Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Keep your eye on the Ball: Mobile Tennis Analysis

Tom Mulligan

Supervisor: Dr. Kenneth Dawson-Howe

School of Computer Science and Statistics

School of Engineering

Trinity College Dublin, *the University of Dublin*

D02 PN40

Ireland

April 2022

A Thesis submitted to the University of Dublin in partial fulfilment of the requirements for the degree of MAI in Computer Engineering

Declaration

I agree that this thesis was completed in line with the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

I agree that this thesis will not be publicly available but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only.

Tom Mulligan, April 2022

Keep Your Eye On The Ball: Mobile Tennis Analysis

Tom Mulligan

Supervised by: Dr. Kenneth Dawson-Howe

SCHOOL OF COMPUTER SCIENCE AND STATISTICS, TRINITY COLLEGE DUBLIN

The game of tennis has a playing population of about 87 Million people, at this point in time there is no product capable of accurately analysing a game of tennis for the casual player. In this report, we propose a Multi-Camera Mobile Tennis Analysis framework for the casual tennis player, which would deliver capabilities currently unavailable outside of the professional game.

Our proposed system features up to four Mobile Phones, each oriented so as to carry out their own analysis of the game at hand. By communicating their findings to each other, each device has a complete picture of the events unfolding throughout the game - In tracking both the players and the ball within multiple camera views, our proposed system would be able to provide useful performance indicators such as the locations of ball bounces, heat maps of the player's movement throughout the court, as well as classifications of the types of shots made.

Within the scope of this project, we are implementing a multi-camera ball tracking and bounce detection system for two devices. More specifically, the objectives of this project are as follows: The detection of a tennis court within multiple camera views; the detection and tracking of a tennis ball within multiple camera views; the determination of bounces points, through multiple views.

Through the testing of our Court and Bounce Detection Systems, we found the former located key points throughout the court within an error of about five pixels, while the latter system's performance in detecting bounces varied greatly depending on the lighting conditions - In overcast conditions using multiple camera views, 64.7% of bounces were correctly detected, while those bounces were on average about eight pixels away from their true locations. In sunny conditions, 73% of bounces were detected, while an error of over 10 pixels was recorded.

Having implemented our desired functionalities, a number of significant obstacles were exposed. Sources of error such as lens distortion, variable frame rates and insufficient camera resolutions all led to our implemented systems carrying error that would be unacceptable for a genuinely trustworthy Tennis Analysis System. Significantly, such errors are intrinsic to the usage Mobile Devices for video processing, and thus poses the question: *Can the Mobile platform be comfortably used in the implementation of Tennis Analysis?.*

Thanks Mom, Dad, Ellen, Basil and Echo for the support through a long five years.

Acknowledgements

I would like to thank Dr. Kenneth Dawson-Howe for his help and guidance throughout this project. Without his advice and support this research would not have been possible. To my family and friends that never failed in their encouragement along the way, thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the following section, we will highlight the main objective and motivations of this study, as well as an Executive Summary of what can be expected throughout this report.

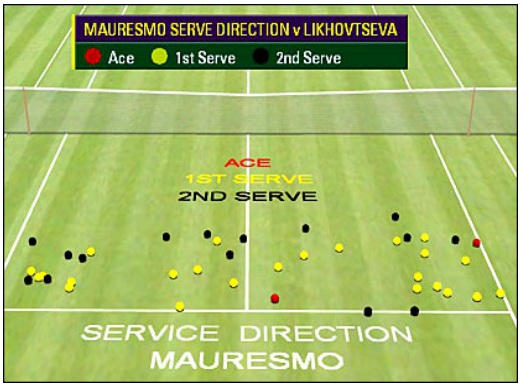
1.1 Objective

The objective of this project is to investigate the use of multiple mobile devices for the development of a Mobile Tennis Analysis system. In particular, our intention is to implement multi-camera tracking of a tennis ball's movement and bounces throughout a tennis court. By carrying out research and development in this area, we hope to set the groundwork for the future implementation of a highly available Mobile Tennis Analysis application for the casual player.

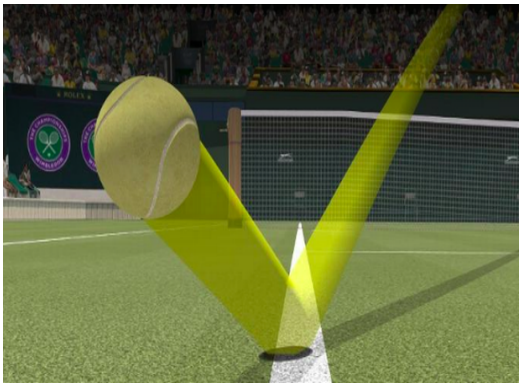
1.2 Motivation

In professional sports, the ability to analyse and critique player performance levels has been revolutionary in how teams and individuals have improved their game[1]. No longer is the focus exclusively on improved materials, equipment, and training regimes - Instead, fields such as Machine Learning and Computer Vision have found themselves at the forefront of performance advancement in sport. Famous technologies such as Sony's Hawkeye are used for their ball tracking and line-calling abilities in competitions such as the US Open, Australian Open and Wimbledon. Since the early 2000s this technology has seen a gradual inclusion into the sport, with the peak being the 2020 Australian Open[2], where Hawkeye Live was used to effectively replace all officials at the event during the Covid-19 Pandemic - *The professional game is now unrecognisable to previous generations.*[3]

Whilst there is no doubting the effectiveness of these technologies, as well as their popularity amongst some of the largest sporting franchises, they are far beyond what the casual player can afford and expect to use themselves. Hawkeye Live, which makes use of the PA system to issue announcements throughout the game, uses 12 cameras positioned throughout the court to watch the tennis ball itself, alongside six foot-fault cameras as well as a large amount of both remotely based officials and



(a) A serve report produced by Hawkeye, during Wimbledon 2005. Image credit: BBC



(b) Hawkeye Line Calling. Image credit: Mirror.co.uk

computers to ensure accuracy and smooth operation.[4]

This project is focused on the development of a low-cost tennis performance analysis application. Swingvision, a popular iOS application providing tennis play analysis, has recently received commendations for its performance as a tool for tennis players to comprehensively review several aspects of their own game. Despite the application's success, much of what it claims to do is outside of what would be deemed practical and realistic. For example, Swingvision claims to be able to gain accurate readings of shot placement for a single's tennis game. The application itself requires that the user's phone is placed at a point (on a tripod[5]) which can gain a clear viewpoint of the whole tennis court. This is however extremely difficult unless the camera is placed well above the height of what an average person can see. The height at which a mobile phone could be placed would suggest a decent viewpoint of the nearest side of the tennis court, but an obstructed viewpoint of the far side – Bringing much of what Swingvision claims to accomplish into question.

Figure 1.2: An advertised example of the SwingVision application. Image credit: SwingVision

1.3 Executive Summary

In this report, the belief is held that creating a system to rely on a single mobile phone is a marketing tactic, rather than a genuine breakthrough in personal tennis analysis. As such, the development of a system using multiple camera viewpoints around a court is required to create a truly functional tennis

analysis system. Thus, it is the belief of the researcher that there has not been a true breakthrough in tennis analysis applications for the casual player. With a playing population of about 87 million people, there is large scope for development in this area.

Immediately following this chapter, a review of the applications and research in the area of tennis analysis and multi-camera applications to sports in general is included 2. In each highlighted piece of research, comments on the study's suitability to the casual setting as well as this project in particular, are included. Also in this chapter, an overview of the Computer Vision theories relevant to this project are discussed.

Instead of relying on a single mobile phone, this project focuses on using multiple devices positioned on the tennis posts, where each device carries out its own analysis, with a collective understanding of the game at-hand generated via the "fusing" of information gathered from each view. In the next chapter, a full-scale implementation of the targeted system is described, where an illustration of how a system would be implemented with multiple mobile devices to provide a means of tracking the ball and the players within a game is also discussed in depth. By carrying out real-time classifications of shots made throughout the game, as well providing information pertaining to the movement patterns throughout the court, the intention is that this system would provide true professional-grade capabilities. To finalise this chapter, a summary of the aspects of this full scale system that this report will focus on are included (3).

Next, the design of this project's Court Detection system is recounted. This system is designed with the intention of localising a series of court points throughout the visible tennis court. The correct operation of this system is crucial to the accuracy of the Multi-View court detection system. As such, a three-step refinement process for localising each point of interest throughout the tennis court is described. Following the detection of the tennis court within the scene, a description of the implementation of the Ball Bounce Detection system (4.2), where each camera view attempts to search for and track the tennis ball is included. By tracing the movement of the ball throughout the scene, ball bounces are localised within each view (4.2.2). In order to gain a more complete outlook of the location of the ball with respect to the court, information is shared via a top down "perspective" view of the court, allowing the system to more accurately expose bounce locations, irrespective of an individual camera's view of the court (4.2.2). For each subsection included in this description of the design of the system, a brief review of the related research in each area is included 4.
Following this, the process of testing the implemented systems is discussed. Beginning with a short description of the footage used for testing, including comments on the scenarios covered, a detailed account of the process of gathering ground truth for each piece of recorded footage (with a particular emphasis on the errors associated) is included. Moving on from this, the various errors intrinsic to the development of a tennis analysis application within the casual setting are outlined. In addition, the

encountered errors associated with the choice of methodologies for each logical step in the design of the implemented systems are highlighted. Closing out the evaluation chapter, performance metrics, results and an accompanying discussion of the performance Court Detection, Single View Bounce Detection and Multi-View Bounce Detection systems are all included 5.

In this report, a background to the problem and application of tennis analysis, a description of the chosen methods for court detection and bounce detection (for single and multiple views), as well as an evaluation of the implemented system's design, and associated final conclusions are discussed. For the Court Detection System, it was found that the designed method yielded an overall error of 5 pixels between the detected Court Points and their intended locations. Additionally, it was found that depending on the lighting conditions, and whether the effect of variable frame rate came into play, the accuracy and precision of the Bounce Detection System to detect bounces varied from close to 100%, to about 60%.

Figure 1.3: Our system's interface. Top left: Camera view from the left tennis post. Bottom left: Camera view from the right post. Rightmost: Top Down (plan) view of the court, via a preset template (Credit to ShutterStock).

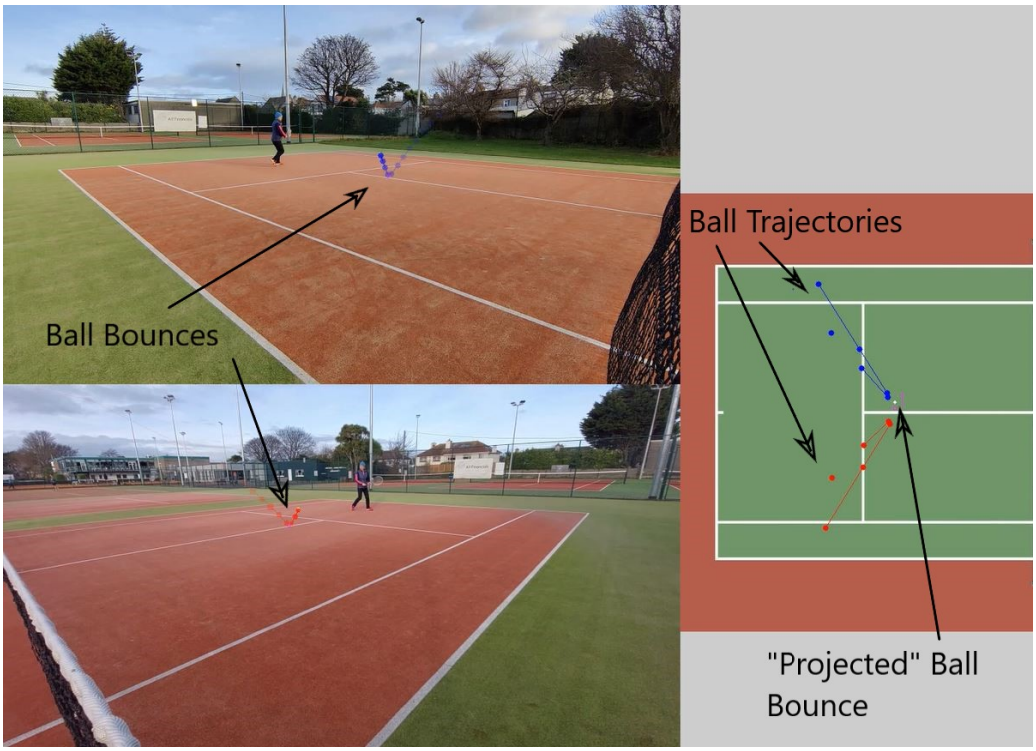


Figure 1.4: A detected ball-bounce. Both trajectories from the individual frames are projected to the planar view. The interpreted bounce location is also displayed (purple).

Figure 1.5: Some time later, the detected seventh bounce of that passage of play. Other purple points indicate previous bounces

Chapter 2

Background

With the objective of exploring the usage of multiple mobile devices for the development of a mobile tennis analysis system, driven by the lack of availability of a true tennis analysis solution for the casual player, we must now explore the background of this topic, and the works that have contributed to its study.

2.1 Tennis

For a complete description of the rules of Tennis, see the **US Tennis Association**.
For later reference, an illustration of the standard tennis court is included below:

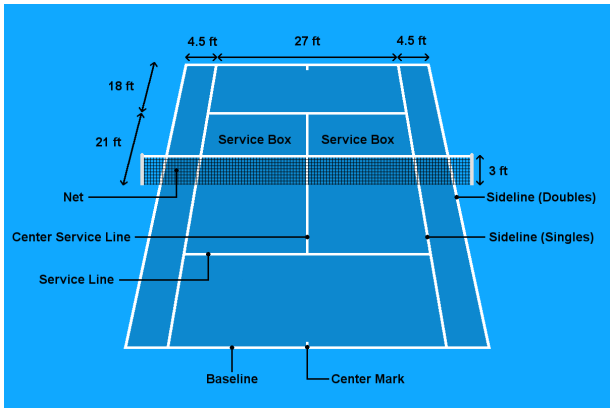


Figure 2.1: Illustration of the tennis court. Image credit: Tennis Uni

2.2 Previous Work

This section of the paper presents a review of related literature and innovations in the field. Firstly, we will focus on the research relating to the requirements of our use case. Following on from this, we will investigate the implementations of current Tennis Analysis systems, with a particular emphasis on how applicable each related innovation is to the average tennis player. Finally, we will summarise those aspects most suitable to build a mobile-capable implementation.

2.2.1 Multi-camera tracking

In Vision-related problems for Sports Analysis, the incorporation of state-of-the-art multi-camera surveillance techniques has proved to be useful for tracking both the ball and the players in a given sport. A key problem to face when tracking in sports is to overcome the issue of occlusion. In Tennis, this is a significant problem when attempting to track the game with a single camera, as there will often be either a player or net in the way of clearly seeing the ball. As such, it is far more effective to work towards tracking with multiple cameras. Whilst this paper is concerned with the implementation of a tennis-based mobile analysis application, we believe that there is value in research papers based on other sports such as Basketball. For this section we are purely interested in the use of multi-camera surveillance techniques across sport.

Studies by Nieto et al.[6] and Xu et al.[7] have both focused on differing techniques for tracking targets of their respective sports. Each of these studies have placed a strong emphasis on the theme of multi-camera tracking, and the difficulties associated. The former study was dedicated to building a system capable of automatically detecting and tracking both the tennis balls and players in a game. Due to the nature of tennis, it was decided by the researchers to position cameras in such a way that each individual viewpoint would cover an individual player's field. This way, the fusion process associated with combining recognised targets in different frames was relatively simple as at any given time, only a single player (in a singles game) would be visible. For a given target, Nieto et al. opted to combine (average) all of the x and y coordinates, from each camera frame, related to the target, resulting in "fused coordinates" for 2D tracking. Unfortunately, the study's overly simplified approach gave way to poor and unpredictable results. Due to poorly calibrated cameras with vastly different lenses, there was a significant error associated when tracking player movement. Additionally, since this paper was primarily concerned with player tracking, there was no consideration of partial or full occlusion in the research. Despite this, the study did find that a weighted-average approach for the fusion process would be much more successful, and that cameras raised as high as possible would (unsurprisingly) result in far better tracking results.

Having learned from Nieto et al's earlier research, Wu et al. decided on a much more complex, and successful, approach to provide efficient 2D tracking even when targets were occluded in certain frames. While no details are provided on the type and configuration of the cameras used, this study does make great improvements in its fusion process to enable 2D tracking in a basketball game. Wu et al. cleverly adapted a camera calibration approach[8] to essentially calculate a mapping of visible points between each camera around the basketball court. This gave the researchers a technique of tracking the basketball within a frame, even if it was occluded by a player. These learned mappings were calculated ahead of time, by establishing at least seven corresponding points between two cam-

era frames. Combined with Deep Learning-based detection techniques, this intricate methodology provided Wu et al. with high precision values of about 0.8, when compared to ground truth.

When considering important aspects such as occlusion, we can see that a complex approach such as what is adopted by Wu et al., is required. In the case of Tennis, an approach that overcomes the likely problem of occlusion is very worth considering.

2.2.2 Commercial Applications

Since the early 2000s, well known technologies such as Hawk-Eye[9] have been widely used for Television Broadcasting purposes, where the real time statistics they measure are presented to viewers live. Hawk-eye has recently seen more widespread usage during the Covid-19 pandemic, due to its highly-accurate officiating capabilities. In the system's original publication, N.Owens et al. claimed the technology to be affordable, easily installed and "near real-time" in its operation. At the time of its original commercialisation, the impressive technology which was made up of four state of the art calibrated cameras, paved the way for its eventual large scale adaptation across other sports such as Rugby, GAA and Soccer [10]. While the technology is the pseudo industry-standard for line calling and performance analytics, it does not offer a product or price range for the average consumer, with some estimates of its cost being up to 500,000 euros per year for the GAA to deploy it[11]. Such inaccessibility has prompted a recent popularity in developing more cost-effective and available Tennis Analysis and Coaching applications.

Thanks to the ever growing processing power that Mobile Phones are equipped with, analysis and coaching services for the more casual players are beginning to see use in tennis. SwingVision[5], an iOS application promising to deliver Professional-grade Tennis analysis capabilities, received commendations in 2021 for its game tracking and score keeping capacity. The product claims to provide high level information on both the tennis players themselves as well as the tennis ball in motion in real-time. These details include: the ball speed; spin on the ball; the stroke used; player footwork and movement patterns. However, when the nature of the product itself is considered, some of these claims seem outlandish. With the application requiring just a single mobile phone mounted at a height around which the average person might stand, the ability to discern ball placement from one end of a tennis court to another, as well as the ability to accurately track the player's foot movement seems very unlikely. The application features no pan, tilt or zoom capability, and whilst the latest iPhone camera technology provides a 12MP wide camera [12], the net of the tennis court still occludes a large proportion of what a single camera can hope to see. We can see that despite SwingVision producing a coaching and analysis application for the average tennis player, there is still some need for a more accurate and reliable analysis system.

