Tracking Football Players

An Investigation into Football Player Tracking in Amateur Football Videos

Paul Lether
ID Number: 10174964
Supervised by Dr Aphrodite Galata
May, 2020

A report submitted as part of the Third Year Project for the degree of BSc (Hons) Computer Science

Acknowledgements

I would like to thank my supervisor, Aphrodite Galata, for her full support throughout the project as without it the project would not have been possible. I would also like to thank my parents and sister for always being a positive voice to hear when needed.

Abstract

This project outlines the build process and evaluation of a piece of non-real time software for a football tracking system using a camera array. The software includes a number of computer vision techniques including player detection by background subtraction and player identification by SIFT with KNN classification. It is written in C++ with the OpenCV library. The dataset used is from a Norwegian football match. The match was recorded using three static cameras with their output videos being stitched together to create a final video output. During this match each player had their locations tracked using highly accurate ZXY radio tracking devices which are used to evaluate the performance of the system.

Contents

List of Tables

List of Figures

1. Introduction

Sport analysis is slowly becoming one of the most important tools in improving the standard of sports. One particular area of sport analysis that is becoming extremely popular is player tracking. This is where players are tracked during the game with their location recorded throughout. After the data from the players is collected, it can be expanded on to find out useful statistics such as distance travelled during the match and sprint speed. These statistics have been used over a variety of sports, yet the majority of the demand has come from football teams.

1.1 Aim

These football teams have been the driving force behind the advancements in this area, with improvements constantly occurring. The current methods being deployed to do this task involve either radio transmitters or a camera array. These methods are extremely effective. However, they are often unattainable for the majority of teams due to the high cost that private companies providing the service are currently charging.

The aim of this project is to investigate whether it is possible to provide an alternative system that could be used by a team to gain the same football analysis at a fraction of the cost that is currently being charged. This third year project has been attempted in previous years by [1] and [2]. Both students achieved a considerable amount during their projects, with both programs developed managing an impressive accuracy for player identification. This project wishes to further improve upon their work by following a similar process to track the players but with an increased number of players tracked.

Furthermore, another aim of this project is to find the distance the tracked players trav-

elled during the match which, due to time restrictions, previous students were unable to attempt. The program does not need to run in real time. Nevertheless, the program should be able to find the statistics of a football match within a reasonable time.

1.2 Challenges

This project comes with many challenges that need to be minimized or fixed. These include:

- During the football matches players will often overlap each other, obscuring the camera view for certain players. This can cause issues with player detection and identification.
- Movement will occur from objects or people that are not players of the football match. These movements will need to be ignored.
- Player identification when players move from the far side of the pitch to the near side can cause issues with identification methods.
- Camera distortion can cause issues with player location.
- Converting the size of a pixel to a known measurement, with football pitches having different height and width sizes.
- Weather conditions.
- Lighting conditions.

1.3 Approach

There are a number of ways to approach this project, with each approach offering their own advantages and disadvantages. In the following section I will run through the most commonly used approaches to track players during a football match. I will go into detail to explain how certain methods within these approaches work.

After the system is created, an evaluation of the results will be undertaken. This will investigate the efficiency of the player tracking process. The distance travelled by tracked players will also be examined to test the accuracy of the measurements taken. From this, a conclusion can be found stating if this approach is successful or not.

1.4 Impact of COVID-19

During the later stages of the project the COVID-19 pandemic was taking place. Due to the risk of infection, the University of Manchester moved all teaching to online and closed all study spaces. This meant a large proportion of this project had to be done at home instead of a study space such as a library. This greatly impacted on my productivity due to my house not having any study space other than my bedroom which was extremely hard not to procrastinate in. Furthermore, my flatmates stayed in the house throughout the lockdown but unlike me they had no work to complete. This became a large distraction which was hard to escape from. Luckily, after a while I adapted to the new study environment and a constant workflow was achieved. No other change in circumstance occurred for the remainder of the time while I was completing the project.

2. Background

A number of approaches have been created to track players during a sports match. There are two main systems that are widely used to track players during sports matches: a wearable device tracking system and a multiple camera array set up.

2.1 Wearable Devices

There are two approaches used with wearable devices: the system radio and a GPS tracking system. ZXY uses small radio trackers which transmit to multiple local radio aerials, while GPS trackers use a satellite. GPS has the obvious advantage of not needing any setting up of aerials. However, [6] provides evidence that ZXY is more accurate than GPS tracking systems. The wearable devices worn by the players for both the ZXY and the GPS systems are small devices that can be incorporated into sport wear such as a belt or wristband. For each player to be tracked, they are required to wear one of these receivers. The cost of each wearable device quickly adds up when considering the number of players needing the devices and the maintenance of each device. ZXY comes with an additional cost of not only requiring the purchase of the wearable devices but the additional cost of buying and installing the radio aerials.

2.2 Camera Array

A camera array is where a number of cameras are placed so that the output of the image covers all of the pitch. Computer vision techniques are then used to work out scene set up, player detection, and identification.

2.2.1 Previous work

Previous work done in this area include:

- A previous student [2] used a three stage approach of player detection, player identification and player tracking. For player detection, a method of a Gaussian background subtraction model was used to detect players in the scene. When combined with temporal coherence this was found to work extremely effectively for the frames where player movement occurred. Unfortunately, this method of detection did not perform to the same high level of detection when frames without player movement were used. For player identification, a number of methods were experimented with the highest achieved being SIFT with a trained K nearest neighbour classifier. The tracking in this project is simply ensuring multiple classifications does not occur and the distance between the last known location of the player and the potential new location is under a threshold, to exploit temporal coherence.

- Another student from a previous year [1] included the same three stages used by [2] but split identification into two separate sections. The newly created sections are an initial identification stage made early on and one final identification stage found late on in the programs pipeline. Methods used within the stages were also changed. Detection was changed to HOG identification which had been trained using a data set to detect humans. This fixed the issue that was occurring in [2] where players would not be identified if they were not moving. Player identification used SIFT features with a K-NN with tracklets being used with Kalman predictions to try to determine the players future location which is used for the measurements used for temporal coherence.

