THE UNIVERSITY OF MELBOURNE

Capstone Project

Department of Mechanical Engineering

Final Report
2015

Project Title: Methods of Projectile Inference in a Tennis Computer Vision System

Date: 5 October 2015

Project Team Information

Identifier: **CP-DOE-152**

Student workers: Robert Chin

Academic supervisor: Denny Oetomo

Academic examiner: Simon Illingworth

Version: 1.1, 12 June 2016

Contents

1 Introduction **4**

2 Literature Review **6**
2.1 Camera Theory . . . 6
2.1.1 Perspective Projection . . . 6
2.1.2 Camera Calibration . . . 8
2.2 Particle Filters . . . 10
2.3 Dynamics . . . 10
2.4 Precedents . . . 11

3 Modelling **13**
3.1 Force . . . 13
3.2 Torque . . . 16

4 Methods **18**
4.1 Alternative Camera Calibration . . . 18
4.2 Trajectory Reconstruction . . . 20

5 Application **24**
5.1 Automated Camera Calibration . . . 24
5.2 Automated Ball Tracking . . . 26
5.2.1 Video Processing . . . 26
5.2.2 Particle Filter . . . 28
5.2.3 Data Association . . . 30
5.3 Testing and Results . . . 31
5.3.1 Stroke Analysis . . . 31
5.3.2 Serve Analysis . . . 32

6 Conclusion **36**

Bibliography **38**

Appendices **45**

A Table of Constants **45**

B Observability **46**

C Alternative Cost Function **47**

Executive Summary

Vision-based sensing systems can come at cheaper cost to conventional sensing methods. However, video can also be rich in information, making it difficult to extract what is desired from the data. Hence the development of algorithms that can interpret the content of video is greatly valued. In this project, we focus on estimating the 3D trajectory of a tennis ball through 2D tennis video captured from a single camera angle.

A model to describe ball behaviour in flight was derived. Using past research on the aerodynamics of tennis balls, a simplified differential equation for translational motion was obtained. A rotational differential equation was also obtained, but was eventually considered negligible enough to be ignored.

A method of camera calibration that only requires point correspondences lying on a single plane, in conjunction with the vanishing point, was proposed to calibrate an image containing a tennis court. An automated approach to detect lines and identify key locations in the court image was used in the testing of this method. This showed that the method did indeed as long as some assumptions were adhered to.

A procedure to reconstruct a nonlinear physics-based trajectory from 2D data was described. A particle filter was implemented on tennis video to track the ball. A state chart used to assign tracks to key events in tennis rallies were also explored. These were combined in an automated system to track and reconstruct the trajectories of tennis balls. The performance of the system was evaluated, however it falls short compared to the ideal. A list of sources of error was provided. It was also found that under ideal conditions the estimated kinematics of trajectories closely matches those as measured by radar guns at tennis stadiums.

From this project, it was shown that it is possible to estimate 3D information from a 2D perspective, however we also make the recommendation that multiple camera angles be used for sensing if accuracy is a performance requirement.

Chapter 1

Introduction

The human vision system is exceedingly sophisticated, more so than many synthetic systems. Signals interpreted by the eyes lead to higher level decision making in the brain. Beliefs are formed not just from what we observe, but also from our past experiences and our understanding of the physical world. It is from understanding that more complex tasks are possible, such as the throwing and catching of a projectile. It is learned from birth that gravity affects midair objects, thus we instinctively infer that such objects will eventually fall to the ground. Artificial systems should also be capable of using a priori knowledge to better predict the future or inform the present. In this project, we focus on estimating the trajectory of a tennis ball projectile in a vision system.

The advent of the image sensor has meant that vision-based systems can be implemented at lower cost. However, the tradeoff is in some loss of precision and accuracy. Data must also undergo further processing in software to perform the actual sensing. Therefore it is one of the goals of this project to explore the various methods which are critical to understand for vision-based sensors to rival their alternatives. This has applications in robotic simultaneous localisation and mapping (SLAM), projectile tracking, control system sensing, and other mechatronic applications.

Currently, the RoboCup contest encourages research in robotics, control systems, artificial intelligence and computer vision in a competitive setting using robotic soccer players. This leaves for a discipline that can be similarly applied in futuristic tennis robots. Furthermore, tennis computer visions can be of aid to tennis players and coaches as an improvement tool, and also to gain strategic insight into opponents.

We define 3 primary aims for this project.

1. Investigate and examine the dynamics behind a tennis ball under drag and spin conditions. Consider past literature and experimental evidence to formulate a physical model for ball behaviour during flight.
2. Develop and present a set of algorithms and methods that allow for a set of 2D points

corresponding the image trajectory to be transformed into a 3D trajectory.

3. Apply the methods developed in Aim 2 into practice on tennis video. Test whether the methods work, and validate and comment on the results.

The organisation of this report is structured as follows. A literary review of the current state of the art is to be provided in Chapter 2. Aim 1 is to be addressed in Chapter 3, Aim 2 is to be addressed in Chapter 4 and Aim 3 is to be addressed in Chapter 5. Finally, a conclusive view on the project is given in Chapter 6.

Chapter 2

Literature Review

In this chapter, background information on topics later referred to in the report is presented. A concise review of past studies in tennis ball aerodynamics is also covered, along with a summary of all precedents to this project.

2.1 Camera Theory

2.1.1 Perspective Projection

We firstly define the geometry of the tennis court as depicted in Figure 2.1. The court is symmetric about both the X-Y and the Y-Z planes. 3 different coordinate systems are to be considered.

Definition 1 *The world coordinate system is an inertial reference frame, with origin at point $\emptyset$ and axes $\{X, Y, Z\}$.*

Definition 2 *The camera coordinate system is a non-inertial reference frame, with origin at point C and axes $\{X', Y', Z'\}$.*

Definition 3 *The image coordinate system is planar, and is the projection of all world points that lie in the viewing frustrum of the camera. Its axes are given by $\{x, y\}$. In a digital image, x and y are both discrete variables from the set of nonnegative integers.*

We make the assumption that the camera operates under a pinhole camera model, which simplifies the projection to being linear. This is a valid assumption in the majority of cases, unless a fisheye camera is being used. Fisheye cameras introduce significant radial distortion in the image, and the projection becomes no longer linear.

Assumption 1 *There is negligible radial distortion in the camera such that a pinhole camera model is allowed to be used to describe the projection of world points on to image points.*

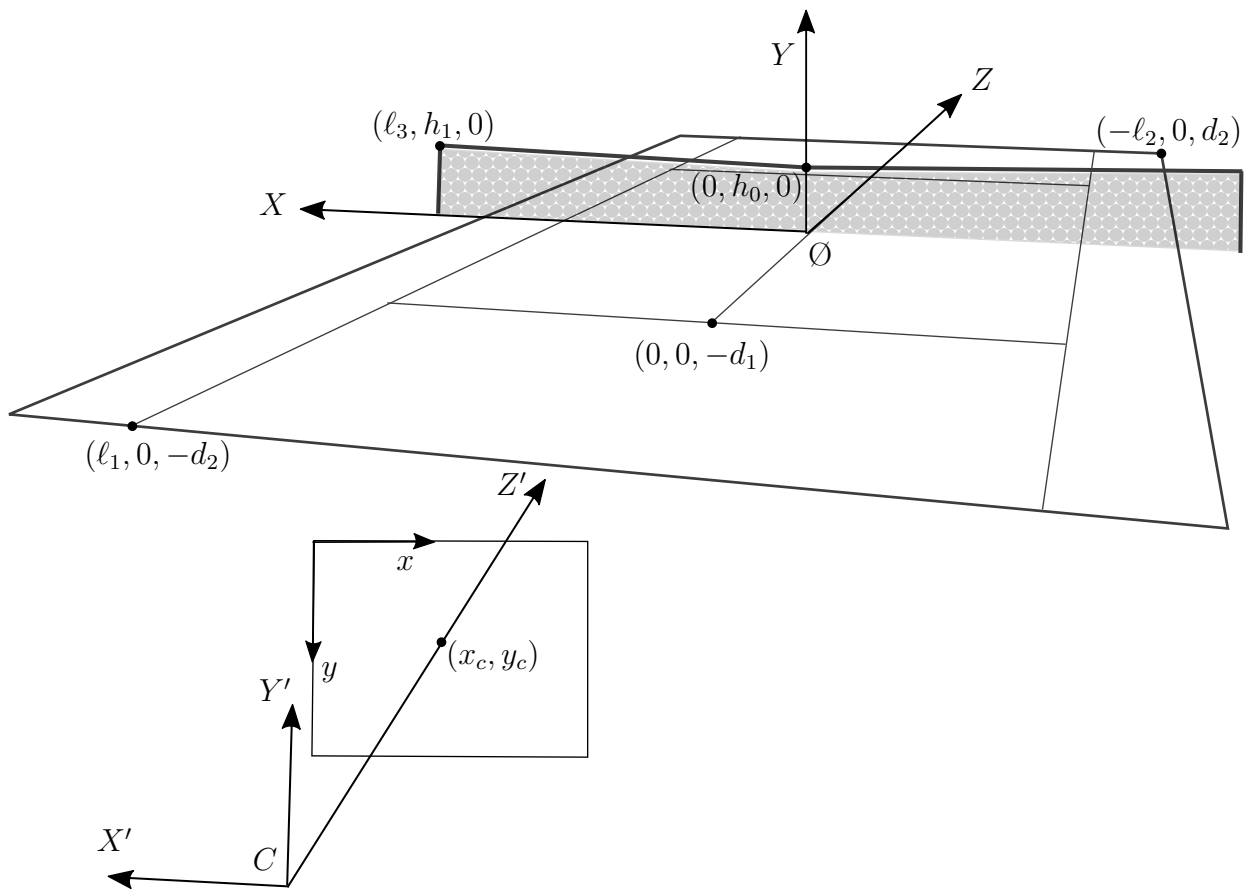


Figure 2.1: A definition of coordinate conventions.

Consider the linear transformation $M : \mathbb{R}^4 \to \mathbb{R}^3$,

$$\mathbf{M} := \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \tag{2.1.1}$$

This transformation maps a world coordinate position $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ to a vector in projection space $\begin{bmatrix} \kappa x & \kappa y & \kappa \end{bmatrix}^T$, ie.

$$\kappa \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1.2}$$

We see that the transformation between world coordinates and image coordinates is given by a nonlinear function $H : \mathbb{R}^3 \to \mathbb{R}^2$ where

$$H_1(X, Y, Z) := x = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \tag{2.1.3}$$

$$H_2(X,Y,Z) := y = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \tag{2.1.4}$$

Proposition 1 *It can be seen from* (2.1.3) *and* (2.1.4) *that multiplying* $\mathbf{M}$ *by any scalar will not change* x *or* y*. Hence there exist infinitely many possibilities of* $\mathbf{M}$ *that can achieve the same transformation between world coordinates and image coordinates.*

2.1.2 Camera Calibration

Given $n \geq 6$ known correspondences between image coordinates and world coordinates, we can use (2.1.3) and (2.1.4) to form a homogeneous system of equations of the form $\mathbf{Ax} = \mathbf{0}$.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1x_1 & -Y_1x_1 & -Z_1x_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1y_1 & -Y_1y_1 & -Z_1y_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_nx_n & -Y_nx_n & -Z_nx_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_ny_n & -Y_ny_n & -Z_ny_n & -y_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \mathbf{0} \tag{2.1.5}$$

For $n = 6$, solution for the m-parameters is given by the null space of $\mathbf{A}$. In the case of $n > 6$, the system is overconstrained and a least squares solution is given by the eigenvector corresponding to the smallest eigenvalue for $\mathbf{A}^T\mathbf{A}$ [19].

$\mathbf{M}$ can be further decomposed into a matrix $\mathcal{I} \in \mathbb{R}^{3\times3}$ of intrinsic parameters and matrix $\mathcal{E} \in \mathbb{R}^{3\times4}$ of extrinsic parameters.

$$\mathbf{M} = \frac{1}{\eta}\mathcal{I}\mathcal{E} \tag{2.1.6}$$

where η is a scalar multiple that arises from Proposition 1. The intrinsic matrix $\mathcal{I}$ consists of parameters defined by the camera itself.

$$\mathcal{I} := \begin{bmatrix} f_x & -f_x \cot\phi & x_c \\ 0 & f_y \csc\phi & y_c \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1.7}$$

where f_x, f_y are focal lengths, x_c, y_c determine the coordinates of the image centre, and ϕ is some angle in the neighbourhood of $\pi/2$ which determines the extent of image skewness (due to camera manufacturing error). For simplicity we assume $\phi = \pi/2$.

