brown DC (1966). "Decentering distortion of lenses.". Photogrammetric Engineering. 7: 444–462.

[Canny_86] Computational approach to edge detection, 1986

[Cata_09] Catarina Maria Brito de Noronha Santiago. Image Analysis of Sport Events, 2009

[CataLuisMar_10] Catarina B. Santiago, Armando Sousa , Luis Paulo Reis. Survey on Team Tracking Techniques Applied to Sports, 2010

[Gue_02] A. Gueziec, Tracking pitches for broadcast television, Computer 35 (2002)

[HerSun_89] G. Herzog, C.-K. Sung, E. Andre, W. Enkelmann, H.-H. Nagel, T. Rist, W. Wahlster, and G. Zimmermann. Incremental natural language description of dynamic imagery. In Wissensbasierte Systeme, pages 153–162, 1989.

[Lum_04] Ville Lumikero. Football Tracking in Wide-Screen Video Sequences. Master Thesis in Computer Science, School of Electrical Engineering Royal Institute of Technology. Stockholm, 2004.

[ManJep_02] R. Mann and A. Jepson. Detection and classification of motion boundaries. In Proceedings of AAAI-2002, Edmonton, AB, July 2002.

[NathFun_03] Nathan Funk. A Study of the Kalman Filter applied to Visual Tracking, University of Alberta Project for CMPUT 652 2003

[OraGua_03] T. D'Orazio, C. Guaragnella, M. Leo, A. Distante. A new algorithm for ball recognition using circle Hough transform and neural classifier. Pattern Recognition 37, pp 393-408, 2003.

[PinOpa_00] G. Pingalli, A. Opalach, Y. Jean, Ball Tracking and virtual replay for innovative tennis broadcast, in Proc. 15th Int. Conf. Pattern Recogn. , vol. 4, 2000

[VioJon_01] Viola, P., and Jones (2001). Robust Real-Time Object Detection. In proceedings of the second International Workshop on Computational Theories of Vision.

[Tsai_87] Roger Y. Tsai.A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV

[Xie_03] Xie, British Machine Vision Conference, 2003

[WuHuan_99] Y. Wu and T. S. Huang. Vision-based gesture recognition: A review. In GW ’99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, pages 103–115, London, UK, 1999. Springer-Verlag.

Appendix A

Results output

In this appendix is possible to see how the results are saved at the file .txt.
The values in the file are the number of the frame in the first column, in the second and third columns are the coordinate of the ball bp_x and bp_y, which are the x coordinate and y coordinate respectively. The final parameter in the file is the area of the ball.
As is possible to see, the area of the ball extracted from my method is not very regular. It tells me, that its values should not be used in order to determine the z coordinate of the ball as explained in the chapter 4.

frame	bp_x	bp_y	area
27	311	289	40
31	333	342	31
32	340	354	38
33	344	369	33,5
34	353	382	39
35	360	394	44
36	367	408	35.5
37	376	420	39,5
38	384	434	42,5
39	391	447	30,5
40	399	459	45,5
41	408	471	35
42	414	484	33
43	424	495	43

Figure 81 - Output file