- Paper [7] made use of a background subtraction method but integrated it with sensor data collected from wearable devices worn by the players. The recorded player's position can be plotted on the image and the area surrounding this reading is noted. The areas that are noted are the only places background subtraction is applied. This decreases the potential area players could be within from the whole

frame to a small section of the frame. By removing these sections of the pitch it removes the chance for potential incorrect identification in these incorrect areas, which has an effect of increasing precision.

2.2.2 Scene Set Up

Scene set up is where image manipulation is conducted to try to fix unwanted changes that have occurred during the capturing process.

Removing Distortion

One main issue with capturing video data is that camera lenses can cause a number of effects to the image produced of the scene. A common effect caused is radial distortion. This effect causes straight lines to appear curved, where the effect intensifies the further away the line is from the center of the image with issues being caused by the curvature of the lens. There are a number of approaches to solve this issue however, according to [9], the most commonly used model is the [10]. Radical distortion is defined by Zhang [10] as:

$$x_u = x_d(1 + \lambda_1 r_d^2 + \lambda_1 r_d^4 + ...),$$

$$y_u = y_d(1 + \lambda_1 r_d^2 + \lambda_1 r_d^4 + ...),$$

Where:

- x_d, y_d = Disorted point
- x_u, y_u = Undisorted point
- λ = Radical distortion coefficients
- R_d = Distance between distorted point and optical centers using euclidean distance

To solve the equation the R_d needs to be found. This can be done by finding the calibration matrix. The calibration matrix is :

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

A well known computer vision library, Opencv [5], documentation states the best way of doing this is to place a known object with a pattern, such as a chessboard, in front of the camera at different positions without changing the Z axis. A number of photos are taken. After this we can work out the distance between the points of the image on the chessboard and these points in 3D space. This is because the chessboard's dimensions are known and the corners of the image points can be calculated from the images themselves. As Z is kept the same we can find the transformation between the object points and the image points. From this we can calculate R_d and then rearrange the equation to find out the distortion coefficients which then can be used to undistort the image.

Figure 2.1: Image showing Opencv displaying what the straight lines of the board should be (from [5]).

2.2.3 Player Detection

Player detection is where sections of a frame are detected as potentially containing a player.

Background Subtraction

Background subtraction is a common method to perform player detection. It allows for a fast and efficient way of finding movements in a scene. There are two ways of performing background subtraction: simple background subtraction and Gaussian mixture model.
Simple background subtraction is where the difference between a frame without any foreground and a new frame of the scene is found. As a static camera is used the difference between the frames will be a change in the scene such as a person entering the frame. This form of background subtraction does have its drawbacks as an empty scene frame is required which is not always available. Furthermore, simple background subtraction does not have any methods to try to distinguish between objects in the frame and lighting such as shadows. This can cause issues with player detection.
[3] suggested a new form of background subtraction to resolve the issue of needing a scene without any foreground. This background subtraction uses a Gaussian mixture model. If a pixel remains the same colour for a certain number of frames or if a pixel has a colour that is used in a high proportion of a frame then the pixel will be classified as background. This paper also presented a method to try to remove shadows. This is done by comparing a potential foreground pixel's chromatic and brightness values with the known background value. If these values are within a certain threshold then the pixel is considered to be a shadow so will be identified as background.

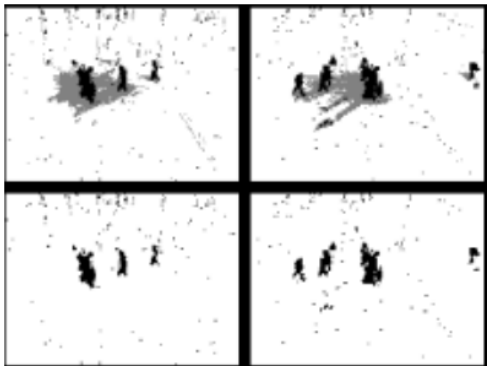

Figure 2.2: Series of frames of background subtraction with and without the shadow remover (from [3]).

2.2.4 Player Identification

There are many computer vision methods to identify objects and humans by using feature detection. One of the most widely used is Scale Invariant Feature Transform (SIFT).

SIFT

Scale invariant feature transform takes an image and returns the key points in the image. This is done in four steps outlined in [4]:

1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation

that allows for significant levels of local shape distortion and change in illumination.

After the key points have been found a KNN classifier can be used to compare the key points found to other images' SIFT key points. If a low KNN value is achieved then it is likely that both images contain the same object or is of the same scene.

Figure 2.3: Objects being identified in the scene with the SIFT algorithm used to locate the key points of each image (from [4]).

3. Development

In this section I will explain the steps I took while creating the system.

3.1 Planning

The system I am proposing is a camera array with a program with computer vision techniques. This was chosen over wearable devices due to the cost associated with buying and maintaining them.
The main feature of the system is the player tracking program, so this will be the main focus of the project. As the program will need some data for ground truth, a relevant dataset has been chosen.

3.2 The Dataset

The dataset I will be using is the 'Soccer Video and Player Position Dataset' [6]. This dataset contains a friendly football match between Tromso IL and Tottenham Hotspur. This match not only contains still camera video of the football match but also ZXY sensor data for some players on the pitch. The data from the radio sensors will be used as ground truth against the data found with the new computer vision system.
The video is constructed from three cameras covering different parts of the pitch which is then stitched together. This stitching process has already been done for us however; if someone were to follow up on this software, it would be advantageous for the software to do this itself as we have the separate video feeds for each camera available.

Figure 3.1: Figure displaying images captured from each camera during the game (from [6]).

[6] explains how the data sensors are recorded. Players wear a ZXY belt that records a 20Hz with the accuracy of 1m. The positions are recorded as a form in distance away from the top left of the pitch in the x and y direction for each timestamp.