Assumption 2 *Error leading to image skewness is negligible, meaning* $\phi \approx \pi/2$.

$$\mathcal{I} = \begin{bmatrix} f_x & 0 & x_c \\ 0 & f_y & y_c \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1.8}$$

The extrinsic matrix $\mathcal{E}$ captures the geometric relationship between the world coordinate origin and the camera coordinate origin.

$$\mathcal{E} = \left[\begin{array}{c|c} {}^{C}_{\emptyset}R & {}^{C}\mathbf{r}_{C\emptyset} \end{array} \right] \tag{2.1.9}$$

where ${}^{C}_{\emptyset}R$ is the rotation matrix from the camera coordinate system to the world coordinate system, and ${}^{C}\mathbf{r}_{C\emptyset}$ is the displacement from the camera origin to the world origin expressed in camera coordinates. We define

$$\mathcal{E} := \left[\begin{array}{c|c} \boldsymbol{r_1}^T & \\ \boldsymbol{r_2}^T & {}^{C}\mathbf{r}_{C\emptyset} \\ \boldsymbol{r_3}^T & \end{array} \right] := \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{14} \\ r_{21} & r_{22} & r_{23} & t_{24} \\ r_{31} & r_{32} & r_{33} & t_{34} \end{bmatrix} \tag{2.1.10}$$

Given a known $\mathbf{M} := \left[\begin{array}{c|c} \mathbf{m_1}^T & \\ \mathbf{m_2}^T & \mathbf{m_4} \\ \mathbf{m_3}^T & \end{array} \right]$ and assuming the camera lies in the negative Z region, we can solve for all extrinsic and intrinsic parameters [19].

$$\eta = \frac{\operatorname{sign}(m_{34})}{\|\mathbf{m_3}\|} \tag{2.1.11}$$

$$x_c = \eta^2 (\mathbf{m_1} \cdot \mathbf{m_3}) \tag{2.1.12}$$

$$y_c = \eta^2 (\mathbf{m_2} \cdot \mathbf{m_3}) \tag{2.1.13}$$

$$f_x = \eta^2 \|\mathbf{m_1} \times \mathbf{m_3}\| \tag{2.1.14}$$

$$f_y = \eta^2 \|\mathbf{m_2} \times \mathbf{m_3}\| \tag{2.1.15}$$

$$\boldsymbol{r_1} = \frac{\eta^2}{f_y} (\mathbf{m_2} \times \mathbf{m_3}) \tag{2.1.16}$$

$$\boldsymbol{r_2} = -\frac{\eta^2}{f_x} (\mathbf{m_1} \times \mathbf{m_3}) \tag{2.1.17}$$

$$\boldsymbol{r_3} = \eta \mathbf{m_3} \tag{2.1.18}$$

$${}^{C}\mathbf{r}_{C\emptyset} = \rho \mathcal{E}^{-1} \mathbf{m_4} \tag{2.1.19}$$

Therefore the position of the camera pinhole with respect to the world coordinate system can be localised using

$${}^{\emptyset}\mathbf{r}_{\emptyset C} = -{}^{\emptyset}_{C}R\, {}^{C}\mathbf{r}_{C\emptyset} \tag{2.1.20}$$

2.2 Particle Filters

Particle filters are a Monte-Carlo approach to state estimation algorithms. Their main advantage over other algorithms such as Kalman filters are that they can be applied to problems with nonlinear state space and non-Gaussian probability densities. Their random sampling and genetic approach in the update step means that they are effective even when little is known about the system. They have powerful applications in video tracking [31] and robotic SLAM.

Consider the prediction of the ith particle at time k out of a system of N particles, each of which has a state vector denoted by $\mathcal{X}_k^i$. Every update of the algorithm begins with a prediction step

$$\mathcal{X}_{k+1}^i = \mathcal{F}\left(\mathcal{X}_k^i\right) + \mathcal{W}_k^i \tag{2.2.1}$$

where $\mathcal{F}$ is an arbitrary function of the previous state and $\mathcal{W}_k^i$ is a random noise term. The probability $P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right)$ denotes the likelihood for an observation of the system $\mathcal{Y}_k$, given the particle at $\mathcal{X}_k^i$. $P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right)$ may be an arbitrary probability distribution $\mathcal{P}$ in both the state and the output.

$$P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right) = \mathcal{P}\left(\mathcal{Y}_k, \mathcal{X}_k^i\right) \tag{2.2.2}$$

In the update step, all particles are resampled with replacement according to the likelihood ratio

$$LR_i = \frac{P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right)}{\sum_i P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right)}, \forall i \tag{2.2.3}$$

This shows that the probability of being resampled to a particle is proportional to the likelihood of the observation given that particle. One cycle of the algorithm is summarised by

1. Update each particle using (2.2.1).
2. For each particle calculate the likelihood using (2.2.2).
3. Resample with replacement all particles amongst each other using the likelihood ratio (2.2.3).

2.3 Dynamics

In this section we review the past studies into the aerodynamics of tennis balls. Some wind tunnel experiments have looked at the effect of the drag force on non-spinning balls [3, 10, 20, 29]. The drag coefficient C_D was measured over a varying range of Reynold's numbers from 80 000 $<$ Re $<$ 300 000, corresponding to velocities between 19 ms^{-1} and 70 ms^{-1} respectively. This range is chosen as it represents the typical speeds of balls during tennis matches. The primary findings are that as the Reynold's number was increased, the measured drag coefficient decreased, until a certain region was reached at a Reynold's number of around 167 000, where the drag coefficient would be largely independent from Reynold's

number, at a value of between 0.6 to 0.7.

There have also been experiments looking into the effect of spin on tennis balls [1,2,4,20,40]. The Magnus effect is a phenomenon caused by a pressure differential at different parts of the ball surface while it is spinning and translating at the same time. This pressure differential will result in another force on the ball, characterised by the lift coefficient C_L. Spin magnitudes up to 4000 rpm were tested, representing the range of spin at which professional players are able to generate. While the lift coefficient was found to increase as the magnitude of spin increased, it was discovered that increasing the magnitude of spin would also increase the drag coefficient.

Other studies [14,28] have attempted to summarise the literature in tennis ball aerodynamics, while [40] developed empirical relations for the drag and lift coefficients, to be demonstrated in simulations by [16, 24].

2.4 Precedents

In this section we review the current literature on video tracking in tennis, and comment on the frontier of 3D trajectory reconstruction in sports. The majority of these studies are motivated by using tracking and reconstruction algorithms as a tool for the enhancement of sports broadcasting and coaching, however little emphasis is placed on their viability in robotic and mechatronic systems. [17, 27, 35, 50, 51] focus solely on tracking the tennis ball in video and a variety of video processing techniques are proposed. There have also been studies which examine methods to track the tennis players in videos [18, 34, 54], and some which perform a combination of ball and player tracking [26,36,42]. Substantial research has been made into the creation of an automated ball tracking system for annotation of tennis matches [45–49]. There are generally reports of success in ball tracking, however a common shortcoming is the robustness. In a video, it is easy to lose track of the ball if it is moving too quick, or confuse it with some other noisy clutter originated from the environment. Video is also very rich in data, and it takes considerable computing power to be able to process video online. This is another common problem. The amount of effort increases as the abundance of cameras which can capture in high definition increases. In all studies discussed so far, the ball was tracked in only a 2D sense. None have attempted to use the tracked ball data to reconstruct the 3D trajectory.

There are a number of other papers which have looked at the 3D reconstruction of trajectories based on video data, but a majority of these have been confined to other sports. In [23,30,38], the trajectories of soccer balls were reconstructed. In [11,12], video tracking of a volleyball followed by trajectory classification was conducted. A similar technique however using basketballs was performed in [13]. A method to reconstruct the trajectory based on a single long exposure image was presented in [8]. In the above, a common procedure to achieve reconstruction was to fit parabolic models of ball flight with the data from the video,

with the homography of the video known.

In tennis, classical tracking of the ball occurred using radar [5] prior to the advent of cheaper and higher resolution cameras. More modern ball tracking systems took advantage of video and were developed in both [25] and [37]. For [25], a simpler parabolic trajectory was fitted to the tracked ball, however in [37], a more complex model featuring drag and spin was used to reconstruct the trajectory. In both cases, multiple cameras situated around the court were used to triangulate the position of the ball.

Among currently existing commercial applications of tennis ball tracking are Hawkeye [32], PlaySight [39] and Mojjo [6]. Hawkeye is primarily used as an officiating system in professional tennis matches, and operates based on multiple high speed cameras. It is considered accurate enough to overrule line judges when deciding whether a ball lands in or out of the court. In practice, it fits polynomial 3D trajectories to the ball motion. PlaySight is designed for installation at tennis clubs and training centres, and its main purpose is to analyse tennis matches, and provide metrics and metadata as feedback to players. Like Hawkeye, multiple cameras are required to be installed on the court for 3D tracking. Lastly, Mojjo uses a single wide lens fisheye camera with a birds-eye angle to track the ball. However it is as of yet unable to reconstruct the 3D trajectory from video sequences.

We now compare previous studies from this project, and discuss how this project differentiates from those above. While 3D tracking is more easily done when there are multiple cameras from different angles, this project focuses on reconstructing the trajectory using only a single camera angle. Also, while physics-based approaches have been used to reconstruct the trajectory, a simple parabolic path has typically been used as the physical model. While this may be acceptable in other sports, it becomes too much of a simplification in tennis. In tennis, spin has a noticeable effect on the curvature of the ball, and players will often incorporate spin into their playing strategy. The textured felt on the surface of tennis balls along with the potential of high speeds means that drag is a significant force acting on the ball. Hence the effects of spin and drag should not be ignored. We incorporate the effects of spin and drag on our model with the intention that it will be able to more accurately foretell the future path of the ball. This is in the spirit of attempting to provide tools that are useful for robotic and mechatronic systems.

Chapter 3

Modelling

In this chapter the behaviour of tennis ball flight is modelled, and reasoning is provided the the choice of model components. The differential equations of motion are stated, and an outline of all assumptions of the model are listed.

3.1 Force

We begin by developing the force components acting on the ball in free flight. The three main forces considered to be acting on the ball are

1. Drag force
2. Magnus force
3. Gravity

The drag force acting on an object is proportional to the squared speed of the object, and acts opposite to the direction of motion. In vector form, it is given by

$$\mathbf{F}_D = -\frac{1}{2} C_D \rho A \|\mathbf{v}\|^2 \frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{3.1.1}$$

where C_D is the drag coefficient, ρ is the density of air, A is the cross-sectional area of the ball and $\mathbf{v}$ is the velocity of the ball relative to the wind.

The Magnus force is also proportional to the speed squared, however it acts in a direction that is perpendicular to both the velocity and the spin axis [24, 33]. The expression for the force is

$$\mathbf{F}_M = \frac{1}{2} C_L \rho A \|\mathbf{v}\|^2 \left(\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \times \frac{\mathbf{v}}{\|\mathbf{v}\|} \right) \tag{3.1.2}$$

where C_L is the lift coefficient, and $\boldsymbol{\omega}$ is the angular velocity vector.

Finally, force due to gravity is assigned to act in the negative Y direction.

$$\mathbf{F}_G = -mg \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.1.3}$$

where g is the constant of acceleration due to gravity.

The drag and lift coefficients C_D, C_L, within a certain Reynold's number range, have both been empirically found to depend on a normalised variable of the spin magnitude, called the spin coefficient $\lambda := \|R\boldsymbol{\omega}\|/\|\mathbf{v}\|$, where R is the radius of the ball [40]. Figures 3.1 and 3.2 show the relationships between these variables across different studies, however we choose to model ball flight using the equations from [16], which are themselves adapted from equations fitted to experimental data in [40].

$$C_D = 0.55 + \left(22.5 + 4.2\lambda^{-2.5}\right)^{-0.4} \tag{3.1.4}$$

$$C_L = \left(2 + \lambda^{-1}\right)^{-1} \tag{3.1.5}$$

We now take stock of the discrepancies between the experimental data and the model we have chosen. Figure 3.1 shows that the correlation (3.1.4) fits somewhat with the data, although in Figure 3.2 there is a mismatch between the correlation (3.1.5) and the results from Alam et. al. [2]. A number of reasons are suggested to explain these discrepancies. Firstly, there is likely to be some variation in environmental conditions and testing methodology within each study, degrading the consistency of data. Secondly and foremost, there are due to be differences between the make of tennis balls used in each test. The results originating from [40] used pressureless Tretorn branded balls, while those from [2] are from pressurised Wilson Rally 2 balls. A variety of balls based on the type of ball felt were used in [20]. In all cases, data were from balls in their 'new' state. Differences in ball composition, as well as the desire of the sport's governing bodies to reduce the speed of the game [22], contribute towards the variation in ball characteristics across studies and over time. We decide to abstract away from these variations, as the impact of factors such as ball felt texture are currently not well studied.