2.2.3 Research Applications

Unsurprisingly, many researchers and academics have also sought to create Vision-based applications for Tennis coaching and analysis. Particularly comprehensive solutions such as O'Connaire et al's TennisSense[13] as well as research carried out by Messelodi et al.[14], Vinyes [15], and Reno et al.[16] have all sought to create systems that combine a number of methodologies, aiming to bring solutions to a host of problems. TennisSense, the oldest of these systems, and a study carried out in collaboration with Tennis Ireland, set about creating a coaching system to "facilitate efficient browsing" or events throughout a play session. The system itself made use of a multitude of high quality cameras, including an overhead camera to "detect moving objects" and provide a clear angle for visualising "tactical shots and movement." Using the acquired low level data, TennisSense would detect ball hits and track player movement around the court. While the results for this early study were promising, it's complicated camera arrangement (nine situated around the court) was unfortunately restricted to a single indoor tennis court. Additionally, only the single overhead cameras is used for the ball and player tracking techniques, resulting in very limited data for further processing. With improvements in camera quality and processing power, studies by Reno et al. and most recently, Messelodi et al., have all incorporated more sophisticated systems that deliver accurate tracking and recognition systems that cope well in both indoor and outdoor environments. Both studies make use of four cameras situated throughout the tennis court to gain sufficient data for full 3D reconstruction of the tennis ball's movement. The purpose of which allowed Reno et al. to accurately identify events involving the ball. For example: Bounces; racket strikes; serves. Messelodi et al. took this high level data acquisition a step further by constructing a "Supervisor" module that would extract data such as the type of Stroke, "player occupancy and movement maps", as well as information relating to the spin put on the ball. Unfortunately, due to the research being implemented and commercialised as EYES-ON[17], it is not known how exactly Messelodi et al. obtain these details. Finally, Vinyes et al. perform a similar high level analysis, except with the addition of Deep Learning-based models to make in depth classifications on stroke types and service types. Though the addition of state-of-the-art models make for an attractive research point; the lack of available data, even when using the THETIS[18] data set, meant that many of the classification's metrics suffered from poor performance.

Rather significantly, despite the particular success of Meseloldi et al's study and the research title suggesting a "Low Cost" alternative, all three of these implementations are still far beyond what a casual tennis player would hope to use themselves. All of these studies required powerful external Computers, connected to at least four high quality cameras positioned at exact predefined locations on the court. There is no denying these project's applicability to a Tennis Club setting, but for a quick plug-and-play configuration these systems are no longer suitable.

2.3 Computer Vision Techniques

2.3.1 Colour Spaces

Traditionally, we represent images using the three channel RGB (Red, Green, Blue) representation. Whereby, at each point in an image (pixel) there will be a grading of 0-255 for each of the three primary colours. Whilst it is useful for us to carry separate information regarding each of these channels, there are other colour representations that allow us to encode colour as a single channel. Such representations also allow us to quantify variables such as hue (colour), luminance (brightness) and saturation (colour intensity). We call these images, HLS images [19]. Normally each of these channels would be represented on a scale of 0-255, though in OpenCV [1] the hue channel is given values from 0-179. Significantly, the minimum and maximum values of Hue are just one unit away. See the figure below. Throughout much of this project, we chose to use HLS image representations due to the ease of use it provides when attempting to restrict the amount of visible colours in the image. For example, we can anticipate ahead of time that a tennis ball will appear at a Hue of around 30 (yellow).

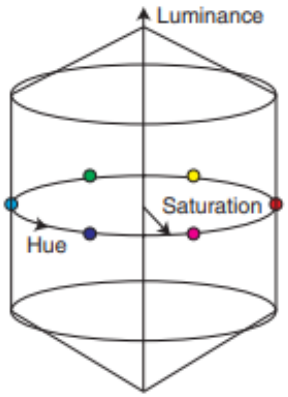


Figure 2.2: The HLS colour space. Credit: Dr. Kenneth Dawson-Howe

[1] https://github.com/opencv/opencv

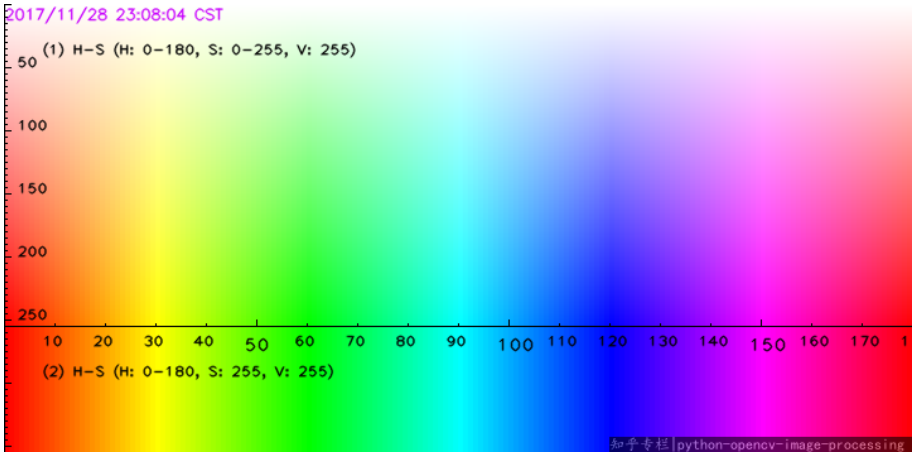


Figure 2.3: The OpenCV Hue Range. Image credit: OpenCV

Throughout the *Development of the Bounce Detection System* chapter, we will regularly refer to the act of applying a "mask" to an image. The applied mask is typically a "binary" image. i.e. Black and white only. This process is described by the following figure:

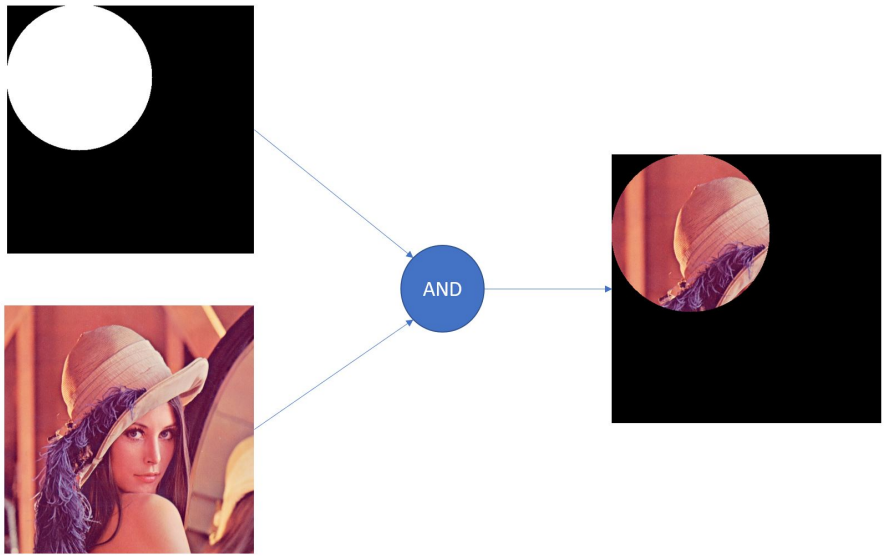


Figure 2.4: An example of image masking with a binary AND operation.

2.3.2 Region Segmentation

When considering distinct objects within an image, we face the issue of actually deciphering what pixels are connected to one another. The connected areas are referred to as regions, and unlike what may initially be assumed, finding these regions is not a trivial problem. Before actually approaching the problem of labeling distinct regions, we must first break the image down into two distinct parts: foreground and background (more on this later). Simply put, the foreground of the image will contain pixels each with a value of 255, while those background pixels will have a value of 0. Determining the foreground and background of an image is a whole other problem. Commonly, there are two [20]

approaches to determining connectedness between any given pixels in a binary image: 4-adjacency and 8-adjacency. The former considers only those points immediately adjacent (non-diagonal), while 8-adjacency considers all neighbors (eight in total).

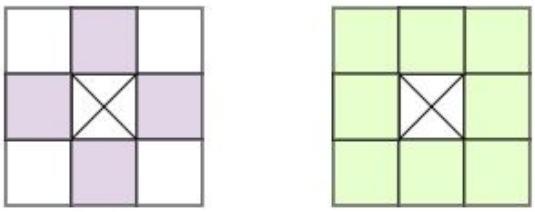


Figure 2.5: 4-adjacency (left) .vs 8-adjacency (right)

The issue here is that if we apply either of the two approaches, we do not achieve the intended result. Instead, the best approach is to use a combination of the two techniques given different scenarios. This gives rise to the Connected Components Analysis algorithm (CCA), which is a two-pass technique that labels points based on previously neighboring pixels. In OpenCV, CCA is implemented using the *findContours()* [21] function. Below are some example outputs:

Figure 2.6: Logo of Trinity College Dublin. Image credit: TCD

Figure 2.7: Segmented regions found in the image, using OpenCV's *findContours*

2.3.3 The Pinhole Camera Model

The most simple model of a camera is that of the Pinhole Camera Model. In this abstraction, the lens of the camera is seen as a pinhole to a flat plane behind it. By taking all light from the scene in front of it, the pinhole projects the image at hand onto the image plane itself. For this very simple case, any given point in the 3D world, visible through the pinhole, can be modeled to the 2D plane.

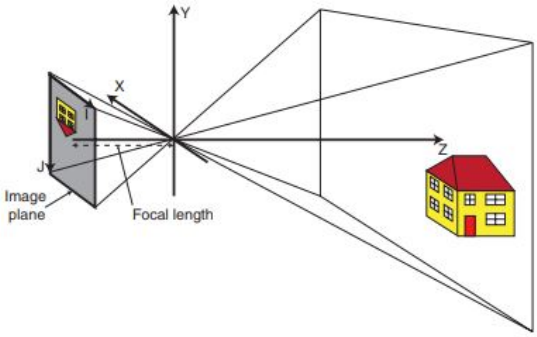


Figure 2.8: The pinhole camera model. Image credit: Kenneth Dawson-Howe

2.3.4 Perspective Transformations

When each point of the 3D world is projected onto this 2D plane, we are carrying out what is referred to as a *Perspective Projection*. If it happens that the object at hand is not face-on to the camera, and we wish to transform the projected 2D image to face directly to the image plane, we can carry out a transformation known as a *perspective transformation*. By taking four matching points between the original image and the to-be-transformed image, we can construct a mathematical operation to apply to the entire image - Effectively translating the region of interest to a face-on view.

Figure 2.9: A sample tennis court image

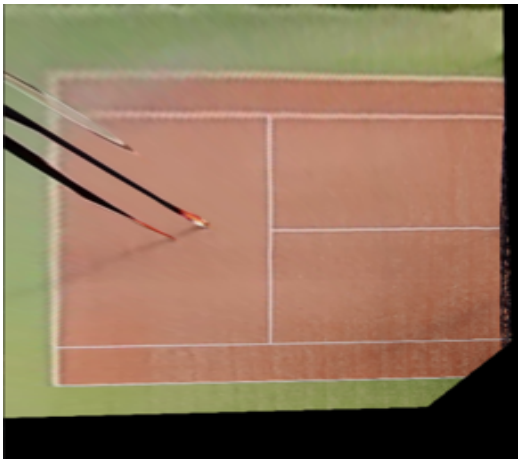


Figure 2.10: The same image, with the court projected face-on to the frame

2.3.5 Background Models

From a human perspective, detecting moving objects in a piece of video footage might seem somewhat obvious. In reality, we have to deal with a host of problems such as brightness changes, our distance to objects, video distortion and background changes. Solving each of these problems, and providing a framework to robustly deliver the foreground of moving pixels separate to a background is very difficult. Especially, for example, if the background also happens to be moving. The Gaussian Mixture Model (GMM), proposed by Stauffer and Grimson [22] is an unsupervised learning technique that provides a

robust framework for handling background pixels which might observe periodic motion (trees moving, water rippling). Each point in the scene is defined as either foreground or background, with each point handed a weighting based on how often that point has appeared throughout the history of frames. Those points with a weight higher than a given figure are referred to as background pixels.

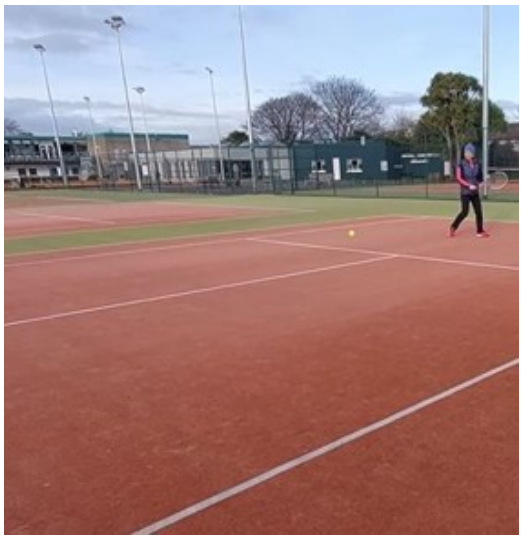


Figure 2.11: A sample tennis court image

Figure 2.12: The extracted foreground of the same image

2.3.6 Edges

Edges represent those points of an image where we observe sharp changes in luminance. Whilst for the human eye the localisation of these points might be trivial, robustly detecting edges of interest to the user and filtering out others is a task in which a large number of Vision techniques attempt to solve. In this section we will discuss the localisation of these edge points, as well as the further processing of such points to identify lines within the image.

Figure 2.13: Single channel (greyscale) image

Edge Detection

With respect to this project, we will consider the detection of images within single channel (greyscale) images, where each point has a gradient associated to how sharp the change in luminance is. First

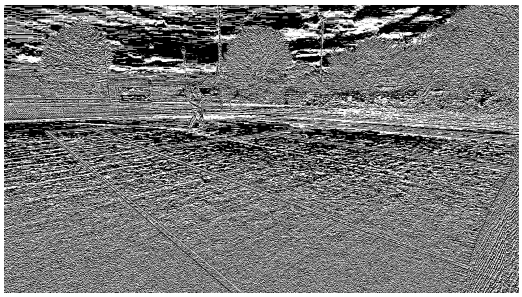


Figure 2.15: The Vertical Partial Derivative

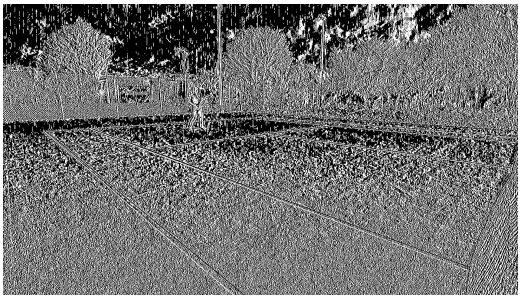


Figure 2.16: The Horizontal Partial Derivative

Figure 2.17: Resultant gradient image

derivative edge detectors are those techniques which output a local maximum to identify edge points. These detection techniques involve the calculation and combination of two partial derivatives, their combination to form a combined image, as well as the thresholding and suppression (filtering) of non-maximum points. The technique considered in this project is the Sobel Edge Detector. This particular method requires the usage of two specific partial derivatives, followed by the combination of their outputs to give the resultant gradient image. Since images do not exist within a continuous domain, partial derivatives are not calculated in the traditional sense. Instead, partial derivatives are calculated as a result of a convolution of a kernel across each point of the image.

$$h_1(i,j) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad h_3(i,j) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2.14: The Sobel Partial Derivative kernels. Image credit: Kenneth Dawson-Howe

Non Maxima Suppression

To filter our noise from those central edge points that we want to consider, Non-Maxima Suppression is carried out by iterating through each edge point in the image, and taking only those edge points whose gradient is greater than either two of its neighbours.

2.3.7 Line Detection

Using the edge points generated using the techniques discussed in the previous section, lines can be extrapolated from within the observed using the following methods:

RANSAC

Random Sample Consensus (RANSAC)[23] is a more modern technique used to estimate the parameters of a given model (line, circle or otherwise), amidst the existence of outliers. For the detection of lines, the technique aims to randomly select two points from a set of observations which give the best "consensus" for the existence of a line. The figure below, describes how with each iteration, a random sample of two points are taken to form a line model, where all those points lying within a certain distance from the line are taken as forming a "consensus" for the existence of the line. Given a big enough consensus, the line model is re-calculated using each of the points included.

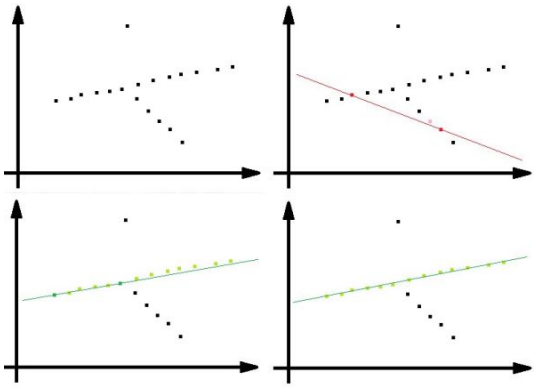


Figure 2.18: The RANSAC method. Image credit: Dr. Kenneth Dawson-Howe

2.3.8 Optical Flow

To augment the ability to track objects moving through a scene, the motion field (how points move from frame to frame) can be computed. This 2D field of motion, which assumes brightness constancy over a given time, is called optical flow. This field illustrates to us the apparent motion direction and magnitude for each point in a scene. In order to reduce the cost associated with searching for related points between two frames, a region of (2w+1 by 2w+1) is taken about the previous location (where w is the region width). By looking at the direction of the change and the magnitude of change, we can compute Optical Flow [24].

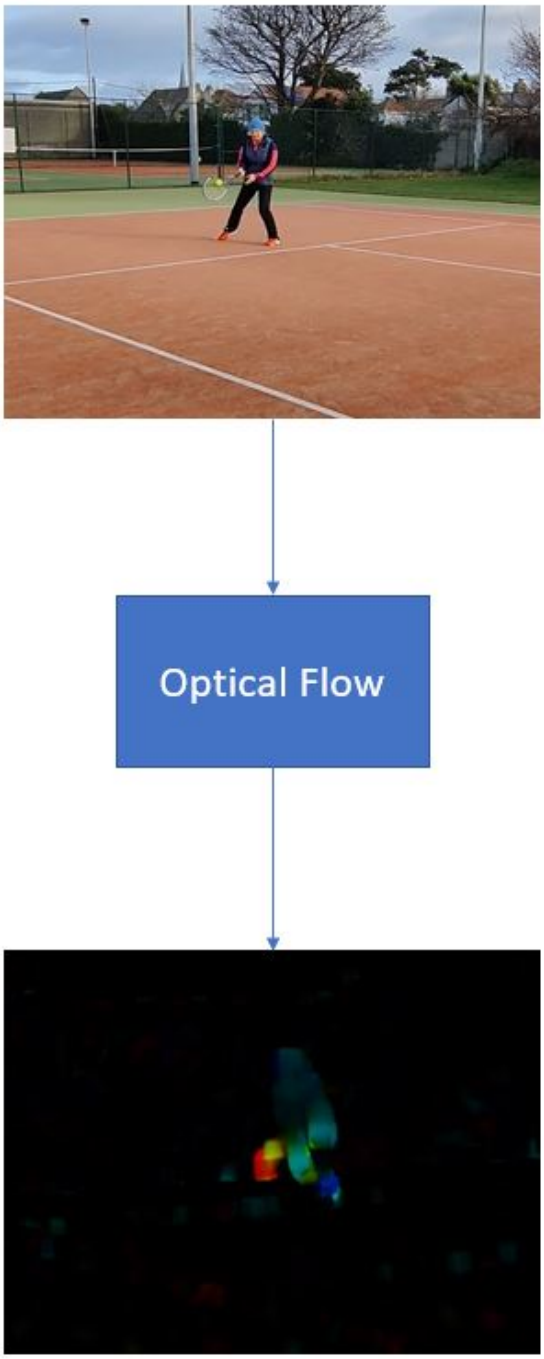


Figure 2.19: An illustration of the motion field obtained via Optical Flow.