3.3 Implementation of the Tracking Software

The software implementation will be split into three parts: 1. Scene set up 2. Player detection, 3. Player identification. The overall work flow of the project is:

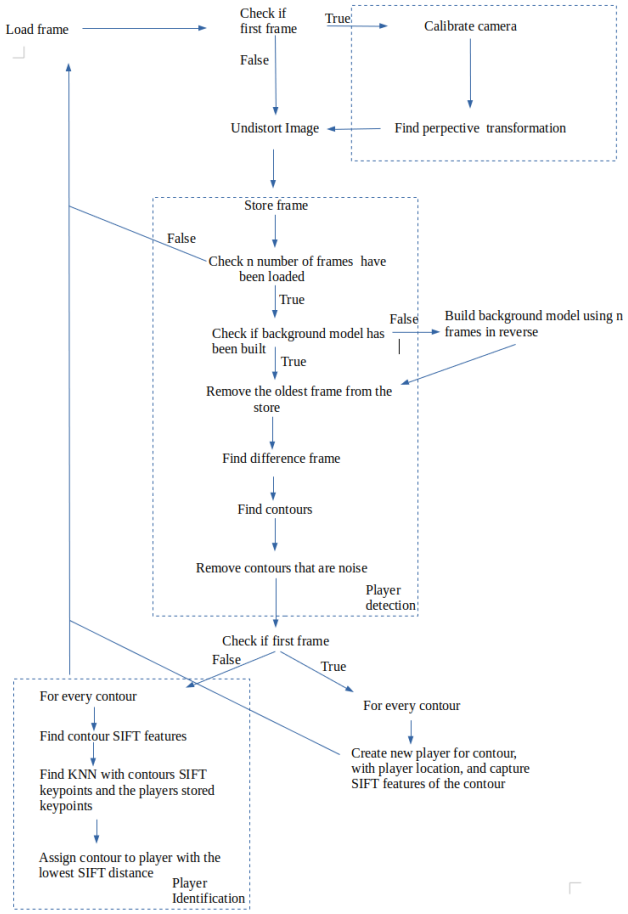

Figure 3.2: Figure showing the general structure of the program.

3.3.1 Scene Set Up

Before any player detection can occur the frame must be manipulated to ensure that any unwanted camera effects are removed. The pitch area of the frame has to be found so it can later be used for player detection.

Calibrate Camera

An unwanted effect that takes place is radial distortion making straight lines appear curved. Normally to solve this problem a camera calibration object with known dimen-

sions would be used at different viewpoints then software would be used to solve equations which would undistort the image as stated in section subsection 2.2.2. However, during this project there is no access to the cameras used to create the dataset nor is the cameras calibration matrix provided. This makes this method non-applicable with this dataset.
An alternative approach was taken where the undistorted frame is estimated in a different manner. This is done by finding the transformation between the coordinates of points on the curved lines of the disorted image and the coordinates of each of these points but with the curved line straightened. To find what the correct location of the points on the curved line should be, the straight lines that would occur without radial distortion are first found. The two straight lines are found by inputting the two corner points which the curved line goes between into the equation of a straight line. Then the points calculated on each curved line have their x value put into the corresponding straight line equation to find what the y value is for each point. The transformation will be applied to the frame to try to undistort the image. This is only an estimation so will have an affect on the overall accuracy of player travel; however, this issue is exclusive to datasets without a calibrated camera. As this issue will only happen on a select number of datasets the points on the curved lines are hard coded in.

Figure 3.3: Screenshots showing the steps taken to try to fix radial distortion.

Finding the Pitch Area

After finding the undistorted image, the area of the football pitch is needed to be found. This is required so only movement on the pitch is detected, and not other movement such as the crowd. To get the area of the pitch the first frame is applied with a green mask.

Erosion and dilation is then applied to get rid of small individual pixels of green and then the contours of the pitch is found. The contours with the largest area is kept as this will likely be the pitch itself.

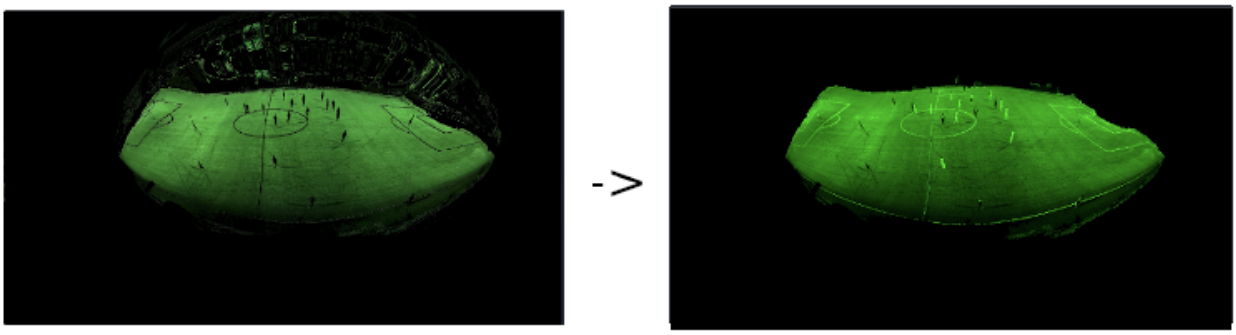

Figure 3.4: Screenshots displaying the frame with only green pixels and the largest green contour.

At this point the area found covers the pitch in full but some small sections outside of the pitch are detected. To make the area a better representation of the pitch, the lines around the pitch are found. This is done through Canny edge detection and then with Hough line detection applied. Then, by only drawing the lines detected on a blank image, the largest contour is found again as the lines around the pitch all join together. This will be the largest contour. Erosion and dilation is then applied to remove small areas detected outside the pitch.