It is pertinent to consider all simplifying assumptions made so far, as although these reduce complexity, they also provide an understanding to the limitations of our model. The differences between new and worn balls were studied in [20, 21] and it was determined that C_D and C_L in general decreased as the balls became more worn, because felt was removed from the ball as it was played with. However prior to that stage, ball felt becomes fuzzier after several impacts, leading to increased C_D compared to when they are new. There also exists a complicated dynamic where fuzzy felt can 'lay down' after sufficient ball speed, leading to decreasing C_D within a certain Reynold's number range. Thus there exists a transient stage in the ball's flow regime, however this can be considered negligible because it affects a ball's initial velocity after impact by less than 1% [14]. Because tennis balls are also constructed

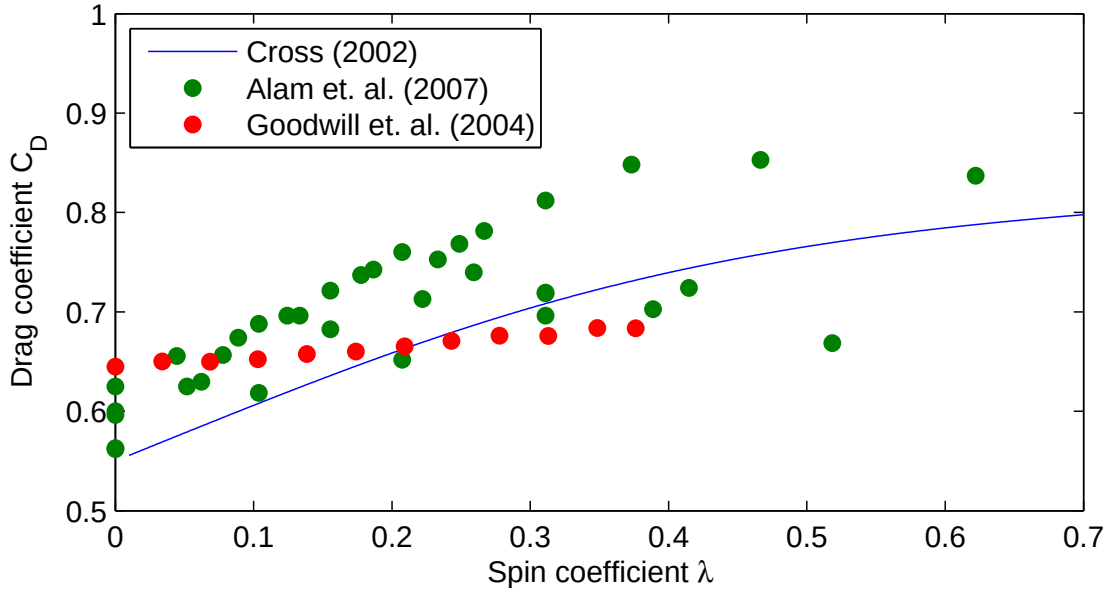


Figure 3.1: The differences for C_D in experimental data found by Alam et. al. [2], Goodwill et. al. [20] and the correlation by Cross [16].

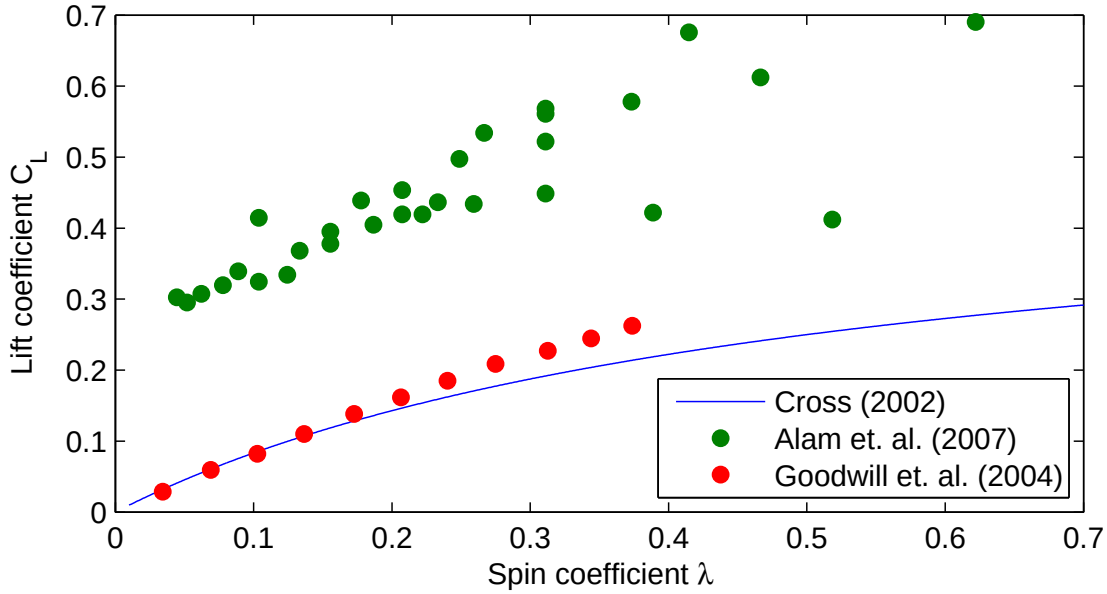


Figure 3.2: The differences for C_L in experimental data found by Alam et. al. [2], Goodwill et. al. [20] and the correlation by Cross [16].

with an asymmetric seam, then the seam orientation relative to the fluid flow can also have an effect on C_D [3]. All these factors are to be neglected. We summarise all assumptions

made on ball translational motion as follows.

Assumption 3 *Ball motion is deterministic, and the effects of all other minor forces such as the Coriolis force are ignored. The effect of wind is also ignored. During motion, only a single flow regime is considered, with coefficients governed by* (3.1.4) *and* (3.1.5)*. These equations are regardless of the the brand of ball, and also of the state of ball condition.*

We arrive at the differential equation for translational motion.

$$m\ddot{\mathbf{r}}_B = \mathbf{F}_D + \mathbf{F}_M + \mathbf{F}_G \tag{3.1.6}$$

where $\mathbf{r}_B$ is the displacement of the ball and m is the mass of the ball.

3.2 Torque

We now develop the rotational equation of motion for the ball. As the ball spins, we model the viscous torque that causes the ball to spin slower down over time. There is a lack of abundance in studies observing the effect of viscous torque on rotating tennis balls. However, viscous torque on table tennis balls was simulated in [43], and the form of the torque component was given by

$$\mathbf{T} = -\frac{1}{2}C_T\rho R^5\|\boldsymbol{\omega}\|^2\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{3.2.1}$$

where C_T is the viscous torque coefficient. Hence the differential equation for rotational motion is

$$\dot{\mathbf{h}} = \mathbf{T} \tag{3.2.2}$$

where $\mathbf{h} := \mathbf{I}\boldsymbol{\omega}$ is the angular momentum of the ball and $\mathbf{I} := \begin{bmatrix} \alpha mR^2 & 0 & 0 \\ 0 & \alpha mR^2 & 0 \\ 0 & 0 & \alpha mR^2 \end{bmatrix}$ is the form of an inertia tensor for a sphere. An estimate for α is given in [7]. The viscous torque coefficient depends on the rotational Reynolds number

$$\mathrm{Re}_\omega = \frac{\rho R^2\|\boldsymbol{\omega}\|}{\mu} \tag{3.2.3}$$

where μ is the dynamic viscocity. However, in [43], empirical relations were specified only for low rotational Reynolds numbers, up to 100. At this number, the viscous torque coefficient simulated on table tennis balls was approximately 1. However, rotational Reynolds numbers of 21 000 are possible in tennis, if considering a heavily spun ball at 3000 rpm. Furthermore, torque coefficient relations based on rotating smooth spheres are not likely to be as applicable to tennis balls, owing to the textured surface of the ball felt.

Using the differential equations for both translational and rotational motion, the trajectory of the ball can be simulated. The Runge-Kutta algorithm implemented in the `ode45` function from MATLAB is the choice of solver for the differential equations. From Figure 3.3 we compare that even with an arbitrary value of $C_T = 2$ against $C_T = 0$, the trajectories are very similar. This is because the radius R is typically in the vicinity of 0.033 m, and so the quintic term in (3.2.1) makes the effect of viscous torque very small. Since a value of C_T for tennis balls is unable to be sourced from literature, it becomes reasonable to neglect the effect of viscous torque.

Assumption 4 *The effect of viscous torque acting on the ball is negligible.*

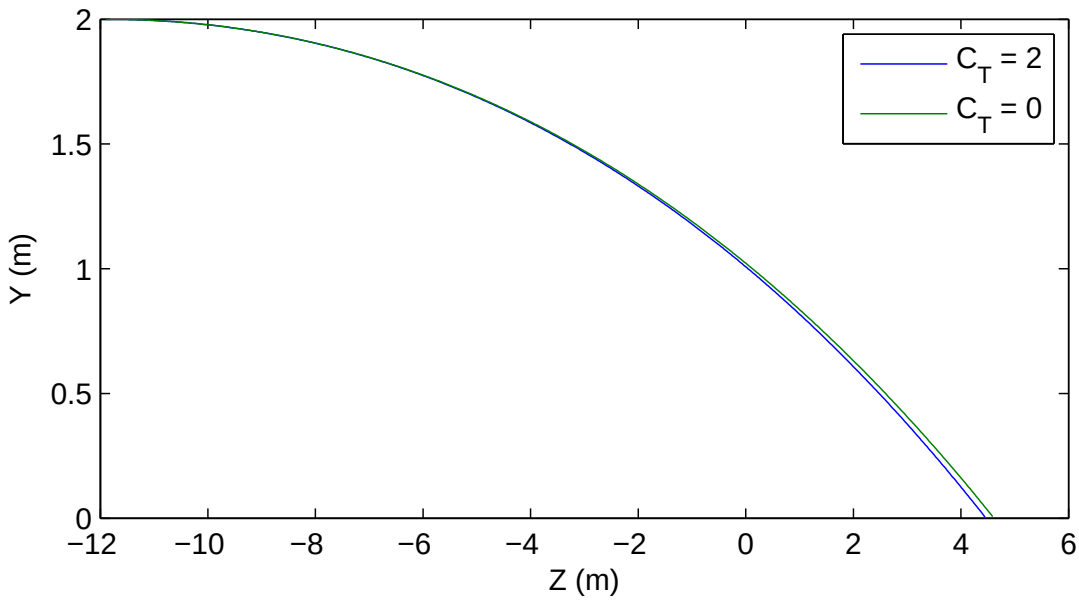


Figure 3.3: Trajectory simulation from an initial condition of 40 ms^{-1} and 300 $rad{\cdot}s^{-1}$.

Chapter 4

Methods

In this chapter, we develop algorithms used in the reconstruction of a 3D trajectory, without consideration of how image-world correspondences or 2D ball trajectory data is obtained. A customised procedure to calibrate images of tennis courts is proposed. A method for the fundamental transformation of 2D information to 3D information is then suggested.

4.1 Alternative Camera Calibration

A least squares approach to camera calibration is presented in Chapter 2, and also found in [41, 52, 53]. These techniques are suited for when there are known correspondences for world coordinate points lying on differing planes. However, for images containing tennis courts, it may only be possible to perform automated identification on points that are on the court surface. The reasoning for this is that the net cord is not perfectly straight by standards, so relying on straight line detection to locate the net may not be feasible.

If $Y = 0$ in all our correspondences, this leads to the problem that m_{12}, m_{22} and m_{32} will no longer be unique. Therefore we would like to seek an approach to calibrate the camera if only correspondences to court surface points are known. Firstly note that lines in world coordinates are projected to other lines in image coordinates. Hence a line painted on a tennis court will be projected to another line on an image. We would like to be able to use the slope of this line as it appears in the image to infer additional information about $\mathbf{M}$. The vanishing points are features of projected images that can be used in camera calibration [9]. However unlike [9] which requires multiple cameras, we develop an approach to calibrate using the vanishing point and one one camera angle.