Chapter 3

System Overview

Having discussed and analysed the various relevant works to our study, we can now further explore the design of this project. In this section, we will highlight how a full-scale implementation may be described and marketed as a product, as well as a closer look at a more narrowed-down list of desired functionalities for such a product. For each listed functionality, we will then provide detail of how such capabilities could be attained. Finally, we will summarise a set of goals that we are setting out to implement for this particular project.

3.1 Product Description

As a final product, our system would give the average tennis player the ability to easily track their own performance throughout a game. By combining functionality to watch player movement, detect ball height and bounces as well as the ability to perform line calling, we are targeting capabilities rarely seen outside of the professional game. This system works by utilising the impressive camera quality and processing power of modern smartphones. By positioning the players' smartphones at both ends of the net, watching both ends of the court, we can gain a complete view of the tennis game being carried out. Each of these phones perform tracking and recognition tasks on their side of the court. With the detected information (court details / player movement and shots / ball location) then combined to retrieve a more complete and accurate outlook of the game at hand. With the high availability of mobile processing power, we believe our combination of multiple smartphone viewing angles allows for levels of performance not currently available to the average tennis player. (Note: Mock-up pictures of this product's interface and functionality are included in: A.1)

In summary, the desired functionalities of this product would be:

- Detect and recognise separate players.
- Track player's movement throughout the court, and deliver classifications on the different shots made by each player.

- Detect the tennis ball, track its movement and height above the court as well as its bounce locations on the court.

- Following a game's completion, provide heat maps of each player's movement through the court, as well as a map of the ball's bounce locations following shots made by each player.

- Given the information gathered from the game, provide information on how likely a given player is to succeed in a) a serve from a certain part of the court b) returning a given type of shot, c) earning points from making a certain type of shot.

3.2 Project Description

In order to provide detail on how each of these desired functionalities would be implemented, we will now discuss the physical make-up of this system, as well as some detail of the system architecture and communications.

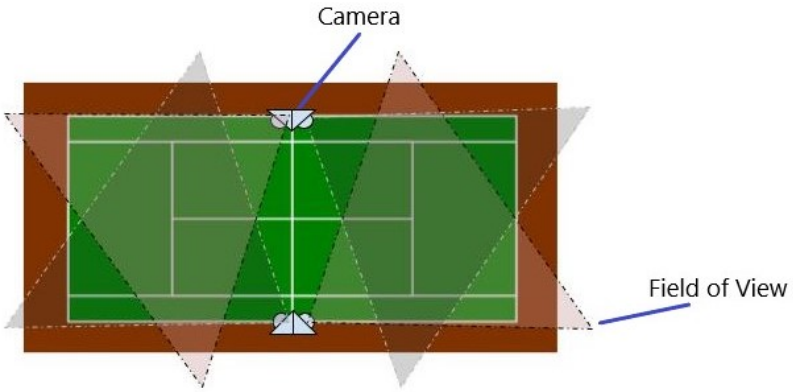


Figure 3.1: An illustration of the proposed camera arrangement

3.2.1 Hardware

For us to build a system that is capable of gaining a complete outlook of a tennis game, using mobile devices, we have decided to use multiple phones positioned on the tennis posts, looking outwards to the court's service lines. At a maximum, there will be four phones positioned on these posts for a doubles game 3.1. We intend for this system to be implemented in Android, with the focus being an application that communicates to multiple other devices also carrying the software. Each device must carry an ultrawide lens capable of recording in 1080p at 30fps. The phones themselves can be positioned using flexible tripods.

3.2.2 Event-based data

In order to gain truly complete information related to the tennis ball and the players in the game, each device carries out its own analysis of the key events throughout the game (shots, ball bounces, player movements). In more detail, a flow of steps required to be completed by each device is described in figure 3.2. As a brief outline, each device begins by searching for and detecting the tennis court within view. Once a list of points throughout the court (intersections of court lines) have been identified, each device can then begin searching for and tracking both the player and the ball. As event-based data, bounce points are identified throughout the court and shots made by the player(s) are detected. Using the previously identified court points as a way of relating two views of the same court section together, the detected information in each view is fused together. By considering the combined information from two views, we can identify the ball's height above the ground by considering the relationship between the separate interpretations of the ball's location 6.3. Additionally, by looking at two separate interpretations of a ball's bounce point, we can estimate a more accurate interpretation of the true bounce point. Finally, since we can expect small movements in the cameras during play (from the ball striking the net), the Court Detection step is repeated by carrying out searches local to each of the previously identified court points.

3.2.3 Communication

Within each group of devices, there will be an assigned "master" device, where each other device will communicate its findings to. Each device, including the master device will carry out the necessary Computer Vision and Machine Learning algorithms to attain the details required for the game at hand. WiFi Direct, a faster and more secure successor to Bluetooth, is to be used as a medium of communication [25].

3.2.4 Performance report

Using these event-based details, information of greater interest to the user can be generated. By compiling the collected information throughout the game, heat maps of player and ball movement can be generated. Additionally, with enough data over a number of games, we can provide predictions on how likely a given player is to earn or lose points from a given shot.

3.2.5 Project Focus

What is presented here is an overall outlook into the design of a fully implemented mobile tennis analysis tool. For the scope of this project however, we will only consider the usage of two devices, pointed at a single side of the court. More specifically the following aspects of this overall system are targeted:

1. The detection of a tennis court within multiple camera views.

2. The detection and tracking of a tennis ball within multiple camera views.

3. The determination of bounces points, through multiple views.

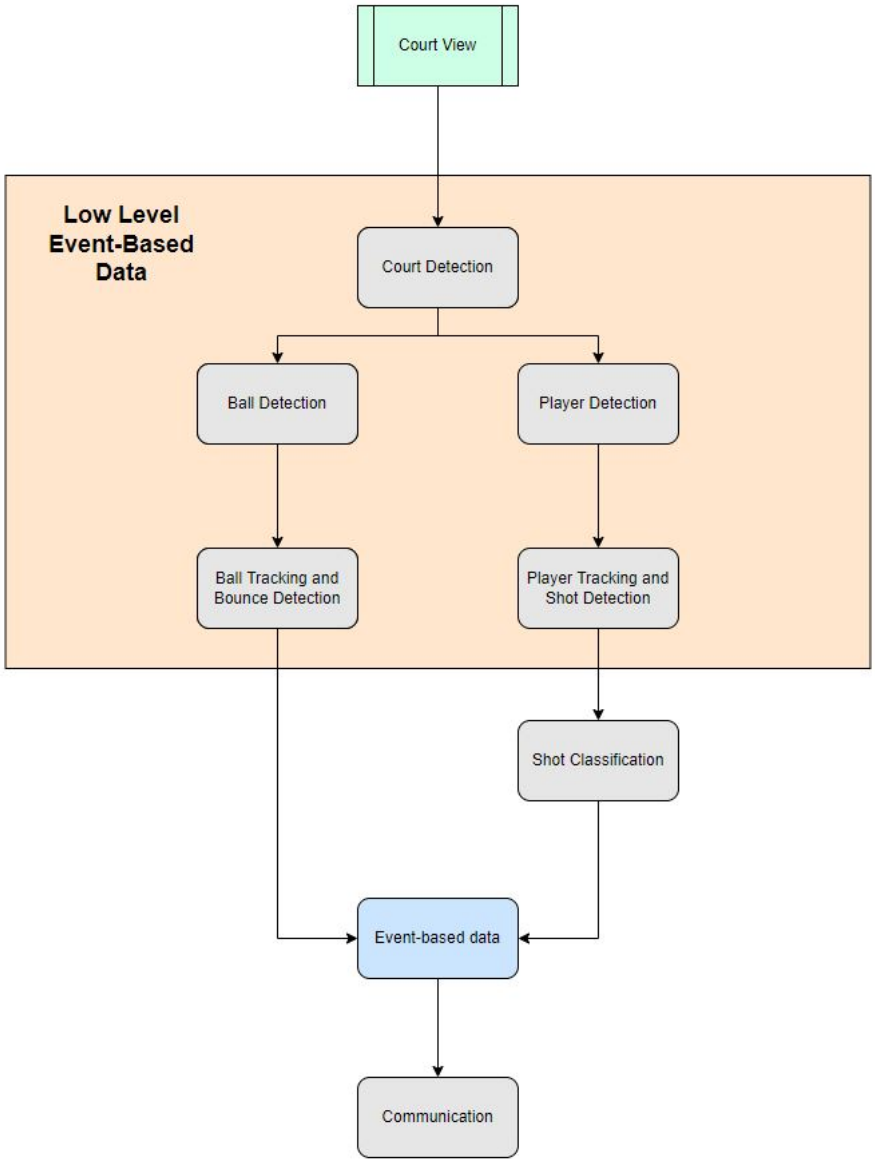


Figure 3.2: An illustration of the proposed flow of events for a single device.

Chapter 4

Development of the Bounce Detection System

Having proposed a full-scale Tennis Analysis Application for mobile, as well as our own more specific goals for this project, we are now in a position to describe the details of the development of our system. For this project, OpenCV 4.5.3 was used to implement our solution. The C++ distribution of this software was chosen for use, due to language's performance benefits when compared to Python. Primarily, the OnePlus 8T, with its 123 degree ultrawide lens was used to record footage. As a Secondary device, the Samsung A51 was used. All devices used were capable of recording from their ultrawode lens at 30fps, in 1080p resolution. In this section, we will begin by providing an overview of the theory related to our work, as well as a detailed breakdown of the following steps:

1. Court Detection
2. Ball Detection and Tracking
3. Synchronisation

4.1 Court Detection

Before carrying out any analysis on the game of tennis at hand, our devices must first locate the tennis court itself. This is done by searching for, finding and tracking a set of key points throughout the court.

4.1.1 Previous Work

In the late 1990s, Sudhir et al.[26] proposed a system to automatically retrieve information from tennis broadcast video. As part of this, the study worked at detecting the court itself by colour, and detecting the court lines by a custom "line growing" technique. The researchers compiled a set of assumed RGB values for each type of court surface (carpet, clay, hard or grass), and would compare a section of the frame under view to each of these known colours to determine if the court was under view. Following a successful court detection, the researchers carried out their "line detection technique, which took as inputs a starting point and a line-growing direction." This technique attempted to search for straight line segments through a given number of directions. In a similar manner, Tien et al [27] relied on using the most dominant colour under view to search for the court in a frame. Unfortunately, while both studies claimed high accuracy in broadcast footage, no quantitative results were provided. Such research also relied too heavily on assumptions regarding the conditions at hand, and would not, in their base form, be suitable for implementation outside of a professional setting. Some years later, Farin et al.[28] built their court detection model with an emphasis on white pixels appearing on the court lines. By isolating these pixels and applying the Hough transform, the researchers applied a heuristic to retain only those white pixels whose neighboring darker pixels appear a given distance away 4.1. With the reduced number of candidate lines, associations were then made between those found lines and lines existing in a real tennis court. Like the previous study, no quantitative results were provided. This study did however make a significant improvement in the computation time of court detection and tracking, with near real-time performance observed on relatively weak hardware by today's standards. Vinyes et al.[29] took significant inspiration from this study, but added some extra steps to ensure the removal of the tennis net from consideration. The study included a method of determining the confidence of detection, having also observed the lack of numerical results from previous studies. Despite the extra computation time, Vinyes reported about 70% detection accuracy across frames, with significant error also recorded for court lines slightly worn away.

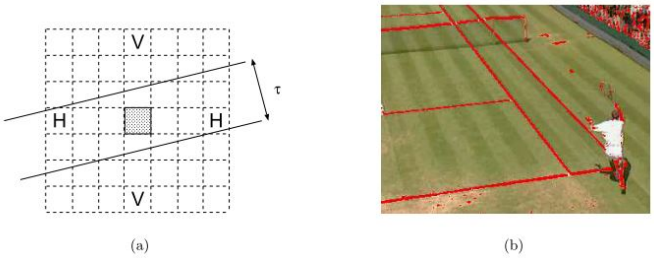


Figure 4.1: The court line heuristic used by Farin et al. This simple rule assumed a constant court line width in the broadcast footage. Image credit: Farin et al.

4.1.2 Our technique

Locating the court

In setting out to locate the key points (court line intersections) throughout the court, it was decided that the court sections themselves would first be located. To do this, we considered the dominant colour within the visible scene. By first converting our original RGB frame to a HLS representation 2.3.1, we searched for the most dominant colour within the image by generating a hue histogram of the frame and taking the greatest bin 4.3. Where the index of this bin was the most dominant hue within the image. We also applied upper and lower bounds (+5 and -20 respectively) to this hue value to allow for noise within the colour of the court itself to also be considered. Significantly, we also had to consider very high hue values, since the colour red takes both very high and very low valies in the HLS image representation. In addition, upper and lower bounds of luminance (lightness) and saturation (colour intensity) also had to be applied. By allowing for a range of luminance and saturation values, our Court Detection System was built with robustness to shadows and sandy patches in the court (brighter regions). Due to the nature of this implementation, a number of different combinations of luminance and saturation bounds had to be experimented with for a given view. For example, upper and lower bounds of 40 and 175, and 25 and 255 were observed for luminance and saturation respectively for an overcast view in the primary camera. In the same conditions, bounds of 20 and 175 were observed for luminance, with values of 70 and 255 observed for the secondary camera - In the secondary camera, it was found that the white of the court lines would regularly blend together with the orange court, as such we required a stricter colour strength for the orange court.

Next, by taking a mask of the image with this identified hue (within some range), we obtained a binary image containing the court regions as well as some noise in the background of the scene 4.4. Finally, using OpenCV's *findContours()* function, along with a set of rules (restricting the length of court sections to be greater than 780 pixels) relating to the expected aspect ratio of each section, we obtained a region for each court section.

Having identified each court region, we also implemented a system to check the coordinates of the highest point in the NEAR region to identify which side of the court the camera is placed on. If this

Figure 4.2: A tennis court used for the development and testing of our Court Detection System.

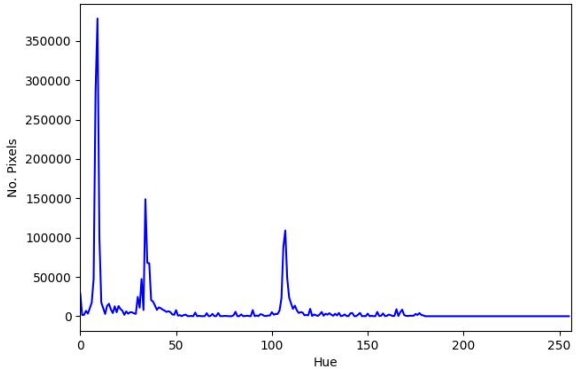


Figure 4.3: A hue histogram taken from the above image.

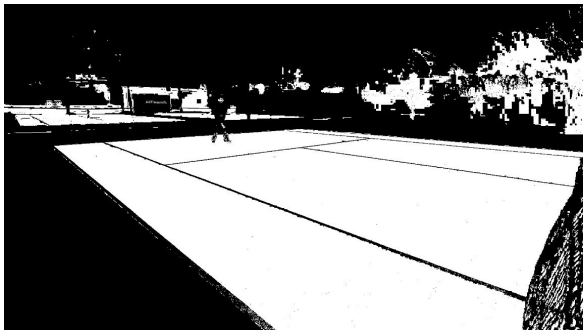


Figure 4.4: Binary image produced by taking only those pixels containing a hue within a set range.

point had an x-coordinate less than half the width of the image, we assumed the camera to be placed on the left side of the court. Otherwise, we assumed the camera to be placed on the right side.

Identifying each court region.

Having obtained each court region, court line intersections were first estimated as existing on the extremities (corners) of each court region. Before carrying this out however, each section needed to be correctly labeled as their equivalent section in a real-life court. This was achieved by stepping through the image from left to right in certain increments (image width divided into 15 increments), where a search upwards through the image was carried out at each step. During each search, the intersecting court sections were registered and stored. A set of rules could then be applied 4.5:

1. If a search has only encountered a single court section, then identify that section as the "NEAR" section.

2. If a search has encountered two sections, then identify the first section as the "NEAR" section and identify the second as the "BACK MIDDLE" section.

3. If a search has encountered three sections, then identify the first section as the "NEAR" section and identify the second as the "MIDDLE NEAR" and the third as the "BACK MIDDLE" section.

4. If a search has encountered four sections, then the third encountered section can either be the "BACK MIDDLE" or "MIDDLE NEAR" section. As such, we can only say that the first, second and fourth identified sections are: "NEAR", "MIDDLE NEAR" and "FAR".

5. If a search has encountered all five sections, then label each in the following order: "NEAR", "BACK MIDDLE", "MIDDLE NEAR", "MIDDLE FAR", "FAR".

For efficiency reasons, the search was carried out in 15 steps through the image from left to right. If the search were to be carried out more frequently, it is likely that only the final rule would be required to identify each court section. Notably, this search assumed that *all* court sections are detected in the first place.

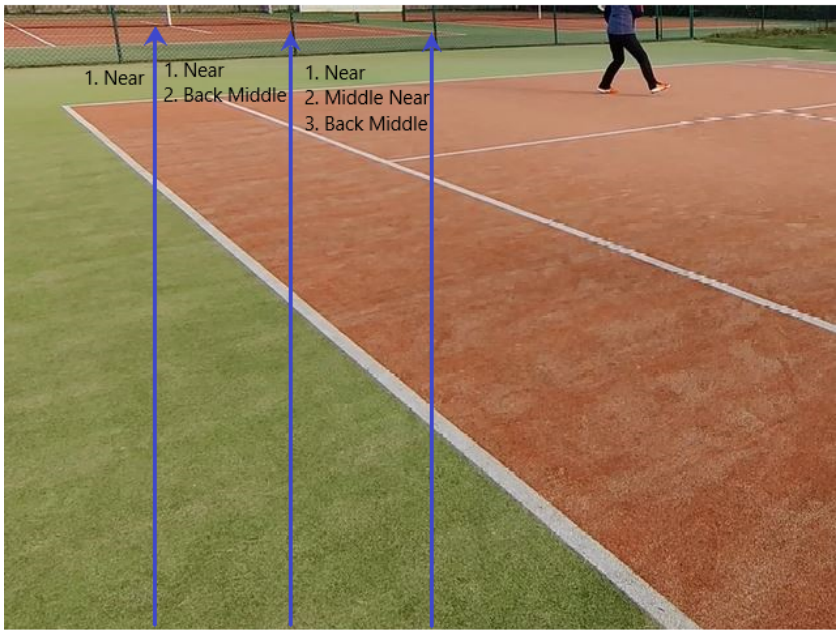


Figure 4.5: An illustration of court-section recognition.

Initial court point estimates

Having identified each court region, the next task was to make an initial estimate of each relevant court point (intersection of court lines). For this, an additional set of rules was observed to determine each point, given an identified section:

1. **Near Section:** "Near B" was found as the contour with the lowest y-coordinate (coordinate system measured from top left of the image). To find "Near A", both the contours with the

largest and smallest x-coordinates were first taken. (To allow for this to work for images from either the left or right side of the court, it was required that we take both of these points.) Then, out of those two points, whichever was closer to "Near B" was identified as "Near A".

2. **Middle Near Section:** In a similar way to the above rule, the contour point with the lowest y-value was first taken. Then to find the "Near Mid" point, both the contours with the largest and smallest x-coordinates were first taken, with the point closest to the contour with the smallest y-coordinate being taken as "Near Mid".

3. **Middle Far Section:** To find "Far Mid," the contour point with the smallest y-coordinate was taken.

