Extracting Field Hockey Player Coordinates using a Single Wide-Angle Camera

David William Higham

A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the degree of Doctor of Philosophy

October 2017

Collaborating Organisations: Engineering and Physical Sciences Research Council, English Institute of Sport and England Hockey

Abstract

In elite level sport, coaches are always trying to develop tactics to better their opposition. In a team sport such as field hockey, a coach must consider both the strengths and weaknesses of both their own team and that of the opposition to develop an effective tactic. Previous work has shown that spatiotemporal coordinates of the players are a good indicator of team performance, yet the manual extraction of player coordinates is a laborious process that is impractical for a performance analyst. Subsequently, the key motivation of this work was to use a single camera to capture two-dimensional position information for all players on a field hockey pitch.

The study developed an algorithm to automatically extract the coordinates of the players on a field hockey pitch using a single wide-angle camera. This is a non-trivial problem that requires: 1. Segmentation and classification of a set of players that are relatively small compared to the image size, and 2. Transformation from image coordinates to world coordinates, considering the effects of the lens distortion due to the wide-angle lens. Subsequently the algorithm addressed these two points in two sub-algorithms: Player Feature Extraction and Reconstruct World Points.

Player Feature Extraction used background subtraction to segment player blob candidates in the frame. 61% of blobs in the dataset were correctly segmented, while a further 15% were over-segmented. Subsequently a Convolutional Neural Network was trained to classify the contents of blobs. The classification accuracy on the test set was 85.9%. This was used to eliminate non-player blobs and reform over-segmented blobs.

The Reconstruct World Points sub-algorithm transformed the image coordinates into world coordinates. To do so the intrinsic and extrinsic parameters were estimated using planar camera calibration. Traditionally the extrinsic parameters are optimised by minimising the projection error of a set of control points; it was shown that this calibration method is sub-optimal due to the extreme camera pose. Instead the extrinsic parameters were estimated by minimising the world reconstruction error. For a 1:100 scale model the median reconstruction error was 0.0043 m and the distribution of errors had an interquartile range of 0.0025 m. The Acceptable Error Rate, the percentage of points that were reconstructed with less than 0.005 m of error, was found to be 63.5%.

The overall accuracy of the algorithm was assessed using the precision and the recall. It found that players could be extracted within 1 m of their ground truth coordinates with a precision of 75% and a recall of 66%. This is a respective improvement of 20% and 16% improvement on the state-of-the-art. However it also found that the likelihood of extraction decreases the further a player is from the camera, reducing to close to zero in parts of the pitch furthest from the camera. These results suggest that the developed algorithm is unsuitable to identify player coordinates in the extreme regions of a full field hockey pitch; however this limitation may be overcome by using multiple collocated cameras focussed on different regions of the pitch. Equally, the algorithm is sport agnostic, so could be used in a sport that uses a smaller pitch.

Acknowledgements

I would like to thank Dr Simon Goodwill, Dr John Kelley and Dr Chris Hudson for their support in completing this study. They have offered guidance on key decisions and have kept the study focussed.

I would also like to thank the Engineering and Physical Sciences Research Council and English Institute of Sport for sponsoring this work. Further to this, the study would not have been possible without the dataset provided by England Hockey.

Team Chestnut, you have provided a vital opportunity to rant when I have needed it. You have given me support when things weren't going well. But most importantly you have all been good friends! Just imagine how quickly this could have been done without Escape Room Fridays! CHESTNUT ASSEMBLE! BarOne for one?

And finally I would like to thank my Mum, Julia, Dad, Chris, and the rest of my family who have supported me through my entire education. I'm sure they are fed up with my whining and I'm pretty sure they thought I would remain a perpetual student; however they have stayed rock steady and supported my decisions throughout. Thank-you. X

Table of Contents

1 Introduction

1.1 Overview of Project

This PhD has been completed as part of a project, sponsored by the Engineering and Physical Sciences Research Council (EPSRC) (Engineering and Physical Sciences Research Council 2017) and the English Institute of Sport (EIS) (English Institute of Sport 2017a), to improve performance in Olympic Sport. Field hockey, one of the sports selected for UK Sports' World Class Performance Programme (UK Sport 2017), was granted two PhDs to improve the likelihood of achieving a medal in future Olympics. The first of these, (McInerney 2017), investigated the performance metrics that predict the outcomes in elite field hockey. Most of these performance metrics are formulated from the player's pitch coordinates at an instance in time. Subsequently, this PhD attempted to automate the process of player coordinate extraction using a single camera; a necessary step for GB Hockey to calculate the defined performance metrics.

1.2 What is Field Hockey?

(Dictionary.com 2015) describes field hockey as "a game played on a rectangular field having a netted goal at each end, in which two teams of eleven players each compete in driving a small leather-covered ball into the other's goal". The team that scores the most goals is declared the winner. While similar games have been played around the globe for 4000 years ago, the modern sport developed in English schools during the 18th century (International Hockey Federation 2015).

International field hockey is played on a 91.4 m x 55 m artificial pitch (Figure 1.1). The pitch must be uniform in colour with white line markings. The game consists of four 15 minute quarters, with a change in sides at half-time. In the modern game the ball is made of solid plastic.

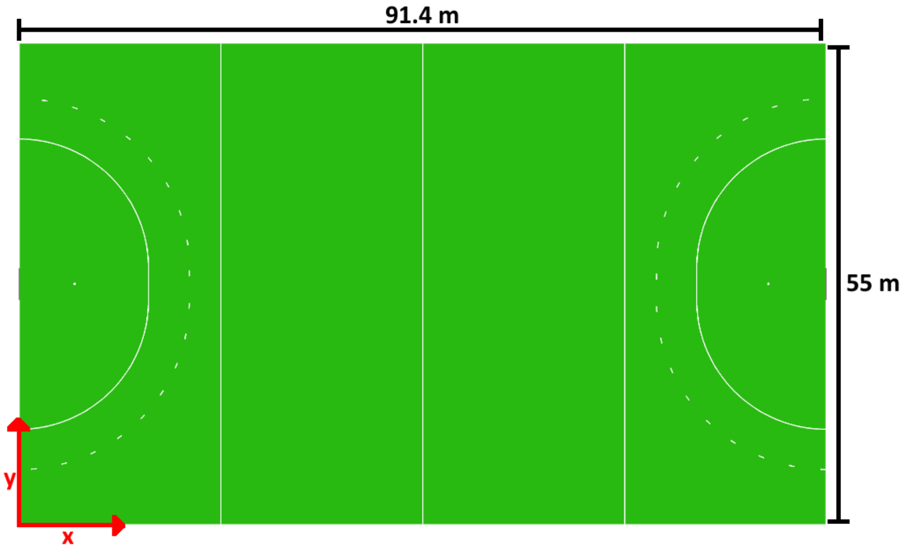


Figure 1.1: The coordinate system used throughout this thesis and the dimensions of a hockey pitch.

Each team consists of ten outfield players, who wear matching uniforms, and a goal keeper, who wears a different uniform. The opposing teams must wear different coloured uniforms. To ensure this, each team has several different coloured kits available to them. In addition, two umpires wear uniforms that contrast with both teams.

The International Hockey Federation (FIH) is the world governing body for the sport. They "exist to raise the global status and popularity of Hockey" (International Hockey Federation 2015) and have a "mandate to the rules and regulations". They organise international tournaments, the pinnacle of which is the quadrennial Olympic competition. The first men's Olympic competition was held at the 1908 London games.

The women’s game had to wait until the 1980 Moscow games to make its first appearance. Great Britain (GB) Hockey is “responsible for the development and administration of hockey in Great Britain related to the summer Olympic Games” (Great Britain Hockey 2015). A collaboration between the hockey federations of England, Scotland and Wales, it aims “to achieve the ultimate performance goal for hockey in Great Britain, Olympic Games success.”

1.3 Performance Analysis within GB Hockey

The EIS state that “Performance Analysis is a specialist discipline involving systematic observations to enhance performance and improve decision making, primarily delivered through the provision of objective statistical (Data Analysis) and visual feedback (Video Analysis).” (English Institute of Sport 2017b). The GB Hockey men’s and women’s teams participate in tournaments around the world. A performance analyst travels with each team to generate objective and visual feedback. They film matches with a single High Definition Pan Tilt Zoom (PTZ) video camera, which enables them to achieve high pixel resolution images of areas of interest. The filming position varies between matches and is often unknown prior to arrival. Post-match, the videos are ‘coded’ in the performance analysis software Sportscode (Sportstec 2015), with timestamps of key events that have been deemed indicators for success. This is an indexing process that allows easy retrieval of video segments of interest for the coach and players to review.

In addition to the video analysis, GB Hockey players wear a Catapult (Catapult 2017) Global Positioning System (GPS) unit during training and matches. Each unit records it’s spatiotemporal coordinate at 10 Hz. Specific to field sports such as hockey, spatiotemporal tracking of players is a vital tool used in the assessment of performance (Leser et al. 2011). This data can be employed at the micro level, to analyse the physiological demands on a player, or the macro level to inform on the formation and tactics a team is employing.

A coach develops a team's tactics to limit the weaknesses in their own team while exploiting those in the opposition team. Only through understanding both teams can this be effective.   While the players of other elite teams also wear GPS units during matches (FIH 2014), the performance benefits of having access to the data means no data sharing agreement exists.

Cameras are an un-intrusive hardware solution that can, with accurate calibration, provide a player's spatiotemporal data without the need for an additional worn sensor. The lack of a necessity for a worn sensor means that positional measurements can be collected for both teams simultaneously. GB Hockey's use of a PTZ camera allows a focus on a single event, while sacrificing the context across the entire pitch relative to this event. It is this context that is vital for observing patterns in a team's play and as such developing tactics that counter these patterns. A single static camera or network of static cameras that cover the whole pitch can provide this whole pitch context.

FIH regulation does not limit the performance analysts to a single camera; however they are usually restricted to a single camera position. Further to this, the regular travel demanded by international hockey means the installation and use of a distributed multi-camera system is impractical. Therefore this thesis will focus on providing individual player coordinates using a single camera. Figure 1.2 illustrates the camera position used to collect the video footage used throughout this thesis.

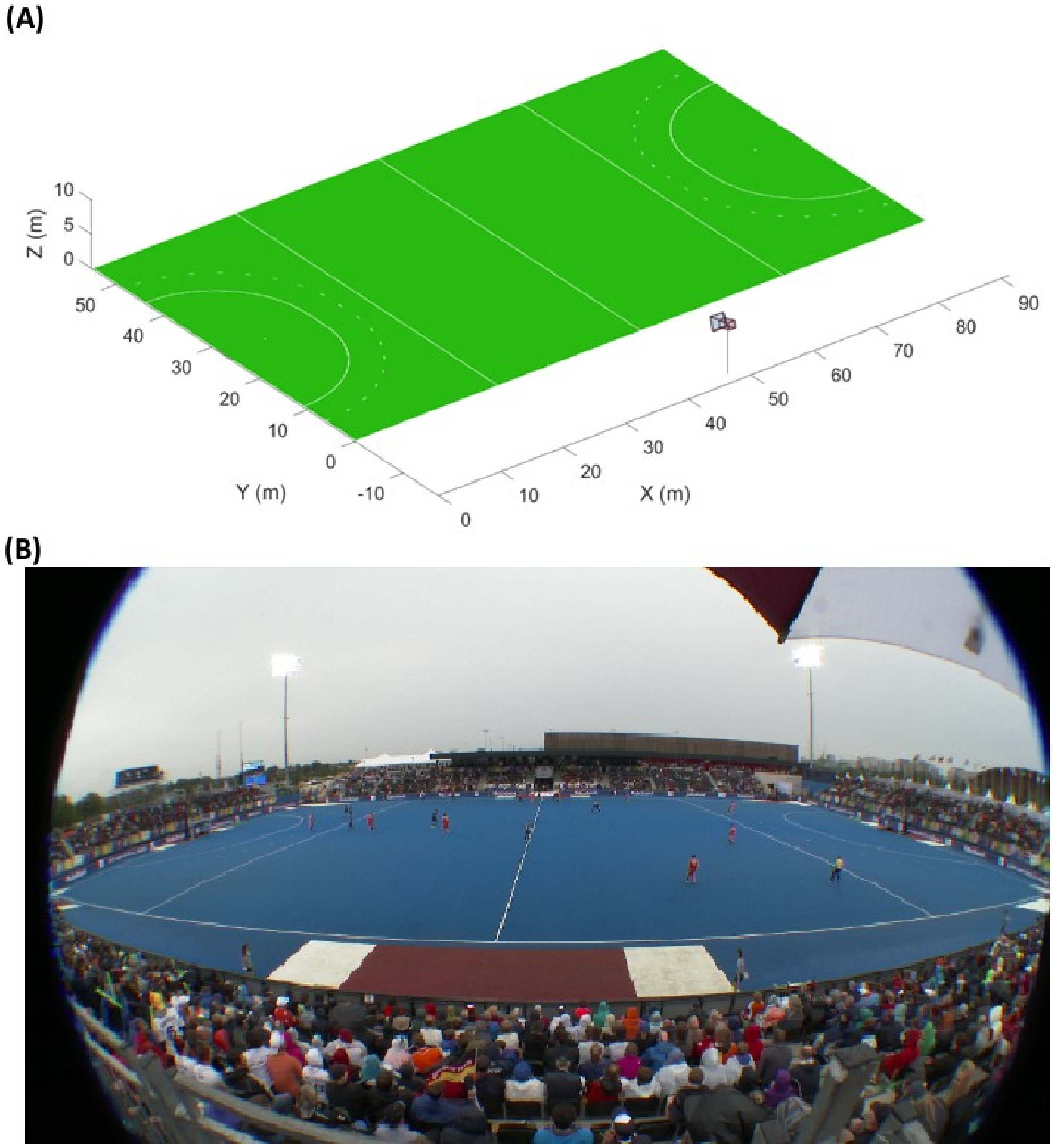


Figure 1.2 (A) The performance analyst's location (X= 48 m, Y= -15m, Z= 7m) at a recent international field hockey tournament. (B) A video frame captured from a camera located at the position in (A).

1.4 Automation of Vision Based Performance Analysis

Manually locating an object in each frame of a video is a time-intensive process. Due to the difficulty of identifying the movements of the correct player in a group of players, anecdotal evidence suggests that accurately identifying one player at 5 Hz for a 15 minute quarter of field hockey takes on average 10 hours. Assuming knowledge of all 22 players is required, it is impractical for a performance analyst to manually identify each of the player's coordinates across the entire quarter. Therefore, if GB Hockey is to use the performance metrics defined by (McInerney 2017), it is a necessity that the time dedicated to identification is vastly reduced. The automation of the identification process using computer vision techniques would reduce this time. Subsequently GB Hockey requires a computer vision algorithm to accurately extract the spatiotemporal coordinates for each of the players for both teams on a hockey field using a single camera.

1.5 Thesis Roadmap

Figure 1.3 illustrates the high level algorithm required to extract the player's world coordinates from match footage. The algorithm can be decomposed into 2 sub-algorithms: (1) Player Feature Extraction, and (2) Reconstruct World Points. This thesis will consider both of these sub-algorithms. Subsequently Chapter 2 highlights the key existing literature for these two components. Chapters 3 and 4 focus on Player Feature Extraction. Chapter 3 identifies a suitable method of segmentation for a field hockey dataset. Chapter 4 proposes the use of a convolutional neural network to classify if a segmented blob contains a hockey player. Chapters 5 – 9 consider Reconstruct World Points. Chapter 5 presents a novel method for identifying a set of world known points on a calibrated plane. Chapter 6 shows that from the expected camera pose the reconstruction accuracy can be improved by minimising the transformation from image coordinates to world coordinates. This is contrary to literature where the inverse of the transformation from world coordinates to image coordinates is typically used. Chapter 7 investigates the effect of control point identification errors on the reconstruction accuracy. Chapter 8 investigates the effect of camera hardware on the reconstruction accuracy. Chapter 9 determines the effect of camera pose on the reconstruction accuracy, important for a performance analyst to determine the suitability of their capture position. Finally Chapter 10 analyses the accuracy with which the player coordinates can be extracted.

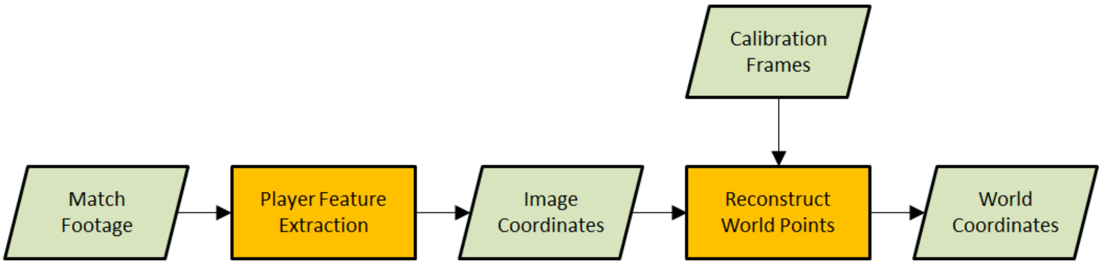


Figure 1.3: The algorithm to automate the process of player coordinates extraction. Data is indicated by parallelograms. Sub-algorithms are indicated by rectangles.

1.6 Research Outputs

The following was published as part of this work:

- Higham, D., Kelley, J., Hudson, C., & Goodwill, S. R. (2016). Finding the Optimal Background Subtraction Algorithm for EuroHockey 2015 Video. In *Procedia Engineering* (Vol. 147, pp. 637–642)

2 Literature Review

This chapter investigates the previous research into vision based sports player tracking. It begins by highlighting the requirements of a performance analyst and the constraints of an international hockey tournament. It then investigates existing camera based performance analysis tools. These tools share a common workflow: (1) Extract a feature representation for each of the players in a single image, (2) Transform the spatial representation from image coordinates to world coordinates, and (3) Associate detections across multiple images into trajectories. This thesis is only concerned with the first two of these and as such a section of this review is dedicated to each. The chapter concludes by reiterating the aim of the thesis and listing the objectives that are needed to achieve this aim.

2.1 Performance Analyst's Requirements

GB Hockey requires a system that can accurately extract all the players of both teams on the hockey field. The accuracy of an extraction system is the percentage of the detections that match the true locations of the players; however this requires a threshold to determine the maximum distance permitted for a detection to match a true location. This threshold distance, the acceptable range, should be selected based upon the needs of the system. Here, this is how much positional error can be in the player's coordinates for the data still to be useful for tactical analysis. (McInerney 2017) notes that in field hockey the acceptable range is dependent upon the 'footprint' of the player and suggests a level of ±0.5 m is reasonable given the reach of the stick.

Data capture at international hockey tournaments is constrained by:

1. The analysis position provided by the tournament organisers. This position is not standard across tournaments and is unknown a priori.
2. The availability of space in which to place the equipment. The performance analysts normally have to share the space with the performance analysts of the other nations competing in the tournament.
3. The international transportation and installation of the capture equipment. The performance analysts do not have prior access to the stadium and must setup and pack down all equipment for each match captured.

Given these constraints GB Hockey require a single camera solution that can easily be deployed in stadia around the world.

Figure 1.2 illustrates the performance analyst's location relative to the pitch at a recent international tournament (X= 48 m, Y= -15 m, Z= 7 m).

2.2 Existing Vision Based Performance Analysis Tools

Research specifically on the tracking of hockey players using vision based methods is limited; however the dimensions of the field of play and the problems encountered are similar to those experienced in other field sports. Therefore this literature review will investigate tracking in all field sports and particularly in football, where a lot of research effort has been dedicated.

Vision based performance analysis approaches are common for in-competition data collection as they are un-intrusive and allow capture of the players on both teams. They record a scene with one or more video cameras. The footage, the sequence of images captured, can then be ‘coded’ to provide further information such as each of the player’s coordinates in the scene. Some commercial systems, such as Sportscode (Sportstec 2015) used by GB Hockey, provide an easy interface for indexing of video and key performance statistics. Others, such as STATS SportVU (STATS 2017) add an element of computer vision to minimise the need for human interaction in the coordinate identification process. Throughout this thesis the term camera will be used to refer to digital video cameras.

Currently the performance analysts at GB Hockey capture the matches using a single Pan Tilt Zoom (PTZ) camera. This is a compact solution that can easily be transported and installed at a venue as required. It allows the analyst to focus on a particular region of interest at the expense of limiting the field of view. The pitch coordinates of players outside the field of view are unknown. Further to this the use of a PTZ camera makes extracting world coordinates more time-consuming as a model to transform from image coordinates to world coordinates must be estimated for each image.

STATS SportVU (STATS 2017), a commercial football solution, estimates the (X, Y) coordinates of each player at 25 Hz using three collocated static cameras. Each camera is focussed on a different section of the pitch to ensure full pitch coverage. The system can be extended with a second three camera installation to provide three-dimensional coordinates and reduce the likelihood of occlusion; a player’s coordinates being unknown due to another player passing between them and the camera. However the

SportVU license is expensive (£150,000 for three years) and the system requires a semi-permanent installation of the cameras in the stadium. In addition the system requires installation at the midpoint of the long dimension of the pitch. As noted earlier it cannot be assumed that the performance analyst will have access to this camera location. For these reasons SportVU is an impractical solution for GB Hockey.

Other performance analysis systems extract player trajectories from broadcast footage (Beetz et al. 2007; Liu et al. 2006; Tong et al. 2011; Lu et al. 2013). These solutions are popular as they do not require specific filming access. However broadcast footage is typically captured using a PTZ camera and suffers the same limitations as the performance analyst's existing method. It generally has a narrow angle of view and the camera's attention focusses on the ball. While this makes for more exciting spectator viewing, it does not allow a player's actions to be analysed in the context of the entire pitch. This wider view is necessary when assessing and developing tactics. Further to this, broadcast footage of complete field hockey matches is rarely available. For these reasons, broadcast footage cannot meet the requirements of GB Hockey and will be discounted from further consideration.

Another solution is to use a single static camera that captures the entire pitch. A standard camera lens is designed to be a rectilinear projection; it projects straight lines in the world as straight lines in the image. The maximum field of view of a rectilinear camera lens is approximately 122° (Zhang 2016). A camera's required angle of view can be calculated by Equations (1) and (2):

$$Angle\ of\ View = 2\tan^{-1}\frac{a}{2r} \tag{1}$$

$$r = \sqrt{Y^2 + Z^2} \qquad (2)$$

Where a is the width of the pitch, Y is the Y component of the camera position, and Z is the Z component of the camera position. Given this equation and the camera position in Figure 1.2, the required angle of view is approximately 140°. A rectilinear camera lens is insufficient to capture the entire pitch from this position; therefore a wide-angle lens must be used. A wide-angle lens is designed to have a wider angle of view by following a different projection model. The distortion due to this projection model must be corrected before accurate coordinates can be extracted. The projection model and methods for correcting it are considered in Section 2.5.

(Erdmann 1992) described a method to extract kinematic data from football and athletics using a wide-angle lens. This method manually identified player's coordinates using a grid system but only at a low spatial resolution. This was a time consuming process that required markings that are not FIH compliant.

(Hudson 2015) used a wide-angle lens to assess the clean swim phase of a swimming race. To do so he automatically estimated the swimmers coordinates in an Olympic swimming pool. While the principle remains the same when applied to a hockey pitch, the area is approximately four times larger than an Olympic swimming pool (50 m x 25 m). As a result, assuming the same camera position a wider angle of view is required and the relative size of players in the image is smaller.

To the author's knowledge there is no existing research into using a wide-angle lens to capture an entire field sport pitch with the aim of automatically extracting the coordinates of the players. This leads to the aim of this thesis:

Develop the algorithm necessary to extract player coordinates from footage captured with a single wide-angle camera at a field hockey tournament.

Most Multi Object Tracking (MOT) systems use a similar algorithm to that illustrated in Figure 2.1. This algorithm is composed of three distinct sub-algorithms:

- **Player Feature Extraction** - Detect each player in an image and extract a feature representation. Typically this feature representation is the image coordinates but may also incorporate appearance statistics.
- **Reconstruct World Points** - Transform a set of image coordinates to world coordinates. This requires a camera model.
- **Trajectory Formulation -** Associate extractions into temporal trajectories.

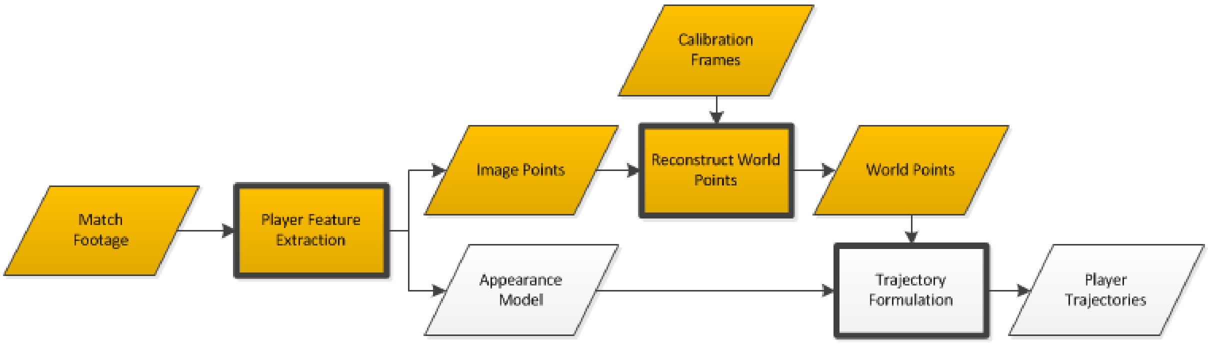


Figure 2.1: Orange: The scope of this thesis, the algorithm to automate the process of coordinates extraction. Data is indicated by parallelograms. Sub-algorithms are indicated by emboldened rectangles. The algorithm is composed of two sub-algorithms: Player Feature Extraction and Reconstruct World Points. White: How coordinates extraction may fit into a Multi Object Tracking framework.

The scope of this thesis is Player Feature Extraction and Reconstruct World Points; the part of Figure 2.1 highlighted in orange. Subsequently, the following sections of this chapter investigate previous research into Image Point Extraction and Reconstructing World Points; however first, the next section will investigate the possible camera assemblies and assess their ability to achieve the aim.

2.3 Camera Assembly

The camera assembly is defined as the combination of the camera and the lens system used. The camera assembly must meet a set of criteria to be suitable to achieve the aim. The most important criterion is a sufficient angle of view. As highlighted in the previous section an angle of view of approximately 140° is required. If this criterion is not met, the angle of view will not cover the whole pitch and the solution would not be suitable for all required camera positions.

A second criterion when choosing the camera assembly is the resolution of the camera. Presently the most common camera resolution is High Definition (HD), 1920 pixels x 1080 pixels; however Ultra High Definition (UHD) is becoming more prevalent. UHD defines two different resolutions: (1) 4K which has 4 times as many pixels as HD, 3940 pixels x 2160 pixels, and (2) 8K which has 4 times as many pixels as 4K, 7680 pixels and 4320 pixels. Generally 8K cameras are currently only used in film production environments but 4K cameras are becoming more conventional in consumer electronics. If the image resolution is doubled in both dimensions, for example from HD to 4K, it follows that an object in the image will be represented by 4 times as many pixels. This larger representation will be more descriptive of the true shape and texture of the object and as such a more accurate feature model can be extracted. This is especially apparent here, as the players are relatively small compared to the pitch and so have small pixel areas. The increase in resolution comes at a cost. It results in larger files, a lower framerate for the same data bandwidth and for pixel-wise image processing algorithms, an increase in computation.

Another criterion is the quality of the footage produced by the camera. Ideally each image in the footage is a true representation of the scene at an instant of time; however the quality of the image produced is dependent upon the quality of camera's sensor, the optical system and the compression of the datum.

A camera sensor is a matrix of photo sensor cells that convert light into electrical charge. Each pixel in the image is formed from the electrical charge from a cell or group of cells. Increasing the size of a photo sensor increases its light sensitivity (Farrell et al. 2006) thereby increasing the signal-to-noise ratio, resulting in a more accurate image.

The optical system of the camera focusses light from the scene onto the sensor. As noted in the last section, wide angle lenses follow a different projection model to achieve a wider angle of view at the expense of added barrel distortion. Non-linear camera calibration must be applied to estimate and then correct this distortion. Optical aberration is the departure of the performance of a lens from the predictions of the projection model. It is caused by the light from a single point not converging on a single point on the sensor and leads to blurring of the image. A lens should have minimal optical aberration so the images are crisp and blur free.

Once the image has been created, the camera's data bandwidth and storage capacity determine how the images are formed into footage and written to memory. Typically the data bandwidth and storage capacity are not sufficient to produce footage from sequences of the raw images. Subsequently compression must be applied to footage, potentially resulting in lower quality images.

The final criterion is the practicality of a performance analyst using the camera. A performance analyst is expected to capture footage of a match in all weather conditions, from an unpredictable analysis location, with limited space, at stadia around the globe. Consequently the camera must be relatively robust to the elements, adaptable to viewpoint, compact and easily installed at a venue.

The five widely available camera types in Table 2.1 were assessed using the outlined criteria.

Table 2.1: Criteria Matrix for the five camera types considered.

Camera	Example	Required Angle of View	Maximum Resolution	Image Quality	Practicality
Camcorder	Sony FDR-AX33 (Sony 2017)	No	4K (3840 x 2160)	Medium	High
Camcorder + wide-angle lens adaptor	Sony FDR-AX33 + Raynox HDP-2800ES (Sony 2017; Raynox 2017)	Yes	4K (3840 x 2160)	Medium	High
Action Camera	GoPro Hero 3+ Black (GoPro 2017)	No	4K (3840 x 2160)	Low	Medium
IP Camera	Axis P1428-E (Axis Communications 2017)	No	4K (3840 x 2160)	High	Medium
Machine Vision Camera	Basler acA3800-14uc + C125-0418-5 (Basler 2017a; Basler 2017b)	Yes	4K (3840 x 2748)	Very High	Low

The camcorder alone does not provide the sufficient angle of view; therefore the wide-angle lens adaptor is necessary. This wide-angle lens adapter, which is mounted on the front of existing lens system, causes optical aberration close to the periphery of its circular image. Despite manufacturer claims, the angle of view of the action camera is not sufficient. No wide-angle 4K single IP cameras were available. The necessary angle could be achieved using a HD IP camera, but due to the drop in resolution this is deemed unsuitable. As only the camcorder with wide-angle lens adaptor and machine vision camera provide the required angle of view, only they will be considered with the further criteria.

Both cameras can achieve at least a 4K resolution. The machine vision camera provides the higher image quality. It has a larger sensor and it provides a stream of uncompressed images. This gives a high image quality but consequently a high data rate, so more storage is required. The image quality of the camcorder with wide-angle lens adaptor is lower. The sensor is smaller and it compresses the images when forming them into a sequence. An advantage of this is a reduction on the amount of necessary storage.

Practically the camcorder with wide-angle lens adaptor is superior. It has built in storage and battery, so can act as a standalone unit. It has optical zoom so the analyst is able to maximise the angle of view dependent upon camera location. The machine vision camera requires external storage and is powered over USB 3.0 and as such requires additional hardware. The performance analysts often have to work in restricted space, so additional hardware is not always possible. The machine vision camera uses a fixed focal length lens and so has no optical zoom. Even with a set of lenses the performance analyst is unable to make small changes to the focal length to maximise the angle of view.

Given the practical considerations it is deemed that the camcorder with wide-angle lens adaptor is the most suitable camera to capture footage at field hockey tournament. The rest of the chapter will investigate the sub-algorithms necessary to extract player coordinates from this footage.

2.4 Player Feature Extraction

The first step in the coordinates extraction workflow presented in Figure 2.1 is to extract the image coordinates for each player in a frame. This can be divided into: (1) detect each player in an image and (2) extract the player's coordinates. As such this section is split into two subsections. The first subsection investigates methods for the detection of players in the image. Given these detections the subsequent subsection explores methods to extract the coordinates. Throughout this review, player will be used a general term for any person on the pitch. Typically this includes the players of both teams and the umpires.

2.4.1 Player Detection

Player detection is finding the regions of the image that are players. These regions, also called blobs, should encompass all of the player pixels, while minimising non-player pixels. This allows the extraction of a reliable player representation with limited non-player noise. Player detection is similar to the field of pedestrian detection, which is in turn a subset of object detection. This review will focus on player detection methods, but will be supplemented with some more general object detection research.

A lot of research has been focussed on pedestrian detection. Much of this is in the field of autonomous vehicles, where a static camera cannot be assumed. The most advanced dataset is the Caltech Pedestrian Detection Dataset (Dollár et al. 2012). This dataset provides footage of resolution 640 pixels x 480 pixels, captured from a camera mounted on a vehicle moving in an urban environment. In the footage any pedestrians of at least 20 pixels in height are annotated. Pedestrian detection algorithms are then assessed by the average miss-rate of the annotations given 0.1 false positive detections per image (Dollár et al. 2012). Considering pedestrians of size 30-80 pixels, the expected size of the hockey players in this work, the state-of-the-art achieves an average miss-rate of 33% (Du et al. 2017). However as noted in (Carr et al. 2012), state-of-the-art algorithms struggle with the complex body poses observed in hockey. This dataset is not representative of these poses and so inferior results are to be expected on hockey poses.

Generally object detection algorithms can be further decomposed into two sub-algorithms. The first sub-algorithm segments the image into potential regions of interest. The second sub-algorithm classifies these potential regions of interest by object type.

