Cardiff University Computer Science
One Semester Individual Project 2016
CM3203 - 40 Credits

AUTOMISATION OF SPORTS VIDEO ANALYSIS

Final Report

Author: Thomas Morley

Student ID: C1217615

Supervisor: Prof. David Marshall

Moderator: Dr. Federico Cerutti

Contents

List of Figures

Abstract

This document covers the final year project based on the automisation of sport analysis. It covers the background to the project, the problem it is trying to fix and the implementation taken to achieve the aims set out. The project looks to give a measurement to the structured plays that can be extracted from phase play in rugby union. Although this project is based around rugby union it could also be used for other sports as the theory behind the project supports any structured team games.

The project looked to understand what measurements could be drawn out from the video footage of a phase play. The measurements taken allows the user to measure and compare plays across a number of phases. To do this players are tracked through a coordinate system and then a frame by frame analysis builds a picture of the phase. Selected players will form basic shapes during these phases and this allows the program to then measure specific properties of the shapes made. It looked at how the team's structures are looked at by the coaches and analysts and tried to use these shapes to help the coaches explain what is happening during a phase play. During phase plays teams will want to maintain a certain value on these measurements. For example keeping the depth of the backline in attack and the distance between players.

A program has also been designed to help develop the measurements and prove that the measurements designed work on the phase plays. This program takes input from the user like selected players in the phase and then the KLT tracking system that is built into Matlab is used to help track the players within the co-ordinate system. The program also then helps display the statistics gathered by graphs to help show the changes over frames. It is also shown graphically to the user by inserting shapes into the frame. This is an important part of the research as the displaying of the results is where the user will have the most interaction and use out of the program.

Acknowledgments

During the project lifetime there have been people that have helped progress the project and its findings. First is my supervisor Professor David Marshall for the constant support given and the meetings set to help keep the project on track. The guidance that has been given has helped during the progress of the project.

Secondly, Aberavon Harlequins RFC coaches and players. During this project questions when asked have been answered and a review of the project has also been conducted by the coaches. The input has been invaluable and was a major factor in the findings of this project.

Table of Acronyms

Acronyms	Description
avi	Audio Video Interleaved (Video Format)
GPS	Global Positioning System
GUI	Graphical User Interface
KLT (feature tracking)	Kanade–Lucas–Tomasi (feature tracking)
RFC	Rugby Football Club
UI	User Interface
WRU	Welsh Rugby Union

Introduction

This project looks to help build on the collaboration the university has with the WRU (Welsh Rugby Union) and Swansea University. Today's professional sports looks at all aspects in fine detail. From the engineering of equipment and plays, to the analysis of a person or team. A lot of this is done by experts in their relevant disciplines to help maximise the potential of a team or player in their sport. Sports are now looking into using technology to help with this process and it is the desire to hit perfection that drives this forward. Sport has always done this through the expert's point of view and this technology should look to help them not take this away.

The project looks to help with the tracking and analysis of plays within rugby. Although this looks at rugby union it could also be used in rugby league as the plays and movement are highly transferable. Currently games are annotated and are analysed using expert sports coders employed by the WRU. During games key points are noted down and these are then analysed and synced to the game video for in depth match reports.

Systems currently allow for the combination of people to work on the video to gather data on the game. Key events during games are events such as line outs, tries and scrums. An issue with the game of rugby is that it is a contact sport and multiple bodies can overlap. This causes the systems issues as the structures like scrums and mauls do not have very well defined edges between players. This is why many systems use a combination of tags (GPS tracker and accelerometer) and camera systems to track players. Examples of systems that use this technology within rugby and other sports are below:

- ChyronHego - http://chyronhego.com/sports-data/tracab
  - Used in Bundesliga (German top Flight Football)
- Sportstec - http://sportstec.com/
- isoLynx - http://www.finishlynx.com/isolynx/

These systems work on the bases that all the data gathered and entered are from external sources. The external sources would be the GPS tracking devices and external notes entered by the expert. This system will look at creating the

data from the analysis it does on the video. Not only would this allow the analysis of the home team but also hard facts about the other team that other systems cannot give you. This feature would give teams an advantage in areas as this data is usually not accessible to the opposition.

Aims and Objectives

This project has 3 main aims. The aims were the areas that the development must focus on for it to be made a success. During the development the aims were checked against and short term targets were changed to make sure the goals were achieved. The goals are not strictly defined due to the nature of the project. The project is to see if creditable information could be retrieved from the tracking of players. The lack of knowledge of what was possible meant that the aims must have been adaptable during the project.

Secondary Aims were set if the development was quicker than originally thought. These extend the implementation of the features and even if these are not achieved the project will look at how easy it would be to implement each aim. The secondary aims are also listed below.

Main Aims

- Understand what analytical features are useful to the target audience
    - If the program is to be used then useful features will be needed
    - Break down plays so a basic analytical statistics can be looked into
    - When basic analytical statistics are found check they are useful to the target audience.

- Produce a working program that allows a person to run multiple phases of play and retrieve tangible data on the events that occurred.
    - To produce a result for the aim above the basic analytical features produced must have a wide variety of both attack and defence.

- Produce a very basic and usable UI.
    - The analytical features are the most important feature but a useable UI must be implemented.

- Depending on the basic analytical statistics found various methods may be used to show the data. The presentation of the data will not be known until approved statistics have been found.

Secondary Aims

- Develop more advance analytical features
    - For more advance moves within a sport different types of data will be captured. This will be an advancement on the features that will be implemented so features are to be made editable.

- Linking tag data to tracking.
    - This would help improve the tracking feature and could possibly give more statistical data that can be used

- The program will only work on clips that have been cut into phases if the program could have a full clip run through it and results produced it would remove the tedious task of clipping the video.
    - This is currently to advanced for the project but could be achievable in future developments

Target Audience

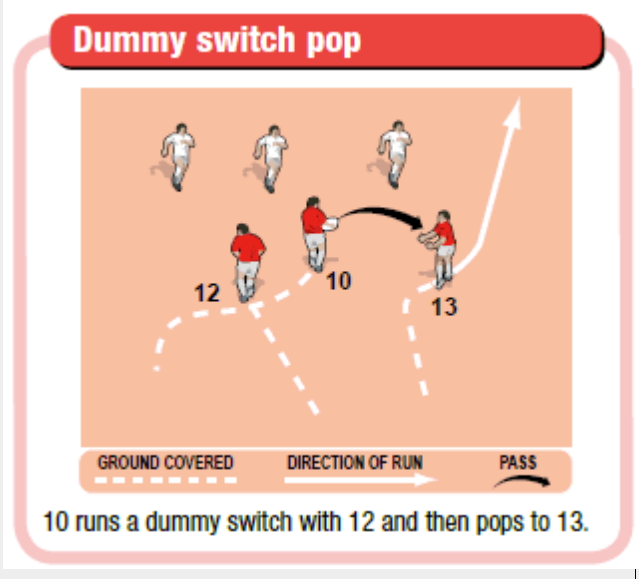
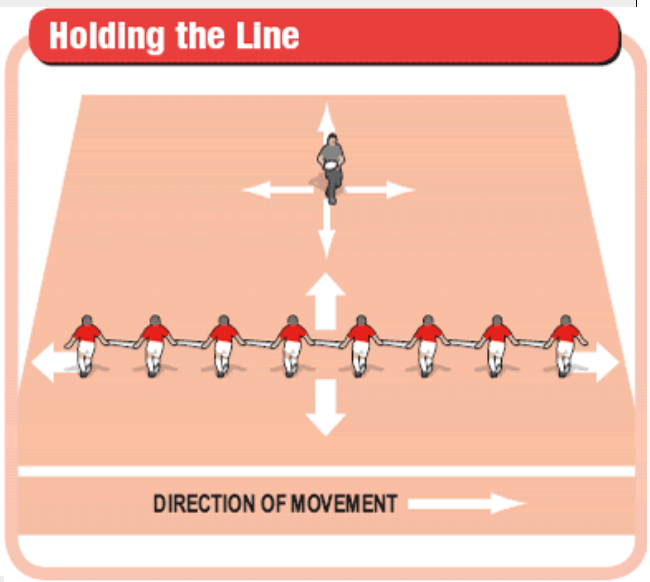
As explained above program will be developed with rugby as its sport. This means the features being implemented will be targeted for specific plays. Although the main target audience is the rugby community if the program achieves the aims set it is also proof that this type of program could be developed for other sports. For example American Football and Australian Football. This would need a change for plays but the theory would stand.

The main target audience for this project are the analysis teams and coaches that would find a system like this useful for post-match reviews and team talks.

Background

As explained in the introduction professional sports now look for every advantage to get into a winning position. Rugby is no different as it looks to use defined tactics and development of team and player styles to create a winning formula. The program being proposed looks at tracking players during plays to be able to define statistics that can be useful for the analysis of the tactics.

Rugby at a certain level becomes a very structured and tactical game. There are 2 main aspects to the game. Attack with the ball and defence, opposition with the ball. Both can be very structured plays and it is this structured play that allows us to develop statistics on the plays that are seen during the game. Attacking structures look to take advantage or manipulate the defence so ground is made. While defence plays looks to shut down the attacking plays and stop the attacking team from gaining ground or for them to turn the ball over. Below are 2 examples of basic attacking and defence moves:

Figure 1: Attacking Move	Figure 2: Defence Move
Figure 1- Attacking Move	*Figure 2- Defence Move*
	
Source: http://www.betterrugbycoaching.com/Article-1694-109-Rugby-Tactics,-Strategies,-Tips-Moves-and-Plays-Rugby-drill-to-coach-a-simple-attacking-move	**Source:** http://www.betterrugbycoaching.com/Article-1432-108-Rugby-Tactics,-Strategies,-Tips-Defence-and-Defensive-Systems-Holding-the-line-rugby-drill

As both aspects of the game looks to manipulate the other it is useful to understand when something is about to happen or find a solution to why something is happening. At the professional game the faults found and the ability to manipulate a team becomes very difficult as teams are trained to understand when to do certain tasks and how to stop the manipulation. This is done through expert analysis of both their own teams and that of the opposition.

In the introduction it stated that the software solutions to help with this analysis relies on the sports experts to enter the data and for external sources. This project will look at creating extra data that has measurable results. At the moment these systems do not give this type of information as it is all from the expert's interpretation not giving any measurable results.

A system that would give them this information would mean that they could become more precise on the analysis of moves and player positioning. It would also allow for better comparison of methods and plays within the game as data could be compared for a better understanding of what works. For example, if a certain play works because of a certain statistic it could mean that this could be introduced into other plays or train the team to perfect such a move.

The main reason for this development is to help produce extra data that could be useful to the analysis team. It will not remove the interpretation of the sports expert but is looking to make his analysis better and easier to reproduce for use in team talks etc.

Tracking System

The tracking system being used in this project is the KLT feature tracking system (Kanade–Lucas–Tomasi). This system has inbuilt libraries within MATLAB and was also set out as the system to use in an engineering student's summer project for tracking players. The student's project outlined the libraries needed and what type of data could be retrieved from this type of tracking.

The coding and use of the KLT tracking system has been based on the references and guides set by Matlab documentation. (The page is located at http://uk.mathworks.com/help/vision/ref/vision.pointtracker-class.html.). KLT

tracking is based on a paper published by Jianbo Shi and Carlo Tomasi (1994) 'Good Features to Track'. This project uses the Max Bidirectional Error property which is a Forward-backward error threshold.

Explanation of the property taken from the Matlab Documentation:

> *If the value is less than `inf`, the object tracks each point from the previous to the current frame. It then tracks the same points back to the previous frame. The object calculates the bidirectional error. This value is the distance in pixels from the original location of the points to the final location after the backward tracking. The corresponding points are considered invalid when the error is greater than the value set for this property. Recommended values are between `0` and `3` pixels.*
>
> *Using the bidirectional error is an effective way to eliminate points that could not be reliably tracked. However, the bidirectional error requires additional computation. When you set the `MaxBidirectionalError` property to `inf`, the object does not compute the bidirectional error.*

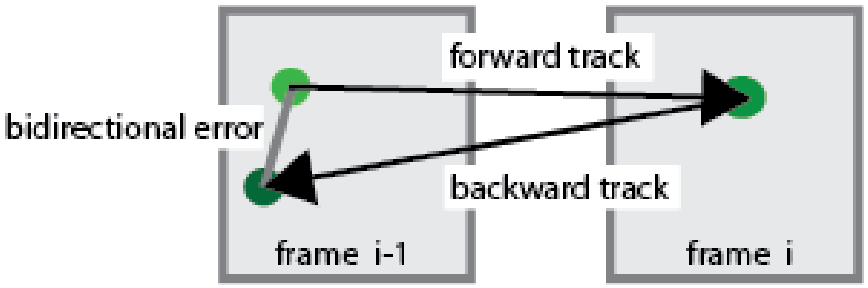
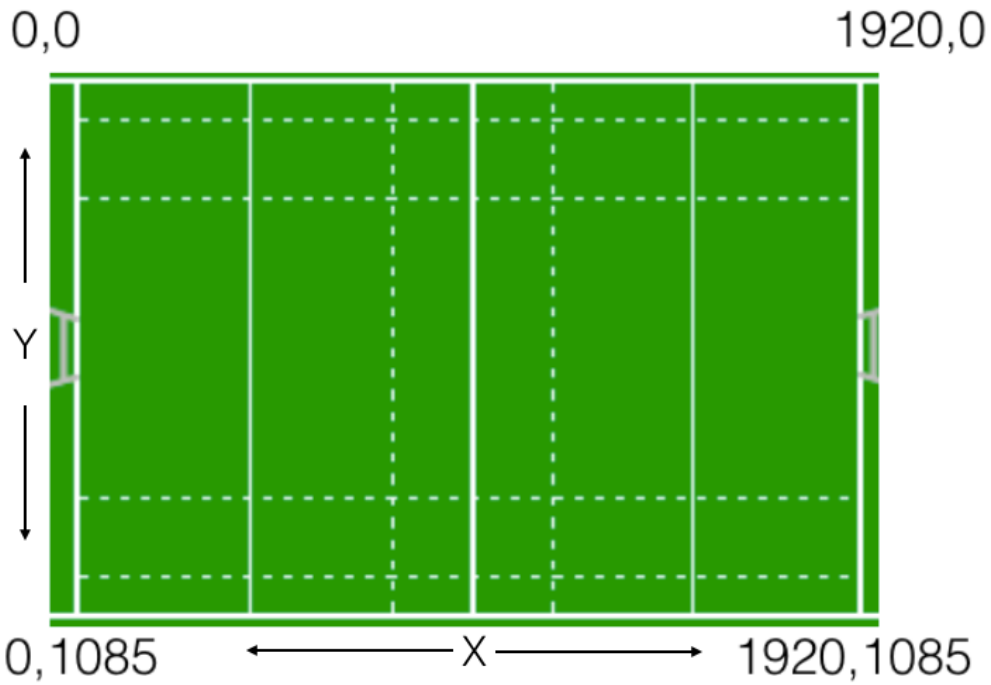


Figure 3 - Bidirectional Error

Data produced from the student's project was good for the use in this project as it produced a frame by frame co-ordinate reference for each player selected. This made it a very valuable system and was the starting point for the project bases. The MaxBidirectionalError was set to 2 pixels which is in the recommended range as suggested by the Matlab Documentation.

Figure 4- Pitch Coordinates



Issues with tracking system

Section of plays within rugby rely on many aspects when playing the game. Below are a list of parameters that effect these decisions.

- Position of play on pitch (X axis & Y axis)
- Number of players in attacking line
- Number of players in defensive line
- Type of player in line (Forward or back)
- Space in behind (When to kick)
- Where the ball is (Who has it)

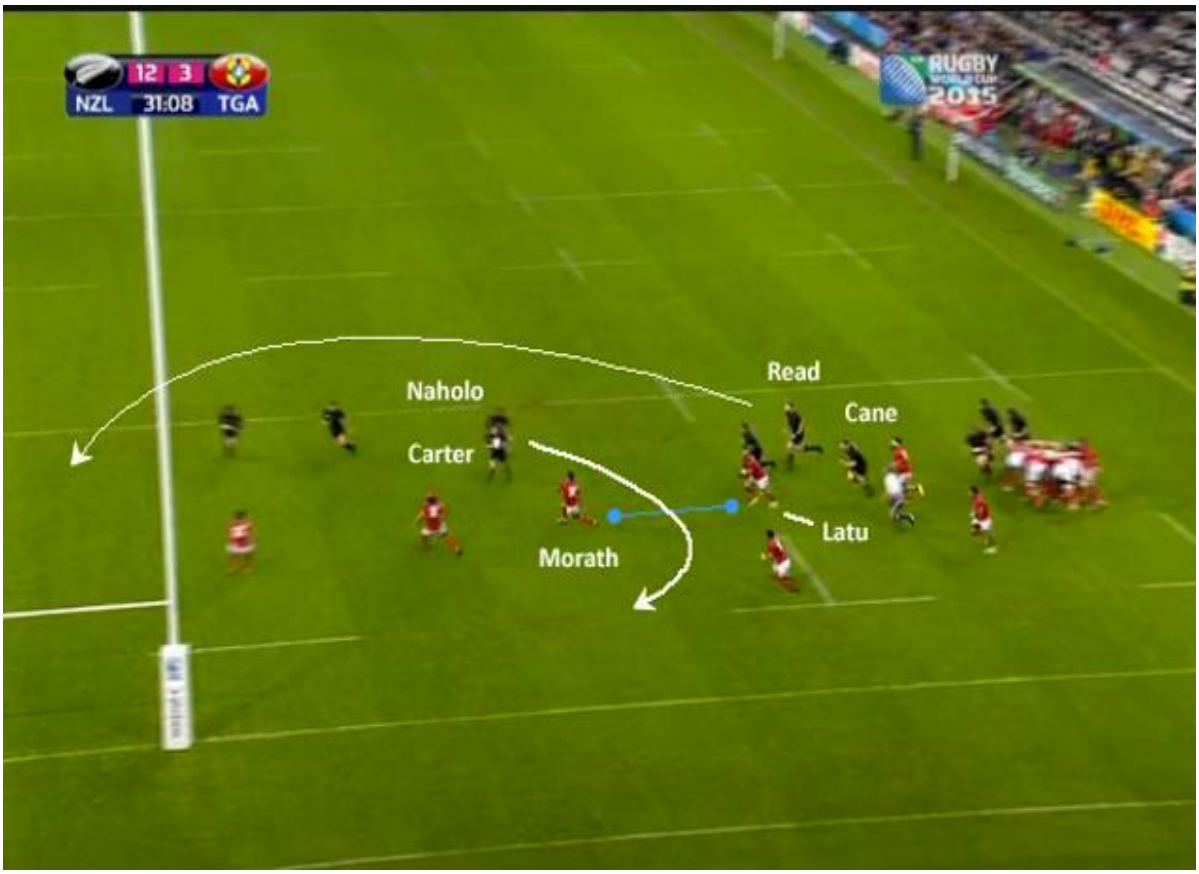
It is a combination of all these parameters that effect the choice made this makes it difficult for the program to decide which play is going to be made and is why the project will only look to help and not replace the experts.

The program will struggle to define some of these parameters using computer vision techniques looked at during this project. It is difficult as the ball and players will sometimes be lost during the video footage. Also the identification of players would be an issue. This means that it makes the computer decision making (learning) difficult on plays. In the future development of this type of

software with GPS tracking and computer vision techniques this could see some of the parameters easier to measure.

Although these issues remove the task of machine learning. There are still issues with the tracking system because of rugby being a contact sport. This means that the project will be limited to phase play within rugby.

Figure 5- Phase Play



http://www.rugbyworldcup.com/news/112396

Phase play is the time the ball is in play between breakdowns (rucks, line outs, restart of play).

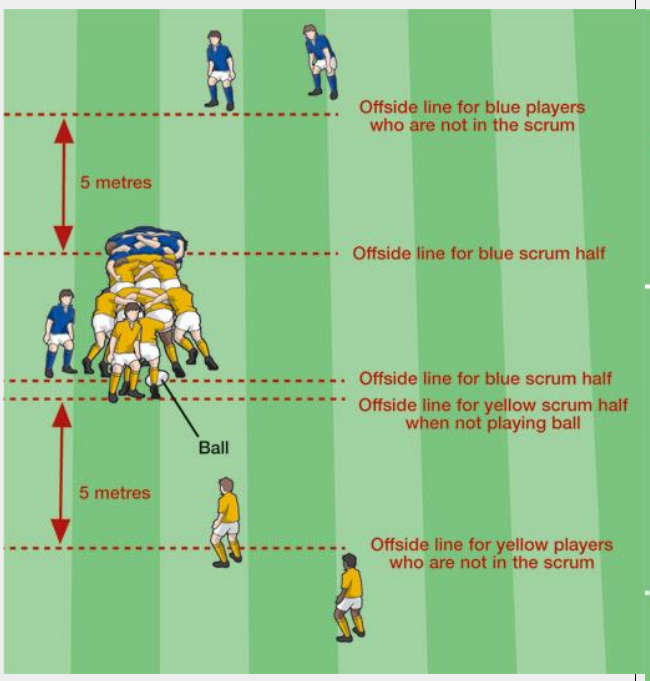
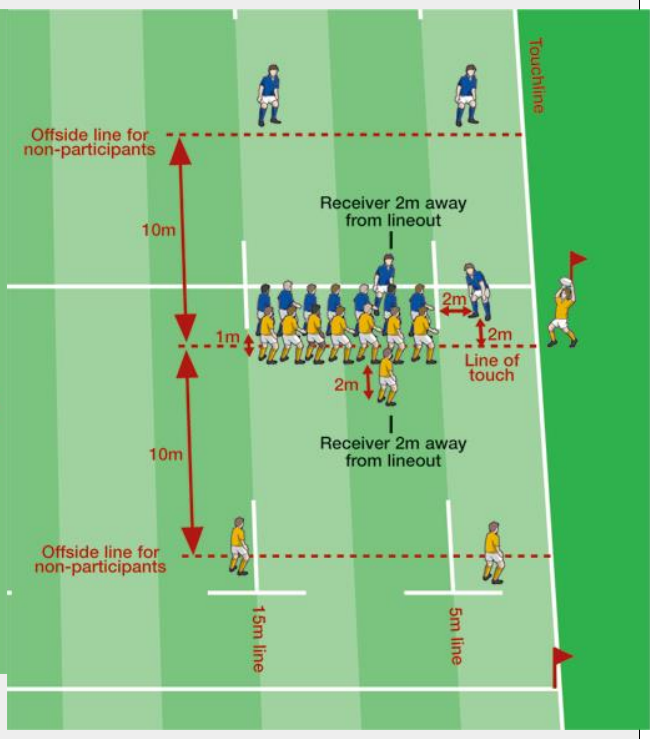
Scrum	Lineout
Figure 6- Scrum	*Figure 7- Lineout*
	
Source: http://laws.worldrugby.org/?law=20.12	**Source:** http://laws.worldrugby.org/?law=19&language=EN

Figure 8- Spinsight Frame

It will also not look into the forwards work during the restart plays. Restart plays are events such as line outs and scrums where plays tend to be technically different to phase play and players are in contact with each other. Therefore the project will not look at gathering data for restart plays.

External Sources for Project

This project required high quality video footage of the games so tracking of the players can be made easier during the processing stage. This has been provided by the university with access to the WRU Spinsight camera. This gives a good overview of the pitch and the players on it. This means that there is no need for additional processing to compute on off centre cameras. Below is an example of a frame captured from the Spinsight camera.

MATLAB

This project has been developed in MATLAB due to the accessible computer vision libraries it holds. Also as stated above the previous tracking method highlighted in the student's summer project used a series of MATLAB libraries that produces a useful result that was useful in the development of the project.

Matlab also allows for very basic UI design that can be used to help develop the measurements. A UI will be needed as the program will need user inputs and a way of displaying the data that has been captured by the program.

Outline of Proposed Program

To make sure that the aims are met in this project a program has also been developed. This section outlines a high level overview of the program that is to be made so the measurements created can be developed and analysed for their usefulness.

The development of a UI is not one of the main aims within the project. So the requirements for the program is currently very basic. They are mainly there to help improve the speed and reliability of the tracking system.

Program Requirements:

- Take user input – Video, player selections
- Output the data captured – Graphs, tables etc.
- Allow user to run multiple scans on a phase
- Try to speed the tracking process up through simple methods.
- Output the phase that has been tracked
- Have a save and load feature so phases are not tracked multiple times by the user.

To understand what information was needed from the user research was done on what are the common factors in the different types of phase play and these were then checked to see if the user would have to define the factor or if the program could do it. Below is the research.

User Inputs

As the project relies solely on the video footage produced it needs extra inputs from the user to allow the program to then make useful statistics. During this project plays were analysed to see what the common inputs needed for the statistics were.

Plays were broken down to see what could be detected by the system and what would be needed from the user. Below are the 3 play types that will be analysed in this project and what would be useful.

Attacking Play

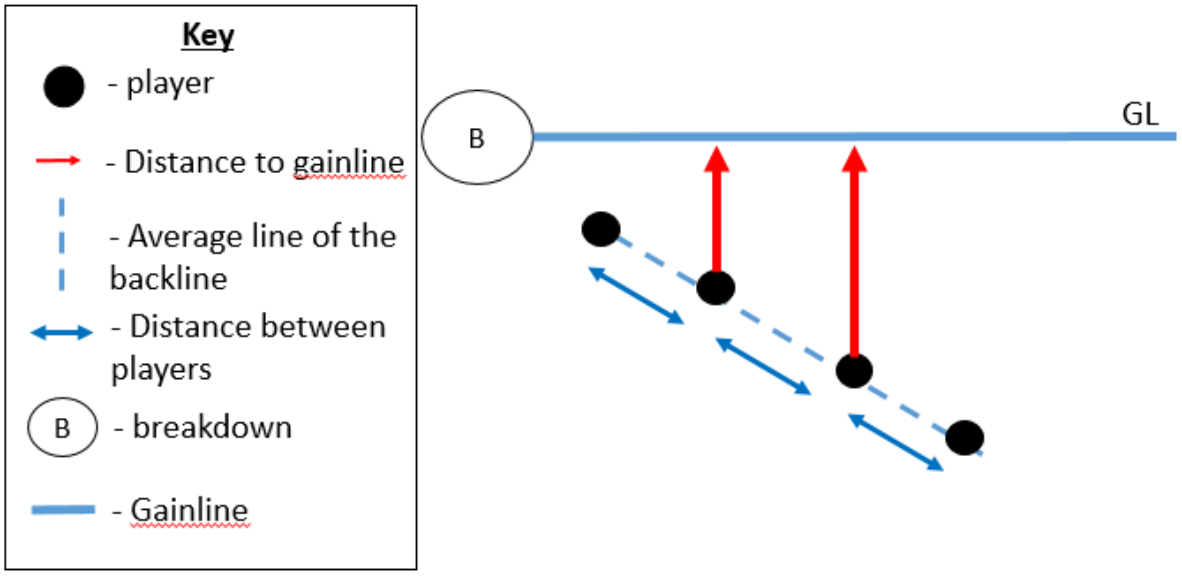


Figure 9 - Attacking structure

(1) **Players** – The players involved in the phase play will need to be identified as the program could not tell who is involved.

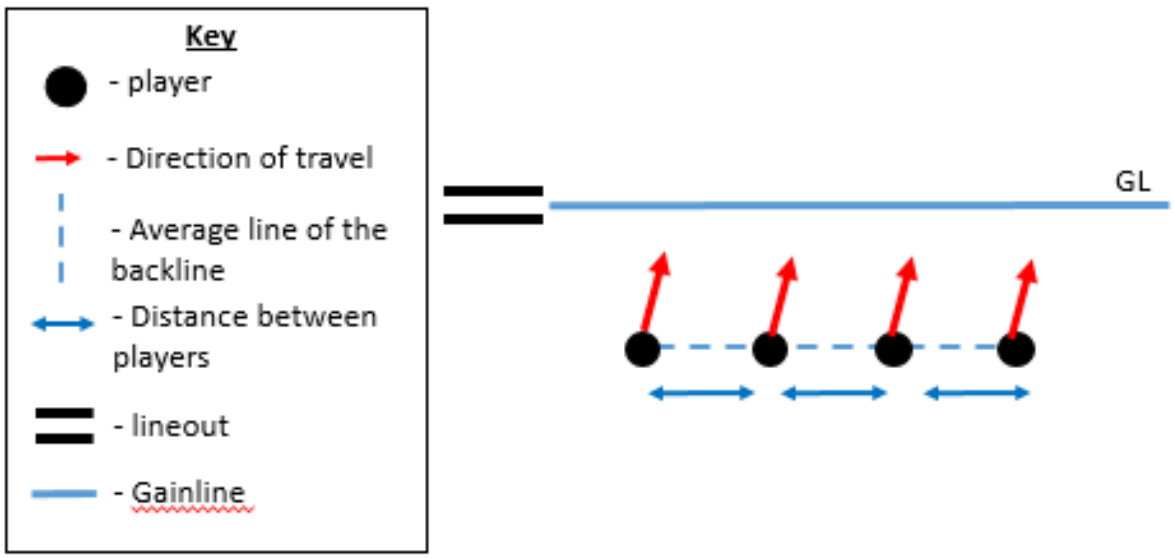
(2) **Gainline –** It may be possible to set the gainline due to it being at the center of the breakdown. But this relies on a detection system being built. So having the user enter it would be simpler and faster.

(3) **Distance to gainline –** This can be measured by the program.

(4) **Distance between players –** This can be measured by the program.

Defence Play

Figure 10 - Defence structure



(1) **Players** – The players involved in the phase play will need to be identified as the program could not tell who is involved.

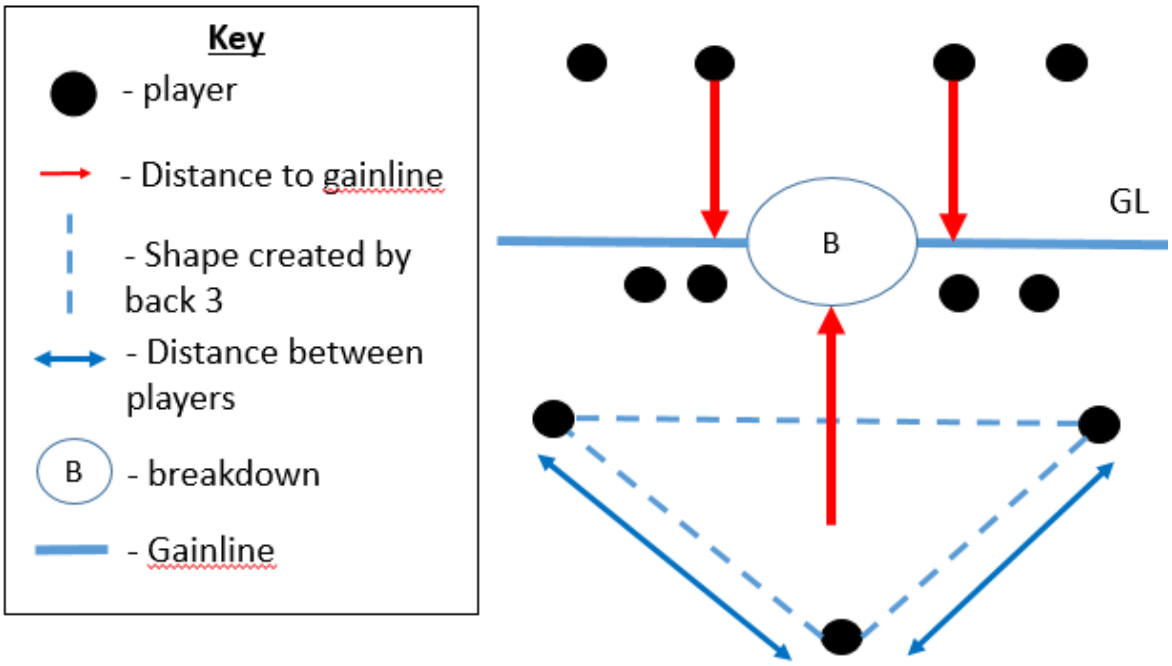
(2) **Gainline –** It may be possible to set the gainline due to it being at the center of the breakdown. But this relies on a detection system being built. So having the user enter it would be simpler and faster.

(3) **Player Direction of travel –** This can be measured by the program.

(4) **Distance between players –** This can be measured by the program.

Kickoff / 22 drop out

Figure 11 - kicking structure



(1) **Players (both sides)** – The players involved in the phase play will need to be identified as the program could not tell who is involved. It also shows that identifying the opposition is useful as their positioning impacts the kick being made.

(2) **Gainline –** It may be possible to set the gainline due to it being at the center of the breakdown. But this relies on a detection system being built. So having the user enter it would be simpler and faster.

(3) **Shape of back 3 –** From the extra analysis the shape of the back 3 has a major factor in the defending of a kick. This can be measured by the program.

(4) **Distance between players –** This can be measured by the program.

(5) **Distance to gainline –** This can be measured by the program.

From the research above 3 main inputs from the user were needed.

Distinguish between 2 teams

The system will not be able to distinguish between the 2 teams therefore when the user is selecting the players it must ask for the user to also help distinguish which player is on what side.