4. **Far Section:** To find the "Far" court point, the contour with the smallest y-coordinate was taken.

Figure 4.6: The labels used for each point. Each of these points are the final, refined versions.

Making refinements

With each estimated key point, there will be a small error associated between itself and the true location. For this system, we required as close to perfectly accurate as possible, thus each point needed to be refined to fit closer to their "true" location.
To do this, we needed to consider the geometry of the court lines local to each estimated court point. Firstly, using each of the previously estimated points, a search was carried out in two directions along the contours from the initial estimate to calculate two separate lines. With each search, RANSAC (See: 2.3.7) was used to determine the line equations for each of the two located lines. By calculating the intersection of these two lines, we obtained our first refinement of each court point.

$$x_{inter} = \frac{c_2 - c_1}{m_1 - m_2}$$
$$y_{inter} = m_1 x_{inter} + c_1$$

Figure 4.7: Finding the intersection of two lines

To further refine the location of each court point, the true court lines within the image were then considered. As part of this, the width of each relevant court line needed to be found. Due to the angle of the camera, an approximate width for the court lines could not be assumed. Instead, it was required that the inner and outer boundaries of each true court line were found. For this, Sobel Edge detection and Non Maxima Suppression (see: 2.3.6) were used to extract a gradient image from the current frame. Using the lines obtained from the previous step, a number of perpendicular lines were taken at specified intervals along each previously estimated line. By searching through each of these perpendicular lines, tracking how the gradient increased and decreased in magnitude, and utilising the already-solved line equations from the previous step, two new line interpretations could be taken along the outside of the court lines. By taking the intersection of these new lines, the outer intersections of the court lines were determined. Notably, since the court lines were very difficult to detect on the far corner of the court, the best option to refine the far court point was to simply subtract (or add, depending on the side of the court) three pixels from the point's x location, and subtract a further three from the point's y-coordinate.

Depending on the particular point, we decided to refine it as the intersection of either the inner or outer boundaries of a pair of court lines. The nearest two court points to the camera were refined as being on the outside boundary, while the middle court points were refined to the inner court lines. This of course could be done in either fashion. See the figure below for a visual flow of events for finding the inner intersection of two court lines.

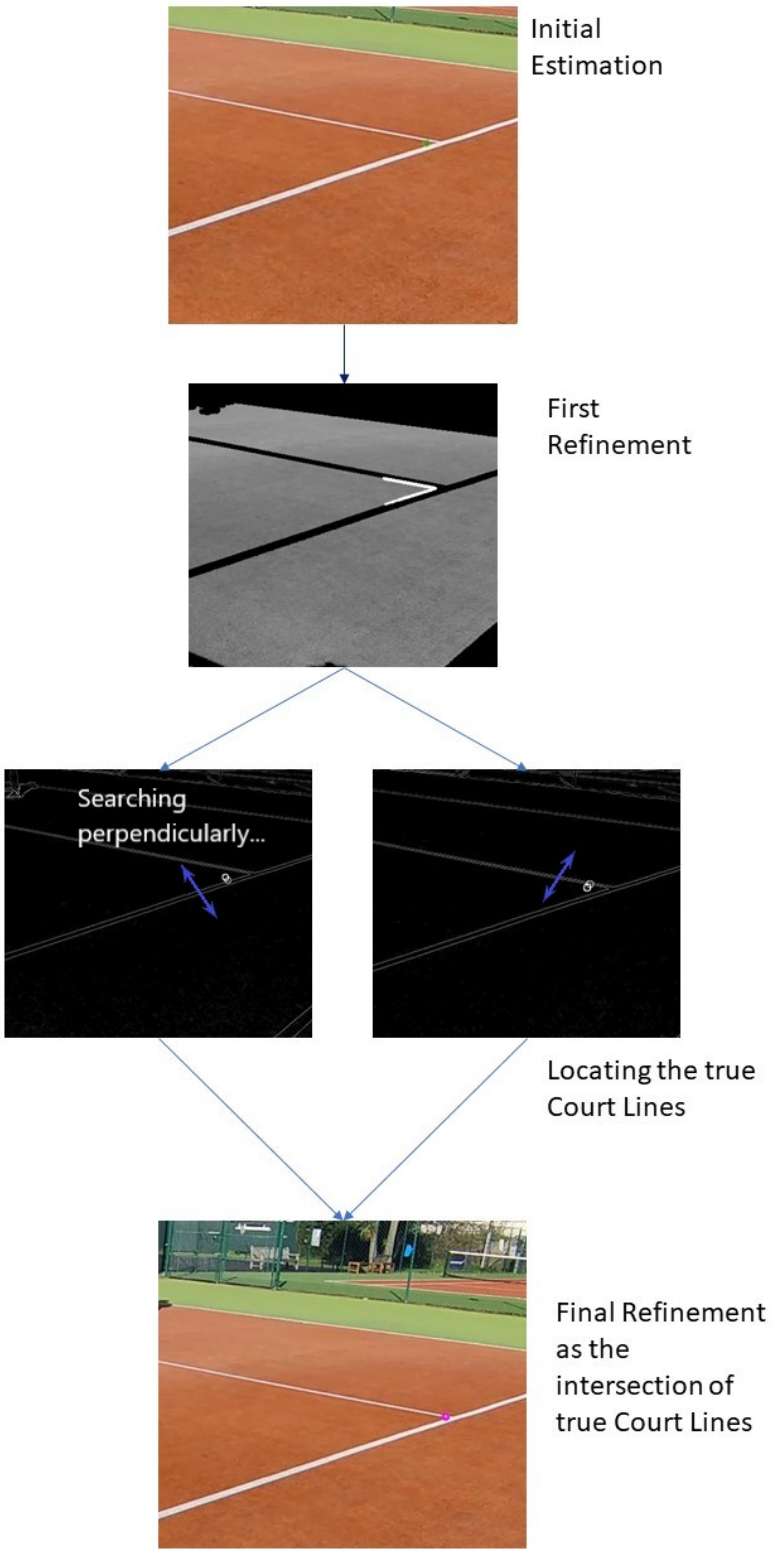


Figure 4.8: An illustration of the flow events through our Court Detection System.

4.2 Ball Detection and Tracking

In a game of tennis, we can expect the ball to move quickly across the court. At high speeds for a camera with a low shutter speed, the circular ball shape will begin to resemble more of a yellowish

ellipse. Many of the popular ball tracking systems [9][30][31] rely on high resolution footage at steep angles to the court. By taking footage in these angles, analysis systems are able to avoid losing the ball as it passes different scenery and lighting. Additionally, researchers can define a somewhat consistent ball size throughout the court. Unfortunately, these issues are unavoidable with our chosen camera setup. As such, we require a robust system that is appropriate to our imperfect recording scenario.

4.2.1 Previous Work

Since the late 1990s, researchers have attempted to address each of these problems to deliver a robust and deployable Tennis Analysis System. Pingali et al. [31], and later Ó'Connaire et al[13], formulated a ball tracking system based on the ball's location, physical size and movement. In order to generate a collection of potential ball candidates in each frame, both studies used frame differencing and thresholding to isolate any candidates within view. The former team of researchers chose to search within a window of the expected ball location, where any ball candidates found within the window would be checked to see that they fit the expected aspect ratio and size. To aid this, the ball's expected location would be calculated based on its observed velocity. Using multiple camera views, the ball's 3D position was found by triangulating multiple camera viewpoints. The bounce of the ball was then defined as when the ball's velocity in the z-component changed from negative to positive. While no quantitative results were provided, Pingali et al suggested excellent, near real-time performance. Ó'Connaire et al followed a similar technique, but implemented a *history* of ball locations, where each location found to lie in a "semi-linear path," would be stored to provide the ability to interpolate a ball position amidst occlusion. The researchers found that their own method, which relied on an indoor court with an overhead camera, performed very well with the precision and recall of ball hits at about 94%. Finally, a study submitted last year by Conor Gilmartin[32], aiming to produce a tennis shot placement report from a single camera, applied a somewhat similar technique of ball tracking through the storage of its location history. By comparing the ball's actual location, to its predicted location, based on its history, Gilmartin filtered out candidate balls in the recorded footage. Ball bounce detection was implemented as the intersection between upward and downward ball trajectories. In high quality footage, a satisfactory precision of 84% was found.

Figure 4.9: Ball history and the predicted ball position, as implemented by Gilmartin. Image credit: Conor Gilmartin

4.2.2 Our technique

For each of the headings below, a flow chart is included, detailing the approach of each method

Single Camera Ball Detection and Tracking

To begin, we needed to locate the moving pixels within the image. To do this, we extracted the foreground image from the moving scene using Gaussian Mixture Modeling (see: 2.3.5) . With the binary image output from this technique, we applied a masking on the original image, which gave us all those moving pixels within the RGB frame. With this resultant colour image, we applied a conversion of the image to the HLS representation, where we applied another mask filtering out all those pixels which did not contain the expected ball Hue (60 degrees). In the following binary image (see: 4.11), we then applied the *findContours()* method as implemented with OpenCV, to obtain all those "candidate" regions which represented potential tennis ball regions within the image. To further filter out those candidate "balls" within the scene, we then applied size and circularity constraints on each contour to result in an image containing only those regions which truly represented a tennis ball. Following a successful detection of the ball within the current frame (see: 4.12), the ball's location is saved to enable a more efficient search within a window of where the ball can be expected to be in the subsequent frame. Effectively, this allowed us to "track" the ball through frames.

Figure 4.10: Court scene.

Figure 4.11: Foreground of the scene.

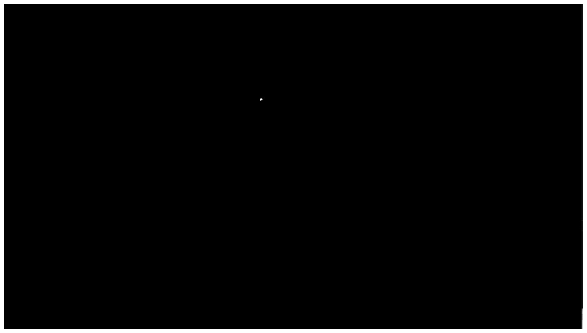


Figure 4.12: Resultant pixels following the Hue mask.

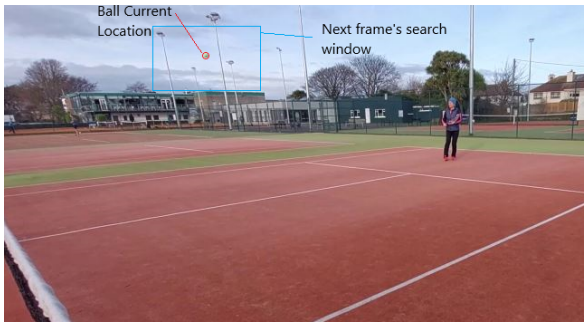


Figure 4.13: An illustration of the ball's tracking window.

$$Circularity = \frac{4\pi A}{l^2}$$

Figure 4.14: Equation for circularity, where l is the circumference of the region.

Single Camera Ball Bounce Detection

By tracking the ball's location through multiple frames, we were able to observe the vertical and horizontal components of the ball's trajectory. In very simple terms, we consider a ball bounce to be a change in the ball's vertical trajectory from downwards to upwards. In development and testing, this simple rule proved robust to upward shots made by the player, since a change of horizontal trajectory from outwards to inwards (of the net) would be detected during a shot. If a shot was detected, then a perceived bounce was ignored five frames either side of the detected shot. While this method gave us a rough idea of both the frame and the location of where the bounce was considered to have taken place, if the bounce of the ball happened between two subsequent frames 4.17, we would not have an accurate picture of where the ball has bounced. To ensure we captured the correct bounce location, Conor Gilmartin's technique was adapted to suit this current implementation. Very simply, this method involves the calculation of two straight lines immediately before and after the estimated bounce point, where the intersection of these two lines gives the true bounce location. If it was found that the originally estimated bounce point was closer to the court (larger y-coordinate) than the newly refined bounce point, then the estimated bounce was taken as the true location.

As an example, see 4.16. In this illustration, a change in vertical trajectory from downwards to upwards is detected at point 3. Using straight lines computed from 1-2 and 4-5, the intersection can be found as the true bounce point.

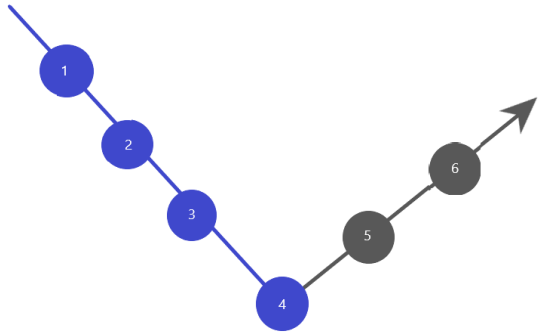


Figure 4.15: A correctly estimated bounce point.

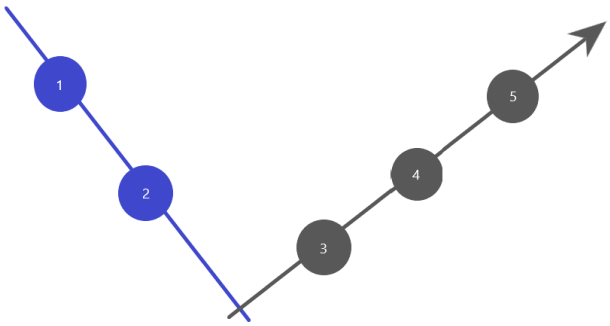


Figure 4.16: A bounce point taken as the intersection of the downward and upward trajectory.

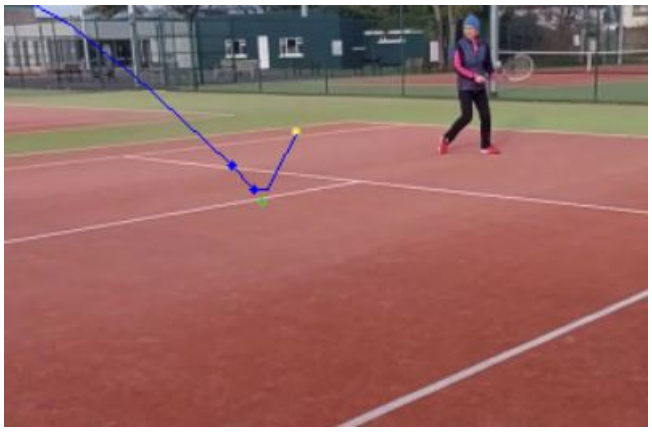


Figure 4.17: An interpolated bounce detected in footage

Multi-view Bounce Detection

Having carried out court detection, as well as the ball and bounce detection within each camera view, the next step was to combine each of the views to obtain a top-down view of the court at hand. Firstly, using the detected court points from the previous section, we carried out a perspective projection of each camera view (see: 2.3.4). To do this, we were required to sort each of the identified court points by their x-coordinate, from smallest to largest if the view in question was on the left side of the court, or from largest to smallest if the view was on the right side. This way, each point could be directly related to their equivalent point in a list of coordinates for the template "top down" court. While this produced a top down view of the entire court, including the tennis ball (within a certain distance of the court itself), we were only interested in the transformed coordinates of the originally detected ball locations. By performing this perspective projection on each camera view, we gained a useful

bi-product where the tennis ball seen in the primary and secondary cameras could be projected so as to appear within the top-down view. Considering the figure below 4.18, both the blue and red trajectories are mapped to the top-down view.

Figure 4.18: Combining the ball-information from each of the views to a top-down perspective.

By considering the path of each perspective projected ball in the top down view, we observed four separate interpretations of the bounce point. Firstly, by applying the same perspective projection to the bounce located in each individual view, each interpreted bounce point in the primary and secondary views were mapped to the top down (plan) view. The third interpretation was estimated as being the midpoint between the projected bounce points found in each of the individual views (see: 4.19). As a final interpretation, we calculated the bounce point as a weighted average of each individually determined bounce location, as projected to the top-down view (see: 4.20). To calculate this, we took into account the size the tennis ball appeared in each view. The following formulae were observed where: X_{Wavg} is the weighted average x-coordinate, d_{size} is the absolute difference in ball sizes, S_{large} is the size of the larger ball, X_{large} is the x-coordinate of the larger ball and X_{small} is the x-coordinate of the smaller ball. The same convention is observed for the y-coordinate equivalent.

$$X_{Wavg} = \frac{(1 + \frac{d_{size}}{S_{large}})X_{large} + X_{small}}{1 + (1 + \frac{d_{size}}{S_{large}})}$$

$$Y_{Wavg} = \frac{(1 + \frac{d_{size}}{S_{large}})Y_{large} + Y_{small}}{1 + (1 + \frac{d_{size}}{S_{large}})}$$

It is worth noting that if one of the views failed to locate a bounce within five frames of the time a bounce was initially detected, both the midpoint and weighted average interpretations would take the location of the bounce that had been reported by either view. Additionally, it was decided that if only

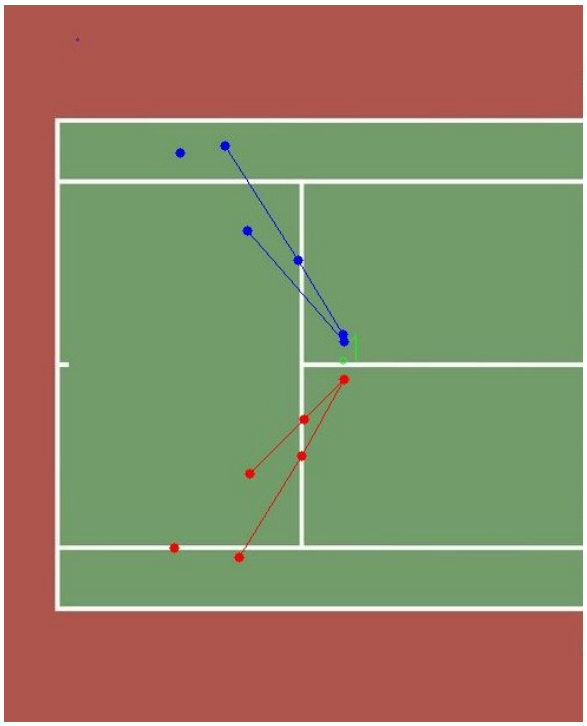


Figure 4.19: Top-down bounce point estimated as the midpoint of the two single-view bounce estimates, mapped to the perspective view.

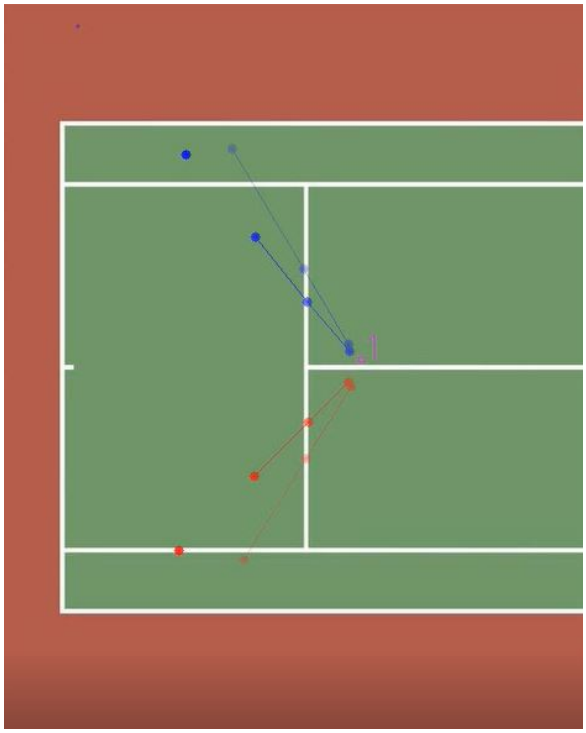


Figure 4.20: Top-down bounce point estimated as the weighted mean of the two single-view bounce points, mapped to the perspective view.

a single bounce was detected, the system would wait a total of ten frames from the initial detection for a detected bounce in the other view. If no matching bounce was detected in the other view within ten frames, then the initially detected bounce was taken.