Segmentation Algorithms

The simplest method for region finding is the exhaustive search. As used in (Viola & Jones 2004; Dalal & Triggs 2005), this supposes that an object can be any part of the image. Therefore the image is split into a set of sub-images and the algorithm applied to each sub-image. As the relative size of the object in the image is not fixed, the operation must be repeated at a pyramid of different sub-image sizes. This is computationally expensive and becomes even more so as the resolution of the image is increased.

Selective search (Uijlings et al. 2013) use a greedy algorithm to group initially small regions into larger regions (Figure 2.2). The grouping is based on a variety of complementary measures. Selective search reduces the computation expense and the number of regions when compared to exhaustive search, however it still requires analysis at different scales.

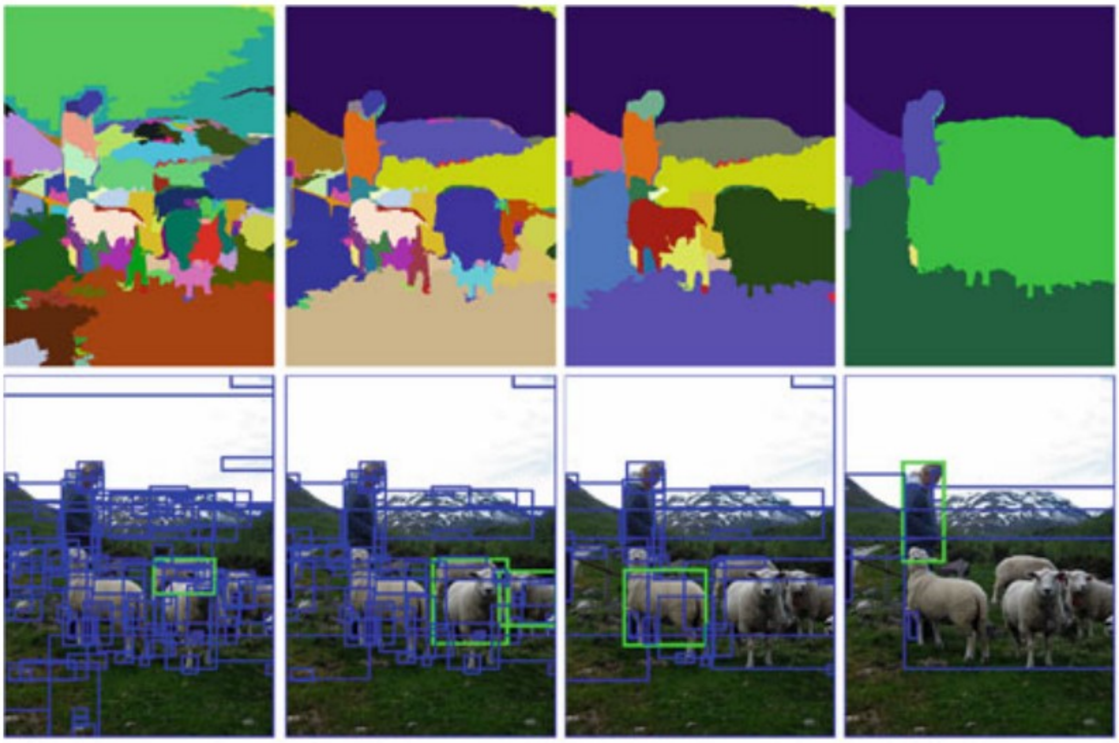


Figure 2.2: The output of the selective search algorithm at four different scales. (Modified from source: (Uijlings et al. 2013)**)**

Exhaustive search and selective search make no assumption on the contents of the image. A field hockey pitch must be of uniform colour; this fact can be exploited to define a model for a pitch pixel. If a pixel matches this model it is defined as a pitch pixel, or more generally a background pixel, and is deemed non-interesting. If a pixel does not match this model it is defined as a non-pitch pixel, or more generally a foreground pixel, and is deemed interesting. By applying this approach, the algorithm to find regions of interest can be decomposed as:

A. Classify each individual pixel as foreground or background.

B. Group foreground regions of pixels into player regions.

(Seo et al. 1997) determine the pitch colour for each of the red, green and blue (RGB) channels as the peak of their respective histogram over all image pixels. A pixel is then defined as background if each of the channels is within some threshold of this peak and the green channel is greater than both the red and blue channel. (Ekin & Tekalp 2003) propose an algorithm that is robust to variation in the dominant colour of the field, weather and lighting conditions. However the pitch markings are also a different colour to the pitch colour and as such are classified as foreground.

An alternative is to model regions of the background individually, thereby having a different model for the pitch markings. As a static camera is to be used, it can be assumed that a pixel is always capturing the same physical point of the background. Subsequently a model can be developed for the background at that pixel. If any deviation from this model is detected the pixel is classified as foreground. An algorithm that learns this model is known as a background subtraction algorithm.

In general background subtraction algorithms find the difference between the current frame and a model frame. This model frame may be as simple as the previous frame or may be more complex, such as modelling each pixel's colour statistics by a Gaussian Mixture Model (Stauffer & Grimson 1999; Zivkovic 2004). (Sobral & Vacavant 2014) review the state-of-the-art in background subtraction. C++ implementations of many of the algorithms are available at (Sobral & Bouwmans 2014).

(Higham et al. 2016) optimised seven pixel-wise background subtraction algorithms and one generic background model algorithm for a field hockey dataset. These eight algorithms were chosen from a set of approximately 30 algorithms, due to their existing use in sports video analysis. For each algorithm, the performance affecting parameters were optimised using particle swarm optimisation (Kennedy & Eberhart 1995). The number of individuals in the swarm and the number of generations were taken from (Erik et al. 2010). The results, displayed in Figure 2.3, indicate that the temporal median (Cucchiara et al. 2003) returns the highest F-score, a measure of the accuracy of the pixel-level classification. The assessment criteria for segmentation are discussed in the following section.

Rather than the pitch's colour, (von Hoyningen-Huene 2011) modelled its uniformity. A 3 x 3 kernel is moved over a grayscale image and the variance computed. A Gaussian Mixture Model is then applied to each pixel of this variance image, thereby learning the expected variance at a pixel. However as the resolution of the image increases, so does the likelihood that a player will have an area of uniformity the size of the kernel. Subsequently the size of the kernel should be dependent upon the image resolution.

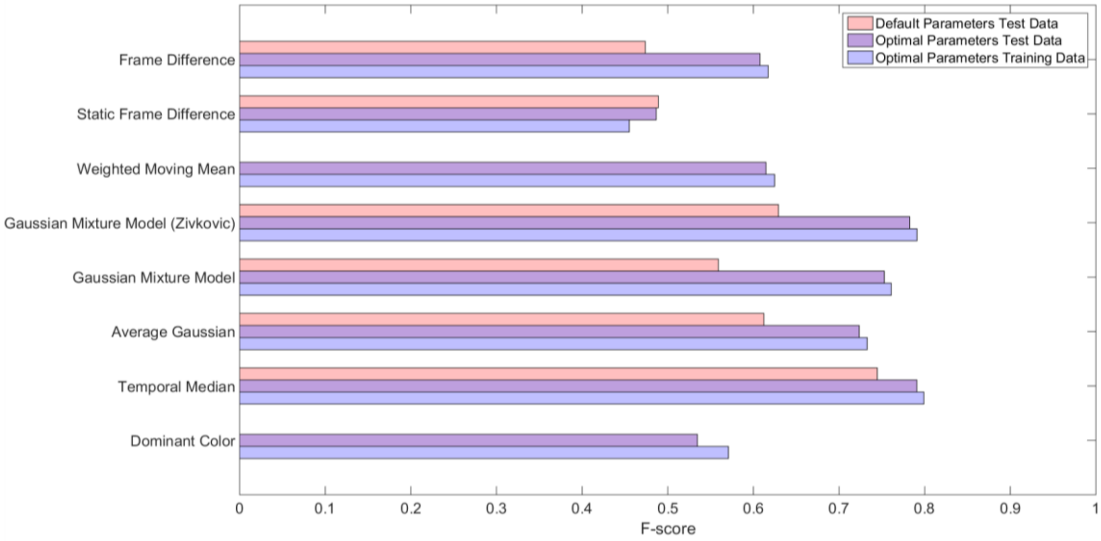


Figure 2.3: The F-scores for background subtraction algorithms optimised on a field hockey tournament dataset. (Source: (Higham et al. 2016)**)**

Once the pixels have been classified, the foreground pixels can be grouped to form blobs. This is performed using 8-connectivity. If any of a pixel's 8 neighbours are also foreground they are labelled in the same blob.

Due to a coincidental similarity between a player pixel and the background model a player may be over-segmented (Figure 2.4) into multiple separate blobs. An approach to resolve this is proposed in (Intille & Bobick 1995). They merge any blobs that are within a specified pixel distance of one another. Another approach is to close the regions between the blobs using morphological operations. Both of these methods are parameterised to the size of the player. As a large range of player sizes is expected, a single parameter cannot resolve all cases. Instead a range of parameters should be used dependent upon expected player size. Yet neither of these methods considers if semantically the blobs should be merged.

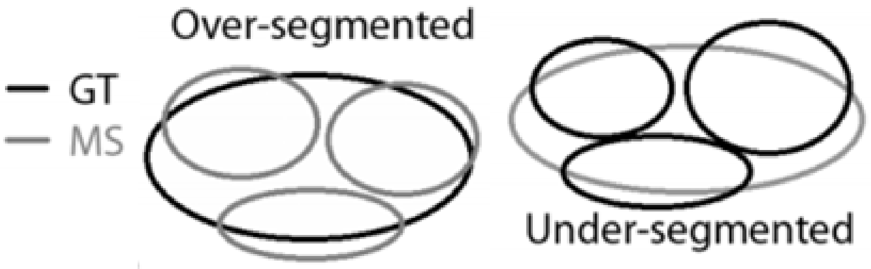


Figure 2.4: Over-segmentation occurs when one ground truth blob is segmented into multiple blobs. Under-segmentation occurs when multiple ground truth blobs are segmented as a single blob. (Figure modified from original (Mazhurin & Kharma 2012)**)**

Conversely a player may be under-segmented; their blob is larger than expected (Figure 2.4). For example, this may occur when a strong shadow is also segmented from the background. Under-segmentation may also result in two otherwise unattached players forming a single blob. This means that neither player has an accurate feature representation. It may be possible to resolve under-segmentation using an open morphological operation.

A special case of multiple players in a single blob occurs when a player occludes another. As demonstrated in Figure 2.5, occlusion occurs when an object O_1 passes between object O_2 and the camera. The pixels of both objects are declared to be of interest, yet the model cannot distinguish one from the other; therefore the objects appear as a single blob. At the foreground/background model level this is the correct segmentation. It is at the blob semantic level that this is incorrect. Therefore throughout this thesis, a blob similar to this will be treated as a correctly segmented multiple player blob.

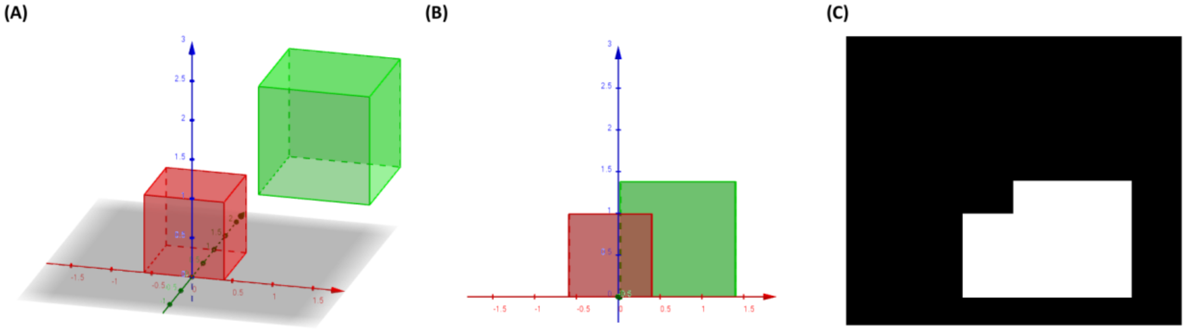


Figure 2.5: An example of occlusion. (A) The green cube overlaps the red cube in the *x* and *y*-axis, but not the *z*-axis. (B) When the cubes are projected onto the two-dimensional image plane the depth information is lost and the red cube occludes the green cube. (C) The cubes both deviate from the background model, so form one large segmented blob. (Produced using: (GeoGebra 2017)**)**

Generally there are three approaches to handling occlusions: (A) Treat the players as a single entity, with a single set of coordinates, (B) Predict coordinates for the occluded players, or (C) Attempt to determine each player's coordinates within the blob. It is clear that of these (A) is the simplest, yet it does result in the loss of individual player extractions for the period of occlusion. (B) retains individual player extractions, however the prediction accuracy decreases over time so this method is only suitable for short occlusions. Under full occlusion (C) becomes very difficult. As such in the case of small occlusions it is preferred to split the players, however for the case of nearly full occlusion, (B) is preferable for occlusions that persist for only a short period of time, while (A) is preferred for temporally longer occlusions.

(Seo et al. 1997) attempt to resolve occlusions using histogram back-projection (Swain & Ballard 1991). This method identifies how well the pixels or subset of the pixels in the blob meet the distribution of pixels in a histogram model. Due to the similarity between the histograms of players on the same team, (Seo et al. 1997) note that this method can only be used to resolve occlusions between players from opposing teams.

As with under-segmentation, in small cases of occlusion a morphological operation may split the players. (Figueroa et al. 2004) erode the blob and then conditionally thicken the new blobs back to the original blob contour. As illustrated in the third row of Figure 2.6, good results are observed when only a small part of a player such as a single leg is occluded.

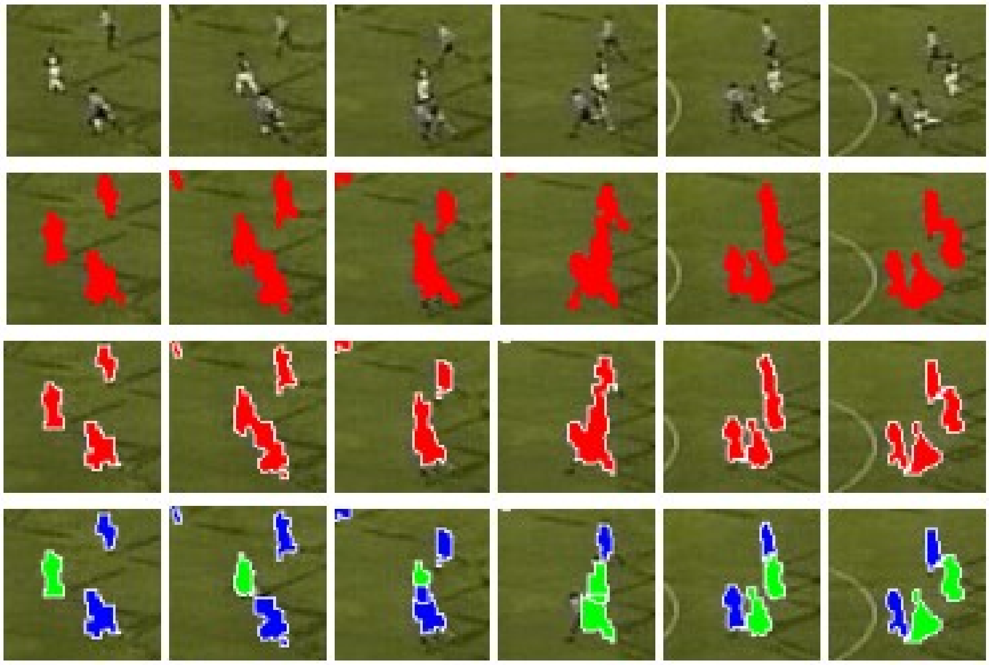


Figure 2.6: The blob splitting algorithm proposed in (Figueroa et al. 2004). Top Row: The original image. Second Row: The segmented players. Third Row: Blobs split using morphological operations. Bottom Row: Blobs split using the blob graph. (Modified from source: (Figueroa et al. 2004))

For cases with a larger area of occlusion, (Figueroa et al. 2004) use a blob graph to determine the number of players in a blob. The blob can then be split vertically or horizontal based upon the expected number of players (Bottom row Figure 2.6). Due to the blob splitting method, they only consider this method viable for blobs of two or three players. If a blob cannot be split, the contained players are treated as a single entity.

(von Hoyningen-Huene 2011) split multiple player blobs by finding the maxima of the convolution of the blob with a rectangular kernel the size of a typical player. This approach is only accurate in cases of limited overlap, for example it will fail when one player is directly between the camera and another player. Also the use of a kernel assumes the player is upright, an assumption that is not valid for all hockey poses (Carr et al. 2012).

In conclusion, this section investigated segmentation algorithms. Background subtraction provides a method for extracting foreground pixels when using a static camera. Many different background subtraction algorithms have been proposed in literature. An investigation should be performed to find the most accurate of these methods for the task of segmenting field hockey. This segmentation can lead to incorrect blobs due to under-segmentation, over-segmentation or noise. The next section will investigate methods to assess the accuracy of the segmentation. The subsequent section will consider ways to classify the contents of a blob.

Assessment of Segmentation

The accuracy of the segmentation can be assessed at the pixel level or the blob level. At the pixel level the algorithm is a binary classification. Those pixels that are classified as foreground are 1 and those that are background, 0. Assuming the ground truth foreground is known, there are many metrics that assess the accuracy of a classifier. The simplest of these is the misclassification rate given by Equation (3).

$$Misclassification\ rate = false\ positives + \ false\ negatives \qquad (3)$$

Here, a false positive is a pixel that has been classified as foreground when it is background and a false negative is the converse. Naively a lower misclassification rate indicates a more accurate classification. Yet the misclassification rate is independent of the prevalence of each class. If the dataset is biased towards one of the classes, the classifier may achieve lower misclassification rate simply by classifying all pixels as this class. In the player detection task the players are relatively small compared to the size of the pitch, therefore the dataset will be biased towards classifying pixels as background and the misclassification rate is unsuitable.

An alternative that does consider the prevalence of each class is the F-score given by Equation (4).

$$F\text{-}Score = 2\,\frac{precision * recall}{precision + recall} \tag{4}$$

Precision is the number of pixels that were correctly classified as foreground, true positives, over the total number of pixels that were classified as foreground, true positives plus false positives (Equation (5)).

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{5}$$

Recall is the number of pixels that were classified correctly as foreground over the total number of pixels that should have been classified as foreground, true positives plus false negatives (Equation (6)).

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{6}$$

Figure 2.7 visualises the calculation of precision and recall.

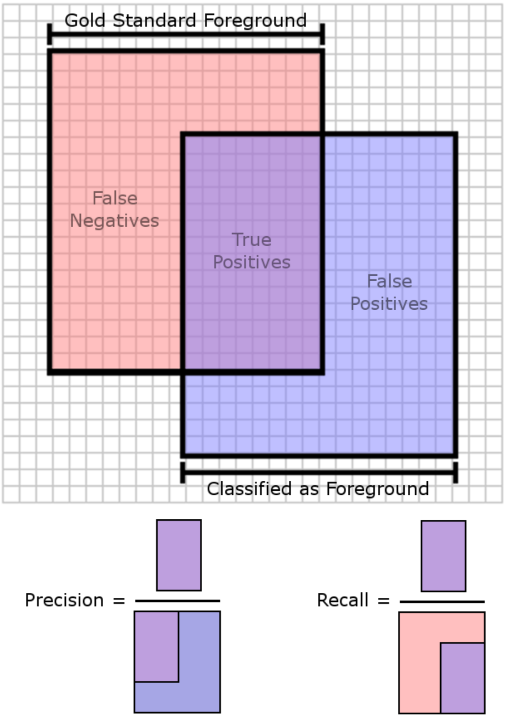


Figure 2.7: Formulation of the precision and recall given the ground truth and the classification by an algorithm.

A precision approaching 1 can be achieved by classifying the majority of pixels as background; however the recall would be approaching zero. Conversely a recall of 1 can be achieved by classifying all pixels as foreground but this would result in a low precision. The F-score, the harmonic mean of the precision and recall, combines them into a single accuracy metric. An F-score of 1 would mean a precision of 1, recall of 1 and perfect segmentation of the foreground.

(White & Shah 2007) used the F-score as the objective function to optimise the parameters of a foreground detection algorithm. This suggests that the F-score is suitable as a metric for segmentation accuracy; nonetheless there is disadvantage to using it. Not all pixels are semantically equal. For example, if pixels from the blobs periphery are misclassified there will only be a small effect on the representation of the blob. However, if a group of pixels that bisect the blob are misclassified as background the blob is split into two and the player is over-segmented (Figure 2.8). Similarly two blobs may be under-segmented by the misclassification of some background pixels as foreground. The F-score weights all pixels equally and thus does not consider the spatial context of the blobs, therefore some criteria is needed that considers accuracy at the blob semantic level.

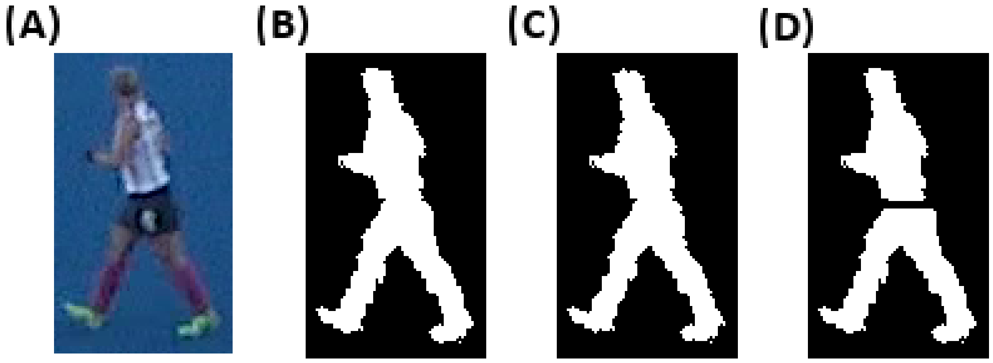


Figure 2.8: Not all pixels are semantically equal. Some pixels can be misclassified without changing the representation of the player, others cannot. (A) The RGB image of a player. (B) The segmented blob. (C) The segmented blob with 60 pixels of misclassification around the periphery of the blob. The shape of the blob remains similar to the ground truth. (D) The segmented blob with 60 pixels of misclassification bisecting the blob. The blob has now been split into two and is over-segmented.

The blob level accuracy can be assessed using the Image Segmentation Assessment Tool (ISAT) proposed in (Mazhurin & Kharma 2012). Figure 2.9 taken from (Mazhurin & Kharma 2012) illustrates the five mutually exclusive sets of blob segmentation.

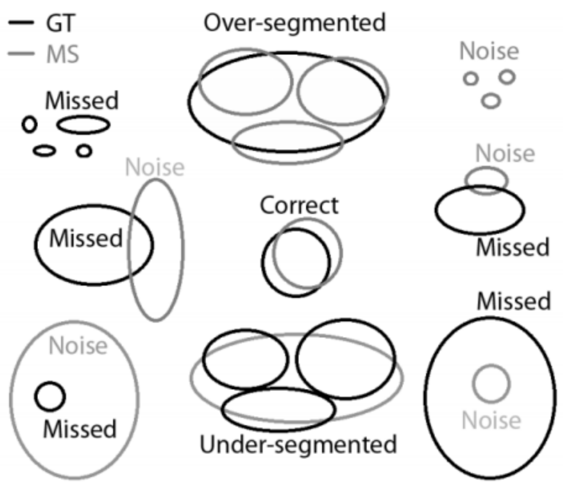


Figure 2.9: The possible resultant sets of blob segmentation. Here GT, ground truth, is the expected blob configuration and MS, machine segmentation, is the output of the blob finding algorithm. Blobs can be: Correct, Missed, Noise, Over-segmented or Under-segmented. Each set is mutually exclusive. (Source: (Mazhurin & Kharma 2012)**)**

Here GT, ground truth, is the set of expected blob configuration. When applied to player detection this is the player's true blob. MS, machine segmentation, is the blob configuration that has resulted from the algorithm. The five sets are:

- Correct – Greater than T% of the area of one ground truth blob overlaps with greater than T% of the area of one machine segmented blob.
- Over-segmented – Greater than T% of the area of one ground truth blob overlaps with more than one machine segmented blobs. AND Greater than T% of the area of each machine segmented blob overlap with the ground truth blob.
- Under-segmented – Greater than T% of the area of one machine segmented blob overlaps with more than one ground truth blobs. AND Greater than T% of the area of each ground truth blob overlaps with the machine segmented blob.

- Missed – Any ground truth blob that does not meet any of the above three criteria.
- Noise – Any machine segmented blob that does not meet any of the above criteria.

A Tgreater than 50% ensures mutual exclusivity of the sets.

The assessment of the segmentation is given as four percentages and an absolute value. The percentage of ground truth blobs that are: correct, missed, under-segmented and over-segmented, plus the number of machine segmented blobs that are noise.

Given the findings of this section the segmentation will be assessed at a pixel level using the F-score and at the blob level using the ISAT classes.

Classifying the Blob

The second step in the player detection algorithm is to determine the semantics of the blob; that is to classify if it is or is not a player. If it is presumed that only players are on the pitch during the match, it can be naively assumed that each segmented blob is an individual player. Due to errors in the segmentation process this assumption is not valid. Errors may occur due to the segmentation of non-player blobs, the inclusion of shadows in the player segmentation or the incomplete segmentation of players.

As noted earlier, the extraction of a reliable player feature representation requires accurate blob segmentation. If the semantics of the blob can be determined not only can non-player blobs be eliminated but the player feature representation can be individually tailored to handle other segmentation errors. Further to this, understanding the semantics of the blob can be exploited to increase the accuracy of the blob merging necessary due to over-segmentation. Equally, as discussed previously, a blob may include multiple players. If it can be determined that a blob contains multiple players, then an algorithm can be applied to handle this.

The classification of blobs as hockey players is a subset of object classification. Until 2012 the state of the art approach to object classification was to: (1) extract a hand-designed feature representation from the image, (2) use a training set to learn the mapping of these feature representations to classes.

The first real-time face detection algorithm, as proposed in (Viola & Jones 2001; Viola & Jones 2004) and then applied to pedestrian detection in (Viola et al. 2005) used a boosted cascade of Haar features. Haar features are rectangular filters, chosen to reflect the expected differences in the intensity of an image given an object. For example, as illustrated in Figure 2.10, it is expected that a nose will result in a band of high intensity pixels between two bands of low intensity pixels in the upper half of the test image. A high output from this filter would be a good predictor of a nose and as such a face.

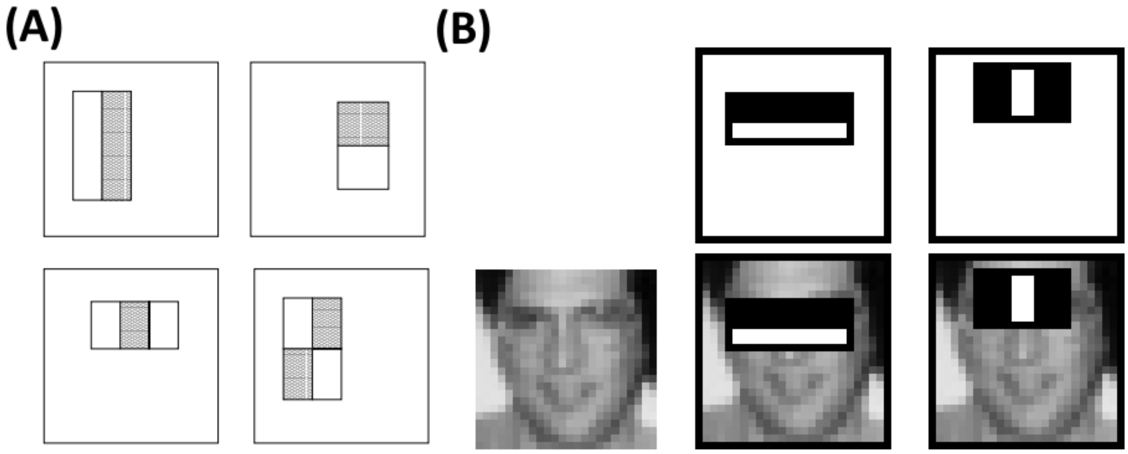


Figure 2.10: The Haar features used for face detection in (Viola & Jones 2001)**. (A) Four of the possible Haar features . (B) The first and second features selected by Adaboost. These features suggest that to identify a face in the image: 1) the eyes should be darker than the upper cheeks, and 2) the area between the eyes should be lighter than the eyes themselves. (Modified from source:** (Viola & Jones 2001)**)**

Haar features are constrained in ratio but not in size or space; therefore in the original formulation of (Viola & Jones 2001) there are approximately 160,000 Haar features in a 24 pixel x 24 pixel image. Many of these Haar features will predict a face no better than chance. The Hughes Effect (Hughes 1968) suggests that including all the Haar features will reduce the predictive power of a linear classifier. Hence Adaboost (Freund & Schapire 1997) is used to produce an ensemble of the N most predictive Haar features. The algorithm is completed by forming a cascade of ensembles with an increasing number of Haar features. Given an image, most of the sub-windows are not faces; the cascade eliminates the need to calculate all Haar features for all sub-windows, thereby decreasing computational time.

(Liu et al. 2009) apply a boosted cascade of Haar features to football video. They first use dominant colour segmentation to find candidate regions in video frames. They then apply the cascade to these candidates. They achieve an F-score of 90.39% for the correct classification of blobs as players. The authors do not comment on the algorithms performance when the candidate regions contain more than one player.

An alternative to Haar features is the Histogram of Orientated Gradients (HOG) (Dalal & Triggs 2005). This feature descriptor counts the occurrences of gradient orientation in sub-windows of the image Figure 2.11. The feature descriptor is used to train a Support Vector Machine (SVM) (Cortes & Vapnik 1995). A SVM learns the hyperplane that maximally separates the data into the correct class. The learnt SVM can then be used to classify a new datum.

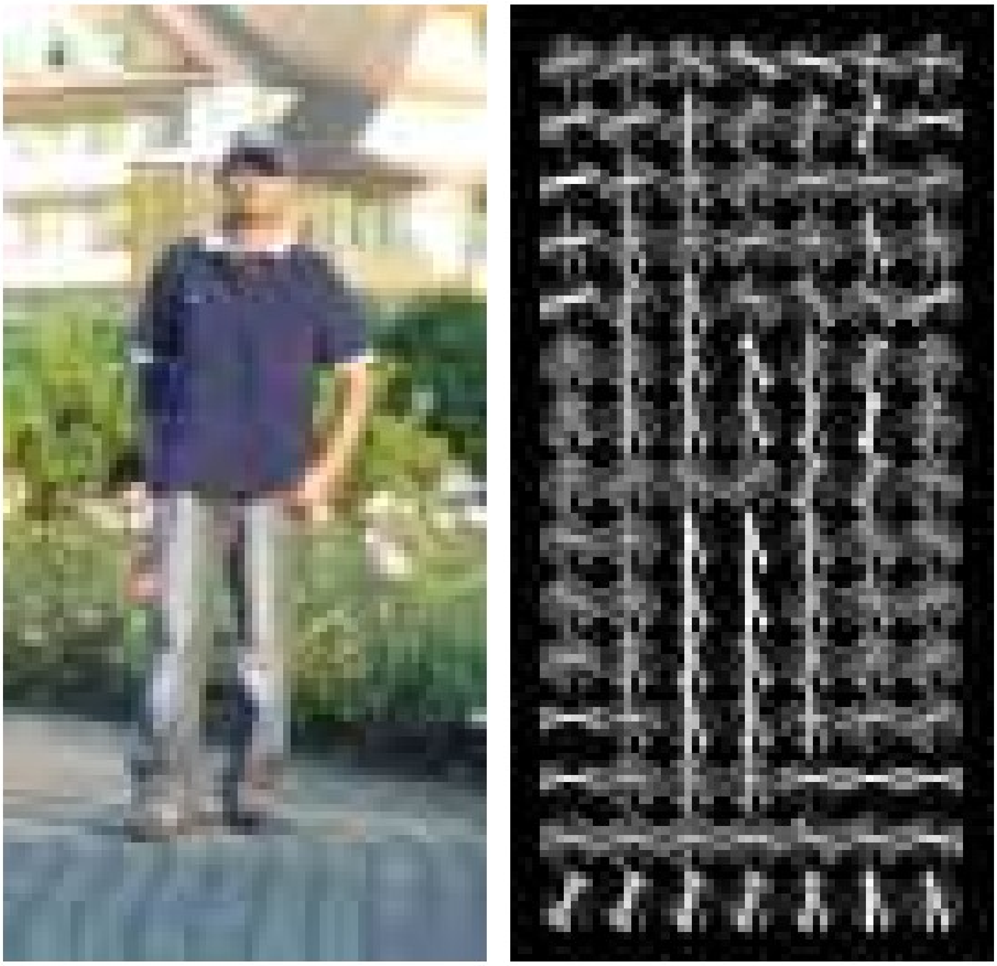


Figure 2.11: Histogram of Oriented Gradients (HOG). Left: The original image. Right: The image's HOG descriptor. (Modified from source: (Dalal & Triggs 2005)**)**

(Lu et al. 2013) apply the Deformable Part Model (DPM) of (Felzenszwalb et al. 2008) to basketball footage. They hypothesise that as sports players are not rigidly upright the poses are more varied. By learning individual HOG models for six body parts and the orientation between those body parts, the algorithm can handle more variation in pose. On basketball scenes they report a 69% precision and 73% recall.

The DPM is unsuited to the classification of players across an entire field hockey pitch. The players are expected to be relatively small in the scene; as such the lack of detail makes it difficult to determine the parts of the model. This will have a negative impact on the accuracy of the classification.