In future development this could be removed by using extra input data. For example the player GPS coordinates from the player packs.

Players involved in the phase

The player selected will have to be the ones that will be involved in the play. When additional players not involved with the phase play are selected they become noise. This noise would skew the group statistics effecting the quality of the results.

Gainline

As stated in the background section the purpose of the game is to manipulate the other team to gain ground. The point at which positive ground is made is the gainline. The gainline is an imaginary line that runs across the pitch parallel to the half way line and is set where the breakdown or restart play has occurred on the pitch. This is a useful feature as a number of statistics are linked to the gainline and shows clearly the effectiveness of the play.

With further research the gainline could be set by detecting when a breakdown has occurred or where the restart play is located. This is not present in this project due to the timescales and extra research needed.

From this the program will take 4 inputs from the user:

1. Video of the phase play
2. Home team players in the phase play
3. Gainline of phase play
4. Away team players in the phase play.

Running of the program

The program will have to have some time saving aspect built into it as the tracking of players will be a long and tedious task that the user will not sit through every time. Although the speed of the program is not in the aims for the project these developments will be fundamental in the future development of the program. Below are the areas where the program will be developed.

- **Re-tracking of players** – Players will only be tracked once in a phase. As more players are tracked the processing on the clip is reduced. It also allows for multiple tracks to be run on one clip.
- **Comparing players positions** – This will be needed if the re-tracking of players is going to work. Here players selected will be compared on their start positions if a player is identified then it will return the comparison as a positive.
- **Save / Load File** – This will allow the users to stop the tracking of a phase and then reload the phase at a later date without the need to track all the players that have been previously tracked. This is a very basic feature but all necessary files must be saved and loaded so the program works when the selected phase is played.

Outputs from the program

The main aim of the program is to find measurements that can be used to help describe the phase plays that are built in rugby. These will have to be displayed to the user in a way that will allow them to use it for analysis. The measurements will be displayed in 2 forms:

- **Graphically on Video** – Today analysis is done through the replaying of video as it is the easiest way to show people exactly what has happened as there is no need for them to imagine it. Therefore the measurements taken will need to be shown on the video to help the users understand the data.
- **Graphing data** – This will show the value of the data and the changes throughout the phase. This way the user can show the exact changes and it will be the only clear way for the user to see the values of the results.

Approach & Implementation

This section looks at the findings and the solutions found to find creditable statistics in the video analysis. It also outlines the program created to allow the inputs and findings to be shown in the program.

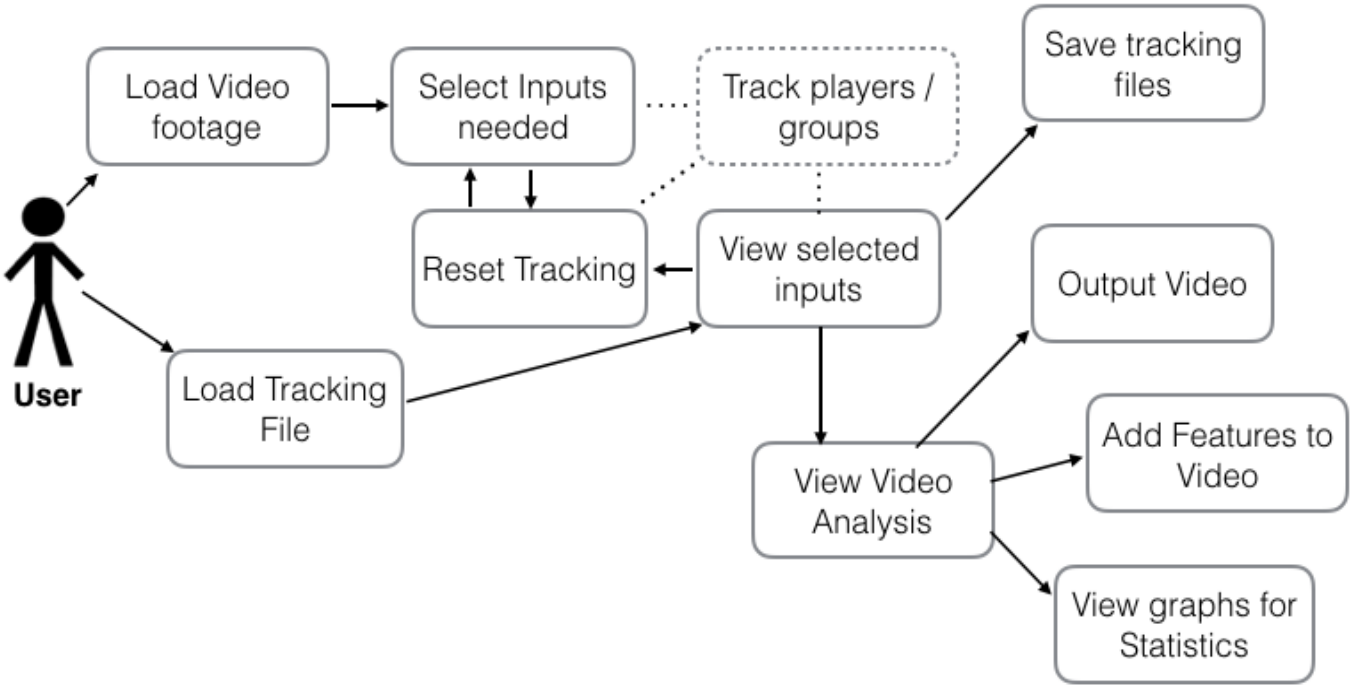
The program interest was in the features and statistics that could be gathered from the video footage. So the GUI was developed to support the showing of statistics. This meant that the GUI constantly changed during the research. The final outcome has now been finalised and is explained below.

System Overview

User Flow Chart

Below are the process that the user can select when opening the program. Certain events are blocked if other process do not occur as the process are needed before others can produce.

Figure 12 -User Flow Chart



Load Video Footage:

This allows the user to select and load the phase play that is to be analysed by the program.

Load Tracking File:

If tracking has already been implemented on a video clip then a saved Matlab file can be loaded with all relevant files needed for that clip. This saves time as clips do not have to be re-tracked to use again.

Select Inputs Needed:

Once a video clip is loaded then inputs are needed from the user. There are 2 main inputs needed from the user. They are the home team players, gainline and away team players. These are then passed to the tracking process.

Track Players / group:

This is not a process that the user can decided to run. It is critical to the program as this is where the statistics of the players and teams are created. More information will be given below.

View Selected inputs:

This is a part of the basic UI (select_Player_GUI). When the selection process and players are tracked the UI will change to show which players have been tracked and offer a new window to view the statistics produced. It also gives the option to run the tracking method again. This would then run the re-tracking method.

Reset Tracking:

Once the first run though of a video has been produced then. A feature that has been implemented checks the previous tracks to see if the selected players have already been tracked. This saves time as there is less processing. More information below.

Save tracking files:

Again when a run thorough has been produced the program will offer the chance to save the appropriate files. This option allows the user to save so phase plays do not have to be tracked again. More information below.

View Video Analysis:

This page can be selected when tracking has been implemented on the selected video clip. This page allows the user to view the graphs created and add features to the video and watch the play back.

Output Video:

This feature allows the user to output the video as they currently view it. For example, if features have been added then this will output with the video to an .avi format at a location of their choice.

Add Features to Video:

The best way to show the statistics is to see a drawing on top of the video as it shows it visually to the user. This function allows the user to select the graphics wanted and then a process is run to add the features to the video clip.

View Graphs for Statistics

This is a function that can be selected on the Play_Video UI. There are 3 graphs that can show all the statistics taken in the video. These can be selected and the relevant graphs are drawn.

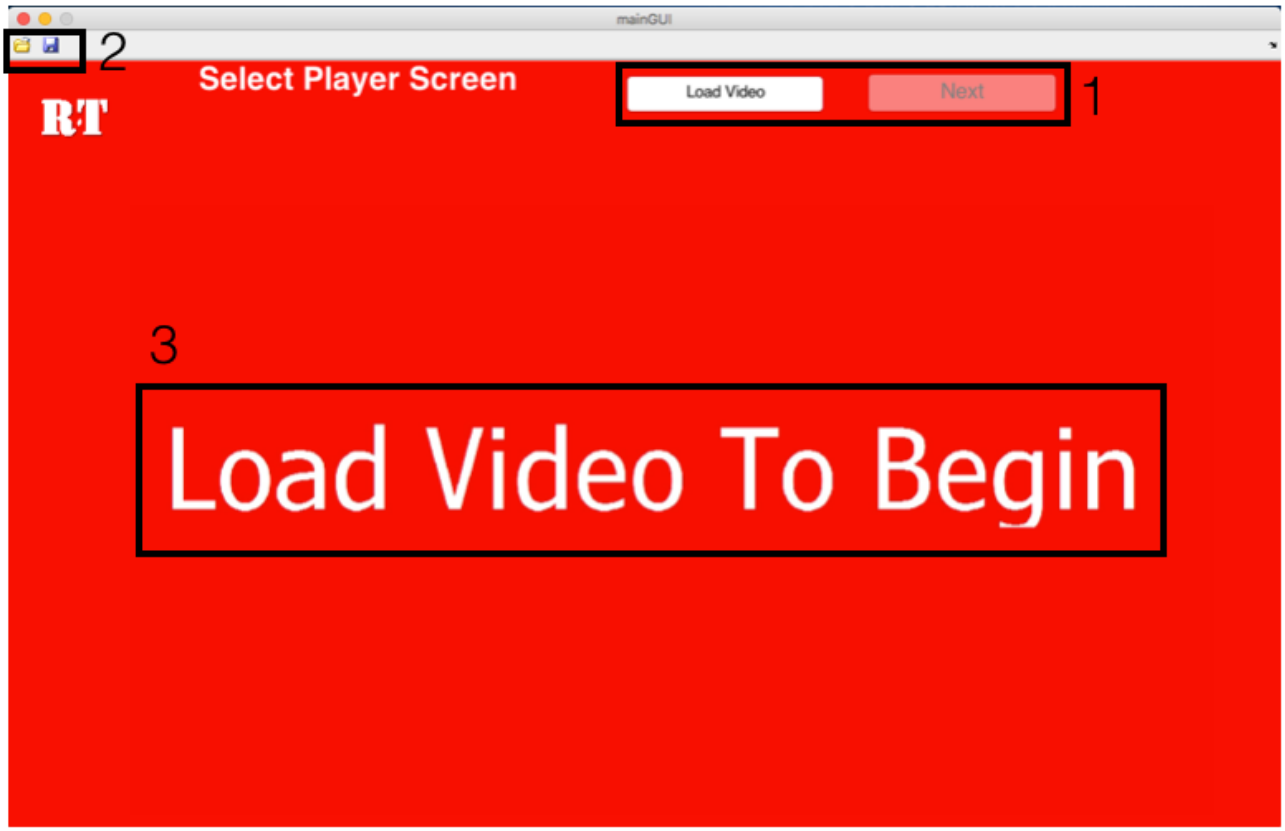
User Interfaces

As stated above due to the nature of the project the UI designs changed as the additional features were needed for the statistics to be gathered correctly. They have been finalised and an overview of both UI's are given below.

select_Player_GUI

This user interface is called to start the program. There are multiple states that this UI can be in as it tries to clearly highlight to the user what is the next step. Screenshots below show each state and the relevant features.

Figure 13 - select_Player_GUI State 1



State 1 - No Video or Tracking Files created

Box 1 - User Interactive buttons

The only button available at this time is the Load Video.

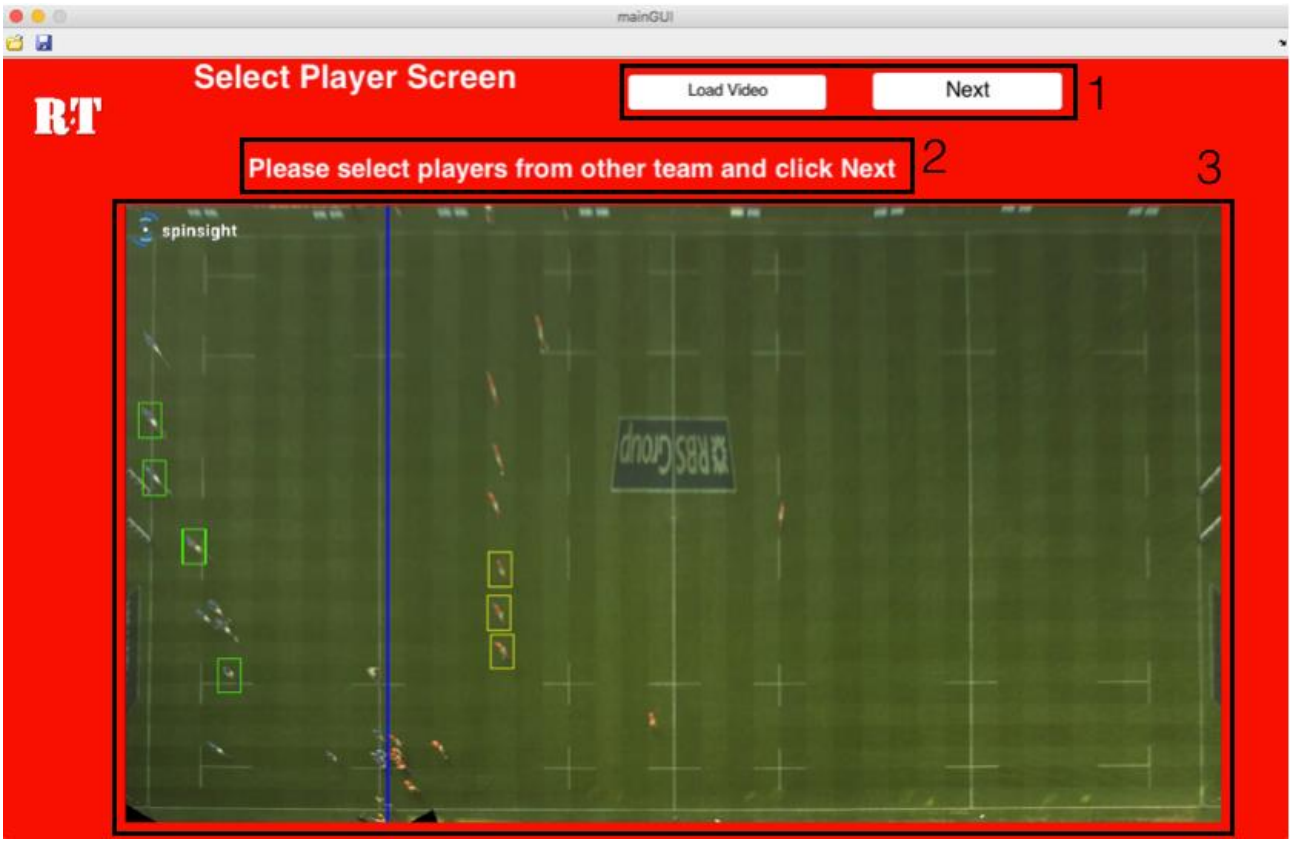
Box 2 - Program Menu

The option to save and load tracking files are available in this corner. Load button can be used here to load a previous phase that has been tracked and save before. More detail given below.

Box 3 - View Window

This is a simple frame where a .jpg image is loaded (‘Load Video.jpg’). To help interact the user as to the next step.

Figure 14 - select_Player_GUI State 2



State 2 - Video file loaded

Box 1 - User Interactive buttons

Now that a video has been loaded the next button has become available as the user is now required to follow steps and click *Next* when complete.

Box 2 - Progress and command *Box*

This box highlights the user next step when needed and also highlight the state of the program currently. It tells the user the steps needed and after requires the user to click *Next*.

Box 3 - View Window

Once a video has been loaded the first frame is captured and displayed to the user in the View Window. While the program is in a certain state for retrieving user selections the window will become interactive. When players are selected the boxes will highlight the areas clicked and a line will be shown when the gainline has been selected. As shown in the figure above.

Figure 15 - select_Player_GUI State 3

State 3 - Tracking has been complete

Box 1 - Program Menu

The save button can now be used to save the reverent files and video for the tracking. More in section below.

Box 2 - User Interactive buttons

The next button has now been removed as the user input is not needed. The load video is still available if a new tracking play is to be used.

Box 3 - New User interactive buttons

These buttons should only show if the tracking method has been completed. These button can be made visible from running the tracking method or loading a previous play that has been tracked. The re-tracking will set the UI back to state 1 and run the retracting method (Explained below).

The View Video button opens the UI “Play_Video” and holds the current UI until “Play_Video” is closed.

Box 4 - Progress and command *Box*

In this state it should only show the message “Player Tracking - Complete”. If it does not show this then an error has occurred.

Box 5 - View Window

In this state the player tracking should be complete. The window should now be disabled and only show the selections made in the player tracking process. As shown in the figure above.

Play_Video

Figure 16 - Play_Video UI

Box 1 - Progress Box

This highlights the progress of certain functions when they are called. This is due to the time taken.

Box 2 - User Interactive buttons

These buttons give the user the ability to run the additional functions that have been implemented.

Close Button - Just closes the UI.

Show Graphics button - This along with the features box allows the user to select graphics to be added to the video.

Output Video - This runs the Output Video function as explained in Features Implemented Section.

Box 3 - Video control panel

This has full control over the video that is being displayed in the frame below. There is the pause and play functions available. While the bar gives the user the option to select the frame position.

Box 4 - View Window

This shows the video of the phase play being tracked. If additional graphics have been added to the window will also display the video with these graphics as shown in the figure above.

Box 5 - Features Box

This is where the user can select what graphics they want on the video. After selecting the graphics wanted the user must click the Show Graphics button to show selected graphics on the video.

Box 6 - Graphs for Statistics

Here the graphs for all the statistics can be selected and viewed in each graph. When a certain statistic is selected the relevant data is retrieved and the graph is drawn. All graphs X axis show the video over frames. While the Y axis will change depending on the measurement being made. The statistics available are available in statistics measured section

Features Implemented

Although this project focuses on the statistics that can be retrieved from the video footage. It has also had additional functions implemented to help with the processing speeds and the future development of the project. These additional features have been listed and explained below.

Graphs for Statistics

It may be a simple but the graphing of statistics is another visual cue that can help with the analysis of a phase. The graphing of results is the easiest way to show the statistics that have been captured.

The graphs gives the user an indication of the changes that occur during the move over a given time. For example if the backline encroaches on the gainline in a move then this is clearly shown in the graph. If this shouldn't happen then the players can be shown clearly that this is happening from the graphing of measurable data.

Output Video

Again a very basic feature but one that helps the analysis team with their job. The ability to output the video with selected features means that the user can take the video and its features that have been added and use it on another system.

This feature is useful as team talks can generally be done on any system and the ability to still use the programs results makes it a system that they are more likely to use.

Load & Save files

This is a complex feature as the program must make sure that is saves the correct files and data to be able to reproduce the tracked phase and its statistics. Again this feature will make the program more user friendly as the processing of the data will not have to be done twice on a phase of play.

Below is an outline of the files used to produce the save file:

BWblob.avi - To help with the KLT feature tracking the original video footage is edited to just produce a blob for every player on the pitch. (Example shown in appendix)

BGRemoved.avi - similar to the video above the original video footage is edited to remove background. This is the step before “BWblob.avi” is produced. (Example shown in appendix)

OtherPlayers - This stores all the players data that have been tracked in the phase play. All players in this file may not be selected but are available when re-tracking is run. (More explanation in re-tracking section below.)

group - This stores the current group statistics of the home team selected. The values that are stored in this file can be seen in the Tracking and Calculating Phase Play section below.

oppGroup - This is exactly the same as the group profile above but stores the group statistics of the opposition team.

player - Stores the players individual statistics that have been selected for the home team in the phase play. Data can be entered straight from OtherPlayers table as the same information is stored.

oppPlayer - Again exactly the same as player profile but all players data in this file is that of selected players for the opposition team.

PlayerLoc – Stores the locations of the selected players in the phase play for both teams.

Gainline – Stores the x axis location of the gainline in the phase play.

Film - This is the original video that has been loaded into analysis the phase play. This is stored as a matlab Video Class so all values for the video are contained. (e.g. number of frames, video playback speed)

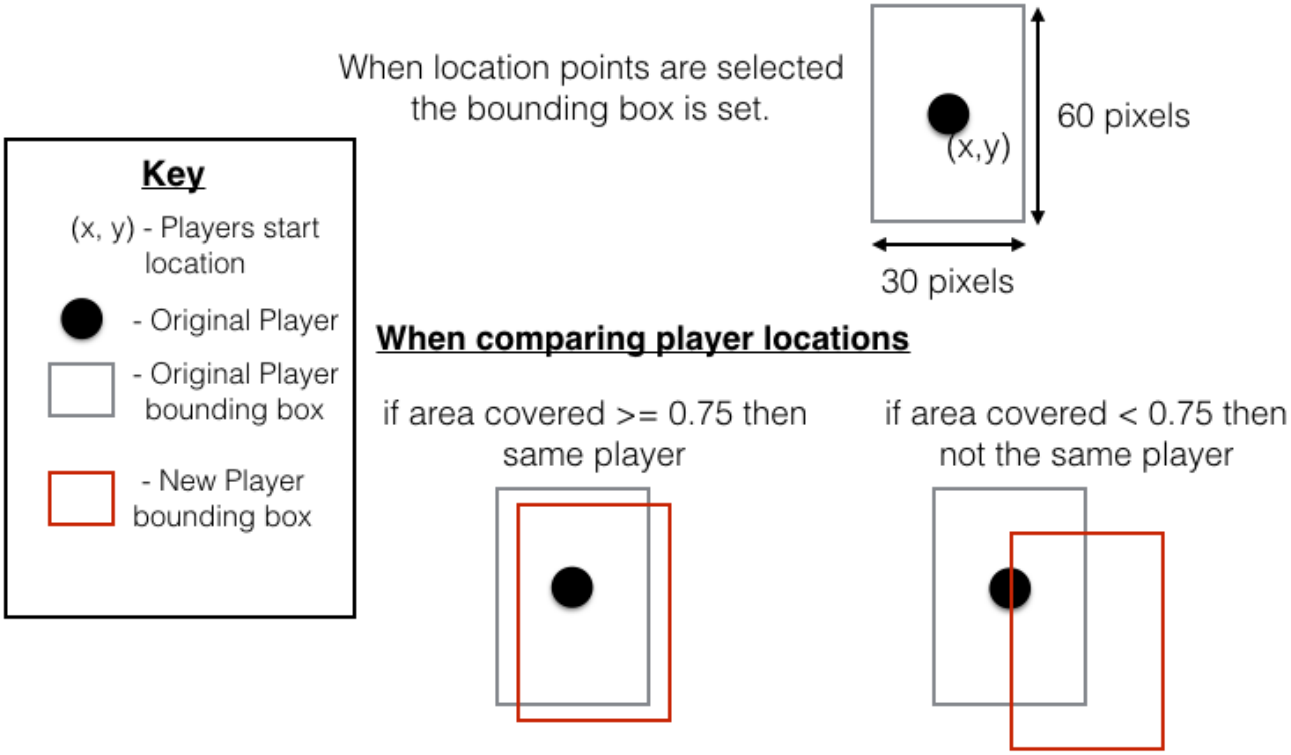
All these files are stored within the default location (local program folder for the program). When saving the program files they are then loaded from the default location and saved to a selected location under one file name as a MatLab file. For loading of a previous tracked phase the save function is revered. Where the files from the selected MatLab file overwrite the files in the default location. Before system variables are changed to complete the load.

Comparing 2 Players

Comparing 2 players is key to the re-tracking system set in this project. Outlined here is the method behind comparing players.

Players are compared by their start positions in frame 1 and the bounding box coverage by the new player selected. This is needed as the user may not click the exact place as the previous tracking. Therefore a comparison must be made. The percentage of area coverage needed for a match could be altered if an adjustment is needed. But this will reduce the accuracy in the program.

Figure 17- Comparing 2 Players



Code

Below is the code used to compare 2 players.

```
% Loop for all new players
for countA = 1:length(oldPlayerLoc)
  matchFound = 0;
  % Set new players bounding box
  A = [oldPlayerLoc(countA,1) oldPlayerLoc(countA,2) 30 60];
  for countB = 1:length(OtherPlayers)  % Loop players in OtherPlayer list
      B = [round(OtherPlayers(countB).X_start …
            round(OtherPlayers(countB).Y_start)  30 60];
      overlap = rectint(A,B)/(30*60); % Percentage of overlap
      if overlap >= 0.75
```

```
            % Percentage has been meet flag as a match
            matchFound = 1;
            break;
        end
    end
    if matchFound == 0
        % If no match found add to Other Players
        OtherPlayers = [OtherPlayers, player(countA)];
    end
end
```

The key line in this code is the line below. This uses a Matlab in built function rectint to return the area of intersection. Then it is divided by the rectangle size to produce the percentage of coverage.

```
overlap = rectint(A,B)/(30*60);
```

rectint - Rectangle intersection area

"area = rectint(A,B) returns the area of intersection of the rectangles specified by position vectors A and B." - Matlab
Source: http://uk.mathworks.com/help/matlab/ref/rectint.html

Convert pixels to meters

In this project tracking is done in pixels and is converted to meters using a variable called pixel2mtere. The value set takes into account the scale and the focal length of the camera. The camera does not move position and has full coverage of the pitch as seen below. This means that the one conversion value is needed for the statistics.

As the rugby pitch has defined dimensions this means that by running the program for selected points it gave values that allowed a conversion value to be defined. Below are some of the calculations run to produce the end results.

$$\frac{5\ m}{90\ pixels} = 0.05555\dot{} \qquad \frac{22\ m}{400\ pixels} = 0.055$$

$$\frac{10\ m}{182\ pixels} = 0.0549$$

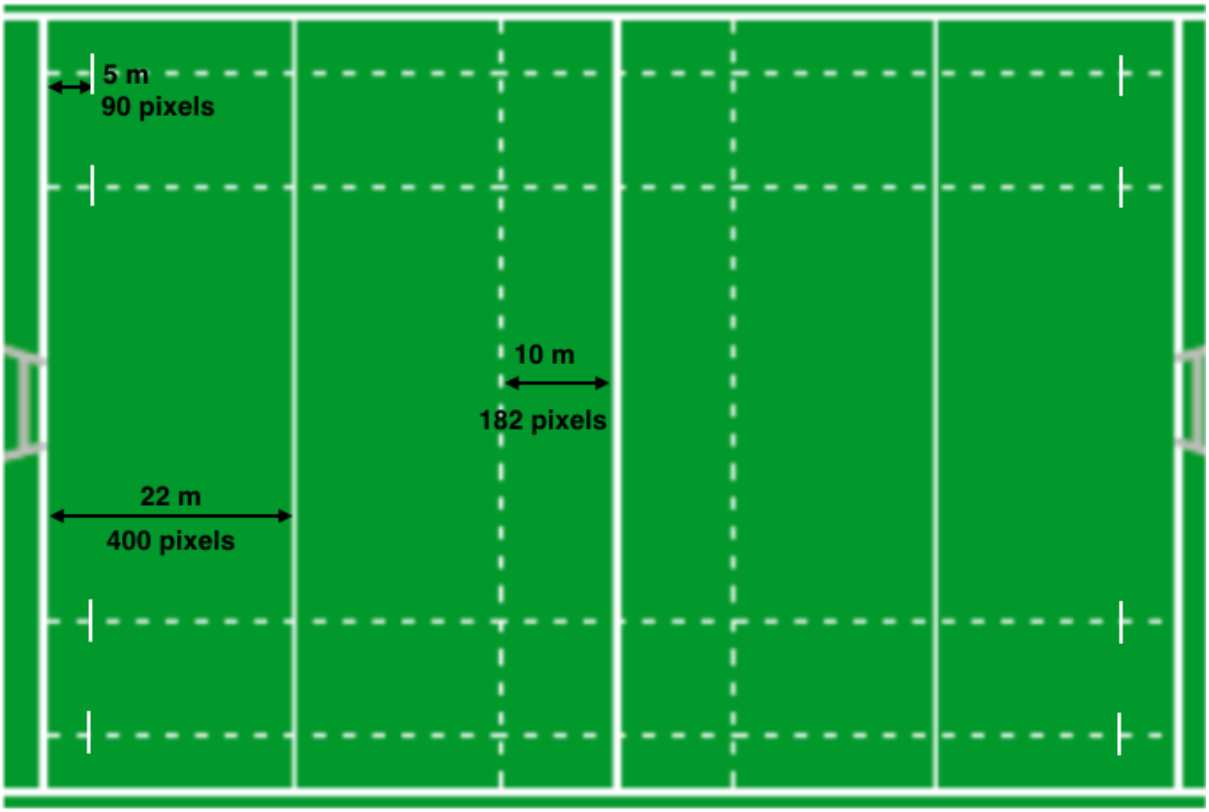


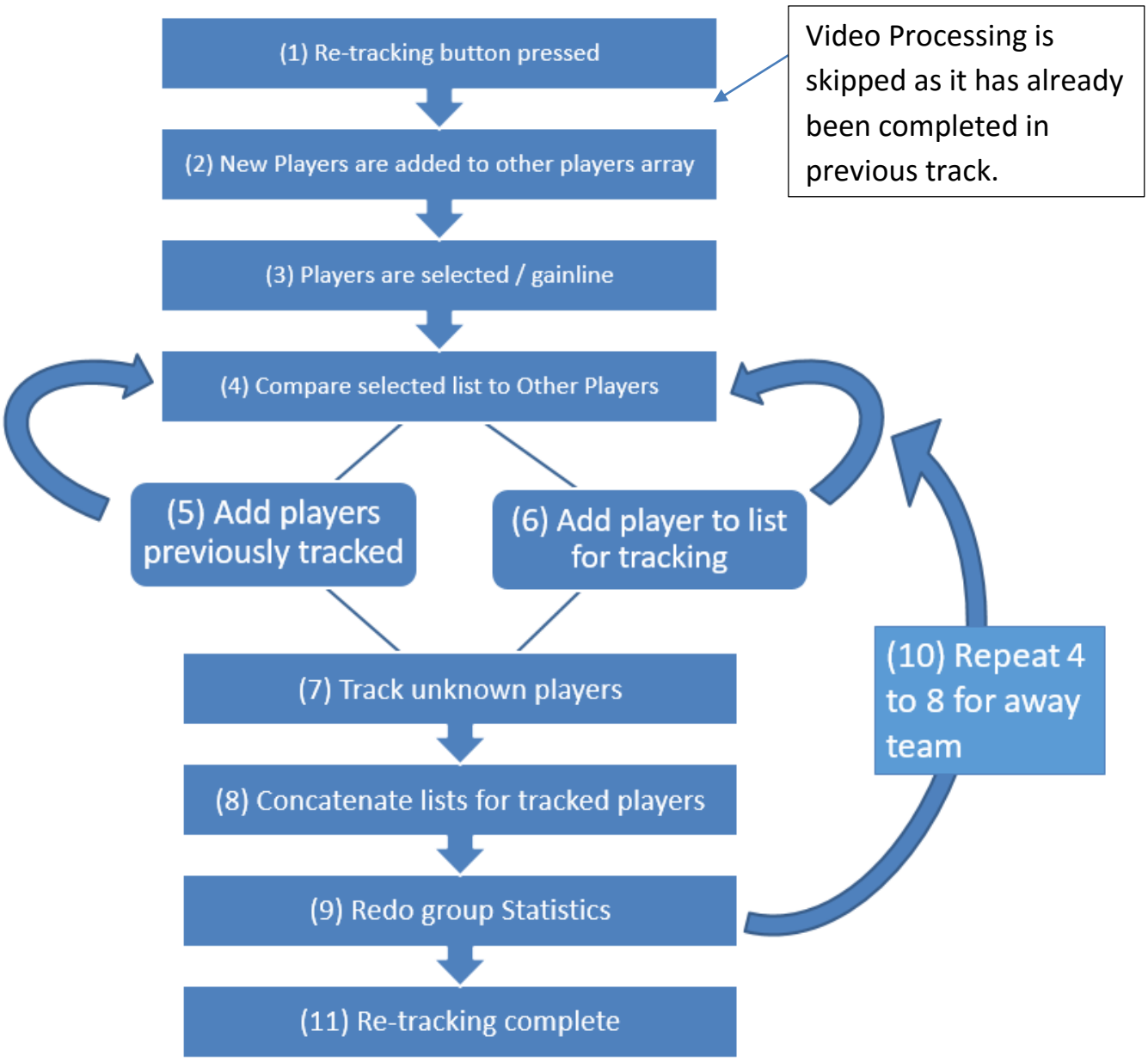
Figure 18- Pixel to meters

From multiple tests and calculations the value was set to $\frac{10}{184} = 0.0543$

Re-Tracking of Players

This function is very useful in the development of the program. This function reduces the amount of work the program has to do as it searches the stored players for tracks that have been completed before. Below is the function has been explained.

Figure 19- Re-tracking Process



(1) Button is pressed

The user initiates the process by clicking the Re-tracking button. Here a flag has been changed to state that a previous run through has taken place.

(2) **Players from current tracking are added to OtherPlayers**

This is the first step the program takes in the re-tracking process. Here players selected are compared to the data stored in OtherPlayer array. If no match is found players are added to the OtherPlayer array. As shown above in section for comparing 2 players.