Definition 4 *The vanishing points of an image are defined as the points in image coordinates where all projected parallel lines in world coordinates meet at infinity.*

To illustrate, consider all sidelines painted on the court, each with $Y = 0$ and constant value for $X = d_i$. Then the vanishing point (x_{vp}, y_{vp}) for all lines on the plane $Y = 0$ and parallel

to the sidelines are found as $Z \to \infty$ in (2.1.3) and (2.1.4).

$$x_{vp} = \lim_{Z\to\infty} \frac{m_{11}d_i + m_{13}Z + m_{14}}{m_{31}d_i + m_{33}Z + m_{34}} = \frac{m_{13}}{m_{33}} \tag{4.1.1}$$

$$y_{vp} = \lim_{Z\to\infty} \frac{m_{21}d_i + m_{23}Z + m_{24}}{m_{31}d_i + m_{33}Z + m_{34}} = \frac{m_{23}}{m_{33}} \tag{4.1.2}$$

It can also be shown that the image slope of any on the plane $Y = 0$ and line parallel to the sidelines is dependent only on the constant d_i and the m-parameters.

$$\frac{dy}{dx} = \frac{\partial y}{\partial Z}\frac{\partial Z}{\partial x} = \frac{d_i\left(m_{21}m_{33} - m_{23}m_{31}\right) - m_{23}m_{34} + m_{24}m_{33}}{d_i\left(m_{11}m_{33} - m_{13}m_{31}\right) - m_{13}m_{34} + m_{14}m_{33}} \tag{4.1.3}$$

Dividing the numerator and denominator of (4.1.3) by m_{33} gives

$$\frac{dy}{dx} = \frac{d_i\left(m_{21} - y_{vp}m_{31}\right) - y_{vp}m_{34} + m_{24}}{d_i\left(m_{11} - x_{vp}m_{31}\right) - x_{vp}m_{34} + m_{14}} \tag{4.1.4}$$

Suppose a point (x, y) is known to correspond with a world point on the court sideline. Then

$$\frac{dy}{dx} = \frac{x - x_{vp}}{y - y_{vp}} \tag{4.1.5}$$

We choose to equate the numerators and denominators of (4.1.4) to (4.1.5). Note that this is possible due to the flexibility offered from Proposition 1, and doing so specifies a unique solution for $\mathbf{M}$. This gives

$$\Delta x := x - x_{vp} = d_i\left(m_{11} - x_{vp}m_{31}\right) - x_{vp}m_{34} + m_{14} \tag{4.1.6}$$

$$\Delta y := y - y_{vp} = d_i\left(m_{21} - y_{vp}m_{31}\right) - y_{vp}m_{34} + m_{24} \tag{4.1.7}$$

Hence given $n \geq 4$ other image-world correspondences, a non-homogeneous system of equations can be found for 9 of the m-parameters, which are estimated through least squares.

$$\begin{bmatrix} d_i & 0 & 1 & 0 & 0 & 0 & -d_i x_{vp} & 0 & -x_{vp} \\ 0 & 0 & 0 & d_i & 0 & 1 & -d_i y_{vp} & 0 & -y_{vp} \\ X_1 & Z_1 & 1 & 0 & 0 & 0 & -X_1 x_1 & -Z_1 x_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Z_1 & 1 & -X_1 y_1 & -Z_1 y_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Z_n & 1 & 0 & 0 & 0 & -X_n x_n & -Z_n x_n & -x_n \\ 0 & 0 & 0 & X_n & Z_n & 1 & -X_n y_n & -Z_n y_n & -y_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{4.1.8}$$

To estimate the remaining 3 parameters, we first make an assumption that allows us to treat the ratio between horizontal and vertical distances translated along the court net, in both the world coordinate and image coordinate sense, to be roughly equal. These become valid assumptions when the pan and tilt on a pan-tilt camera are minimal.

Assumption 5 *The X-Y plane of the world coordinate system is approximately parallel to the X'-Y' plane of the camera coordinate system, and the camera focal lengths f_x and f_y are approximately equal.*

Firstly, the X-axis in world coordinates has the analytical projected image coordinate slope

$$\delta_1 := \frac{m_{21}m_{34} - m_{31}m_{24}}{m_{11}m_{34} - m_{31}m_{14}} \tag{4.1.9}$$

The perpendicular slope is

$$\delta_1' := -\frac{1}{\delta_1} = \frac{m_{31}m_{14} - m_{11}m_{34}}{m_{21}m_{34} - m_{31}m_{24}} \tag{4.1.10}$$

The Z-axis in world coordinates has the analytical projected image coordinate slope

$$\delta_2 := \frac{m_{22}m_{34} - m_{32}m_{24}}{m_{12}m_{34} - m_{32}m_{14}} \tag{4.1.11}$$

Let h_0 be the standard height in metres of the court net at the centre. Then define (x_0, y_0) to be the known projection of the world coordinate origin $(0, 0, 0)$, (x_L, y_L) to be the known projection of $(0.5h_0, 0, 0)$ and (x_R, y_R) to be the known projection of $(-0.5h_0, 0, 0)$. Then by Assumption 5, an estimate for the image projection of the top-centre of the net (x_p, y_p) is given by

$$\begin{bmatrix} \hat{x}_p \\ \hat{y}_p \end{bmatrix} = \begin{bmatrix} x_0 + y_R - y_L \\ y_0 - x_R + x_L \end{bmatrix} \tag{4.1.12}$$

We know that analytically,

$$x_p = \frac{h_0 m_{12} + m_{14}}{h_0 m_{32} + m_{34}} \tag{4.1.13}$$

$$y_p = \frac{h_0 m_{22} + m_{24}}{h_0 m_{32} + m_{34}} \tag{4.1.14}$$

Hence by equating (4.1.13), (4.1.14) with (4.1.12), and also equating (4.1.10) with (4.1.11) by Assumption 5, we formulate a system of equations for the unknown variables in terms of the known variables.

$$\begin{bmatrix} h_0 & 0 & -h_0 x_p \\ 0 & h_0 & -h_0 y_p \\ \delta_1' m_{34} & m_{34} & \delta_1' m_{14} - m_{24} \end{bmatrix} \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} = \begin{bmatrix} x_p m_{34} - m_{14} \\ y_p m_{34} - m_{24} \\ 0 \end{bmatrix} \tag{4.1.15}$$

This is sufficient to estimate the remaining parameters.

4.2 Trajectory Reconstruction

Suppose there is a known point (x, y) in image coordinates. We can rearrange (2.1.3) and (2.1.4) to find the equations of two planes.

$$(m_{11} - x m_{31})\, X + (m_{12} - x m_{32})\, Y + (m_{13} - x m_{33})\, Z = x m_{34} - m_{14} \tag{4.2.1}$$

$$(m_{21} - ym_{31})\,X + (m_{22} - ym_{32})\,Y + (m_{23} - ym_{33})\,Z = ym_{34} - m_{14} \tag{4.2.2}$$

Hence there are infinitely many solutions for (X, Y, Z) which lie on the intersection of these two planes. For a system of two planes $a_{11}X + a_{12}Y + a_{13}Z = b_1$ and $a_{21}X + a_{22}Y + a_{23}Z = b_2$, the normal directions are (a_{11}, a_{12}, a_{13}) and (a_{21}, a_{22}, a_{23}) respectively. A vector parallel to their line of intersection is found through the cross product of those vectors. Then normalising the Z-component gives a parametrisation in ϱ of the line containing the solution of (X, Y, Z).

$$\begin{bmatrix} X(\varrho) \\ Y(\varrho) \\ Z(\varrho) \end{bmatrix} = {}^{\emptyset}\mathbf{r}_{\emptyset C} + \frac{\varrho}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{12}a_{23} - a_{13}a_{22} \\ a_{11}a_{23} - a_{13}a_{21} \\ a_{11}a_{22} - a_{12}a_{21} \end{bmatrix}, \varrho \geq \Upsilon \tag{4.2.3}$$

where Υ is the smallest value such that the parametrisation begins on the boundary of the viewing frustrum. Hence we show that it is not directly possible to find world coordinates based on image coordinates. We also show is via an alternative method given in Appendix B.

Although knowing the image coordinates of the tennis ball at any time confers no unique solution about the state of the ball, it is when these image coordinates together form a trajectory that we are able to reconstruct the 3D flight path. Define a single flight path for the ball to be the time interval for $t \in [t_0, t_f]$, where t_0 is the time directly after the ball undergoes a collision, and t_f is the time directly preceding the next collision. A collision shall be considered most of the time to be, although not strictly limited to, an impact with the ball against a racquet, or the bounce of the ball against the court surface. Other likely occurring collisions include those with the net or court fencing. In the case where the ball bounces on the surface, it is then known that the ball must be in the vicinity of $Y = R$. From (2.1.3) and (2.1.4), this then leads to an explicit solution for the ball in world coordinates. In the case where the ball is struck by a racquet, we make the following assumptions.

Assumption 6 *The body of the player when striking the ball is constrained to a single plane, and this plane is parallel to the X-Y plane.*

Assumption 7 *The feet of the player as the ball is struck are connected to the ground.*

These assumptions are not always satisfied, because players will often make contact with the ball in front of them, and players can sometimes be airborne as they strike the ball. However they allow us to solve for the world coordinate position of the ball when struck by a racquet. If the image coordinate position of the feet are known, then by Assumption 7 we know the world coordinate location of the feet. From this, then by Assumption 6 we know the Z-value at which the entire player lies in, which must be the same Z-value of the ball being struck at that instant. The world coordinate position of the ball can be aptly determined.

So if at times t_0 and t_f it is known the world coordinates of the ball $\mathbf{r}_{B0}$ and $\mathbf{r}_{Bf}$ respectively, then reconstructing the trajectory between this time leads to a solution by optimisation

of a two point boundary value problem. Let $\mathbf{Q}$ denote a matrix which contains in each row the image coordinates of the tracked ball corresponding to the points in time from $\mathbf{t} = \begin{bmatrix} t_0 & t_1 & \dots & t_f \end{bmatrix}^T$.

$$\mathbf{Q} = \begin{bmatrix} x_{Q0} & y_{Q0} \\ x_{Q1} & y_{Q1} \\ \vdots & \vdots \\ x_{Qf} & y_{Qf} \end{bmatrix} \tag{4.2.4}$$

Let $\mathbf{W}\left(\mathbf{r}_{B0}, \mathbf{v}_0, \boldsymbol{\omega}_0\right)$ be the solution at times $\mathbf{t}$ to the equation of motion (3.1.6) with initial conditions $\mathbf{r}_{B0}$, $\mathbf{v}_0$ and $\boldsymbol{\omega}_0$, then transformed with the output function h from (2.1.3) and (2.1.4) to give the simulated image coordinates.

$$\mathbf{W}\left(\mathbf{r}_{B0}, \mathbf{v}_0, \boldsymbol{\omega}_0\right) = H\left(\begin{bmatrix} X_0 & Y_0 & Z_0 \\ X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_f & Y_f & Z_f \end{bmatrix}\right) = \begin{bmatrix} x_{W0} & y_{W0} \\ x_{W1} & y_{W1} \\ \vdots & \vdots \\ x_{Wf} & y_{Wf} \end{bmatrix} \tag{4.2.5}$$

We seek to minimise the trajectory error $\mathbf{Q} - \mathbf{W}$, and also constrain the simulated solution at t_f terminally with the known $\mathbf{r}_{Bf}$. Let the cost function J be defined as a quadratic in the trajectory error.

$$J = \text{trace}\left(\left(\mathbf{Q} - \mathbf{W}\right)^T \left(\mathbf{Q} - \mathbf{W}\right)\right) \tag{4.2.6}$$

Then the optimisation problem is declared as

$$\begin{aligned} &\min_{\mathbf{v}_0, \boldsymbol{\omega}_0} J\left(\mathbf{v}_0, \boldsymbol{\omega}_0\right) \\ &\text{subject to } \mathbf{r}_{Bf} - \begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix} = 0 \end{aligned} \tag{4.2.7}$$

An alternative specification of a cost function is also given in Appendix C. The resulting estimate of the initial conditions $\mathbf{v}_0$, $\boldsymbol{\omega}_0$, along with the original estimate of $\mathbf{r}_{B0}$ characterises the trajectory of each ball flight from t_0 to t_f.

The dynamic programming solver `fmincon` in MATLAB implements a constrained optimisation algorithm that can solve (4.2.7). Many optimisation algorithms also require an initial guess to seed the algorithm. It is desired that the guess be close to the true initial velocity, since will allow for faster convergence to solutions, and convergence to solutions closer to the true solution. To generate guess of initial velocity close to the true solution, the analytical solution without spin can be used. Consider the 1-dimensional differential equation with drag

$$\frac{d^2 s}{dt^2} = -\text{sign}\left(\frac{ds}{dt}\right) \frac{\zeta}{m} \left(\frac{ds}{dt}\right)^2 \tag{4.2.8}$$

with $\zeta = \frac{1}{2}\rho A C_D$. A solution for s is

$$s = \text{sign}\left(\frac{ds}{dt}\right)\frac{m}{\zeta}\ln\left(c_1 m + \zeta t\right) + c_2 \tag{4.2.9}$$

where c_1 and c_2 are constants of integration. Given initial condition s_1 at t_1 and s_2 at t_2, and taking sign $\left(\frac{ds}{dt}\right)$ to be sign $(s_1 - s_2)$, we have

$$c_1 = \frac{\zeta}{m}\left(t_2 - t_1\right)\left[\exp\left(\text{sign}\left(s_2 - s_1\right)\frac{\zeta}{m}\left(s_2 - s_1\right)\right) - 1\right]^{-1} \tag{4.2.10}$$

which is used to explicitly find the velocity at t_1.

$$\left.\frac{ds}{dt}\right|_{t=t_1} = \frac{\text{sign}\left(s_2 - s_1\right)m}{c_1 m + \zeta t_1} \tag{4.2.11}$$

This was performed to find the starting velocity components for the X and Z directions individually. In the case of the Y-direction with gravity, drag is neglected for simplicity because the component of velocity is assumed to be relatively small. Hence the guess of initial velocity in the Y-direction will be

$$\left.\frac{ds}{dt}\right|_{t=t_1} = \frac{s_2 - s_1 + \frac{g}{2}\left(t_2 - t_1\right)^2}{t_2 - t_1} \tag{4.2.12}$$

For the guess of initial angular velocity, a small non-zero guess can be used. This is because if a zero initial guess is used, divide-by-zero errors may be encountered while computing the drag and lift coefficients.