The Imagenet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015) classification task is an annual competition to find the best visual classifier. The task requires an algorithm to classify images into 1000 different classes. The classes cover a broad range of objects. Some classes are generic objects such as desk or microwave, whereas other classes are visually similar, such as breeds of dog. An image is deemed correctly classified if the true class is in the top 5 predicted classes. The error of an algorithm is then the percentage of incorrectly classified images.

The winner of the ILSVRC 2011 classification task (Perronnin et al. 2010) used the Fisher Vector (Jaakkola & Haussler 1999) encoding to achieve an error of 26%. The winner of the ILSVRC 2012 classification task was AlexNet (Krizhevsky et al. 2012) with an error of 16%. This large increase in accuracy is attributed to the use of a Convolutional Neural Network (CNN). A CNN is a type of feed-forward neural network (NN), or multi-layer perceptron.

A perceptron (Rosenblatt 1958) is a binary linear classifier modelled on a neuron in the brain. It maps an input feature vector x to a single value output value $f(x)$ (Equation (7)).

$$f(x) = \begin{cases} 1 & if\ w \cdot x + b > 0 \\ 0 & otherwise \end{cases} \tag{7}$$

Where w is a vector of weights, $w \cdot x$ is the dot product as given in Equation (8) and b is the bias.

$$w \cdot x = \sum_{i=1}^{m} w_i x_i \tag{8}$$

The weights and bias are the parameters of the model and are learnt over a training set using the perceptron learning rule.

A neural network extends the brain analogy. The brain is a complex network, where the outputs of neurons become the input for other neurons. A neural network attempts to replicate this by grouping perceptrons into layers, with a perceptron in layer $L+1$ taking as input the outputs of layer L (Figure 2.12). A non-linear activation function must be applied to the output of each perceptron; otherwise the neural network could be represented by a single perceptron. The weights of the perceptrons in the network are learnt over a training set by minimising an objective function using gradient descent. Back-propagation (Rumelhart et al. 1986) is used to determine the gradient of each weight with respect to the objective function.

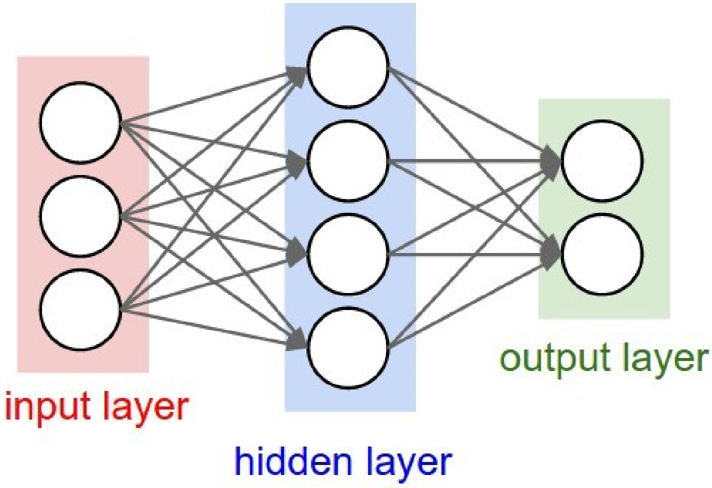


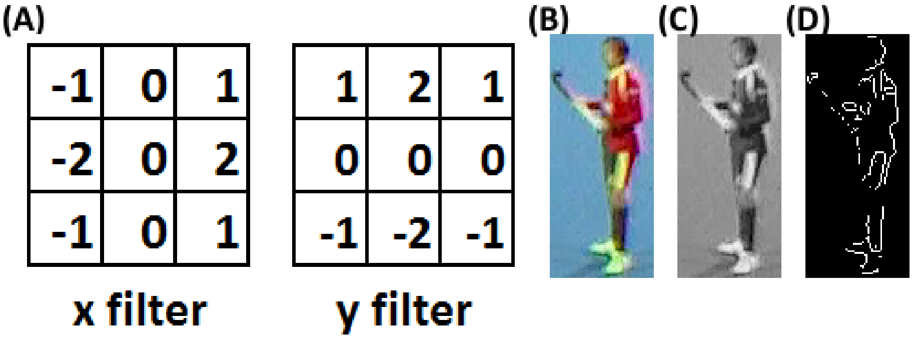
Figure 2.12: A two layer fully connected neural network. (Source: (Karpathy 2016))

The universal approximation theorem (Cybenko 1989) states that a neural network with a single hidden layer containing a finite number of perceptrons can approximate any continuous function. Therefore, it can be expected with enough data any classification problem can be learnt. However, fully connected neural networks do not perform well on image classification tasks. Consider an image from the Imagenet dataset. This image is 224 pixels x 224 pixels x 3 colour channels, giving a total feature vector of 150,528 elements. In a fully connected neural network the number of parameters for each perceptron is the $number\ of\ elements + 1$, i.e. 150,529 parameters. Therefore the total number of weights in the model is a linear function of the number of perceptrons. While adding perceptrons, and as a result parameters, increases the complexity of the function that can be learnt, it also increases the possibility of over-fitting to the training set reducing the models generalisability. Overfitting occurs when the model learns the noisy distribution described by the training set, rather than the task's underlying distribution. The model is effectively memorising the training data. It results in good classification performance on the training set but poor generalisation to a test set. This may be resolved by reducing the number of parameters or by increasing the number of training examples, which may be impractical. Additionally, as the model learns to weight specific pixels, a per pixel transformation, such as a spatial translation, can lead to poor classification results.

An alternative, that still exploits the power of a neural network while mitigating the limitations, is to hand craft a feature which is then passed to a neural network for classification. Yet the developed feature may not be optimal. A convolutional neural network (CNN) overcomes this by classifying directly on the normalised pixel values.

CNNs again borrow from the theory of the brain, this time a simplified model of the visual system. The receptive field of a neuron is the set of inputs that stimulate the firing of the neuron. At the lowest layer of the visual system this is a particular region of the retina. The neuron is fired if a specific simple structure is recognised in that receptive field. Here the structure may be a texture or colour pattern. The receptive field of a neuron in a subsequent layer is the outputs of a group of neurons in a previous layer. This allows the identification of a more complex structure constructed from the structure at the previous layer. As the number of layers increases, more complex structures can be represented.

Similarly a convolution matrix is a mask that identifies a specific structure in an image. A region of the image can then be convolved with this convolution matrix to determine if the region contains the structure. For example, the two Sobel convolution matrices presented in (Sobel & Feldman 1973) identify pixels that are likely to be edges (Figure 2.13).



-1	0	1
-2	0	2
-1	0	1

x filter

1	2	1
0	0	0
-1	-2	-1

y filter

Figure 2.13: The Sobel filter. (A) The convolution matrices that make up the Sobel filter. (B) The original image. (C) The grayscale image. (D) The output of the Sobel filter.

A CNN is a feed forward neural network comprised of layers of convolution matrices (LeCun et al. 1998). At runtime each of the convolution matrices in a layer are applied to the input image to return a new space that has as many channels as the number of matrices in the layer. This new space is the input for the succeeding layer.

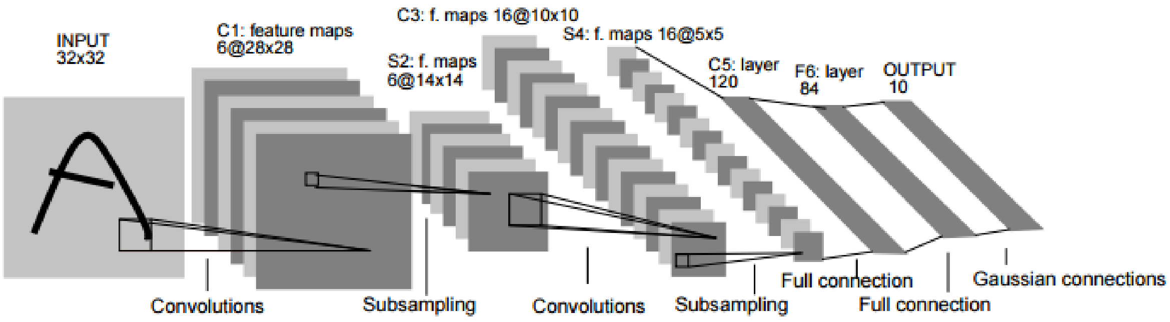


Figure 2.14: The character recognition Convolutional Neural Network architecture proposed by (LeCun et al. 1998) and commonly known as LeNet.

The Sobel filter is an example of a convolution matrix that may be in the first layer. It is more difficult to visualise convolution matrices at higher layers, however it can be abstracted that higher layers find more complex structures. For example the classification of a car may have the following abstraction:

- The convolution matrices in low layers will identify simple structures such as edges.

- The middle layers combine these structures to identify shapes such as circles.
- The upper layers identify that these simple shapes form parts of the car such as the wheel.
- The highest layers classify the contents of the image based upon the configuration of the car parts.

As with neural networks, the weights of a CNN are learnt over a training set using gradient descent.

Whereas in the brain each neuron has a specific receptive field, in a CNN the convolution matrices are applied across the whole image. This is by design, for two reasons:

1. It means the convolution matrices are space invariant. If a structure is important in one part of the image, it is likely to be important throughout the image.
2. It reduces the number of weights to be learnt and thereby reduces overfitting. If convolutions were location specific, it can be supposed that there would be more of them, therefore increasing the number of weights.

As stated earlier, CNNs have won all the ILSVRC classification tasks since 2012 (Krizhevsky et al. 2012; Zeiler & Fergus 2014; Szegedy et al. 2015; He et al. 2016). The winning CNNs have progressively got deeper; the number of layers has increased. While the universal approximation theorem states that any function can be represented by a neural network with a single hidden layer, (Simonyan & Zisserman 2015) showed that classification accuracy using CNNs is dependent upon the depth of the network. (He et al. 2016) the 2015 winner of ILSVRC used 152 layers to achieve a top 5 error rate of 3.6%. This classification rate is better than the 5.1% achieved by human performance (Russakovsky et al. 2015), and their analysis has suggested that contextual scene understanding is needed to further improve performance.

In general, increasing the depth of a CNN increases the number of weights in the network. This increase in weights increases the possible complexity of the network and as such the likelihood of overfitting the network to the training data. As stated earlier, overfitting to the training set can be reduced by increasing the number of training examples or by reducing the complexity of the network. One of the reasons for the rapid improvement in the top 5 error rate has been the availability of vast numbers of training examples on the internet. For example the ImageNet training set consists of 1.2 million images. Another reason is the more efficient use of the parameters. Despite being deeper, (Szegedy et al. 2015) the 2014 winner of ILSVRC contains fewer weights than the runner up (Simonyan & Zisserman 2015). It achieves this by using smaller convolution matrices.

Even with the reduction in parameters, training an entire CNN with a small dataset will lead to poor generalisation. To overcome this, a CNN can be trained via transfer learning. Transfer learning takes a network that has been trained on a much larger dataset, ImageNet for example, and then fine-tunes the parameters using a smaller domain specific dataset. (Yosinski et al. 2014) suggest that transfer learning achieves better results than training from scratch because the convolution matrices learnt at low layers are very generic. Good low level structures remain good structures independent of the image contents. It is only the higher layers that identify content specific structures that need to be fine-tuned. As such they perform an investigation into how much of the CNN should be retrained. They found the best classification accuracy was achieved when the CNN is fine-tuned from the middle layers.

Given the recent performance of CNNs on classification tasks, this seems a promising direction for the handling of incorrect segmentation. Therefore an investigation should be performed to determine if a CNN can be used to classify correct player blobs.

2.4.2 Player Coordinate Extraction

The spatial location of a player can be represented by either:

1. The point in image coordinates (u, v) extracted directly from the image.
2. The image point transformed into world coordinates (X, Y, Z), where it is assumed a player is on the ground plane and subsequently $Z = 0$. This transformation requires the calibration of the camera which is discussed in more detail in Section 2.5.

Commonly the image point is found as the projection of the blob's estimated centre of mass onto the base of its bounding box. The transformation to world coordinates assumes the player is in contact with the pitch, the base of the bounding box is an estimate of this point. However Figure 2.15 illustrates that this may not be accurate if the player's motion is not perpendicular to the camera or the blob has a long thin protrusion, such as a hockey stick.

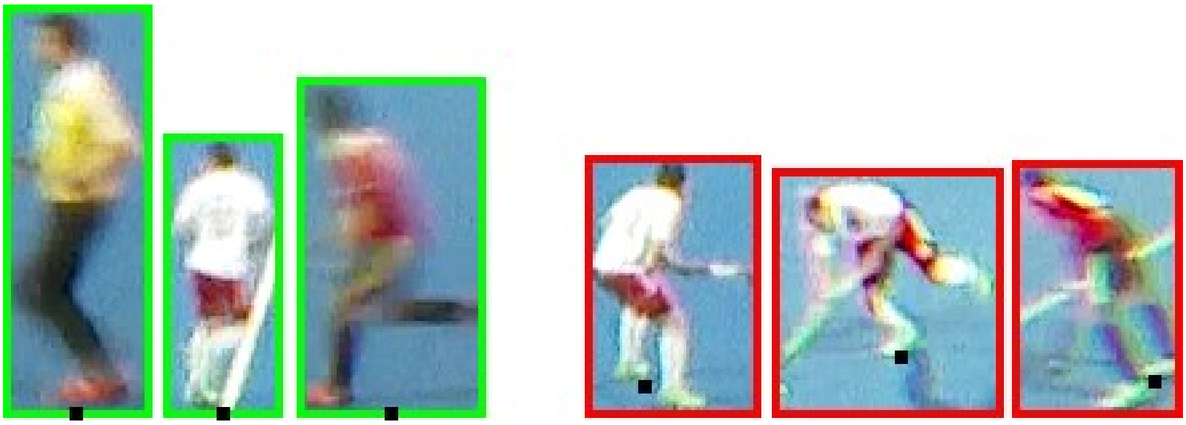


Figure 2.15: The automatic image coordinate of a blob is estimated as the midpoint of the base of its bounding box. For the images with a green bounding box this gives an accurate image coordinate. However for the images with a red bounding box the estimate is poor. The black dot illustrates the manual identified image coordinate.

Alternatively a player's coordinates can be calculated using a Probability Occupancy Map (POM) (Fleuret et al. 2008). POM discretises the world space into a grid. The probability of a player at a grid location is calculated based upon the extracted foreground and expected foreground given a player at that location. The location of a player is then the grid location with the highest probability. (Carr et al. 2012) adapted POM to detect the location of field hockey players. They achieved an approximate precision of 55% and recall of 50% at a tolerance of 1 m. However their footage was captured using a multi camera system, with each camera focussed on a subsection of the pitch. This allowed for very well segmented players, something that cannot be assumed in this work. Also as POM requires the discretisation of the world, the players are bound to certain pitch locations. The number of pitch locations can be increased to improve the player location resolution; however this increase also increases the computation requirements. For these reasons POM is unsuitable in this work.

2.4.3 Summary

This section investigated the procedure for the extraction of players from a video. It found that the extraction of players is a two-step process. First the player blob is extracted from the frame and then their coordinate is extracted from the blob. The player blob extraction process can be decomposed further to: (1) classify the pixels of interest, (2) group these pixels and (3) classify their content.

Pixels of interest can be found using the method of background subtraction. This method develops a background and classifies pixels as foreground if they deviate from this model. The pixels are then formed into blobs using 8-connectivity. Many different background subtraction methods are proposed in literature. This leads to the objective:

Determine an accurate method to segment player blobs from an image.

The segmentation of the blobs in the frame can result in well segmented players but also noise and poorly segmented players. The extraction of a players coordinates requires accurately extracted players; therefore a method is needed to classify if a player is well segmented. Convolutional neural networks provide state of the art performance at classification tasks. This leads to the objective:

Train a Convolutional Neural Network to classify the contents of a blob.

2.5 Reconstruct World Points

A camera is a device that captures an image of the objects in a scene. As displayed in Figure 2.16, it performs a three-dimensional projection of points in the world coordinate system into the two-dimensional image plane coordinate system. Camera calibration is the estimation of a model for this projection. Given this model it is possible to: (A) project world coordinates onto the image plane, and (B) reconstruct points from the image plane into world coordinates. The former of these is the basis for augmented reality (Azuma 1997; Krevelen & Poelman 2010). Of more interest in this thesis, the latter allows player coordinates in the image to be reconstructed into world coordinates (Dunn et al. 2012).

Assuming a single camera, it is not possible to reconstruct a complete three-dimensional model of the scene. The three-dimensional projection onto the image plane discards the point's depth information and it cannot be recovered. However, it is possible to determine the ray projected from the centre of projection that passes through a point. If it is assumed that all points lie on a plane, then the object's planar coordinate is given as the point of intersection of the projected ray and this plane. This plane is known as the calibrated plane.

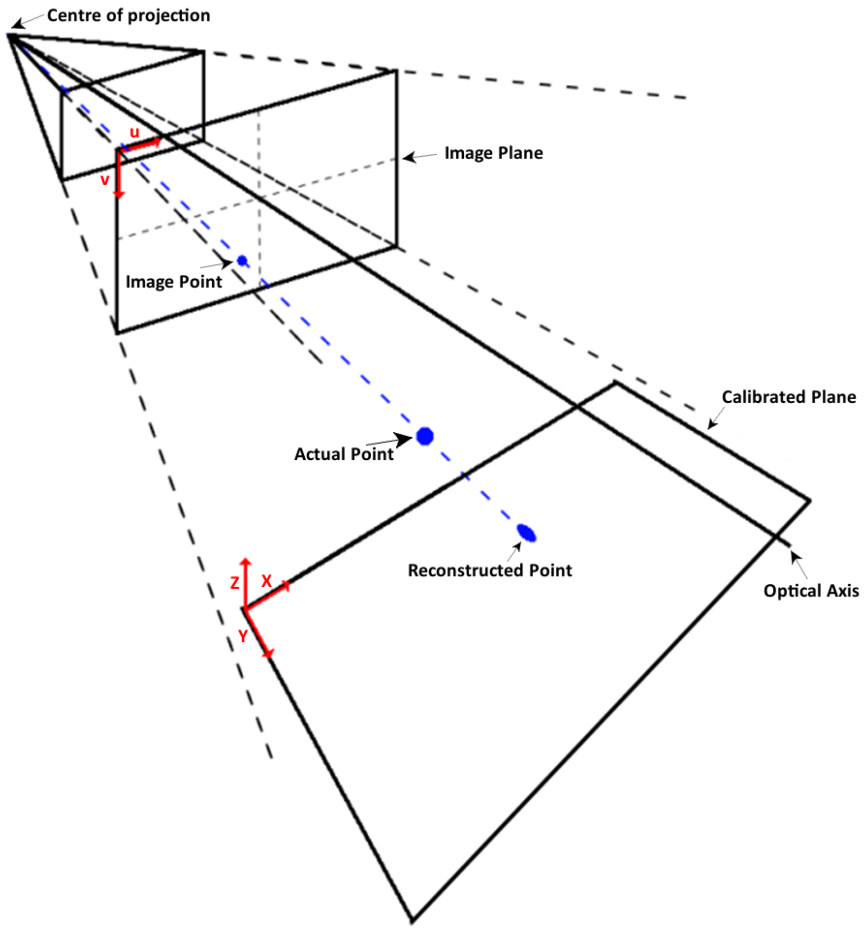


Figure 2.16: The three-dimensional projection of a world point onto the image plane. Given a camera model, two operations can be performed: (A) World Points can be projected onto the image plane to augment reality. (B) Image points can be reconstructed into planar world coordinates. The three-dimensional projection discards depth information therefore only the direction of the ray that passes through the centre of projection and the point can be determined, here the blue dashed line. If it is assumed all points are on a calibrated plane, the world point is found as the intersection of the ray and this calibrated plane.

2.5.1 Camera Model

A point in the homogeneous world coordinate system is projected into homogeneous image plane coordinates by Equation (9).

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P_{projection}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{9}$$

Where w is a scaling factor. $P_{projection}$ is the camera matrix and is given by Equation (10).

$$P_{projection} = K[R|T] \tag{10}$$

Here K encompasses the intrinsic parameters of the camera. These are the parameters that transform from camera coordinates to pixel coordinates. K is given as in Equation (11).

$$K = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

f_x and f_y are the focal lengths, the distance in units of horizontal and vertical pixels between the lens and centre of projection, the point where all parallel rays converge. γ is the aspect ratio, the ratio of pixel width to pixel height. u_0 and v_0 are the principal point; the image coordinates of the point where the optical axis, a straight line passing from the centre of projection through the geometrical centre of lens, intersects the image plane.

The extrinsic parameters, R and T, comprise the rotation and translation necessary to perform the rigid transform from world coordinates to camera coordinates as in Equation (12).

$$R|T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Equation (9) describes the perfect perspective camera; however deviations from this model occur due to the optical system and sensor misalignment (Clarke & Fryer 1998). These deviations cause two types of image distortion, tangential and radial. The intrinsic model must be extended to correct for these distortions.

Tangential distortion is the result of: (1) thin prism distortion, imperfections in the lens manufacturing process (Weng et al. 1992), and (2) misalignment of a camera's lens and image plane as in Figure 2.17 (Bradski & Kaehler 2008). (Mallon & Whelan 2004) suggest tangential distortion is not readily observable in the presence of radial distortion and can be somewhat reduced by principal point estimation.

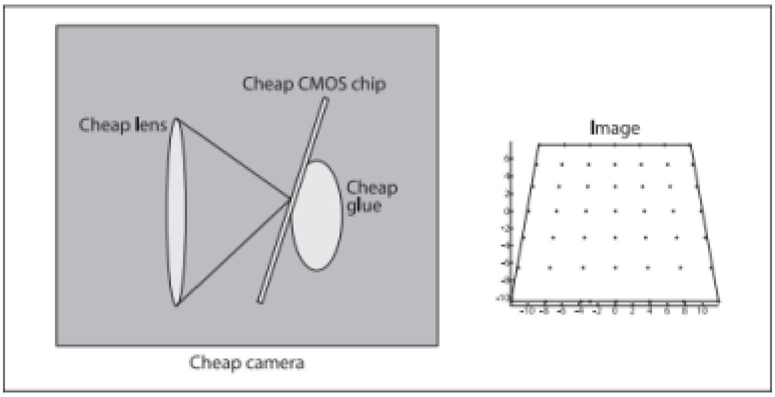


Figure 2.17: Tangential distortion as a result of misalignment of a camera's lens and image plane. (Source: (Bradski & Kaehler 2008)**)**

Radial distortion is a deviation from a rectilinear projection. It causes straight lines in a scene to appear curved in an image. A point is displaced by a non-linear function, along the radial axis from the principal point (Hughes et al. 2008).

(Brown 1971) extended the camera model to account for radial and tangential distortion. First the world coordinates are transformed to camera coordinates using the extrinsic parameters (Equation (13)).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = [R|T] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{13}$$

Then the normalised image coordinates are given by Equation (14).

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} \tag{14}$$

Subsequently distorted normalised image points are given by Equation (15).

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = distortion_{radial} * \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \ distortion_{tangential} \tag{15}$$

Here radial distortion is modelled as the first three even terms of a Taylor series of the radial distance (Equation (16))

$$distortion_{radial} = 1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6 \tag{16}$$

Where $kc(1)$, $kc(2)$ and $kc(5)$ are the radial distortion coefficients r^2 is given by Equation (17).

$$r^2 = \ x^2 + y^2 \tag{17}$$

The tangential distortion is modelled as 2 coefficients, $kc(3)$ and $kc(4)$, and given by Equation (18).

$$distortion_{tangential} \tag{18}$$

$$= \begin{bmatrix} 2kc(3)x_n y_n + kc(4)(r^2 + 2{x_n}^2) \\ kc(3)(r^2 + 2{y_n}^2) + 2kc(4)x_n y_n \end{bmatrix}$$

Finally the coordinates are converted to the pixel coordinate system by Equation (19).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{19}$$

While all lenses exhibit some radial distortion, a wide-angle lens is specifically designed to obey some other projection model (Kannala & Brandt 2006), which has so much barrel distortion that an entire hemisphere is imaged as a finite circle (Kingslake 1989). This wide angle of view permits the capture of large objects using a single camera, however, (Shah & Aggarwal 1996) showed that an image captured with a wide-angle lens still exhibits distortion when corrected using the radial distortion model in Equation (16). Therefore a different distortion model is required for wide-angle lenses.

(Kannala & Brandt 2006) modelled the radial distortion as a function of the angle between the principal axis and the incoming ray, rather than the radial distance. The first four terms of this Taylor series are used to model the radial distortion and tangential distortion is omitted. Their results suggest that the Root Mean Square Error (RMSE) of projection is comparable to state-of-the-art methods for both conventional and wide-angle lenses. As such this camera model will be used throughout the thesis.

2.5.2 Assessment of Calibration Accuracy

It is vital that a camera model is able to reconstruct a scene with sufficient accuracy. Without sufficient accuracy a user cannot be confident that the estimated coordinate is the objects true coordinate. Therefore a measurement of the accuracy of the model is necessary to determine if it is sufficient. (Salvi et al. 2002) reviews the common measures used to assess the accuracy of a camera model. These measures can either be in the image coordinate system or the world coordinate system. This literature review will consider two of the most popular.

The mean projection error, or in (Salvi et al. 2002) the ‘Accuracy of distorted image coordinates’, is computed by:

1. Capture an image of a set of known world points.
2. Extract the image points for each of the world known points.
3. Project the known world points onto the image plane using the projection model.

4. The mean projection error, $\varepsilon_{projection}$, is then the mean distance in pixels between the extracted test coordinates and their corresponding projected test coordinates (Equation (20)).

$$\varepsilon_{projection} = \frac{1}{N}\sum_{i=1}^{N} d(x_i, F_{projection}X_i)^2 \qquad (20)$$

where N is the number of known points, $d(A, B)$ is the Euclidean distance between two points, x_i is the image point, $F_{projection}$ is the function that transforms from world coordinates to image coordinates and X_i is the world point.

The mean projection error, $\varepsilon_{projection}$, is important when augmenting reality, for example, if an object is to be overlaid with a graphic, any error in the projection will be perceived as misalignment in the augmented world. However, this thesis is concerned with the opposite transformation, from image coordinates to world coordinates; therefore it is more appropriate to use a metric measurement.

The mean reconstruction error, or in (Salvi et al. 2002) the 'Radius of Ambiguity in the calibrating plane', is a metric measurement. It is computed as follows:

- Capture an image of a set of known world points.
- Extract the image points for each of the world known points.
- Reconstruct the image coordinates of the test points onto the calibrated plane.
- The mean reconstruction error, $\varepsilon_{reconstruction}$, is then the mean distance in metres between the known world points and their corresponding reconstructed test points (Equation (21))

$$\varepsilon_{reconstruction} = \frac{1}{N}\sum_{i=1}^{N} d(X_i, F_{reconstruction}x_i)^2 \tag{21}$$

Where $F_{reconstruction}$ is the function that transforms from image coordinates to world coordinates.

Despite the mean reconstruction error being a more appropriate assessment criterion, there is more uncertainty in the mean reconstruction error than in the mean projection error. This is due to potential errors in the known world points object, $\varepsilon_{world\ points}$, and the extraction of the image points, $\varepsilon_{image\ extraction}$. Equation (20) can be expanded to include $\varepsilon_{world\ points}$ and $\varepsilon_{image\ extraction}$ (Equation (22)).

$$\varepsilon_{projection} = \frac{1}{N}\sum_{i=1}^{N}(d(x_i, F_{projection}(X_i + \varepsilon_{world\ points}))^2 \\ + \varepsilon_{image\ extraction}) \tag{22}$$

As the known world points object can be accurately machined, it can be assumed they contain no error and this simplifies to Equation (23).

$$\varepsilon_{projection} = \frac{1}{N}\sum_{i=1}^{N} d(x_i, F_{projection}X_i)^2 \\ + \varepsilon_{image\ extraction} \tag{23}$$

Equivalently Equation (21) expands to give Equation (24).

$$\varepsilon_{reconstruction} = \frac{1}{N}\sum_{i=1}^{N}(d(X_i, F_{reconstruction}(x_i \\ + \varepsilon_{image\ extraction}))^2 + \varepsilon_{world\ points}) \tag{24}$$

Which simplifies to Equation (25).

$$\varepsilon_{reconstruction} = \frac{1}{N}\sum_{i=1}^{N} d(X_i, F_{reconstruction}(x_i + \varepsilon_{image\ extraction}))^2 \tag{25}$$

As illustrated in Equation (23), $\varepsilon_{image\ extraction}$ is independent of $F_{projection}$ and as such it is a constant scalar. Yet when calculating $\varepsilon_{reconstruction}$, Equation (25), $\varepsilon_{image\ extraction}$ is propagated through the reconstruction function. As such its value is dependent upon the reconstruction function, which adds uncertainty to $\varepsilon_{reconstruction}$. Consequently it is important to minimise $\varepsilon_{image\ extraction}$ so it can be assumed all error is due to $F_{reconstruction}$.

(Hudson 2015) identified the known world points using the intersections of a black and white checkerboard (Figure 2.18). Checkerboard intersections demark a set of dimensionless points which can be precisely extracted from an image with sub-pixel accuracy using a Harris corner detector (Harris & Stephens 1988).

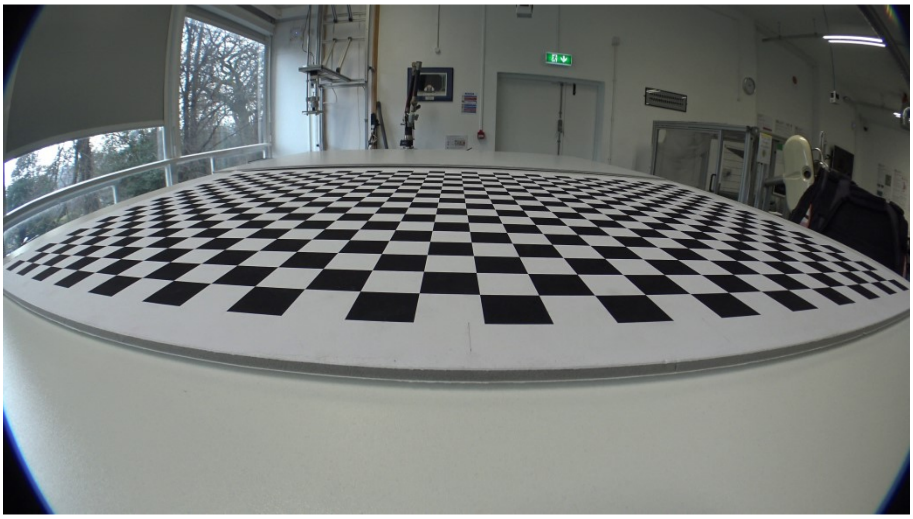


Figure 2.18: Checkerboard intersections identifying known world points in the image.

Alternatively (Mateos & Tsai 2000; Kannala & Brandt 2006) denote the known world points as the centres of a grid of circles (Figure 2.19). The known image points can be inferred by finding the centre of the circles in the image. However (Mateos & Tsai 2000) note that the centre of the ellipse that results from projecting a circle is not the centre of the original circle. This is due to:

1. The projective transformation means the planar resolution is not consistent across the ellipse. Pixels further from the camera account for a larger area than those closer to the camera. Therefore the centre of the ellipse will denote a point that is shifted from the true centre towards the camera. This effect is particularly apparent when the camera's angle of incidence is very high giving a large range of planar resolution.
2. The amount of distortion is not consistent across the ellipse. Pixels further from the principal point will exhibit more distortion. Therefore the centre of the ellipse will denote a point that is shifted from the true centre towards the principal point.

Both (Mateos & Tsai 2000; Kannala & Brandt 2006) correct for issue 1. (Mateos & Tsai 2000) use the common tangents of the grid of circles to identify known points on the periphery of each of the circles. Given these it is simple to identify the centres. (Kannala & Brandt 2006) propose a numerical method to find the centres. Both methods also consider solutions to issue 2 in cases where the intrinsic parameters are known.

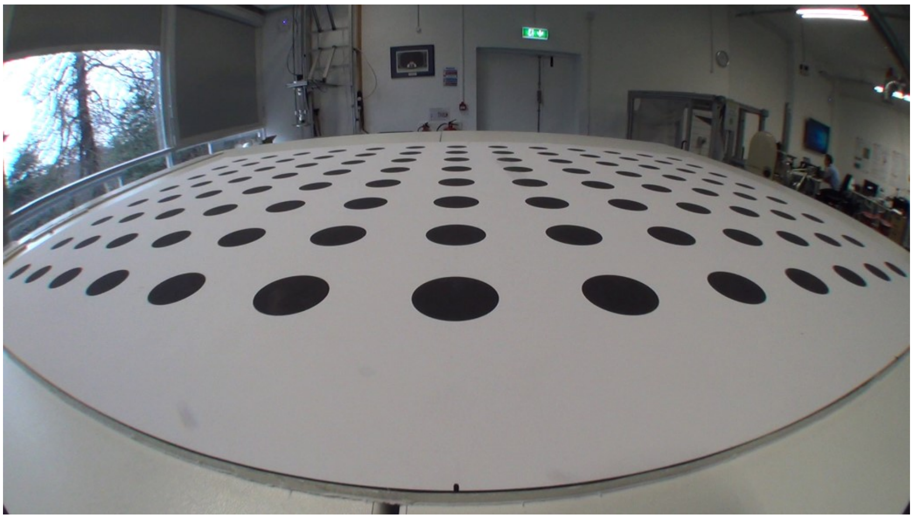


Figure 2.19: Circle centres identifying the known world points in the image

Checkerboard intersections and the centres of a set of circles both provide methods to extract known world coordinates on a plane. Due to the camera's wide angle of view and relatively high angle of incidence to the plane it is unknown if either of the two methods are suitable for the expected pose. Therefore an investigation should be performed to discover which of these methods is most suitable to demark known world points from the expected camera pose.