(3) **New players from both teams and gainline are selected**

This is no different to the regular selection of players and gainline. The difference is due to the flag set at point 1.

(4) **Compare new selected players with OtherPlayers array**

Once players and gainline has been selected. Then the tracking function is called. The difference is that the run_through flag set at point (1) causes the tracking to compare the new players selected with previous selected players.

(5) **Add players found to new Player list if found**

If a match is found then the players data is copied straight across to a temporary array called newPlayer.

(6) **Add player to list for tracking.**

If no match is found then the location of the player is set for tracking. The locations are set into x_temp and y_temp arrays.

(7) **Track unknown players**

Here unknown players are tracked using the normal tracking method as stated in the tracking and calculating phase play.

(8) **Concatenate lists for tracked players**

Once unknown players have been tracked the 2 lists will be concatenated and newPlayer list will be changed back to the standard player list for use.

(9) **Redo group statistics**

As the selection of players may have changed then the group statistics will have to be recalculated. This is completed exactly the same as a normal tracking method.

(10) **Repeat steps 4 to 8 for away team**

For the same reasons the away team will run through the same procedure.

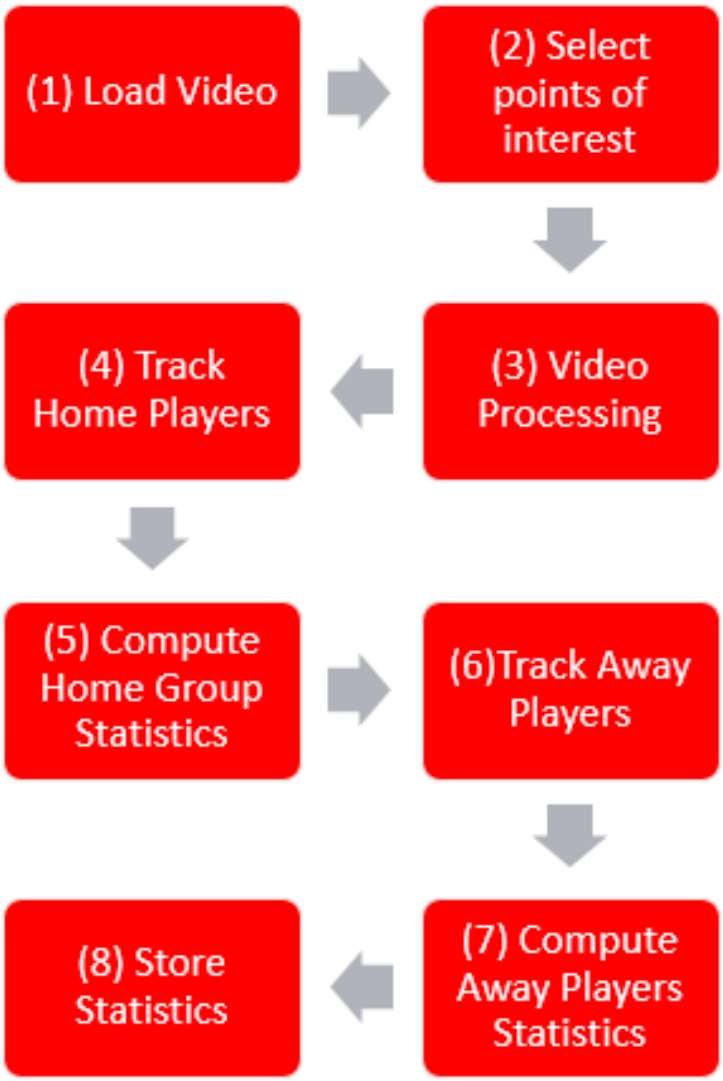
(11) **Re-tracking complete**

All procedures should now be complete and the select_Player_GUI should be in state 3 as explained in the User Interface section.

Tracking and Calculating Phase Play

This section outlines the tracking system that has been implemented to track phase play that is selected. The tracking method uses the KLT tracking system to track the players over the video and then these are passed to the group function for analysis.

Figure 20 - Tracking system



(1) Load Video

The video must first be loaded so the program can initialise the tracking method. The load function should enable the select_Player_GUI to enter state 1 as mentioned in the user interface section.

Here the video is loaded in using the VideoReader (http://uk.mathworks.com/help/matlab/ref/videoreader.html) function built

into Matlab. This creates a Video class where the video footage and frame 1 is extracted. The footage and frame are then saved for future use.

Code

The variable OriginalVid is the full file location of the selected clip.

```
% Reads Original video clip
testvid = VideoReader(OriginalVid);

% Get video properties
set(handles.txtInfo, 'String', 'Reading Video ...');
drawnow;
Film = VideoClass(read(testvid, [1 Inf]),testvid.NumberOfFrames,...
   testvid.FrameRate,testvid.Height,testvid.Width);% Film is a
MovieObject
Vid = Film.video; % Just get video

%Save files needed later
frame = Vid(:,:,:,1);
imwrite(frame,'frame00.jpg');
save('-v7.3','OriginalFrames','Film')
```

(2) Select Points of interest

As stated in the user input section above the user must enter additional information for the tracking system to be successful. The program will ask for 3 inputs home team players, gainline and away team players. From UI section the select_Player_GUI will be in state 2.

Here 3 arrays are created hold the information. These are passed on to the next stage for tracking. These are also the same data structures that are compared when re-tracking. The 3 arrays are below.

PlayerLoc - holds all the x and y co-ordinates of selected players for the home team.

Example

X-Coordinate	Y-Coordinate
800	400

OtherTeamLoc - holds all the x and y co-ordinates of selected players for the away team. Same as example above.

gainline - Is a single value that stores the x-coordinate of the selected position of the gainline

(3) Video Preprocessing

Here the video is edited to remove the pitch and turn the players into white dots. This makes the feature tracking easier as there is less noise within the photo. First an average pitch image is generated to help remove the background.

Code

```
avIm = mean(MovieObj.video, 4);
avIm = uint8(avIm);
```

Secondly, each frame is then converted and 2 videos are produced. BWBlob.avi and BGRemoved.avi. This is done through the first removing the background from the video to produce BGRemoved.avi. Then thresholding is run on the new frame to give the blob effect to produce BWblob.avi. Frame captures can be seen on the appendix.

These files are then saved for use in the tracking method.

Code

This is the relevant code used in the frame editing to produce the new videos.

```
mov(i).cdata = (video(:,:,:,i) - avIm); % capture one frame and covert to
                                          double precision

writeVideo(writerObj1,mov(i).cdata);      % write BG removed to file

mov(i).cdata = im2bw(mov(i).cdata, 0.1);  % thresholding
mov(i).cdata = imerode(mov(i).cdata,se1); % erode image
mov(i).cdata = imdilate(mov(i).cdata,se2); % dilate image
mov(i).cdata = im2double(mov(i).cdata);   % convert to double
                                            precision

writeVideo(writerObj2,mov(i).cdata);      % write BW blob video to file
```

(4) Track Home Players

This is where the system now uses the KLT feature tracking system using the BWblob.avi video produced in the process above. The selected home team players blobs are tracked through the frames to produce a frame by frame coordinate for each player. Selected players are located using PlayerLoc array from process 2.

Here the individual player statistics (more in the Statistics Gathered section below for player statistics) are also calculated and are then placed in the data structure. The data structure is called 'player_profile.mat' and is structured to store all relevant information. This file is then saved to the workspace for future use. Below is the layout of the file each row in the table is a selected player.

Name of Column	Type	Description
boundingbox	Double array	Stores the bounding box used by the KLT tracking system
X_start	Double	Stores the X coordinate of the players position in frame 1.
X1	Double Array	Sores the X coordinates for the player in each frame. Array length depends on the number of frames.
Y1	Double Array	Sores the Y coordinates for the player in each frame. Array length depends on the number of frames.
Y_start	Double	Stores the Y coordinate of the players position in frame 1.
X2	Double Array	Used to store the X coordinates when tracking players.
Y2	Double Array	Used to store the Y coordinates when tracking players.
x	Double Array	Stores the distance travelled by the player in the x direction from the start position. Array length depends on the number of frames.
y	Double Array	Stores the distance travelled by the player in the Y direction from the start position. Array length depends on the number of frames.
Dis	Double Array	Stores the distance travelled by the player in both X and Y direction from the start position. Array length depends on the number of frames.

Name of Column	Type	Description
v_x	Double Array	Stores the velocity of the player in the x direction from the start position. Array length depends on the number of frames.
v_y	Double Array	Stores the velocity of the player in the Y direction from the start position. Array length depends on the number of frames.
Vel	Double Array	Stores the velocity of the player in both X and Y direction from the start position. Array length depends on the number of frames.
a_x	Double Array	Stores the acceleration of the player in the x direction from the start position. Array length depends on the number of frames.
a_y	Double Array	Stores the acceleration of the player in the Y direction from the start position. Array length depends on the number of frames.
Acc	Double Array	Stores the acceleration of the player in both X and Y direction from the start position. Array length depends on the number of frames.
theta	Double Array	Stores the direction of travel of the player. Array length depends on the number of frames.

Table 1 – player_profile.mat layout

(5) Compute Home Team group statistics

Once all player statistics and tracking is complete the group statistics are calculated using the 'player_profile.mat' data structure. This file is loaded and the group statistics are then calculated. (Calculations can be found in section Statistics Gathered section of the report.)

Once calculate the results are again saved to a file for future use. The file name is 'group_profile' and is stored in the workspace. Below is the layout of the file. Each row is a frame within the video.

Name of Column	Type	Description
Back_3	Double	This stores the area coverage of the back 3
Back_3_Ang	Double	Stores the angle of the back 3.
Depth	Double	Stores the depth of the players selected
PlayerDis	Double Array	This array stores the distance between each player within the back line.

Name of Column	Type	Description
x_group	Double	Stores the average group displacement from the origin (first frame) in the Y direction. Not currently used in the statistics.
y_group	Double	Stores the average group displacement from the origin (first frame) in the Y direction. Not currently used in the statistics.
DisFromGainline	Double	Stores the group distance from the gainline.

Table 2- group_profile.mat layout

(6) Track Away Team Players

This is exactly the same process when tracking the home team players. The differences here is that the name of the file outputted is 'OtherTeam_player_profile'. Secondly the selected players are located using OtherTeamLoc array.

(7) Compute Away Team group statistics

Again similar to the process above the group statistics are computed the same but using the file 'OtherTeam_player_profile.mat'.

(8) Completing the tracking

After the away team has been processed the tracking is complete. Here variables are set to control the select_Player_GUI UI and put it into state 3 as above in the UI section of the document.

All files for the statistics should now be stored in the program folder and the save feature as stated above is now available to copy the files into one moveable file.

Statistics Measured

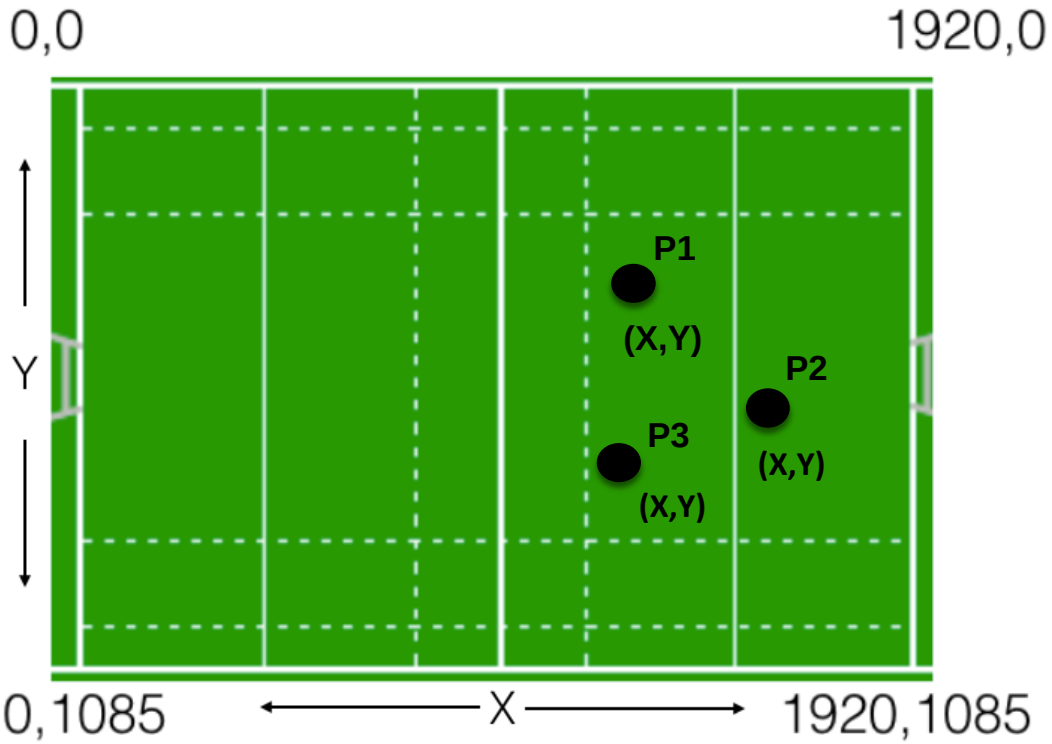
Player Statistics

Below are the statistics that are measured for each tracked player within the phase. This section explains the reasoning and how it is measured.

Players Coordinates

This is vital to the whole project and is the bases for every statistic calculation that is made. As shown in the tracking section the coordinate system gives the location on the pitch and from this we can start to build a picture of the team and the players positioning throughout the phase.

Figure 21 - Player Coordinates



Displacement / Velocity / Acceleration

These individual statistics are useful to the analysis team when looking into the movement of the team during a play. For example if someone was slow in their acceleration off the line then this could be an issue as gaps will open up in the defence.

These values combined can give a good comparison between players and are values that are used by analysis teams when analysing a player's performance within a team situation.

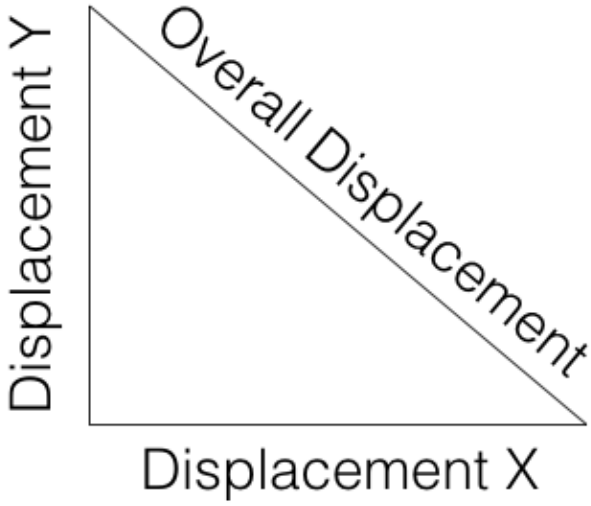
How are they calculated?

First of all the statistics are measured in both the X direction and Y direction separately. Before being computed to create a value in both directions using Pythagoras theorem.

$$c^2 = a^2 + b^2$$

$$\text{Overall Displacement} = \sqrt{\text{Displacement } Y^2 + \text{ Displacement } X^2}$$

Figure 22- Pythagoras Triangle (Displacement)



Calculating Displacement

This is done by comparing the new frame position with the start location to produce a displacement.

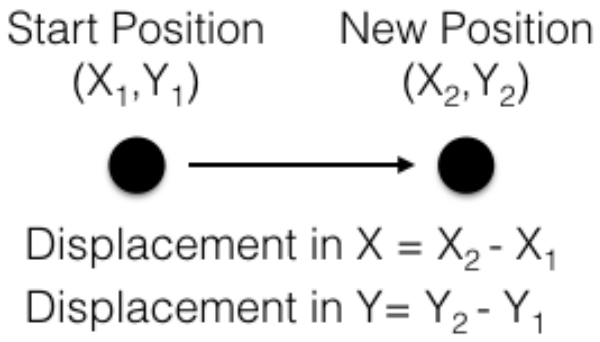


Figure 23 - Displacement Calculations

Code

```
% Displacement
player(i).x = smooth(player(i).X1 - player(i).X_start)*pixel2metre;
player(i).y = smooth(player(i).Y1 - player(i).Y_start)*pixel2metre;
player(i).Dis = smooth(sqrt((player(i).x.^2)+(player(i).y.^2)));
```

Graphs for Displacement

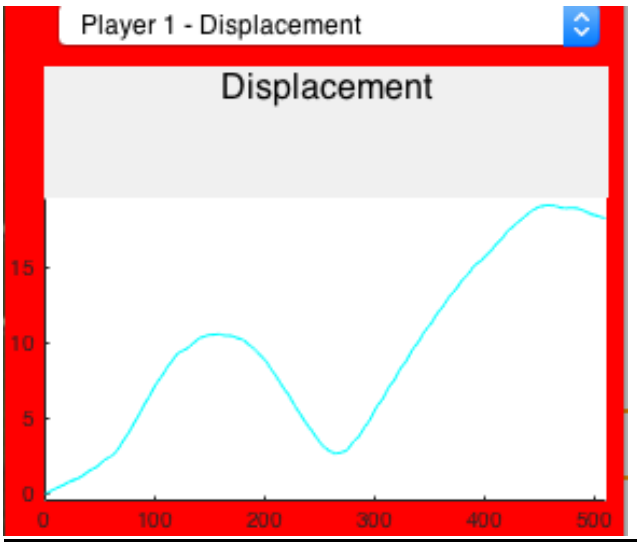


Figure 24 - Displacement graph

Calculating Velocity

Once the displacement is calculated then velocity and be calculated by the gradient of displacement. The velocity is calculated through X and Y before an overall value is created.

$$\text{Velocity} = \frac{\blacktriangle \text{Position}}{\text{time}} = \frac{\text{Displacement}}{\text{time}}$$

Code

```
% Velocity
player(i).v_x = gradient(player(i).x);
player(i).v_y = gradient(player(i).y);
player(i).Vel = sqrt((player(i).v_x.^2)+(player(i).v_y.^2));
```

Graphs for Velocity

The Y axis would be in meters per second. All players have the acceleration graphs available.

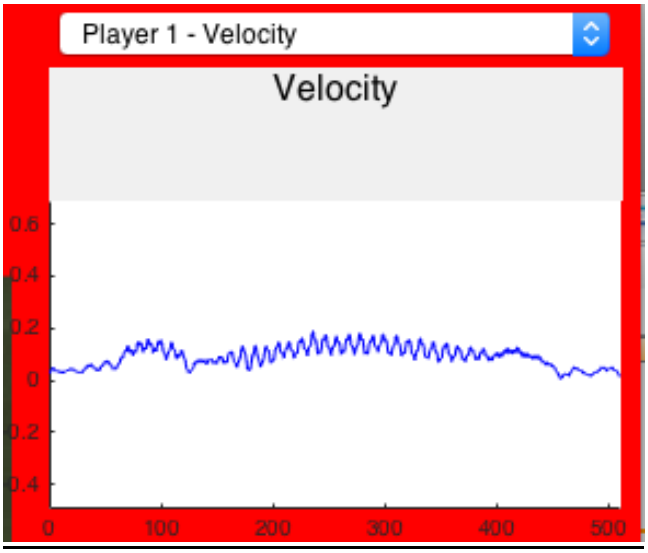


Figure 25 - Velocity graph

Calculating Acceleration

Acceleration is calculated in the same way where the gradient of the velocity is used for the calculation. Again calculated through X and Y axis before the overall value is created.

Graphs for Acceleration

The Y axis would be in meters per second squared. All players have the acceleration graphs available.

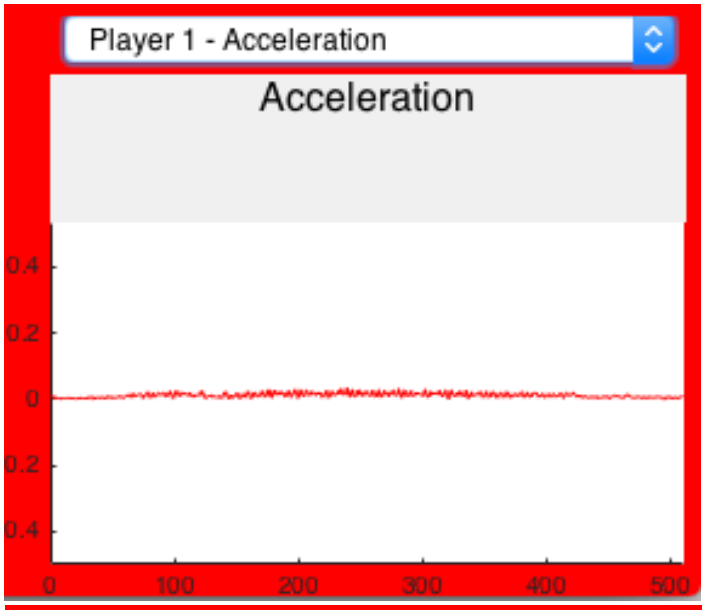


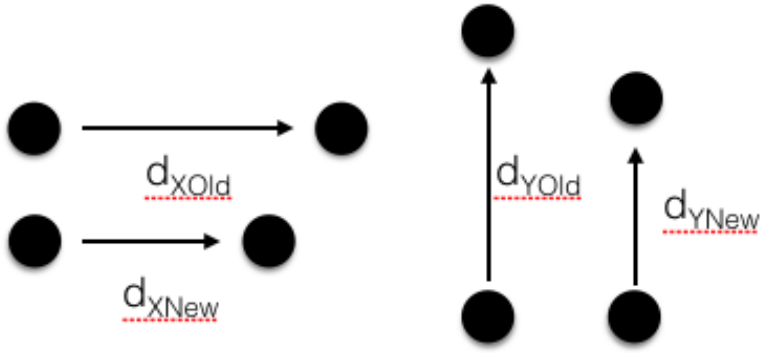
Figure 26 - Acceleration graph

Direction of travel

Although not currently used in any of the group statistics the direction of travel could be a very useful feature as the statistics measured become more complex. For example when a drift defence occurs all players need to push at the same rate and at the same time. (More information in the appendix)

This statistic would help the system to analysis a drift defence accurately and other such plays like it.

Figure 27 - Direction of travel



$$dx = d_{XNew} - d_{XOld}$$

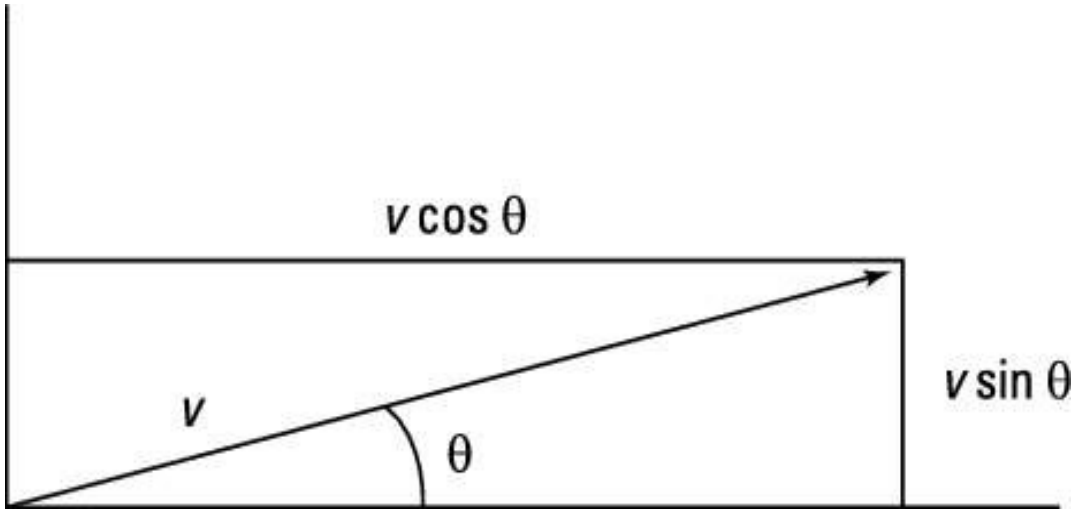
$$dy = d_{XOld} - d_{XNew}$$

if dx < 0
  direction = $\tan^{-1}(dy/dx) + 180$
else
  direction = $\tan^{-1}(dy/dx)$

How is it calculated?

The displacement is taken from the previous frame and the current frame and the difference is used in the final computation. If the difference in x is negative then the person has changed direction and therefore 180 degrees is added to the x direction.

Figure 28 - Vectors, Magnitude and Direction



Source:
http://www.dummies.com/how-to/content/how-to-find-a-vectors-magnitude-and-direction.html

Code

```
% Direction of Travel
x_old = player(i).x(1);
y_old = player(i).y(1);
for j = 2:length(player(i).x)
    dx = player(i).x(j) - x_old;
    dy = y_old - player(i).y(j);
    % Set current displacement to old for next calculation
    x_old = player(i).x(j);
    y_old = player(i).y(j);
    if (dx<0)
        % if direction is backwards then flip theta
        player(i).theta(j) = atand(dy/dx)+180;
    else
       player(i).theta(j) = atand(dy/dx);
    end
end
player(i).theta = smooth(player(i).theta);
```

Issues

At the moment the issue with this is that the direction reading is very sensitive. A small change in the next pixel location and the payer direction is heavily affected. In the future this sensitivity will have to be addressed so a comparison between players can be made easier.

There are 2 ways that this could be addressed. The first way is to take an average of the direction over a number of frames. This would give a smoother direction change. Secondly, the direction could be computed to the neatest compass point (e.g. N,E,S,W,NE,SE,SW,NW). This would give violent changes but would give a better comparison between players.

Showing Direction

At the moment there is no graphs to show the direction of travel for a player. With more research a new way would have to be designed to best show this statistic.

It does have a graphic as shown in the frame capture below.

Figure 29 - Direction of player frame

Group Statistics

This is where the large amount of the research has taken place. The group statistics is where analysis teams will see the biggest benefit from a system like this. The research broke plays down into key components and looked at the components that could be measured by the system using the statistics that have been gathered above.

The research proved that there were measurements that were common in phase plays and that plays are defined by a number of these measurements combined. Below are the group statistics that were extracted by the research and each measurement has been explained in the detail below.

During the research and development of the statistics it was noticed that the players needed to be ordered so that the group statistics could be measured accurately. Below is the code used to sort the selected players into a sorted list for the group analysis. The players are organised into a list where the player with the minimum Y value is first to the player with the largest Y value last.

Code

```
[sortedY, SortIndex] = sort(Y);
% Sort X into same order as Y. X values must not change
 sortedX = X(SortIndex);
```

(Angle) Depth of Line

What does it show?

The depth of a backline can be used for most phase plays in both attack and defence. In an attacking line the depth of the line is looking to be deep so it has time to spread the ball wide and outside of the defence line. Where as in defence depending on the play the depth would change but most plays in defence want a flat line to close the attack quickly.

Maths

Here the angle is measured between a line running parallel to the x axis and the average line set by the back line. As shown in the figure below. The angle of the back line is measured from 0 to 90 degrees. If the line being measured is

passed 90 then the angle is swapped to the opposite direction. Future implementations could consider a bigger angle if the direction of the line is also considered in the calculations.

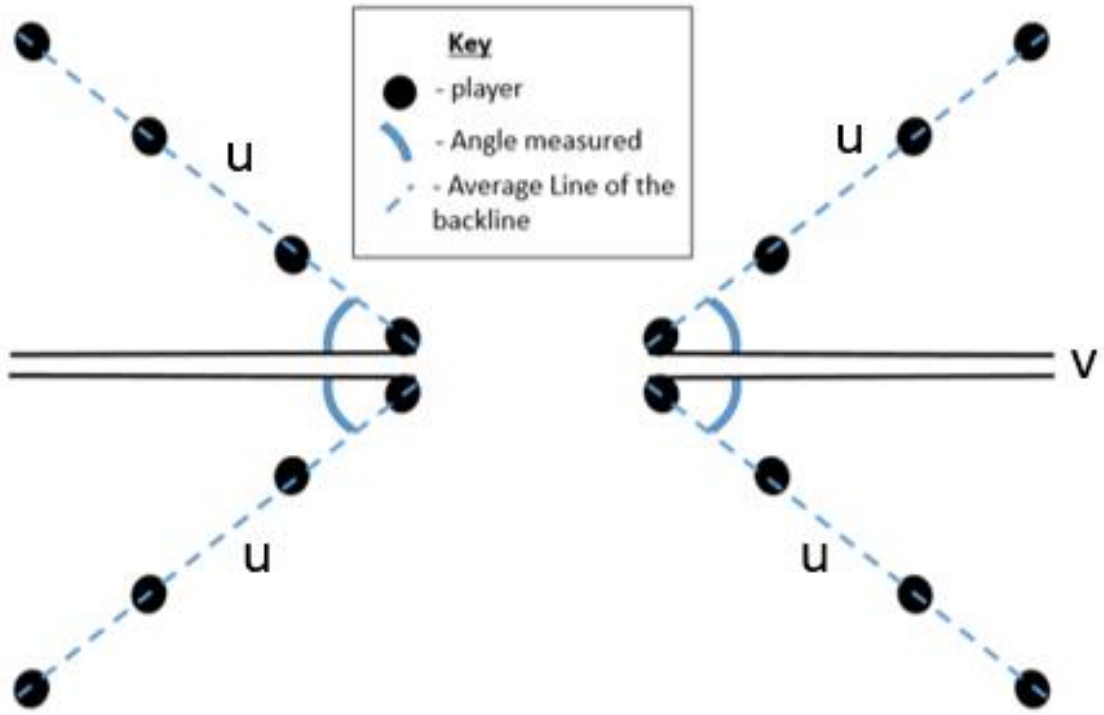


Figure 30- Angle of back line

$$\emptyset = \cos^{-1}\left(\frac{\vec{u}.\vec{v}}{[\![\vec{u}]\!][\![\vec{v}]\!]}\right)$$

Code

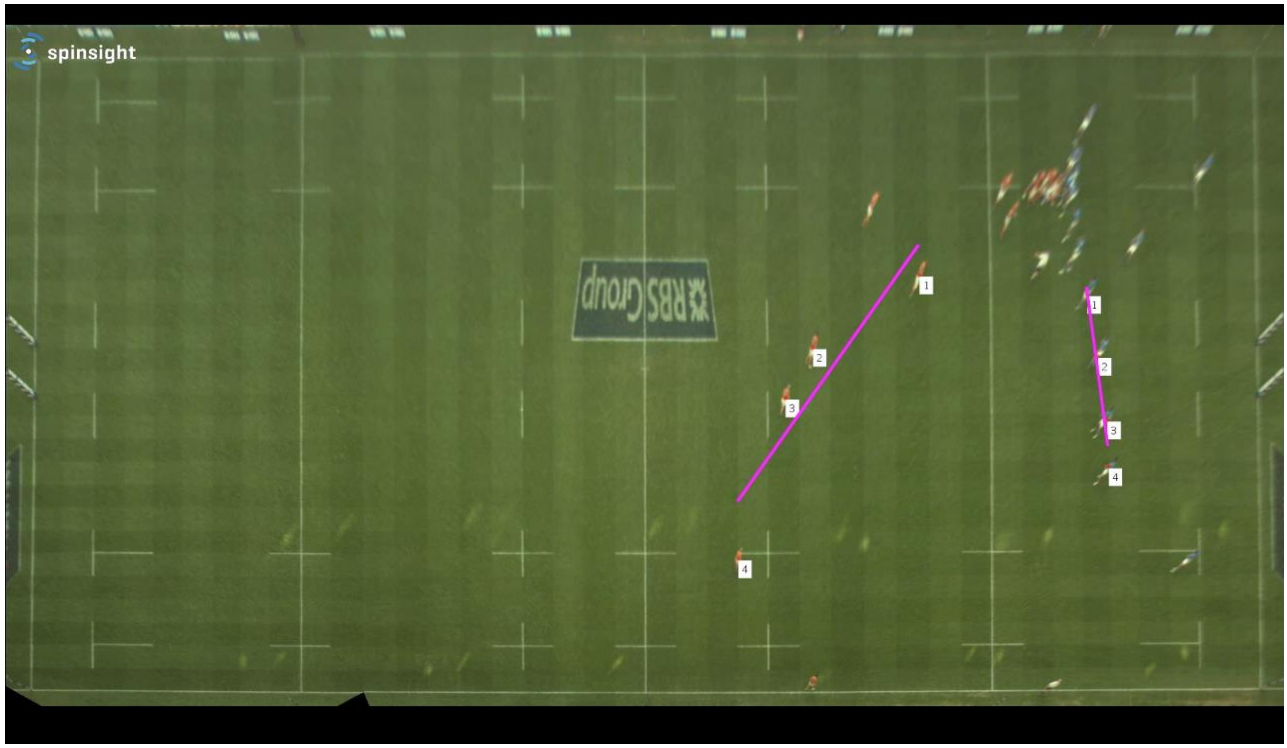
```
% Depth of Line
z = z + 1;
p = polyfit(sortedX,sortedY,1);
line = polyval(p,sortedX);
% Get angle
x1 = sortedX(1);
y1 = line(1);
x2 = sortedX(end);
y2 = line(end);

angle = abs(round(radtodeg(atan((y2-y1)/(x2-x1)))));
group(i).Depth = angle;
```

Features on Video

To help the user with explaining the statistics to a team. A graphic has been developed to show what is being measured. Below is a frame capture of a video with the graphic placed on it.