Chapter 5

Application

In this chapter, the methods from Chapter 4 are applied to practice and attempted to be performed automatically. Several algorithms are presented in order to achieve this. Finally, testing and analysis is demonstrated on actual ball trajectories to validate the methods.

5.1 Automated Camera Calibration

In this section two automated approaches for automated camera calibration are presented and summarised. We desire an automated method of camera calibration primary because in tennis ball tracking situations, the camera position and orientation will not always be fixed. Tennis robots that are moving will require recalibration at every change of state to localise itself, and similarly in broadcast situations, the camera will pan to follow the path of the ball.

A naive approach is firstly presented. Consider a ray extending extending from a candidate vanishing point at some angle ψ. We sample in some subset of all possible ψ, and for each sampled ψ, we then sample a thick band of points along the ray. Analysis is performed on these points to determine the likelihood of the ray being collinear with a white band corresponding to a court line, and for each ψ sampled, a likelihood score Σ is assigned. If the candidate vanishing point is the actual vanishing point, then 5 prominent peaks are expected in Σ, corresponding to the court lines (Figure 5.1). With this procedure, an iterative search can then be conducted on regions of the entire image to find the best possible location of the vanishing point matching to the signature plot. Finally, the terminal points of the white bands are found and camera calibration is applied using the aforementioned techniques.

While this algorithm performs soundly, in practice it is slow, since multiple nested iterations occur during the search process. Also, many tuning parameters are required, such as the angle and line sampling resolutions, as well as those within the objective function used to characterise Σ.

A more efficient Hough Transform-based approach is presented. Firstly, a sequence of images

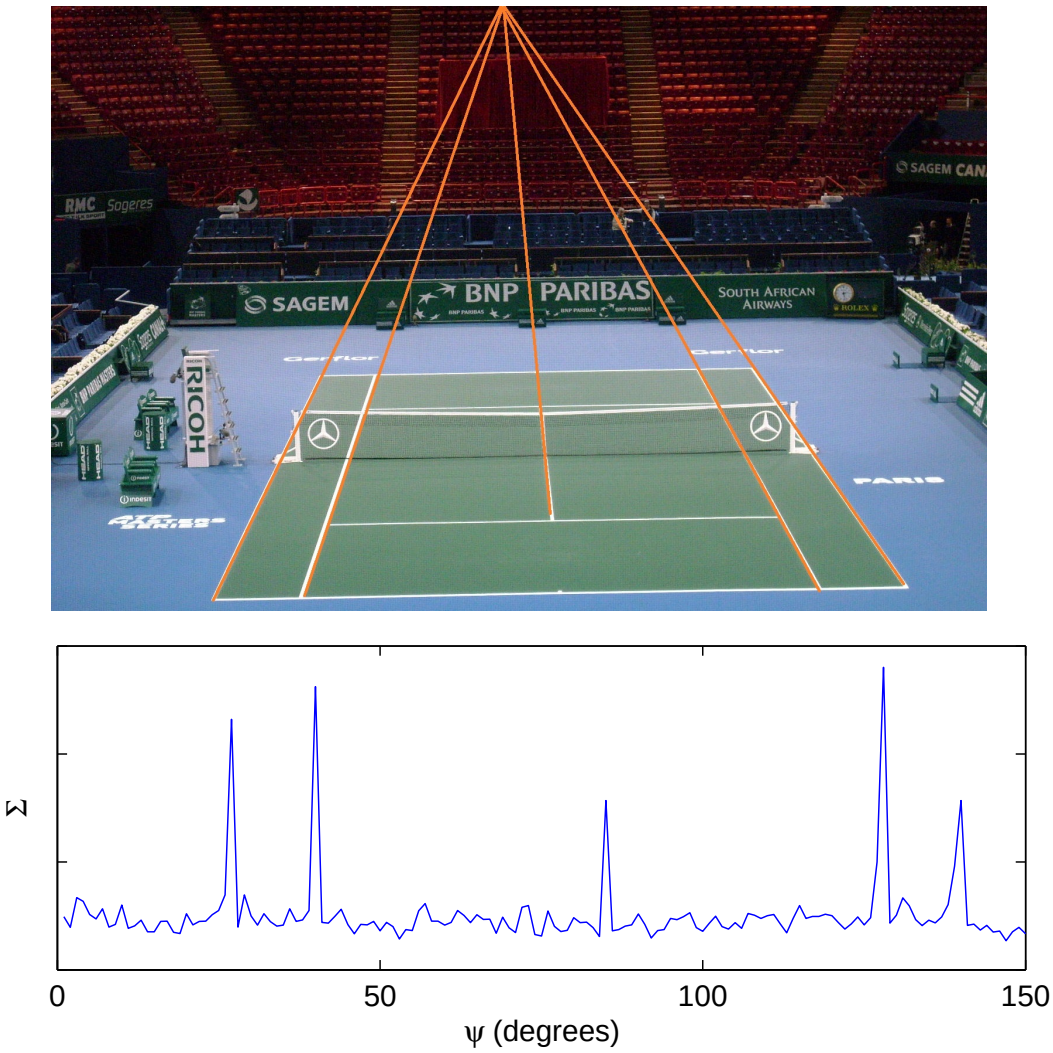


Figure 5.1: The signature plot of a vanishing point. The 5 peaks correspond to all court lines parallel to the Z-axis.

are median filtered to remove the players from the image (Figure 5.2a). As players move around the court during the point, using a median filter should filter out the players and result in a 'background' image for the court. Secondly, the median image is then transformed into a grayscale image as colour information is no longer important, and then blurred using an averaging filter on each pixel's neighbourhood (Figure 5.2b). This procedure is necessary to reduce the noise in the image. A tophat operation is then performed (Figure 5.2c). The tophat operation emphasises regions brighter than its surroundings, which is suitable for finding court lines [17]. The result is then binarised using Otsu's method [44] (Figure 5.2d). Region analysis is used to find and remove any small regions from the binarisation process, and then the Hough transform is used to identify all line segments (Figure 5.2e). An algorithm is used to find the minimum area quadrilateral that bounds all the identified

line segments, and this quadrilateral forms the template of the court outline. From this, it is then trivial to calculate the vanishing point by finding the intersection of opposing sides of the quadrilateral.

The result of camera calibration can be seen in Figure 5.2f. The red lines show the projection of the standardised court coordinates using the estimated perspective projection matrix. The green line shows the estimated projection of the net. Note that there is a close match with the actual net outline, which validates the technique for estimating the entire projection matrix $\mathbf{M}$ when only 9 parameters are found.

Compared to the naive method presented previously, this algorithm finds an estimate faster. However this method may also be more susceptible to noise since any misidentified line segment outside the court will alter the detected quadrilateral, leading to calibration error.

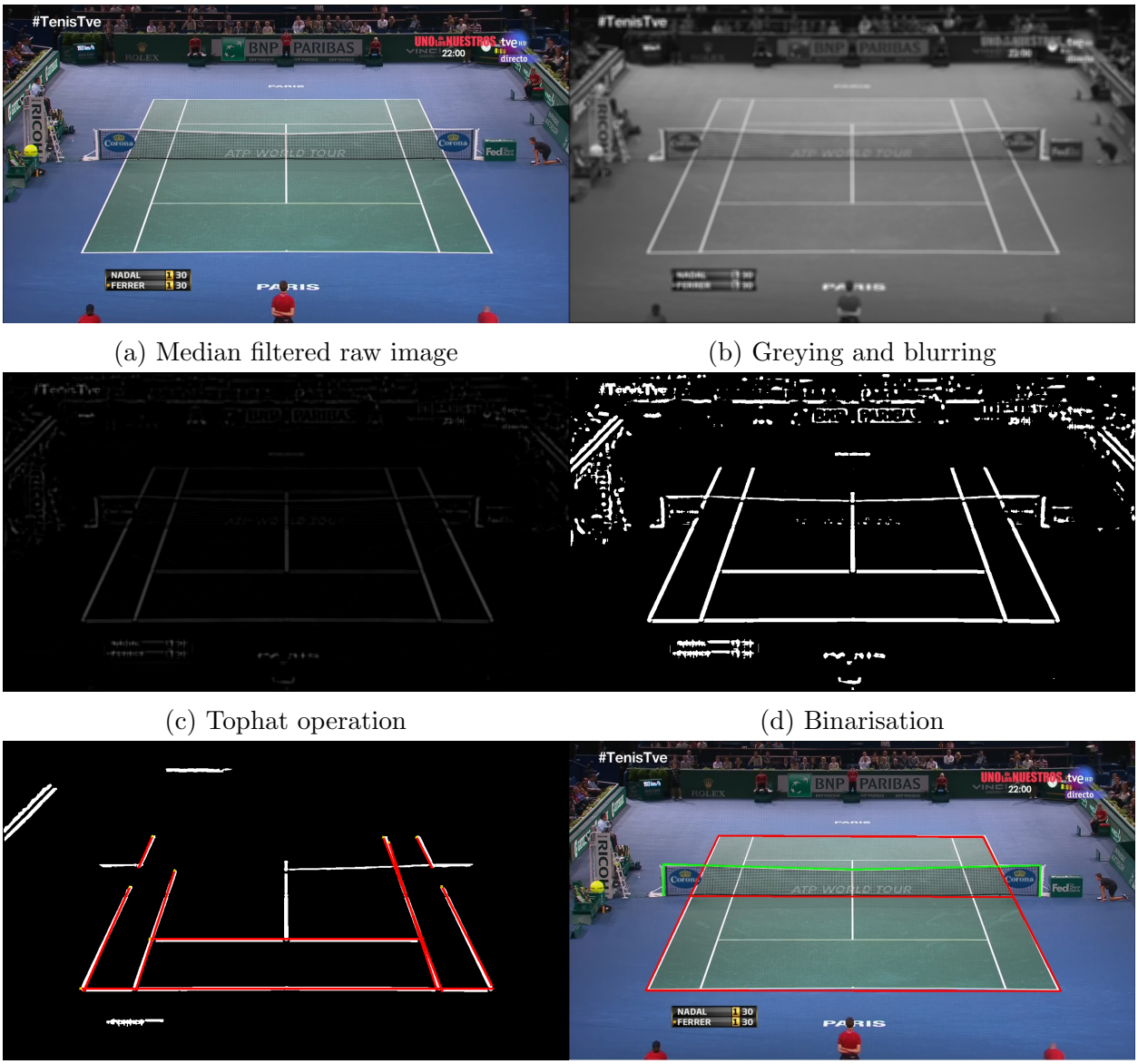
5.2 Automated Ball Tracking

We apply video tracking on a series of video clips containing highlights of a tennis match between Nadal and Ferrer at the 2013 Paris Masters. The purposes of choosing this selection of clips are

- The tournament is played indoors, so the effect of wind is small. This partially satisfies Assumption 3.
- The video is captured in 50 frames per second, so there is ample ball trajectory data. Also, it slightly reduces the effect of error by frame differencing.
- The tournament features a wide view of the court, so there will be minimal panning of the camera during points.
- Both participants in the match are known for playing primarily from the baseline, so associating events with data becomes easier if there are fewer occurrences of play at the net.

5.2.1 Video Processing

We begin video tracking by firstly isolating the motion of the ball. Frame differencing is used as the tool to segment motion in the video. This is achieved by taking the absolute difference in pixel values between successive frames (Figure 5.3b). Regions that contain stationary pixels, such as the court, will not be included in the differenced image, whereas regions that contain moving pixels such as the players and the ball will be included. This image is then converted to greyscale, and then blurred. A morphological closing is then performed on the binarised image, to transform the pixels representing the players into large regions. All resulting regions larger in area than some threshold (Figure 5.3c) are subtracted from the blurred greyscale image (Figure 5.3b). This isolates the final image to only motion



(a) Median filtered raw image

(b) Greying and blurring

(c) Tophat operation

(d) Binarisation

(e) Region filtered and lines identified

(f) Calibration

Figure 5.2: Image processing procedure.

of small objects (Figure 5.3d).

Simultaneously, player tracking can also be incorporated on the regions from the Figure 5.3c operation. A minimum area bounding box is assigned to the largest region in each of the lower and upper halves of the court. The bottom of the boxes become the estimated location of the players' feet. This estimate is smoothed using a low pass filter, because this estimate can occasionally be subject to bobbing up and down as players move to play a stroke.

(a) Raw image (b) Absolute difference

(c) Binarisation, closing and area thresholding (d) Ball motion isolation

Figure 5.3: Video processing procedure.

5.2.2 Particle Filter

To quantitatively track the ball in video, a particle filter was used. A more complicated particle filter was used for tennis ball video tracking in [45, 47], however we demonstrate ball tracking using a simpler particle filter.