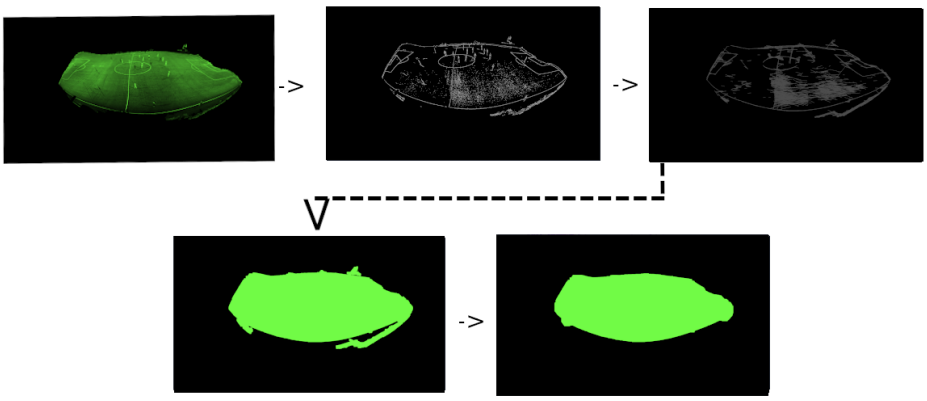

Figure 3.5: Screenshots showing the process of getting a better pitch area representation.

Perspective Transformation

The last step in setting up the scene is to find the perspective transformation of the pitch. This is where a translation is found that will solve the issue of the far left and right side of the pitch appearing to curve inwards caused by perspective. This is solved by finding the perspective transformation between the location of the corners of the curved pitch and the location of them if there was no perspective.

To find the corners of the curved pitch Harris corner detector is used. This finds all the corners of the pitch but also finds some non-pitch corner points. To determine which found point is the corners of the pitch, the pitch is split into a grid. For every grid square the average Harris corner point is found for that area. The grid is then searched from the top left across every column then row. The first grid square with a Harris corner within will be the top left corner of the pitch. The top right corner of the pitch will be the furthest away grid square containing Harris corners on this grid row. The bottom left corner of the pitch will be grid square containing Harris corners which is the closest to the left hand side of the image while the bottom right will be the grid square containing Harris corners furthest to the right. The effect has occurred due to perspective making the goal lines bend inwards the further they're away from the camera.

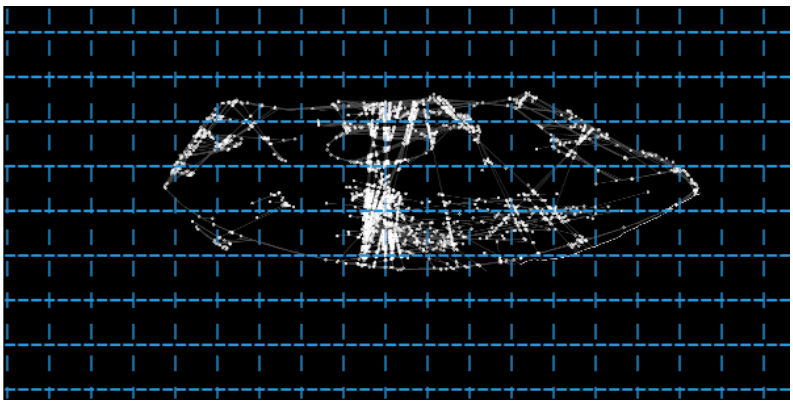

Figure 3.6: Screenshots showing Harris corner with a graphical representation of the grid.

Once the corners of the curved pitch are found, we can find what the top two corners should be without perspective. This is done by setting the x values of the top left and right corners to the x value of the bottom left and right corners. Once this is found we can find the perspective transformation between the perspective pitch and the non-perspective pitch corners. This gives the ability to obtain an un-perspective pitch. However, this will be applied after player identification, due to player identification working more effectively without this transformation.

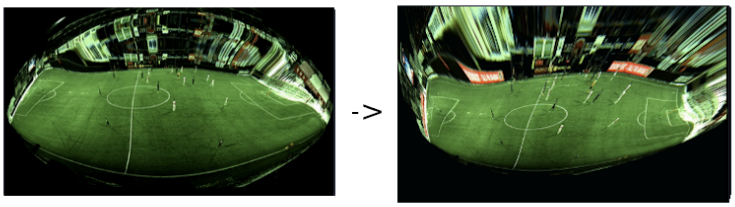

Figure 3.7: Screenshots showing the pitch with and without perspective.

Pixel to Known Measurements

To allow the distance travelled to be later calculated, the pixel to a known measurements have to be found. This can not be found by the pitch width or height as a football pitch has to be within a set range and not an exact value. One part of the football pitch that is a consistent size throughout every football pitch is the distance between the side line closest to the goal keeper and the six yard box. The number of pixels is found between

these lines and then six yards is divided by this number to find out the distance of each pixel. Due to time restrictions of the project this was only managed to be hard coded in.

3.3.2 Player Detection

Once the scene has been set up, player detection can now occur.

Background Subtraction

The first step in player detection is background subtraction. Initially simple background subtraction was used in the program. This method did not handle shadows nor marks that appeared on the pitch well as they were falsely detected as people. Due to this a Gaussian mixture background subtraction model with a shadow remover was used. This method dealt with these issues better as the model would be updated throughout the video, allowing it to adapt to the marks on the pitch while some of the shadows were removed prior to player detection. This method had the added advantage of not needing an image without any players present having to exist.

Nevertheless, a method was needed to allow the model to be built prior to player detection on the first frame of the video. This method involved storing the first few frames before any played detection processing occurred. After a number of frames were stored these frames were loaded into the background model in reverse. When the background model was built the difference between the frame, starting with the first frame stored, and the background model would be found and further processed. On every frame the background model would be updated using the current frame. The difference between the background model and the current frame can be seen as player movement.