The pink line shows the average position and depth of the line. Above the line on the left is showing the welsh team attacking. As stated above the attacking line looks to hold its depth and during the video as players move the angle changes. It also shows the defensive side holding flat.

Figure 31 - Depth of line frame



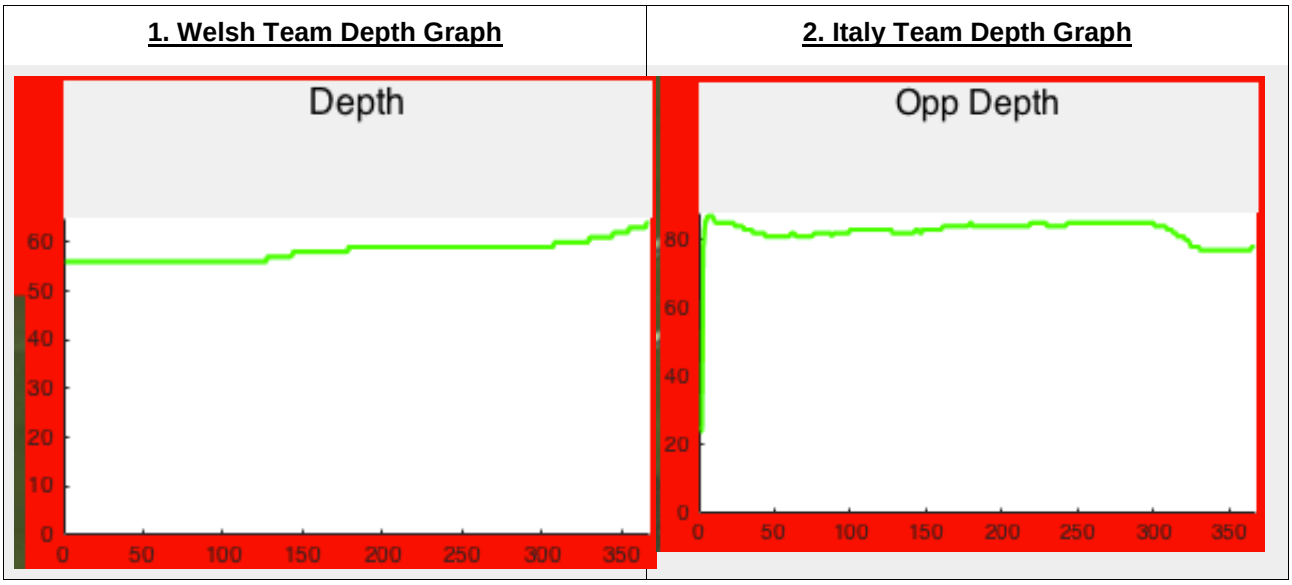
Graphs

Below are the graphs that allow the user to see the statistics that have been gathered. These are linked to the features shown above. The graph allows the user to see the changes that occur during the phase.

The graphs Y axis is the angle created by the backline. Graph 1 shows the welsh team and from the graph you can see that the line encroaches and flattens. While graph 2 shows that Italy are flat but at the end of the phase they don't hold that line.

This detail would allow the user to highlight this to the team and if needed corrections could be made.

Figure 32 - Depth of line graphs



Distance from Gainline

What does it show?

A teams backline tends to stand further from the gainline when they are about to kick. When standing further back the kicker has more time to kick and the backline needs to be behind him so it is a very useful measurement to see when a team is likely to kick. It is also a useful measurement in the defence of a kick. The back 3 players (wingers and fullback) form a triangle to defend against this. The distance they stand can affect the decisions that are made for where to kick. (e.g. The back 3 may stand further back if the wind is in their face. As the kick will travel further.)

It also useful in the attacking and defence lines. In attack if a team is to far away then they will probably lose ground on the gainline. Where as in defence the further away you are the more ground you are likely to give up. Therefore, in a team analysis session a measurement like this is heavily analysed as it can be a bug factor in the phase. Also at the end of the phase a play has worked if ground has been gained over the gainline.

Maths

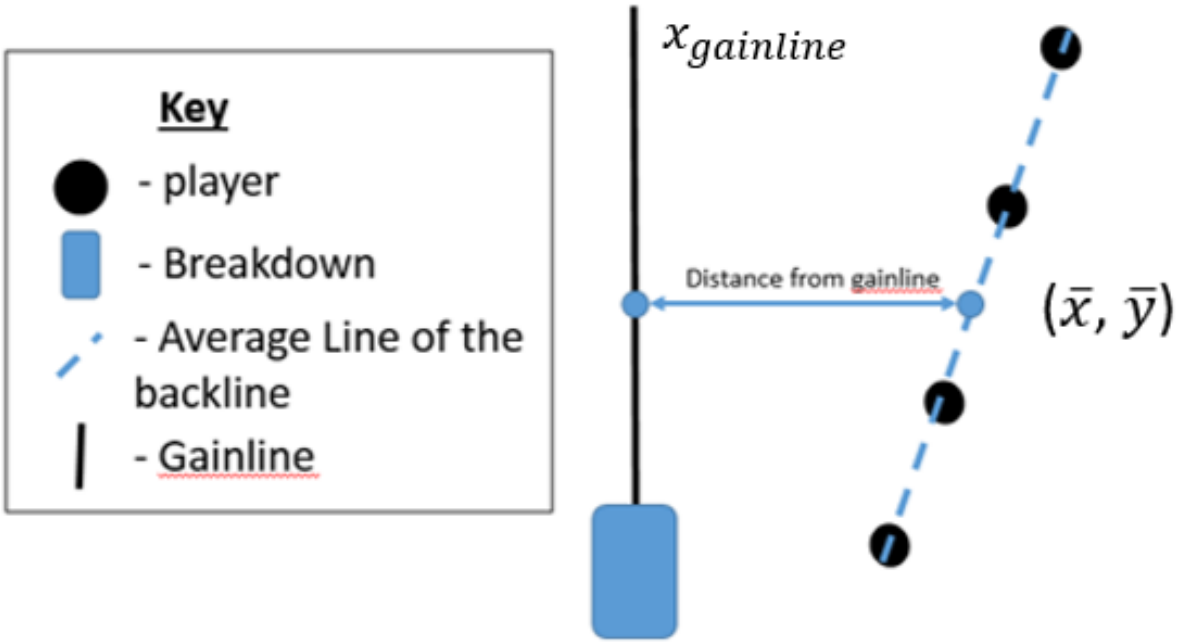


Figure 33 - Distance from gainline

The average x value of the back line is taken away from the x value set for the gainline to get the distance. The values taken are negative if the backline is on the left and positive on the right.

$$Distance\ from\ gainline = x_{gainline} - \bar{x}$$

Code

The code is one line as the work is done before to set the extra variables. As stated the average value is taken from the mainline value and converted to meters before stored.

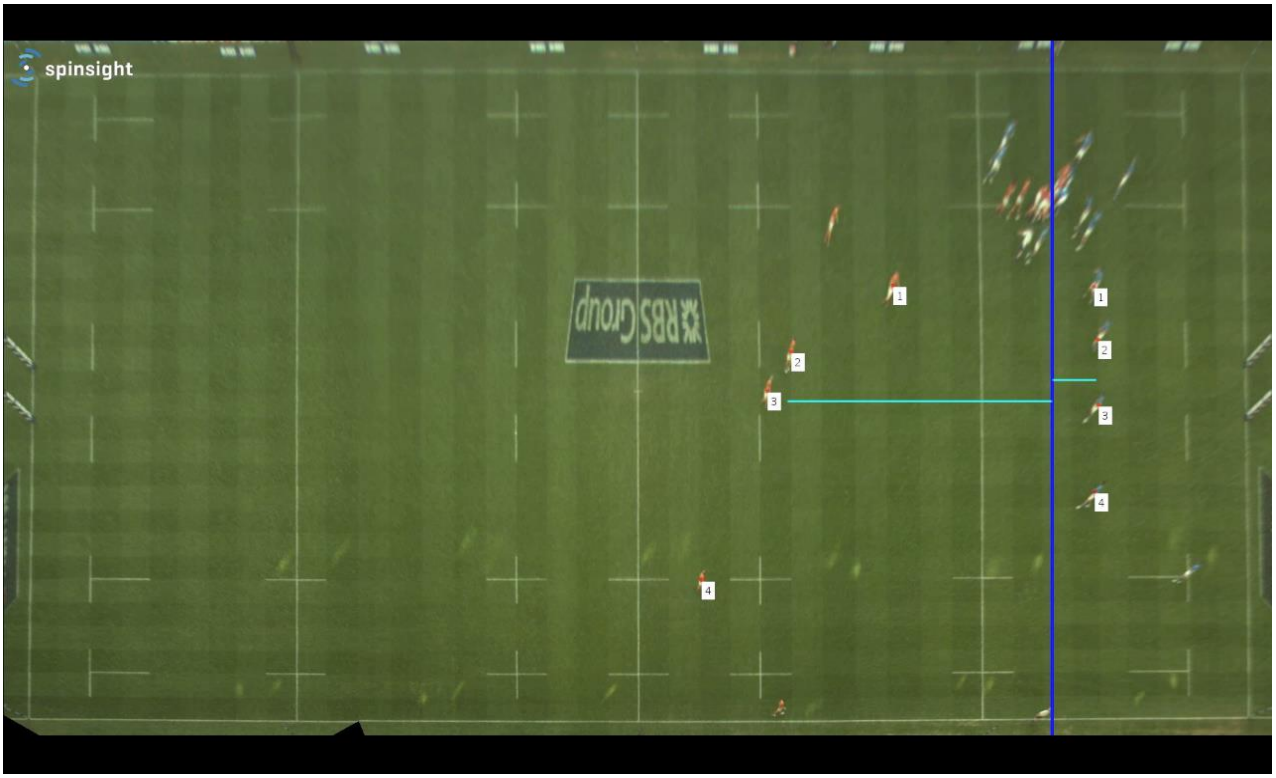
```
group(i).DisFromGainline = round((x_group(i)-gainline)*pixel2metre);
```

Features on Video

To show this on the video the graphic is a cyan colored line that it connected to the average X and Y value of the selected players and run parallel to the X axis to the gainline. The length of the line clearly shows the distance value and displays the measurement clearly.

Additional to the distance measurement the graphic also shows an additional measurement that could be used if more research is done in to it. As the line uses the selected players average Y position it shows the width of a backline. In plays some teams may look to stand outside the defence to be able to exploit space outside. This would need more research in future projects.

Figure 34- Distance from gainline frame



Graphs

The graphs are a simple line graph that show the change in the value. The Y axis is in meters.

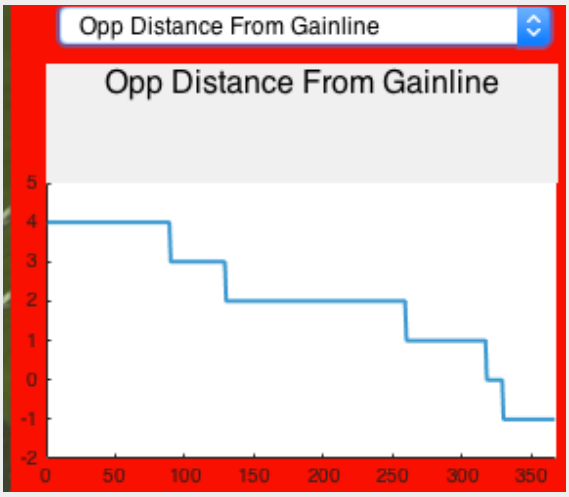
1. Welsh Team Distance from Gainline	2. Italy Team Distance from Gainline
	

Figure 35 - Distance from Gainline graphs

Tracking Back 3

What does it show?

As stated in the distance from gainline the back 3 look to defend from kicks from the opposition. These form a triangle and try to cover the pitch so it is difficult for the opposition team. The triangle that is formed is a right angle triangle with the fullback usually the corner in the right angle triangle.

As the breakdown / restart play moves across the pitch the 3 players will move to make sure spaces are covered but throughout the play a triangle is always kept as shown in the diagram below.

Maths

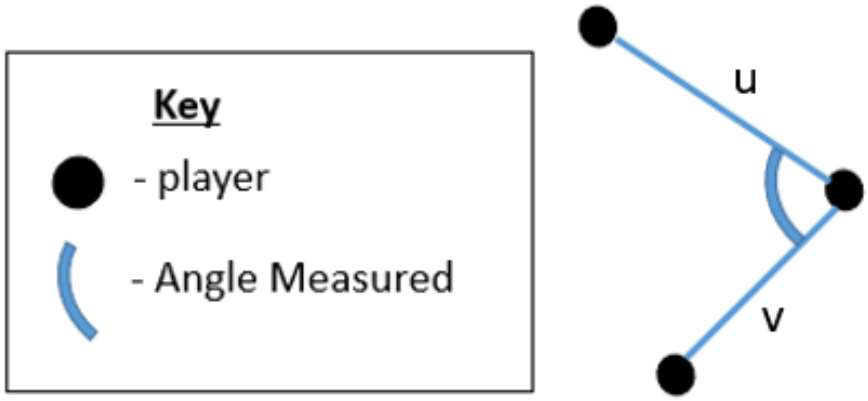


Figure 36 - Back 3

The angle that is measured is always the inside angle of the middle man in terms of the Y axis.

$$\emptyset = \cos^{-1}\left(\frac{\vec{u}.\vec{v}}{[\![\vec{u}]\!][\![\vec{v}]\!]}\right)$$

Code

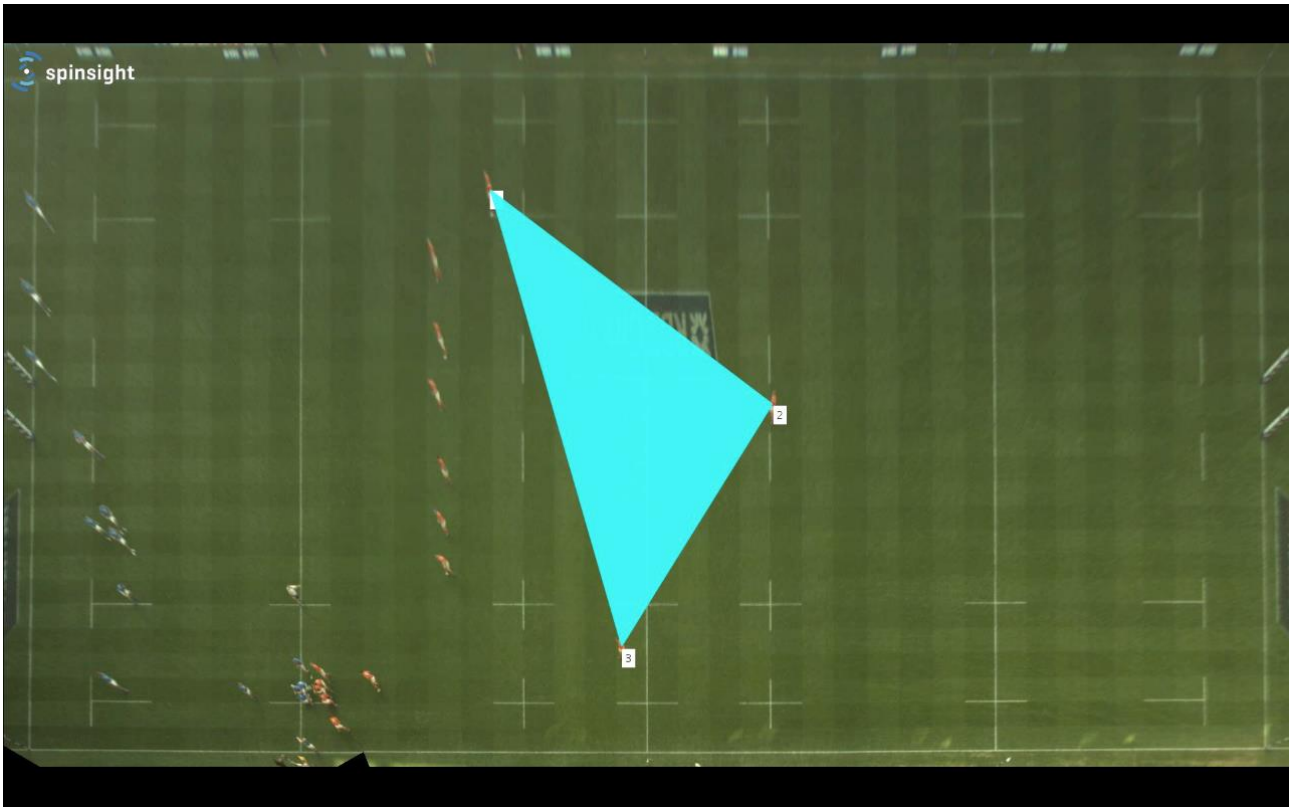
Here it will only run the code if 3 players have been selected. This is due to the complexity when less or more players are selected as it is no longer a triangle. Future work on this could look at further possibilities when extra player step back to help the back 3.

```
if length(player) == 3
        P0 = [sortedX(2), sortedY(2)];
        P1 = [sortedX(1), sortedY(1)];
        P2 = [sortedX(3), sortedY(3)];


        ang = round(radtodeg(atan2(abs(det([P2-P0;P1-P0])),dot(P2-P0,P1
                        P0))));
        group(i).Back_3_Ang = ang;

 end
```

Features on Video

The graphics used here is quite primitive and could be upgraded in future developments. Currently the graphic is a basic blue triangle to show the user the connectivity between the players. It also shows how they move together as the triangle deforms.

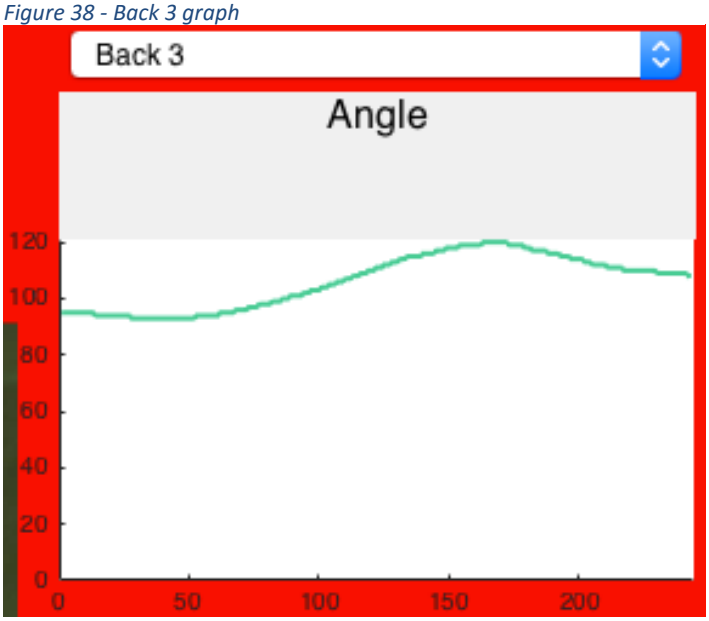
Figure 37 - Back 3 frame



Graphs

The graph is a line graph as this give the best representation of the changes that occur to the angle during the phase. The Y axis is the angle size is degrees. Below is the graph that shows the changes that occur in the featured video. From this graph the analyst teams can see what angle works best for them in a play.

Figure 38 - Back 3 graph



Distance between Players

What does it show?

Plays within rugby is all about the movement of people and the ball. So the standard measurement in all plays is the relative distance between players. In defence players are usually a set distance apart. This is so players don't defend an area too big for them or if they are all to close space is left on the outside. Whereas in attack teams will play around with the relative distance between them to try and pull the defence out of shape.

This measurement was always going to be set due to the amount of analysis it has during team meetings. Some attacks and defences rely on this statistic a lot as if it is out then plays will start to fail.

Maths

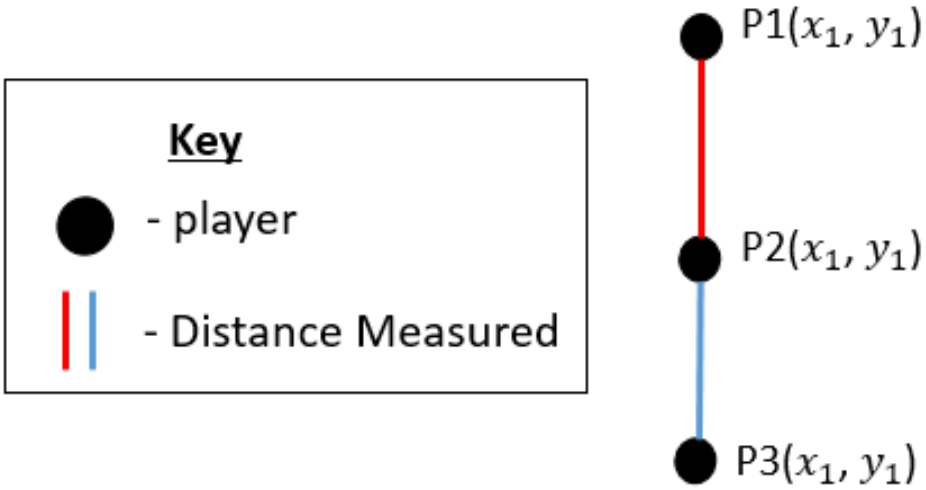


Figure 39 - Distance between Players

As stated the player are sorted by the Y axis and the distance values needed will always be NumberOfPlayers-1. It is calculated using the pixel distance between each player.

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Code

The code loops through each pair of players in the sorted list. It calculates the pixel distance between the two players and stores it in an array which is then stored in the group file.

```
% Distance between players
   for count = 1:(length(sortedX)-1)
      points = [sortedX(count),sortedY(count);
                        sortedX(count+1),sortedY(count+1)];
      pixelDis = pdist(points, 'euclidean');
      dis = round(pixelDis*pixel2metre);
      group(i).PlayerDis = [group(i).PlayerDis, dis];
   end
```

Features on Video

The graphics on the video are important in this as they needed to highlight the different gaps and the distance between the relevant players. Below in the frame capture you can see that this is done by a different coloured line between each pair of players. The line shows the gap clearly and the angle to the player next to them.

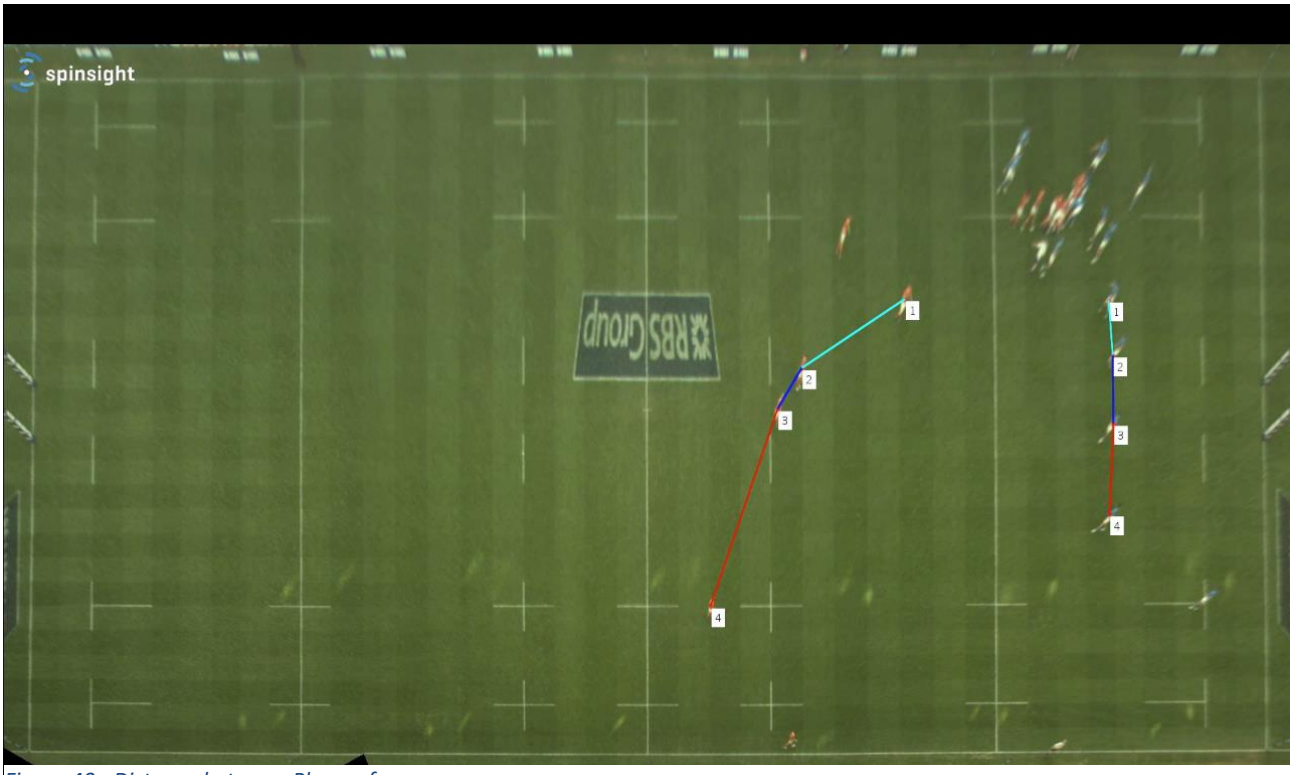


Figure 40 - Distance between Players frame

Graphs

The graph had a similar issue to the graphics being displayed on video where it needed to highlight the different gaps. The extra issue with the graphs is relating each line to the graphics displayed on the video. This is done through the colour of each line. Each line shown in the graph corresponds to its coloured line in the video.

The Y axis is in meters and the values are shown using a line graph for better comparison over time. The graphs below link to the frame capture above.

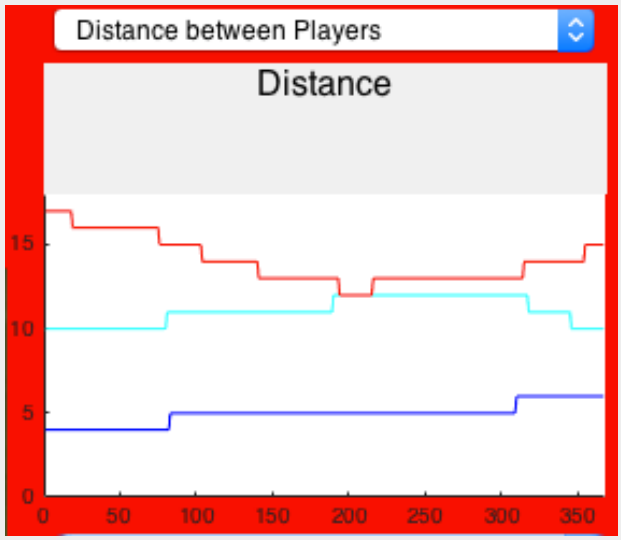
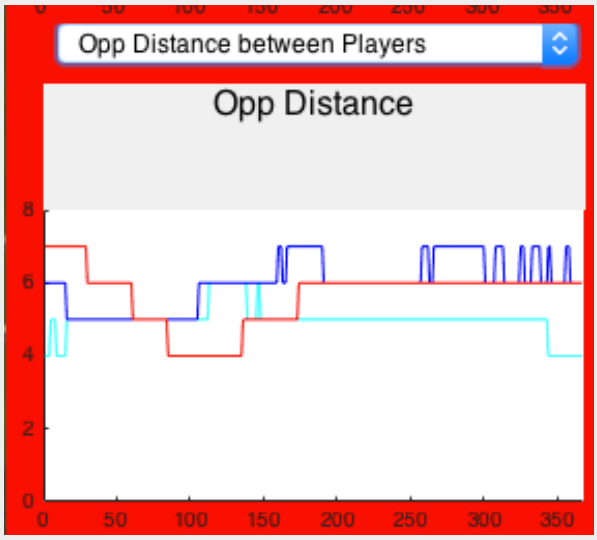
1. Welsh Team Distance between Players	2. Italy Team Distance between Players
	

Figure 41 - Distance between Players graphs

Group Displacement

What does it show?

This was a very simple measurement that just gives the analysis team another value to work with. This measurement is measured using the selected players average X and Y start positions and location in the current frame. In some plays they may not want to know the relative distance to the gainline as the ball stays with the forwards. Where this is useful is to see if they encroach on the breakdown when they should have been hold position.

Maths

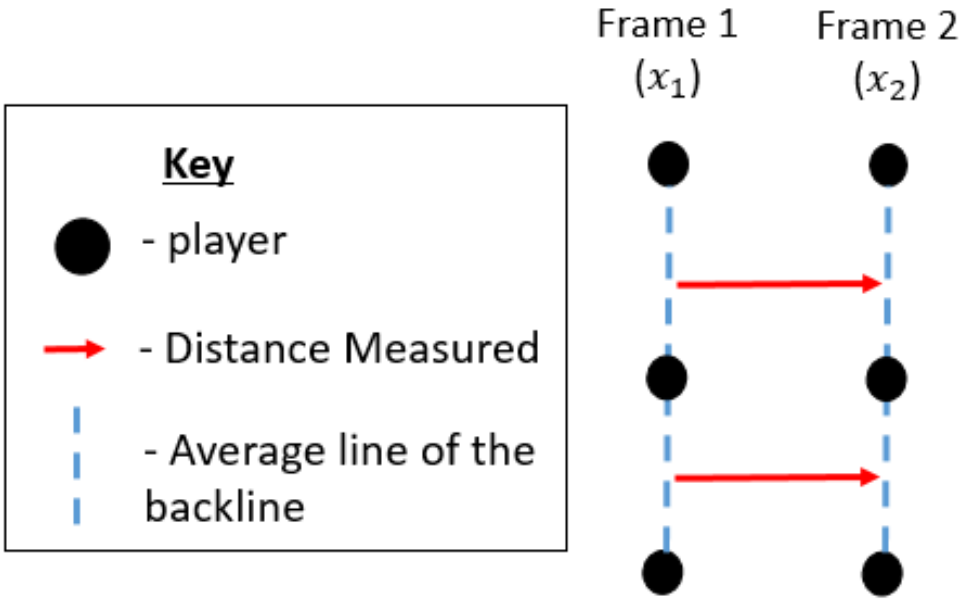


Figure 42 - Group Displacement

Here the starting average x value of the line is taken away from the line current frame average x value of the line. This gives the displacement of the line during the phase play.

$$Displacement = x_2 - x_1$$

Code

The code measures both the X axis and the Y axis but only the X axis is currently used with the program.

```
%X axis
group(i).x_group = (x_group(i)-x_group_start)*pixel2metre;
%Y axis
group(i).y_group = (y_group(i)-y_group_start)*pixel2metre;
```

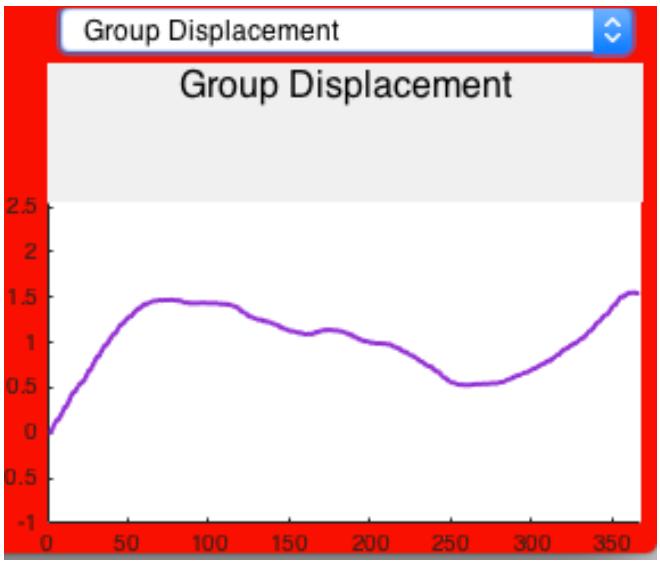
Features on Video

There is no graphics implemented for this measurement as it is quite a primitive measurement. If it is required by the user in the future a graphic could be added.

Graphs

As there is no graphics it relies on the graph to display the measurements captured. This is a simple line graph that shows the displacement of the group in the X axis. With the Y axis of the graph in meters. Although the Y is measured it is not currently used. This is due to the fact that the X axis shows the movement up and down the field which is more interesting to teams.