We define our state vector by

$$\mathcal{X}_k^i := \begin{bmatrix} x_k^i & y_k^i & \dot{x}_k^i & \dot{y}_k^i & \ddot{x}_k^i & \ddot{y}_k^i \end{bmatrix}^T \tag{5.2.1}$$

A linear update during the prediction step simulates particles with acceleration and noise.

$$\mathcal{X}_{k+1}^i = \begin{bmatrix} 1 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathcal{X}_k^i + \mathcal{W}_k^i \tag{5.2.2}$$

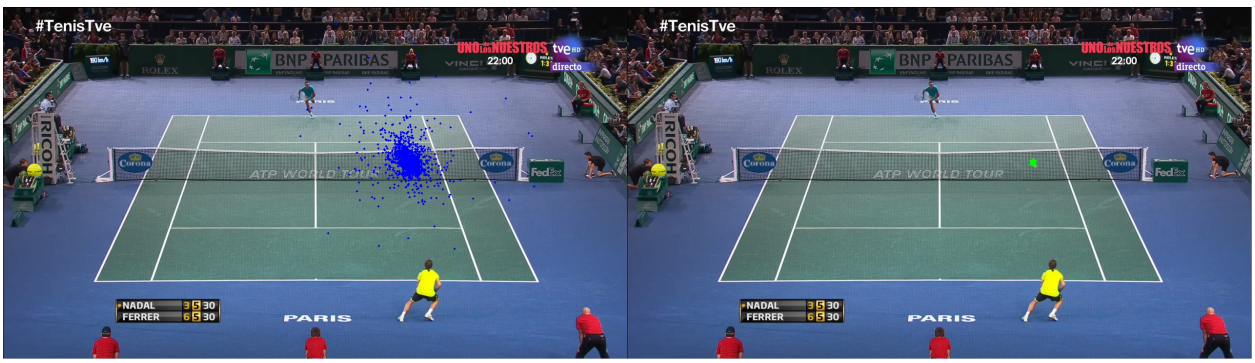
where $\mathcal{W}_k^i$ is selected to be Gaussian distributed noise for simplicity. For the likelihood

function, we use

$$P\left(\mathcal{Y}_k|\mathcal{X}_k^i\right) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right) \tag{5.2.3}$$

where $\epsilon \geq 0$ is a function that decreases with the intensity of the image at point (x_k^i, y_k^i). Hence the stronger the intensity of the differenced image, the higher the likelihood given the particle.

The estimated location of the ball $(\hat{x}_k, \hat{y}_k)$ at time k can be found using the average position of all particles $(\bar{x}_k, \bar{y}_k)$. A measure for the confidence of the estimate can also be calculated from the covariance $\mathrm{cov}\left(x_k, y_k\right)$, with a lower measure representing a higher confidence. A demonstration of particle filter tracking in tennis video is provided in Figure 5.4.



(a) The particle field

(b) The estimated state

Figure 5.4: Particle filter results.

In implementation, the particle filter is robust enough to retrack the ball after loss of tracking. Loss of tracking usually occurs when the ball is near or occluded by the players. In this case, the ball is removed by the morphological closing operation (Figure 5.3c) and the particle filter can no longer give a confident state estimation. Tracking may also momentarily be lost as the ball changes directions after impact. Because the nature of tennis rallies means for frequent changing of directions for the ball, it makes fitting an accurate dynamic model to ball motion difficult. However, this performance is hypothesised to be better compared to if an alternative filter were used, such as a Kalman filter.

A long rally consisting of 27 strokes was tracked using the particle filter along with player tracking. The blue curve in Figure 5.5 shows the x and y positions where the the ball was tracked with a high level of confidence, charted over time. The gaps in between the blue curves are the periods where the ball was tracked with a low level of confidence. Rather than rely on the low quality data, an algorithm was developed to fill in the missing data. For every gap, a polynomial was fitted from the left hand and right hand curves. Given that the motion of the ball must follow in a continuous curve, the location where these polynomials joined was determined to be the location that minimised the distance between the join of

the left polynomial and the right polynomial. This results in a natural looking fit in between the blue curves, although there are some aberrations that may have arisen from insufficient tracking.

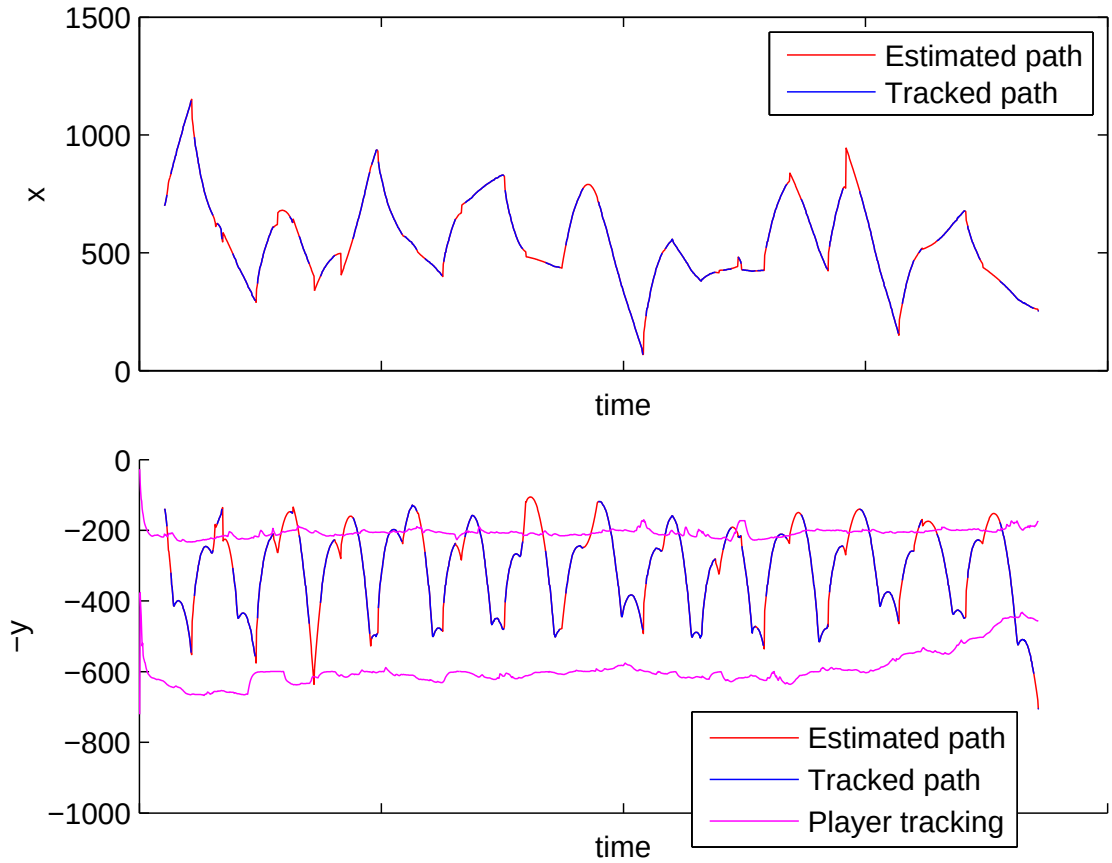


Figure 5.5: Ball and player tracking over the course of a rally.

5.2.3 Data Association

To associate tracked data to ball events such as a bounce or hit, a state chart for a rally is constructed in Figure 5.6. This state chart contains all possible transitions during a point, however for simplicity all "Volley", "Net" and "Miss" transitions shall be ignored. State transition logic is required to detect key events such as a "Bounce" or a "Serve/Hit". All tracks of data in between Serve-Bounce, Hit-Bounce or Bounce-Hit transitions are classified into a single trajectory, for which the kinematics of the trajectory can then be estimated. Judging from the y-value data from Figure 5.5, characteristic changes in the data such as local minima and maxima can signify a transition, based on the current y-location. Thus a relatively simple machine which detects events based on the y-value data can be developed,

however the abstractness of the the state chart allows for more complicated designs if necessary.

Figure 5.7 shows the y data from Figure 5.5 applied to the state chart with transition logic. The uppermost peaks represent the ground truth for hits, as does every second trough directly following each of these peak hits. The ground truth for all bounces come as either a local trough or peak in between the hits. As the figure shows, most events are correctly detected however there are still several misdetections and detection errors.

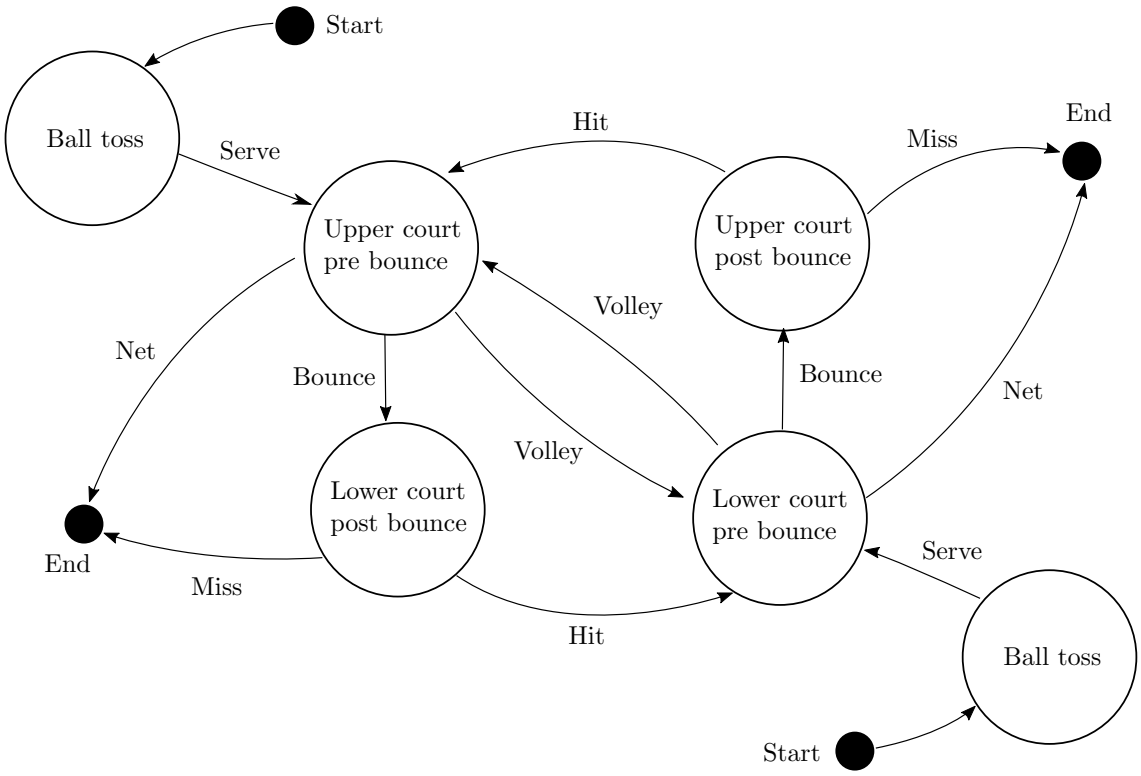


Figure 5.6: Full state chart for a rally.

5.3 Testing and Results

5.3.1 Stroke Analysis

We now perform trajectory reconstruction on 5 correctly detected strokes from Figure 5.7, each played by Nadal, who occupies the upper court in the video clip. The results are displayed in Table 5.1 and the reconstructed trajectories are shown in Figure 5.8.

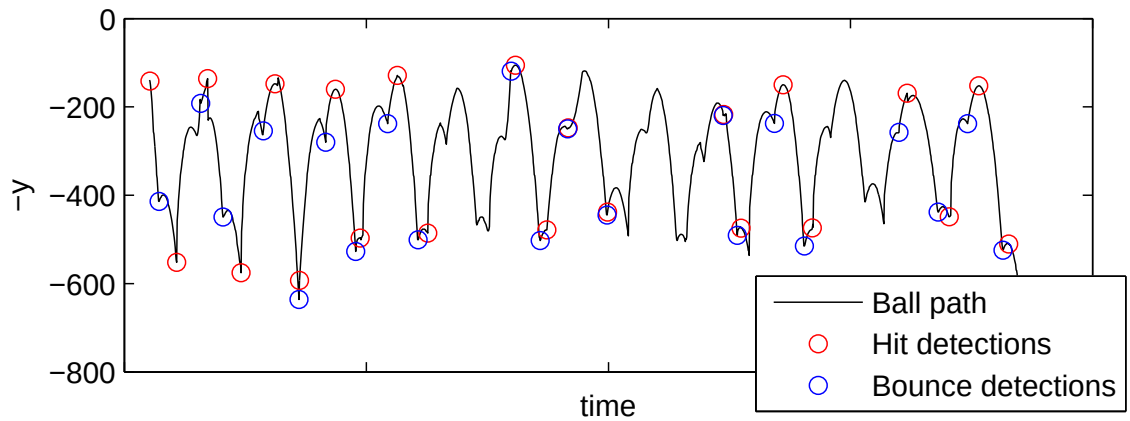


Figure 5.7: Bounce and hit detections using state transition logic on tracked y data.