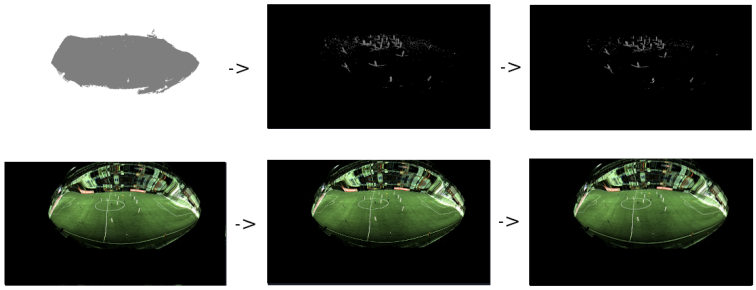

Figure 3.8: Figure showing the background model being built by frames in reverse.

Checking For People

After the difference frame is calculated, the different sections of movement need to be grouped together. To do this the contours of the image needs to be found. Prior to this happening, a bit-AND function needs to be applied between the difference frame and the current frame with the resulting image being in the RGB colour space. This is done because the difference image produced only detects differences between pixels of the current frame and the built background model. This means if contours were found on the difference frame it would be grouping difference between pixels values instead of similar pixel values. As any pixel on the pitch can have any pixel values this would mean a poor grouping achieved by the contours, as the pixels belonging to the same player could have a large variation for pixel differences. By merging the image, the contours found would be on the similarity of the pixel values which increase the chance of correctly identifying the correct contours. After the contours are found we need to check they are from human movement and not noise. To do this we check each contour has a height larger than its width as this is a characteristic of any human.

Creating and Updating Stats

After finding the player contours, one of two methods are called depending if this is the first frame or not. On the condition that it is the first frame, we assume that all the player contours found are players themselves. Each contour gets assigned an ID, their current

location is recorded (pixel position * perspective transformation), and SIFT keypoints of the image are detected in the contour area. No new players will be assigned after this first frame. Alternatively, if it is not the first frame, player identification occurs.

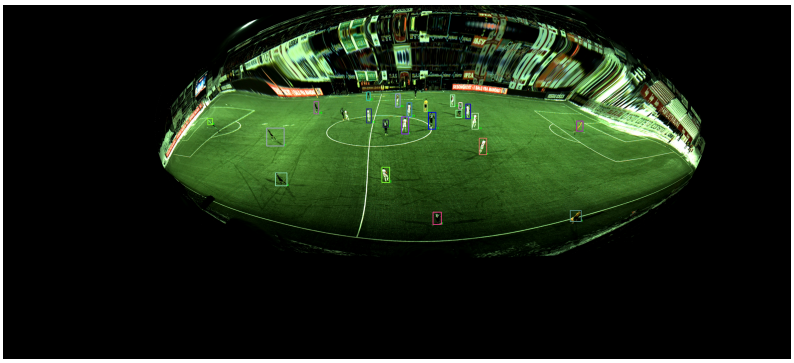

Figure 3.9: Screenshot showing classification of players in first frame.

3.3.3 Player Identification

At this stage there are a number of players detected with SIFT key points found from a number of images, their last recorded location on the pitch and some potential player movement contours. These contours need to be matched to the players. To do this we need to go through every contour, where for each contour we do the following steps.

Temporal Coherence

Temporal Coherence is the idea that if a person is at location at a certain time, they will be near this location if only a small amount of time has past. Exploiting temporal coherence is the first step in player identification. We can apply temporal coherence as the football match was recorded at a constant 25 FPS [6]. This means between frames there is only 0.04 seconds. Therefore, a player can only move a small distance from their last frame. As a result of this, players with their last recorded location over a certain distance away from the contour can be disregarded as it is impossible for the player to cover this distance within the time between frames. Additionally we also disregard players that have already been assigned a contour on the current frame.

SIFT Features

After discarding certain players for the contour, the player the contour most likely belongs to needs to be found. This is done by finding the SIFT key points of the contour, and using KNN to find the smallest distance between the contour and the player stored SIFT key points found from a number of images. The most similar SIFT key points will have the smallest KNN distance between them. Once the player with the lowest KNN distance is found the contour is assigned to that player. The player's data will then be updated. This involves updating the player's current location and distance travelled by working out the euclidean distance between the player's old and their new coordinates. Each point is multiplied by the perspective transformation and the pixel to known measurement found earlier on in section section 3.3.1. The contour's SIFT keypoints are stored for that player where each player can store a number of SIFT keypoints. When the maximum number of images that SIFT key points has been reached a random image SIFT key points is replaced.

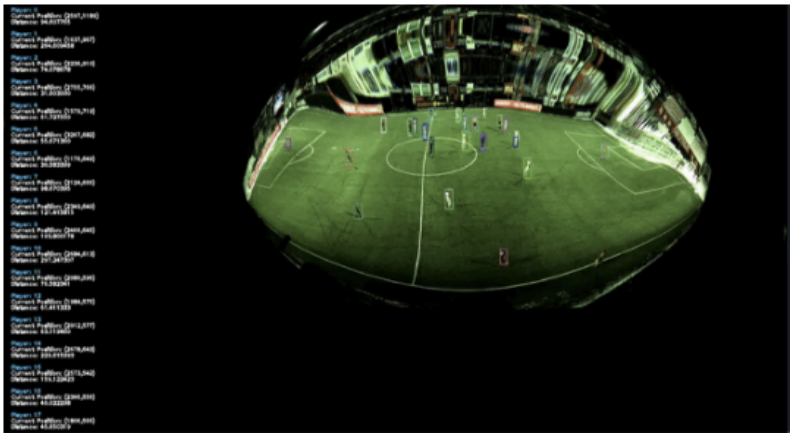

Figure 3.10: Screenshot displaying the players being tracked with the distance travelled by each person displayed on the left.

After all the contours have been run the next frame is loaded and the process is started again, finding the difference between the new frame and the background model.

4. Evaluation

In this section I will evaluate the effectiveness of the methods used during this project. This will be carried out by conducting several tests on the program created. Additionally, I shall investigate how different parameters for methods used within the program change the outcome produced.