Figure 43 - Group Displacement graphs



Live Analysis

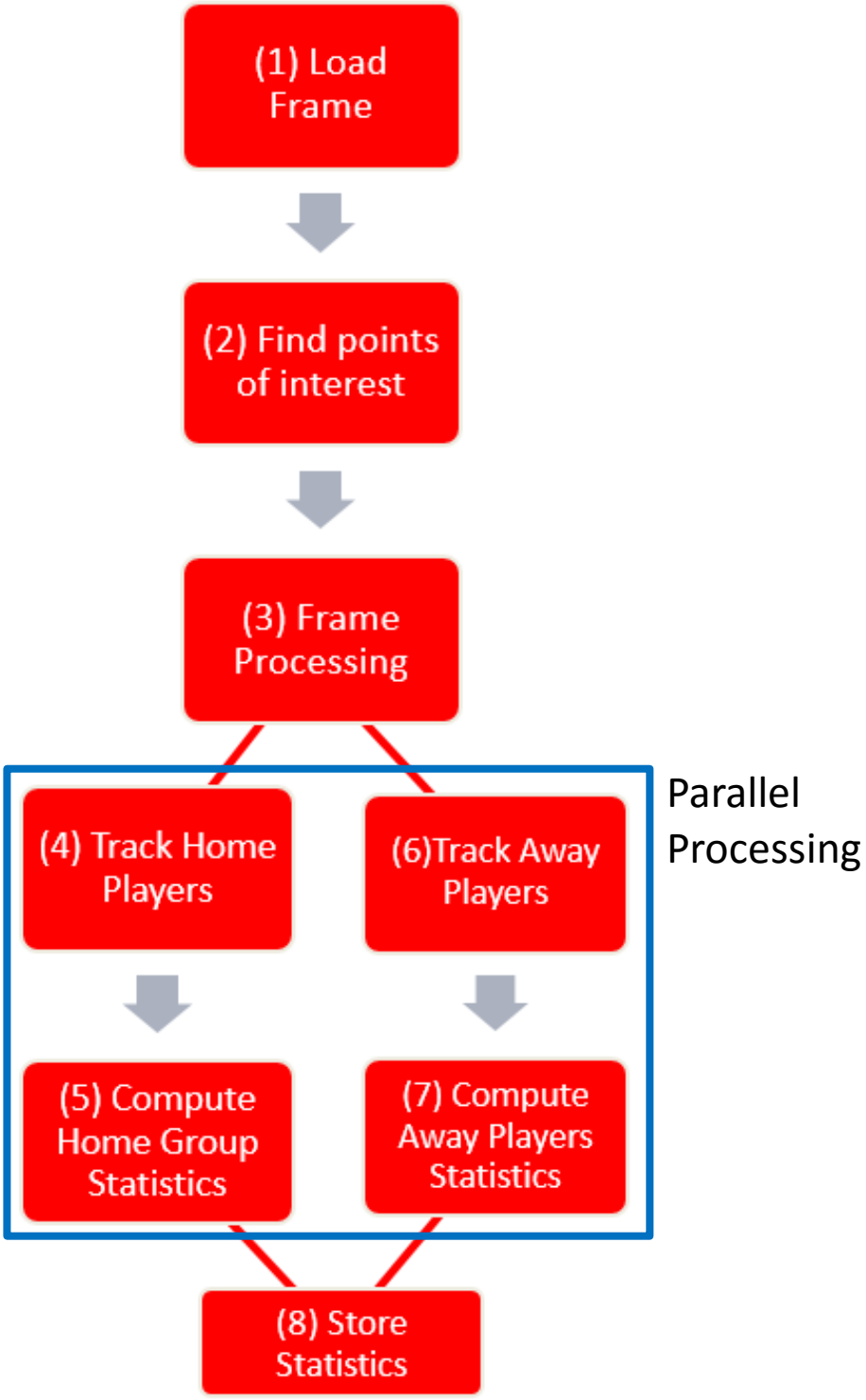
During the project it was looked to see if the analysis could be done through a frame by frame analysis of the videos. This would include tracking the players and updating the results of the group analysis after each frame has completed. It would also involve the live plotting of graphs and graphics on to the video.

Due to time scales it was not progressed further than an initial development of trying to plot live graphics and graphs. Although from the initial development and research into how it could be done it would be an area that could be looked at with confidence that it would bring a good value to the program.

The initial development worked at plotting the graphics live along with the graphs. This caused issues to the usability of the program as it slowed the processing up considerably. This could have been down to a number of factors. Firstly, the program was developed on a laptop and this could be an issue as the processing power may have been an issue. On investigation of this it did not seem to be an issue with memory levels. Secondly, the code is not optimised for live tracking. If live analysis was to be done then this would have to be addressed for example running the code in parallel and reducing the number of lines needed to produce a result. An effort was made to reduce the number of lines being run but when plotting the graphs the video frame rate decreased. This made the third aim of a workable UI difficult to hit so progress on this was halted.

As stated the other area for live analysis would be the tracking system. For this the tracking system would have to be deconstructed and rebuilt so it does the tracking frame by frame rather than a whole clip in each section of the tracking method.

Figure 44 - Live tracking example



This is certainly an area of the project that would be continued in future developments as the benefits for live analysis during games are huge to the coaches and teams.

Results & Evaluation

Testing

The testing has been run throughout the project as the group statistics needed to work for them to be analysed for their usefulness. Below are the tests used on the features and group statistics. These were run multiple times during the project but due to time and resources not all have been captured below. The tests below are more formal tests and can be re-run in the future if changes are made. Tests below are not based on the UI as the project was not set to build a UI therefore it does not test every button click and pathway through the program.

The values used during the tests are set by eye. It is done this way due to the time scales of the project and it is the easiest way to test many phases at once. If more time was available then the values could have been measured accurately and over the entire frame. But due to the development of the group statistics and the confidence in the results from intermediate tests throughout. It was thought that a slimmer line of testing would work for the project. With testing the features it is known that they have not been implemented to hold up to situations outside the normal process. (For example not selecting players). This will show in the tests but understanding the limitations of the program will help in the future developments. The tests where possible will have extra comments and issues that have been raised when running the tests.

In the tests below the issues are given a classification of risk. The levels set are unclassified, low, medium and high. These are the levels at which they impact the running of the program. Below are the meaning of the classifications.

Unclassified - An issue has been raised by the tester but is not sure if it is an issue as it cannot be replicated confidently or if it was a miss understanding by the tester.

Low - An issue has been identified but runs no threat to the program running or the outputs from the tested function. (This is usually a UI error on displaying information.)

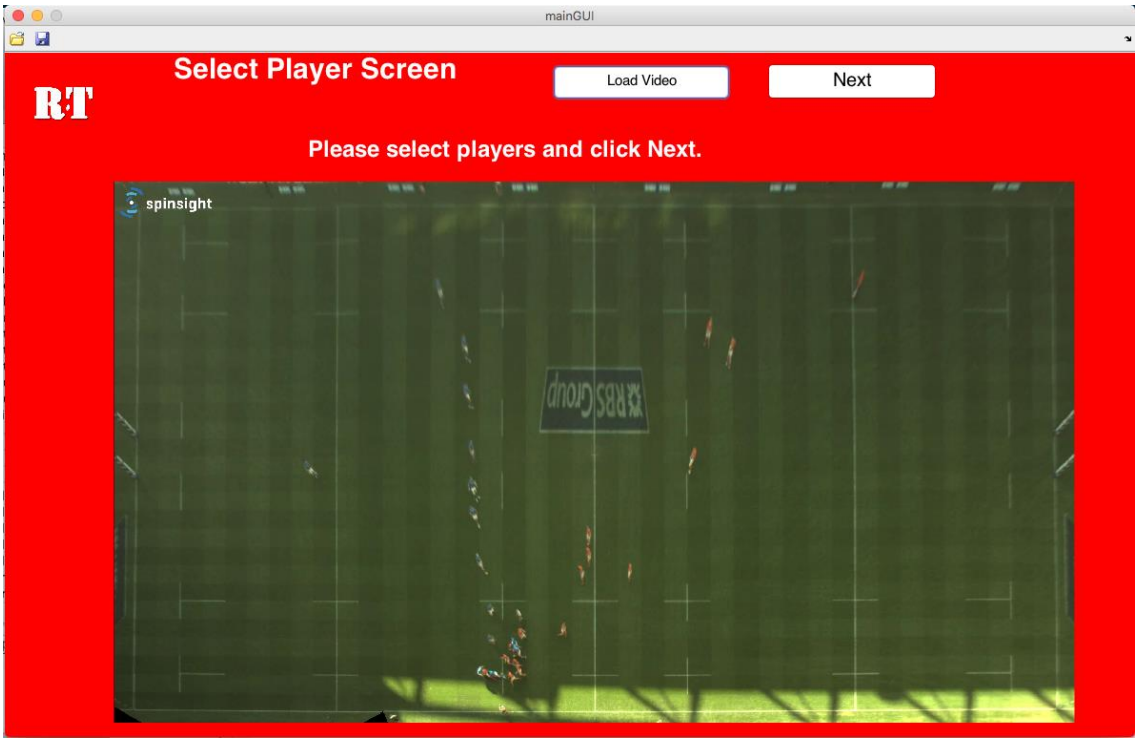
Medium - Here the issue will start to effect the running of the program. It does not affect the output of the function. This could be related to optimisation problems.

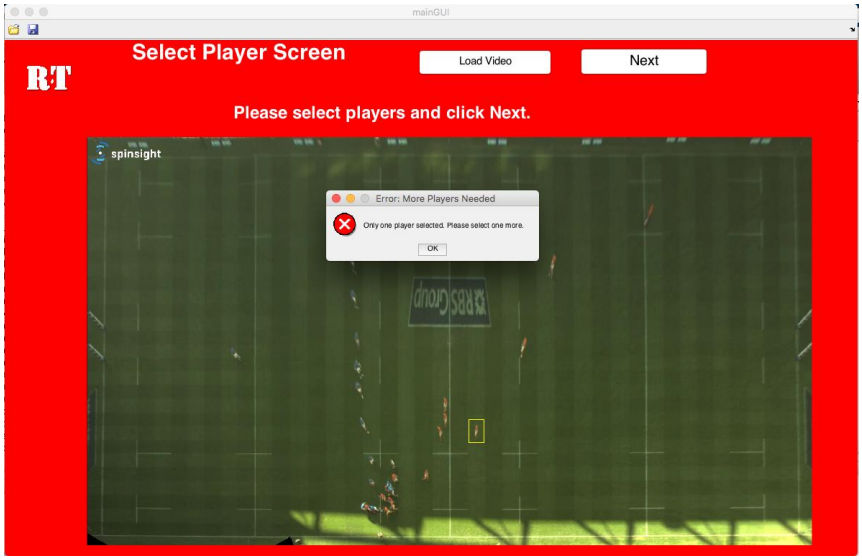
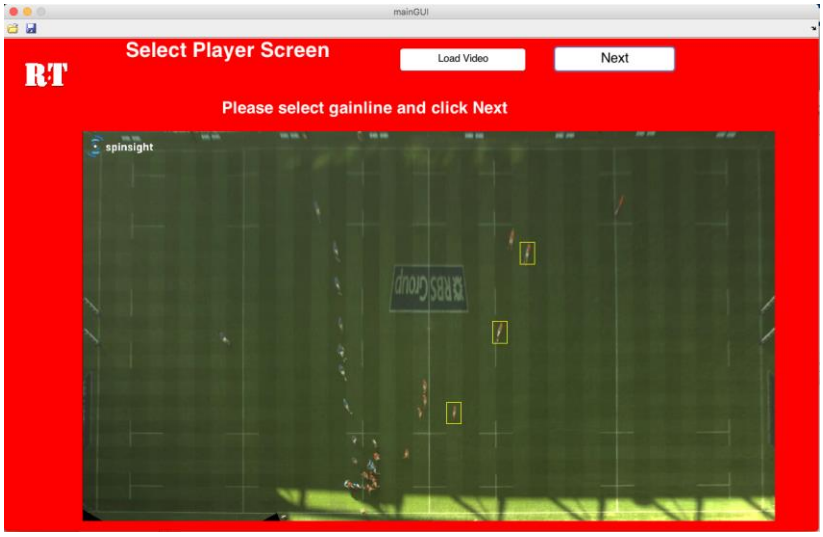
High - The issue effects the output of the function and or the run time of the program. The issue here will need to be dealt with the highest priorities in future developments.

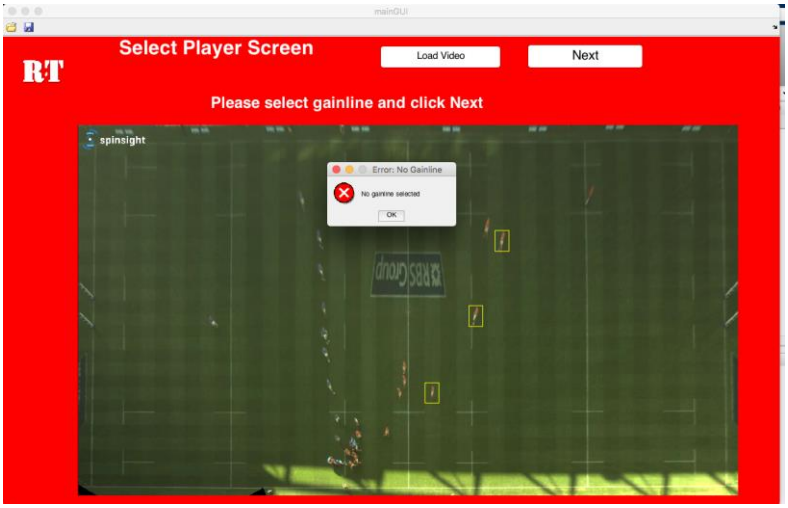
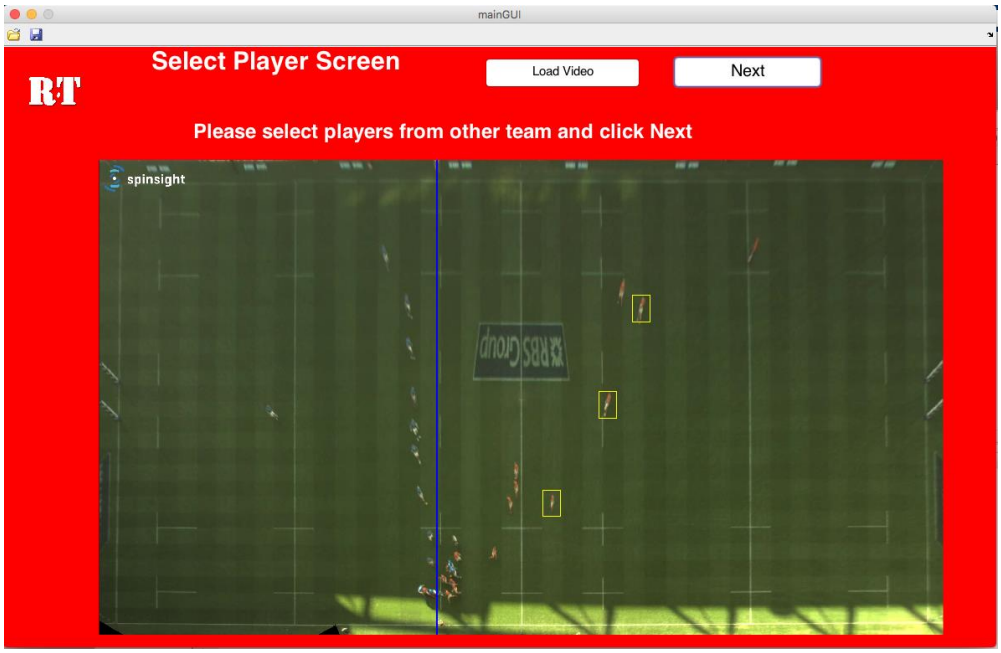
Also during the test words will be highlighted in italics. This is to highlight that it is a UI object that is to be interacted with.

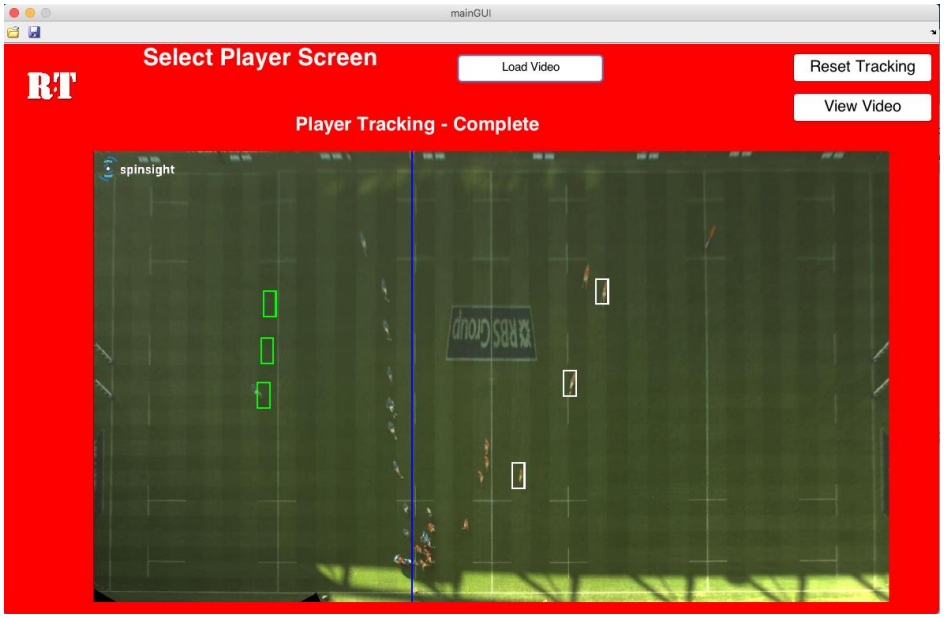
Selecting players

This looks at testing the method of selecting players and gainline. Here there are some error captures as the program relies on clear select points for the tracking to work. Below are the testing results.

	Action	Result	Passed / Failed
1	Select and load a phase play video	Video should be selected and the first frame should be loaded into the select_player_GUI window. select_player_GUI should enter State 2 as described in the User Interfaces section	Passed
			

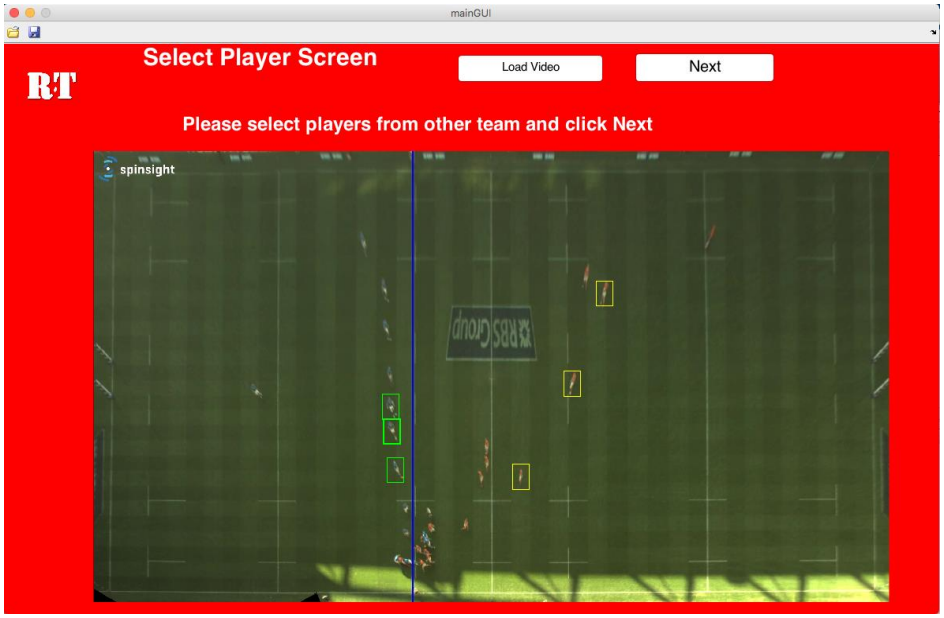
2	Select only one Home team player	A yellow bounding box should show around selected player. Click *Next* button. An error message should show asking for 2 or more players to be selected. Click *OK* and message box should disappear	Passed
			
3	Select 2 or more home team players	Home team players should show with a yellow bounding box around the players selected. Click *Next* - UI should now ask for gainline.	Passed
			
4	Do not select gainline and click *Next*	When selecting the gainline a blue line running parallel to the Y axis should show where the user has selected. (Gainline only has X axis)	Passed

			
5	Select Gainline and click *Next*	When selecting the gainline a blue line running parallel to the Y axis should show where the user has selected. (Gainline only has X axis). The UI should ask for away team to be selected.	Passed
			
6	Select only one Away team player	A green bounding box should show around selected player. Click *Next* button. An error message should show asking for 2 or more players to be selected. Click *OK* and message box should disappear	Fail



This has failed as it has taken away team players from a previous tracked phase. This looks to be a case that the away players selected location file has not been reset.

7	Select 2 or more away team players	Away team players should show with a green bounding box around the players selected.	Passed



8	Click *Next* to complete selection.	The tracking method should now run and select_player_GUI should now enter stage 3 as described in the User Interfaces section.	Passed

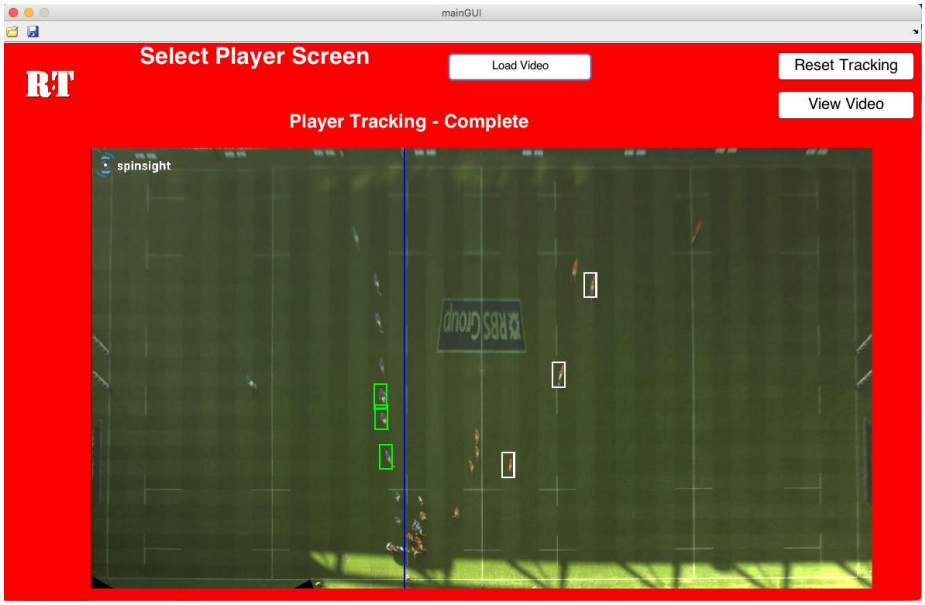
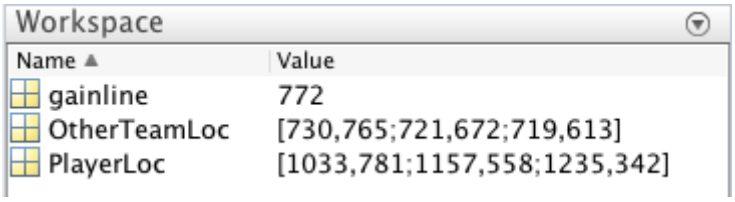
			
	The yellow box for the home team does change to white this is a small change and is not a risk to the running of the program.		
9	Check files have been created. (To do this load a file called 'Selected_Points.mat' into Matlab workspace)	The program should save the 3 arrays for future use. It should be saved as 'Selected_Points.mat' and hold 3 arrays as listed below. **gainline** Single value to hold the X value of gainline selected. **OtherTeamLoc** Holds the X and Y coordinates of the selected away team players. **PlayerLoc** Holds the X and Y coordinates of the selected home team players.	Passed
			
	Values have also been checked and match the players that have been selected.		

Table 3 - Selecting Players Testing

Extra Comments:

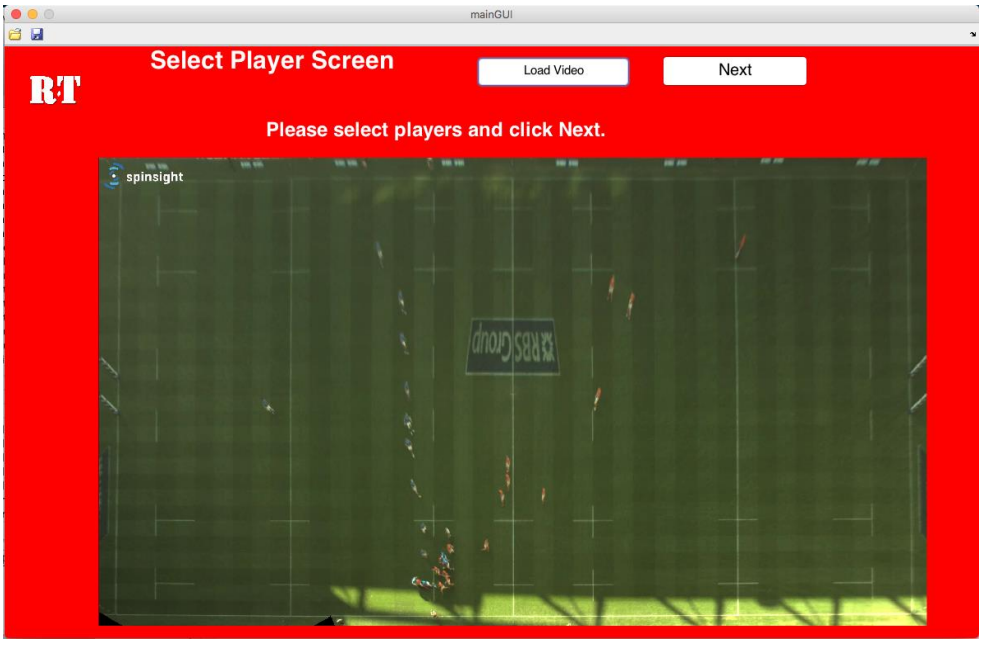
- There is no method to be able to remove a selection if a mistake is made. For this the user must close and start the process again.
- There is no check when players have been selected. It may be possible to add a check feature as players are being selected.
- Sometimes when the process/program is closed early the reading of the next video is slow.

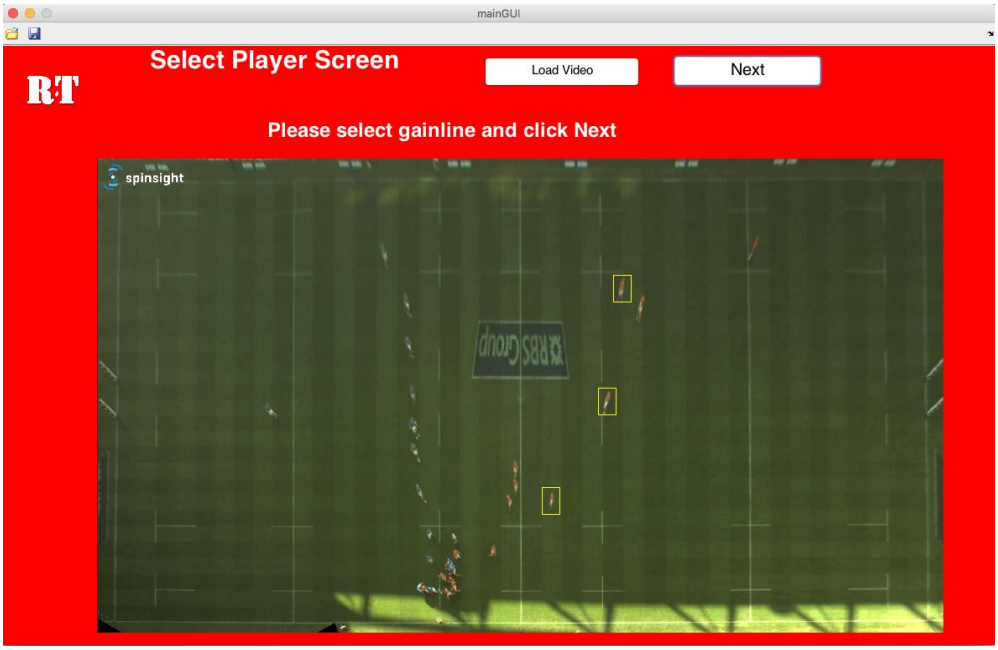
This is a very basic UI that works and allows the selection of players. It works well and has passed all but one of the basic tests above. If this issue can be fixed then it will improve the reliability of the feature. Also from the additional comments raised these are simple features that would be good as future implementations as they will also improve both reliability and robustness of the feature implemented.

Compare 2 players

When running the re-tracking method players must be compared to allow for less processing as players are not tacked twice. This section looks to see if the implementation of the feature works and if the percentage value set for the comparison method needs to be changed.

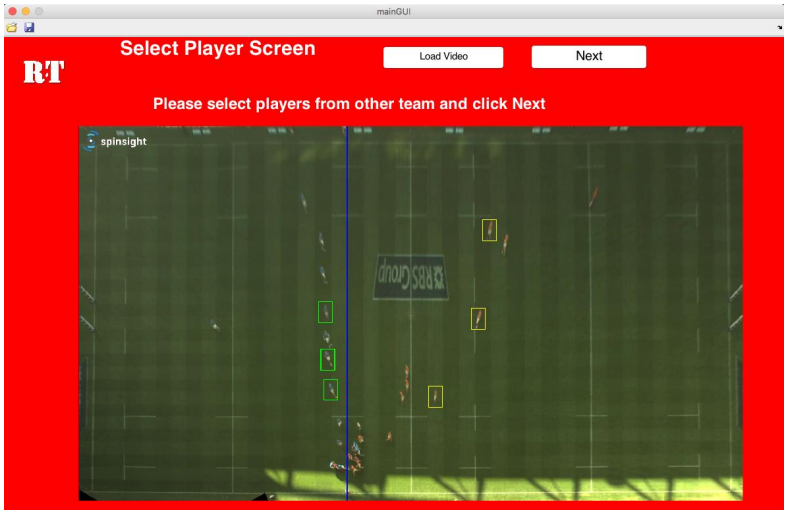
This carries on from the previous testing above:

	Action	Result	Passed / Failed
1	Click the *Reset Tracking* button	Program should enter back to state 2.	Passed
			
2	Select 2 of the same home team players and one new player. Then click *Next*	Bounding boxes should show and the program should ask for the gainline to be selected.	Passed



The player at the top of the home team players is the new player selected.

3	Select the gainline in the same position. Click *Next*	The gain line does not have a comparison method so it acts the same as before. The UI should now ask for away team players.	Passed
4	Select 2 of the same away team players and one new player.	Bounding boxes should show and the program should ask for the gainline to be selected.	Passed



The top player on the away team is the new player selected.

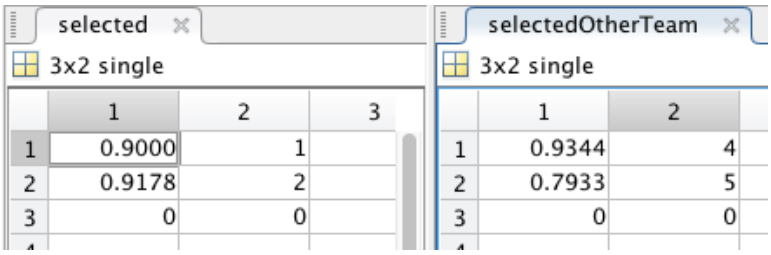
5	Place breakpoint to allow the analysis of the code and arrays created. Click *Next to run code.*	The code should produce 2 arrays for the home and away team. The arrays should be layout as below. For each player selected the array should hold the percentage covered and the number of the player that compares to it in OtherPlayer array. The 2 players reselected should have a comparison value over 70 percent and a number for the selected player. The new player should have a value less than 70 percent and an empty value for selected player from OtherPlayer array.	Passed
			

Table 4 - Compare 2 Players Testing

Extra Comments:

The player comparison is done after the preprocessing of the video. This could be swapped to help speed up the comparison and errors that are produced.

Additional test

An additional test was also run to see what happens when players stand close as the bounding boxes will overlap. This tests the percentage value selected. The video selected has 3 players on the away team close to each other. As shown in the frame capture below the bounding boxes overlap. This test looked to see if a comparison would identify the correct players from the Other Players array.

Figure 45 - Players standing close

The same home team players are selected and 2 of the players along with a new player is selected for the away team. Here it should identify the new player and correctly identify the 2 players that are already known and their linked players position in the Other Players array.

The new selected players with the bounding boxes.

Below are the results from the test. It shows the program has been able to correctly identify the players that are new and correctly identify the position of the found player in the Other Player array.

New players selected. Location Array

PlayerLoc

3x2 double

	1	2
1	658	782
2	652	714
3	656	638

OtherTeamLoc

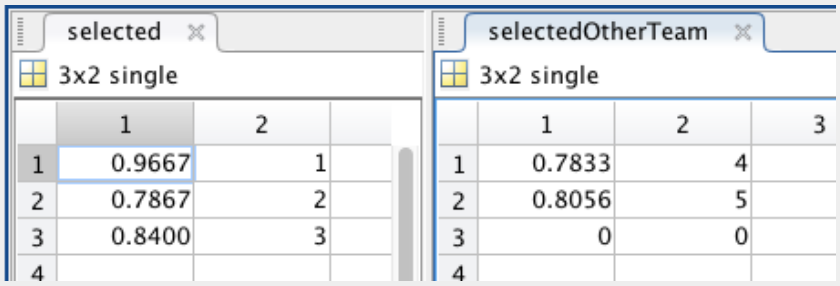
3x2 double

	1	2
1	178	746
2	138	723
3	157	701

Other Players array with previous scanned players

Fields	boundingbox	X_start	X1	Y1	Y_start	
1	[645.5000,761....	660.5596	1x205 si...	1x205 si...	776.9993	1
2	[643.5000,695....	652.3973	1x243 si...	1x243 si...	708.7796	1
3	[652.5000,631....	660.4378	1x243 si...	1x243 si...	641.6703	1
4	[160.5000,724....	174.3294	1x243 si...	1x243 si...	735.0963	1
5	[133.5000,715....	142.9501	1x243 si...	1x243 si...	723.1135	1
6	[154.5000,701....	164.3164	1x243 si...	1x243 si...	710.8764	1

Comparison arrays for both teams.