2.5.3 Planar Camera Calibration

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of the camera. Camera calibration methods are proposed in (Heikkila & Silven 1997; Sturm & Maybank 1999; Zhang 2000). The method of (Zhang 2000), which has been widely adopted and has implementations in C++ (Bradski & Kaehler 2008) and Matlab (Bouguet 2015), uses a planar checkerboard so is an example of planar camera calibration.

The first step of the planar camera calibration procedure is to estimate the intrinsic parameters. For each of N different images of a planar calibration object, W known points are extracted to give a collection of N point sets. Each of these point sets has a unique optimal set of extrinsic parameters but the optimal intrinsic parameters are global across the collection. The optimal intrinsic parameters, K, as well as R_i and T_i for i=1, 2… N, can be found by minimising the projection error, $\varepsilon_{projection}$ (Equation (26))

$$\varepsilon_{projection} = \sum_{i=1}^{N}\sum_{j=1}^{W} d(x_{(i,j)}, K[R_i|T_i]X_{(i,j)})^2 \tag{26}$$

Where $d(A, B)$ is some distance measure between A and B, in the case of (Zhang 2000) it is the Euclidean distance. $x_{(i,j)}$ is the jth point extracted from the ith image and $X_{(i,j)}$ is the corresponding world coordinate of the point. For brevity the distortion correction has been omitted from the equation but these parameters are also optimized.

Subsequently the extrinsic parameters for a new calibrated plane can be estimated using control points, points of known coordinates on the desired plane. Given at least four correspondences between image coordinates and world coordinates, the extrinsic parameters are initially estimated by the following procedure.

a) Normalize and undistort the image coordinates.
b) Use Singular Value Decomposition to compute the homography that transforms from world coordinates to normalized undistorted image coordinates.
c) If the number of correspondences is greater than four, refine the homography by minimising the projection error.

The extrinsic parameters are then refined by the following procedure.

d) Extract R and T from the homography.

e) Refine R and T by minimising $\varepsilon_{projection}$ using Equation (26). This time K is fixed, W is the number of control points and N is 1. Given these constraints Equation (26) can be simplified to Equation (27).

$$\varepsilon_{projection} = \sum_{i=1}^{W} d(x_i, K[R|T]X_i)^2 \tag{27}$$

The procedure outlined in steps a-e estimates the optimal parameters to transform from world coordinates to image coordinates. An alternative is to estimate the projection matrix that transforms from camera coordinates to world coordinates. (Hartley & Zisserman 2003) suggest that this could be achieved by minimising the reconstruction error for the control points as in Equation (28).

$$\varepsilon_{reconstruction} = \sum_{i=1}^{W} d(X_i, K^{-1}[R|T]x_i)^2 \tag{28}$$

As noted earlier, $\varepsilon_{reconstruction}$ is subject to more uncertainty than $\varepsilon_{projection}$ due to the errors in the image point extraction. Subsequently this minimisation is traditionally not performed.

As highlighted in the performance analyst requirements in Section 2.1, due to the different configurations of stadia, a performance analyst does not have a standard camera pose. It is therefore of interest to understand the effect of the camera pose on the reconstruction error. (Brewin & Kerwin 2003) investigated the effect of tilt angles on the reconstruction error when using two-dimensional direct linear transformation (2D-DLT). They found that tilt angles in the range -2° to +6° had little effect on the reconstruction error. However in this work the expected angle is approximately 80°, so these results are not directly applicable.

(Hinrichs et al. 2005) also investigated the effect of tilt angle on the reconstruction error on 2D-DLT, but they considered angles up to 60°. They found that while the yaw angle had minimal linear effect on the y dimension of the reconstruction error, there was an exponential effect in the x dimension. At 60° of tilt this resulted in a reconstruction error of approximately 0.9 m in the x dimension and less than 0.05 in the y dimension. Yet these results were computed using 2D-DLT, which is not suitable here due to the large amount of lens distortion, so may not be applicable when considering (Kannala & Brandt 2006)'s camera model. Subsequently the effect of the expected camera pose on the reconstruction error should be investigated.

2.5.4 Summary

This section outlined the procedure for the transformation of image coordinates to world coordinates. This transformation requires a camera model comprised of intrinsic parameters and extrinsic parameters. Intrinsic parameters are those that transform from image coordinates to camera coordinates. The extrinsic parameters are those that describe the perspective projection that transforms from camera coordinates to world coordinates.

Camera calibration is the process for estimating the parameters of the camera model. Traditionally the parameters of the camera model are estimated by projecting known points onto the image plane and minimising the difference between the projected points and the actual image points.

The accuracy of the camera calibration can be assessed using the reconstruction error, the mean difference between a set of reconstructed points and their corresponding known world points. The reconstruction error is dependent upon the accuracy of the image point extraction; therefore the error in the image point extraction should be minimal. This leads to the objective:

Determine an accurate method to extract planar world points from a frame captured at the expected camera pose.

This thesis is concerned with extracting accurate player coordinates; therefore image points should be reconstructed to world points with minimal error. This leads to the objective:

Develop an accurate method for the reconstruction of planar points from a frame captured at the expected camera pose.

At an international hockey tournament, a performance analyst may not always have the same camera pose. The reconstruction error may not be consistent across these camera poses. This leads to the objective:

Assess the effect of camera pose on the reconstruction error.

As noted in Section 2.3 the use of a 4K camera gives larger objects in the scene and so is beneficial for their extraction and classification. Yet it is unknown if the increase in camera resolution has an impact on the reconstruction error. This leads to the objective:

Assess the effect of camera assembly on the reconstruction error.

The estimation of the extrinsic parameters requires control points that are known in the world coordinates and in image coordinates. The previous objectives in this section assume that the control points are extracted with zero error. This assumption is invalid which leads to the objective:

Assess the effect of control point errors on the reconstruction error.

2.6 Aim and Objectives

This chapter reviewed literature associated with extracting player coordinates for all hockey players using a single camera. Given the findings the following aim has been identified:

Develop an algorithm to extract player coordinates from footage captured with a single wide-angle camera at a field hockey tournament.

To meet this aim the following objectives have been identified:

1. ***Determine an accurate method to segment player blobs from an image.***
2. ***Train a Convolutional Neural Network to classify the contents of a blob.***
3. ***Determine an accurate method to extract planar world points from a frame captured at the expected camera pose.***
4. ***Develop an accurate method for the reconstruction of planar points from a frame captured at the expected camera pose.***
5. ***Assess the effect of camera pose on the reconstruction error.***
6. ***Assess the effect of camera assembly on the reconstruction error.***
7. ***Assess the effect of control point errors on the reconstruction error.***

A chapter is dedicated to each of these objectives. Following this, a final chapter will bring the work of the previous chapters together to assess the overall accuracy of the algorithm. This chapter will have the following aim:

8. ***Investigate how accurately the player's coordinates can be extracted from wide-angle field hockey footage.***

3 Segmenting Player Regions

3.1 Introduction

As noted in the literature review, a player's coordinates are extracted based on a blob segmented from the image. The segmentation of this region must be accurate to ensure that the coordinates are reliable.

The segmentation of an accurate region is a two-step process:

i. Classify the individual pixels as either player or non-player.

ii. Group player pixels into player regions.

In this chapter, step ii will only consider the 8-connectivity when grouping pixels in to player regions. Subsequently Chapter 4 will consider the grouping of these blobs into 'super'-blobs using semantic understanding of their contents.

This chapter investigates which of six segmentation algorithms is the most accurate. The six segmentation algorithms were chosen from existing literature:

1. Dominant Colour (Seo et al. 1997)
2. 3 x 3 Variance GMM (3V GMM) (von Hoyningen-Huene & Beetz 2009)
3. 9 x 9 Variance GMM (9V GMM) (von Hoyningen-Huene & Beetz 2009)
4. 17 x 17 Variance GMM (17V GMM) (von Hoyningen-Huene & Beetz 2009)
5. Red Blue Green (RGB) GMM (Zivkovic 2004)
6. Temporal Median (Cucchiara et al. 2003)

The accuracy is assessed at the pixel level using the F-score and at the region level using the Image Segmentation Assessment Tool (ISAT) ratios displayed in Figure 3.1. These assessment criteria are explained in detail in Section 2.4.1.

This chapter addresses the objective: *"Determine an accurate method to segment player blobs from an image"*. The novel contribution to knowledge for this chapter is the systematic analysis of a set of background subtraction algorithm for wide angle field hockey footage.

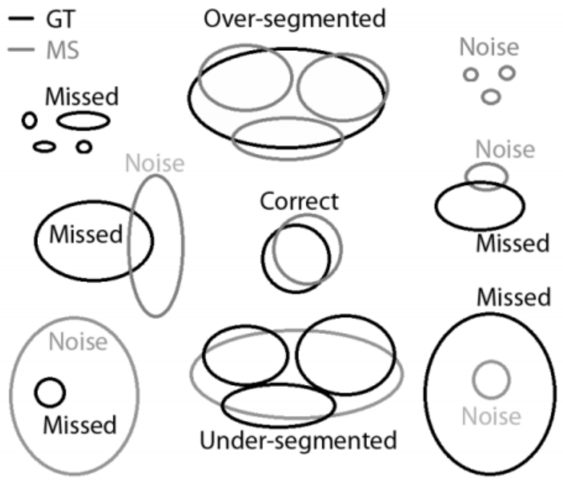


Figure 3.1: The possible resultant sets of blob segmentation. Here GT, ground truth, is the expected blob configuration and MS, machine segmentation, is the output of the blob finding algorithm. Blobs can be: Correct, Missed, Noise, Over-segmented or Under-segmented. Each set is mutually exclusive. (Source: (Mazhurin & Kharma 2012)**)**

3.2 Dataset

EuroHockey 2015 was an international field hockey tournament played in London in the summer of 2015. The male and female tournaments, which ran concurrently, had a combined total of 40 matches. The majority of matches were captured with a static wide angle camera to give a total of 153 quarters. The matches were played from 8:30

am until 10:30 pm and were not postponed due to weather conditions. Subsequently the captured video contains variation in lighting, weather conditions and the teams' playing kit colours. The footage, provided by England Hockey, was captured at 25 frames per second at a resolution of 3840 pixels x 2160 pixels. As illustrated in Figure 3.2, the camera was positioned at X= 48 m, Y= -15 m and Z= 7 m.

3.3 Method

For each algorithm the mean F-score and mean ISAT was calculated across the 20 frames of a test set. The 20 test frames were sampled from footage from four different matches in the EuroHockey dataset.

Each test video began prior to the start of one of the match's quarters. As illustrated in Figure 3.3, each of the test quarters was chosen to reflect the different environmental conditions experienced throughout the tournament.

Some of the tested segmentation algorithms developed a background model over a sequence of frames; thus there was a 100 frame initialisation period before the test frame. Subsequently the test set consisted of the 100th, 150th, 200th, 250th and 300th frame for each test video. The frames were taken from the opening sequence of the footage to ensure a distribution of players across the whole pitch. A 50 frame gap between test images allowed two seconds for player movement to ensure variation in the foreground across test frames.

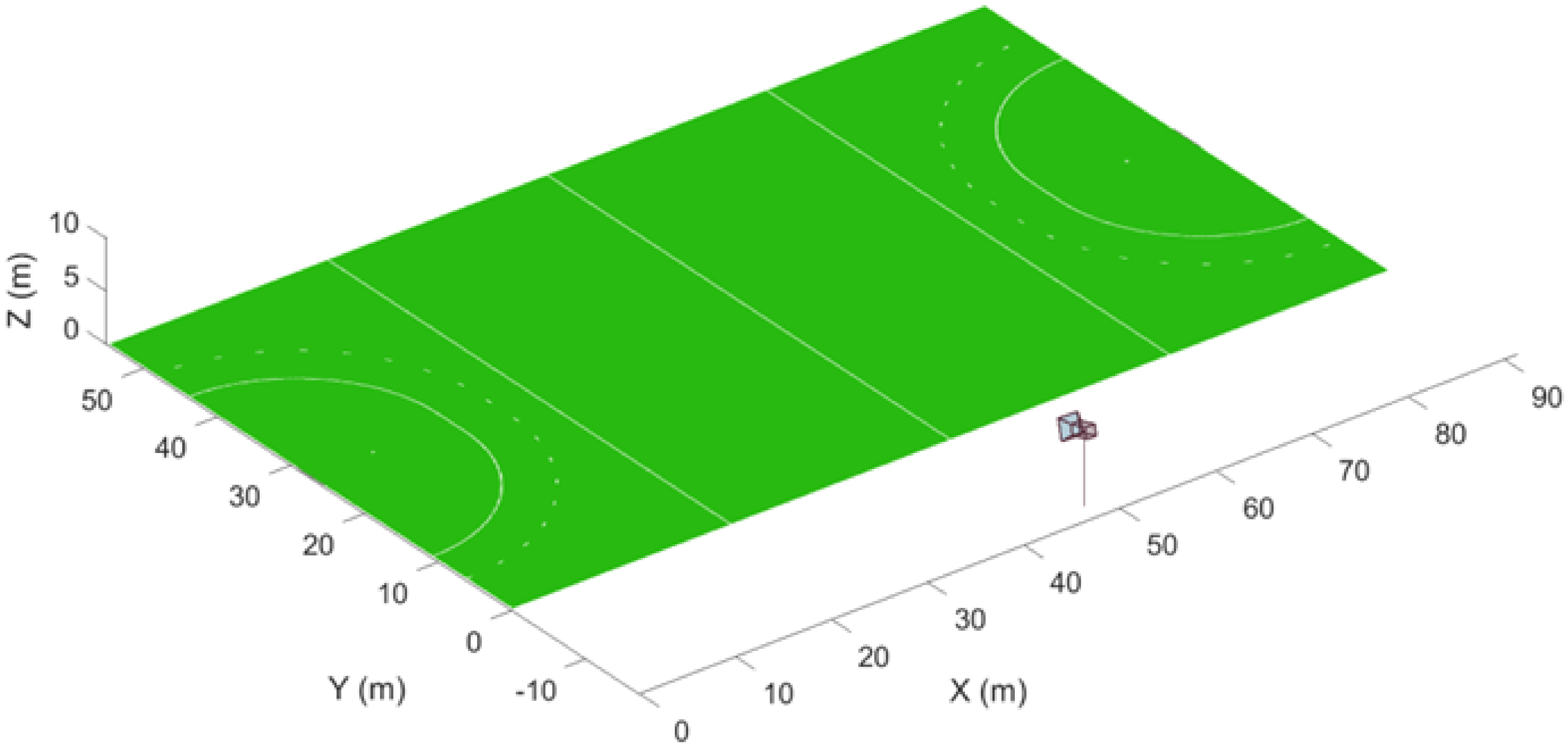


Figure 3.2: The camera position (X= 48 m, Y= -15m, Z= 7m) for the videos used in the investigation.

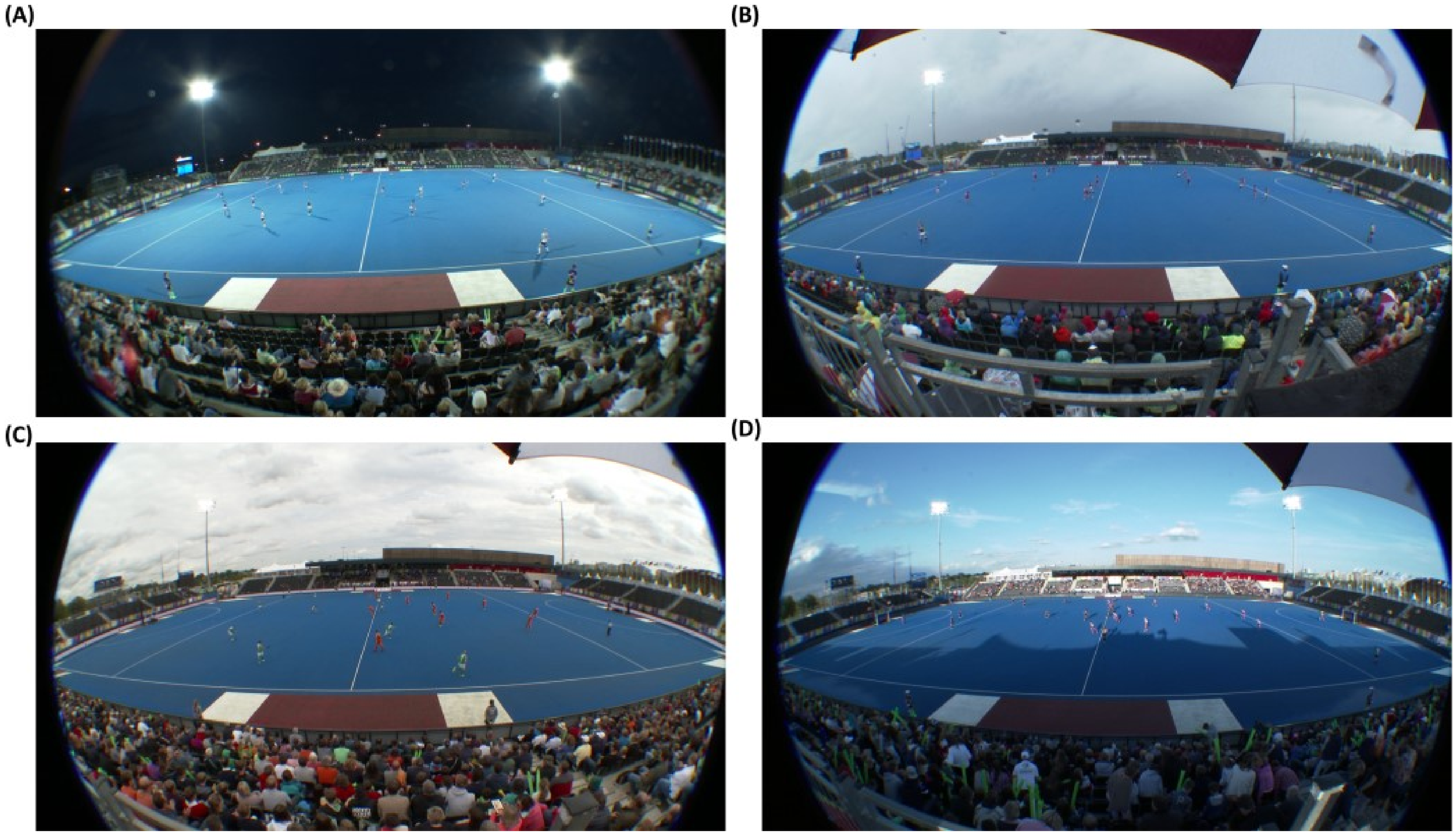


Figure 3.3: An example frame from each of the videos in the dataset. Each video was chosen due to different environmental conditions: (A) Night, (B) Rain, (C) Overcast, (D) Day-time Shadows.

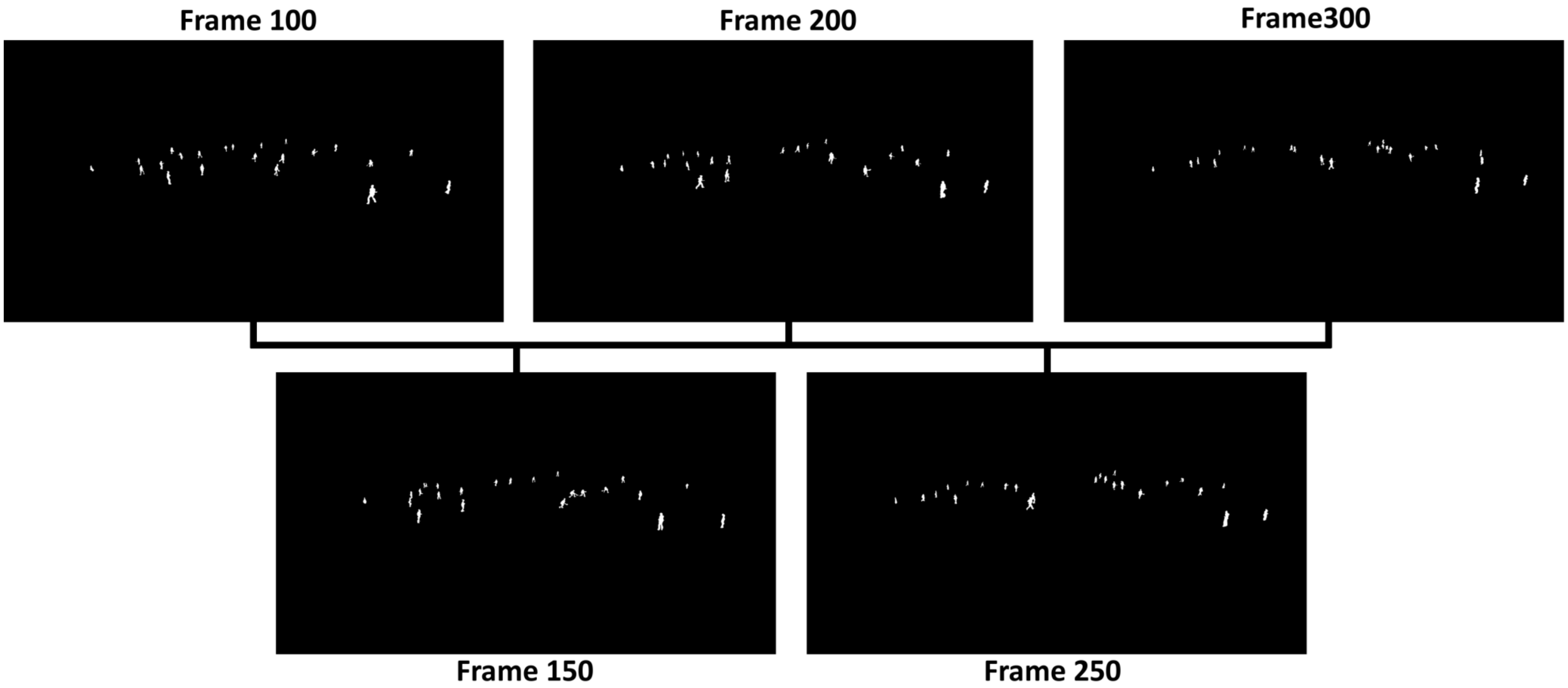


Figure 3.4: The sequence of ground truth foreground images for Video 1.

For each of the test frames, the RGB image was extracted from the video and a ground truth foreground binary mask was created manually by setting any pixel that formed a player as 1 and all other pixels as 0. Figure 3.4 displays the ground truth foreground sequence for Video 1.

For each test video, each of the segmentation algorithms was applied and the foreground mask at each test frame was extracted. A 3 x 3 median filter was applied to each foreground mask to remove any small blobs. The result of this was a set of machine segmented binary masks. To avoid noise in the spectator area, both the ground truth binary masks and the machine segmented binary masks were masked to the pitch region.

For each algorithm:

1. The F-score and ISAT was calculated for each test frame. To calculate the ISAT, T was set to 66% as in (Mazhurin & Kharma 2012).
2. The mean F-score and the mean ISAT was calculated for the frames in each video and for the frames in the entire dataset.

The mean rather than the accumulated F-score was calculated to account for any bias due to a different number of ground truth foreground pixels in each test frame. Similarly the mean ISAT was calculated due to a different number of blobs in each test frame.

3.3.1 Segmentation Algorithms

Six foreground segmentation algorithms were considered. These six algorithms were used in existing literature to segment field sport players from the field. The descriptions below identify the specific papers each was taken from. Each algorithm modelled the background of the scene. The foreground was those pixels that deviated from this model. The algorithms were:

Dominant Colour – This algorithm exploited the rule that the pitch must be a uniform colour. A pixel was classified as foreground if, for all channels, its colour was not within a threshold of the image's dominant colour. It was similar to the implementation by (Seo et al. 1997) without the assumption that green was the dominant channel. The algorithm's parameters are listed in Table 3.1 and were optimised using particle swarm optimisation on a field hockey dataset (Higham et al. 2016).

3 x 3 Variance GMM (3V GMM) – This method exploited the uniformity of the pitch. The image was transformed to grayscale. A 3 x 3 filter was moved over the grayscale image and the variance calculated at each pixel. The expected per pixel variance was modelled using a Gaussian Mixture Model (GMM). If a pixel's variance fell outside this model it was declared foreground. This method was very similar to that proposed in (von Hoyningen-Huene 2011). The only difference being the use of the GMM of (Zivkovic 2004). The algorithm's parameters were left as the defaults.

9 x 9 Variance GMM (9V GMM) – This method was similar to 3 x 3 Variance GMM except a 9 x 9 filter was used. Huene applied a 3 x 3 filter to an image of 352 pixels x 288 pixels. At this low resolution the pixel area of a player is relatively small and so high variance is observed across the whole player. In this Chapter the tests were performed with images of 3840 pixels x 2160 pixels. At this higher resolution the players have larger blocks of similar colour. These larger blocks will exhibit little variance so will be classified as background. The use of a larger filter attempted to resolve this issue. The algorithm’s parameters were left as the defaults.

17 x 17 Variance GMM (17V GMM) – This method was again similar to 3 x 3 Variance GMM except a 17 x 17 filter was used. The algorithm’s parameters were left as the defaults.

Red Green Blue (RGB) GMM – This was the GMM method proposed by (Zivkovic 2004). It used the OpenCV implementation with the parameters listed in Table 3.1. The parameters were optimised for a field hockey dataset using particle swarm optimisation (Higham et al. 2016).

Temporal Median – This was the temporal median method proposed by (Cucchiara et al. 2003) using the parameters listed in Table 3.1. A pixel was background if it was within a threshold of the median value of that pixel and foreground otherwise. The parameters were optimised for the field hockey dataset using particle swarm optimisation (Higham et al. 2016).

3.4 Results

Figure 3.5 displays the mean precision, mean recall and mean F-score across all the frames in the test set for each of the six segmentation algorithms tested. The RGB GMM and Temporal Median algorithms are superior to the other algorithms, however at the pixel level there is little difference between these two algorithms.

Table 3.1: The parameters used in the investigation as optimised in (Higham et al. 2016)

Algorithm	**Parameter**	**Description**	**Value**
Dominant Colour	*thresholdChannel1*	Threshold on channel 1 of colorSpace	0
	thresholdChannel2	Threshold on channel 2 of colorSpace	23
	thresholdChannel3	Threshold on channel 3 of colorSpace	0
	numberBinsChannel1	Number of histogram bins on channel 1 of colorSpace	1
	numberBinsChannel2	Number of histogram bins on channel 2 of colorSpace	231
	numberBinsChannel3	Number of histogram bins on channel 3 of colorSpace	1
	colorSpace	Color Space that the algorithm works in. 0 = RGB, 1 = HSV, 2 = Lab, 3 = YCrCb	3
RGB GMM	*alpha*	Parameter that defines the exponentially decaying envelope	0.034272
	backgroundRatio	Ratio of Gaussians that account for the background	0.447127
	threshold	Pixel foreground if greater than this many variances from background mean	59.7615
	fVarInit	Variance of new Gaussian model	26.4657
	fCT	Complexity reduction parameter	0.71691
Temporal Median	*threshold*	Pixel foreground if greater than value	22
	samplingRate	How often background resampled	10
	historySize	Number of frames in sample	65
	weight	Amount of influence given to previous samples	8

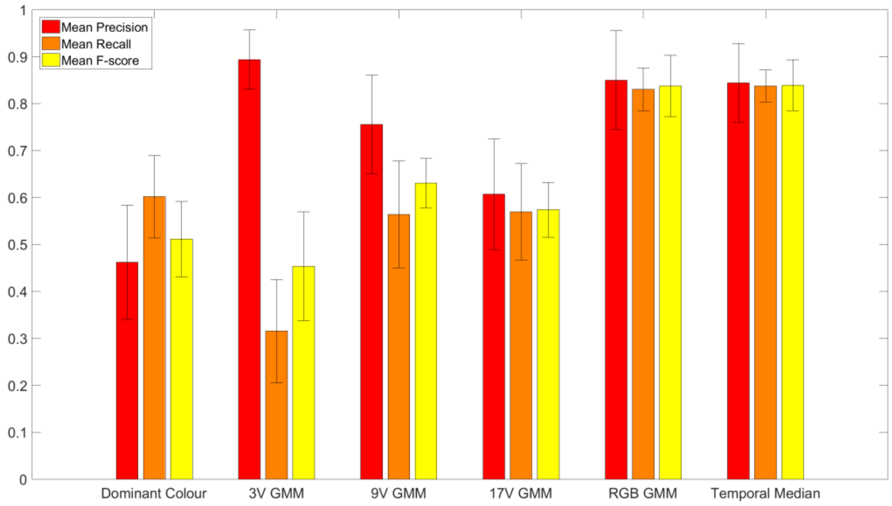


Figure 3.5: The mean precision, mean recall and mean F-score for each of the six algorithms. Error bars indicate the standard deviation across the frames.

Figure 3.6 displays the mean ISAT ratios across all the frames in the test set for each of the six segmentation algorithms. Again RGB GMM and Temporal Median are superior; an unsurprising result as a correctly segmented blob is dependent on correctly classified pixels. Yet the Temporal Median algorithm returns a higher percentage of correct blobs due to RGB GMM's tendency to over-segment.

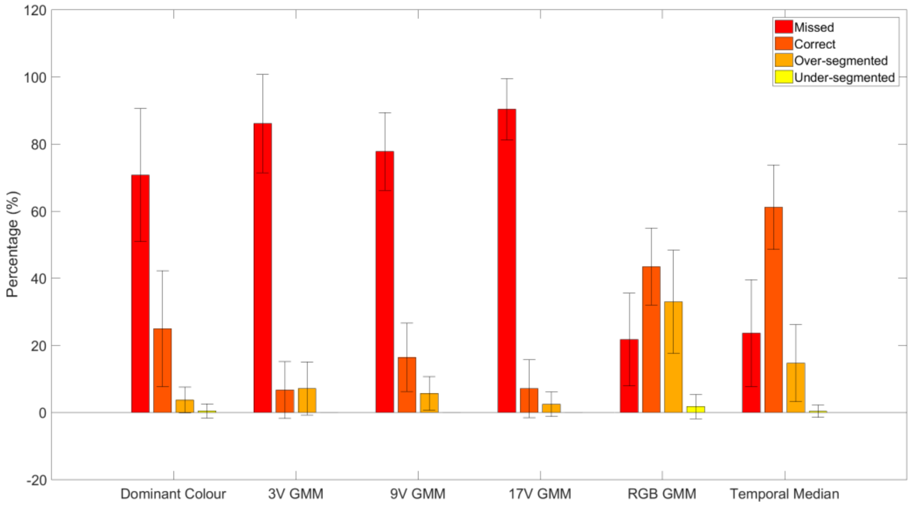


Figure 3.6: The mean ISAT ratios for each of the six algorithms. The ratios are the number of ground truth blobs that are classed as: Missed, Correct, Over-segmented, Under-segmented. Error bars indicate the standard deviation across the frames.

Figure 3.7 displays the mean number of noise blobs in each frame across all the frames in the test set for each of the six segmentation algorithms tested. This illustrates that the Dominant Colour and 3V GMM have over 125 noise blobs per frame. Assuming 24 blobs in each frame, this means there is approximately 5 noise blobs for every true blob. On the other hand Temporal Median has only 11 noise blobs per frame, less than 0.5 noise blobs for every true blob.

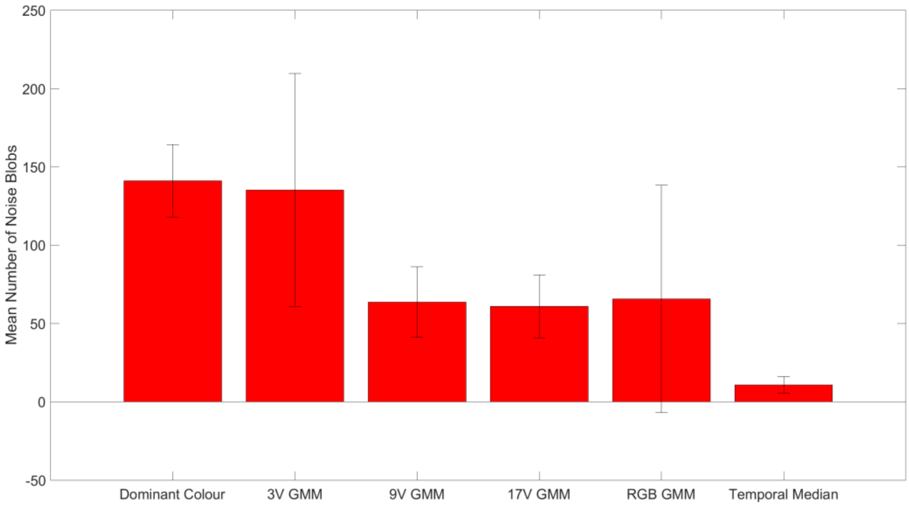


Figure 3.7: The mean number of noise blobs per frame for each of the six algorithms. Error bars indicate the standard deviation across the frames.

3.5 Discussion

Dominant Colour segmentation is commonly used to segment players in broadcast footage (Seo et al. 1997; Ekin & Tekalp 2003). It is employed because it does not make an assumption about the camera being static, though as indicated in Figure 3.5 it only achieved an F-score of 0.51. Despite achieving the third best recall and thereby correctly classifying 60% of those pixels that should be foreground correctly, only 46% of the pixels that were classified as foreground were correct. Figure 3.8C illustrates that the majority of the pixels incorrectly classified as foreground can be explained by the pitch markings that also deviate from the dominant pitch colour. A subsequent processing step could handle these pixels, yet both RGB GMM and Temporal Median achieve better recalls so it is deemed unnecessary of further investigation.