4.3 Synchronising the views

Unfortunately, when recording with mobile phones instead of dedicated video cameras, we are adding a large amount of complexity to how frames are processed. For example, if a phone were to experience a spike in CPU usage, it might happen that frames are dropped from the recording [33]. With even a slight difference in the timing of frames from each view, it is necessary that the system we develop is able mitigate the risk of this happening in the first place, as well being able to accommodate for the inevitable small errors in the timing of each video. While the videos being one or two frames out of sync would not pose a huge issue to the operation of our system, the accumulation of error would result in the incorrect interpretation of events happening throughout the tennis game.

While it could be assumed that the problem of multi-camera synchronisation is one that is commonly addressed in research concerning the application of multi-view tennis analysis systems, it is instead either assumed that frames from separate views are at all times synchronised, or just ignored altogether. For our system, including the full scale implementation as a mobile application on multiple phones,

we can consider synchronous issues from both video processing and timing error in the instant that video recording commences on each device.

4.3.1 Previous Work

As a method of synchronising videos based on the dynamics of the scene at hand, Irani et al.[34] and Elhayek et al.[35] both focused their research on carrying out synchronisation based on the matching of visual features related to the trajectories of moving points within the scene. Irani et al. introduced the notion of using trajectory-based features for matching within the scene, with a particular emphasis on its benefits related to "sub-frame" accuracy. I.e. synchronisation between multiple views that is not wholly reliant on exact frames, rather the information drawn through a collection of subsequent frames. While the researchers were steadfast in the benefits of their solution, including synchronising views at different zooms as well as different types of sensors (e.g. visible and infra-red light), Elhayek et al. made significant advancement some years later. This study adapted the notion of time-based alignment based on trajectories of moving objects, by instead searching for trajectories of any moving points within the scene. While this approach undoubtedly adds complexity, it meant that multiple videos could be synchronised without having to rely on any one moving object. Considering the nature of our project, where there is no guarantee of a moving object (player or ball) at any one point in time, this approach would be more suitable for implementation.

Also worth consideration is the synchronisation of multiple videos based on audible features (i.e. a serve or ball bounce).

Video alignment via audio "fingerprint" was presented by Weda et al.[36], whereby fingerprints (32 bit binary string representing 11.6ms of audio) would be created and compared to other recordings. The study found that perfect synchronisation could be achieved with videos of at least three seconds overlap of a single audio signal. Casanovas et al.[37] included both the audio and visual features in their multi-camera synchronisation strategy, with the emphasis being that audio-visual events (those visual events that included an audio feature) would provide better anchor points to identify temporal shifts (offset between frames). This did however add a dependency on those visual features related to "sharp" audio spikes actually being identified. Across data-sets of footage for concerts, basketball matches and even office spaces, the study reported admirable performance with precision of about 94% and recall of 83%.

While each of these studies presented valuable techniques for dealing with synchronous issues through time, the research only accounted for predictable drift between two separate recordings (from different frame rates) or from different start times of footage - Misalignment due to unpredictable events throughout the run-time of the footage was never addressed

4.3.2 Our technique

Due to the time constraints of this project, only a simple synchronisation strategy was implemented. This was based entirely on the time in which bounces were detected in the primary and secondary cameras. For example, if the primary camera detected a bounce two frames ahead of the secondary, then for the next two frames considered in overall video, only frames from the secondary camera were considered. This way, the secondary camera was effectively "fast-forwarded" to align with the primary. Since this method only based its synchronisation on a single event that took up a very small number of frames through the entire run-time of the footage, the videos still tended to drift apart in a way that could not be predicted.

As such, to implement a truly functional synchronisation scheme, other visual (or audio) events would have to be considered (see: 6.3).

Chapter 5

Evaluation

Having described the development of our system, from the implementation of a Court Detection system, to the method behind a single and multi-camera ball tracking and bounce detection procedure, the next logical step is to discuss the performance of each. In this section, we will begin by describing the process in which we recorded footage and generated ground truth. Next, we will outline the various errors intrinsic to the problem of Mobile Tennis Analysis for the casual game, as well as those errors associated to our choice of devices and methods. Following this we will provide our metrics used to assess the performance of our system, describe our testing procedure, as well as provide a complete overview of the numerical results we have found.

5.1 Test Footage

In obtaining footage for testing, a mobile phone was placed on top of each tennis post, pointed towards the centre of the baseline on a given side of the court. The "primary" device used was a OnePlus 8t, while the secondary was a Samsung A51. Each phone was set to record at 1080p in 30fps with their respective ultrawide lenses. The device's positions were fixed using a flexible tripod, while (close to) simultaneous commencement of recording was helped by using Bluetooth clickers. The tennis play itself was at a beginner-intermediate standard - At 30fps there was never a stage where the ball's speed impacted its ability to be tracked. Footage was recorded in two courts, which will be denoted as Court A and Court B respectively (see: 5.1).

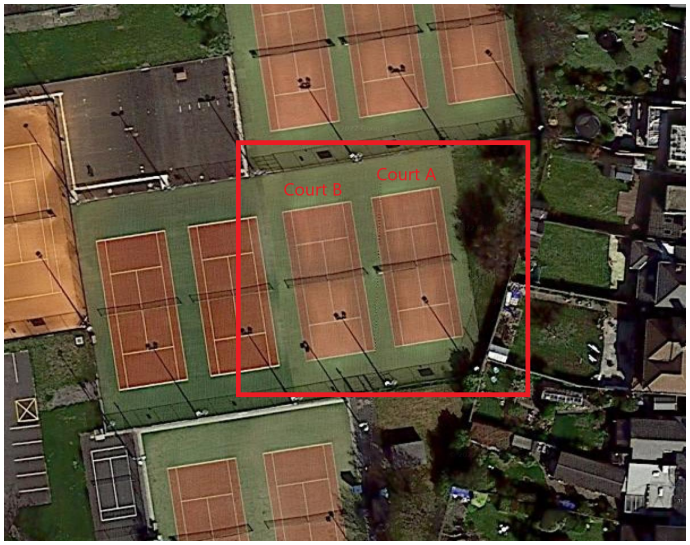


Figure 5.1: A view of the courts used in testing. Image credit: Google Maps

5.2 Ground Truth

In order to carry out testing, we first required ground truth to compare the output of our algorithms against. For this, a custom testing framework was designed and built to allow for the generation of ground truth as well as the testing of our system's performance. This truth was accumulated by surveying both friends and family (four in total), and collecting their interpretations of the exact bounce location within our recorded footage. For each piece of recorded footage, ground truth was recorded for the primary, secondary and top-down views. I.e. For each piece of footage, four interpretations of each bounce were sampled, in both the primary, secondary and top down views. Ground truth for Court Detection was also determined in the same way, though with just a single interpretation of each point.

The comparisons were made in each of the three domains:

1. Primary View Ground Truth vs Bounces detected in the Primary View
2. Secondary View Ground Truth vs Bounces detected in the Secondary View
3. Top down View Ground Truth vs Bounces detected from the Weighted Average Interpretation, and the Midpoint Interpretation.

To make sure that the truth across all three domains could be compared, the primary and secondary bounces, as well as the Ground Truth were also mapped to the top down view prior to comparison. This way, we could explore the similarities and differences between the performance of each of the four types of bounce detections (primary, secondary, midpoint and weighted average).

Our testing framework was implemented with C++, while our surveying tool was built with Python 3.10. The latter tool was built to output a series of CSVs for the observed bounces in the primary,

secondary and top down views. The former testing framework then accepted these CSVs and compared each of the detected bounces against their respective truth. In a similar way, ground truth was also recorded to compare the localisation of court points in our Court Detection system. Both tools used OpenCV.

5.2.1 Reliability and Intrinsic Error

As with any survey, we can expect some level of human error and inaccuracy related to the recorded ground truth. Given a piece of recorded footage, the participant was asked to click the exact location that they perceived the ball to bounce at. While we ensured that the recorded footage itself was played at a slow enough rate (60 millisecond delay between each frame) to allow for reaction times, we still saw variation in the location recorded by the participant. The geometry of the cameras themselves relative to the court, also contributes both to the reliability and the associated error of the ground truth - We found that participants would regularly interpret balls to have bounced on the wrong side of the center service line 5.2 5.3. When this was reviewed later, we ourselves found it difficult to determine which side of the line the ball had actually landed on. In both pieces of recorded footage, the true bounce was not visible in a single frame. Instead, the bounce was "missed," and the prior and subsequent frames would trick the eye into interpreting the bounce point wrongly. Interestingly, the bounce point localisations for both of the single camera views as well as the top down view were all noticeably different to their respective ground truth. This raised the important observation: In recording ground truth for an application such as this, the devices used in recording should not be subject to the same errors encountered by the devices in which we hope to test.

Figure 5.2: A perceived bounce from the perspective of the left-side view

Figure 5.3: That same bounce, from the right-side view

Similarly, in recording ground truth for the top-down view, we were unfortunately forced to accept that a significant amount of error would be present in the recorded truth. Since we had no real-life birds-eye perspective of the court to offer authentic ground truth, the closest we could come to obtaining truth was to manually record as close to perfect court points as we could, then perform a perspective projections on each view, blend together the resultant frames and record truth as the points in which the ball was seen to bounce. This is far from ideal, since in recording ground truth we

are again including a significant amount of error associated with our ability to select the appropriate court points, as well as other errors associated in the production of the top down view (see: 5.3.1). In addition, the two projected tennis balls often do not converge to a single point, resulting in the participant of the survey having to make a rough estimation of where they believe the true bounce location to be.

5.3 Sources of Error

5.3.1 Sensor Distortion and Error

Before considering the error intrinsic to each of the steps in our Bounce Detection system, we must first consider how our choice of equipment introduces error - In this section, we will highlight errors related to sensor distortion, camera movement, synchronisation issues, resolution and colour clarity. Such errors are unavoidable to vision applications on the mobile platform. Those same errors would largely be mitigated or entirely non-existent in a professional setting with higher-end equipment.

Image Distortion

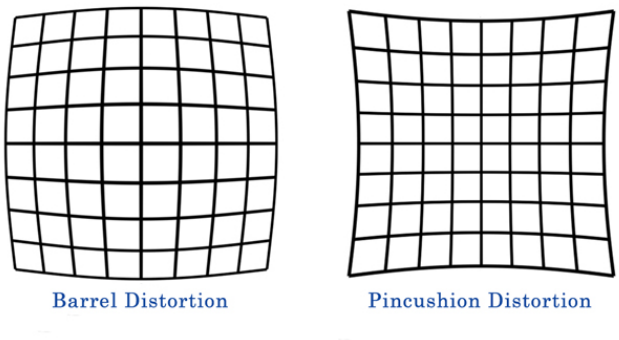


Figure 5.4: An illustration of lens distortion. Image credit: clickitupanotch

When using any type of wide lens, image distortion 5.4 is likely to be visible along supposedly straight lines. This effect is produced as a bi-product of the lens attempting to absorb as much of the scene within view to a single frame [38]. While the post-processing applied by the camera software in mobile devices will fix much of the distortion in the image, we can still observe some distortion in individual views, as well as the top down (plan) view (see: 5.5 and 5.6). In our case, neither the camera software nor the sensors themselves were the same in each device, as such we observed differing and inconsistent effects of distortion in the combined view. Unfortunately, by relying on ultrawide lenses we are introducing some amount of error into how accurate our line-calling (determining whether a ball is in or out) ability would be for a ball bounce in the top-down view.

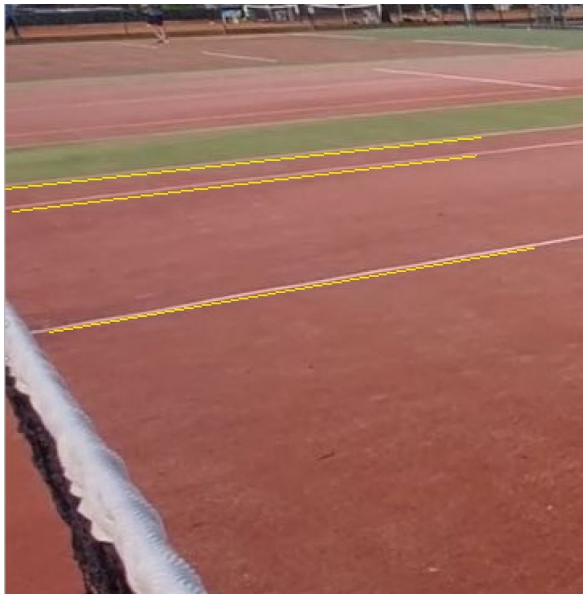


Figure 5.5: Image distortion visible from the secondary secondary camera, where the yellow lines indicate true straight lines.

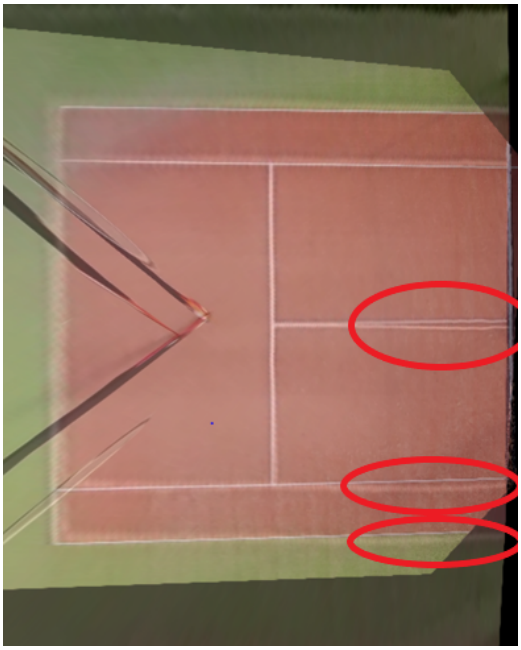


Figure 5.6: The same distortion visible from the top-down perspective. In this perspective the secondary camera is on the upper tennis post.

Camera Movement

While the flexible tripods used to secure each mobile device provided ample stability in the recorded footage, we did see subtle movements of the camera views through time. This movement was exaggerated when the ball hit the net itself, as well as when a gust of wind blew through the court. Again, we see how the application of tennis analysis to a casual setting faces many obstacles that could be ignored in a professional setting. While the movement of cameras through time immediately reduces the accuracy of the located court points and hence the perspective projection applied to the court, we describe how a system designed to track these court points would mitigate the error introduced: 6.3.

Synchronisation Issues

As described in 4.3, we saw how the usage of mobile devices for the analysis of video footage is an especially difficult task due to the design choice of variable frame rates for video in a mobile setting. This unpredictable behaviour is also compounded by having two separate mobile devices with largely different software and hardware platforms. While a small difference in the alignment of frames between the two separate videos would not throw the system completely off in its detection of bounces, larger separation over the course of a number of minutes would result in mistimed interpretations of bounces. Additionally, a more exaggerated difference in alignment without a dedicated synchronisation system means that a history of events needs to be stored within each device to ensure that those same events can be compared when possible. Since we also rely on the top down view to create ground truth

for our system, the existence of a sychronisation issue between the two frames inevitably results in improper ground truth recorded for later testing.

Colour and Resolution

In using two different camera sensors, we are also establishing some risk in how capable our system is in its ability to analyse a given scene. Our primary camera, the Oneplus 8t, features a higher megapixel (16MP) ultrawide sensor than the secondary's 12MP equivalent. The difference in the two is significant, and presented issues in how able the system was in its ability to extract information from the scene. In the screenshots below 5.7 5.8, taken during an overcast day, we can see that despite there being adequate contrast between the orange court and the white lines, the secondary camera would incorrectly process the colours within the scene, and blend the colour together. The same effect visible in the primary camera, though to a lesser extent. It is worth noting that that both cameras record at the same resolution of 1080p. At this resolution, the court points on the far side of the court suffer greatly in terms of the sharpness between the court colours and overall quality of the picture. We believe that this resolution is insufficient for the application of a truly accurate Mobile Tennis Analysis System.

Figure 5.7: A zoomed-in image of the far-side court lines, as viewed from the primary camera.

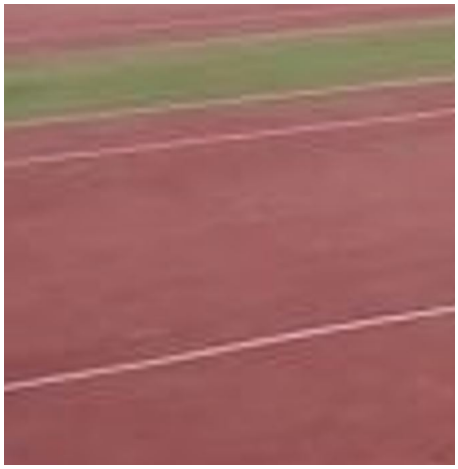


Figure 5.8: A zoomed-in image of the far-side court lines, as viewed from the secondary camera.

Additionally, the secondary camera would often misinterpret the colour of the moving tennis ball entirely. In footage recorded in a high-lighting scenario, the ball was often displayed as a white circle instead of the expected yellowish hue.

Figure 5.9: The ball processed as white by the Secondary Camera.

5.3.2 Court Detection Error

In carrying out Court Detection, the system performed the following three steps, depending on the point we are searching for: Court Point Estimation; Initial Refinement; Second Refinement. Here, we will discuss the error that featured in two of these steps:

1. **Court Point Estimation:** Often when seeking to the find each court section, and their respective extremities, the court sections themselves would contain noise in their colour that results in misshaped outputs. This was particularly noticeable during days with poor lighting. It was found that on the far side of the court, in all conditions, the number of pixels per square centimetre of the image was simply not sufficient, often resulting the blending of the white, orange and green court colours 5.8. This poor resolution meant that the far-side court points would regularly be thrown off their intended location 5.10 5.11.

2. **Initial Refinement:** Having made an estimation of each court point, the system then made a refined estimate of the location by taking each neighbouring contour, extracting a line along each of these contours, and finding their intersection. The error in this step was found to rely on the choice of the threshold when using RANSAC. I.e. the maximum distance that points can be included in consideration to a candidate line. It was found that a threshold of seven pixels yielded the best results.

Figure 5.10: Noise visible between the white, orange and green boundaries, giving incorrect interpretations of the initial estimated far-side court point.

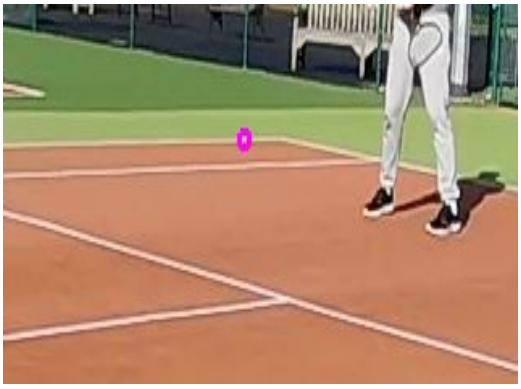


Figure 5.11: The far-side court point

5.3.3 Bounce Detection error

Missed Ball Detections

The primary source of error in a single camera view was missed ball detections. If the detection of a ball failed, it would happen that the bounce detection would either fail entirely out of the system being unaware of the ball's location, or in the event of a few missed detections, the bounce point would be misinterpreted. For example, if two of the immediate ball locations following (or prior to) a bounce are not detected, then those points that are eventually detected will be used to identify the bounce point, resulting in an improper localisation.