Two equations that will be used to evaluate the program are:

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Where:

- TP = True Positive = Correctly identified Player
- FP = False Positive = Incorrectly identified Player
- TN = True Negative = Player not present in the frame
- FN = False Negative = No player identified for the player ID during the frame

4.1 Altering the Number of Starting Frames

When initially running the program, a number of frames will be run prior to player detection so the background model can be built. The number of frames selected changes how many players are detected. After running the program numerous times with the number of build frames being changed the results were:

Number of frames	Precision	Recall
1	0.02	0.045
2	0.8	0.72
3	0.76	0.72
4	0.81	0.77
5	0.86	0.86
6	0.86	0.86
7	0.86	0.81
8	0.90	0.81
9	0.90	0.81
10	0.90	0.81

Table 4.1: Table showing the effect of altering the starting number of frames on the background subtraction model.

If too few frames are selected then the background model will not have enough data to fully build the model. This means static parts of the scene are picked up as foreground due to being detected as movement. This is why building the model with only one frame has such a low precision and recall value. When increasing the number of frames loaded into the model, the precision slowly increases as the static parts of the pitch keep the same pixel values meaning incorrect player detection is decreased. Precision does not achieve 100 percent as the linesmen and the referee are moving during the frames so are incorrectly identified as players.

Recall becomes constant after seven frames. This is because the players who are not detected remain still during these first few frames that are being fed into the background model. The frames are loaded into the model in reverse, so increasing the number of frames used to build the model will not fix this problem. This is because the background model will still use the frames with the static players to update the background model at the end of the build process. Therefore, the static players will still be detected as static and thus background.

4.2 Using the Ground Truth

The original aim of the project was to use the ZXY data as a ground truth for both player identification and distance travelled. This would have be done by checking the classified person is within a certain distance of the recorded location. If they are, the contour image is saved within the correct images for that player. If not, it is saved in the incorrect player identification folder for the player. These images would have been manually checked later for mistakes.

To create a program to do this the following would have to be extracted from the ZXY table provided in [6]:

- timestamp (string) – Local Central European Time (CET) time encoded as ISO-8601 format string.
- tag id (int) – The sensor identifier.
- x pos (float) – Relative position in meters of the player in the field's x-direction.
- y pos (float) – Relative positions in meters of the player in the field's y-direction.

After extracting this data the sensors x and y position need to be plotted on the video pitch. To do this the x and y positions of the sensor data is first converted from meters to yards and then to a number of pixels by multiplying the pixel to yard measurement found in section 3.3.1. Then by taking the coordinates of the top left of the pitch, found in section 3.3.1, and adding new x and y position just found, the sensor point for each player should be plotted onto the pitch. However, this caused the sensor readings to be placed incorrectly off the pitch. After some investigating it was found that during the process of transforming the pitch, the ratio of the width to height was changed causing the pixel to yard value to be incorrect. This problem was attempted to be solved by finding the correct pitch height and width, found in [8], and the image of the pitch was transformed to the same ratio. To get this ratio, the pitch width had to be increased equally at the top and bottom of the pitch. The pixel to known measurement is also found again but this time using the height of the pitch and this known measurement from [8].

Even after this, the player sensor data did not match the players of the frame correctly. As this issue could not be resolved, the ZXY data could only be used as ground truth for the distance travelled, with the sensor data being plotted and the players being matched visually to the sensor ID by comparing the position to the video itself.

Furthermore, the issue of the sensors taking 20 readings per second compared to 25 frames in the video would need to be addressed. If this is not addressed then the sensor readings will slowly start to run ahead of the frames causing incorrect results. A potential solution could be on every 5th frame the sensor data used on the 4th frame is used again. However, the issue of not having the correct position on the pitch for the sensor data still meant this possible method was infeasible.

A previous year student [2] used colour to group the extracted contour to sort the players, however; the data set that was used to do this had players wearing different coloured clothing meaning it was easy to categorize each player. The dataset in this project had only players wearing one of two colours for their team so this technique would not work with this dataset.

The method I finally decided to use to check player classification was to manually check the video frame by frame. Because of the manual checking required, only the first 20 seconds of frames are checked as without this it would not be possible to run the tests required in the time frame available. All players detected initially will be manually tracked.

4.3 Player Identification with Stored SIFT Key Points from a Different Number of Images

The program stores a number of image SIFT key points for each person. The program is run with 5 frames to build the background loader. This results in:

Images SIFT key points Stored	Precision	Recall
1	48.75	26.30
5	53.61	29.45
10	51.33	28.15
20	53.73	29.33
25	54.99	27.89
30	51.06	28.54

Table 4.2: Altering the number of image SIFT key points stored

When running the experiments it was found that some players got the same recall and precision throughout the altering number of image SIFT key points stored. This occurred as these player stayed a distance that was far enough away from other players that the search radius of the contours produced by the other players would not have their location within. This meant the only checking contours for these players would be the occlusion noise contours and the correct contours by the player. This meant increasing the number of image SIFT key points had no effect on the outcome as no other contours were produced. Some of these players had the contour that should of been assigned to them lost. This occurred as player contours was not detected and when the players contour was detected again the player was outside the radius that was allowed for classification for the contour.

Images SIFT key points Stored	Precision	Recall
1-30	77.84	31.97

Table 4.3: Altering the number of image SIFT key points stored: only values with Constant value

This recall and precision displayed in table 4.3 is not higher due to an issue with player detection. Player detection finds the square around the player. This area can sometimes

contain key points found by SIFT that does not belong to the player themselves such as shadows. If these noises are detected as a contour then it will have similar key points to the player's due to its key points being found prior. This will mean the player could be assigned incorrectly to a incorrect contour by mistake. This incorrect contour will be stored meaning it becomes increasingly likely it will be assigned to this unwanted contour again.

Figure 4.1: Screenshots showing shadow being detected and identified as the player incorrectly.