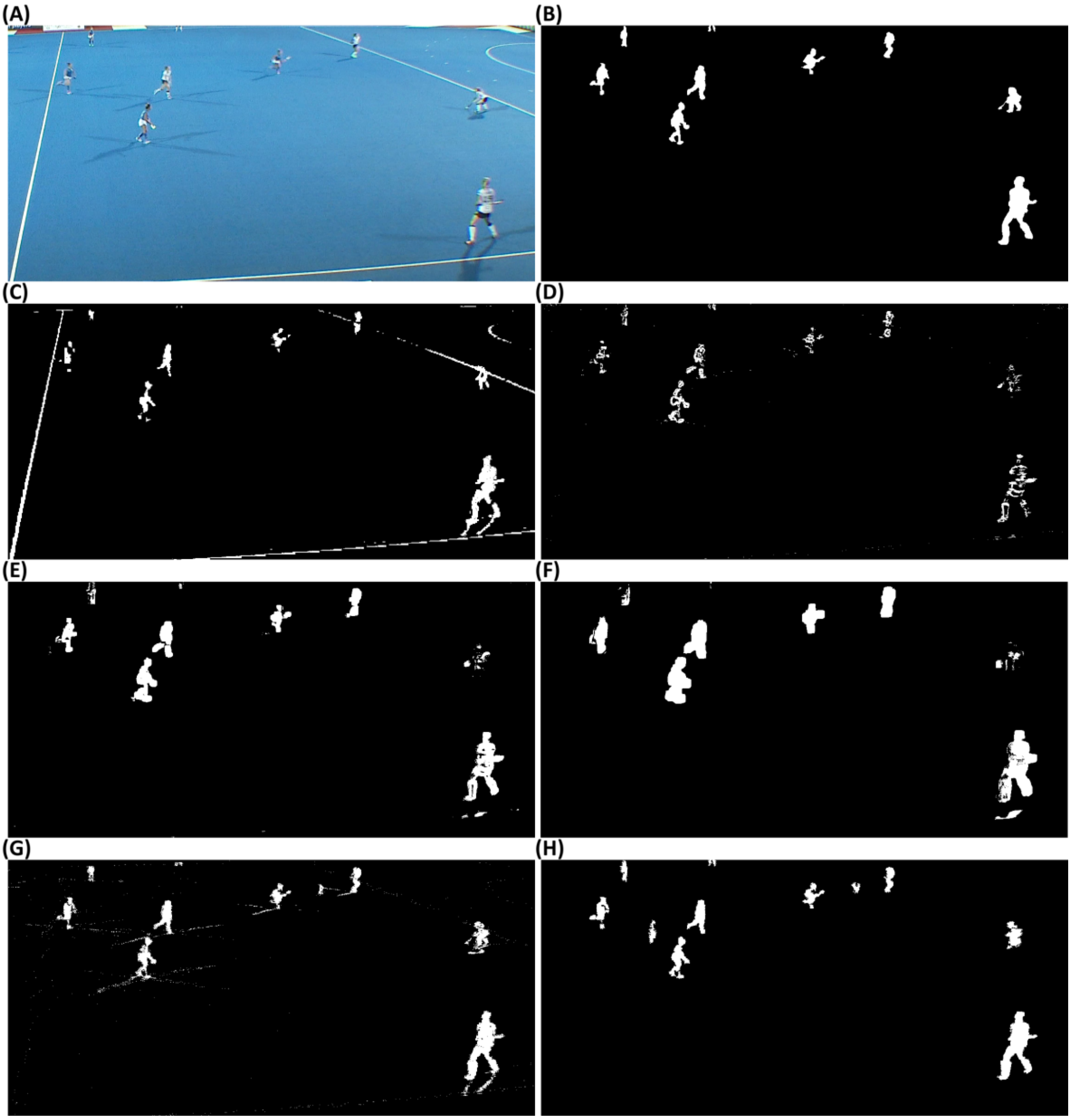


Figure 3.8: A segment extracted from one of the test frames. (A) The original RGB frame. (B) The ground truth foreground binary mask. (C) The binary mask output of the Dominant Colour algorithm. (D) The binary mask output of the 3V GMM algorithm. (E) The binary mask output of the 9V GMM algorithm. (F) The binary mask output of the 17V GMM. (G) The binary mask output of the RGB GMM algorithm. (H) The binary mask output of the Temporal Median algorithm. For all binary masks, black is background and white foreground.

The 3V GMM method proposed by (von Hoyningen-Huene 2011) achieved the highest precision, however also the lowest recall and as a result lowest F-score. Figure 3.8D shows the algorithm correctly classified the areas around the edge of players where there was a large variance across the 3 x 3 filter, however the internal areas of the player blobs that had low variance, for example the player's shirt, were classified as background. This resulted in player outlines being classified as foreground with the internal area of the player being classified as background. As the resolution of the images is higher than that used in (von Hoyningen-Huene 2011), this result was the expected prior to the experiment and was the justification for including 9V GMM and 17V GMM. These two algorithms were similar to 3V GMM, but used a larger filter to compute the variance. Figure 3.5 suggests that a larger filter does increase the recall but at the expense of reducing the precision. The application of the filter has a dilating effect on the blob. As illustrated in Figure 3.8D, E and F, the larger the filter the more the dilating effect and the higher the number of false positives around the blob's edge. Of those filter sizes tested the 9 x 9 is the most accurate. This filter dilates enough to fill the blob's interiors, but not so much that the blob's detail is lost. Other sizes such as 7 x 7 or 11 x 11 may improve of the accuracy; however both RGB GMM and Temporal Median achieved approximately 0.2 higher on the mean F-score, suggesting far superior performance, so it was felt unnecessary to investigate further.

It can be observed in Figure 3.5 that RGB GMM and Temporal Median return very similar mean F-scores of approximately 0.84. These two segmentation methods are superior to the other methods considered. This is supported by the ISAT ratios in Figure 3.6. Ideally the Correct ratio would be 100% and all other ratios 0% but the importance of the other ratios is not equal. Missed, Over-segmented and Under-segmented are all false negative errors, they are incorrect segmentations of expected blobs. Over-segmented and under-segmented errors happen when there is evidence for a player in a location but incorrect segmentation has occurred, they are known errors. Image processing techniques can be applied to split or merge these errors into correct blobs. Yet missed errors occur when there is limited evidence for a player at the location, they are unknown errors. With no evidence for a player it is more difficult to rectify missed detections. For this reason it is important to minimise the number of misses. RGB GMM missed a mean of 22% and Temporal Median missed a mean of 24% of the players in a frame, the other methods missed at least 70%. For this reason from this point on only these two methods will be considered.

Missed errors can occur when a player is stationary for a period of time. Both RGB GMM and Temporal Median build a per pixel model for the background and find deviations from this background. If a player is stationary for a period of time the model will be updated to include them in the background. At this point they will no longer be detected and will become a miss. Further to this, when they do start to move again, while their true blob will be correct, the previous stationary blob will result in noise until the background model is updated back to the true background. This kind of missed detection can be handled by assuming a stationary player remains stationary until movement is detected again.

While Temporal Median and RGB GMM achieve very similar F-scores, Temporal Median achieves a mean of 18% more correct detections (Temporal Median – mean 61% vs RGB GMM – mean 43%). This can be explained by the RGB GMMs tendency to over-segment players (Temporal Median – mean 15% vs RGB GMM – mean 33%). RGB GMM is also more susceptible to noise blobs (Temporal Median – mean 11 per frame vs RGB GMM - mean 66 per frame). The removal of non-player blobs is addressed in

Chapter 4. For these reasons, despite having a slightly higher missed rate (Temporal Median – mean 24% vs RGB GMM – mean 22%), Temporal Median will be used as the segmentation method throughout the rest of this thesis.

One of the weaknesses in the ISAT method is the lack of consideration for the semantics of a blob. To demonstrate this, consider a simplified ISAT that only considers Correct, Missed and Noise. To be classified as correct, the pixels of a machine segmented blob must overlap with at least T% of the pixels of a ground truth blob and vice versa. T is set to ensure that the machine segmented blob is a good representation of the expected ground truth blob. This T% of pixels must be 8-connected but can be located anywhere across the ground truth blob. Yet, a player's image coordinates are determined using the base of their bounding rectangle; if all the overlapping pixels are in the upper body region then poor coordinate accuracy can be expected. Equally if a machine segmented blob overlaps with a ground truth blob but less than T% of the ground truth blob, the machine segmented blob will be classed as noise and the ground truth blob will be classed as missed.

The experiment could have been improved by selecting test frames from throughout the test quarters. This would have ensured complete independence of the blobs in each of the test frames. It would also have tested how each of the algorithms copes with variations in light over a much longer period.

3.6 Summary

This chapter addressed the objective: *"Determine an accurate method to segment player blobs from an image"*. To do so it investigated which of six segmentation algorithms gave the most accurate foreground segmentation. The accuracy of the segmentation was assessed at the pixel level using the F-score and at the blob level using the Image Segmentation Assessment Tool (ISAT) ratios.

Temporal Median was found to be the most accurate segmentation algorithm. It correctly segmented 61% of the expected blobs. If it is assumed that the majority of blobs are single players, it follows that only approximate 60% of the players have been detected and subsequently can have a coordinate extracted. If a method can be developed to reform correct blobs from over-segmented blobs then the number of detected players increases to approximately 75%. While this result shows the promise of using Temporal Median as a background subtraction technique to find blobs, it does mean that approximately 25% of all players were not correctly segmented and subsequently cannot have their coordinates extracted.

The Temporal Median algorithm did not exploit any structure specific to field hockey; therefore with adequate tuning of the parameters similar results should be achievable on other field sport datasets.

The next chapter investigates using a convolutional neural network to classify blobs as players or non-players. Given this classifier it should be possible to merge over-segmented blobs to form correct ‘super’-blobs while simultaneously minimising the number of noise blobs in each frame.

4 Classifying Player Regions

4.1 Introduction

The output of the background subtraction procedure discussed in Chapter 3 is a collection of blobs that have been classified as foreground, however each of these foreground blobs may be classified as: correctly segmented, under-segmented, over-segmented or noise. Figure 4.1 illustrates examples of each of classes.

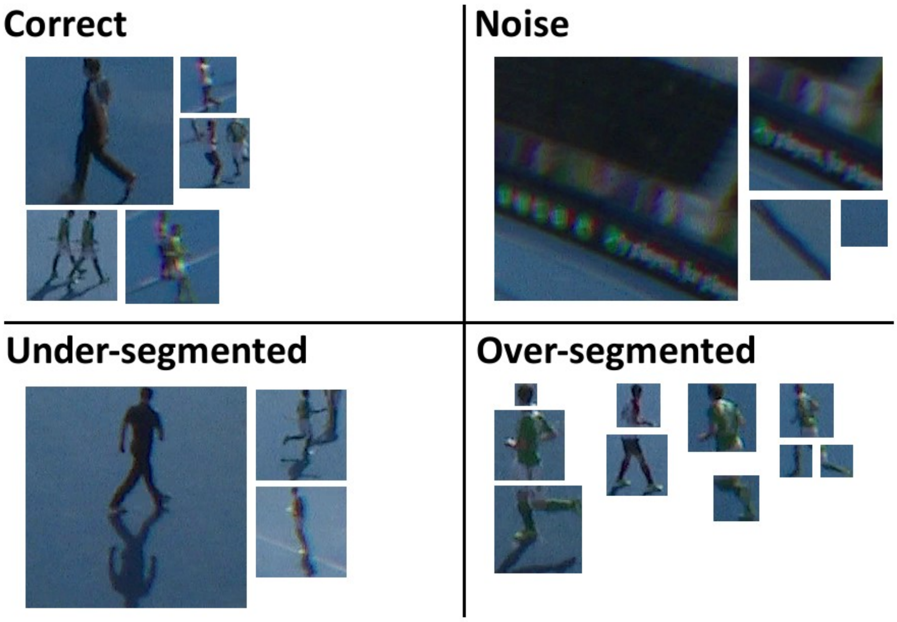


Figure 4.1: Scale examples of the four different classifications of blobs following the background subtraction of chapter 3. All blobs have been padded to be squares, as this is necessary for the input of the convolutional neural network.

Ideally only blobs that contain a correctly segmented player should be considered to have their coordinates extracted; yet understanding a blob's contents may aid in the handling of cases of poor segmentation. For example, take the large under-segmented blob in Figure 4.1, if it can be determined that the player's feet are not on the base of the bounding box then it may be possible to develop a procedure to determine the true location of the player's feet, and as such their coordinates. Consequently it is of benefit to classify each blob by its contents. As noted in the literature review, Convolutional Neural Networks (CNNs) have won each of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) classification challenges since 2012.

A CNN is a feed forward neural network inspired by the visual system. It is formed from layers of banks of convolution matrices. Each convolution matrix is a set of parameters that cause an activation when a specific structure is present in the input. Each region of the input is convolved with each convolution matrix to give an activation map, a different representation of the input. Each layer takes the activation map of the previous layer as input. The final layer of the CNN classifies the input based upon the representation of the penultimate layer.

Applying a CNN to a specific classification task is a case of learning the parameters that classify the data correctly. To do so a loss function is minimised over a training set, a collection of input and class label pairs. As CNNs are susceptible to over-fitting to the training data, accuracy over the independent validation set is used to determine which iteration of the learning process has the best generalisability. Accuracy over the validation set is also used to choose between the different models. Finally the accuracy of the best model is assessed over a test set.

This chapter addresses the objective: *“Train a Convolutional Neural Network to classify the contents of a blob”*. This Convolution Neural Network can be used to remove non-player blobs, thereby increasing the precision of the coordinate extraction algorithm, and, as will be presented in Chapter 10, to reform over-segmented blobs, thereby increasing the recall of the coordinate extraction algorithm. The novel contribution to knowledge for this chapter is the application of a convolutional neural network in the domain of field hockey player classification.

4.2 Method

4.2.1 Classes

The model was trained to classify four possible classes:

1. Single player – well segmented.
2. Single player – bottom poorly segmented. If it is known a player's feet are not in contact with the base of the bounding box, an attempt could be made to estimate the true location of the player's feet. This class predominately occurs when the player has a strong shadow protruding towards the camera. An attempt was made to remove shadows using the expected Hue, Saturation and Value (HSV) of the background (Cucchiara et al. 2003); however this method did not eliminate all shadows.
3. Multiple players.
4. Not a player. This class contains not only the blobs that contain no player but also the blobs that are only segments of a player. The player's position cannot be extracted from a player segment; therefore the blob should be discounted from further analysis.

Examples of these classes are illustrated in Figure 4.2.

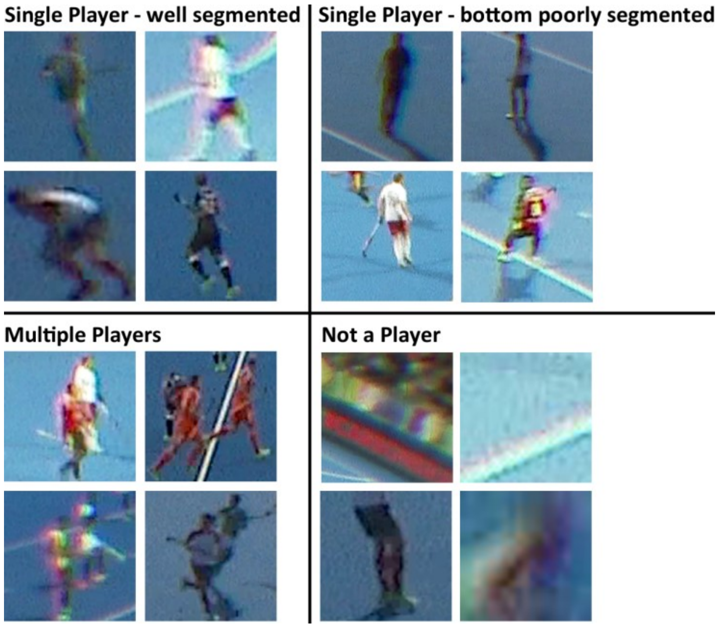

Single Player - well segmented
Single Player - bottom poorly segmented
Multiple Players
Not a Player

Figure 4.2: The four classes included in the study were: Single Player-well segmented, Single Player – bottom poorly segmented, Multiple Players and Not a Player. The Not a Player class included both noise blobs and blobs that contain segments of players.

While other classes such as 'Single player – top poorly segmented' could have been included, the overall aim of this work was to identify player coordinates. Throughout the work it is assumed that the player coordinates are the centre of the base of the bounding box; having information about the top of the bounding box does not aid the identification of this point. Therefore the first class could be named 'Single Player – bottom well segmented' and these blobs would be included here.

4.2.2 Image Collection

The aim of the learning process is to learn the parameters that correctly classify the contents of an image. Or alternatively, if each class is thought of as a distribution in the feature space, to find the parameters that correctly partition the distributions. Therefore the training set should be representative of these distributions.

Five quarters were chosen from the EuroHockey Dataset (Section 3.2) to form a training video dataset. As the training set must be labelled manually, five quarters was chosen to give sufficient data while not making the labelling task excessively time intensive. The five quarters were chosen to be representative of the general tournament based upon two factors:

- Each team used two difference uniforms throughout the tournament. The quarters were chosen to include the majority of the different uniform colour's used.
- Each quarter of the tournament was classified based upon the environmental conditions at the start of the quarter. The classes were 'Morning Shadows', 'Day – Raining', 'Day – Clear Sky', 'Evening Shadows' and 'Night'. The selected quarters comprised one from each of these classes.

The same criteria were used to select a further five quarters to form a test video dataset that could be used to assess the generalisability of the model.

One of the assumptions when training a CNN is that the data is independent and identically distributed. As video is time series data, the independence assumption is violated across single frames; the content of frame N is dependent upon the content of frame $N-1$. This violation can be mitigated by sampling the video at a lower frequency. Both the image training set and validation set were formed by sampling the training video dataset at 0.2 Hz. 0.2 Hz was chosen as it provided sufficient data, yet also maintained a reasonable amount of time between samples, five seconds. To achieve independence, the two datasets were sampled from different segments of the video. The allocation schema is formulised in Equation (29).

$$Dataset = \begin{cases} Training & frame \leq trainMax \\ Validation & frame \geq valMin \end{cases} \tag{29}$$

Where for each video in the training dataset, $trainMax$ is given by Equation (30) and $valMin$ is given by Equation (31).

$$trainMax = 4 * \frac{finalFrameOfVideo - 1500}{5} \tag{30}$$

$$valMin = trainMax + 1500 \tag{31}$$

The frames in the range between $trainMax$ and $valMin$ were not sampled. This gave an approximate training frame to validation frame ratio of 4:1.

The image test set was formed by sampling the test video dataset at 1/40 Hz. This lower frame rate was chosen because the test set did not need to be as large as the training set.

For each sampled frame, the following procedure was applied to each segmented blob:

1. The bounding box was extracted. A bounding box is defined by the (u, v) coordinate of the upper left corner, its width and its height.
2. The bounding box was padded to make it square. The padding was always applied to ensure the centre of the base of the bounding box, the point taken as the blobs coordinate on the calibrated plane, remained constant.
3. The image bounded by the new bounding box was extracted.
4. The image was resized to 224 pixels x 224 pixels x 3 colour channels, the dimensions necessary for input to the CNN.

5. The image was manually labelled with one of the four included classes.
6. For images in the training or validation set: if the correct class was 1-3, one of the player classes, then the dataset was augmented by adding a duplicate image reflected about the y-axis. As noted in (Krizhevsky et al. 2012), the semantics of an image are invariant to reflection and as such this is a low cost way to increase the training and validation set size. This augmentation was not performed for Class 4 as it would exaggerate the class imbalance discussed in the following paragraph.

Table 4.1 displays the number of examples in each class in each dataset. The dataset has a class imbalance because it is constructed by manually labelling all the blobs from a sequence of frames and not all the classes occur with the same likelihood. A class imbalance can lead to poor classification results because the model biases towards predicting the over-represented class. For instance in a binary classification problem with a class imbalance of 99 negative examples for every 1 positive example, the model will achieve a 99% classification accuracy simply by classifying everything as negative. To overcome this, under-sampling, balancing the number of examples in each class by randomly sampling, was employed and is described in more detail in Section 4.2.5.

Table 4.1: Number of examples in each class of the datasets

	Training Set		Validation Set		Test Set	
Class	Number	Percentage	Number	Percentage	Number	Percentage
1	15212	48%	3406	46%	1360	42%
2	1294	4%	350	5%	79	2%
3	3122	10%	684	9%	219	7%
4	12228	38%	3004	40%	1576	49%
Total	31856		7444		3237	

4.2.3 Classification Methods

To train a CNN from scratch requires a vast amount of data; the training set for the ILSVRC classification task is approximately 1.3 million images (Russakovsky et al. 2015). As noted in the previous section the training set here is approximately 32000 images, therefore poor classification results can be expected. An alternative is to take a CNN learnt on a much larger dataset, such as ILSVRC, and transfer it to the required task. This process, known as transfer learning, exploits the fact that the learnt low level features are similar across classification task and dataset (Yosinski et al. 2014).

Transfer learning can take two different approaches:

1. Feature Extractor - Use the existing CNN as a feature extractor for the new dataset. Train a Support Vector Machine (SVM) on the extracted features.
2. Fine-tuning - Train the CNN on the new dataset so it adapts to the new task.

Typically the first approach is employed when the dataset is small and the second when the dataset is larger. The dataset used here is midsized when considering training CNNs; therefore both the approaches were considered to find which gave the best classification accuracy.

In the case of fine-tuning the CNN, the classification accuracy is dependent upon how many layers of the model are allowed to adapt. If it is assumed that the low layers of the base CNN are similar for all datasets, over fitting to the training dataset may be reduced and better accuracy achieved by only fine-tuning the higher layers. Subsequently the study investigated how the classification error was affected by fine-tuning to different depths of the network.

A similar decision must be made for the feature extractor. Again, if it is assumed that the low layers of the base CNN are generic and that the higher layers are more specific to the original task, it follows that the features extracted at different layers will achieve different classification errors. Therefore, the study investigated how a Support Vector Machine (SVM) trained on different layers of the existing CNN affected the classification error.

4.2.4 GoogLeNet

The existing CNN used in the study was GoogLeNet (Szegedy et al. 2015). This CNN was the winner of the 2014 ILSVRC classification task achieving a top 5 classification error of 6.66% (Russakovsky et al. 2015). The main contribution of GoogLeNet was to achieve improved performance while keeping the computational budget constant. To achieve this the authors introduced the Inception module (Figure 4.3), an extension of the Network in Network (Lin et al. 2013). Whereas prior to GoogLeNet each layer was one operation and the layers were performed sequentially, the Inception module parallelises convolutions of different sizes into a single layer. This allows image structures of different sizes to be determined at each layer and therefore increases the representative power of the network.

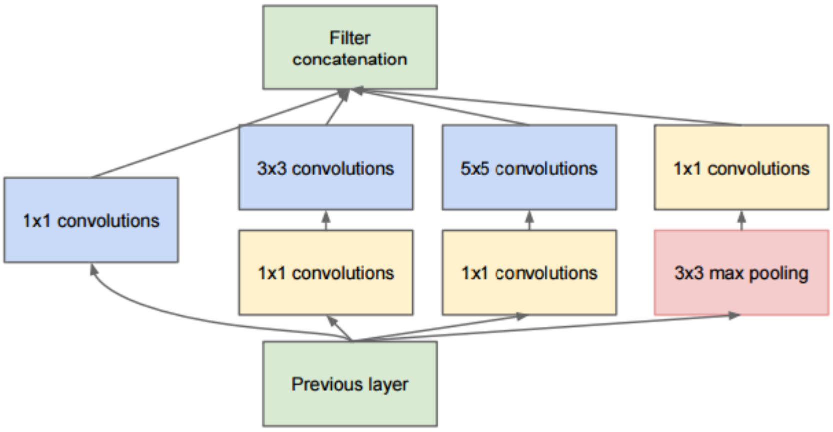


Figure 4.3: The architecture of an Inception module. (Source: (Szegedy et al. 2015)**)**

The complete architecture of GoogLeNet stacks nine Inception modules to form a 22 layer network (Figure 4.4).

4.2.5 Classification Procedure

As noted in Section 4.2.2 the dataset had a class imbalance. This imbalance was addressed in the training set by random under-sampling.

The under-sampling procedure was as follows:

1. Determine the class with the lowest number of training images. This was class 2 which had 1294 images. All of these images were added to the balanced training set.
2. For each of the other classes, 1294 randomly sampled images were added to the balanced training set.

The two classification methods use training sets in different ways so more details can be found in the relevant sections.

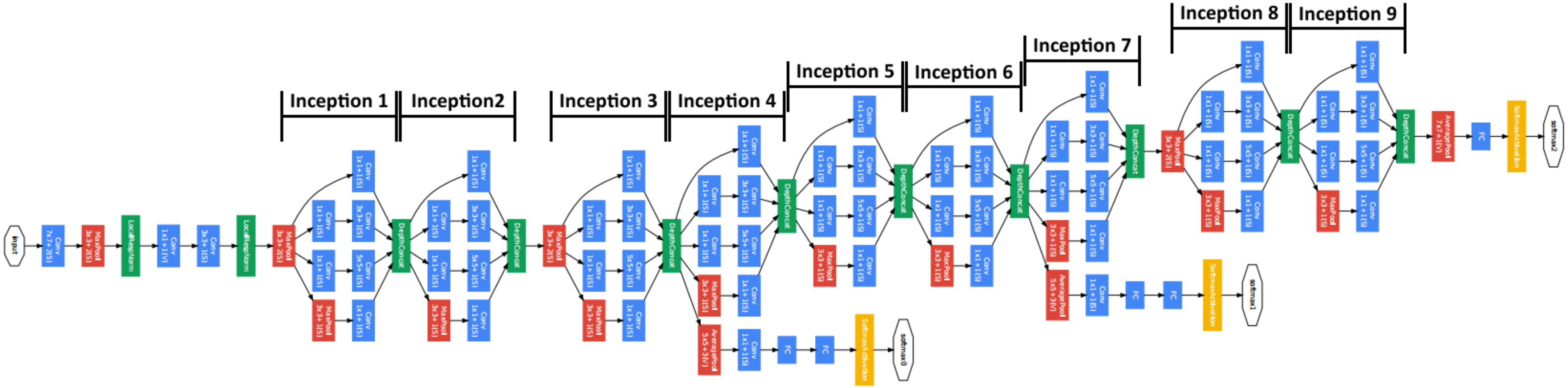


Figure 4.4: The GoogLeNet (Szegedy et al. 2015) **architecture is formed from 9 Inception modules for a total of 22 computation layers.**

As the validation set classification error was used for model selection, the validation set also needed to be balanced. Yet to ensure the comparison was valid, it had to be consistent across all the different models. Therefore the validation set was under-sampled once and held constant for all models. This procedure gave a balanced validation set with a total of 1400 images.

Class imbalance was not an issue within the test set as the classification error of the individual classes can be analysed using the confusion matrix.

All CNN operations were performed using MatConvNet (Vedaldi & Lenc 2015) version 1.0-beta20, a CNN framework for Matlab. In pre-processing, each image was normalised. The feature extractor approach normalised the image by subtracting the mean image computed over the ILSRVC 2014 dataset. This followed the normalisation procedure used when GoogLeNet was originally trained. The fine-tuning approach computed the mean RGB triplet over the combined training and validation dataset and subtracted this from each pixel in the image.

Feature Extractor

The feature extractor approach used the CNN to extract a feature vector from each image and then learns a SVM to classify the image based upon this feature. The Inception modules provide a set of logical points in the network from which to extract the feature vector. Therefore for each image the activation after each Inception module was extracted to give nine different feature matrices. All SVM models will be referred to by the Inception module prior to the point of feature extraction, for example **Inception 9** was formed from the activation after 9th Inception Module.

Using the entire activation map would result in a very large feature vector. For example the activation map after the 5th Inception module is a 14 * 14 * 512 matrix, a total of 115,200 elements. A feature vector with this many elements is highly likely to over-fit to the training data. Therefore the size of the feature was reduced by average pooling along the 3rd dimension. In the example of the 5th Inception module this results in a 512 element feature vector, created by taking the mean of each of the 14 x 14 element tensors (Figure 4.5). This dimension reduction procedure was chosen as it was used after the 9th Inception module in the original GoogLeNet architecture.

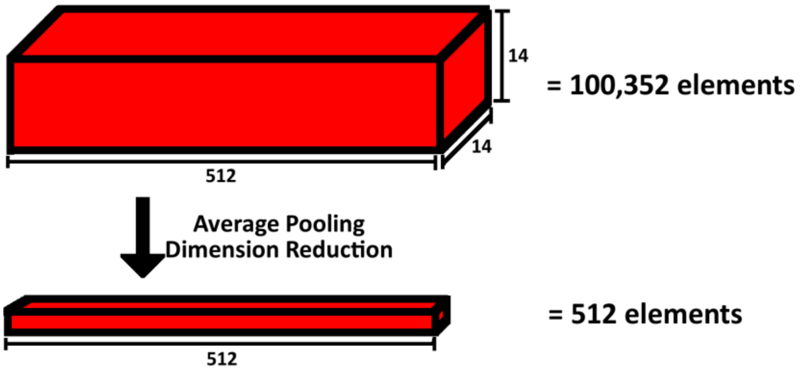


Figure 4.5: Average pooling dimension reduction was used to reduce the number of elements in the feature vector. Here, for the 5th Inception module, average pooling takes the mean of each of the 14 x 14 tensors to reduce the vector from 100,352 elements to 512 elements.

Table 4.2 lists the nine feature vectors that were extracted from each image.

Table 4.2: The nine different feature vectors extracted for each image from the GoogLeNet architecture.

	Number of elements in feature vector
Inception 1	256
Inception 2	480
Inception 3	512
Inception 4	512
Inception 5	512
Inception 6	528
Inception 7	832
Inception 8	832
Inception 9	1024

As nine features are extracted for each image, there are nine different extracted datasets. Each extracted dataset was normalised by subtracting the mean vector and then dividing by the vector of the per channel standard deviations. Any channel that had no standard deviation was discarded as it gave no discriminative ability.

For each of the nine different extracted datasets 100 fold Monte Carlo like validation was employed. Monte Carlo validation is a way to ensure that a model's validation accuracy was not achieved by chance due to similarities between the training and validation set. In a Monte Carlo validation a random segment of the data set forms the training set and the remaining proportion forms the validation set. The model is trained on the training set and then the classification accuracy calculated on the validation set. This procedure is repeated N times and the mean validation classification accuracy is a better estimate of the generalised classification accuracy. Due to the distinction between the training set population and the validation set population a traditional Monte Carlo simulation could not be performed here. Instead, for each of the N (N=100) iterations the training set was sampled using the under-sampling procedure outlined in the previous section. As noted earlier, the validation set was kept constant to facilitate the comparison of models.

Therefore the training and assessment procedure for the Feature Extractor models was as follows:

- For each Feature Extractor model:
    - For each of 100 iterations:
        i. Under-sample the training set.

    ii. Use the training set to train a four class linear SVM. A different SVM kernel may have improved the results, but was beyond the scope of this study.
    iii. Calculate the classification accuracy on the validation set.
2. Calculate the mean classification accuracy across the 100 iterations.

Fine-tuning

The fine-tuning approach re-trained the CNN for the task of player blob identification. To do so the existing classifier layer was removed and replaced with a four class softmax classifier. The parameters of the classifier layer were initialised using the procedure in (He et al. 2015). In addition to this, a Dropout layer set to 0.5 dropout was added between the 9th Inception Module and the final fully connected layer.

Dropout (Srivastava et al. 2014) is a method of regularisation used to stop the model over-fitting to the training data. For each batch in the training procedure, a percentage of the parameters in the network are randomly set to zero. These zero parameters have no impact on the output and the network must learn to classify without them. This reduces the networks dependency upon a limited set of features.

To ensure model convergence, the network was trained for 300 epochs. An epoch is one complete pass through the training dataset. The model after epoch 300 is not necessarily the best; the validation classification error may increase as the model over-fits to the training data. Therefore the model at each epoch was retained so the best model could be returned.

The training dataset was under-sampled once per epoch following the procedure outlined previously. (Krizhevsky et al. 2012) employed data augmentation to increase the variation in the dataset. They state that certain augmentations can be made to an image without changing its semantical content. This included adding random noise to the images, randomly reflecting the images, randomly rotating the images and randomly cropping the images. Here only random noise was added to the images. As noted in the Image Collection section (Section 4.2.2), the dataset was augmented offline by reflecting the images in classes 1-3 about the y-axis. Random rotations were not included because due to the perspective camera view it can be assumed that a player's feet will always be at the bottom of the blob. Random crops were not included because the point of this study was to find blobs that contain entire well segmented players.

As this is a classification task, the cross-entropy loss was used. The original GoogLeNet architecture has two additional loss branches to facilitate training in the lower layers. These additional loss branches were not considered here so were disposed of. Mini-batches of 16 images were used to estimate the loss of the model. The mini-batch size was limited by the memory of the GPU. The loss was back-propagated (Rumelhart et al. 1986) through the network and the parameters updated using adam (Kingma & Ba 2015). Adam is an adaptive learning rate update procedure. A key problem when training a CNN is setting the learning rate. To achieve optimal classification accuracy the learning rate may need to be reduced over time and may vary across the different parameters. Adam handles this by adapting the learning rate for each parameter dependent upon the magnitude of its recent updates. Further to this, as mini-batches are used, the gradient of the loss can be noisy, resulting in suboptimal updates. Adam uses momentum to smooth the updates handling any noise in the loss. The hyper-parameters for adam were set to the defaults in (Kingma & Ba 2015). This was as follows: *base learning rate* = 0.001, *beta1* = 0.9 and *beta2* = 0.999.