In testing, we found that the ball would be missed from detection in high-lighting (sunny) scenarios, and when the ball was passing the sky or other low contrast backgrounds 5.13. At the time when footage was recorded during the fair-weather day, the sun was facing the player. As such, the side of the ball facing the sun appeared pure-white, and thus escaped detection 5.9.

Multi-View Bounce Detection Error

In the top down view, errors were carried forward from the Court Detection and Single-View Bounce Detection steps. In addition, a significant amount of error presented itself at this stage from our choice of obtaining the weighted average of the two interpretations of each bounce 4.2.2. In the high-lighting (fair-weather) scenario, the secondary camera would often struggle to correctly detect the ball's bounce location if it was on the opposite side of the court. What we found was that our linear approach to determining the weighted average bounce location placed too much of a weighting on the smaller ball sizes detected, causing the weighted average location to be thrown off by weaker detections made by a camera further away from the ball 5.12. To summarise, it was intended that the weighted average bounce points would avoid the issues that bounces from the midpoint of each individual view would face. Unfortunately, by only considering a linear method of creating the Weighted Average, we

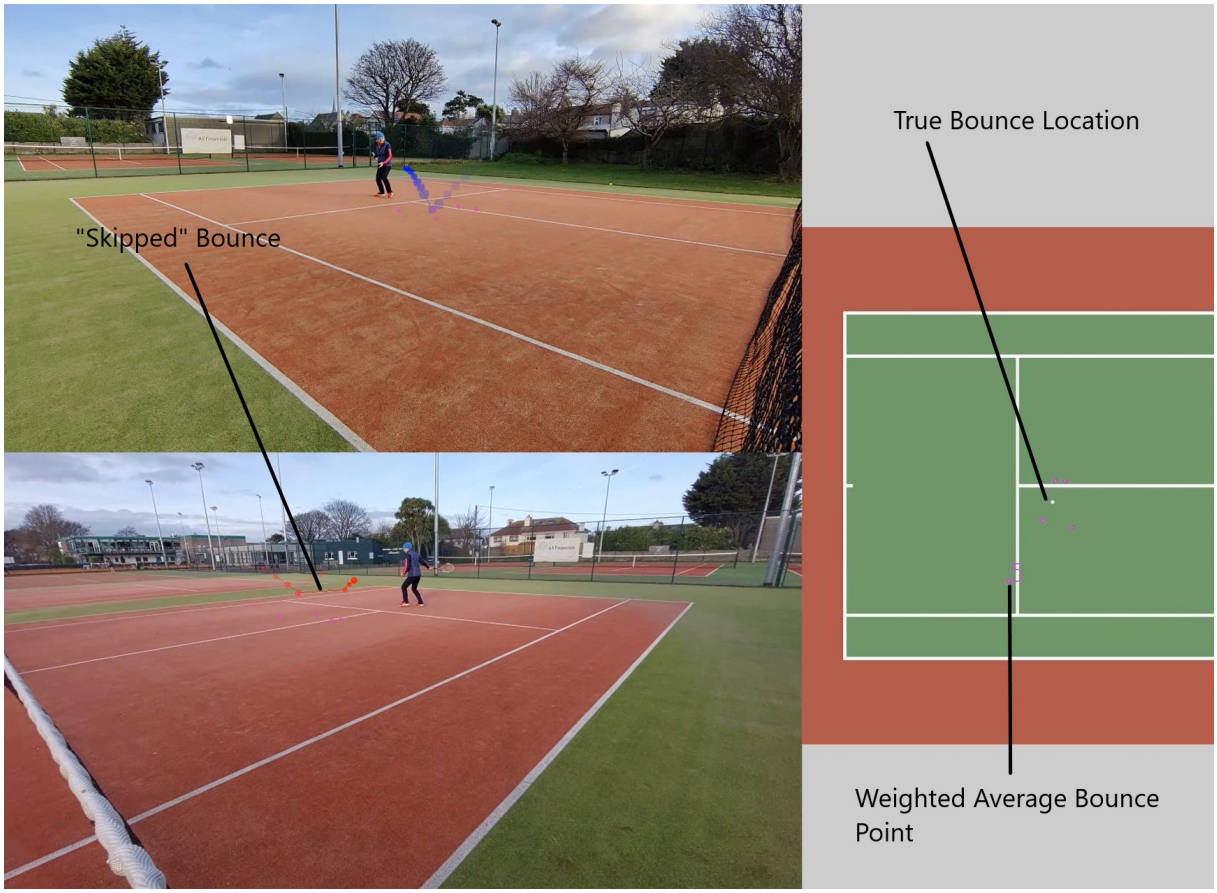


Figure 5.12: An example of a poor detection in the secondary camera, significantly worsening the accuracy of the Weighted Average Bounce Point

Figure 5.13: A ball (circled manually in pink) rendered nearly invisible against its background.

allow for incorrect interpretations to carry a significant impact on the result. The perspective issue of correctly localising the bounce location in individual views (as discussed in: 5.2.1), also has an impact on the bounce points determined using multiple views. In an ideal scenario, each of the two individually projected balls in the planar view would converge to a singular point, allowing for a quick

and simple determination of the associated bounce point in that view. Unfortunately, the perspective issue seen in individual views was also visible in the planar view, resulting in a much more ambiguous location of the true bounce point. This ambiguity also meant that Ground Truth determined in the Planar view also suffered significant inaccuracy 5.14.

Figure 5.14: An example of the two plan-view balls not converging to a point.

5.4 System Performance

5.4.1 Metrics

As a measure of performance of our Court Detection system, we used the euclidean distance (in pixels) between the detected point and the true point as the error of a detection.

To Measure the success of our Bounce Detection System, we considered two aspects: A) Whether a bounce wass detected within a similiar time as the ground truth (classification). B) How accurate was the bounce localisation.

With respect to a single view, we considered a True Positive (TP) to be a bounce detected within five frames of the mean frame number (out of the four recorded instances of ground truth) for that particular bounce (in that particular domain). If the bounce were to be detected, but outside of the five frame threshold, then this was referred to as a False Positive (FP). If there was no bounce detected at all, then a False Negative (FN) was assumed. The obvious omission here is a true negative (TN), but in our use case we were not interested in the correct negative classifications of bounces, thus we omitted this figure from consideration. Otherwise, it would be calculated as all those frames in a piece

of footage that do not contain bounces. Using the figures mentioned above, we used the following metrics to quantify the performance of our Bounce Detection System:

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

Precision gave an impression of the ability for the Bounce Detection system to detect bounces within five frames of the recorded ground truth, while accuracy took into account those bounces that were missed entirely, thus giving an overall picture of the system's performance.

In terms of the accuracy of detected ball bounces, we observed the following procedure:

1. Calculate the Standard Deviation of each bounce in the Ground Truth, for each view. This involved finding the mean observed bounce, out of the sample size of four, and calculating the standard deviation, for each of the primary, secondary and top down domains (views).

2. Calculate the x and y coordinates from the mean truth, to the detected location for that bounce. I.e. ((303, 295) – (295, 286)) to give a set of coordinates (8, 9). This was carried out for each type of detected bounce (primary, secondary, midpoint of primary and secondary, weighted average of primary and secondary).

3. Using this list of coordinates for each type of detection, plot the resultant points.

4. Take the mean of the standard deviations calculated for each view (from step 1), and plot the 1st and 2nd standard deviations.

5. For further information, calculate the median of the error between the detected bounces and their related mean truth.

5.4.2 Court Detection Performance

When testing court detection, we considered two separate tennis courts, Court A and Court B. Court A was located beside trees and foliage, with a noticeably lower quality to the court colours. This court was recorded in both sunny and overcast weather. Court B, a newer court, was surrounded by other courts. There was a much stronger contrast between the white, orange and green of this court.

Below are the included results for Court Detection on a sunny day for Court A and B. As well the results for the detection on an overcast day for Court A.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	1824	480	1827	480	3
Near B	1649	456	1645	453	5
Near Mid	1443	519	1441	517	2.828427
Far Mid	745	385	739	384	6.082763
Far	986	370	962	369	24.02082
				Mean	8.186403

Table 5.1: Court B, viewed in sunny weather, with the primary camera.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	55	371	56	373	2.236068
Near B	235	363	232	366	4.242641
Near Mid	462	446	460	444	2.828427
Far Mid	1119	373	1115	372	4.123106
Far	894	337	880	336	14.03567
				Mean	5.493182

Table 5.2: Court B, viewed in sunny weather, with the secondary camera.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	159	473	158	472	1.414214
Near B	330	462	323	464	7.28011
Near Mid	541	537	550	537	9
Far Mid	1213	448	1220	450	7.28011
Far	980	419	973	417	7.28011
				Mean	6.450909

Table 5.3: Court A, viewed in overcast weather, with the primary camera.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	1662	365	1650	365	12
Near B	1511	356	1507	357	4.123106
Near Mid	1333	432	1320	431	13.0384
Far Mid	675	356	680	359	5.830952
Far	901	320	901	324	4
				Mean	7.798492

Table 5.4: Court A, viewed in overcast weather, with the secondary camera.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	51	500	49	500	2
Near B	237	484	232	486	5.385165
Near Mid	466	559	466	557	2
Far Mid	1152	457	1151	457	1
Far	914	431	913	430	1.414214
				Mean	2.359876

Table 5.5: Court A, viewed in sunny weather, with the primary camera.

	Truth		**Detected**		
	x	**y**	**x**	**y**	**Error (px)**
Near A	1756	461	1759	461	3
Near B	1595	443	1594	440	3.162278
Near Mid	1403	508	1403	506	2
Far Mid	750	395	748	396	2.236068
Far	972	374	968	373	4.123106
				Mean	2.90429

Table 5.6: Court A, viewed in sunny weather, with the secondary camera.

From each of these tables, a mean of means for error is given as: 4.99px

The above tables are included as a demonstration of the results of our court detection system, and the oftentimes unpredictable nature of Court Detection.

Court A in sunny conditions stood out with the best performance here. We believe the high accuracy recorded for this court is a consequence of the tree cover providing slightly lower lighting than Court B, which was recorded in the same conditions. The slightly lower lighting resulted in an ideal scenario where court lines were nicely contrasted against other colours, but not overexposed to the point where the boundaries of the lines carried noise. This noise was visible in the results for Court Detection on Court B, during the same sunny conditions. The poor performance in detecting the far point on this court can be explained by figure 5.17, where the extremity of the far region is mistaken to be the corner circled in the image below. Interestingly, the primary camera performed worse than the secondary for Court B, which might point out that despite the Primary ultrawide sensor being of markedly higher quality, both inevitably suffered from the same lack of resolution at the far side of the court.

The issues of resolution and noise in the processing of the court colour presented itself in the results for Court A recorded on an overcast day. Across both cameras, there are consistent small displacements in detected court points, stemming from the located lines in the first round of refinement. Due to the lower lighting, the sharpness of colours around the court lines gave a poor interpretation of the true orientation. These mistaken line orientations then resulted in an incorrect intersection taken as the second refinement. A second possibility for the poor estimations of court lines is the

existence of distortion in the image, though upon inspection there is no obvious effect near to the court intersections.

Figure 5.15: Located court points for Court A, taken from the Primary Camera on a overcast day.

Figure 5.16: Located court points for Court A, taken from the Secondary Camera on a overcast day.

Overall, the relatively high error throughout each of the courts suggests that a more accurate Court Detection methodology is required when dealing with so many sources of error. In particular, we believe a stricter approach needs to be taken in terms of fitting a court "model", instead of having to rely on locating points throughout the court amidst the impacts of poor resolution, colour processing and lens distortion 6.3. We believe that had we used higher resolution video from each camera (e.g. 2K video), the effect of whether the day was sunny or overcast would be lessened.

Figure 5.17: Wrongly estimated "Far" court point.

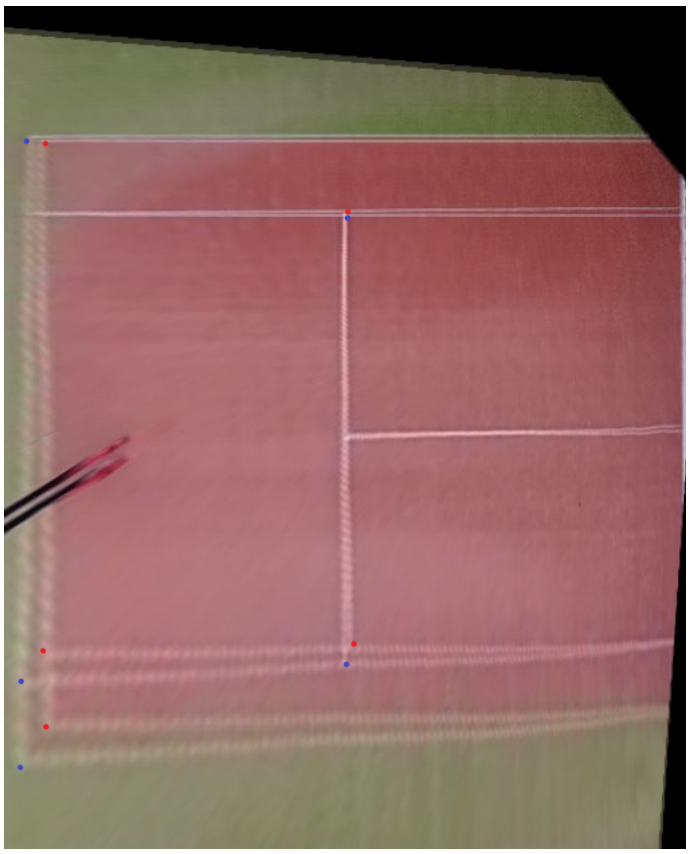


Figure 5.18: A display of the error between the "true" (red) and detected (blue) court points, for Court A, using the secondary camera on an overcast day. Note how for "far" points, a more exaggerated error is observed, rendering the plan view from these detected points unusable - Again indicating that a higher video resolution would be preferable

5.4.3 Bounce Detection Performance

Below are the results of our Bounce Detection System, for both single views, as well as for both interpretations in the top down-view (Midpoint and Weighted Average). In each case, we consider the performance of the system, both in its ability to detect the existence of bounces (within five frames of the recorded truth), as well as its ability to accurately localise those bounces. When carrying out

the perspective projection step for this portion of testing, we used the ground truth from our testing in the previous section. If we had used our own detected court points, the Bounce Detection system would have been almost unusable due to the high error observed 5.18.

Bounce Detection (Classification)

Overcast Weather

Table 5.7: Primary view classification performance in overcast weather

		Detected	
		Pos	Neg
Truth	Pos	17	1
	Neg	0	n/a

Table 5.8: Secondary view classification performance in overcast weather

		Detected	
		Pos	Neg
Truth	Pos	17	1
	Neg	0	n/a

Table 5.9: Top down (midpoint & weighted average) view classification performance in overcast weather

		Detected	
		Pos	Neg
Truth	Pos	11	1
	Neg	6	n/a

Table 5.10: Performance metrics for each view in overcast conditions.

	Primary	**Secondary**	**Top**
Precision	1	1	0.647
Accuracy	0.94	0.94	0.61

Sunny Weather

Table 5.11: Primary view classification performance in sunny weather

		Detected	
		Pos	Neg
Truth	Pos	15	0
	Neg	0	n/a

Table 5.12: Secondary view classification performance in sunny weather

		Detected	
		Pos	Neg
Truth	Pos	6	9
	Neg	0	n/a

Table 5.13: Top down (midpoint & weighted average) view classification performance in sunny weather

		Detected	
		Pos	Neg
Truth	Pos	11	0
	Neg	4	n/a

Table 5.14: Performance metrics for each view in sunny conditions.

	Primary	**Secondary**	**Top**
Precision	1	1	0.73
Accuracy	1	0.4	0.73

In overcast weather, both the primary and secondary cameras performed nicely, with just a single False Negative recorded overall. In the fused top down view result we noticed a markedly poorer precision and accuracy, caused directly as a result of the secondary video stream becoming misaligned

and recording bounces a number of frames later than the primary. This resulted in a series of False Positives recorded in the top down views (for both the midpoint and weighted average interpretations). Even with our implemented synchronisation system, the video stream from the secondary camera still managed to misalign. This result suggested that a more comprehensive Synchronisation Scheme was required.

In the fair weather (sunny) tests, we found an overall poorer ability to detect and track the tennis ball. As mentioned earlier, we believe this to be caused by the ball becoming overexposed in the high lighting and turning pure white in each camera view. Since our multi-view bounce detection system was able to accept a single view's bounce interpretation 10 frames later than the initial detection if the other view failed, there was inevitably a sizable delay in the recorded times for each bounce in the top down view. Overall, this resulted in the top down view suffering a poorer precision and accuracy than the equivalent overcast results. It should be noted that a synchronisation issue was not observed in this footage.

Bounce Detection (Accuracy)

To fully test our system's ability to correctly localise a bounce, we have accepted both True and False Positives as part of our accuracy testing. While the latter group are considered as misclassifications, the ability for the system to localise them close to the recorded ground truth is still important.

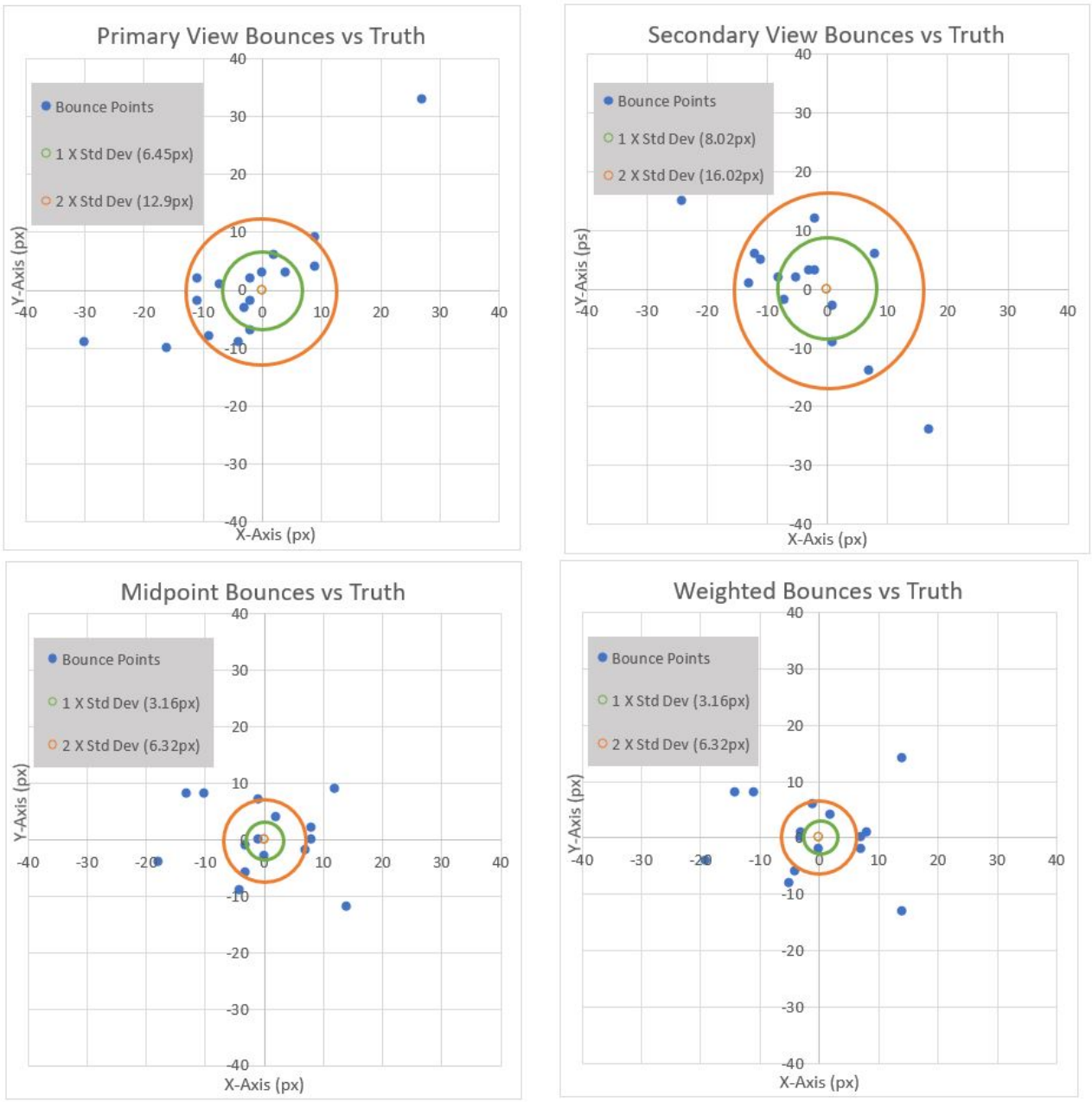


Figure 5.19: Court Detection accuracy (in pixels) for overcast weather.