To fix this issue, a method which better encases the player could be used where unwanted key points are not found when SIFT is applied. Furthermore, this issue not only occurred for noise but also when players are detected within the same contour. This normally happens when occlusion occurs which is where a player overlaps another player. After the players move apart it is likely that the incorrect player will be misclassified as the player the occlusion occurred with. If this occurs it is likely the incorrect player will be tracked until another misclassification occurs or the correct player is classified by chance.

Images SIFT key points Stored	Precision	Recall
1	28.62	22.38
5	36.90	27.82
10	32.98	25.50
20	37.04	27.50
25	39.76	24.90
30	32.52	17.76

Table 4.4: Altering the number of image SIFT key points stored: Constant value removed

Table 4.4 shows an extremely low recall and precision when number of SIFT stored is equal to one. The main reason for this is that when storing only one feature, if a misclassification contour occurs, the misclassified image will replace the correct image stored. The program will be trying to find contours that are similar to the misclassified key points found instead of any correct key points. This means that the only way the program can classify this player to what the correct contour would be is to misclassify a contour again to the original player.

Additionally, one potential reason for the difference in recall and precision between storing one SIFT contour key points and a larger number is the fact that SIFT is rotation invariant in the X and Y axis but only partially in the z axis. This means if a player rotates too much in the z axis, which happens often in a football match, new SIFT features will need to be captured to identify this player. When SIFT size is large there is an increased chance that the player has already had their SIFT key points captured at this position. This means if the player turns into this position it is more likely to be classified to the correct player with a larger SIFT.

One final potential reason is that if a player is correctly identified in the first few frames and a large SIFT key point storage is used then this player will have many correct key points. This increases the chances of the player being correctly identified during the match, as with more correct key points stored there is an increased probability of the correct player being identified for a contour. However, this has the side effect that if many incorrect image SIFT key points are stored early on, the more likely it is an incorrect contour is classified incorrectly.

Table 4.4 also shows a varying recall and precision between 5-30 SIFT key points stored. One potential reason for this is due to the randomly removal of player SIFT features. After a certain number of key points of an image has been reached a random image key points are replaced with the new contours image key points. This could be replacing a potentially good classification image key points with an incorrect image key points or the opposite where an incorrect contour image key points are replaced with a correct image contours key point. This could cause the varying in values shown.

Overall, recall was extremely low for 4.4. This issue was caused by the players not being detected on every frame. This happened partly because of the background model not removing all shadows. This means when the contours are found the shadow and the player is found as one contour. This sometimes gives the contour a width larger than the height. This meant the program identified this contour as noise instead of a player. The requirement of having a larger height than width cannot be removed due to the fact many noise contours are filtered with using this method. This could be a more effective shadow remover.

Another reason for the low recall was due to the method of background subtraction. As the model updates if players remain still the pixels containing the players remain constant. Because of this, the background model assumes this area is static and therefore is part of the background, meaning the player will not be detected. This should not affect the final result for distance travelled as if the player is not moving the distance travelled is not increasing either. If wanted, this effect could be minimised by applying a lower learning rate to the background model after the model has been initially built. Then players would have to remain still for longer to be considered as background, which would be unlikely to occur due to the constant movement required for the sport.

4.4 Player Identification with Different Search Area

For each player a contour must be within a certain radius to be able to be classified as that person. By changing this radius you affect how many potential players a contour belongs to, affecting the classification that takes place, overall changing the precision and recall of player identification. By changing the radius the results were:

Table 4.5 shows an extremely high precision when search radius is small. With a small radius there are less contours in a players radius. It will be likely that the correctly detected contour will be in this area near the start of the program and be assigned correctly. However, over the course of the video, player movement will occur without being detected. With a small radius the player stored location will be far enough away from the real player in the image that it is not within the radius anymore. After this occurs

Search area	Precision	Recall
20	82.28	18.46
60	73.05	17.44
120	48.1	26.32
All Pitch	12.92	12.17
Mixture of 60,120	55.34	27.79

Table 4.5: Altering search area

the contour will stay in the same position until player movement is detected within this radius which it could potentially classify to. As the radius is extremely low it is unlikely a player will come into this radius. This causes a small number total assignment and a even smaller number of correct identified classifications causing a small recall.

When the radius size is increased, precision decreases while recall increases. This is because the number of correctly classified players is increasing although the number of incorrect classification is increasing by a larger amount. Incorrect classifications are increased as with the increased radius of the players where more contours are within it. This increases the number of contours the player could be incorrectly classified to, increasing the probability of a miss-classification occurring.

The importance of temporal coherence is shown in 4.5. Without temporal a much worse precision and recall is achieved. Furthermore, it was tested to see if a mixture of having two radii could gain the advantage of having potential incorrect players within the contours area while still being able to track players that have been lost due to poor player detection. This was done by having one search area radius of 60 and another of 120. The achieved outcome resulted in a high precision and the second highest recall achieved.

4.5 Distance Accuracy

Before finding the distance travelled by players some parameters values have to be defined. These include:

- The number of frames to build the model.
- The number of image SIFT key points stored for each player.

- The size of the radius a contour has to be in to be considered needs to be defined.

The number of frames to build the model is kept at eight as this achieved the joint highest precision and recall as shown in table 1.4. For the number of image SIFT key points stored for each player, five will be used as it achieved the second highest precision along with the second highest recall in table 4.4. For the radius a contour has to be in to be considered a possible match, two radii will be used as this achieved a average precision with a good recall as shown in table 4.5. With these parameters set the achieved result was:

Sensor ID	XYZ Distance	My Results	Accuracy
1	43.93	5.82	13.24
2	47.78	18.47	38.66
4	47.86	6.43	13.44
5	44.24	2.11	4.76
6	39.56	2.11	5.33
8	22.77	7.5	32.94
11	59.24	4.27	7.21
14	49.24	6.43	13.06

Table 4.6: Distance Accuracy

Total Precision	Total Recall	Total Accuracy
54.24	26.17	16.08

Table 4.7: Distance Accuracy precision, recall, and total accuracy

The results from table 4.7 show an extremely low total accuracy. Part of the reason of such a low accuracy is because of the low precision of the system. If the player cannot be correctly identified then the player's position cannot be fully tracked, thus the distance travelled will not be accurate. Only when the issue of player identification is solved will this accuracy see any increase. Furthermore, the issues with the player detection will also need to be resolved. The issue of some player movements not being detected because of occlusion will cause the accuracy to go down due to the player's movement not being recorded. One other potential reason why the accuracy could be so low could be down to the same issue with the sensor data being placed incorrectly onto the pitch. This issue maybe one big problem or many smaller issues together.