Extra Comments

- The comparison also works well for people close to each other. This is good as bounding boxes overlap in this case and values are seen.
- The players were close but not overlapping. Players that overlap may see an issue in tracking.

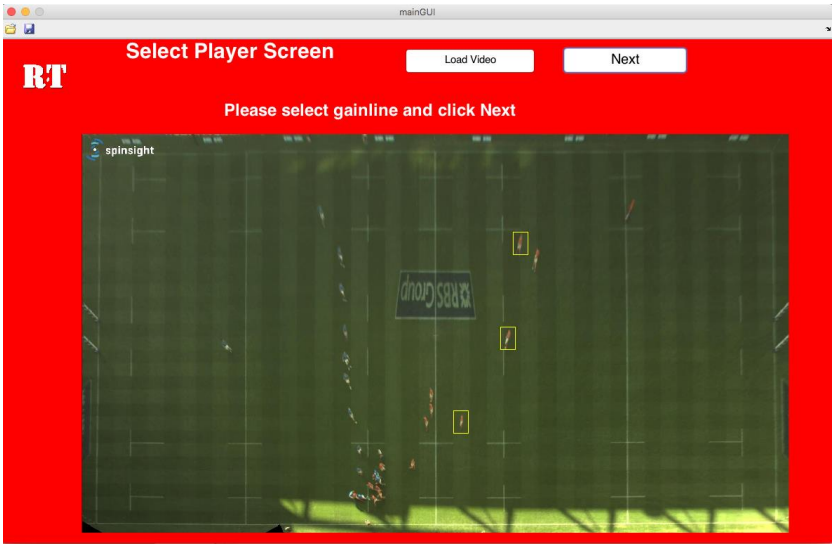
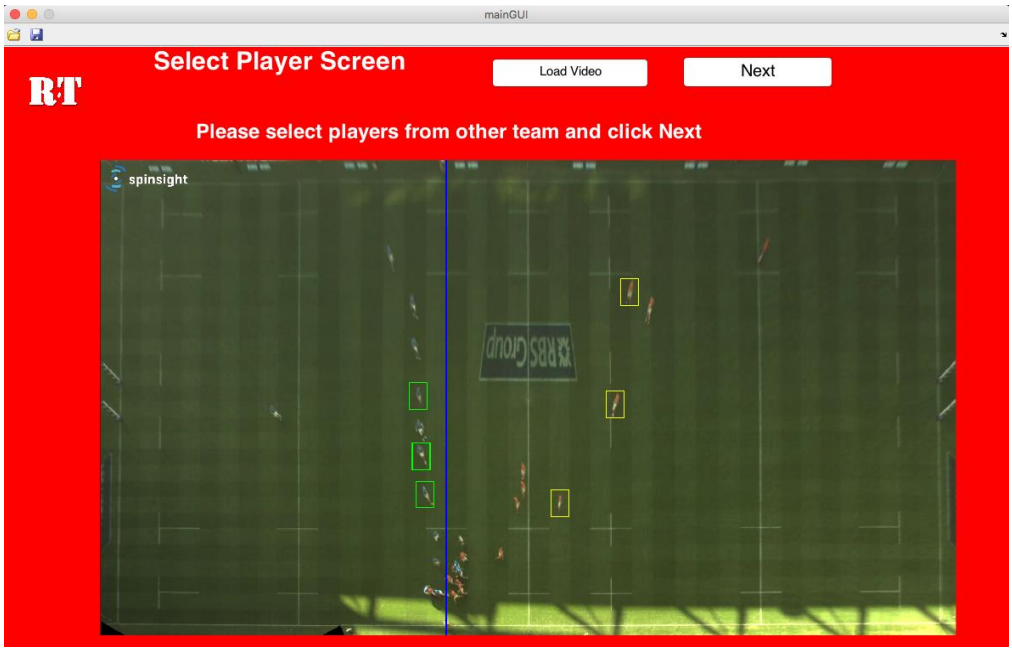
This feature works well and is robust for the system. This is good as the re-tracking method bases its work on the results from the comparison of two players. The feature is only effected by the fact that the video processing is done before it slowing it up. This does not need to be the case as it doesn't need to wait for the video processing therefore it could be moved. The only variable I would alter is the percentage covered for a match to be found. If in future projects it needs to be changed then it is simple and easy to do and would not affect the code produced.

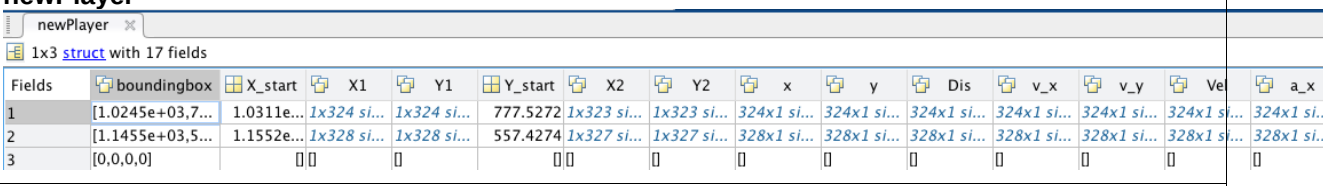
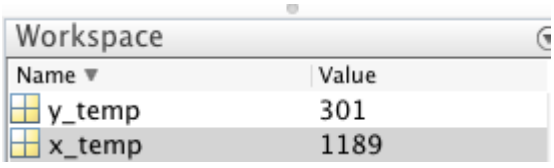
Re-tracking Players

This feature makes sure that players are not tracked again so there is less processing is done by the system. The decisions made are based on the arrays passed from comparing 2 players. The test looks to see if each step of the process (as described in the implementation section) is run correctly.

This test carries on from the selecting players test:

	Action	Result	Passed / Failed
1	Click Reset Tracking method	select_player_GUI should enter state 2.	Passed
2	Load 'OtherPlayers.mat'	All players selected and their statistics in the select players test should be copied across to Other Players.	

3	Select 2 of the same home team players and one new player. Then click *Next*	Bounding boxes should show and the program should ask for the gainline to be selected.	Passed
			
	The player at the top of the home team players is the new player selected.		
4	Select the gainline in the same position. Click *Next*	The gain line does not have a comparison method so it acts the same as before. The UI should now ask for away team players.	Passed
5	Select 2 of the same away team players and one new player.	Bounding boxes should show and the program should ask for the gainline to be selected.	Passed
			

	The top player on the away team is the new player selected.		
6	Place breakpoints in tracking method to identify the arrays listed in the steps below.	N/A	N/A
7	Click *Next*	Pre-Processing of the Video should not be run. The video has already been through the process. This saves time.	Passed
8	First Breakpoint hit. For home team.	Players have now been split. The arrays below should be created based on the compare 2 players arrays. **New players array** Based on the compare 2 players array this should hold 2 players as have been tracked before. It should also have one empty slot for the unknown user. **x_temp** This should hold one x value for the new player that has to be tracked **y_temp** This should hold the corresponding Y value for the new player that has to be tracked	Passed
	newPlayer 		
			
9	Click ***Continue*** to allow code to run.	Home team player should now be tracked. Time for tracking should be reduced	Passed
10	Second Breakpoint hit. For home team.	The 2 arrays should now be concatenated into one array with all player statistics available.	Failed

player

1x3 struct with 17 fields

Fields	X2	Y2	x	y	Dis	v_x	v_y	Vel	a_x	a_y	Acc	theta
1	23 si...	1x323 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...	324x1 si...
2	27 si...	1x327 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...	328x1 si...
3	15 si...	1x615 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...	[]

This has failed as the theta value has not be copied across when concatenating the new players and old players. This does not cause an issue at the moment as theta is not used to graph results in the program. This should be a quick fix to insert a line that copies the value across.

11	Click ***Continue*** to allow code to run.	Home team players group statistics should now be recomputed.	Passed
12	First Breakpoint hit. For away team.	Players have now been split. The arrays below should be created based on the compare 2 players arrays. **New players array** Based on the compare 2 players array this should hold 2 players as have been tracked before. It should also have one empty slot for the unknown user. **x_temp** This should hold one x value for the new player that has to be tracked **y_temp** This should hold the corresponding Y value for the new player that has to be tracked	Passed

newPlayer array

newPlayer

1x3 struct with 17 fields

Fields	boundingbox	X_start	X1	Y1	Y_start	X2	Y2	x	y	Dis	v_x
1	[721.5000,740....	731.3252	1x616 si...	1x616 si...	758.3127	1x615 si...	1x615 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...
2	[711.5000,647....	719.2510	1x616 si...	1x616 si...	661.3620	1x615 si...	1x615 si...	616x1 si...	616x1 si...	616x1 si...	616x1 si...
3	[0,0,0,0]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
4											

Workspace

Name	Value
y_temp	535
x_temp	713

13	Click ***Continue*** to allow code to run.	Away team player should now be tracked. Time for tracking should be reduced	Passed
14	Second Breakpoint hit. For home team.	The 2 arrays should now be concatenated into one array with all player statistics available.	Passed

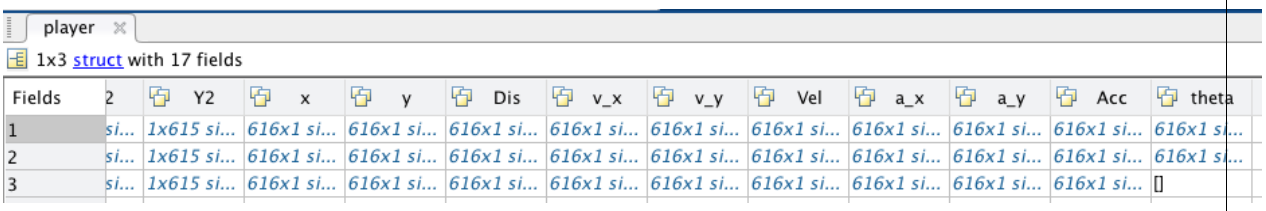
	 This has failed the same way as the previous failed step because the code is the same. The fix needs to add a line to the code to copy across the theta value.		
15	Click ***Continue*** to allow code to run.	Away team players group statistics should now be recomputed. select_player_GUI should enter state 3 with all players and gainline correctly selected.	Passed

Table 5 - Re-tracking Players Testing

Additional test

When players are moved to Other Players array a comparison between the players being added are made so the same player is not entered twice. The test carries on from the test above.

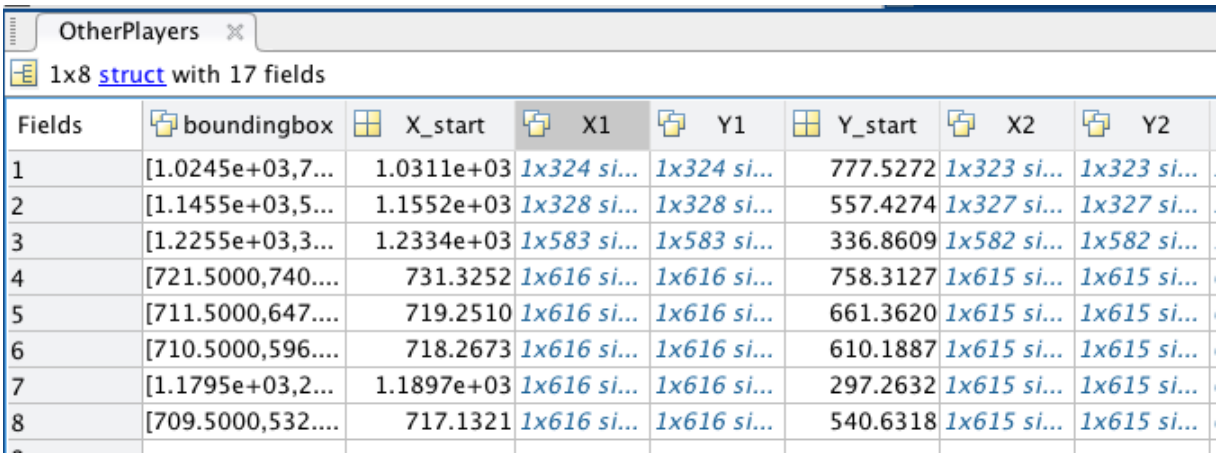
	Action	Result	Passed / Failed
1	Click Reset Tracking method	select_player_GUI should enter state 2.	Passed
2	Load ‘OtherPlayers.mat’ into Matlab workspace.	The array should hold 8 players from the 2 previously run tracking methods. Players should be unique with different start locations.	Passed
			

Table 6 - Additional Test: Re-tracking players

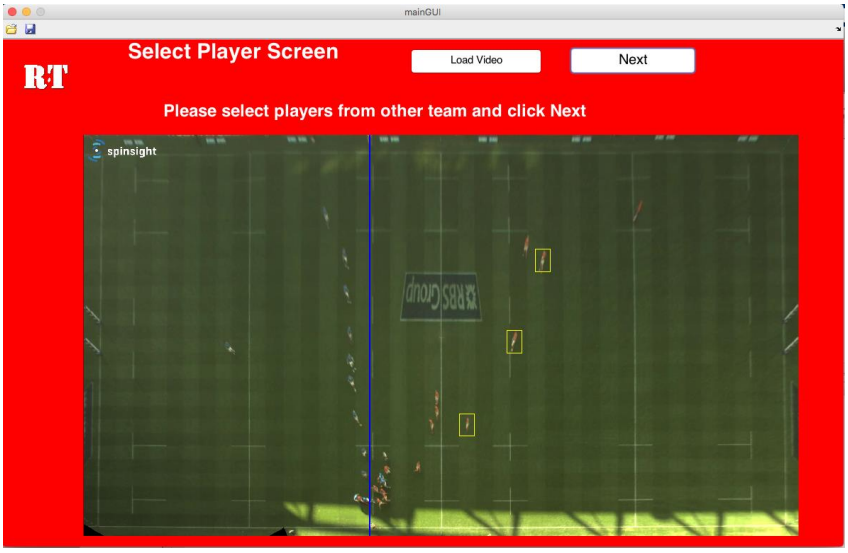
The testing shows that the re-tracking method works and that it does reduce the processing needed. There are some issues that were found within the tests that at the moment are low level risks as the values are not currently used. In the tracking method.

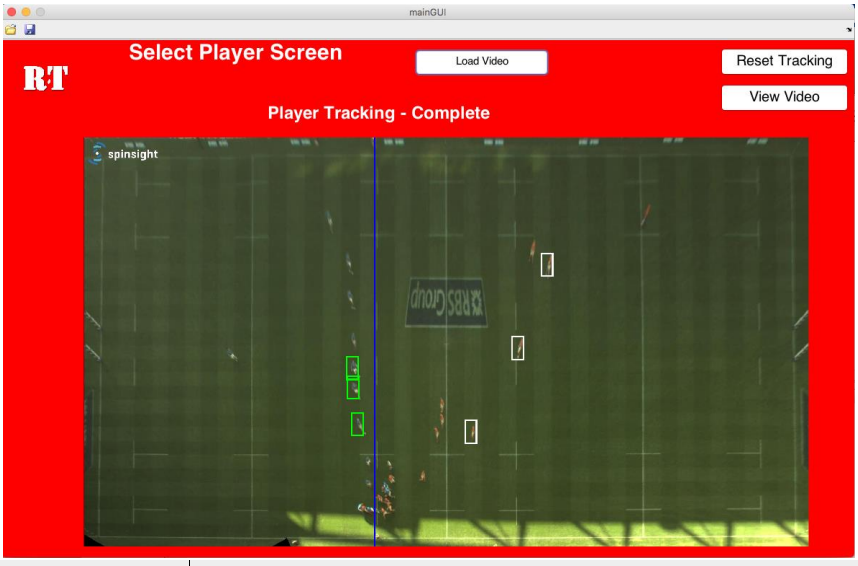
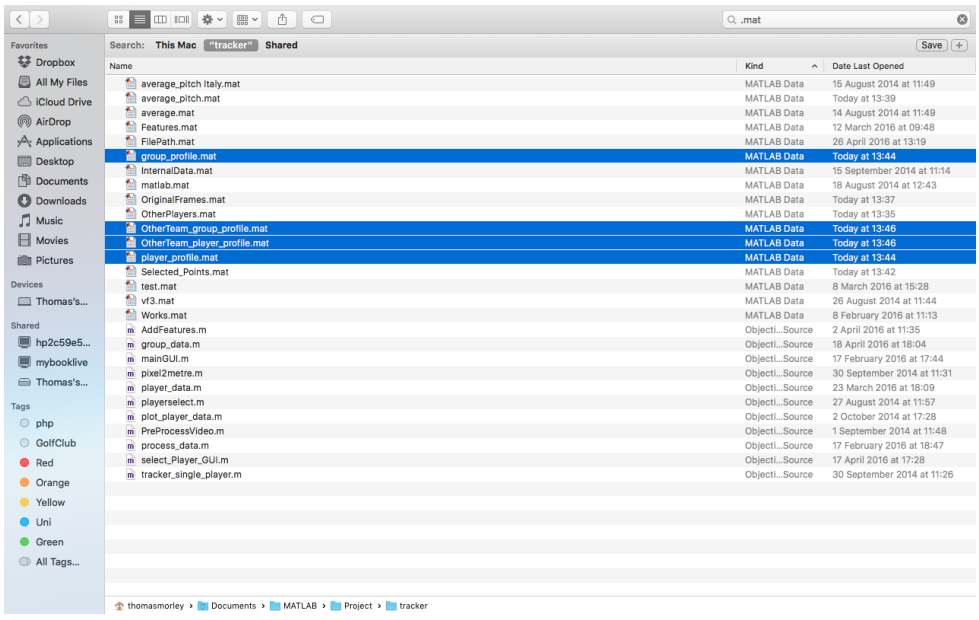
Tracking System

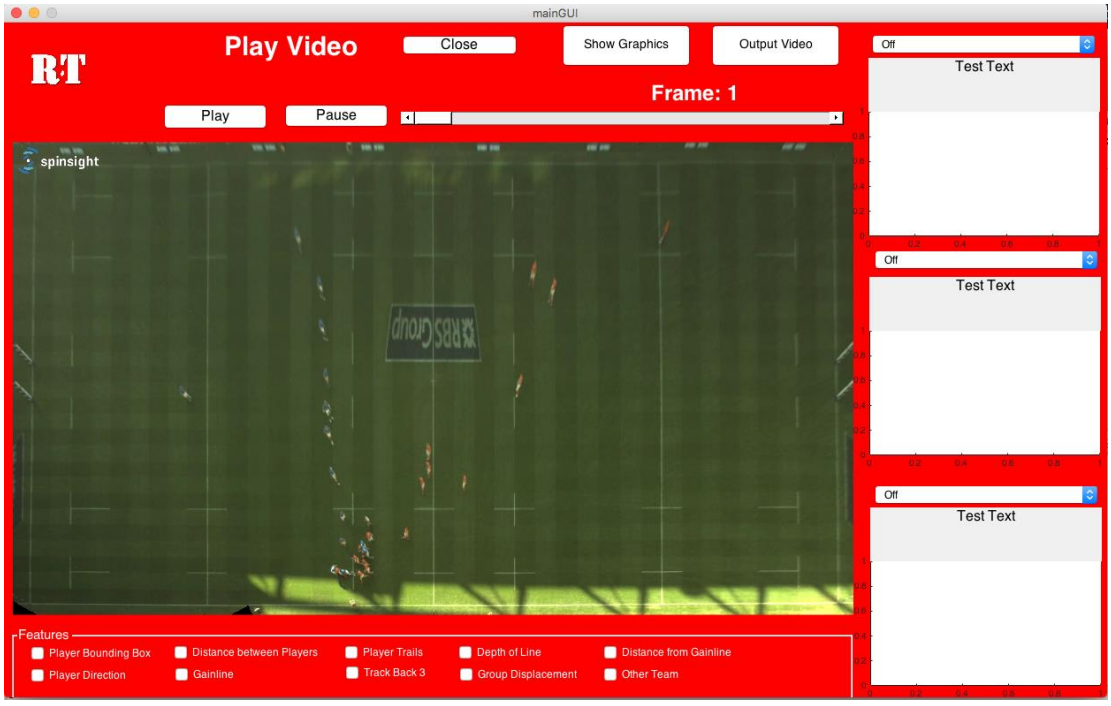
The system is heavily influenced by the tracking feature being reliable and robust. The tracking method has been tracked through as the player and group statistics rely on this and for every new track the method is used.

For the tracking method a graphic has been implemented to be able to see visually if the method has worked. Also the player files are checked for any sudden changes or loss of data in the system. Below is the test that was run various clips. The clips used tried to get players in different situations. For example, players standing close and players crossing can have an effect on the tracking method.

	Action	Result	Passed / Failed
1	Select and load a phase play video	Video should be selected and the first frame should be loaded into the select_player_GUI window. select_player_GUI should enter State 2 as described in the User Interfaces section	Passed
2	Select 3 home team players	Home team players should show with a yellow bounding box around the players selected. Click *Next* - UI should now ask for gainline.	Passed

			
3	Select Gainline and click *Next*	When selecting the gainline a blue line running parallel to the Y axis should show where the user has selected. (Gainline only has X axis). The UI should ask for away team to be selected.	Passed
4	Select 3 away team players	Away team players should show with a green bounding box around the players selected.	Passed
			
5	Click *Next* to complete selection.	Processing of the Video will be run first. Then the tracking method should be completed in the order as displayed below: (This should be visible from the UI text) 1. Tracking Home team players 2. Computing Home Group Statistics 3. Tracking Away team players 4. Computing Away Group Statistics. Once the tracking method has been run the Ui should enter state 3 and tracking should be complete	Passed

			
6	Check tracking files. Open project file location of the project.	In this file should be 4 files as named below all with a size value more than 0. • group_profile.mat • OtherTeam_group_profile.mat • player_profile.mat • OtherTeam_player_profile.mat	Passed
			
	The 4 files have been highlighted in the screenshot above.		
7	Click *View Video*	A new window should open as Play_Video UI. (As described in the implementation section)	Passed

			
8	Select *Player Trails* and *Other Team* selection boxes. Then click *Show Graphics*	This should start the overlay of graphics onto the video. Once complete the players tracks should be highlighted by a black line in the video.	Failed
			
	All players expect for one has been tracked through the phase correctly. On the away side number 2 player has crossed paths with the ref and the tracking system has taken the ref as the player. Closeup is below.		

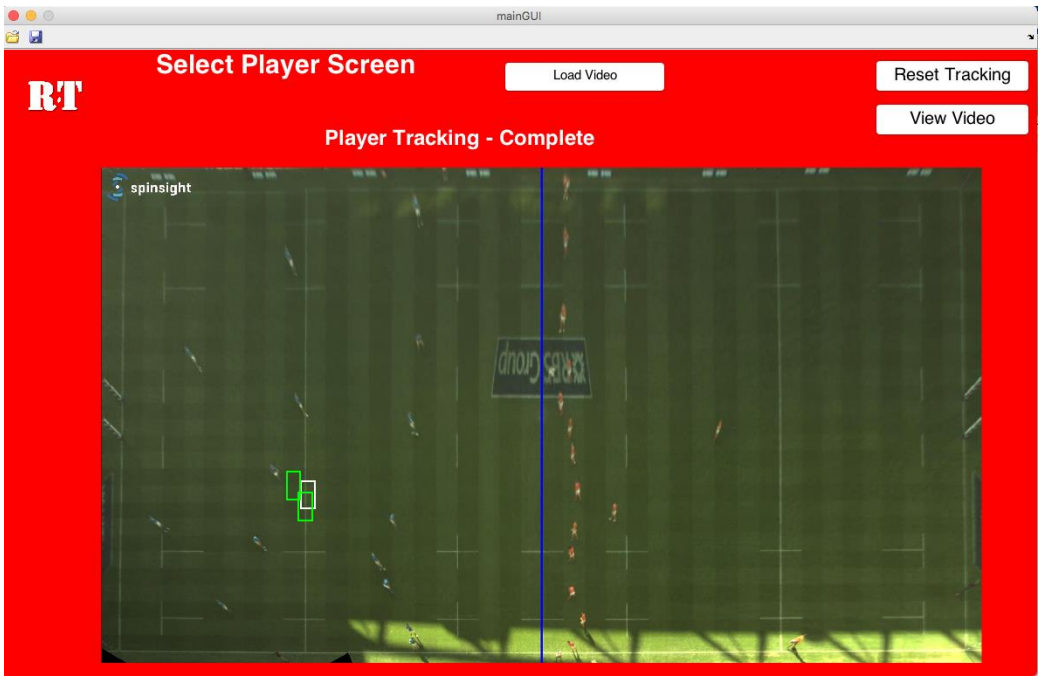
			
9	Load 'player_profile.mat' into the Matlab Workspace for analysis.	Once loaded into Matlab make sure that the data looks good and that there is no values missing. There should be 3 rows for the 3 players selected.	Passed
	player 1x3 struct with 17 fields Fields | boundingbox | X_start | X1 | Y1 | Y_start | X2 | Y2 | x | y | Dis | v_... 1 | [1.0245e+03,7... | 1.0311e... | 1x324 si... | 1x324 si... | 777.5272 | 1x323 si... | 1x323 si... | 324x1 si... | 324x1 si... | 324x1 si... | 324x1 s 2 | [1.1455e+03,5... | 1.1552e... | 1x328 si... | 1x328 si... | 557.4274 | 1x327 si... | 1x327 si... | 328x1 si... | 328x1 si... | 328x1 si... | 328x1 s 3 | [1.2255e+03,3... | 1.2334e... | 1x602 si... | 1x602 si... | 336.8609 | 1x601 si... | 1x601 si... | 602x1 si... | 602x1 si... | 602x1 si... | 602x1 s		
10	Load 'OtherTeam_player_profile.mat' into the Matlab Workspace for analysis.	When loaded into Matlab make sure that the data looks good and that there is no values missing. There should be 3 rows for the 3 players selected.	Passed
	oppPlayer 1x3 struct with 17 fields Fields | boundingbox | X_start | X1 | Y1 | Y_start | X2 | Y2 | x | y | Dis | v_... 1 | [721.5000,740.... | 731.3252 | 1x616 si... | 1x616 si... | 758.3127 | 1x615 si... | 1x615 si... | 616x1 si... | 616x1 si... | 616x1 si... | 616x1 s 2 | [711.5000,647.... | 719.2510 | 1x616 si... | 1x616 si... | 661.3620 | 1x615 si... | 1x615 si... | 616x1 si... | 616x1 si... | 616x1 si... | 616x1 s 3 | [710.5000,596.... | 718.2673 | 1x616 si... | 1x616 si... | 610.1887 | 1x615 si... | 1x615 si... | 616x1 si... | 616x1 si... | 616x1 si... | 616x1 s		

Table 7 - Tracking System Testing

Additional Test:

	Action	Result	Passed / Failed
1	Select and load a phase play video	Video should be selected and the first frame should be loaded into the select_player_GUI window. select_player_GUI should enter State 2 as described in the User Interfaces section	Passed
2	Select 2 positions of empty ground. Click *Next* when selected	Bounding box should highlight the selected area	Passed

3	Select gainline and click *Next*	When selecting the gainline a blue line running parallel to the Y axis should show where the user has selected. (Gainline only has X axis). The UI should ask for away team to be selected.	Passed
4	Select 2 positions of empty ground.	Bounding box should highlight the selected area	Passed
			
5	Click *Next*	The program will run the video processing first. Then the program should highlight to the user that they have not selected a player and stop the process.	Fail
6	Select only one Away team player	A green bounding box should show around selected player. Click *Next* button. An error message should show asking for 2 or more players to be selected. Click *OK* and message box should disappear	Fail




```
>> select_Player_GUI
Starting Mean Operation
Players not selected, Try again!!
Players not selected, Try again!!
>>
```

This has failed as the program has completed the tracking method. It does know that the players had not been selected but it has still gone and used values from a previous track. This can be seen in the screen captures above. The top shows it has used a previous track and the second is the Matlab command window with the error message from the program.

This is a high risk issue as it impacts all parts of the system

Table 8 - Additional Test: Tracking System

Extra Comments:

- As seen in the test above players crossing causes the tracking to sometimes lose tracking of the players. It is an issue that will need to be looked at in future developments. This carries a high risk to the program and can cause issues as it effect the results of the group statistics.

- Another effect seen in the testing was that longer clips seem to impact the reliability of the program. It is understandable that the time taken is longer but at times this seemed to impact the program / code running. This was only an observation and it could not be classified as an actual fault due to the clips were being completed. One of the issues could have been down to the hardware being used and this would have to be

looked at. This runs an unknown risk at the moment as it cannot be defined as an actual problem. It is known that the code was not fully developed with optimisation in mind. This should be addressed in any development on this project

- It is known that the system cannot track players that cannot be clearly defined at the start of the phase. But it has not been identified what the extreme limits are for the program. This could cause an issue as the user will not now this and could select players that cause an issue. This should be looked at in future developments as again it run a high risk to the program.

Overall the tracking method is very useful due to the outputs it gives. The coordinates and statistics of each player means that computing the group statistics is a lot easier as the data is simple and easy to use. The testing has shown that there is still work to do here as there have been vulnerabilities shown. It was known at the start that there would be vulnerabilities but the testing has shown that they carry a high risk to the project. Due to the time frame of the project and the main aims the tracking system hasn't been worked on as much which has shown as a result.

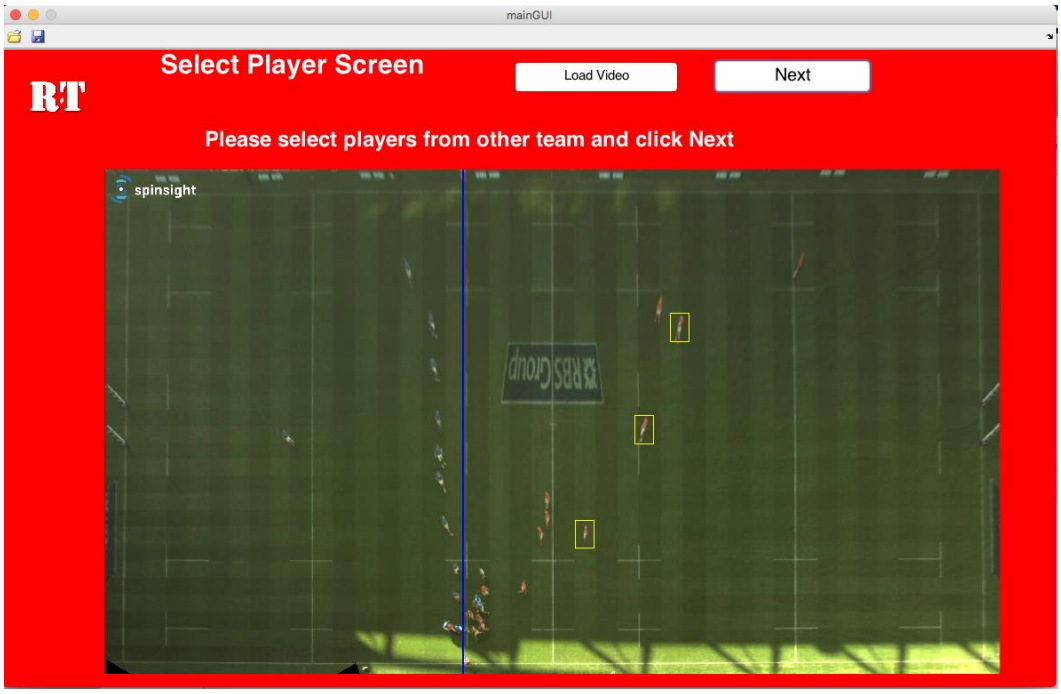
The development of the tracking system could be a project in itself as the number of variables that can effect it and the different methods that can be used are high. A project that looks into the methods and what can affect the tracking running alongside the development of this current project would work well. As improving the tracking will also help the improvement of the group statistics.