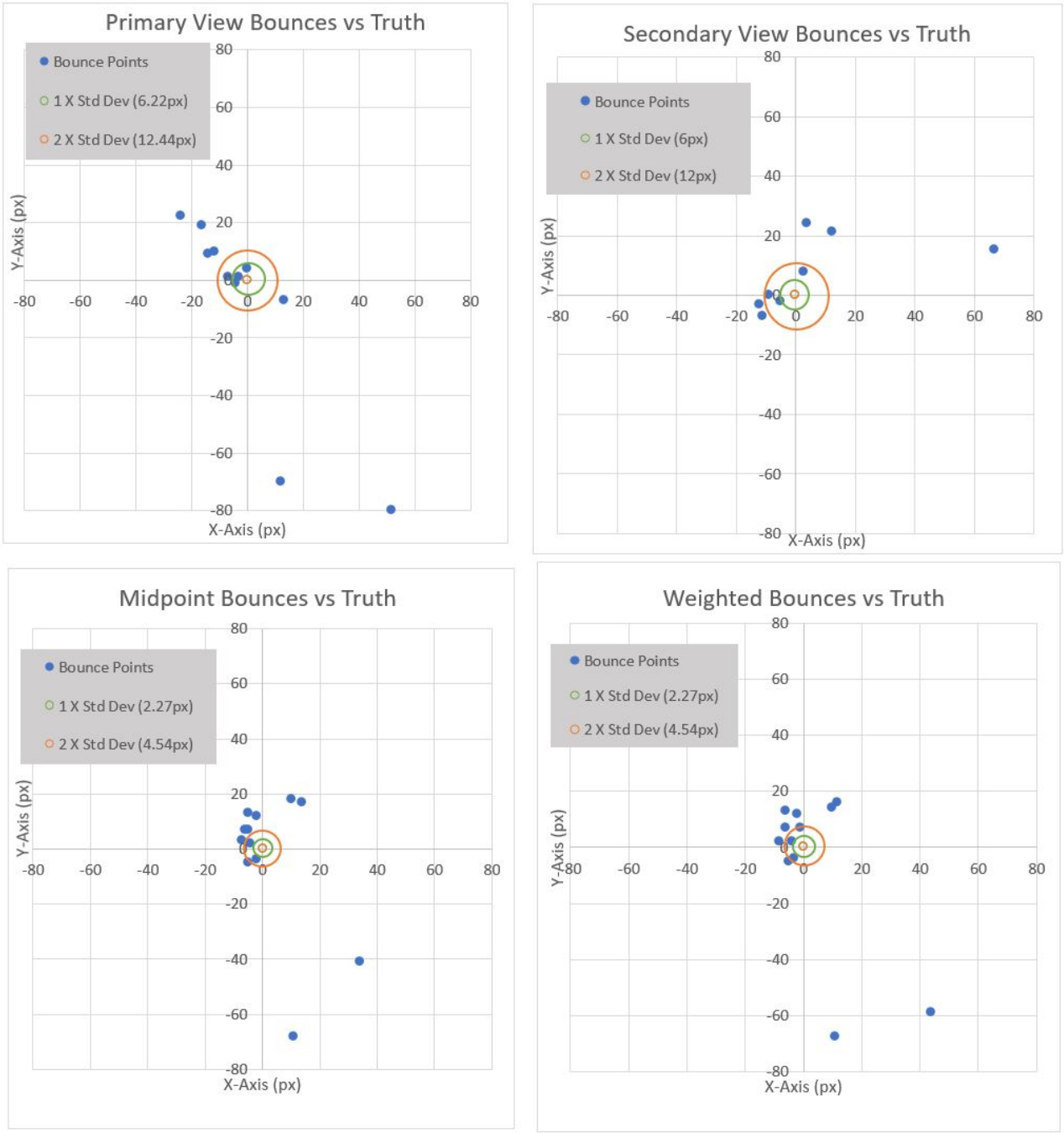


Figure 5.20: Court Detection accuracy (in pixels) for sunny weather

As an aside, the smaller standard deviation observed in the midpoint and weighted bounce plots can be explained by the way in which that truth was recorded, where the topdown view itself (ideally) presents two balls converging to a singular point, the bounce point to be selected by the participant is more obvious than either the primary or secondary views.

Throughout both the overcast and sunny groups of scatter plots, the effect that "skipped" bounces had as outliers in both the primary and secondary views was observed. Such extremities were far more noticeable in the latter group of plots, where due to the poor level of detection for the ball and its bounces, the "skipping" effect made frequent appearances. In a correct application, such

bounces would not be considered as detected bounces, but again, since in this section of testing we are interested in the system's ability to accurately localise bounces, it seemed appropriate that they be included. Also in both groups, we saw marginally better performance for the Weighted Average bounce interpretations, with more shots landing within two standard deviations. However, this difference in accuracy between this method and the Midpoint Bounce interpretation is too small to suggest much of an advantage in choosing one over the other. In order to mitigate the effect of poor detections from a camera further from the ball within the scene, we believe a stronger weighting than our own linear procedure is required. Alternatives to our own method could be a polynomial or exponential weighting scheme to better offset the effects of a poor secondary bounce detection. That being said, this would not be required if the level of detection throughout different lighting conditions was consistently high. In the sunny group of plots, the outliers are more numerous and extreme in their distance from the origin. Again highlighting the systems poorer performance in high-lighting conditions. Alongside these outliers, is a generally poorer performance when compared to the equivalent plots in overcast conditions - In the table below is a direct comparison between the median errors for each view. The median error was chosen as opposed to the mean in order to nullify the effect of outliers in both groups.

	Primary	**Secondary**	**Midpoint**	**Weighted Average**
Overcast	9.85	12.083	8.25	8.06
Sunny	15.62	12.0386	10.69	10.69

With these results, it can be noted that for sunny conditions: a) the median error from the origin is the same for both the Midpoint and Weighted Average observations; and b) the primary view seems to perform worse than the secondary. These results can be explained by the fact that in sunny conditions, it would happen that only one of the views detected a bounce, and as was mentioned in 5.4.3, if it is the case that just a single view detects the a bounce, then that location will be used by both the Midpoint and Weighted Average observations. As such, they share a number of bounce locations.

Throughout all of our results, it was also observed that the ball's distance from the net did not affect the level of error found in the detection. Across both overcast and sunny conditions, a mean R^2 value (measure of spread between the points and an overlayed regression line) of 0.05 and 0.295 were found respectively, with the latter exhibiting a trend of lower error further from the net. However, we don't believe this suggestion is significant. considering the small amount of data as well as the presence of large outliers. The scatter plots related to this discussion have been included in B.1.

Chapter 6

Conclusion

At the beginning of this report, we discussed how despite the popularity of Tennis Analysis systems such as Hawkeye in the professional game, there is still no truly reliable system available to the casual player (1). Following this, we discussed the relevant works in the areas of Tennis Analysis and Multi-Camera tracking, as well as the Computer Vision theories related to the development of our systems (2). With this mind, we proposed a full-scale Mobile Tennis Analysis System, capable of tracking a tennis ball and players throughout multiple views, with the intention of delivering functionalities such as the classification of shots made throughout a game (forehand, backhand, serve etc.), the localisation of bounces within the court, and the generation of useful performance-related statistics such as player or ball-bounce heat-maps. Other useful performance indicators such as shot-success/failure rates and the likelihoods of earning/losing points from a given shot were proposed. With this full-scale implementation in mind, we finalised our System Overview, with a more focused list of objectives for this project (3), aimed at analysing a single side of the court with two cameras. In the next chapter, we discussed the development of our Court and Bounce Detection systems, with each section featuring a brief overview of the related works (4). Also in this chapter, we highlighted our implementation of a simplistic view-synchronisation system - A problem we felt required a more robust solution in future. Finally, we provided a complete description of the variety of errors encountered both in relation to our choice of devices, as well in relation to our Court and Bounce Detection Systems. Which we followed, by providing the results of each of these systems (5).

Having reviewed the performance of our Court and Bounce Detection Systems, and highlighted the errors encountered in each, we can now conclude this report with a review and critique of our work with respect to each of our initially stated objectives.

6.1 Critique and Review

The detection of a tennis court within multiple camera views.

To begin, we believe that our implementation of a Court Detection system, despite the high number of errors present within the functionality, broadly achieved what we set out to do. Regardless, it

is important that we highlight how obstacles such as: lens distortion; camera movement; inadequate camera resolution; unpredictable colour processing affect our system. We believe that the first of these is solvable with an implementation to track court lines associated to each located point (see: 6.3). Lens distortion can also be addressed via the calibration [39] of each lens prior to video processing, though since we were focused on delivering systems that could be used by casual tennis players, calibration seemed beyond what should be included. The latter two however are intrinsic to our choice of devices, and would only be solved by choosing the highest quality devices available at this time. Despite our Court Detection System using two separate rounds of refinement, we still observed a global (across all points) mean error of about 5 pixels. While error is small in relation to a 1920x1080 sized image, there is a severe impact on the accuracy of the planar view of the court. With respect to our own implemented methods locating the court, the usage of heuristics to identify each court section and to extract the initially estimated court points, meant that the adaptability to other scenarios may not always be guaranteed.

The detection and tracking of a tennis ball within multiple camera views.

In terms of the detection of the moving tennis ball through multiple views, we are satisfied with the performance of our system. Since we had neither ground truth of whether the ball appeared within a scene, nor of where it appeared, we had no way of testing the system's ability to localise the ball within a scene. While this was not ideal, the primary focus of our system was to localise bounces, and as a bi-product of testing our Bounce Detection System, the ability to correctly detect the ball location was also tested. Nevertheless, the weather conditions and background had a significant effect on whether the ball could be consistently detected. And as we saw in high-lighting (sunny) days, the ball would often disappear entirely as it moved through the scene. As a critique however, the usage of RANSAC to better identify the ball as a moving circle within the scene would have been a valuable addition. Also, the over reliance on the physical features of the ball meant that detecting the ball was especially difficult given its appearance in high lighting. As such, a more suitable system might have been to track the ball based only on its shape, and *how* it moves through the scene. I.e. the ball should take a somewhat predictable path through an image, setting it apart from most other moving objects. Finally, since our system was only tested on footage of play at a beginner-intermediate standard, we have no guarantee that the same approach or choice of devices would carry the same results for a higher play standard. In fact, it is likely by recording at 30fps, the ball will become untrackable as it will resemble more of a greenish ellipse within frames.

The determination of bounces points, through multiple views.

Our primary and overall objective of this project was to implement a multi-camera bounce detection system for mobile devices. We believe that our work to develop and explore the implementation of

this system has been successful. In working towards the development of this system, we introduced the idea of using multiple mobile camera views to determine bounce points in a top down, planar view. We also highlighted the various issues and shortcomings associated with the Bounce Detection system. In particular we brought attention to how the misalignment between separate camera streams resulted in poor classification accuracy and precision, with results 0.61 and 0.647 recorded respectively. This de-synchronisation affect came into play rather unpredictably, and resulted in a situation where we were totally at the mercy of whether the variable frame rate effect decided to take hold during recording. As a large source of ambiguity in the determination of bounce points, we found how each individual view would often have a different interpretation of the location of a ball bounce, which meant that in the associated top down perspective, the two balls would not appear to converge to a point. This effect exposed the difficulty in determining the true bounce location, and added further levity to the motivation of finding a truly reliable bounce detection system. We saw how our choice of applying a linear weighting in finding the weighted average, resulted in poorer detections as a result of weaker detections made by a camera further from the ball being weighted too strongly. Additionally, we saw how the ball's distance from the tennis net did not greatly effect the accuracy of the system in localising bounces, rather it was the system's ability to consistently locate the moving ball in different environments that affect the accuracy, irrespective of distance from the net. As a significant critique of our approach, the lack of correct ground truth in the evaluation of our systems meant that the testing that was carried out, was put under significant doubt since the accuracy of the recorded truth was under serious doubt. Given more time for this project, a far more suitable alternative to gathering ground truth would have been to position a high-quality camera at a relatively high angle to the court, where a more accurate picture of the location of bounces could be ascertained. Finally, and importantly, we highlighted how our own method of gathering ground truth using our imperfect camera setup is not a practice that should be repeated in future. Instead, we ascertain that it would be far more useful if a separate video camera, of higher quality than 1080p, be positioned at a height relative to the half of the court under question. This way, perspective issues in the bounce locations, as well as resolution problems for balls at a higher distance from the mobile phones would be avoided in the recording of ground truth.

6.2 Conclusions

As of writing this report, despite the high availability of Tennis Analysis systems to the professional game, there is still no true Mobile Tennis Analysis System available to the casual tennis player. With this in mind, we proposed a multi-camera Tennis Analysis system for mobile, which would utilise up to four mobile phones to generate a complete overview of a players' performance throughout a game of tennis by detecting ball bounces and carrying out shot classifications.

Having carried out a thorough evaluation of our implemented systems, we can now address the question: *Can the Mobile platform be comfortably used in the implementation of Tennis Analysis?* As we discussed in 5.3.1, by choosing to use mobile phones in our system, we are exposing ourselves to a host of errors, all of which affected our own systems, and would also cause issues in a future full-scale implementation - Fundamental requirements such as being to judge whether a ball is in or out, or being able to accurately specify the moment in which a ball struck the court, would be greatly troubled by variable frame-rates in mobile devices, as well as insufficient resolutions and distortion existing across supposedly straight lines. Overall, with our chosen devices and video resolutions of 1080p, we believe that until those errors are close to negligible, such a system would not be suitable for a genuinely trustworthy Mobile Tennis Analysis System.

As an interesting point of discussion, it is also worth considering the extent to which an implemented Tennis Analysis System can be trusted in its ability to officiate a game through the determination of bounce points and other game-related statistics. At the onset of the Covid-19 pandemic, Hawk Eye Live was granted complete officiating control over line calling in tournaments such as the US and Australian Open. Unfortunately, the ability for this technology to accurately officiate the game has since been a regular source of complaint throughout the tennis community [40] [41]. While the sophistication of Hawk Eye remains exemplary, there still does not exist proof that the system can be wholly trusted to give accurate line calling abilities. With such a lack of transparency, the question about whether *any* Tennis Analysis System can deliver truly accurate capabilities.

In spite of this, we are happy with our systems' ability to carry out each of our previously outlined requirements. In building our implementations, from Court Detection through to Multi-Camera Bounce detection, we not only explored the performance of our own methods to achieve our outlined objectives, we also gained an insight into the various obstacles that are unavoidable and intrinsic to the problem of Mobile Tennis Analysis. it is the view of the researcher that having carried out this research and development, and having exposed the large amount of errors present in working with this type of application, (something that is more often avoided in discussions for similar applications) this report is ultimately a valuable piece of research.

6.3 Future Work

Since this project considers just the Court and Bounce Detection systems of the overall suggested implementation, we have the following suggestions for future work, both for the development of the full-scale Mobile Tennis Analysis product, as well as in relation to the obstacles highlighted throughout this report:

1. **Higher Resolution-capable mobile devices:** As was discussed in 5.3.1, a significant issue faced in the development of our Court Detection system was the insufficient resolution for ul-

trawide video recordings. Whilst 1080p has been the industry standard in video for some time now, we saw how the low pixel density around the "far" Court Point led to difficulty in detecting those court points, as well as an exaggerated effect in the top down (plan) view of the court. Thus, in order to develop a truly functional system for Tennis Analysis, the usage of devices with higher resolution capabilities would be preferred.

2. **Court Detection via a preset Court "Model":** In addition to requiring a higher image resolution, we believe the Court Detection System also requires a stricter more repeatable technique for determining the court within the scene. For this, the application of a court "model" is proposed, whereby using the initially estimated court points, a preset court model could be overlayed onto the estimated court points, and refined to the point where the number of white court-line pixels covered by the overlaying model is maximised. Such an approach was first proposed and followed by Mora [42].

Figure 6.1: An example of the Court Model technique observed by Mora

3. **Synchronisation System:** In 4.3, we discussed how given two mobile devices, a number of variables such as the start time of the video streams, and (unpredictably) the variable frame-rate design choice by manufacturers, results in a situation where we can expect misalignment between multiple videos of the same scene. In order to address this, we believe a more comprehensive approach is need to be taken for synchronising the view - Instead of just relying on the bounce to synchronise video streams, other events such as sounds in the court, player movements or events on the opposing side of the court could be used as a means of re-aligning footage.

4. **Court-point Tracking System:** As was discussed in 5.3.1, a significant issue faced in this project is the movement of cameras during recording. This error resulted in the incorrect perspective transformation of each individual view to the planar view. As a suggestion for future

work, we propose a method of tracking each located Court Point, by searching perpendicularly along the previous court line locations until those same lines have been relocated. This way, the implemented system can maintain its accuracy to the originally located court points.

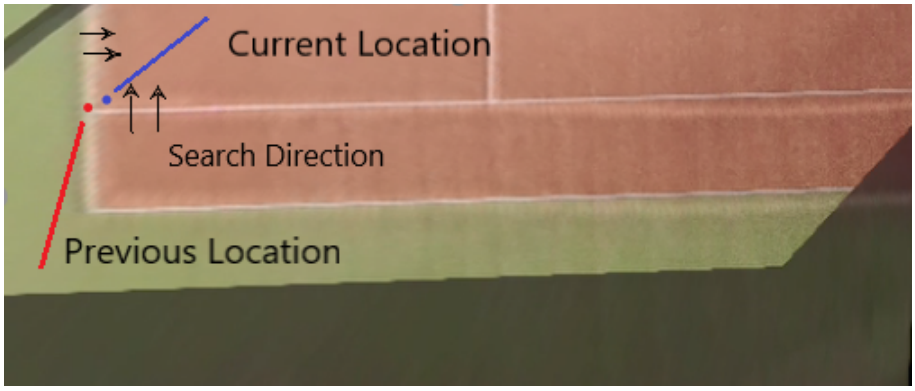


Figure 6.2: An illustration of the suggested Court Update procedure.