Table 5.1: Table of estimated kinematics

Shot	Type	Speed (kmh^{-1})	Spin (rpm)	$t_f - t_0$ (s)	$\mathbf{v}_0$ (ms^{-1})	$\boldsymbol{\omega}_0$ (rad s^{-1})
1	Forehand	148	3260	0.84	$[-7.5 \quad 5.8 \quad -40]^T$	$[-188 \quad 60 \quad 278]^T$
2	Forehand	142	3300	0.86	$[2.1 \quad 5.3 \quad -39]^T$	$[-254 \quad 116 \quad 202]^T$
3	Backhand	104	390	1.02	$[-0.7 \quad 2.6 \quad -29]^T$	$[34 \quad -7 \quad -21]^T$
4	Forehand	128	610	0.88	$[6.3 \quad 4.0 \quad -35]^T$	$[-52 \quad 19 \quad -32]^T$
5	Backhand	114	440	1.00	$[-0.3 \quad 3.4 \quad -32]^T$	$[46 \quad 3 \quad 3]^T$

In terms of the validity of these results, Nadal is a player well known for being able to generate a large amount of spin on the forehand. Thus, spin magnitudes of around 3000 rpm are well within the possible range. Considerably less spin was generated on the two backhand shots, however the values are still not abnormal. The speeds also fall within the typical expected ranges.

5.3.2 Serve Analysis

We now compare automated trajectory reconstruction against manual trajectory reconstruction. Figure 5.9 shows two reconstructions of a single serve. In automated software reconstruction, the ball during the serve was tracked using the particle filter, segregated using the state chart and the data was estimated by optimisation. Only data with high tracking confidence was used in the estimate. In manual trajectory reconstruction, the ball tracking was hand coded however player tracking was still performed automatically.

Speed radars are installed at most tennis stadiums, so we are also able to use the radar

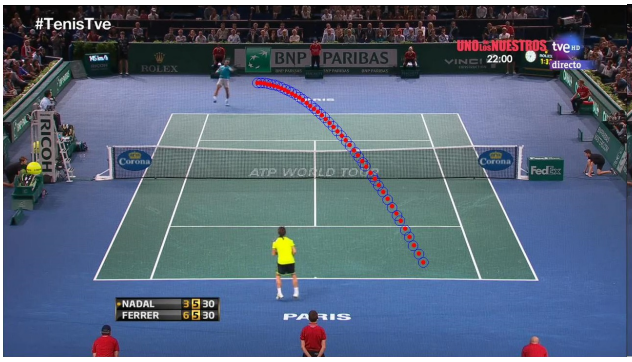
readings to validate these reconstructions. Table 5.2 lists these results. The recorded spins of around 5000 rpm which are within the admissible range for second serves. We also see that the manually reconstructed speed of 157 kmh^{-1} yields a slightly higher speed than the radar reading of 145 kmh^{-1}. If we make the argument that the radar gun uses a Doppler radar pointed in a direction parallel to to Z-axis, then the radar gun may perhaps ignore components of velocity in any tangential direction. Given that the manual estimate is an absolute measure, we can extract the Z-component of velocity and compare that, giving 151 kmh^{-1}. The manual estimate is now within the same range as the radar reading allowing for some tolerance. However, the automated estimate is very far from the radar reading. The performance of trajectory reconstruction is generally degraded as the transition is made from manual to automated estimation. We identify the following sources of errors and discrepancies during the automated process, and in trajectory reconstruction in general.

- Inaccurate tracking from the particle filter, caused by the ball moving too quick or being occluded by the player.
- Variations in tracking due to the random nature of the particle filter.
- Frame differencing causes both the current and previous ball to show in the same image. This may cause tracked ball location to be skewed.
- Mistimings in the detection of key events. A 180 kmh^{-1} serve misdetected by 0.02 s can cause an error of 1 m.
- The discrete sampling of video meaning that the exact moment of impact for the ball is not captured.
- Motion blur in the camera making it difficult to pinpoint the exact location of the ball at a particular instant.
- A mismatch between the dynamic model of the ball and the actual dynamics present in the video.
- Slight errors in camera calibration also causing small errors in projection.
- Violations of any of the assumptions.

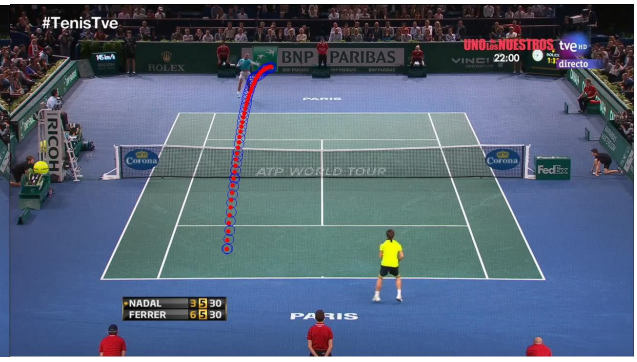
Despite this, the results of Table 5.2 and the closeness between the manual and radar figures show that in principle, higher dimensional information of an object can be estimated from a sequence of 2D images.

Table 5.2: Serve speed readings

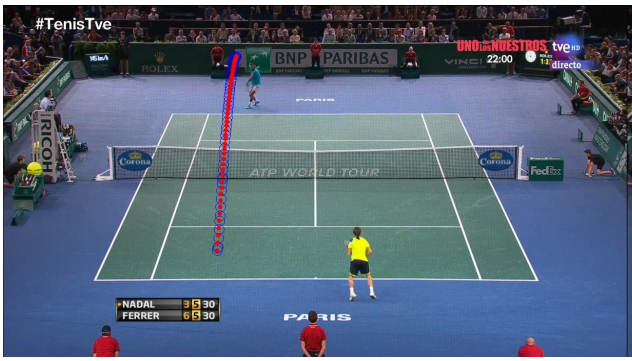
Source	Speed (kmh^{-1})	Spin (rpm)
Radar	145	-
Manual	157	5350
Manual (Z-component)	151	-
Automated	191	4910



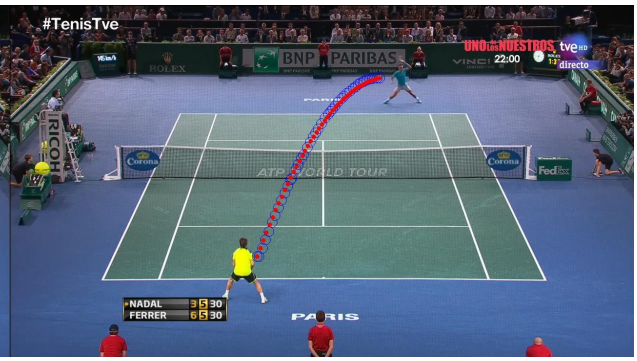
(a) Shot 1 (Forehand)



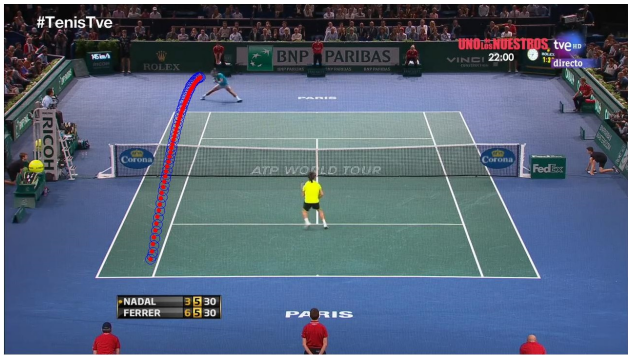
(b) Shot 2 (Forehand)



(c) Shot 3 (Backhand)

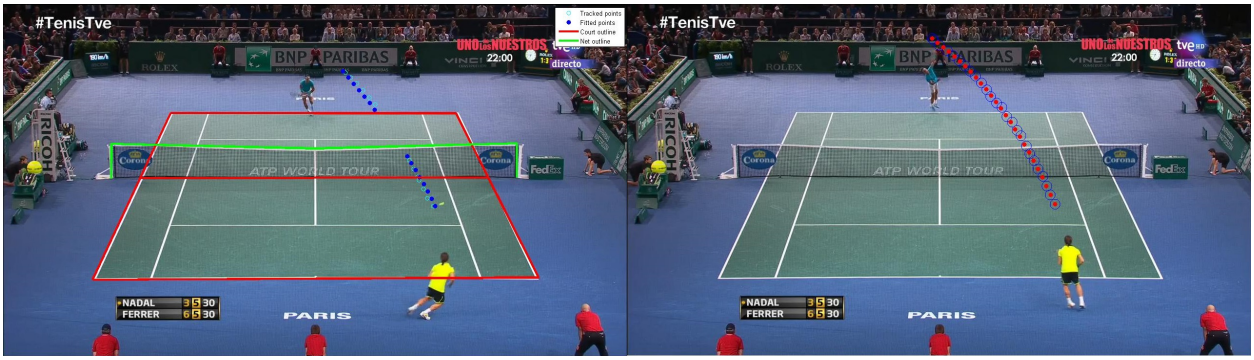


(d) Shot 4 (Forehand)



(e) Shot 5 (Backhand)

Figure 5.8: Trajectory reconstruction. The blue circles consist of the tracked data, and the red dots consist of the projection of the estimated trajectory.



(a) Automated reconstruction (b) Manual reconstruction

Figure 5.9: Trajectory reconstruction. The blue circles consist of the tracked data, and the red dots consist of the projection of the estimated trajectory.

Chapter 6

Conclusion

We see that the 3 aims as stated in Chapter 1 have been achieved. In Chapter 3, equations describing the motion of the tennis ball were found. In Chapter 4, two algorithms were presented, which were then validated in Chapter 5. The most significant outcomes from this project are demonstrating the ability to systematically infer accurate 3D information from 2D trajectories, when there are seemingly a multitude of possible solutions. The modularity in design of this project also means that different ball tracking, data association and court identification algorithms can be substituted without affecting its fundamental operation. To that end, we now identify key areas for improvement in this project, and direction for future work.

Firstly, the ball tracking and data association algorithms are imperative to successful automated trajectory reconstruction. As was seen, the automated method performed poorly when compared to the idealised case from manual tracking. This highlights that there is more potential for improvement in the algorithms, and perfect tracking can yield substantial results. Secondly, the court calibration and ball tracking algorithms were tested on a very small subset of tennis courts. Different environments, such as outdoors, and clay or grass surfaces will introduce new challenges that need to be accounted for. Techniques to make the the algorithms more robust should be investigated. Machine learning approaches may be one potential field that can help automated classification tasks perform as well as humans.

We envisage these methods applied in a video-based form of system identification. For example, if it is known that the ball has some state prior to a bounce and another state after, then the coefficients of restitution of the court surface can be empirically determined. Likewise, this could be used to find bounce characteristics of balls and surfaces in other sports, such as basketball, soccer and table tennis.

Although it was not considered in this project, fast and efficient algorithms can be a desired feature in any scenario. Future research could explore different software or hardware setups that can achieve online performance of the tracking and estimation algorithms. In terms of being able to use this system in a robotic or mechatronic application, we are able to advise

that the single camera approach will be considerably less accurate, due to the prevalence of numerous errors as discussed. We make the recommendation of a multiple camera setup in any scenario where precision is a requirement, because triangulation of coordinates will reduce errors. A stereo camera setup can be investigated, because this is a viable option on an autonomous robot. In any regard, the contributions from this project will be able to pave the way for future advancements.

Bibliography

[1] F. Alam, A. Subic, J. Naser, M. G. Rasul, and M. M. K. Khan. A study of spin effects on tennis ball aerodynamics. *WSEAS Transactions on Fluid Mechanics*, 3(3):271–278, 2008.

[2] F. Alam, W. Tio, S. Watkins, A. Subic, and J. Naser. Effects of spin on tennis ball aerodynamics: an experimental and computational study. In *16th Australasian Fluid Mechanics Conference, 2007. Proceedings.*, pages 324–327, 2007.

[3] F. Alam, S. Watkins, and A. Subic. The aerodynamic forces on a series of tennis balls. In *15th Australasian Fluid Mechanics Conference, 2004. Proceedings.*, pages 13–17, 2004.

[4] F. Alam, S. Watkins, and A. Subic. Effects of spin on aerodynamic properties of tennis balls. In *The 5th International Conference on the Engineering of Sport, 2004. Proceedings.*, volume 1, pages 83–89. International Sports Engineering Assocation, 2004.

[5] R. Berger. Method of and system for determining position of tennis ball relative to tennis court, and tennis ball provided therefor, January 21 1992. US Patent 5,082,263.

[6] C. Bialik. Even pretending to be a professional tennis player is terrifying. `http://fivethirtyeight.com/features/even-pretending-to-be-a-professional-tennis-player-is-terrifying/`, July 2015. [Online; accessed 20-July-2015].

[7] H. Brody. The moment of inertia of a tennis ball. *The Physics Teacher*, 43:503–505, 2005.