Weight-decay, another form of regularisation, was employed to further reduce the chance of overfitting to the training set. At each update, weight-decay subtracts a fraction, *a*, of a weight's magnitude from the weight. This has the effect of decaying the parameters towards zero, which ensures that all the parameters have a similar relative size and as such all impact on the output. In this study *a* was set to 0.0005, the default value for MatConvNet.

As noted previously, improved accuracy may be achieved by only fine-tuning the higher layers of a CNN. Again the Inception modules give a logical way to decide which layers can be fine-tuned. Subsequently eleven different GoogLeNet models were trained. The model with the designation **All** allowed all layers to be fine-tuned. The models designated **Inception *N***, where *N* is an integer between 1 and 9, allowed the fine-tuning of the layers in that Inception module and up. The model designated **Classifier** only tuned the final layer, the softmax classifier. This model is similar to the **Inception 9** Feature Extractor, but with a softmax classifier rather than the SVM.

Monte Carlo validation is not typically performed when training a CNN and as such was not employed in this work.

Therefore the training and assessment procedure for the fine-tuned models was as follows:

- For each model:
    - For 300 epochs:
        i. Under-sample the training dataset
        ii. For each training batch:
            a. Calculate the loss
            b. Back propagate the loss through the network to calculate the gradient at each layer
            c. Use adam to update the layer’s parameters
        iii. Calculate the loss over the validation set

Each model was trained on a HP Z230 Workstations (Intel® Core$^{\text{TM}}$ i7-4790 processor - 3.6 GHz, 16 GB RAM, Intel HD Graphics 4600).

4.2.6 Results on test set

Once all the models had been trained, the best model was identified as the one with the lowest classification error. This model was then used to classify the test set. A confusion matrix was used to interrogate the misclassification between classes.

4.3 Results and Discussion

This section will first consider the Feature Extractor models before moving on to the fine-tuned models.

Feature Extractor

The line in Figure 4.6 displays the mean classification error for the validation set for each of the nine SVM models. The mean was calculated across 100 randomly sampled training sets. The error bars indicate the standard deviation. The mean classification error for all nine SVMs was between 20% and 23%. The following paragraphs attempt to explain the differences in classification error for each of the SVM models.

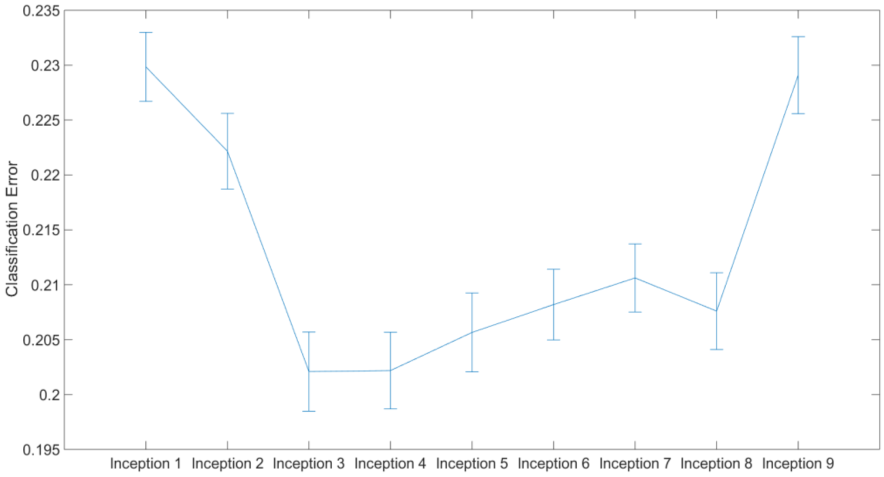


Figure 4.6: For each of the nine SVM models, the mean classification error on the validation set. Each SVM model was trained on a feature vector extracted after one of the nine Inception modules. The mean of each model was calculated across 100 randomly sampled training sets. The error bars indicate the standard deviation.

The classification error for **Inception 1** and **2** are comparatively high. This is because:

1. These two feature vectors are those closest to the original image. Therefore there has not been sufficient opportunity to abstract the data into a form that is suitable for classification. This highlights why depth is so important to the accuracy of a CNN.
2. These two feature vectors contain a relatively small number of elements. A smaller feature vector has fewer elements with which to discriminate between data. It is possible, that better classification accuracy could be achieved by using the entire feature vector rather than reducing the feature dimensions by average pooling. This was not investigated because as demonstrated later, superior results can be achieved by fine-tuning the network.

The error is at a minimum for **Inception 3** and **4**. This suggests that these feature vectors are at the optimal point; far enough from the input to have sufficient abstraction of the data, but not far enough to have learnt abstractions specific to the dataset. After this, the classification error progressively increases through **Inception 5-7** as the feature vectors abstraction becomes more task specific. **Inception 8** is somewhat of an anomaly as the classification error decreases, however this decrease is insufficient to make the model competitive. Finally **Inception 9** has performance comparable to **Inception 1**. **Inception 9** forms the last layers before the classification itself. Therefore the data has been abstracted to a form very specific to the original classification task.

Fine-tuning

As displayed in Figure 4.7, the classification error of a CNN is assessed at each epoch to determine how the training is affecting the performance of the network. Here, the classification error is illustrated for the training and validations sets for the **All** CNN.

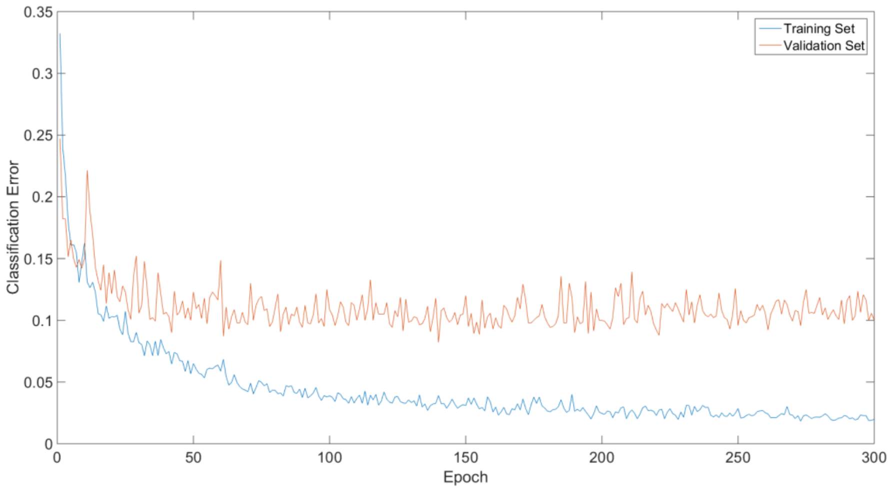


Figure 4.7 The classification error over time for the All CNN on the training set and the validation set.

It is the performance on the validation set that indicates how the network will perform on unseen data. It is clear that the lines for the training set and validation set diverge, this is a sign that the network is over-fitting to the training dataset. This may suggest that the GoogLeNet model is too complicated, has too many parameters, for the task with the current training dataset. The following would resolve this:

- Use a simpler CNN model. The task is only a 4 class classification so the 6.8 million parameters of GoogLeNet may be more than necessary. (Howard et al. 2017) recently proposed MobileNets, a CNN architecture that gives similar performance to GoogLeNet on the ILSVRC task. It is designed to run on mobile devices and subsequently contains only 4.2 million parameters. Using fewer parameters should reduce overfitting, which may mean higher validation classification accuracy.
- Increase the regularisation in the network. This should increase the generalisability of the network. Options include increasing the weight decay parameter or increasing the amount of dropout.
- Increase the size of the training dataset. A CNN trained on more data is better able to estimate the data's underlying distribution and subsequently generalise to unseen data. Class 2 is under-represented in the dataset, which made the under-sampling of the other classes necessary. More examples of this class could be sourced to rectify this imbalance. While the other classes are a count of the number of players in a blob, this class identifies if a single player is poorly segmented. Chapter 3 attempted to optimise the segmentation; therefore it

follows that actual examples of this class should be scarce. Instead the class could have been augmented by simulated examples.

Figure 4.8 illustrates the validation classification error for each of the CNN models. At epoch 300 most of the CNN models had a classification error less than 20% which meant they outperformed the best SVM model. This suggests that for this task, fine-tuning the model is the more suitable method for the application of an existing CNN. This may be because the task is significantly different from the ImageNet classification task the original GoogLeNet model was trained upon. The exception to this was the **Classifier** model, which gave very similar performance to the best SVM model. This may be expected as the only difference between the **Classifier** CNN and **Inception 9** SVM is the different classifier. However, it goes against the findings of (Tang 2013), who suggest that a SVM typically outperforms a softmax classifier.

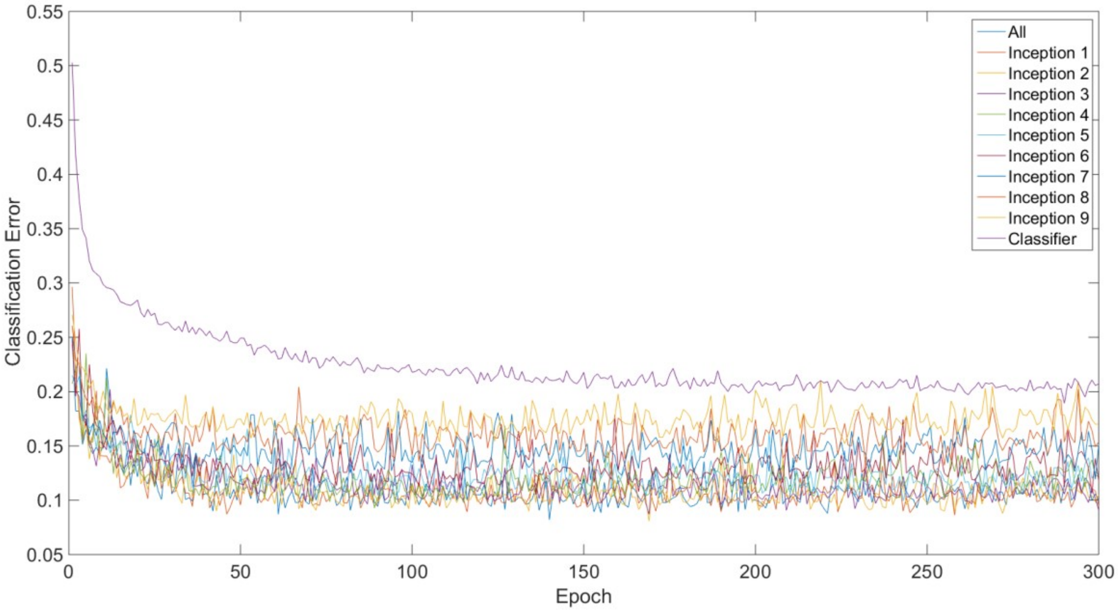


Figure 4.8: The validation classification error for each of the CNN models across the 300 epochs.

Figure 4.8 also suggests that it was unnecessary to train the majority of models for 300 epochs; there was little improvement in the validation classification error after about epoch 50. This combined with displaying all 11 traces makes it difficult to visually determine which model is superior. Subsequently Figure 4.9 displays the validation classification error for the best four models: **All**, **Inception 1**, **Inception 2** and **Inception 3**. It is clear from these traces that these four models all gave a similar validation classification error of just over 10%. This suggests that the existing features at lower layers are already suitable for the task and that training has little effect on them. This is further support for the notion that the lower layer features are generic across task.

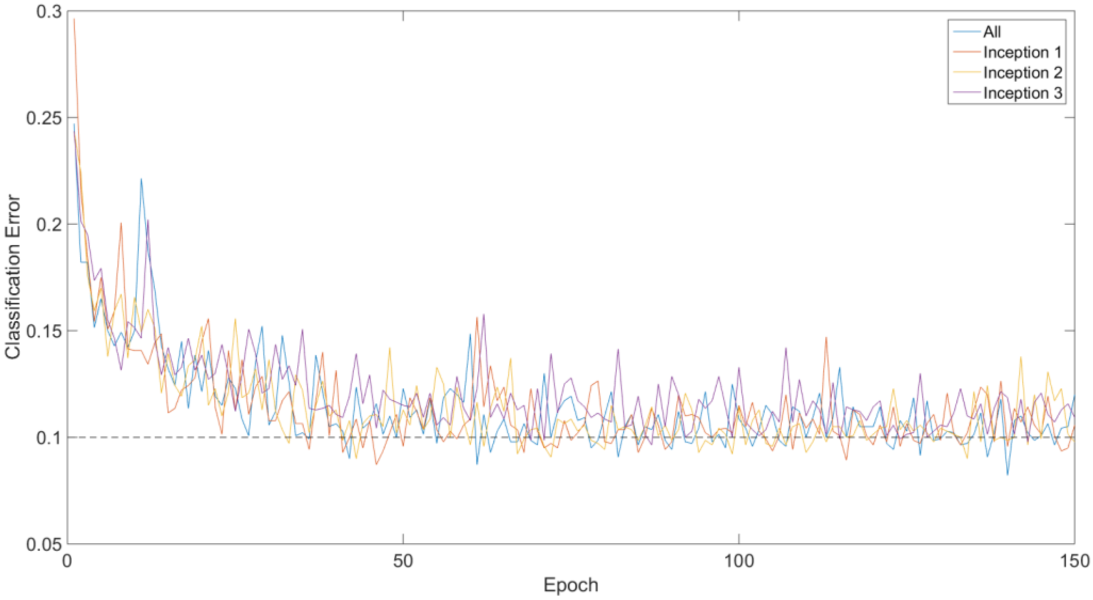
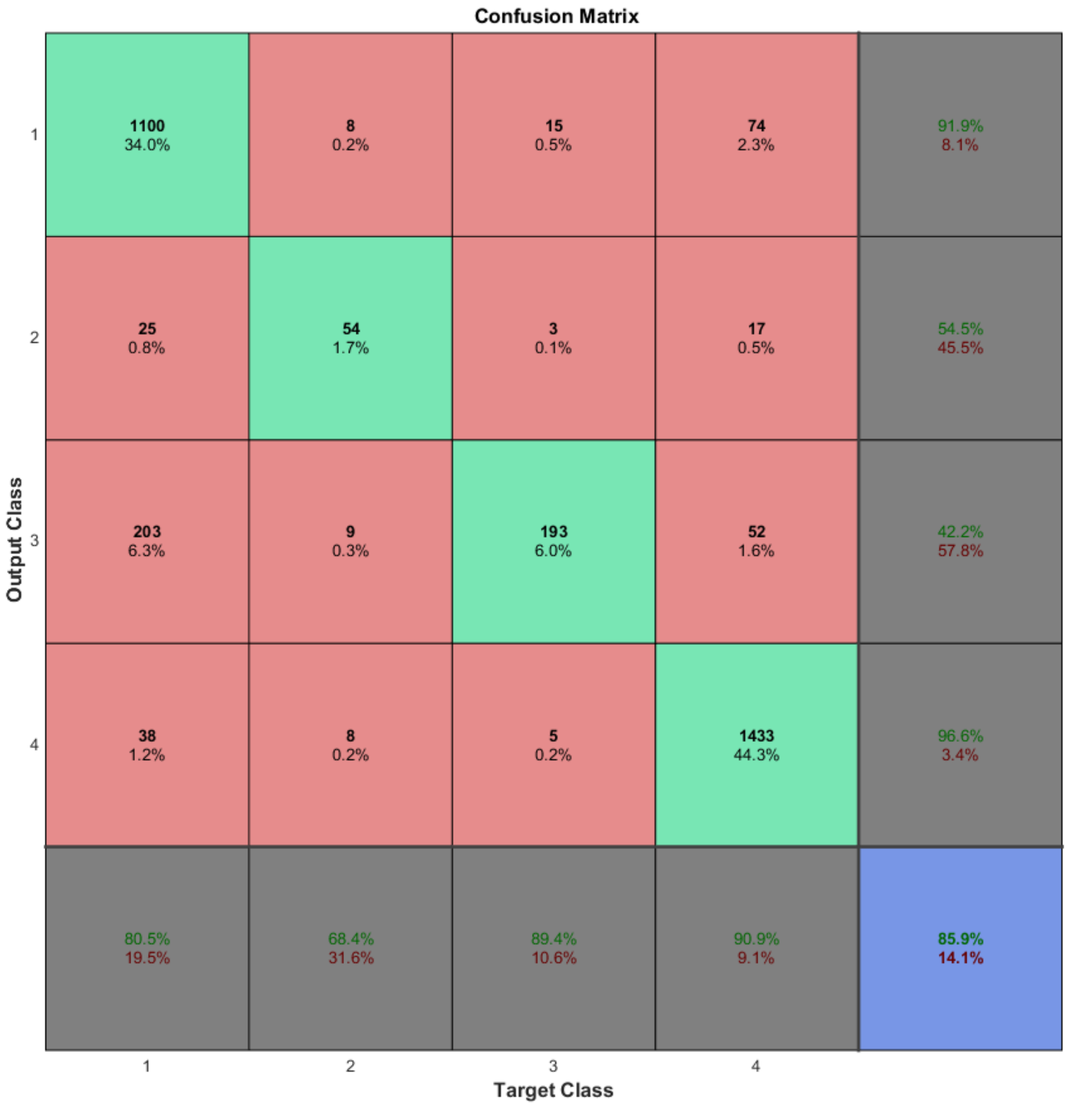


Figure 4.9: For epochs 0-150, the validation classification error for the best four CNN models: All, Inception 1, Inception 2 and Inception 3. The dashed line indicates a classification error of 0.1.

From the traces in Figure 4.9 it is clear that the validation classification error is very noisy; a 5% change can be observed between epochs. This is a result of a relatively small validation set and the similarity between the classes. Three of the classes are similar in the fact they contain parts of one or more hockey players. This means they share similar textures and colours and that the difference between them is very subtle. It follows that a small change in parameters may shift the boundary between classes a sufficient amount to have this impact on the classification error.

Despite the four models all giving similar validation classification error, assuming only one can be used in the final algorithm, it is logical to use the model and epoch that gave minimal validation classification error. Epoch 140 of the **All** model gave an error of 8.2%. It is this model which will be used in the algorithm but first its performance on the test set will be assessed.

Figure 4.10 displays the Confusion Matrix for the test set. Overall the model achieved a classification error of 14.1%. This is inferior to the 8.2% observed on the validation set. The superior validation performance may be due to this specific model by chance over-fitting to the noise in the validation data. As noted previously the validation error across epochs is noisy due to the size of the dataset. The test set does not have the same noise in its distribution and subsequently is not as accurate.



Confusion Matrix

Output Class \ Target Class	1	2	3	4	
1	1100 34.0%	8 0.2%	15 0.5%	74 2.3%	91.9% 8.1%
2	25 0.8%	54 1.7%	3 0.1%	17 0.5%	54.5% 45.5%
3	203 6.3%	9 0.3%	193 6.0%	52 1.6%	42.2% 57.8%
4	38 1.2%	8 0.2%	5 0.2%	1433 44.3%	96.6% 3.4%
	80.5% 19.5%	68.4% 31.6%	89.4% 10.6%	90.9% 9.1%	85.9% 14.1%

Figure 4.10: The confusion matrix for the test dataset.

Alternatively this may be due to the formulation of the datasets. The training set and validation set were formed from sampling different segments of the same five videos. Therefore while independent across time, the training set and validation set are dependent across the teams represented and the environmental conditions. The test set was formed from five independent videos, which had different teams and environmental conditions. Consequently the difference in performance could be due to the model learning characteristics specific to the training/validation videos, i.e. the model has learnt to classify a player in that set of videos rather than a generic player.

This form of overfitting was not considered prior to the experiment but suggests that better results may be expected with a larger and more diverse dataset.

The confusion matrix also suggests that the model tends to over-predict classes 2 and 3. Only 54.5% of those predicted as class 2 are correct, this drops to 42.2% for those predicted as class 3. Similarly, only 68.4% of those blobs that should have been class 2 were predicted correctly. This is further support for the fact that the model struggles to separate the classes.

While the best model achieved good accuracy, further accuracy may be achieved by the following:

- **Hyper-parameter optimisation** (Bergstra & Bengio 2012)**.** The update algorithm, adam, has four hyper-parameters that affect the learning process. Not all sets of hyper-parameters will find the optimal solution. Hyper-parameter optimisation attempts to find the hyper-parameters that find the optimal solution in as shorter time as possible.
- **Use the average probabilities from an ensemble of models**. An ensemble (Dietterich 2000) is the combination of a number of classifiers to increase the overall classification performance. (He et al. 2016) show that the ILSVRC classification error can be decreased by predicting the class from the mean probabilities of a number of independently trained models. The four models **All**, **Inception 1**, **Inception 2** and **Inception 3** are good candidates to form an ensemble. Yet, inferring the class with multiple models does add to the computational cost at runtime.

- **Use the average probability for augmented images**. This takes advantage of the fact that certain transformations can be applied to an image without changing its class. The image is augmented a number of times with transformations such as a reflection about the y-axis. The mean class probabilities across the augmented images are more likely to correctly classify the image. Similar to the ensemble of models, this improvement requires multiple inferences at runtime.

4.4 Summary

This chapter addressed the objective: *"Train a Convolutional Neural Network to classify the contents of a blob"*. To do so, it investigated the performance of two different methods for the transfer of an existing Convolutional Neural Network (CNN) to the task of player blob classification. The existing CNN used the GoogLeNet architecture and had been trained on the ImageNet dataset. The performance was assessed using the classification error.

The first method, labelled Feature Extractor, used a feature vector extracted from GoogLeNet to train a Support Vector Machine (SVM) for the classification task. A classification error of approximately 20% was achieved when using a feature vector extracted after **Inception 3** or **4**.

The second method fine-tuned the parameters of GoogLeNet for this specific classification task. The fine-tuning method outperformed the Feature Extractor method. A validation error of 8.2% was achieved when the model was allowed to adapt the entire network. This model gave a test set classification error of 14.1%. Part of this error is accounted for by the similarity between three of the classes and may be resolved with more data.

This is a novel use of a convolutional neural network however it did not exploit any hockey specific knowledge; therefore this model could be used to classify player blobs in any other field sport. However, better results should be expected if a model is trained with domain specific data. While the technique could also be used to classify players in non-field sports, the increased image resolution of the players may mean the data has different characteristics and the results may not be comparable.

The trained model can be used as a filter to eliminate non player blobs, thereby reducing the noise in the set of player coordinate extractions. Further to this, if groups of blobs rather than single blobs are classified using the model, then it can be used to reform over-segmented blobs into players. Chapter 10 outlines the method to do this as part of an investigation into the accuracy with which player coordinates can be extracted; however the next five chapters investigate the sub-algorithm Reconstruct World Points.

5 World Point Extraction

5.1 Introduction

As highlighted in the literature review, the reconstruction error is assessed by measuring the difference between a set of known world points and the corresponding image points reconstructed on to the calibrated plane. This requires the accurate extraction of the known world points corresponding image points. For a given camera pose each known world point, W_a, has an exact correspondence with a unknown image point, I_a. The extracted image point, I_e, is the measured image point of the unknown world point, W_e. Unless I_a and I_e are equal, there will be some difference between W_a and W_e, which adds uncertainty to the reconstruction error.

There are two possible sources of uncertainty when assessing the reconstruction error. The first source of uncertainty is due to the misplacement of the world known points. By machine manufacturing the calibration object, the world known points are marked with high accuracy and errors due to world point misplacement are minimal. The second source of uncertainty is due to the accuracy with which the image position of a world known point can be extracted. Typically, when assessing the reconstruction accuracy, the known world points are either marked by a grid of checkerboard intersections or a grid of circles. Due to the camera's wide angle of view and relatively high angle of incidence to the plane it is unknown a priori if either of the methods of demarcation are suitable for the expected camera pose. Therefore this chapter investigates which of these two methods is suitable to extract known world points for a frame captured at the expected camera pose. This method will be used in future chapters when the reconstruction accuracy is to be assessed.

This chapter addresses the objective: “*Determine an accurate method to extract planar world points from a frame captured at the expected camera pose*.” The novel contribution to knowledge for this chapter is a method to extract image known points when using a camera with a high angle of incidence and a wide angle lens. To assess the accuracy of this method, a pattern was designed that allows a known world point to be visually resolvable at a range of different planar resolutions.

5.2 Data Collection

In this and the following camera calibration chapters assessments are made using a 1:100 scale model of a hockey pitch. This allows:

- For machine accurate placement of known points.
- The testing of camera poses that are not accessible in the available hockey stadia.

All results presented are relative to the size of the scale model. The scale 1:100 was chosen as this could be produced on an A0 (841 mm x 1189 mm) rigid board.

A test plane for each of the board types was created and printed onto a flat board. The checkerboard plane was 0.93 m x 0.57 m with 0.03 m squares as in Figure 5.1A. The checkerboard intersections, which are dimensionless points, provided a set of 540 known world points on the plane (Figure 5.2A).

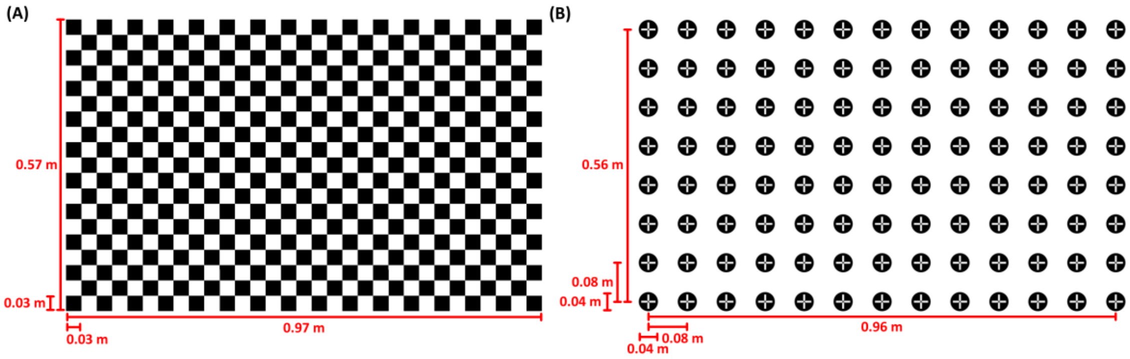


Figure 5.1: (A) The checkerboard plane. (B) The circle plane.

The circle plane was 0.96 m x 0.56 m (Figure 5.1b). In both dimensions, at intervals of 0.08 m, modified circles (Figure 5.2B) of diameter 0.04 m formed a 13 x 8 grid, the centres of which denoted 104 known world points. Unlike a checkerboard intersection, a circle is not a dimensionless point so remains resolvable at a lower planar resolution, the number of pixels per metre (pixels/m). A world known point can be inferred as the centre of the circle, however due to lens distortion and the perspective transformation the centre of the projected circle is not the true centre of the circle. The circle must be modified so its true centre can be determined in the projected image. The novel pattern in Figure 5.2 was designed to make the circle centre visually resolvable at a range of different planar resolutions. The 0.3 mm white dot provides an accurate location for the centre of the circle at a high planar resolution, the 1 mm black diagonal cross an estimation of the centre of the circle at a lower planar resolution and the 4 mm white cross an estimation at an even lower planar resolution.

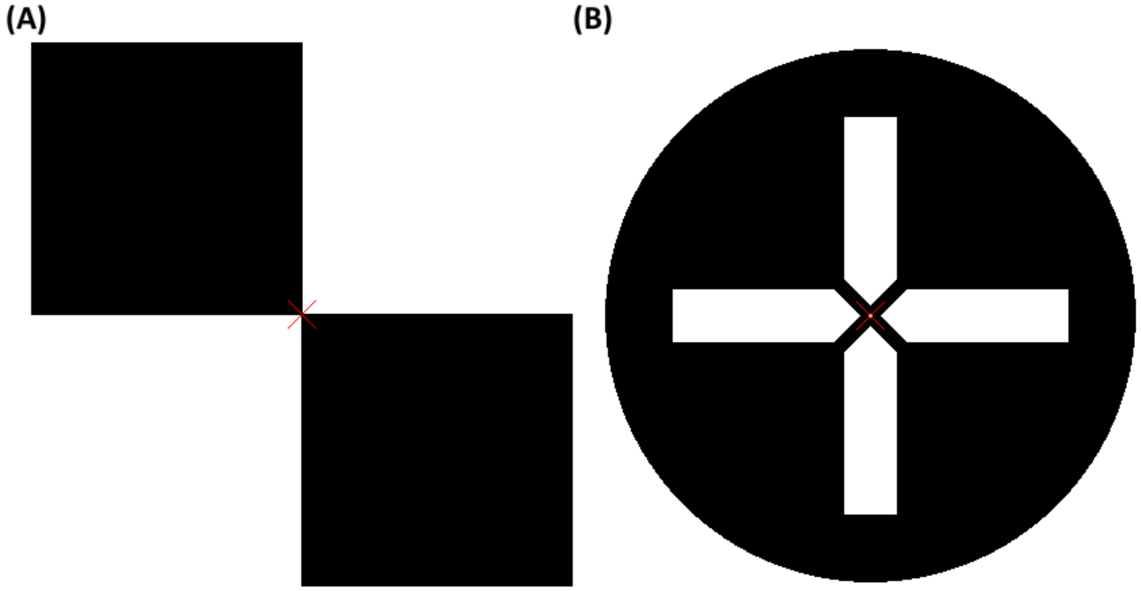


Figure 5.2: The two patterns used to define the known points on the two test planes. The red cross marks the known point relative to the pattern. (A) A checkerboard intersection as used on the checkerboard plane. (B) The novel pattern used on the circles plane. At a high planar resolution the circle centre is denoted by the white dot. At a lower planar resolution the circle centre can be estimated by the black diagonal cross and at an even lower planar resolution the circle centre can be estimated by the broad white cross.

The image collection procedure for each test plane was as follows:

i. A video of the plane was captured using a Sony FDR-AX33 mounted with a 0.28x Raynox HDP-2800ES wide-angle lens converter. The video had a resolution of 3820 pixels x 2180 pixels (4K). 4K was the highest available image resolution and as such gave the highest available planar resolution. The camera was positioned at the midpoint of the long dimension of the test plane, 0.15 m back from the test plane and elevated by 0.08 m (Figure 5.3). Assuming viewing straight up as zero rotation, it was rotated 110° about the x-axis and 0° about the y and z axes. The camera was zoomed to maximise the plane in the field of view and then focussed manually to achieve focus across the entire board.

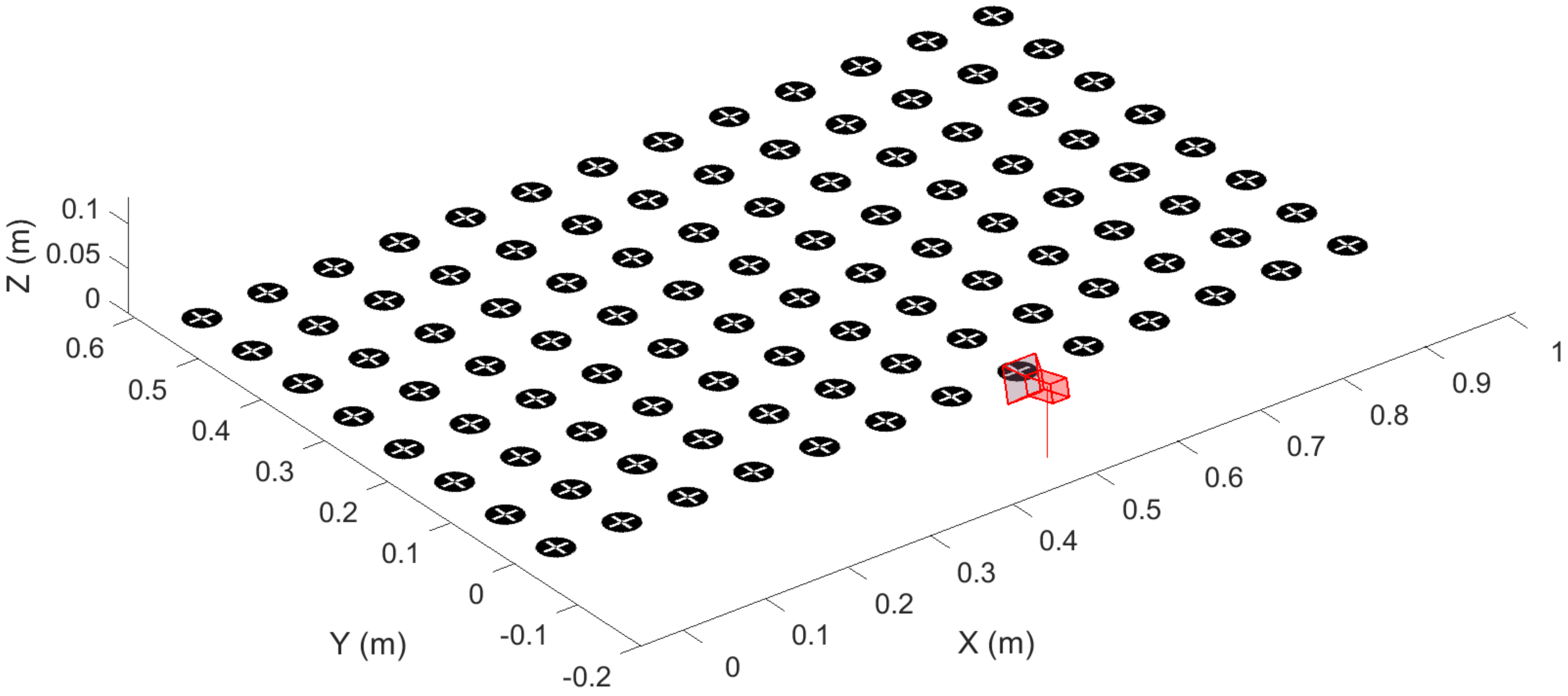


Figure 5.3: The camera position relative to the circle model plane.

ii. A single frame, the plane frame, was extracted from this video.

iii. Without changing the zoom or focus of the camera, a video of a planar calibration board in different orientations was captured. The calibration board was an 8 square x 8 square, 30 mm checkerboard, to give a total of 49 intersections. The board was orientated through a range of different angles and translations to achieve full checkerboard intersection lens coverage.

iv. Frames were extracted from the calibration video at 1 Hz to form the set of potential calibration frames.