5. **More reliable Ground Truth:** In testing our Bounce Detection system, we required the use of manually chosen Court Points to generate our Ground Truth. This method of creating our truth was inherently flawed, and a far more suitable alternative for future work would be to record Ground Truth from a high quality camera, positioned looking downwards at the court.

6. **Ball Height Determination:** At the earlier stages of this project, it was proposed that part of this research investigate the determination of the ball's height above the court, using the two fused planar views of the court. In an ideal scenario, without any synchronisation or perspective issues, the two projected balls will approach and converge to a singular point. As part of this process, the two balls can be seen to take a linear path, as their distance from each other gradually closes to zero. With this information, it is feasible that the balls' distance apart, could be modelled with the true ball's distance above the ground, so as to give a robust technique of ball height determination at any point in the court. Unfortunately, this task is made more difficult by the synchronisation issue we have discussed in this report, as well as the fact that the projected balls will regularly not converge to a singular point 6.3.

7. **Player Tracking and Shot Classification:** A useful addition to this project would have been the implementation of a Player Tracking and Shot Classification system. As a bi-product of our Bounce Detection System, the player moving throughout the scene was visible within the foreground 2.12. This player could be tracked based on the colour of their clothes, and using Optical Flow 2.3.8, a system could be built to consider the flow vectors around the player's racket-hand. For example, a movement of the player's racket-arm across their body from right to left moving upwards would be a forehand shot, while the opposing motion would be a back-hand. This is obviously over-simplified since we are assuming that the player is right-handed, however

by looking at the length of each upper-limb of the player, it would be possible to determine the hand that the player chooses to hold their racket in. Such a system would certainly be within the scope of future work.

8. **Player Performance Reports:** As was discussed in the System Overview Chapter, a popular addition to full-scale Mobile Tennis Analysis Application would be player movement and ball bounce heat maps, as well as compiled information relating to how a player responds to (or carries out) certain types of shots. With the current implementation, heat maps of bounce points throughout the court would be trivial, however in order to deliver information relating to how well a player performs with different shots would require the *opposing* player to be tracked simultaneously. Which naturally leads to the next suggestion for future work -

9. **Four-Camera Tennis Analysis:** In the proposed complete-implementation, we described the use of a maximum of four cameras in the case of a doubles game. As a piece of future work, the initial task of implementing the same Court and Bounce detection systems for an additional two cameras would not be much of a challenge, however the challenge of synchronising each of the four views to (roughly) the same scene would prove difficult. Overall, the addition of an additional two cameras for the opposing side of the court would allow for a more complete outlook of events throughout a game, as well as the generation of information relating to players' ability to earn or lose points, given a type of shot.

10. **Real Time Capabilities:** For the implementation of this project, developing a system capable of analysing the scene in real time was not one of our priorities. Having layed out a foundation for future works in this area however, the development of a more optimal application, capable of carrying out its analysis on a game of tennis in real time would encourage its later production as a mobile application. The use of GPU acceleration with OpenCV, as well as the use of pre-trained Machine Learning models would be important inclusions in working towards the development of a real-time system.

11. **Mobile Development and Optimisation:** Finally, we hope that our Court and Ball Detection Systems, as well as our proposed full-scale implementation, encourage future work towards the development of Mobile Tennis Analysis System, implemented on either Android or iOS. Such an application would be capable of each of the functionalities highlighted in 3 - Ball tracking and bounce detection through multiple views, player tracking and Shot Classification through multiple views, score keeping and line calling. On top of this, the application would be capable of delivering useful performance related statistics and visualisations, as mentioned above. A significant portion of work for development of this system for mobile, would be the communication between multiple devices to accurately distribute a complete knowledge base of the game

at-hand, throughout a maximum of four devices. For this, we suggest a device is allocated as the "lead" device within a session to compile the information found in each of the others, where a secure channel of communication is established between these devices through the use of WiFi Direct.

Figure 6.3: An example of the two projected balls converging to a single point.

Bibliography

[1] Mar 2018. URL https://onlinemasters.ohio.edu/blog/how-technology-is-revolutionizing-sports-training/. Accessed: 2022-03-07.

[2] Feb 2021. URL https://www.espn.com/tennis/story/_/id/30877297/hawk-eye-live-gains-more-support-australian-open. Accessed: 2022-03-07.

[3] Kevin Loria. Science is creating super-athletes — and making sports unrecognizable to previous generations, Aug 2015. URL https://www.businessinsider.com/how-science-and-technology-are-changing-sports-2015-8. Accessed: 2022-03-07.

[4] Robert Wood. Hawkeye tennis line-calling system, Feb 2022. URL https://www.topendsports.com/sport/tennis/hawkeye.htm. Accessed: 2022-03-07.

[5] n.d. URL https://swing.tennis/. Accessed: 2022-03-07.

[6] Rafael Martin Nieto and Jose Maria Martinez Sanchez. An automatic system for sports analytics in multi-camera tennis videos. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2013. doi: 10.1109/avss.2013.6636679. URL https://dx.doi.org/10.1109/avss.2013.6636679.

[7] Wanneng Wu, Min Xu, Qiaokang Liang, Li Mei, and Yu Peng. Multi-camera 3d ball tracking framework for sports video. *IET Image Processing*, 14(15):3751–3761, 2020. ISSN 1751-9659. doi: 10.1049/iet-ipr.2020.0757. URL https://dx.doi.org/10.1049/iet-ipr.2020.0757.

[8] Banglei Guan, Yingjian Yu, Ang Su, Yang Shang, and Qifeng Yu. Self-calibration approach to stereo cameras with radial distortion based on epipolar constraint. *Appl. Opt.*, 58(31):8511–8521, Nov 2019. doi: 10.1364/AO.58.008511. URL http://www.osapublishing.org/ao/abstract.cfm?URI=ao-58-31-8511.

[9] N. Owens, C. Harris, and C. Stennett. Hawk-eye tennis system. In *2003 International Conference on Visual Information Engineering VIE 2003*, pages 182–185, 2003. doi: 10.1049/cp:20030517.

[10] Oct 2021. URL https://www.hawkeyeinnovations.com/index.html.

[11] Martin Breheney. Huge annual costs of hawk-eye technology a major turn-off for gaa, Jun 2011. URL https://www.independent.ie/sport/gaelic-football/huge-annual-costs-of-hawk-eye-technology-a-major-turn-off-for-gaa-26755869.html.

[12] Oct 2021. URL https://www.apple.com/ie/iphone-13-pro/.

[13] Ciaran O Conaire, Philip Kelly, Damien Connaghan, and Noel E. O'Connor. Tennissense: A platform for extracting semantic information from multi-camera tennis data. In *2009 16th International Conference on Digital Signal Processing*. IEEE, 2009. doi: 10.1109/icdsp.2009.5201152. URL https://dx.doi.org/10.1109/icdsp.2009.5201152.

[14] S. Messelodi, C. M. Modena, V. Ropele, S. Marcon, and M. Sgrò. *A Low-Cost Computer Vision System for Real-Time Tennis Analysis*, pages 106–116. Springer International Publishing, 2019. ISBN 0302-9743. doi: 10.1007/978-3-030-30642-7_10. URL https://dx.doi.org/10.1007/978-3-030-30642-7_10.

[15] Silvia Vinyes Mora. *Computer Vision and Machine Learning for In-Play Tennis Analysis: Framework, Algorithms and Implementation*. Thesis, Department Of Computing, 2017. URL https://www.doc.ic.ac.uk/~wjk/publications/vinyes-2018.pdf.

[16] V. Reno, N. Mosca, M. Nitti, T. D'Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. A technology platform for automatic high-level tennis game analysis. *Computer Vision and Image Understanding*, 159:164–175, 2017. ISSN 1077-3142. doi: 10.1016/j.cviu.2017.01.002. URL <GotoISI>://WOS:000404422100013.

[17] n/a EYES-ON. Eyeson, Oct 2021. URL https://www.eyeson.tennis/.

[18] Sofia Gourgari, Georgios Goudelis, Konstantinos Karpouzis, and Stefanos Kollias. Thetis: Three dimensional tennis shots a human action dataset. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2013.

[19] Kenneth Dawson-Howe. *Images*, page 18–19. John Wiley and Sons, 2014.

[20] Kenneth Dawson-Howe. *Connectivity*, page 67–70. John Wiley and Sons, 2014.

[21] Opencv: Finding contours in your image, n.d. URL https://docs.opencv.org/3.4/df/d0d/tutorial_find_contours.html.

[22] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.

[23] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692.

[24] Kenneth Dawson-Howe. *Dense Optical Flow*, page 182–185. John Wiley and Sons, 2014.

[25] n.d. URL https://www.wi-fi.org/discover-wi-fi/wi-fi-direct.

[26] G. Sudhir, J. C. M. Lee, and A. K. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*. IEEE Comput. Soc, 1997. doi: 10.1109/caivd.1998.646036. URL https://dx.doi.org/10.1109/caivd.1998.646036.

[27] Tien Ming-Chun, Wang Yi-Tang, Chou Chen-Wei, Hsieh Kuei-Yi, Chu Wei-Ta, and Wu Ja-Ling. Event detection in tennis matches based on video data mining. In *2008 IEEE International Conference on Multimedia and Expo*. IEEE, 2008. doi: 10.1109/icme.2008.4607725. URL https://dx.doi.org/10.1109/icme.2008.4607725.

[28] Dirk Farin, Susanne Krabbe, Peter de With, and Wolfgang Effelsberg. *Robust camera calibration for sport videos using court models*, volume 5307 of *Electronic Imaging 2004*. SPIE, 2003. URL https://doi.org/10.1117/12.526813.

[29] Silvia Vinyes Mora. *Computer Vision and Machine Learning for In-Play Tennis Analysis: Framework, Algorithms and Implementation*. Thesis, Department of Computing, 2017. URL https://www.doc.ic.ac.uk/~wjk/publications/vinyes-2018.pdf.

[30] Fei Yan, W Christmas, and Josef Kittler. A tennis ball tracking algorithm for automatic annotation of tennis match. In *British machine vision conference*, volume 2, pages 619–628. University of Surrey, 2005.

[31] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. IEEE Comput. Soc, 2000. doi: 10.1109/icpr.2000.902885. URL https://dx.doi.org/10.1109/icpr.2000.902885.

[32] Conor Gilmartin. Producing a tennis shot placement report by detecting bounce points in tennis shots. n.d., 2021.

[33] Dec 2019. URL https://momofilmfest.com/smartphone-iphone-filmmaking-what-is-variable-frame-

[34] Yaron Caspi, Denis Simakov, and Michal Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006. ISSN 0920-5691. doi: 10.1007/s11263-005-4842-z. URL https://dx.doi.org/10.1007/s11263-005-4842-z.

[35] A. Elhayek, C. Stoll, K. I. Kim, H. P. Seidel, and C. Theobalt. *Feature-Based Multi-video Synchronization with Subframe Accuracy*, pages 266–275. Springer Berlin Heidelberg, 2012. ISBN 0302-9743. doi: 10.1007/978-3-642-32717-9_27. URL https://dx.doi.org/10.1007/978-3-642-32717-9_27.

[36] Prarthana Shrstha, Mauro Barbieri, and Hans Weda. Synchronization of multi-camera video recordings based on audio. In *Proceedings of the 15th ACM International Conference on Multimedia*. ACM Press, 2007. doi: 10.1145/1291233.1291367. URL https://dx.doi.org/10.1145/1291233.1291367.

[37] Anna Llagostera Casanovas and Andrea Cavallaro. Audio-visual events for multi-camera synchronization. *Multimedia Tools and Applications*, 74(4):1317–1340, 2015. ISSN 1380-7501. doi: 10.1007/s11042-014-1872-y. URL https://dx.doi.org/10.1007/s11042-014-1872-y.

[38] Nasim Mansurov. What is lens distortion?, Aug 2013. URL https://photographylife.com/what-is-distortion.

[39] Camera calibration, n.d. URL https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.

[40] lovetennis. Jelena ostapenko unhappy with hawk-eye live - tennis news, Jan 2021. URL https://www.lovetennis.com/tennis-news/jelena-ostapenko-complains-about-hawk-eye-live-during-opening-round-win/.

[41] Feb 2021. URL https://tennishead.net/hawk-eye-is-not-at-all-accurate-gilles-simon-blasts-el

[42] Silvia Vinyes Mora. *Computer Vision and Machine Learning for In-Play Tennis Analysis: Framework, Algorithms and Implementation*. Thesis, Department of Computing, 2017. URL https://www.doc.ic.ac.uk/~wjk/publications/vinyes-2018.pdf.

Appendix A

Sample Application

A.1 Sample Interface

Figure A.1: A sample display of a fully implemented Mobile Tennis Analysis Application..

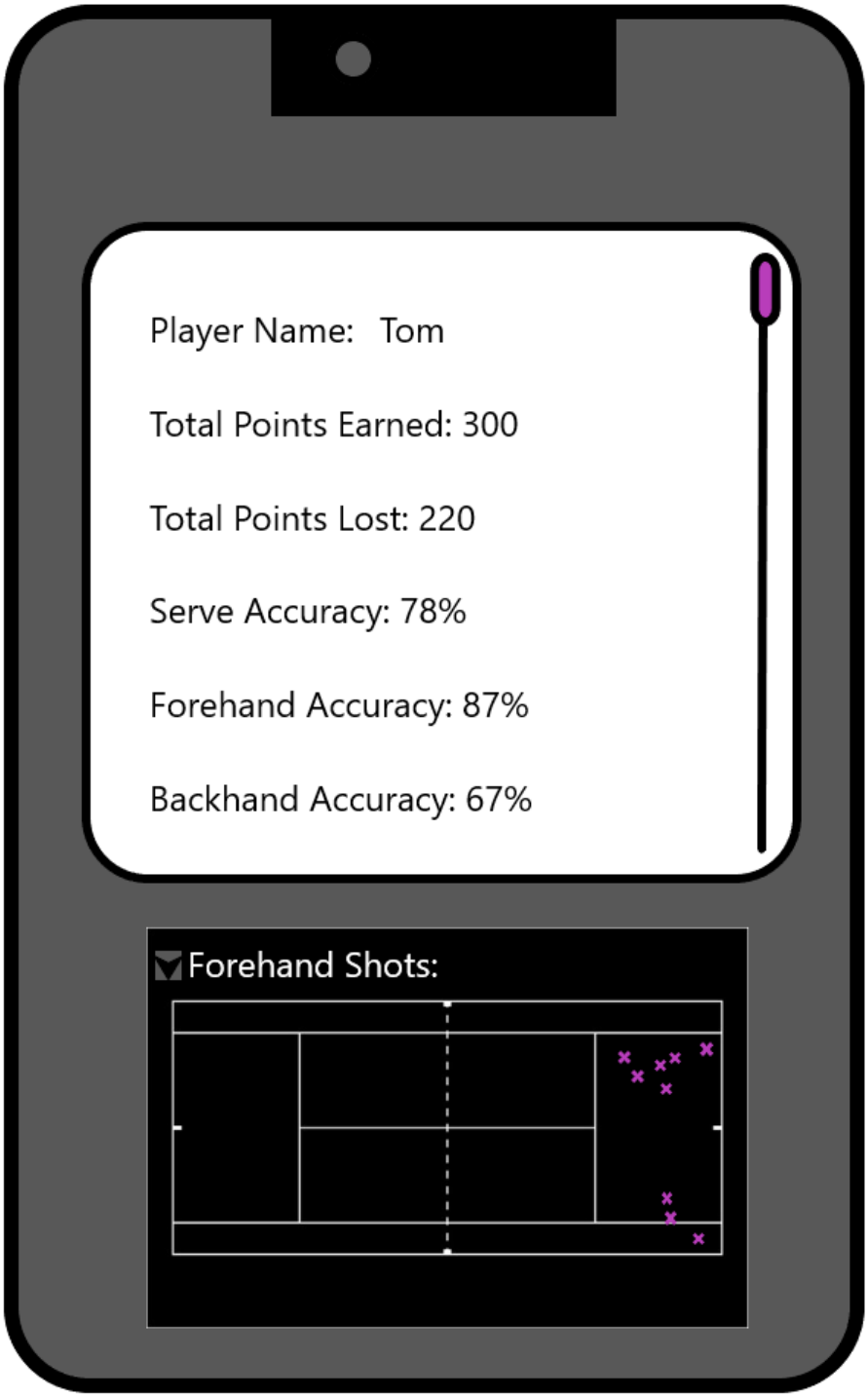


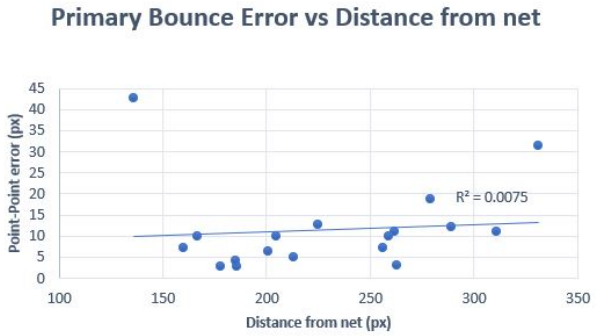
Figure A.2: A sample interface for that same application, where information performance-related information and a history of shots are displayed.

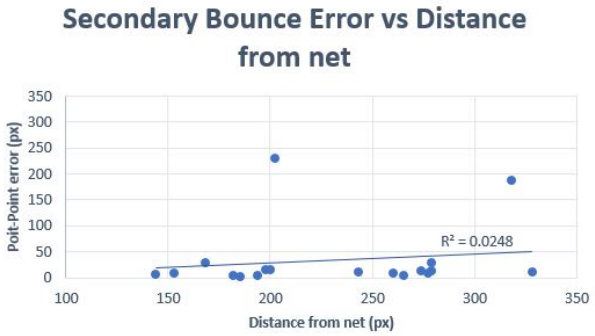
Appendix B

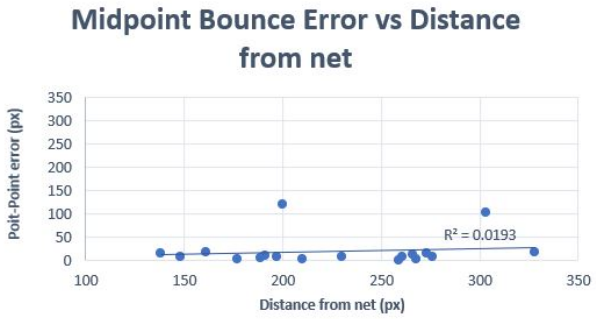
Bounce Detection

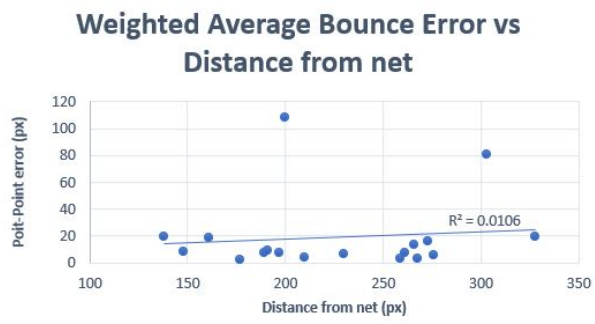
B.1 Error of Bounce Points vs Distance from the net

Bounce Point Error (px) vs Distance From Net (Overcast Conditions)









Bounce Point Error (px) vs Distance From Net (Sunny Conditions)