[8] V. Caglioti and A. Giusti. Ball trajectory reconstruction from a single long-exposure perspective image. In *Workshop on Computer Vision Based Analysis in Sport Environments, 2006. Proceedings.*, 2006.

[9] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[10] S. G. Chadwick and S. J. Haake. Determination of the coefficient of drag of three different tennis balls using a wind tunnel. In *16th International Symposium on Biomechanics in Sports, 1998. Proceedings.*, pages 436–439. International Society of Biomechanics in Sports, 1998.

[11] H. T. Chen, H. S. Chen, and S. Y. Lee. Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. Proceedings.*, volume 1, pages 1097–1100. IEEE, 2007.

[12] H. T. Chen, C. L. Chou, W. J. Tsai, and S. Y. Lee. 3d ball trajectory reconstruction from single-camera sports video for free viewpoint virtual replay. In *Visual Communications and Image Processing, 2011. Proceedings.*, pages 1–4. IEEE, 2011.

[13] H. T. Chen, M. C. Tien, Y. W. Chen, W. J. Tsai, and S. Y. Lee. Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video. *Journal of Visual Communication and Image Representation*, 20(3):204–216, 2009.

[14] A. J. Cooke. An overview of tennis ball aerodynamics. *Sports Engineering*, 3(2):123–129, 2000.

[15] J. Cortes. Lecture notes in nonlinear controllability, May 2010.

[16] R. Cross. Ball trajectories. In *The Physics and Technology of Tennis*, pages 367–374. Racquet Tech Pub.

[17] B. D. Ekinci and M. Gokmen. A ball tracking system for offline tennis videos. In *1st WSEAS International Conference on Visualization, Imaging and Simulation, 2008. Proceedings.*, pages 45–48. World Scientific and Engineering Academy and Society, 2008.

[18] M. Y Fang, C. K. Chang, N. C. Yang, C. M. Kuo, and S. K. Guang. Robust player tracking for broadcast tennis videos with adaptive kalman filtering. *Journal of Information Hiding and Multimedia Signal Processing*, 5(2):242–261, 2014.

[19] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Pearson, 2005.

[20] S. R. Goodwill, S. B. Chin, and S. J. Haake. Aerodynamics of spinning and non-spinning tennis balls. *Journal of Wind Engineering and Industrial Aerodynamics*, 92(11):935–958, 2004.

[21] S. R. Goodwill and S. J. Haake. Aerodynamics of tennis balls-effect of wear. In *The 5th International Conference on the Engineering of Sport, 2004. Proceedings.*, volume 1, pages 35–41. International Sports Engineering Assocation, 2004.

[22] S. J. Haake, S. G. Chadwick, R. J. Dignall, S. Goodwill, and P. Rose. Engineering tennis-slowing the game down. *Sports Engineering*, 3(2):131–143, 2000.

[23] T. Kim, Y. Seo, and K. S. Hong. Physics-based 3d position analysis of a soccer ball from monocular image sequences. In *Sixth International Conference on Computer Vision, 1998. Proceedings*, pages 721–726. IEEE, 1998.

[24] F. Klvana. Trajectory of a spinning tennis ball. In *Solving Problems in Scientific Computing Using Maple and MATLAB®*, pages 27–37. Springer, 2004.

[25] A. Kumar, P. S. Chavan, V. K. Sharatchandra, S. David, P. Kelly, and N. E. O'Connor. 3d estimation and visualization of motion in a multicamera network for sports. In *Irish Machine Vision and Image Processing Conference, 2011. Proceedings.*, pages 15–19. IEEE, 2011.

[26] W. Lao, J. Han, and P. H. N. With. Ball-path inference based on a combination of audio and video clues in tennis video sequences. In *Benelux/DSP Valley Signal Processing Symposium, 2006. Proceedings.*, pages 1–4. IEEE, 2006.

[27] J. Mao. Tracking a tennis ball using image processing techniques. Master's thesis, University of Saskatchewan, 2006.

[28] R. Mehta, F. Alam, and A. Subic. Review of tennis ball aerodynamics. *Sports technology*, 1(1):7–16, 2008.

[29] R. D. Mehta and J. M. Pallis. The aerodynamics of a tennis ball. *Sports Engineering*, 4(4):177–189, 2001.

[30] J. Metzler and F. Pagel. 3d trajectory reconstruction of the soccer ball for single static camera systems. In *IAPR International Conference on Machine Vision Applications, 2013. Proceedings.*, pages 4–22. IAPR, 2013.

[31] E. Ota. Algorithm development using MATLAB - object tracking by particle filter, 2015.

[32] N. Owens, C. Harris, and C. Stennett. Hawk-eye tennis system. In *International Conferenceon Visual Information Engineering, 2003. Proceedings.*, pages 182–185, 2003.

[33] G. Palmer. *Physics for Game Programmers.* Apress, 2005.

[34] J. Pansiot, A. Elsaify, B. Lo, and G. Z. Yang. RACKET: Real-time autonomous computation of kinematic elements in tennis. In *IEEE 12th International Conference on Computer Vision Workshops, 2009. Proceedings.*, pages 773–779. IEEE, 2009.

[35] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *15th International Conference on Pattern Recognition, 2000. Proceedings.*, volume 4, pages 152–156. IEEE, 2000.

[36] G. S. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings.*, pages 260–265. IEEE, 1998.

[37] A. Poliakov, D. Marraud, L. Reithler, and C. Chatain. Physics based 3d ball tracking for tennis videos. In *International Workshop on Content-Based Multimedia Indexing (CBMI), 2010. Proceedings.*, pages 1–6. IEEE, 2010.

[38] I. D. Reid and A. North. 3d trajectories from a single viewpoint using shadows. In *British Machine Vision Conference, 1998. Proceedings.*, pages 51–52, 1998.

[39] C. Shachar, E. Khazanov, and B. Z. Yoram. Smart-court system and method for providing real-time debriefing and training services of sport games, February 21 2013. US Patent App. 14/370,006.

[40] A. Stepanek. The aerodynamics of tennis ballsthe topspin lob. *American Journal of Physics*, 56(2):138–142, 1988.

[41] P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. Proceedings.*, volume 1. IEEE, 1999.

[42] K. Teachabarikiti, T. H. Chalidabhongse, and A. Thammano. Players tracking and ball detection for an automatic tennis video annotation. In *11th International Conference on Control Automation Robotics & Vision, 2010. Proceedings.*, pages 2461–2494. IEEE, 2010.

[43] Y. Tsuji and Y. Muguruma. Numerical simulation of motion of table tennis ball; effect of ball diameter. *Table Tennis Science*, 4&5:38–53, 2002.

[44] E. Turkel. Lecture notes in automatic thresholding, 2011.

[45] F. Yan. *Tennis Ball Tracking for Automatic Annotation of Broadcast Tennis Video.* PhD thesis, University of Surrey, 2004.

[46] F. Yan, W. Christmas, and J. Kittler. Ball tracking for tennis video annotation. In *Computer Vision in Sports*, pages 25–45. Springer.

[47] F. Yan, W. Christmas, and J. Kittler. A tennis ball tracking algorithm for automatic annotation of tennis match. In *British Machine Vision Conference, 2005. Proceedings.*, volume 2, pages 619–628, 2005.

[48] F. Yan, J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox, and Q Huang. Automatic annotation of tennis games: An integration of audio, vision, and learning. *Image and Vision Computing*, 32(11):896–903, 2014.

[49] F. Yan, A. Kostin, W. Christmas, and J. Kittler. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006. Proceedings.*, volume 1, pages 634–641. IEEE, 2006.

[50] X. Yu, N. Jiang, and E. L. Ang. Trajectory-based ball detection and tracking with aid of homography in broadcast tennis video. In *Visual Communications and Image Processing, 2007. Proceedings.* International Society for Optics and Photonics, 2007.

[51] X. Yu, C.H. Sim, J. R. Wang, and L. F. Cheong. A trajectory-based ball detection and tracking algorithm in broadcast tennis video. In *IEEE International Conference on Image Processing, 2004. Proceedings.*, volume 2, pages 1049–1052. IEEE, 2004.

[52] Z. Zhang. Camera calibration. In *Emerging Topics in Computer Vision*, pages 4–43. Prentice Hall.

[53] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[54] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *14th annual ACM international conference on Multimedia, 2006. Proceedings.*, pages 431–440. ACM, 2006.

List of Figures

2.1 A definition of coordinate conventions. 7

3.1 The differences for C_D in experimental data found by Alam et. al. [2], Goodwill et. al. [20] and the correlation by Cross [16]. 15
3.2 The differences for C_L in experimental data found by Alam et. al. [2], Goodwill et. al. [20] and the correlation by Cross [16]. 15
3.3 Trajectory simulation from an initial condition of 40 ms^{-1} and 300 $rad{\cdot}s^{-1}$. . . 17

5.1 The signature plot of a vanishing point. The 5 peaks correspond to all court lines parallel to the Z-axis. 25
5.2 Image processing procedure. 27
5.3 Video processing procedure. 28
5.4 Particle filter results. 29
5.5 Ball and player tracking over the course of a rally. 30
5.6 Full state chart for a rally. 31
5.7 Bounce and hit detections using state transition logic on tracked y data. . . 32
5.8 Trajectory reconstruction. The blue circles consist of the tracked data, and the red dots consist of the projection of the estimated trajectory. 34
5.9 Trajectory reconstruction. The blue circles consist of the tracked data, and the red dots consist of the projection of the estimated trajectory. 35

List of Theorems

1 Definition 6
2 Definition 6
3 Definition 6
1 Assumption 6
1 Proposition 8
2 Assumption 8

3 Assumption 16
4 Assumption 17

4 Definition 18
5 Assumption 20
6 Assumption 21
7 Assumption 21

Appendix A

Table of Constants

Table A.1: Constant values used throughout the project

Symbol	Value	Units
ℓ_1	4.11	m
ℓ_2	5.49	m
ℓ_3	6.4	m
d_1	6.4	m
d_2	11.89	m
h_0	0.914	m
h_1	1.07	m
R	0.033	m
m	0.057	kg
ρ	1.21	kg m^{-3}
μ	1.983 $\times 10^{-5}$	kg m^{-1}s^{-1}
g	-9.81	kg ms^{-2}
α	0.535	-

Appendix B

Observability

Consider a simplified linear state space system for ball motion in three translational directions

$$\begin{bmatrix} \dot{\chi}_1 \\ \dot{\chi}_2 \\ \dot{\chi}_3 \\ \dot{\chi}_4 \\ \dot{\chi}_5 \\ \dot{\chi}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \\ \chi_5 \\ \chi_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g \\ 0 \\ 0 \end{bmatrix} = f(\boldsymbol{\chi}) + g(.) \tag{B.0.1}$$

and nonlinear output equation

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = H(\chi_1, \chi_3, \chi_5) \tag{B.0.2}$$

We now show that the system is not locally observable (that is, it is not possible to reconstruct the states by observing the output for any given state). For local observability, we require

$$\dim(d\mathcal{O}(\boldsymbol{\chi})) = 6 \tag{B.0.3}$$

where $d\mathcal{O}(\boldsymbol{\chi}) := \text{span}(\nabla\mathcal{H}(\boldsymbol{\chi}) | \mathcal{H} \in \mathcal{O})$ and $\mathcal{O}$ is the observation space in $\mathbb{R}^6$ containing H_1, H_2 and the lie derivatives $\mathcal{L}_f\mathcal{L}_g H_1$, $\mathcal{L}_f\mathcal{L}_g H_2$ [15]. However

$$\sup\{\dim(d\mathcal{O}(\boldsymbol{\chi}))\} = 4 \tag{B.0.4}$$

so the system is not locally observable for any $\boldsymbol{\chi}$.

Hence by observing the motion of the ball in image coordinates, no information about the ball in world coordinates can be explicitly obtained.

Appendix C

Alternative Cost Function

The MATLAB function `fmincon` can perform slowly. Here is presented an alternative cost function with an alternative optimisation algorithm, using soft constraints. Let the cost function J be defined as affine with a quadratic in the trajectory error, and with a quadratic in the terminal error.

$$J = \beta \text{ trace}\left(\left(\mathbf{Q} - \mathbf{W}\right)^T \left(\mathbf{Q} - \mathbf{W}\right)\right) + (1 - \beta)\left(\mathbf{r}_{Bf} - \begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix}\right)^T \left(\mathbf{r}_{Bf} - \begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix}\right) \tag{C.0.1}$$

with $0 \leq \beta \leq 1$, determining the weighting placed on the terminal constraint. Then the optimisation problem is declared as

$$\min_{\mathbf{v}_0, \boldsymbol{\omega}_0} J\left(\mathbf{v}_0, \boldsymbol{\omega}_0\right) \tag{C.0.2}$$

This may be used with the nonlinear unconstrained optimisation solver `fminunc` in MATLAB, which implements the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm. In practice, it does give an estimate faster than `fmincon`, however the terminal constraint will not always be satisfied.