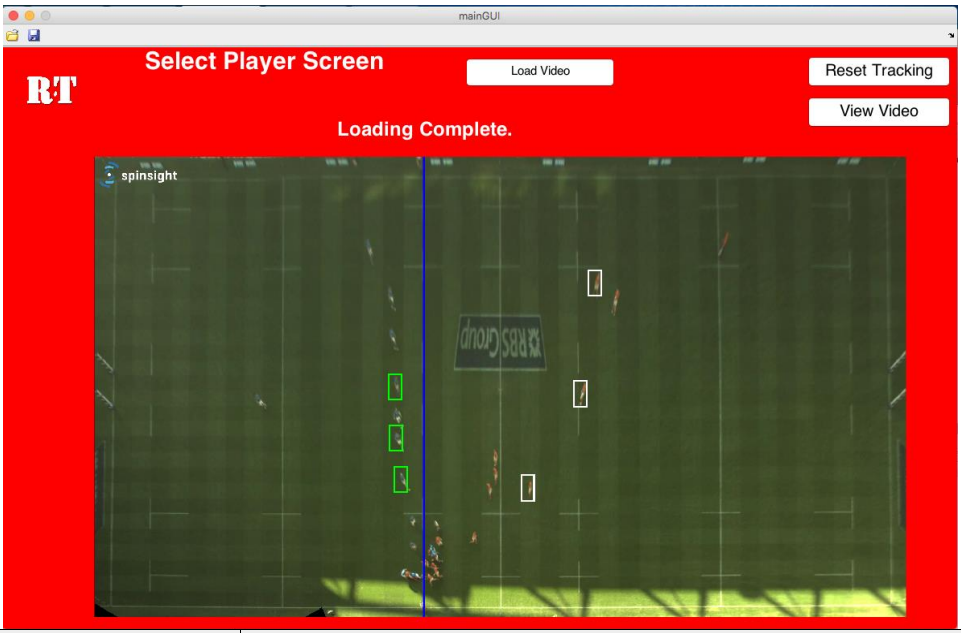
Group Statistics

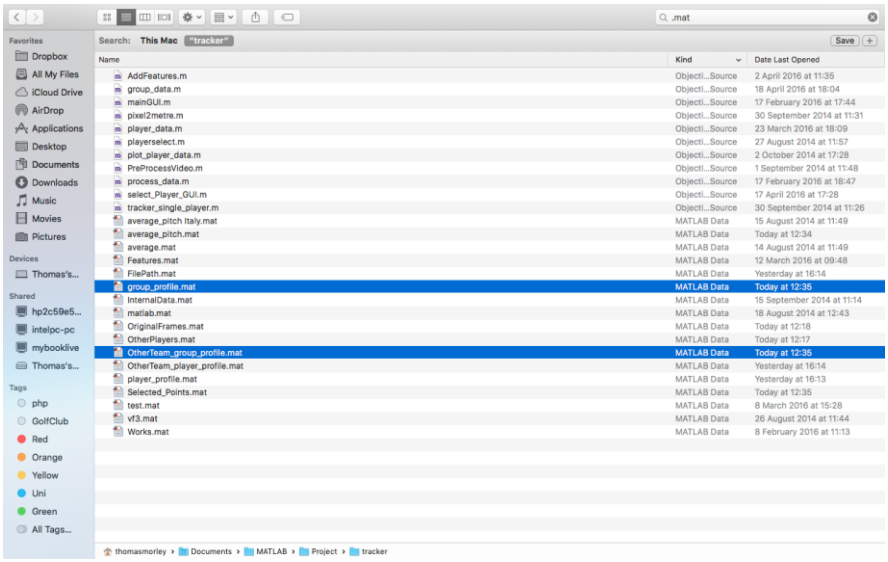
With the group statistics the tests have been run by estimating the result to be shown and comparing it to the angle given by the program. The video is watched throughout to see if any changes in the data does not match the video. The graphics and graphs are then compared to the values stored. Not all clips have been written up but below are examples of the tests run.

The first test just looks to see if the code runs in the correct order and that all files and values are correctly loaded. Then the statistics have been tested separately to check that the values measured are correctly shown in the UI.

Table 9 - Group Statistics Testing

	Action	Result	Passed / Failed
1	Select and load a phase play video	Video should be selected and the first frame should be loaded into the select_player_GUI window. select_player_GUI should enter State 2 as described in the User Interfaces section	Passed
2	Select 3 home team players	Home team players should show with a yellow bounding box around the players selected. Click *Next* - UI should now ask for gainline.	Passed
			
3	Select Gainline and click *Next*	When selecting the gainline a blue line running parallel to the Y axis should show where the user has selected. (Gainline only has X axis). The UI should ask for away team to be selected.	Passed
4	Select 3 away team players	Away team players should show with a green bounding box around the players selected.	Passed

			
5	Click *Next* to complete selection.	Processing of the Video will be run first. Then the tracking method should be completed in the order as displayed below: (This should be visible from the UI text) 1. Tracking Home team players 2. Computing Home Group Statistics 3. Tracking Away team players 4. Computing Away Group Statistics. Once the tracking method has been run the UI should enter state 3 and tracking should be complete	Passed
			
6	Check group files. Open project file location of the project.	In this file should be 4 files as named below all with a size value more than 0. • group_profile.mat • OtherTeam_group_profile.mat	Passed



The 2 files have been highlighted in the screenshot above.

7	Load 'group_profile.mat' into the Matlab Workspace for analysis.	Once loaded into Matlab make sure that the data looks good and that there is no values missing. There should be the same number of rows as frames and all cells should be filled with a value.	Passed

group

1x324 struct with 7 fields

Fields	Back_3	Back_3_Ang	Depth	PlayerDis	x_group	y_group	DisFromGainline
314	611	130	68	[27,3]	-6.8243	-8.4584	13
315	578	126	68	[27,3]	-6.9397	-8.4883	13
316	547	121	68	[27,3]	-7.0564	-8.5002	12
317	547	121	68	[27,3]	-7.0564	-8.5002	12
318	522	117	68	[27,2]	-7.1562	-8.5322	12
319	482	111	68	[27,2]	-7.2830	-8.5994	12
320	445	74	69	[28,2]	-7.3886	-8.6486	12
321	397	86	69	[28,2]	-7.4974	-8.6791	12
322	379	78	69	[28,2]	-7.4992	-8.6007	12
323	342	79	69	[28,1]	-7.6885	-8.5211	12
324	293	75	69	[28,1]	-7.7980	-8.4569	12

8	Load 'OtherTeam_player_profile.mat' into the Matlab Workspace for analysis.	When loaded into Matlab make sure that the data looks good and that there is no values missing. There should be the same number of rows as frames and all cells should be filled with a value.	Passed

Fields	Back_3	Back_3_Ang	Depth	PlayerDis	x_group	y_group	DisFromGainline
1	14	174	84	[3,5]	0	0	-3
2	14	174	84	[3,5]	0	0	-3
3	15	173	84	[3,5]	-0.0154	0.0162	-3
4	16	173	84	[3,5]	-0.0327	0.0329	-3
5	19	172	84	[3,5]	-0.0617	0.0450	-3
6	20	171	84	[3,5]	-0.0958	0.0516	-3
7	20	171	84	[3,5]	-0.0958	0.0516	-3
8	19	171	85	[3,5]	-0.1344	0.0652	-3
9	19	171	85	[3,5]	-0.1741	0.0898	-3
10	20	171	85	[3,5]	-0.2168	0.1214	-3

Below is the clip being used to show the values working. The frame capture shows the player selected and the gainline location. In most phase plays all statistics can be tested. The exception in most cases is the back 3 angle where it is used on mainly defence plays.

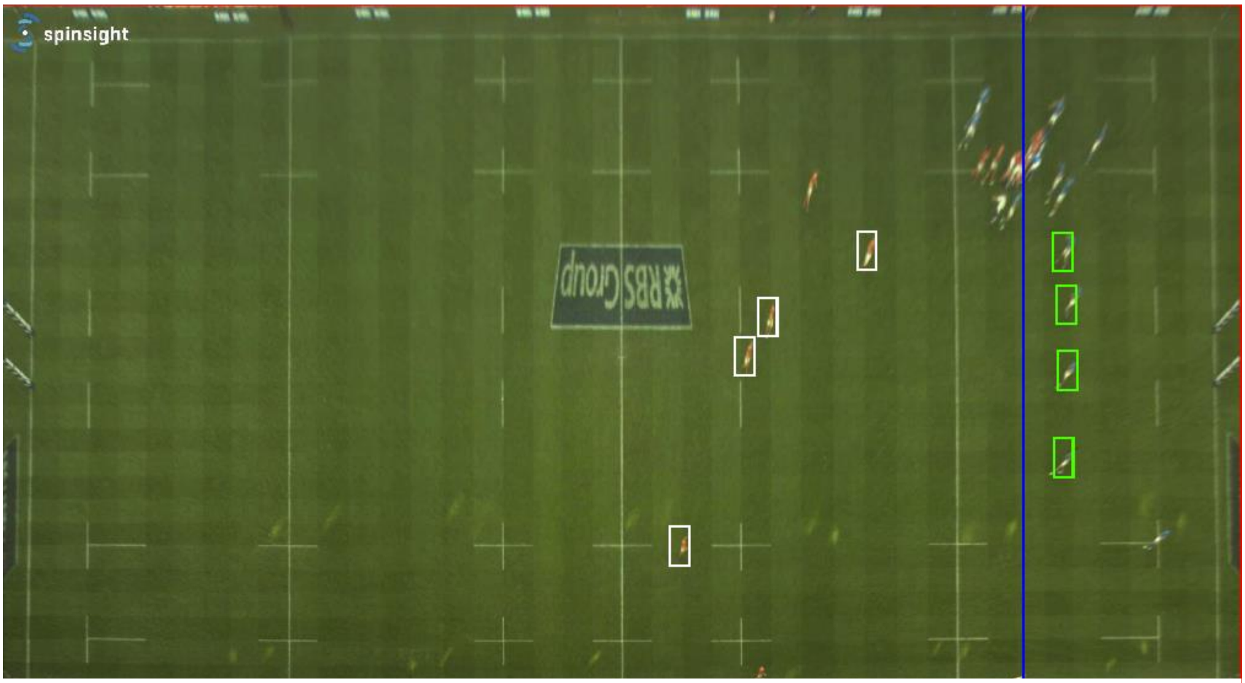


Figure 46- Clip 1 for testing

(Angle) Depth of line

Here the for the welsh team the angle to be measured should start around 55 - 70. While the Italian team should be around the 80 - 90 degree mark.

Below are the first 78 values that are stored for both the Welsh depth of line and Italian depth of line. That shows the values are in the range wanted.

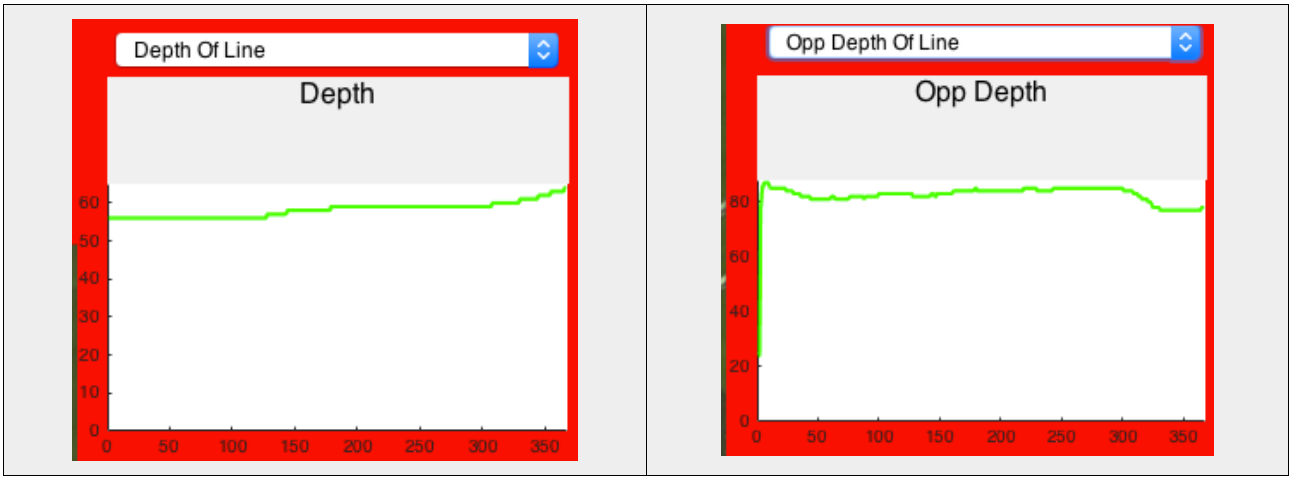
Welsh

56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56
56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56
56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56 56

Italian

24 24 77 84 86 87 87 87 87 86 85 85 85 85 85 85 85 85 85 85 85 85 85 84 84 84
84 84 84 83 83 83 83 83 83 82 82 82 82 82 82 82 82 81 81 81 81 81 81 81 81 81
81 81 81 81 81 81 81 81 82 82 82 81 81 81 81 81 81 81 81 81 81 81 81 82 82 82

Wales depth of line	**Italy depth of line**



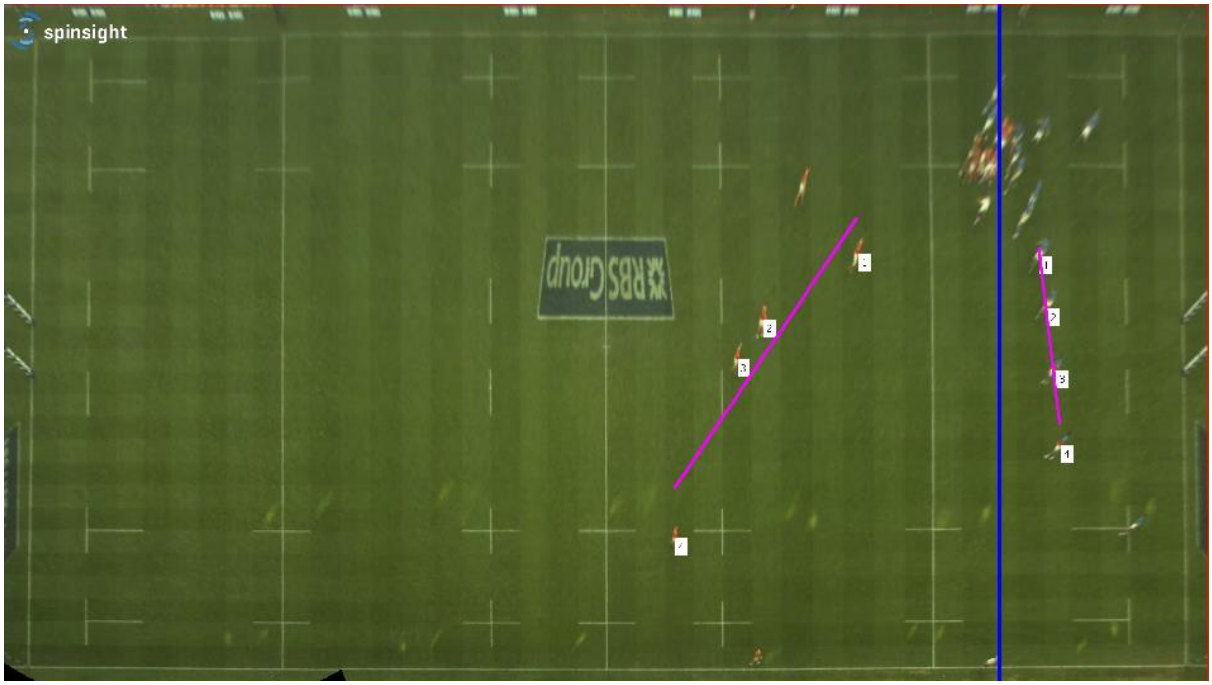


Figure 47- Test Depth of line

From analysing the values and the graphs they work correctly and clearly show the correct results. Also the graphs help show the angle clearly on the video and mimic the values collected.

Overall this is a very sound statistic but there is an issue that has been found. When the depth of line moves over 90 then the angle swaps to measure the opposite angle. This is how is it designed but it causes the values for the depth of line to flip. (This can been seen on the graph above for the Italian defence in the first couple of frames). This could possibly be changed as it does not look right when looking at the data. It runs a low risk to the program as it will not hit the performance.

Distance from Gainline

Using the same phase the line will be a certain distance from the gainline. The Welsh time is about 20 - 30 meters away from the gainline. Where the Italian team is close to about 5 meters away from the gainline

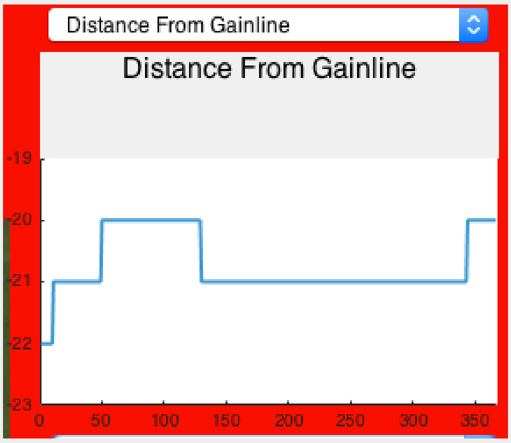
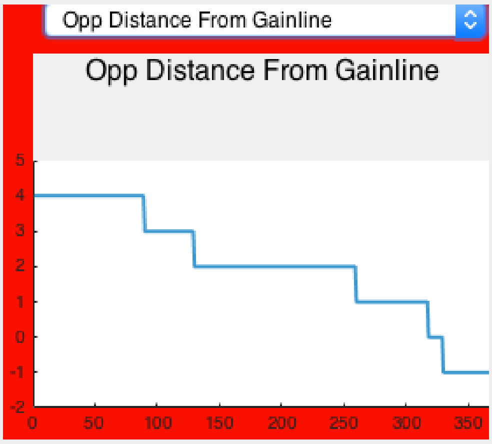
Below are the first 78 values that are stored for both the Welsh and Italian distance from the gainline. That shows the values are in the range wanted.

Wales

-22 -22 -22 -22 -22 -22 -22 -22 -22 -22 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -21 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20 -20

Italy

4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Wales Distance from gainline	**Italy Distance from gainline**
	

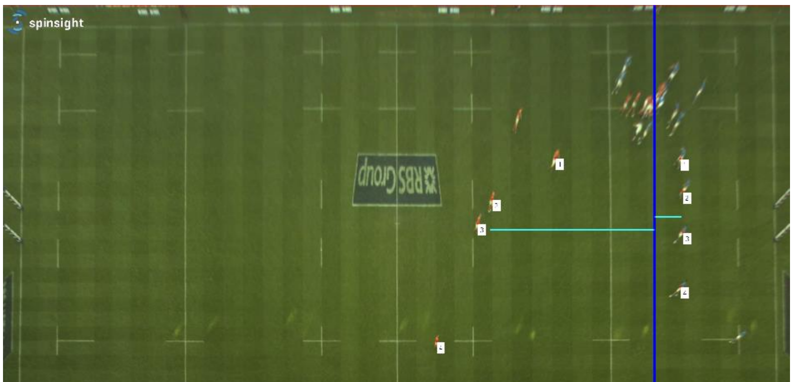


Figure 48- Test Distance from gainline

Distance between players

In the video the players have been numbered below are the expected values from the video.

Wales 1-2 = 5m - 10m 2-3 < 5m 3-4 = 15m - 20m

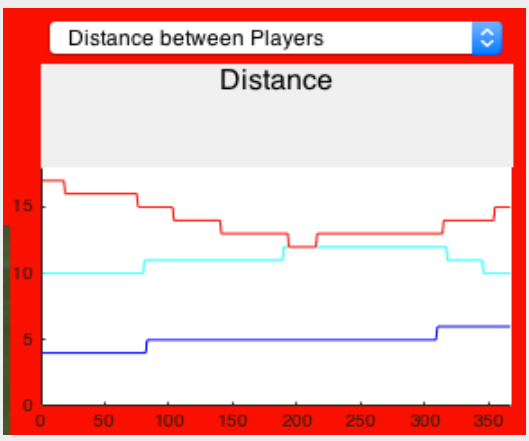
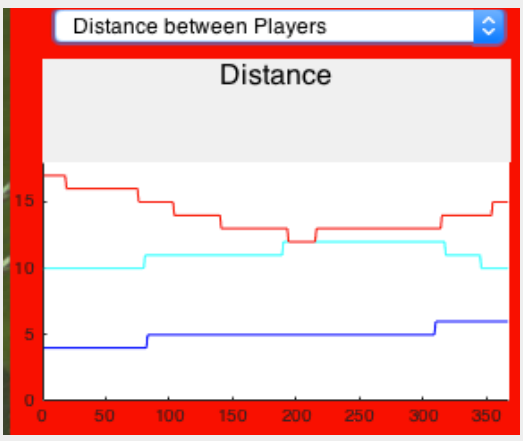
Italy 1-2 <= 5m 2-3 <= 5m 3-4 = 5m - 10m

Wales

[10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17]
[10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17] [10,4,17]
[10,4,17] [10,4,17] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]
[10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]
[10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]
[10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]
[10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]
[10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16] [10,4,16]

Italy

[4,6,7] [4,6,7] [4,6,7] [4,6,7] [5,6,7] [5,6,7] [5,6,7] [5,6,7] [4,6,7] [4,6,7] [4,6,7]
[4,6,7] [4,6,7] [4,6,7] [4,6,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7]
[5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,7] [5,5,6] [5,5,6] [5,5,6] [5,5,6]
[5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6]
[5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6]
[5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,6] [5,5,5] [5,5,5] [5,5,5] [5,5,5]

Wales Distance between Players	Italy Distance between Players
	

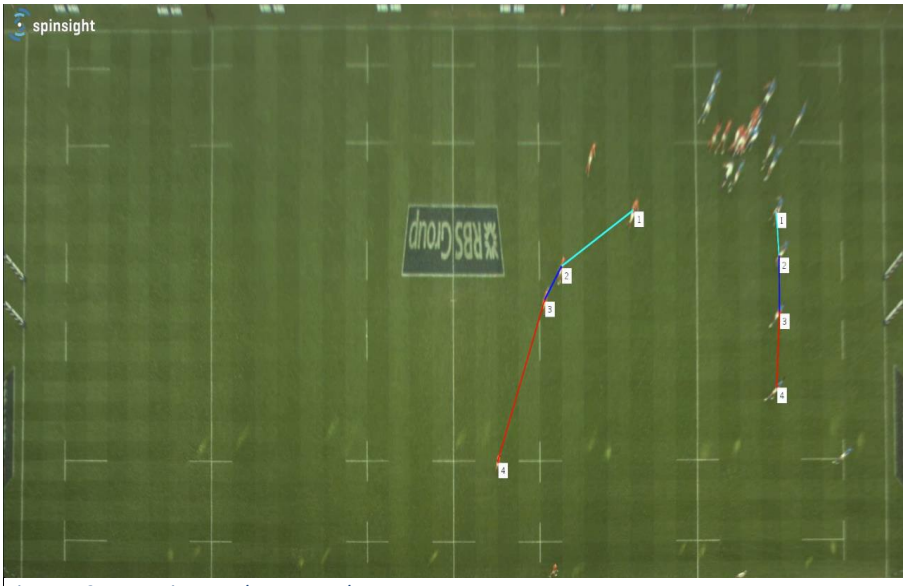


Figure 49- Test Distance between Players

Group Displacement

Here the group displacement should always start at 0 as it is relative to the start distance. In the video the teams end displacement has been measured. For the welsh team the displacement is less than 5m. For Italy it's about 5m.

Wales

First 25

0 0 0.0392 0.0765 0.1123 0.1456 0.1456 0.1792 0.2115 0.2509 0.2894 0.2894 0.3336 0.3736 0.4134 0.4502 0.4502 0.4810 0.5100 0.5333 0.5590 0.5590 0.5859 0.6223 0.6586

Last 25

1.2254 1.2498 1.2751 1.2928 1.3069 1.3069 1.3324 1.3573 1.3849 1.4126 1.4126 1.4425 1.4672 1.4907 1.5065 1.5065 1.5180 1.5313 1.5440 1.5466 1.5466 1.5507 1.5483 1.5458 1.5447 1.5447

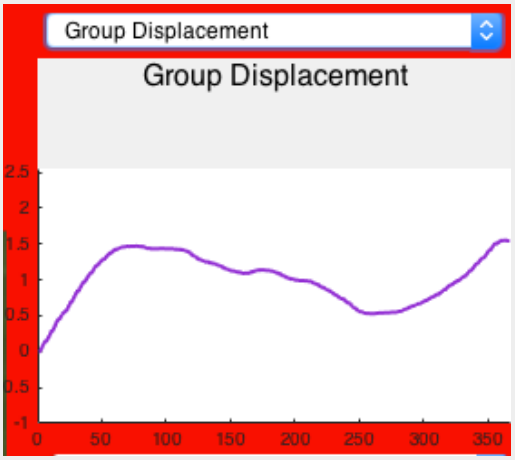
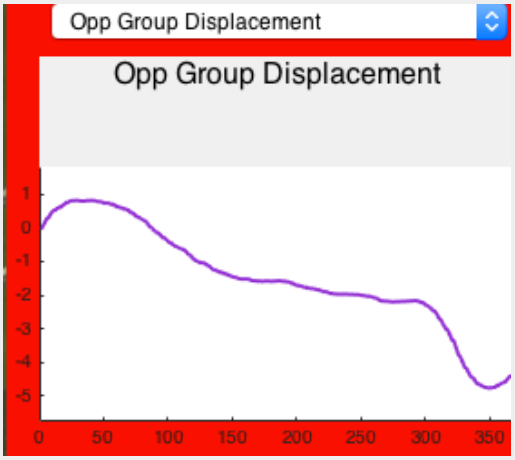
Italy

First 25

0 0 0.0689 0.1341 0.2116 0.2892 0.2892 0.3593 0.4197 0.4751 0.5193 0.5193 0.5564 0.5810 0.5988 0.6219 0.6219 0.6585 0.6922 0.7223 0.7545 0.7545 0.7796 0.7980 0.8077

Last 25

-4.6460 -4.6757 -4.6950 -4.7183 -4.7379 -4.7379 -4.7529 -4.7598 -4.7575 -4.7522 -4.7522 -4.7448 -4.7353 -4.7118 -4.6757 -4.6757 -4.6466 -4.6243 -4.6131 -4.5869 -4.5869 -4.5349 -4.4886 -4.4438 -4.4057 -4.4057

Wales Group Displacement	**Italy Group Displacement**
	

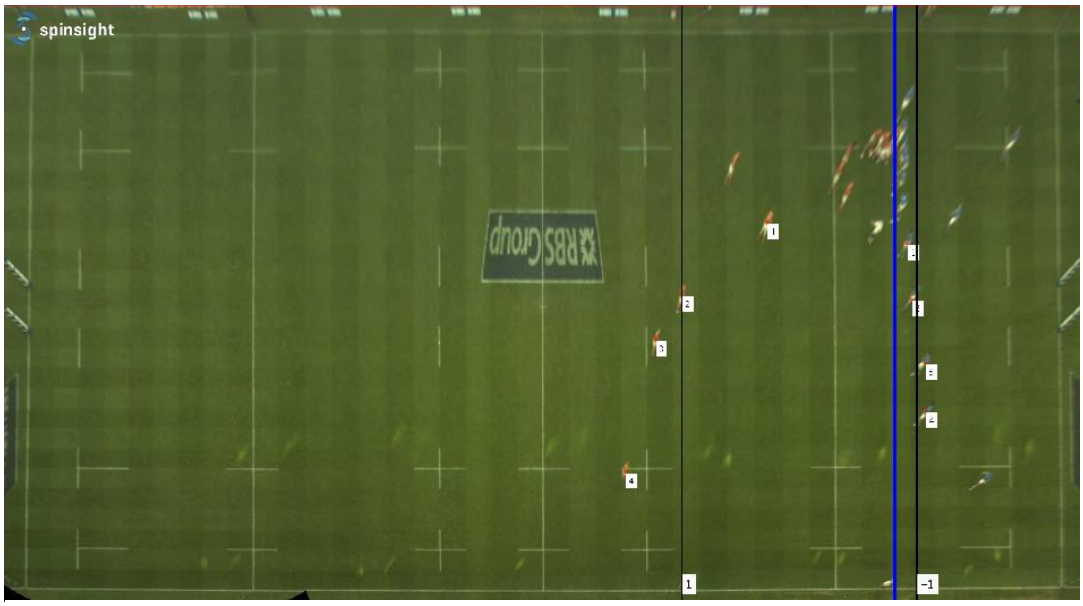


Figure 50 - Test Group Displacement

Tracking Back 3

The back 3 cannot use the same video as they don't have the formation needed to test the measure for this a new phase play is used a frame capture to show the players selected and the gainline position. In the frame capture Italy has not been selected as they do not have a formation that allows the measurement to be tested. In this frame the angle created is about 90 - 100 degrees.

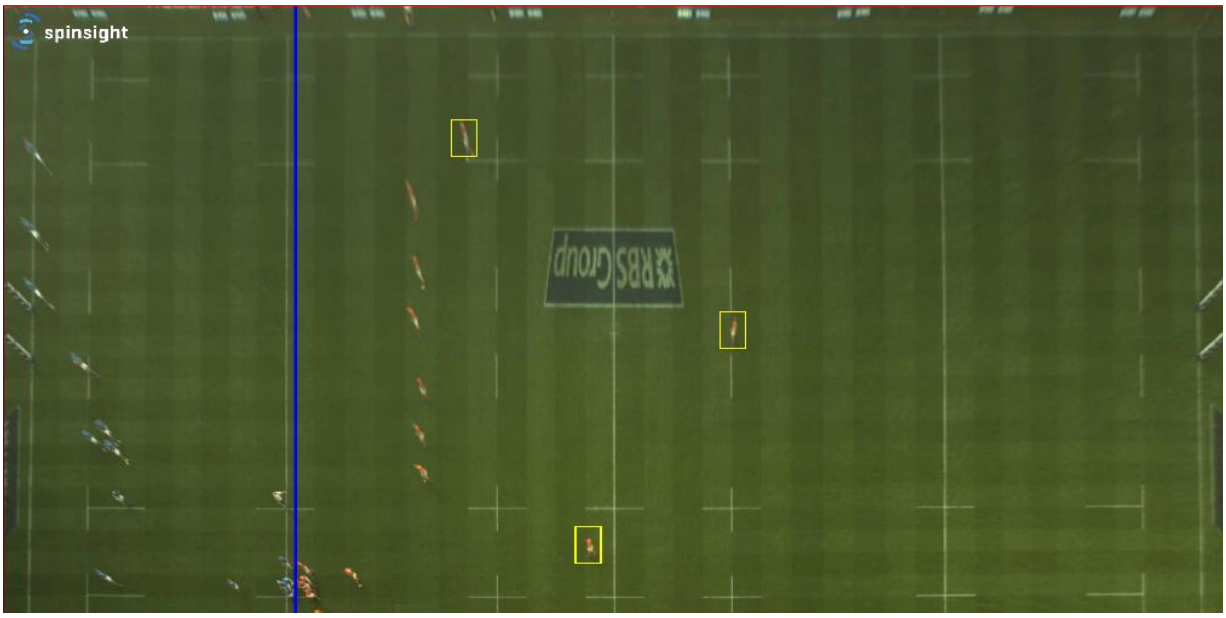
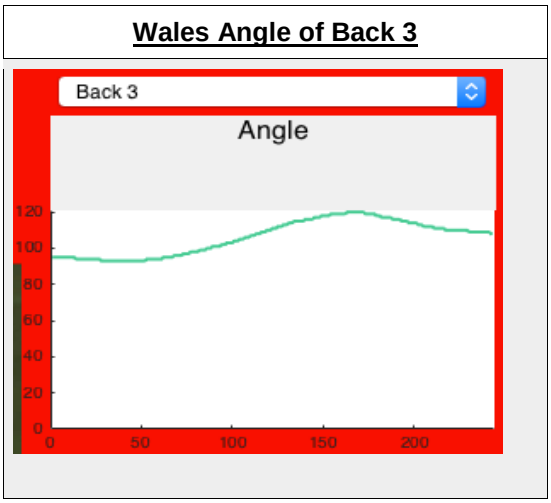


Figure 51 - Clip 2 Testing

Wales

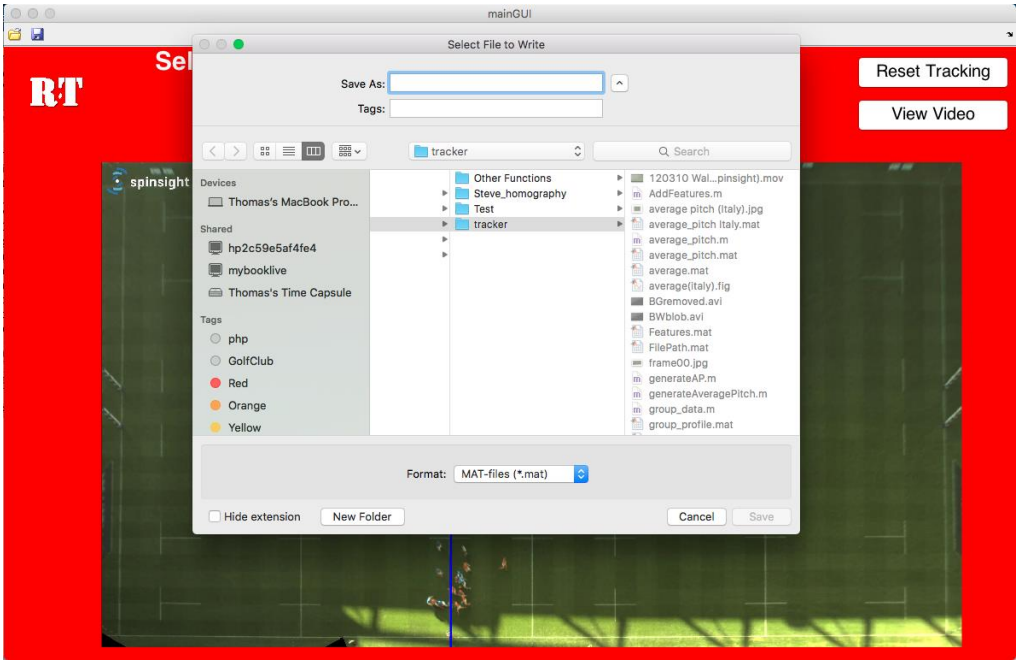
95 95 95 95 95 95 95 95 95 95 95 95 95 94 94 94 94 94 94 94 94 94 94 94 94 94
94 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93 93
94 94 94 94 94 94 94 94 94 95 95 95 95 95 95 96 96 96 96 96 97 97 97 97 98 98