Chapter 5

Reflection

Overall this project was successful. It provided the basework to locate the pitch area, identify the players and track the distance travelled during a football match.

5.1 Project Achievements

The project covered many areas of computer vision and some inventive methods were created to solve some problems. The results from the program itself may not have achieved high accuracy, yet with the number of issues that occurred during the project it is an achievement that the program outputted a reasonable result. Some notable sections of the project were:

- Locating the pitch area: This was done using a number of computer vision techniques which should be able to perform on many other pictures automatically with reasonable accuracy.
- Locating the corners of the pitch: The method created worked extremely effectively which can again be deployed on other pitches.
- Detect players from the first frame: This method allowed the players on the pitch to be found without needing to use and remove an image of the foreground.
- The number of methods used: This project was overall extremely large covering

many computer vision techniques. This gives others the ability to build upon this work to further improve the result.

5.2 Project Improvements

The project had many unexpected challenges that with hindsight could have been avoided or better fixed. Some of the main issues were:

- Non-calibration camera causing the barrel distortion: This issue was partly fixed with the estimation of the calibration coefficients but the majority of the distortion was still present. This issue has a simple fix of using a dataset where the calibration matrix is known. If there is not a dataset with the correct data available then the dataset itself could be created, as the time taken to create the estimation of the calibration far exceeds the time this would have taken.

- Another way the project could have been improved is by using tracklets. This was avoided at the time due to time restraints but [1] and [2] achieved a more accurate system when tracklets were used.

- A third way the project could be improved is by using more cameras round the pitch. This would help fix the occlusion problem where the system finds it difficult to detect two separate players if they overlap. By having multiple cameras round the pitch the system should be able to find an angle where no one is hidden behind any other player.

5.3 Future Work

The project could be further developed in many ways. Some examples of future work that could be done include:

- More statistical analysis could take place. This could be along the lines of tracking the top sprint speed of each player, measuring the areas of the pitch most players

spend time in, and estimating how many calories a player has burnt. If the ball was able to be tracked then the number of passes, tackles and headers could be recorded adding to the collection of statics. However, before any work on this is possible the program needs to start correctly identifying players the majority of the time.

- The program could be furthered by making it run in real time. This could help assist managers in making decisions during the match. For example, one way the program could assist is by providing stats to see who has run the furthest. The players who have run the furthest are more likely to be the most tired so it could be argued they should be substituted.

- The program could be adapted to learn many statistics from a number of matches. This would mean the program could use the data from a number of opponent team's matches and predict their potential line ups for upcoming games. This would mean the manager could adapt the team sheets accordingly.

5.4 Conclusion

Overall, this project was extremely difficult but incredibly rewarding. Throughout the project there were constant issues that took days if not weeks to be solved. For a large proportion of the project I believed I would not have the time or the knowledge to complete the code to finish the program. The program does not work fully as intended with very poor accuracy for distance travelled. However, I believe with the right knowledge and time this accuracy can be vastly improved.

The aim of this project was to produce a cheap alternative solution to tracking players. The project did not fully achieve this objective, but it was unrealistic to achieve such a large aim within the limited time available. What the project does provide is groundwork which, if further worked upon, could allow this system to provide a cheap alternative tracking system solution. Because of this I believe the project was a success and that the work done has been helpful for potential future systems.

Bibliography

[1] Akerman, S. Tracking Football Players, 2018.

[2] Dunlop, A. Tracking Football Players, 2017.

[3] KaewTraKulPong, P. Bowden, R. (2002). An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow, *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, Boston, MA, Springer US. pp. 135-144.

[4] Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 60(3), pp. 91-110.

[5] Opencv Library and Documentation. (2020). Camera Calibration Tutorial. [Online]. Available at: `https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html`, (Accessed 11/03/2020).

[6] Pettersen, S. A. Johansen, D. Johansen, H. Berg-Johansen, V. Gaddam, V.R. Mortensen, A. Langseth, R. Griwodz, C. Stensland, H.K. and Halvorsen, P.(2014).'Soccer Video and Player Position Dataset', *The International Conference on Multimedia Systems (MMSys)*, Singapore, March 2014. Association for Computing Machinery, pp. 18-23.

[7] Tennøe, M. Helgedagsrud, E. Næss, M. Alstad, H.K. Stensland, H.K. Gaddam, V.R. Johansen, D. Griwodz, C. and Halvorsen, P. (2013). Efficient Implementation and Processing of a Real-time Panorama Video Pipeline, *Proceedings of the International Symposium on Multim*, Anaheim, CA, USA, December 2013. IEEE.

pp. 76-83. Available at: `http://home.ifi.uio.no/paalh/publications/files/tomccap2014-bagadus.pdf` (Accessed: 23/03/2020).

[8] Tromso IL, 14/03/2017, *Alfhem Stadium*, Avaliable at: `https://www.til.no/english/alfheim-stadium`, (Accessed: 20/03/2020).

[9] Wu, F. Wei, H. and Wang, X. (2017). 'Correction of image radial distortion based on division model', *Optical Engineering*, 56(1), pp. 1-12, Available at: `https://www.researchgate.net/publication/312872991_Correction_of_image_radial_distortion_based_on_division_model` (Accessed 11/03/2020).

[10] Zhang, Z. (2000).'A flexible new technique for camera calibration',*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pp. 1330-1334. Available at: `https://www.researchgate.net/publication/3193178_A_Flexible_New_Technique_for_Camera_Calibration` (Accessed 11/03/2020).