5.3 Checkerboard

Figure 5.4 displays the plane frame for the checkerboard condition. In regions of high planar resolution the checkerboard intersections are well defined points; however the cut-out illustrates that in areas of low planar resolution the checkerboard intersections do not appear as resolvable points. Here the points are blobs of high or low intensity pixels. The Harris corner detector algorithm, employed to find checkerboard intersections, finds well-defined corners in the image and as such no corner is found. This issue cannot be resolved by increasing the size of the checkerboard squares. The checkerboard intersections are dimensionless points, the ability to resolve them is solely dependent upon their planar resolution. As the points are unresolvable a checkerboard is not a suitable method to indicate the known planar points when capturing from the required camera pose.

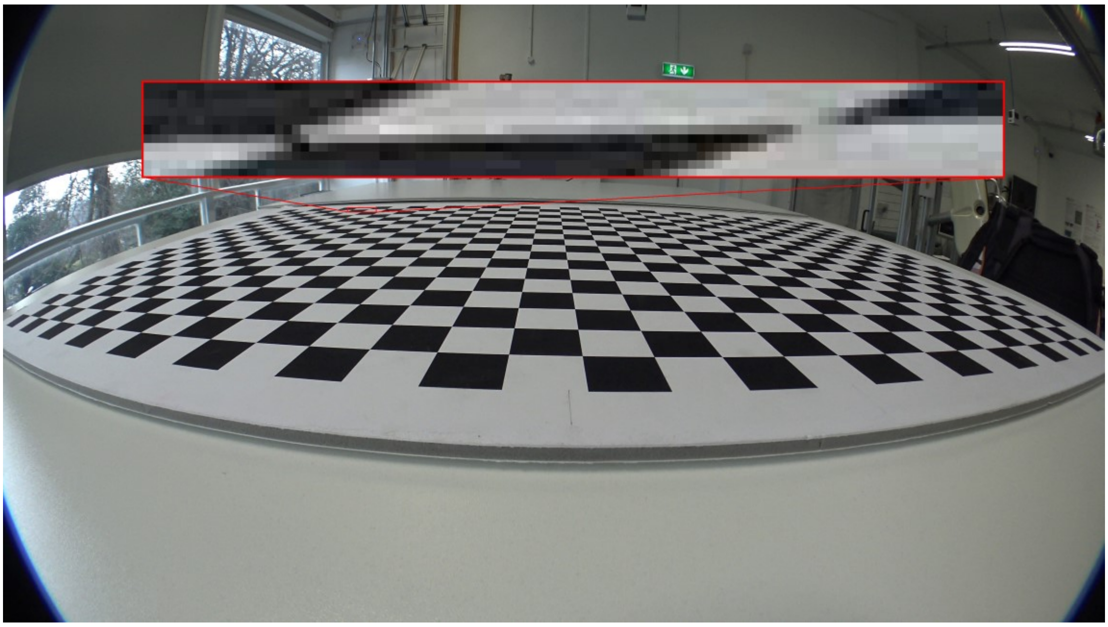


Figure 5.4: Plane frame for the checkerboard condition. The cut-out displays two checkerboard intersections neither of which are sharp, well defined points, but instead are blobs of pixels of low and high intensity.

5.4 Circles

Figure 5.5 illustrates that the circles are also susceptible to diminishing resolution due to the perspective of the camera; however even with this effect they remain visible in the most extreme regions of board's planar resolution. This suggests that circles have the potential to be a suitable method for demarcation of the known world points at the expected camera pose. Subsequently an accurate method is required to extract the circle centres from the image. The following two sections consider a manual process and an automatic process for the extraction of the circle centres.

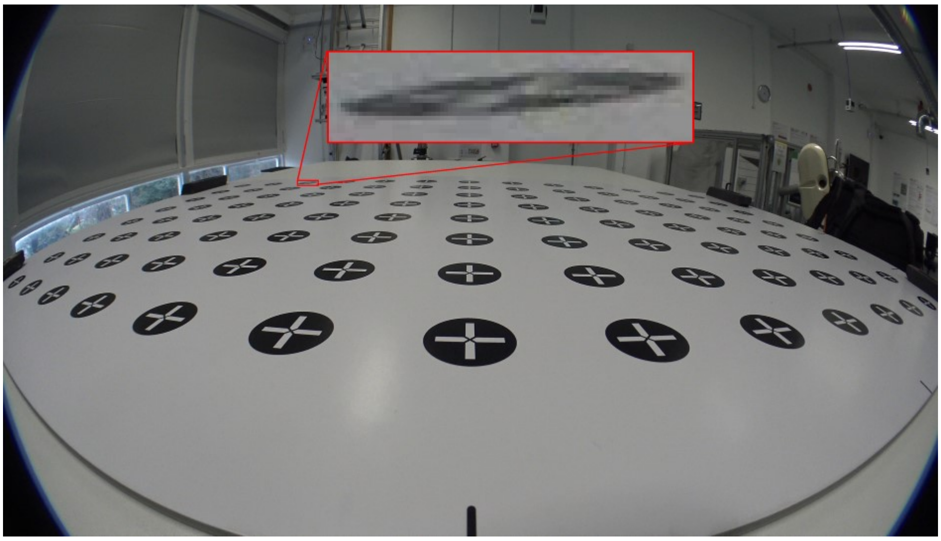


Figure 5.5: Plane frame for the circles condition. The cut-out displays the circle with a similar coordinate to the checkerboard intersections highlighted in Figure 5.4. The circle centre can be inferred from the white cross however the black diagonal cross and the white are unresolvable due to the low planar resolution.

5.4.1 Manual Extraction

The modifications made to the circles mean their true geometric centres are indicated in the image. It follows that the grid of known world points can be created by clicking each of these circle centres. This simple procedure has two issues:

- Identifying the centre of a circle in the image is a subjective judgement which is likely to display variance over repeated clicks.
- As highlighted in the cut out in Figure 5.5, the white cross added to the circles estimates the centre, however the planar resolution is insufficient to accurately resolve the more accurate black cross or white dot. This makes it difficult to identify I_a which increases the uncertainty. Due to the camera pose restriction, this can only be resolved by increasing the camera resolution, not possible with the available hardware.

Each of these issues may mean that I_e does not equal I_a, however the effect of this error is not consistent across the plane. It is clear that a one pixel error will give a larger difference between W_a and W_e at a lower planar resolution. Consequently the following two ideas were investigated:

- Does the planar resolution affect the variance of the clicked point?
- How does a one pixel difference affect the reconstruction across the calibrated plane?

The analysis procedure was as follows:

Camera Intrinsic Parameter Estimation

i. For each potential calibration frame, the 49 calibration points, the checkerboard intersections, were extracted. Those frames for which the calibration points could not be extracted were disposed of. The extractions from the remaining frames formed the set of point configurations. There were 40 point configurations.

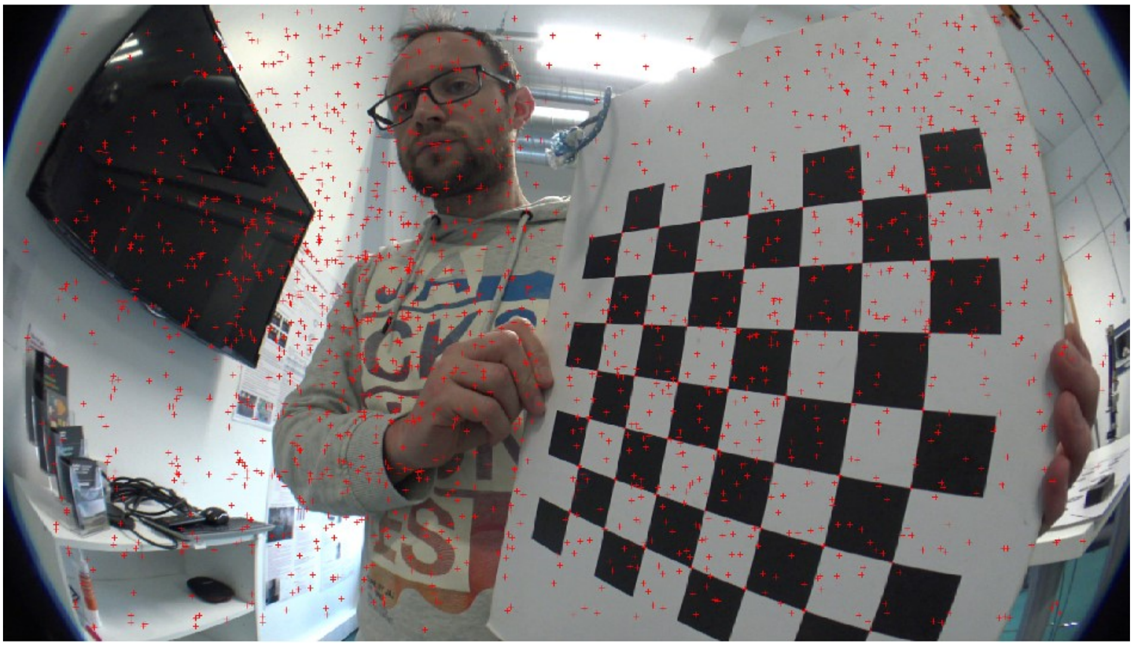


Figure 5.6: One of the calibration images overlaid with the set of point configurations.

ii. Bouguet's toolbox was used to calculate the camera's intrinsic parameters using the set of point configurations. The fisheye camera model presented in (Kannala & Brandt 2006) was used. The following parameters were optimised: the focal length, the principal point and the 4 term radial distortion model. The standard deviation of the pixel error was: x = 1.42 pixels, y = 1.33 pixels

Circle Centre Clicking

i. For five repeats each circle centre was clicked. 24 hours was left between each repeat to limit any effect due to memorising the point that was clicked.

ii. The mean and standard deviation for each of the 104 circles was calculated.

Camera Extrinsic Parameter Estimation

i. The transformation from image points to world points was estimated using the method proposed in Chapter 6. All 104 circles were used as control points, with the known image points being the set of mean clicked values.

Calculate the Metres per Pixel

i. For each circle:

   a. The mean u value was rounded up and rounded down to give two values with a one pixel difference. These were combined with the original v value to give a pair of coordinates $c_1 = (u_{up}, v)$ and $c_2 = (u_{down}, v)$.

   b. c_1 and c_2 were reconstructed using the camera model.

   c. The effect of a single pixel change in u was then calculated as the vector $c_2 - c_1$.

ii. This procedure was repeated for the v value to give two grids of two-dimensional vectors.

Figure 5.7 displays the standard deviation in the u and v dimensions for each of the clicked points. The standard deviation is similar across the entire plane in both u and v, i.e. the clicker's precision is similar irrespective of the coordinate on the plane. This suggests that the clicker is able to perceive a circle centre with the same consistency, even at a low planar resolution. However no conclusion about the effect of planar resolution on accuracy of the click can be made because the true circle centre is unknown, i.e. the clicker's perceived circle centre may not be the true circle centre.

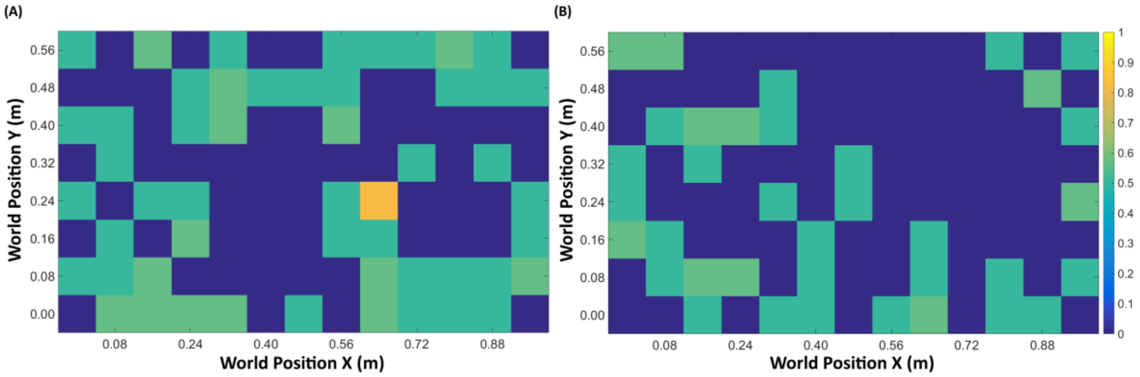


Figure 5.7: For each of the world known points, indicated by their (X, Y) position on the calibrated plane, the standard deviation in pixels of the clicked circle centres in: (A) the u image dimension and (B) the v image dimension.

Figure 5.7 displays the effect of a one pixel difference on the reconstructed coordinate for each of the circle centres. The colour of a cell indicates the magnitude of the difference while the arrow indicates the relative magnitude and direction of the difference vector. Figure 5.8A displays the effect of a one pixel difference in the u dimension. Ignoring the barrel effect of lens distortion, the u dimension of the image is parallel to the X dimension of the calibrated plane. This is reflected in the direction of the difference vectors which have a major component along the X dimension. It also means that the magnitudes of the differences in u are much smaller when compared to the differences in v. The magnitude of the difference increases further from the camera, however this effect is much more pronounced with a one pixel change in the v dimension (Figure 5.8B). At the most extreme point a one pixel difference in the v dimension equates to more than a 0.005 m resultant difference. This means a single pixel accounts for as much as 1/192th of the entire dimension of the board model. This large value illustrates the potential for a small pixel difference resulting in a large distance between W_a and W_e, and the necessity for sub-pixel accuracy for I_e.

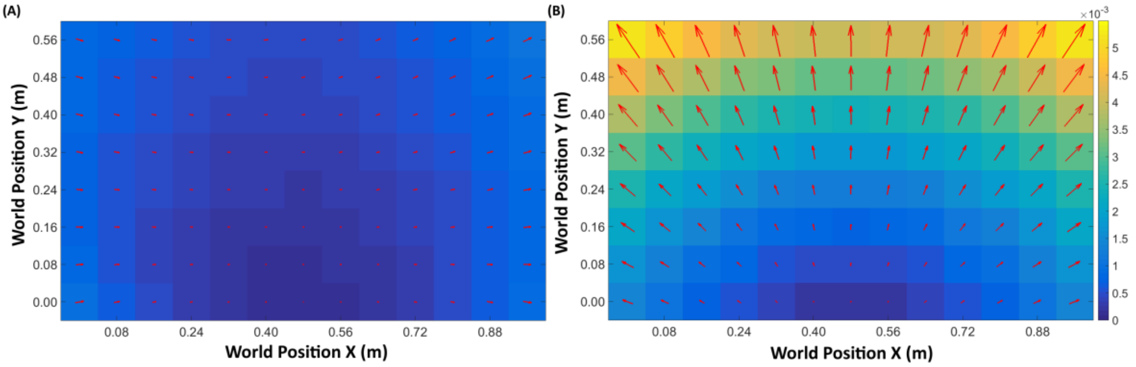


Figure 5.8: For each of the world known points, indicated by their (X, Y) position on the calibrated plane, the effect of a one pixel difference on the reconstructed coordinates in metres. (A) A one pixel difference in the u dimension. (B) A one pixel difference in the v dimension. The colours indicate the absolute magnitude of the difference. The arrows indicate the direction and the relative magnitude of the difference.

The calculation of the metres per pixel requires the estimation of an extrinsic model. This estimation is made using the clicked image points, so any errors are propagated through to the model. This error is somewhat mitigated by the using all 104 circle centres as control points; each control point has less of an effect on the overall solution. It is also somewhat irrelevant as Figure 5.8 is only meant to be indicative of the range in planar resolution across the calibrated plane.

As illustrated in Figure 5.7, the user does not consistently click the same point across repetitions. This suggests some difficulty in perceiving the circle centres and results in error in the ground truth points. As it has been identified that a single pixel can have a large impact on the world position, it is inappropriate to use the clicked points when assessing the reconstruction error. The following section proposes an automatic extraction process that uses the geometry of the circles on the plane. It does not require the exact circle centre to be resolvable and calculates the centres at the subpixel level.

5.4.2 Automatic Extraction

The automatic method to extract the centre of the circles used the common tangency of neighbouring circles (Mateos & Tsai 2000). To determine neighbouring circles the grid of known world points was projected onto the image plane. This required an initial estimate for the extrinsic parameters. The extrinsic parameters were calculated using the 6 control points illustrated in Figure 5.9.

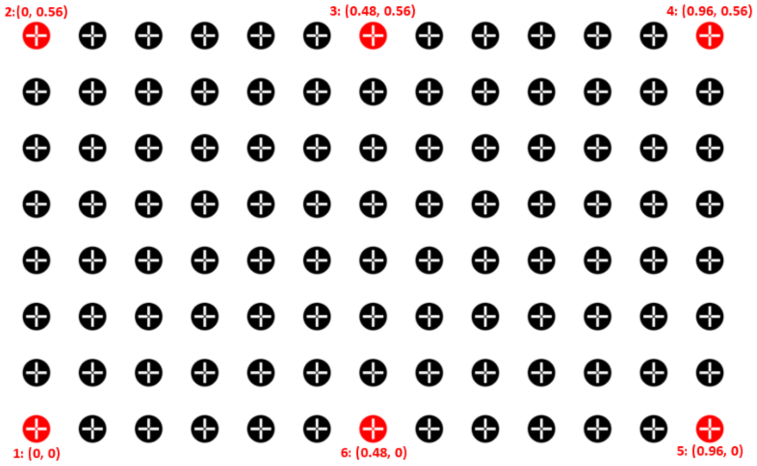


Figure 5.9: The six control points used to calculate the extrinsic parameters.

The extraction of the centre of the circles was broken into the following sub-algorithms:

A. *Initial Circle Centre Estimation*

i. The plane frame was masked to the area of interest, thresholded and then inverted to ensure the projected circles had a strong contrast against non-circles.

ii. An image fill was applied to each of the projected circles to ensure they were solid white blobs. This resulted in the blob frame.

iii. An estimate for the centre of each circle was found in the blob frame as the centre of the blob. This formed the set of initial circle centres (Figure 5.10).

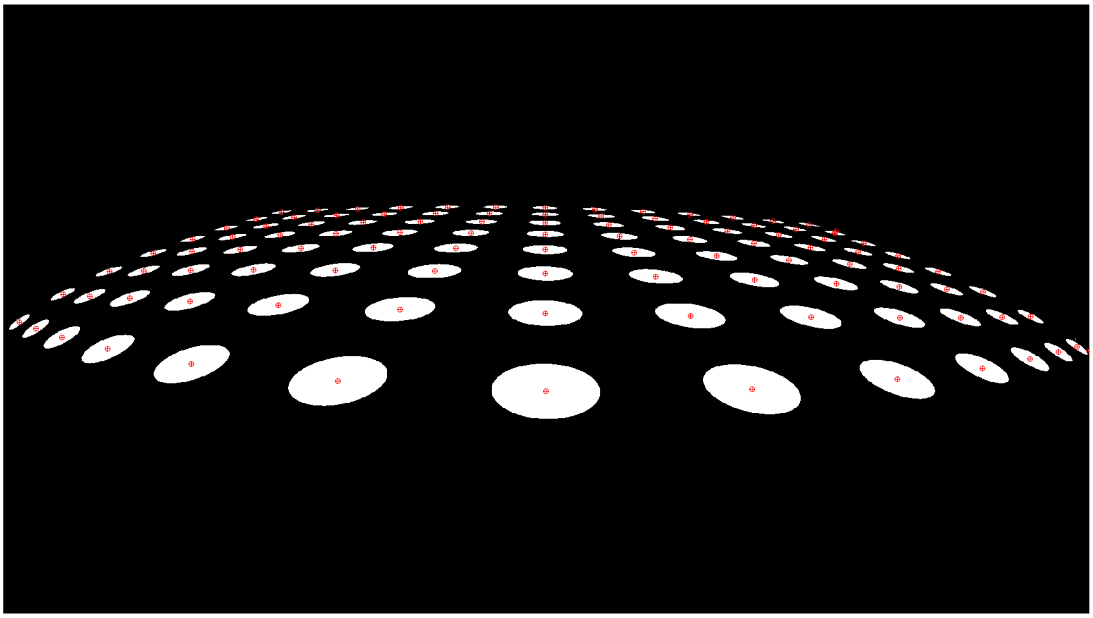


Figure 5.10: The blob frame overlaid with the initial circle centres.

B. *Camera Extrinsic Parameters Estimation*

i. In the plane frame, the 6 control image points were clicked in ascending order to give the set of estimate control image points.

ii. For each of the estimate control image points the nearest initial circle centre was estimated by minimising the Euclidean distance. This gave the set of correspondences between the control point's world coordinates and image coordinates.

iii. Bouguet's camera calibration toolbox, along with the intrinsic parameters found in Section 5.4.1 and the 6 control points, was used to estimate the extrinsic parameters. The estimated extrinsic parameters were used to project the grid of known world points onto the distorted image plane. This gave the projected grid.

Mapping Circles to the Projected Grid

i. The projected grid was undistorted to give the undistorted projected grid.

ii. The intrinsic parameters were used to create the undistorted blob frame from the blob frame (Figure 5.11).

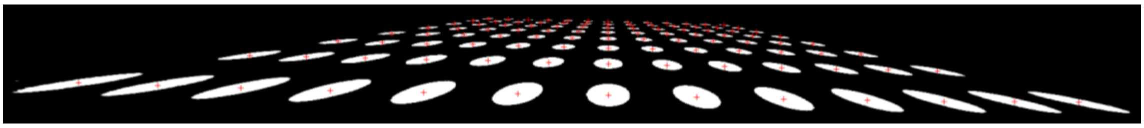


Figure 5.11: The undistorted blob frame, overlaid with the initial guess for the circle centre, which is the centre of the blob.

iii. An estimate for the centre of each circle was found in the undistorted blob frame as the centre of the blob.

iv. For each point in the projected grid the nearest estimate circle centre was found by minimising the Euclidean distance. This gave the mapping necessary to determine which circles were neighbours of one another.

Estimate Circle Centres

i. An ellipse was fitted to each segmented blob.

ii. For each ellipse:

   a. For each of its 8-connected neighbouring ellipses, the two common external tangents were calculated using the method in (Mateos & Tsai 2000). For the edge ellipses the reduced set of neighbours was used.

   b. Each point of tangency was classified by the direction on the model board of the neighbour used to calculate it. The classes were: Left-Right, Up-Down, Diagonal Up-Down or Diagonal Down-Up. Figure 5.12A illustrates which class each neighbour belongs to. Figure 5.12B visualises the points of tangency for one of the circles.

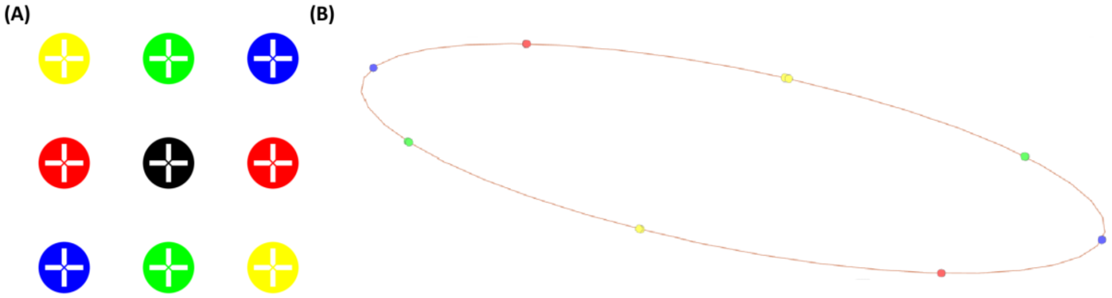


Figure 5.12: The points of tangency for one of the circles. (A) The neighbours that form each of the classes: Red – Left-Right, Green – Up-Down, Yellow – Diagonal Up-Down and Blue – Diagonal Down-Up. (B) The corresponding points of tangency for a circle in the segmented image. The fitted ellipse is indicated in orange. The colour scheme is the same as in (A). Note: i) The points of tangency for each class form a line perpendicular to the class label. ii) The upper two yellow points are not collocated suggesting error in one or both of the ellipses used to form these points of tangency. This may be due to residual distortion in the undistorted image or the circle segmentation.

   c. Linear regression was used to fit a straight line to the points in each of the classes. This line was perpendicular to that suggested by its class label.

d. Least squares minimisation was performed to find the intersection of the four lines. This point is the centre of the circle (Figure 5.13).

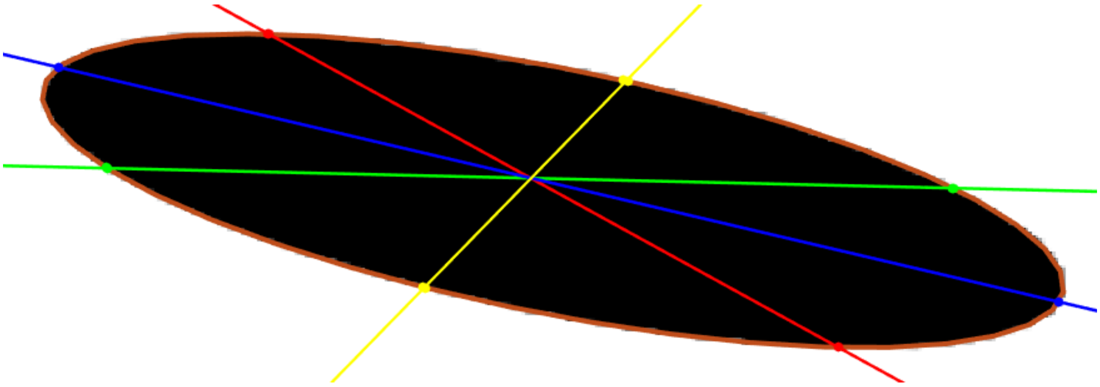


Figure 5.13: The circle centre is found as the intersection of the four lines fitted to the points of tangency. The circle is the same as that in Figure 5.12B but the ellipse has been filled with black to improve the visualisation of the lines.

iii. The distortion model was used to transform the circle centres (Figure 5.14) to plane frame points. This gave the set of refined circle centres (Figure 5.15).

As the true centre of the circle is unknown, the accuracy of the circle centre extractions cannot be assessed objectively. Subsequently there may be an unquantified error between the true centre of a circle and the extracted centre of a circle. This error would partially explain the errors in the reconstruction accuracy computed in future chapters.

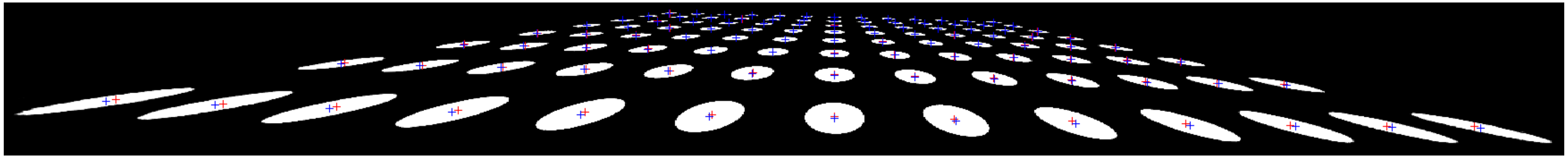


Figure 5.14: The circle centres extracted from the image overlaid on the undistorted blob frame. The blue crosses mark the initial estimate for the centre of each circle, given as the centre of the blob. The red crosses mark the refined circle centres.

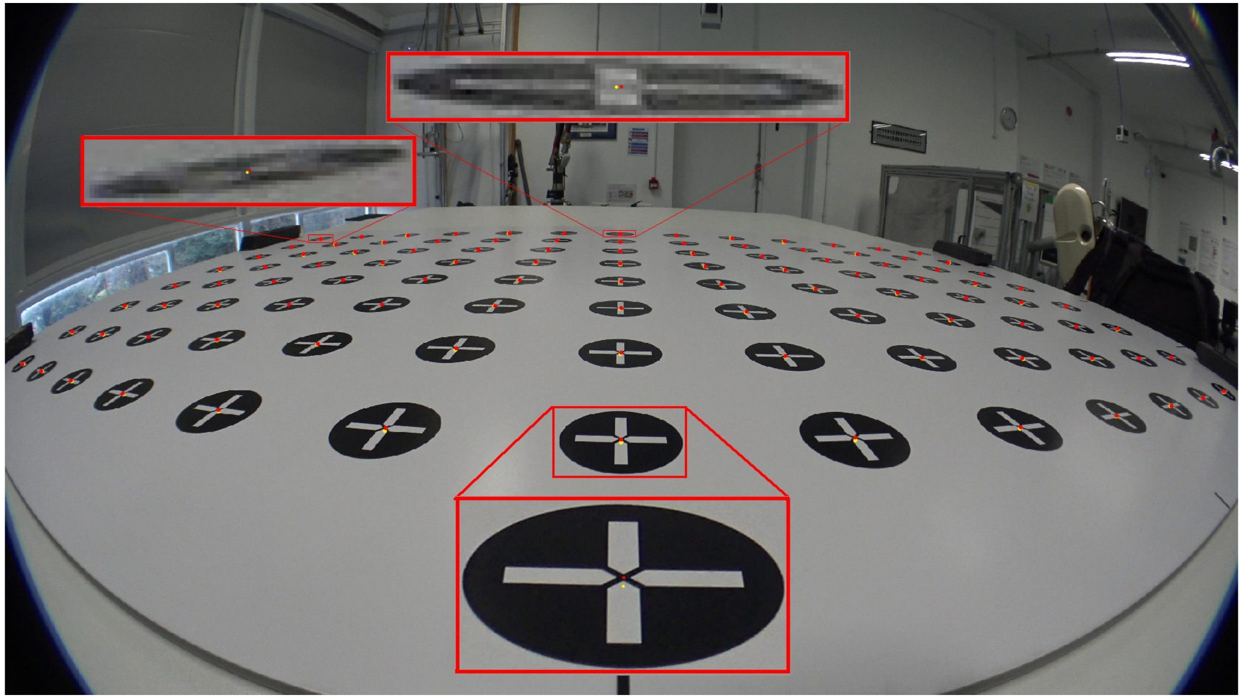


Figure 5.15: The plane frame overlaid with: Yellow Dots – The initial circle centres. Red Dots – The refined circle centres. The cut-outs magnify three of the circles.

Instead the circle centres were assessed qualitatively. A visual comparison can be made between the extracted circle centres and the true centres indicated by the circle modifications. The cut-outs in Figure 5.15 magnify three of the circles. The yellow dots indicate the initial circle centres, the red dots the refined circle centres. There is a systematic error in the initial circle centres. As noted earlier, this is a result of the perspective transformation and the lens distortion. The perspective transformation means the planar resolution decreases from the front of the circle to the back of the circle. This inconsistency means the centre of the projected circle is not the centre of the circle, but instead is shifted towards the camera.

The lens distortion increases as the radial distance from the principal point is increased. Points of the circle further from the principal point will be more distorted than those closer to the principal point. This results in a compression of the blob in the direction of the principal point, which shifts the blob centre towards the principal point. As the blobs are small the intra-blob variation in distortion is small and this shift is relatively minor when compared to the shift due to the perspective transformation. The initial circle centres do not provide accurate circle centres and should not be used as known image points.

Figure 5.15 also displays the refined circle centres. For each circle the refined circle centre is aligned with the centre of the modified circle. It can be observed that for the circle closest to the camera the refined circle centre falls within the 0.3 mm white dot that denotes the centre. For the other two cut-outs, which have much lower planar resolution, the white dot falls within the white cross, visually close to the expected centre. This suggests that the method provides an accurate way of extracting the image coordinates of the known points on the test plane at the expected camera pose.

Due to the structure of the grid, the expected world coordinates of the points of tangency are known. As each circle has eight distinct points of tangency, it follows that the number of the world known points could be increased by using the points of tangency rather than the circle centres. Yet Figure 5.12B illustrates that the points of tangency are not errorless. For example, the upper two yellow points are not collocated suggesting error in one or both of the ellipses used to form these points of tangency. The small error in the ellipses may be due to poor image un-distortion due to ill-fitting intrinsic parameters or poor circle segmentation. Instead the proposed method uses the redundancy of multiple erroneous points to formulate a better estimate of the circle centre.

The structure of the grid also means that a single straight line should bisect the centre of each circle in a row, column or diagonal. Given this assumption, a single line could be fitted to all the points of tangency in a row, column or diagonal. With more points of tangency it could be supposed that the line is a better fit as it is less subject to error in any single point of tangency. However, due to an imperfect intrinsic model, the undistorted image plane is not rectilinear and the expected straight lines appear as curves. Calculating a line local to each circle reduces the curve to a short segment which can be better represented by a straight line.