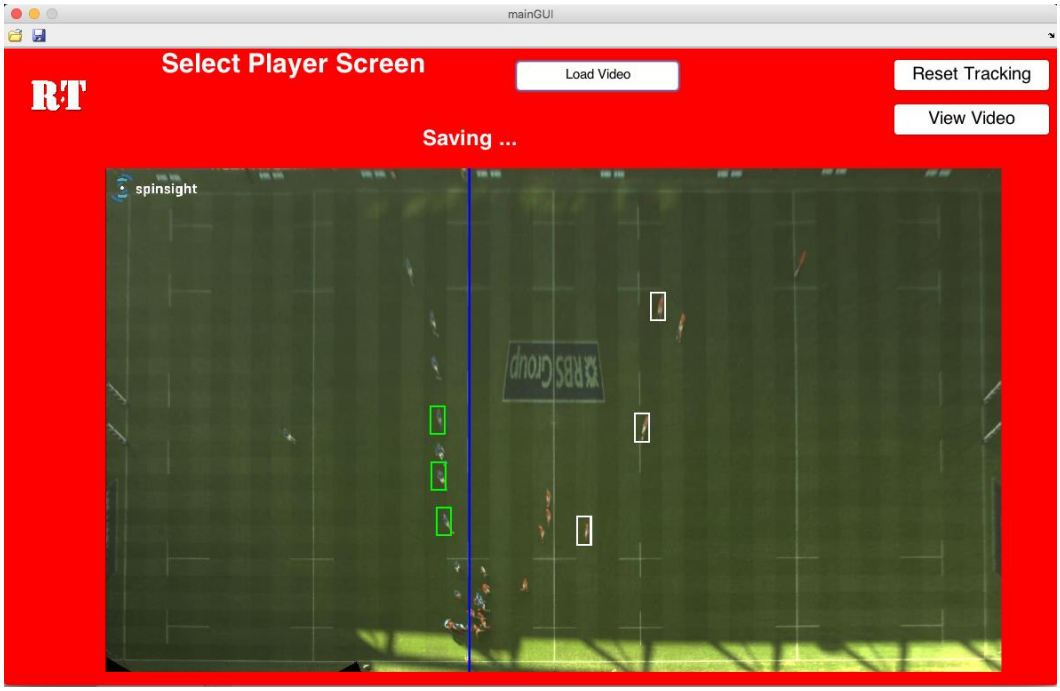


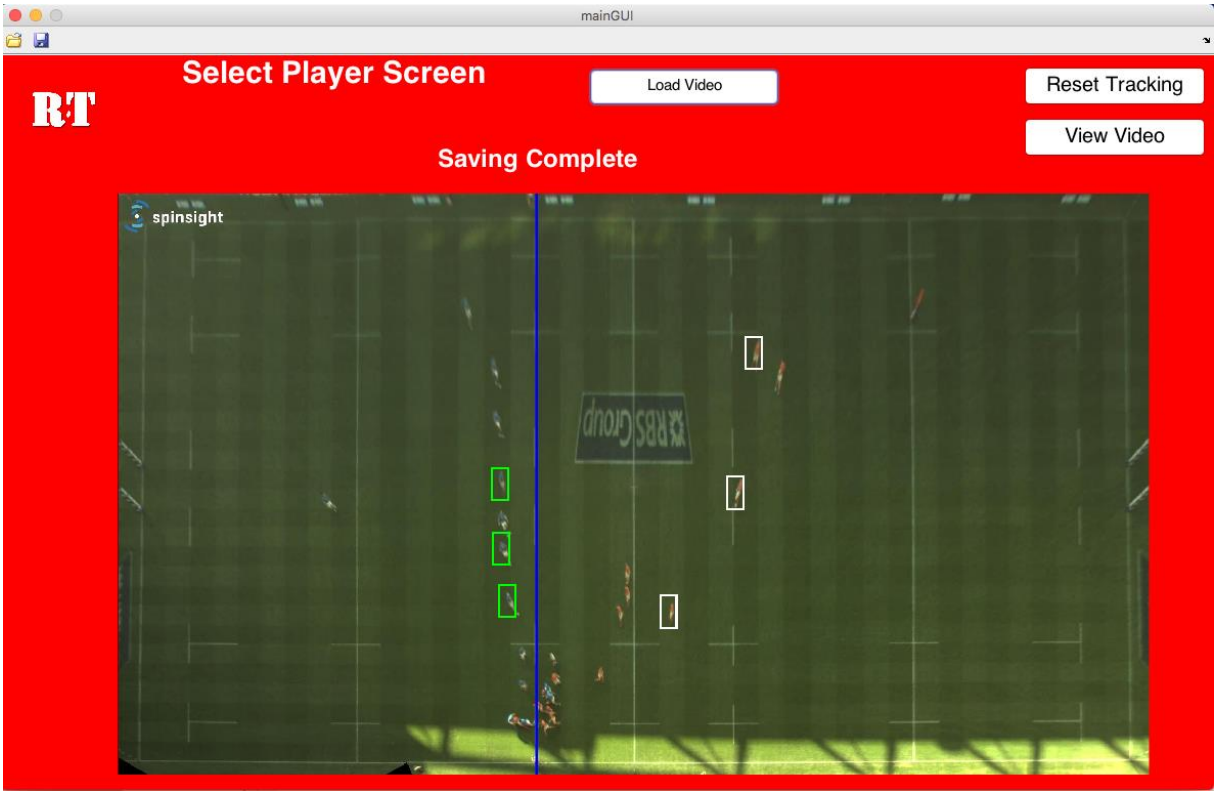
Save / Load File

This looks at the saving and loading of a file that is created by the program so the tracking method does not have to be run twice. This test carries on from the testing done on the re-tracking method.

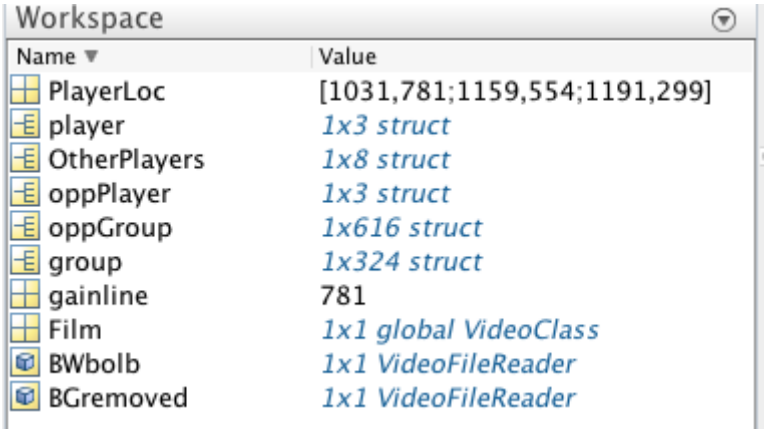
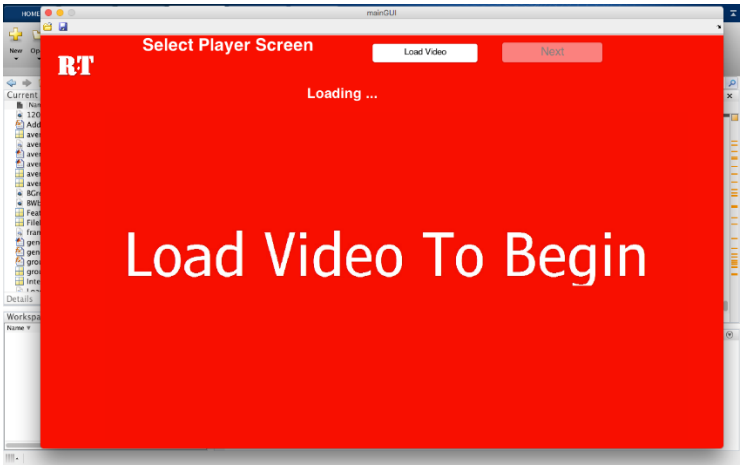
Table 10 - Save / Load Testing

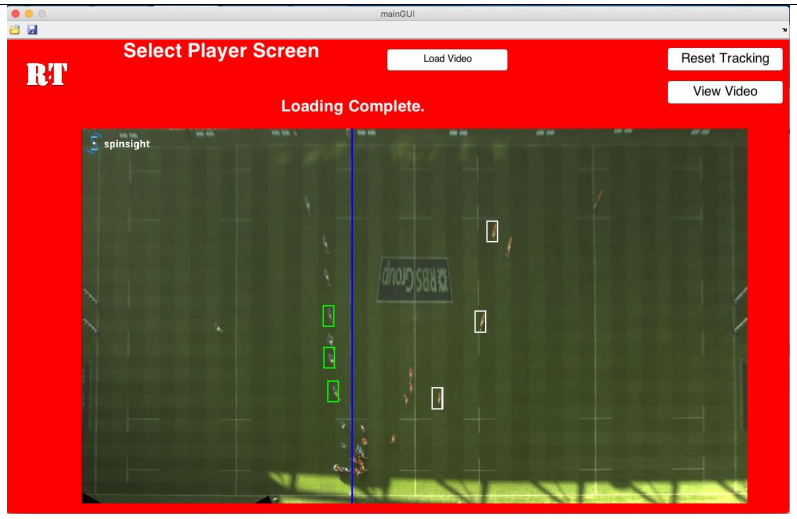
	Action	Result	Passed / Failed
1	With the select_player_GUI is state 3. Click the *Save* icon in the menu.	A save file window should display to enter the name and select a location of the file.	Passed
			
2	Save a file with a name 'SaveFile' on the desktop	The program should now highlight to the user that the files are being saved on the select_player_GUI UI. Each stage should be highlighted.	Passed





3	Once the file has completed. Load the 'SaveFile.mat' into the MatLab workspace.	The files that should be loaded in the Matlab workspace are listed in the implementation section.	Passed

			
4	Close the program	Program should close	Passed
5	Open new select_player_GUI and click the *Load* icon in the menu bar	A load file window should appear to allow the user to select a .MAT file.	Passed
6	Load the file ‘SaveFile.mat’.	The UI should now highlight to the user that the files are loading. The UI will then enter state 3 with all the correct players and gainline selected as saved before the file was saved.	Passed
			



Extra Comments

- The loading and saving of files is quick except for the Video Class. This is where most of the saving and loading times are spent. Don't know how this can be edited but for the user it would be good if the time to save and load files are reduced.
- If there is a file missing then the save function will fail and Matlab will produce and error of file not found. This means the save function will not work. This runs a low risk to the program as it only stops the saving of a tracked phase. But if a file is missing it means that there is an issue with the tracking itself.
- If the load file is missing a data structure then it will not stop the loading process. This is due to old files being available. This runs a high risk for the program. It does not highlight the error and if it does occur then the tracking values will not correlate. This will have to be addressed in future developments.

Overall the save and load feature works and is a feature that is vital in a system like this. It saves the correct files as it allows the program to save time and not re-track a play for a second time. Although it works well it is open to some additional implementations to make the feature robust. Firstly it does not check for files that are missing in both the save and load functions. This would need to be fixed as it is a medium to high risk to the program. Also the time taken by the program does have an effect as it reduces the program's effectiveness.

Summarising Tests

Overall the tests have shown that the group statistics and very solid in the results they return. All aspects of the group statistics are very reliable and consistent. The graphs are correctly displaying the values and the graphics are mimicking the data correctly when being added to the video. This is good as the project focus was heavily on these statistics which shows the time invested in them was good.

The testing has also showed that the additional features that were implemented have a very good bases and that them when running normally don't see an issue. The problems arise when the user does something that is not in the normal operation. This is understandable as the focus was not fully on these features so they have been developed to handle exceptions outside of their normal operations. From the testing it has shown what needs to be addressed with the features and in future developments these can be addressed.

The main point raised from the testing is the tracking method could be improved. The systems bedrock is the tracking and if that is slightly off then it effects the overall results of the group statistics. The tracking outputs a very simple and organised data set that is useful for the statistics when computing. This is good for reasoning with when testing and this should be something that is kept in the future. As stated above the tacking system could in itself be a project and the issues with regards to the swapping of players must be addressed as this is an issue that carries a major risk to the project.

Finally the testing has been constant throughout the program and the standard program is very solid and it does what the aims wanted at the start of the project. If more time was available then the issues with the features would be addressed first as it effects the overall experience of the project and the user experience.

User Evaluation

At the end of the implementation the program was shown to the coaches of Aberavon Harlequins RFC. This was to get an idea of how they would react to a program that could help with analysing phase plays. The issue with the group of users used is that at the amateur level of rugby technology is not used in the

development of a team. Generally this is down to money and the fact that the level of detail these systems produce is not needed. The idea of showing them is that if they could see a potential in the program then it shows that the project has produced results that could be expanded and refined in future builds.

A quick presentation and demo was used to show the program and the results that could be produced. Then a general discussion allowed for any comments and questions that they had about the project. Below are the results from these meetings.

The program was spilt into areas of development and during the talks each area was discussed to get the users options and comments. Each development area was then given a rating.

User Interface	Average
Usability	Poor
Graphics	Good
Graphs	Good
Group Statistics	Very Good
Implemented Program Features	Good
General Impressions	Very Good
Can you see it being Used	Yes

Overall the response from the coaches was very positive. Throughout the project they have been asked questions and kept up to date with what has happened but the end product is the first chance for them to look. What was interesting is that it had to be suggested how the program could be used in certain situations before they started to think how they would use it. This is understandable as the program does only work for phase plays and systems like this are not used at amateur levels. Once this link was established then the evaluation worked better.

The main point raised by these talks was that the usability of the program would have to be improved along with the design of the UI so that it would be used by them. This was expected as this had very little focus and it was raised during the testing. This shows that if it was to be released then further development will be needed before product could enter the market. For the

users this was the biggest block for a system like this being used. Also a comment was raised on how only short phase plays may cause people to get annoyed with the system. Once the UI and usability was passed they started to see the benefits the statistics can have. A point raised was this project looks to be focused slightly more towards defence. This comment is understandable as the group statistics at the moment are very basic and attacking plays tend to be slightly more complex within the phase. This could be addressed in the future by looking into more complex measurements for additional runners. The question that was constantly asked was the linking of the GPS tags to the program. It was highlighted to them that this used a computer vision system and a future implementation could use the data. The point that was liked by the coaches is that the program allowed the user to also gain information on the opposition team within the same phase where the GPS tags wouldn't give you this information.

Program implementations like the comparison of two players and the re-tracking method was explained. The response from the users was limited due to their knowledge of programming but they understood the idea that improving the methods will help to improve the usability and when showing how it helped improve the system it changed their overall look on the program. The main evaluation wanted from the users was on the group statistics. Having good measurements was one of the main aims set at the start of the project. When giving the presentation phases were shown with the group statistics calculated. This is where the best responses were given. The coaches were positive that the measurements helped clearly describe the plays and that they did show clearly what was happening. During the demonstration the graphics was where the coach's attention was mainly focused. This showed that the users would tend to use the video with graphics more to relay the information back to people. Although they found the graphs helpful in backing up what was seen. What this shows is that future statistics should look at how they can be developed and shown in graphical terms as this is where the user will spend most of their time.

Overall the user's evaluation was very positive and when asked at the end if they see this as something that could be used in the future all said yes if the developments proposed were to be introduced and updates were made. A meeting with a representative from the WRU was organised as this would have

been more like the target audience that would use a system like this as they are the top end of the professional ladder in Wales. But due to other commitments this did not occur. Although this did not happen in time the response from the coaches has proven that the statistics measured do help define plays and that the one of the main aims have been hit.

Conclusions

This section looks at the overall result from the project and evaluates it to the aims set at the start of the project. Below are the aims that were set. For descriptions please see the introduction section.

Main Aims

- Understand what analytical features are useful to the target audience
- Produce a working program that allows a person to run multiple phases of play and retrieve tangible data on the events that occurred.
- Produce a very basic and usable UI.

Secondary Aims

- Develop more advance analytical features
- Linking tag data to tracking.
- The program will only work on clips that have been cut into phases if the program could have a full clip run through it and results produced it would remove the tedious task of clipping the video.

Overall the results from the project have been really promising and the statistics that have been measured show that there is some reasoning to be able to produce a program that will help the analysis teams. At the start of the program it was unclear what could be measured and if there was a reason for researching in this area of sports analysis. Now it is clear that that it could help the analysis teams and that it is possible to measure the plays with statistics that make a difference.

The main objectives have been achieved for this project. As stated they were left a little vague due to the unknowns involved but as the project has

progressed it has become clear where the project can progress to. The first aim was to understand what could be measured and if they were useful. This has been achieved as there are clear measurements that when combined can clearly describe a phase play. Also from the research it has given extra measurements that can be looked at in future developments. The second aim was to produce a working program that allowed for multiple phase plays to be run and data to be retrieved. Again this has been met as shown in the implementation section where the program is described. The document outlines the steps needed to be run to produce a full analysis of a phase play. It explains it in a way that if it was to be produced in another project parts could be copied from this report. During the project the tracking system worked more as a black box to the measuring of the statistics. This means that the measurement of statistics could be taken and used on a new tracking method as long as the output from the new tracking method uses the coordinates system that is currently used. This flexibly means that it is useable in future developments.

The final main aim was to produce a very basic and usable UI. This has been achieved to some extent. The UI is very basic and is usable to be able to review and produce the analysis of a phase play. The area where it falls is when given to the target audience. What can be said is that it is usable but is not user friendly for the user as it is not clear what can be done. Although this is a down point this was not the aim of the project to produce a user friendly UI. The UI was developed over time and changes were made when statistics needed to be viewed in a certain way. If more time was available this could be worked on as it is now clear which is the best way to show the information and the users usability measure could be considered.

Secondary aims were always there to make sure that the program could be developed in the future. Although tangible results have not been made for these aims there is some progress towards them in this project. Where research has led to some insights. The first aim is the most researched area and from the research this should become a main aim in a future development. The project has shown that there is a case for more complex moves to be broken down and analysed. The second secondary aim was to look at linking the GPS tag data to the tracking. From the project research this aim was unattainable in the time scales available. Although the project has

shown that it may be possible and some useful features could be made it would not be an aim in future developments. The final aim was to move from clips and work on longer videos. This has not been looked at as a viable option as the research was needed in other areas. Again with this aim it was not needed as the main reason for the project was to develop measurements. From the research the possibility of this occurring is small as the game changes rapidly and it would be taking measurements that would be pointless for the analytical teams. (e.g. when ref blows his whistle to stop play.) Therefore this aim would be removed in future projects.

To summarise the project has given a clear insight as to what is possible in terms of measuring statistics. It is clear as to what direction the project could move in and the aims of the project could be defined more as the understanding is better. Finally, the deliverables have been met and the program gives a result that shows the statistics working correctly. From user feedback it still needs work to become a better system but shows the potential is there for this type of system.

Future Work

This section looks at the future work that could be done to develop the program further and address the comments and issues found in the Results and evaluation section.

As stated in the conclusion of the project the direction of the project has meant that there are a number of possibilities for future work on this project. Below are some of the development areas that could be addressed next.

Live Analysis

This is certainly an area that can be investigated. The research into being able to do it has shown it is possible. From the results and evaluation of the project, comments from testers have asked if this could be run live for quick updates during the game. This would then allow the analysis teams to get the results straight down to the team where the impact can be made instantly.

Although this is a really desirable feature to have it would probably be used only a few times during a season of games. The area it will be mainly used is for

the team analysis sessions during a training week. In these sessions they have time to look over the finite details of phases and what has happened. Therefore, although it is a desirable feature it would be best to develop more complex measurements as the system would see more of a benefit.

Graphic (Feature) Overlays

The live analysis is effected by the graphics not being overlaid in time for the video to be able to play at the correct speed. Although this will help the live analysis it would also help the overall usability of the program. At the moment the user has to wait while the program edits the video with the graphic overlays. If this time could be removed / reduced it would allow for the showing of the analysis results live.

Graph Plotting

As with the graphic overlay it effected the play back of the video when plotting live data. This would have to be addressed if live analysis was to be used. The reason for the video frame rate dropping is not fully understood and this would have to be researched before the live graph plotting could be become usable.

Tracking Multiple Players (At the same time)

This is the biggest improvement that could be made to the current system. The tracking system currently tracks each player separately. If they could be tracked at the same time then this would dramatically improve the speed at which the tracking is completed. As the group statistics could also be completed as all the data needed for the measurements are available. The time saved here would make the live analysis much easier as this is where the heavy processing is done in the program.

Better UI design for Future use

The project focused on what statistics could be gathered from the video footage and the UI was developed after the statistics to help show the statistics as a concept. This means that if it was to be taken by an analysis team it would have to be developed with more usability in mind.

As stated the aim was to create a usable UI which has been the case and this has also shown in the reflection from testers. When showing the UI they found it hard to follow. This could be down to not understanding what a program like this could do for them. This may involve training to help with this transition with a program like this.

All of the above would have to be considered when looking into this area of work and for it to be used in the field it would have to be changed from its current state.

Improved Tracking System

When first looking into the project there are systems that use the GPS tags to help identify and track players. This could be something that is implemented into this program to help develop and improve the tracking. The main issue is that as an analysis team would have is that they would not have the data from the opposition for obvious reasons. The benefits from a link to this could be helpful in creating more complex features and measurements. For example if a player is meant to be in a certain location then knowing which player he is could allow us to detect if they are in that position.

Linking a tracking system to a GPS tag would be good if it is possible but there is still work for the vision method used now. During some of the phase plays some of the players will cross. During these phases players can be lost or their tracking is swapped. This is an issue that has a heavy risk factor and is an area of work that would be high on the list of possible threats to the system. If more time was available this would be the area of research that would be high on the priority factor.

Overall the tracking system works well now but a more reliable and advanced tracking system would befit other areas of work as it is the bases that the program is built on.

More Complex measurements

The statistics currently only cover very structured and simple shapes. So this area of research starts to look at more complex shapes that can start to measure more complex plays.

For example, in defence a play that is run is an umbrella defence. This means the outside players speed up and the line becomes a curve to force the attacking team back towards the breakdown and not go out wide. As shown in the image below. The measurement of depth of line currently works but is not the best measurement for this. A measurement on the curve would be better.

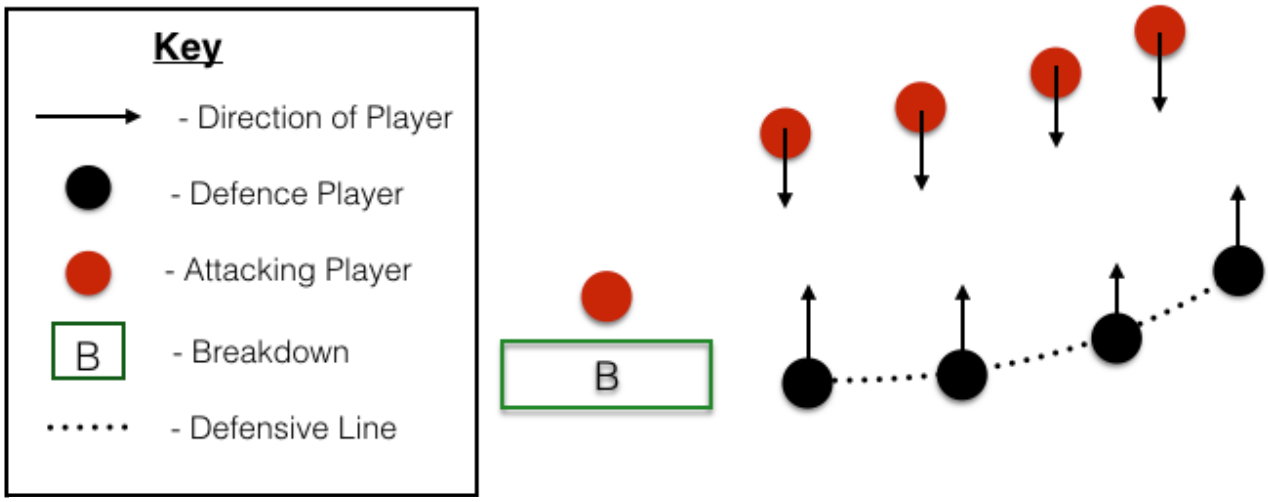
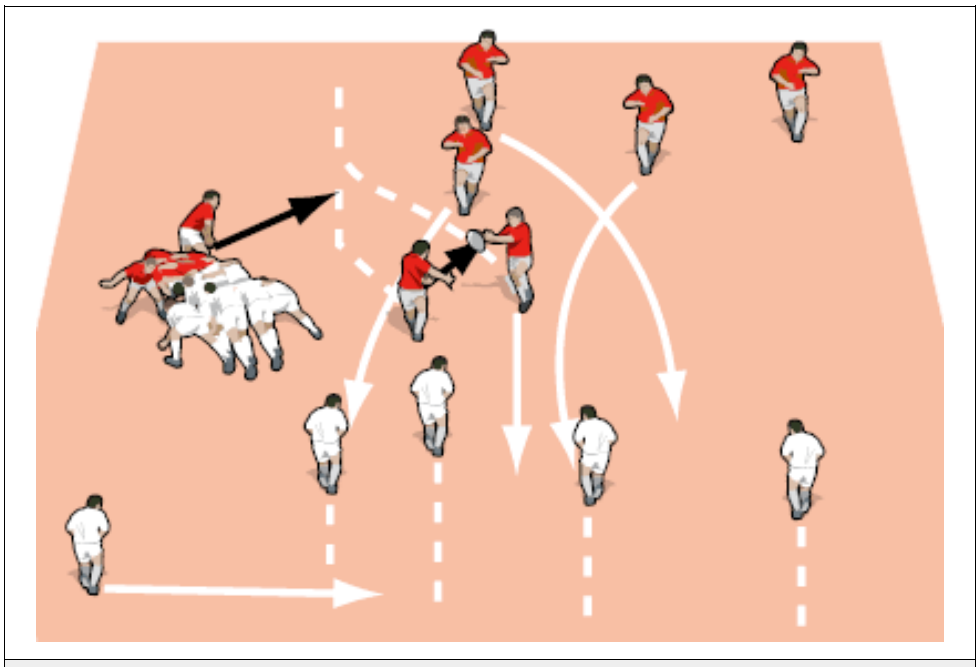


Figure 52 - Complex Plays

Attacking moves also have players looping and running angles. If a measurement could be used to detect this then the program would start to be more useful for attacking plays than it currently is.



Source:
http://www.betterrugbycoaching.com/Article-772-77-Rugby-Tactics,-Strategies,-Tips-How-to-trick-opponents-with-a-deceptive-ploy#

Figure 53 - Running Angles

This is certainly an interesting area of research as it looks to improve on the statistics found here and if an improved tracking system is produced then it would open the possibility of the measurements that could be found.

Machine Decision Making (learning)

The final area that I see as a future development is for the program to start making decisions on when plays have not been run properly. Throughout this report it has talked about the structured plays and these have parameters where the play will break down. If the program could start to highlight when plays have broken down then it may help the analysis teams in their work.

For example, in a blitz defence the line will stay flat and the gaps between people will stay roughly the same. If a player was to slow or to step in then a gap would open up. This could be detected by the distance between players decreasing to a value that is too small. This could then be highlighted to the user. This feature would only assist the user as during phases players will make decisions as the situation will change. Here the user will have to then decide if it was the right decision but highlighting it saves the user time from scanning each play.

The other improvement would be for the program to start learning patterns and maybe try to understand what play is being made as it could compare measurements of previous plays.

Overview on Future Work

The development of machine decision making could be a goal that is set as an end target for the program. The complexity and research needed into this big as it would have to look at plays from the analyst point of view not just measure. It would also rely on a system that is complex enough and reliable to allow for advance processing and the program is currently in an infant state that could not support this.

The final point to make is that the analysis teams are not looking for these types of programs that make decisions. For now future development should focus on helping them and on improving the overall tracking and measurements used. This would bring a bigger benefit and maybe get the program to be used.

Reflection & Summary

Overall the research and development of the project has been really enjoyable. It has given me insight in to the analysis data produced on rugby and that there is a market out there for programs that automate the data gathering in games.

The main achievement that I take from this is that the main aims set at the start of the project have been hit. It went from not knowing what measurements would be useful to having multiple measurements that allow the user to describe what has happened during a phase. For me this means that I can say that it has achieved what I set out to do.

The project has run smoothly with only a few issues along the way. There has been a few items during the project that has taken longer than expected. This has been down to a number of reasons. Firstly, with a family member passing away took a couple of days out but I am proud of how it was handled and it hasn't affected the end result of the project. Secondly, due to the timescale of the project being so long there has been small incidences like illness that has just disrupted the flow of the project. In the project notes there is a quote of Hofstadter's Law:

> It always takes longer than you expect, even when you take into account Hofstadter's Law.
>
> **Source** D. R. Hofstadter. G¨odel, Escher, Bach: an eternal golden braid. Vintage Books, New York, 1980

This has been the case in this project and with as much planning as I have put into each week it has still been a challenge. The constant meetings with my supervisor have been a help throughout the project. It has kept me motivated and discussions about how to develop and improve the measurements have kept the aims in constant check. Overall the timescales have been met and a working program was developed on time which shows that the timescales and meetings set have worked.

With regards to the project there are some areas that I would have liked to develop more. Firstly I would of liked to of done more comprehensive testing and run different games to see where the extreme limits of the program was in terms of the tracking method used. It would have also meant testing the

reliability of the program. Due to the time limits I didn't investigate this and is something I would chase next time. The other change I would make is to the UI. When showing people the program the look does impact their first impressions. This means that there was a reluctance at first as to whether they would use it. Although this wasn't an aim it would have been nice to have time to develop a more user friendly UI.

Finally, the project has tested me and has shown areas where I do need to improve. For example some of my time management skills where I would sometimes put unnecessary stress on myself by trying to force certain tasks through. It did impact my motivation in the project and for other projects it would have to change as it did leave an impact during the Easter break where I didn't hit my aim for the work done.

To finish I have enjoyed challenge and I am pleased with the outcome of the project. Being able to use my knowledge of the sport and being able to rely on people with experience in the sport has helped produce the end result. It kept ideas moving and made sure the measurements being researched were useful. To approach to the project as more of a research project I believe was the best approach. If it was approached as a project to produce a program / UI it would have been harder as the information on what statistics and features were needed was thin. This way it has been shown what can be achieved from a computer vision system and from here the users can decided if it is worth taking on.

References

ChyronHego Corporation. 1966-2016. TRACAB OPTICAL TRACKING [Website] http://chyronhego.com/sports-data/tracab [Accessed: 22/04/2016]

Green Star Media Ltd. 2016. Holding the line rugby drill [Website] http://www.betterrugbycoaching.com/Article-1432-108-Rugby-Tactics,-Strategies,-Tips-Defence-and-Defensive-Systems-Holding-the-line-rugby-drill [Accessed: 22/04/2016]

Green Star Media Ltd. 2016. Rugby drill to coach a simple attacking move [Website] http://www.betterrugbycoaching.com/Article-1694-109-Rugby-Tactics,-Strategies,-Tips-Moves-and-Plays-Rugby-drill-to-coach-a-simple-attacking-move [Accessed: 22/04/2016]

Hofstadter, D. R. 1980 Godel, Escher, Bach: an eternal golden braid. New York: Vintage Books

John Wiley & Sons. 2016. How to Find a Vector's Magnitude and Direction [Website] http://www.dummies.com/how-to/content/how-to-find-a-vectors-magnitude-and-direction.html [Accessed: 22/04/2016]

Lynx System Developers, Inc. 2013. IsoLynx [Website] http://www.finishlynx.com/isolynx/ [Accessed: 04/05/2016]

Rugby World Cup Limited. 2016. Doing basics well is always New Zealand's first instinct [Website] http://www.rugbyworldcup.com/news/112396 [Accessed: 22/04/2016]

Shi. J, Tomasi. C, 1994. Good Features to Track [Conference] IEEE Conference on Computer Vision and Pattern Recognition (CVPR94), June 1994

Sportstec. 2016. Sportstec [Website] http://sportstec.com/ [Accessed: 22/04/2016]

The MathWorks, Inc. 1994-2016. Rectint [Website] *http://uk.mathworks.com/help/matlab/ref/rectint.html* [Accessed: 22/04/2016]

The Welsh Rugby Union Limited. 2013 [Website] http://www.wru.co.uk/eng/home.php [Accessed: 22/04/2016]

Tomasi, Carlo and Takeo Kanade. 1991. *Detection and Tracking of Point Features*, Computer Science Department, Carnegie Mellon University,

World Rugby. 2009-2016. Laws of the Game Rugby Union [Website] http://laws.worldrugby.org [Accessed: 22/04/2016]

Appendix

Normal Video frame capture

BGRemoved.avi frame capture

bwblob.avi frame capture