Similarly for each circle, the number of points of tangency used to formulate a line could be increased by using all the circles in a line direction rather than just the direct neighbours. Again this method is unsuitable due to residual distortion in the undistorted image plane.

The proposed method requires accurate segmentation of the circles on the plane. Poor segmentation can be lead to inaccurate ellipses, which in turn leads to high variance in the points of tangency and errors in the circle centres. In areas of low planar resolution, accurate segmentation is complicated by the modifications made to the circles; therefore in subsequent chapters filled circles will be used.

It should be noted that the proposed automated method is not error free. Any errors in the method are subject to the same metres per pixel error presented in Figure 5.8.

5.5 Summary

This chapter addressed the objective: "*Determine an accurate method to extract planar world points from a frame captured at the expected camera pose*." To do so, it considered using checkerboard intersections and circles to indicate the known points on the plane.

Checkerboard intersections are unsuitable to indicate the known points on the plane. At low planar resolution the intersections are unresolvable and as such the known points cannot be accurately extracted.

As circles have a dimension they remain resolvable at a lower planar resolution. The known point can be inferred as the centre of a circle; however due to the perspective transform and lens distortion the centre of the projected circle is not the true centre of the circle. Manually clicking the centre of the circles is unsuitable because: 1. the low planar resolution makes the precise centres unresolvable, and 2. human error leads to variance in the point clicked. Therefore, a novel automatic method of extraction was presented that finds the true centre of the circle in the image.

The true image points of the circle centres are unknown; therefore the circle centres could not be objectively assessed. This is a fundamental limitation of the work in this chapter and a potential source of error in the assessment of the reconstruction in future chapters. Instead the true centres were assessed qualitatively using a novel pattern that means the world known points are visually resolvable a range of different planar resolutions. The true centres displayed good alignment with the expected coordinates.

Using a grid of circles and the method presented allows accurate extraction of known image points from the expected camera pose. This method will be applied in the future camera calibration chapters to extract the known image points.

6 Planar Reconstruction

6.1 Introduction

As highlighted in the literature review, image reconstruction is the transformation of image plane coordinates into world coordinates. Following camera calibration it is possible to determine the ray that a point falls on; however as depth information is lost during the imaging process, it is impossible to determine the distance to the object. However, if it is assumed that all points lie on a calibrated plane, the coordinate of the point is then the intersection of the ray and the plane. This is the process of planar reconstruction.

Extrinsic camera calibration is the process of estimating the pose of the camera relative to a world coordinate system. In the case of planar reconstruction this is the camera's pose relative to the plane. The classic camera calibration method of (Heikkila & Silven 1997) estimates the projection matrix $P_{projection}$ that transforms from world coordinates to camera coordinates. $P_{projection}$ is estimated by minimising the projection error, the distance between a set of world coordinates projected into image coordinates and their expected coordinates. The planar reconstruction is then performed using the inverse projection matrix, $P_{projection}^{-1}$. This inverse projection will only be optimal for reconstruction if the camera plane and the calibrated plane are parallel. When parallel, the planar resolution, the number of pixels per meter (pixels/m), of the calibrated plane is consistent across the whole plane. If each control point has minimal projection error and the same planar resolution, then $P_{projection}^{-1}$ must also be optimal. Yet when the calibrated plane is captured from a perspective view, the planar resolution is no longer consistent. Consequently $P_{projection}^{-1}$ biases the reconstruction towards those control points with a higher planar resolution. At a high range of planar resolution this results in poor mean reconstruction accuracy.

Alternatively, the projection matrix, $P_{reconstruction}$, that transforms from image points to world points can be estimated directly by minimising the reconstruction error, the mean distance between a reconstructed point and it's expected coordinates. Traditionally this estimation is not performed because of errors introduced in the image point extraction process. As demonstrated in the literature review these errors propagate through the minimisation procedure, causing errors in the planar reconstruction.

This chapter addresses the objective: "*Develop an accurate method for the reconstruction of planar points from a frame captured at the expected camera pose*." The novel contribution to knowledge for this chapter is that for camera poses that have a high angle of incidence with the calibrated plane, estimating $P_{reconstruction}$ gives lower reconstruction error than estimating the inverse of $P_{projection}$.

6.2 Preparing the Data and Intrinsic Calibration

A 0.96 m x 0.24 m plane was created and printed onto a flat board. As in Chapter 5, the long dimension (0.96 m) of this board was a scaled representation of the length of the hockey pitch. The short dimension (0.24 m) was chosen to be as long as possible given the most extreme of the performance analyst's possible camera positions and no wide angle lens converter. As this chapter is investigating the method to reconstruct points onto the calibrated plane and assuming the points have been undistorted, it is irrelevant whether the wide angle lens converter is included. The chapter is more concerned with demonstrating that the proposed method of reconstruction gives superior results, rather than the absolute value of those results. The wide angle converter does however affect the quality of the image produced; therefore it was removed to eliminate any aberrations it may cause in the image.

In both dimensions, at intervals of 0.08 m, circles of diameter 0.04 m formed a 13 x 4 grid of 52 known points. Chapter 5 demonstrated a method to extract accurate known points from a grid of projected circles.

The four corner points plus the two mid points along the long dimension formed the set of six control points. The numbering of these started with 1 at (0, 0) and proceeded in a clockwise direction. As illustrated in Figure 6.1, the board was captured by a camera located at the midpoint of the short board dimension, set-back 0.22 m from the board and elevated by 0.11 m (X= 1.18 m, Y= 0.12 m, Z= 0.11 m). As in Chapter 5 the camera was a Sony FDR-AX33, but this time without a wide-angle lens converter. The capture resolution was 4K (3840 pixels x 2160 pixels).

The image collection, intrinsic calibration and known world point extraction followed the procedure listed in Chapter 5. It was assumed that the extracted circle centres were error free. The image control points were taken from the set of extracted circle centres. As no wide-angle lens converter was mounted, Browns distortion model (Brown 1971) was used. The focal length, principal point, tangential distortion and three term radial distortion were estimated. Figure 6.2 displays the plane frame with the overlaid circle centres.

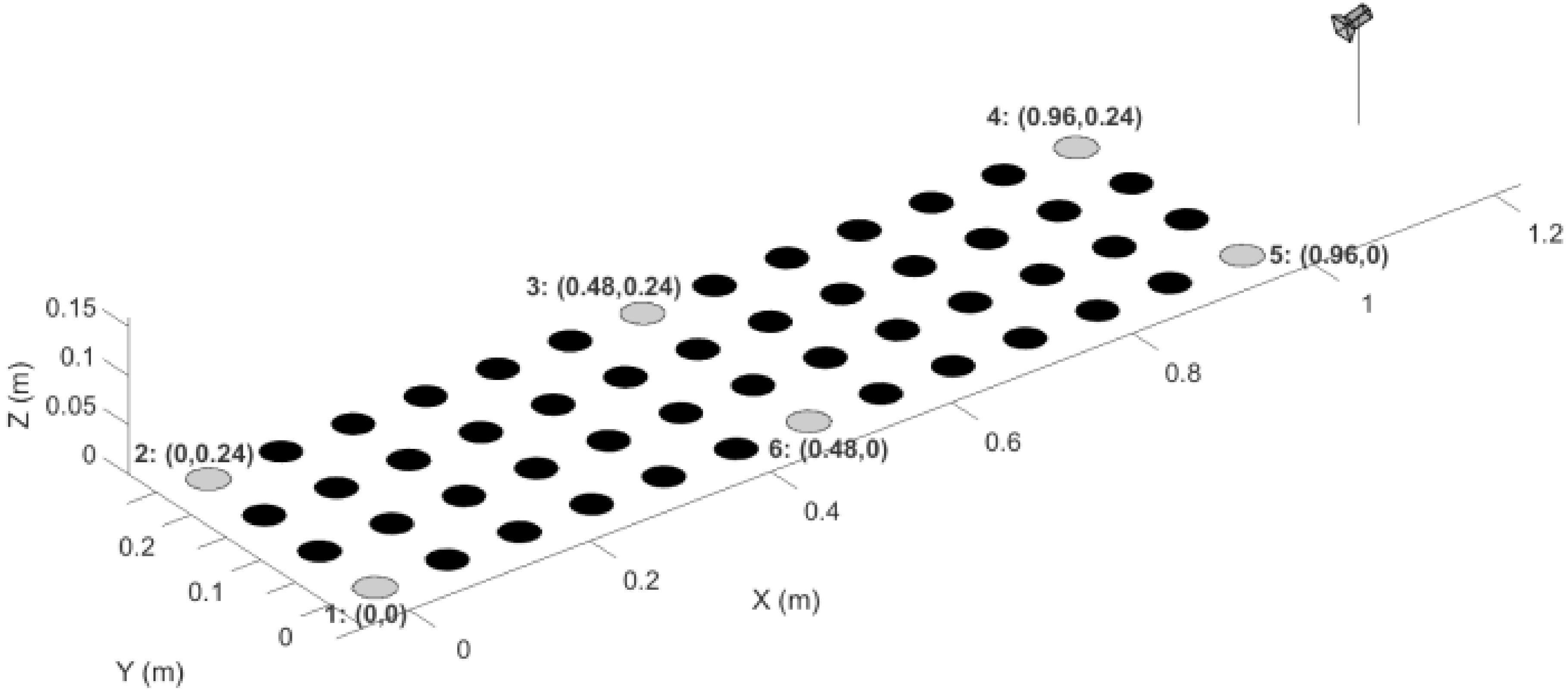


Figure 6.1: The experimental setup. The control points are indicated and labelled in grey. The camera was positioned at X= 1.18 m, Y= 0.12 m and Z= 0.11m.

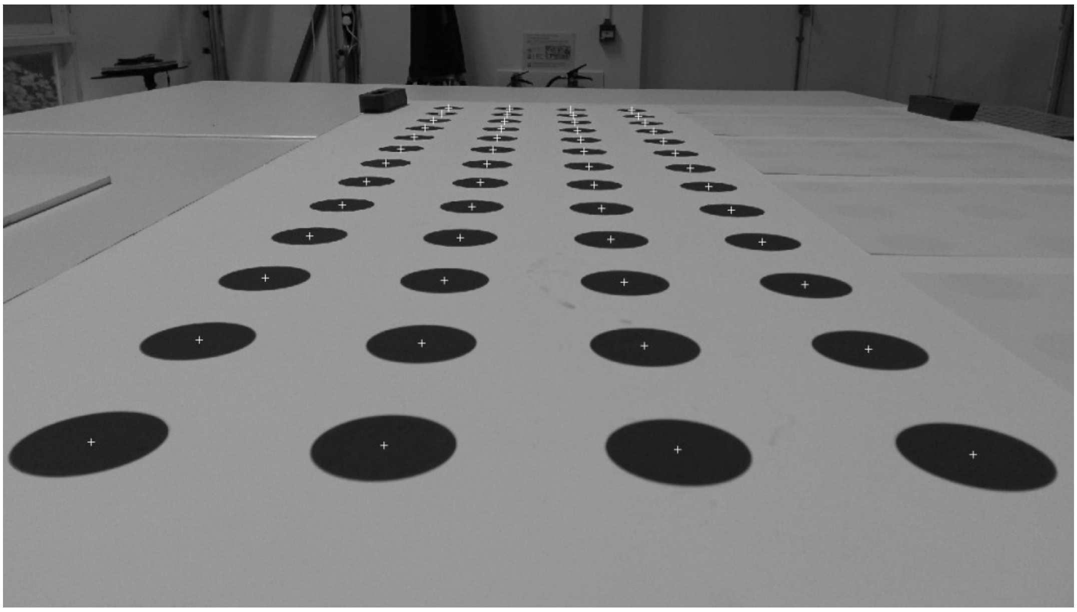


Figure 6.2: The plane frame overlaid with the extracted circle centres.

6.3 Classic Method of Reconstruction

Bouguet's (Bouguet 2015) implementation of (Heikkila & Silven 1997) was used to determine $P_{projection}$. This is a two-step process:

1. Initial Estimate - Estimate the homography that transforms from world coordinates to camera coordinates
2. Refine the Estimate - Convert the homography to $P_{projection}$ and refine the estimate by minimising the projection error.

The fit of $P_{projection}$ can be examined using each control point's resultant projection error. Figure 6.3A compares the resultant projection error after the initial extrinsic estimate and the refine procedure. It can be observed that after the initial estimate control point 5 has a relatively large projection error. As expected, the refine procedure adjusts $P_{projection}$ to minimise the total projection error. This reduces control point 5's projection error at the expense of other control points. After this refine procedure, $P_{projection}$ has the minimal projection error.

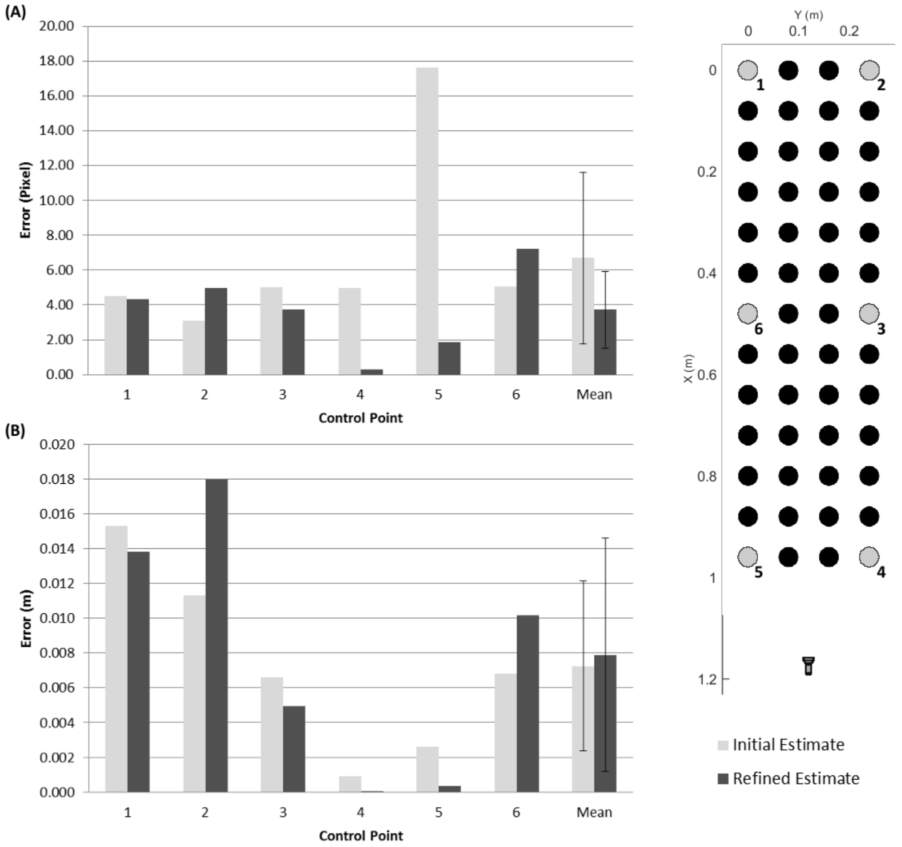


Figure 6.3: Classic Method - For each control point and the mean, after the initial estimate and the refine procedure: (A) The resultant projection error. (B) The resultant reconstruction error. Error bars on mean indicate the standard deviation.

The minimisation procedure weights all pixels the same irrespective of their position on the calibrated plane. Yet it can be observed in Figure 6.4 that there is a large range of planar resolution over the control points; a one pixel error at control point 1 accounts for a larger world error than a one pixel at control point 5. If $P_{projection}^{-1}$ is used for reconstruction, there is a bias towards those points with a higher planar resolution. Figure 6.3B displays the resultant reconstruction error, computed using $P_{projection}^{-1}$ and the method outlined in (Dunn et al. 2012), after the initial estimate and the refine procedure. If optimal, the reconstruction error would be similar across all control points. Instead the reconstruction error is biased towards control points 4 and 5, those points with high planar resolution.

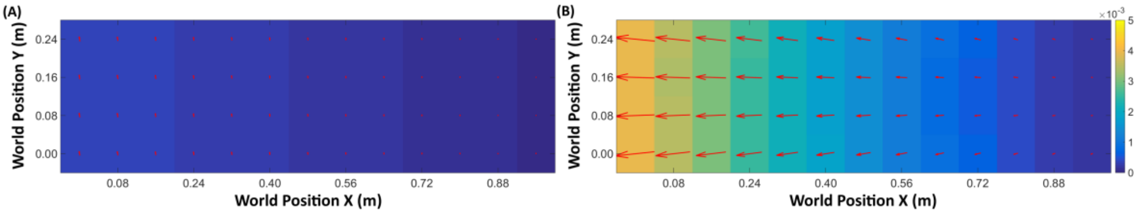


Figure 6.4: For each of the world known points, indicated by their *(X, Y)* position on the calibrated plane, the effect of a one pixel difference on the reconstructed coordinates in metres. (A) A one pixel difference in the *u* dimension. (B) A one pixel difference in the *v* dimension. The colours indicate the absolute magnitude of the difference. The arrows indicate the direction and the relative magnitude of the difference.

The non-optimality of $P_{projection}^{-1}$ can also be observed by the increase in the mean reconstruction error as a result of the refine procedure. After the initial estimate, despite having the largest projection error, control point 5 has the second lowest reconstruction error. As stated previously the refine procedure reduces control point 5's projection error. Control point 5 has a high planar resolution so the large reduction in projection error results in a relatively small reduction in the reconstruction error. Subsequently, the projection error for control point 2, which has a much lower planar resolution, is increased, which has a much larger negative effect on its and the mean reconstruction error.

6.4 Proposed Method of Reconstruction

The previous section illustrated that for a camera pose with a high range of planar resolution, $P_{projection}^{-1}$ is not optimal when reconstructing the scene. This is a result of the minimisation objective function not reflecting the assessment criteria. Instead, I propose to calculate the projection matrix, $P_{reconstruction}$, by minimising the total reconstruction error.

The proposed method uses the same procedure to estimate the intrinsic parameters and to determine the control point correspondences. Following that the homography from undistorted image control points to world control points is calculated using

Singular Value Decomposition. As there are 6 control points, the resulting homography is refined by minimising the reconstruction error. The proposed method is only equivalent to the initial estimate of the classic extrinsic estimation method. Similar to the classic method it may be possible to further optimize the projection matrix in a refine step.

Figure 6.5 compares the projection error and the reconstruction error for the classic method and the proposed method. A lower mean reconstruction error (Figure 6.5B) indicates the proposed method is superior if the desire is to reconstruct the points. The smaller standard deviation indicates more consistency in the reconstruction error of the proposed method over the classic method.

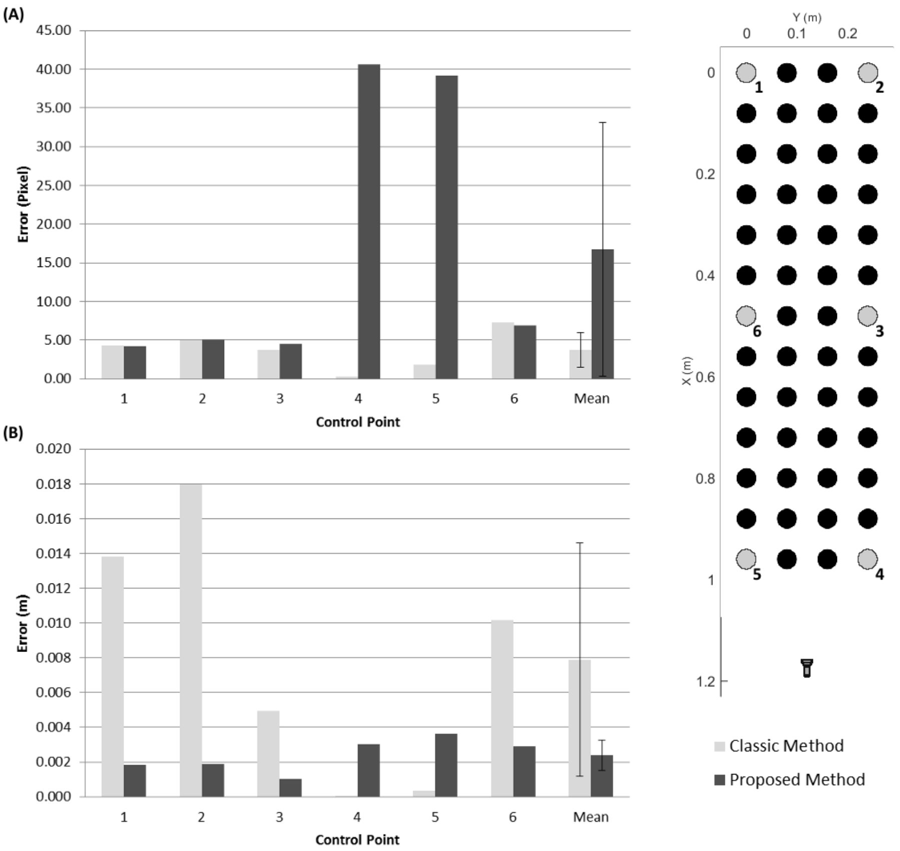


Figure 6.5: Classic Method vs the Proposed Method - For each control point and the mean: (A) The resultant projection error. (B) The resultant reconstruction error. Error bars on mean indicate the standard deviation.

A consequence of the proposed method is control point 4 and 5 have relatively high projection error (Figure 6.5A). $P_{reconstruction}^{-1}$ would be worse at augmenting reality than $P_{projection}$, as the projection error is no longer optimized. This suggests the extrinsic calibration procedure should be chosen based upon the desired operation of the projection matrix.

6.5 Accuracy across the Calibrated Plane

In the previous section it was shown that, for the control points, more accurate reconstruction could be achieved by using $P_{reconstruction}$ rather than $P_{projection}^{-1}$. However for this to be practically useful the improvement must generalise to the entire calibrated plane. (Hudson 2015) assess the quality of a calibration using the mean reconstruction error. This metric is often accompanied by the standard deviation to indicate the variation in the reconstruction errors. Reporting the mean and standard deviation assumes the reconstruction errors follow a normal distribution, yet as the reconstruction error is an absolute value it is a folded normal distribution (Tsagris et al. 2014). Subsequently the mean and standard deviation are not representative of the true distribution of the reconstruction errors. Instead the median reconstruction error was calculated for all 52 known points on the calibrated plane. Additionally, the maximum and minimum reconstruction errors were computed.

Table 6.1 lists summary statistics for the two methods.

Table 6.1: Reconstruction statistics across the 52 known grid points for the classic method and the proposed method

	Classic Method	Proposed Method
Median Reconstruction Error	0.0047 m	0.0034 m
Maximum Reconstruction Error	0.0182 m	0.0112 m
Minimum Reconstruction Error	0.0000 m	0.0004 m

The maximum reconstruction error has been reduced and the minimum reconstruction error has been increased suggesting more consistency across the calibrated plane. This is supported by the reconstruction error maps in Figure 6.6.

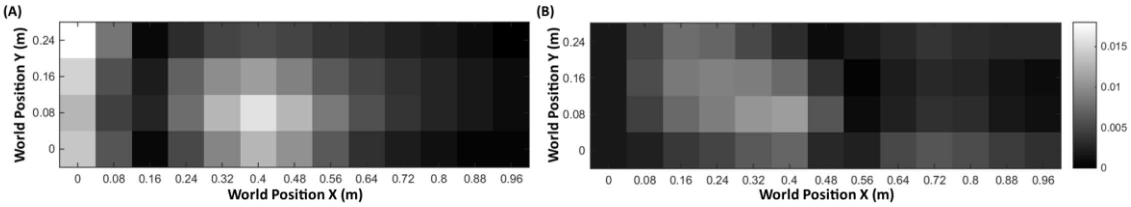


Figure 6.6: For each of the world known points, indicated by their (X, Y) position on the calibrated plane, the reconstruction error in metres. (A) Classic method. (B) Proposed method.

Both error maps display relatively high reconstruction error in the range: X = 0.16 m – 0.4 m, Y = 0.08 m – 0.16 m. This is a result of the intrinsic model; standard deviation of the projection error over calibration points: u = 1.69 pixels, v = 1.41 pixels. The intrinsic parameters are used to undistort the image; therefore any error is compounded in the estimation of the circle centres. A better intrinsic model will have an effect on the reconstruction error; however due to the systematic nature of this error, it is still valid to state that the proposed method gives better reconstruction accuracy than the classic method.

Apart from the anomalous region explained above, the proposed method has a reconstruction error less than 0.01 m across the entire plane. In contrast, the classic method exhibits a trend of relatively high reconstruction errors in the region furthest from the camera, the area of lowest planar resolution.

6.6 Summary

This chapter addressed the objective: "*Develop an accurate method for the reconstruction of planar points from a frame captured at the expected camera pose*." To do so, it investigated if better planar reconstruction accuracy could be achieved by estimating $P_{reconstruction}$ rather than estimating $P_{projection}^{-1}$, for a camera pose with a high angle of incidence to the calibrated plane.

$P_{projection}$ is the optimal projection matrix that transforms from world coordinates to image coordinates. However, for a camera pose with a high range of planar resolution, using $P_{projection}^{-1}$ to reconstruct world points is not optimal. The reconstruction is biased towards the points with high planar resolution.

Instead, $P_{reconstruction}$, the projection matrix from image coordinates to world coordinates, can be found directly by minimising the reconstruction error. Traditionally this minimisation is not performed because errors in the image point extraction are propagated through to $P_{reconstruction}$. This method gives a significant reduction in the reconstruction error over the classic method.

This chapter has demonstrated that for a camera position with a high angle of incidence to the calibrated plane, such as those expected at an international hockey tournament, minimising the projection, $P_{reconstruction}$, from image coordinates to world coordinates, gives more accurate mean reconstruction than using the traditional inverse of $P_{projection}$. It also displays less variance in the reconstruction error across the calibrated plane. Using this projection has not previously been considered in the sports analytics literature. By simply changing the algorithm used to compute the extrinsic parameters, a performance analyst can be more confident in the accuracy of coordinates across the entire pitch. This change does not affect their data collection procedure. The subsequent chapters will use this proposed method to investigate the effect of different camera conditions on the reconstruction accuracy.

7 Reconstruction Accuracy

7.1 Introduction

In Chapter 6, the image control points were extracted from the set of circle centres. Given the limitations of the extraction method it was assumed that the extracted circle centres were error free and deterministic; ergo the image control points were error free and deterministic. In an applied environment the image control points are identified by the performance analyst clicking directly in the image. This is a stochastic process due to human error and the above assumption does not hold. Chapter 6 showed that there is a difference in planar resolution across the calibrated plane. This difference means that the metric area accounted for by a single pixel is not consistent across the plane. Subsequently it can be hypothesised that some control points will have a larger effect on the reconstruction error than others. Consequently this chapter investigates the impact of error in the control points on the reconstruction error. First the reconstruction accuracy for the camera pose experienced at a recent international field hockey tournament is calculated. Following this, the chapter reports how the reconstruction accuracy is affected by variation in the control points.

This chapter addresses the objective: "*Assess the effect of control point errors on the reconstruction error*"*.* The novel contribution to knowledge for this chapter is how typical control point identification errors affect the reconstruction error when considering a performance analyst's typical pose at an international hockey tournament.

7.2 Method

A Monte Carlo method (Metropolis & Ulam 1949) is a simulation method that uses repeated normal distribution random sampling to estimate numerical results. This can be applied to estimate the distribution of the reconstruction error given the expected variance in each of the control points. Assuming a probability distribution defined for each control point, a random control point vector can be formed by sampling each distribution. Using this control point vector the extrinsic parameters can be estimated and the reconstruction error calculated. If this process is repeated multiple times, each with a different random control point vector, the distribution of the reconstruction error given the expected control point variance is estimated.

The model frame for the circle condition in Chapter 5 was reused. As displayed in Figure 7.1, this frame captured the modified circles plane. This modification was designed to aid locating the circle centre at different planar resolutions. The camera pose was similar to that in the EuroHockey footage used in Chapters 3-4.

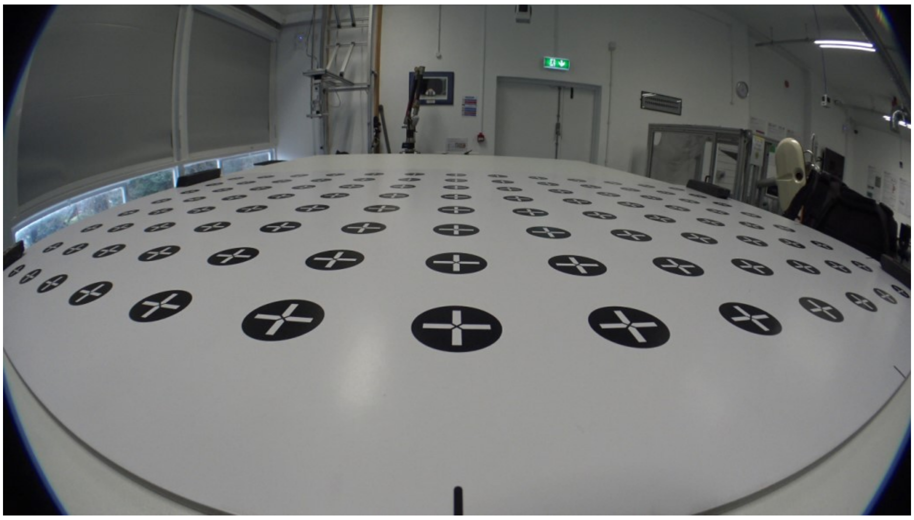


Figure 7.1: Chapter 7 reused the plane frame used for the circle condition in Chapter 5. This plane frame captured the plane of circles that had been modified so the circle centres could be identified.

A hockey pitch has fourteen line intersections. Each of these line intersections has a defined location and can therefore be used as a control point. To replicate this in the scale model, the circle centre closest to each intersection was used as a control point. These control points were labelled **A-N**. The labelling was clockwise around the model plane, with **A** at (0, 0). The Figure 7.2 displays a hockey pitch schematic overlaid with the modified centres. The control points are indicated in red.

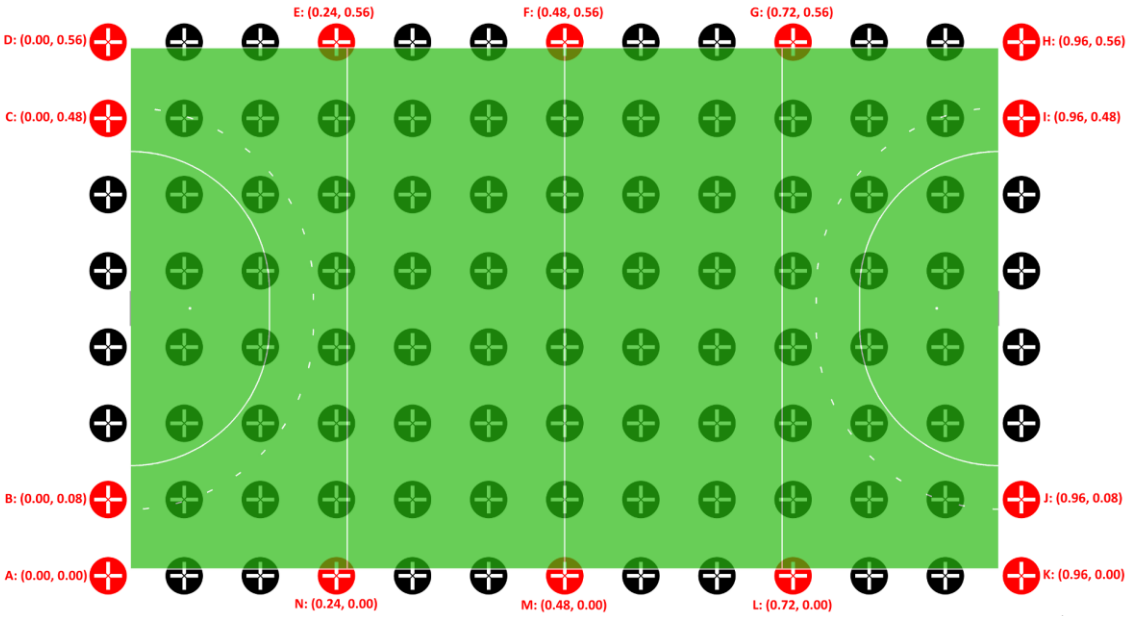


Figure 7.2: A hockey pitch schematic overlaid with the model plane. Each of the fourteen line intersections on the hockey pitch has a defined world coordinate, meaning they are suitable to act as control points. This is replicated on the model by using the nearest circle centre to each intersection. These control points, indicated in red, are labelled A-N clockwise from A at (0, 0).

In this thesis the control point clicked coordinate is assumed to follow a two-dimensional Gaussian distribution. The difference described by this distribution can be attributed to:

1. Misidentification of the correct point in the image. This is unpredictable and cannot be modelled.

2. Misalignment of the cursor with the correct point in the image. This is modelled here by the Gaussian distribution.

A Gaussian distribution can be represented by the mean value and a covariance matrix. Chapter 5 showed that the variance of the clicked points is similar across the calibrated plane; however it cannot be assumed that this is also true of the covariance and as such a unique covariance matrix was used for each of the control points. For each of the 14 control points the mean value and covariance matrix were calculated by manually clicking the point twenty times. Two hours was left between each sample to limit the learning effect caused by memorising the point that was clicked.

Sixteen different control point conditions were considered. The conditions were defined by how the control point vector was constructed. The first condition constructed the control point vector from the mean clicked vector. Condition 2 used the mean clicked vector for all points apart from control point **A** which was randomly sampled from its calculated Gaussian distribution. Similarly, conditions 3-15 used the mean clicked vector for all points apart from **B-N** respectively. Finally condition, 16 was constructed by sampling all the control points. The construction of each condition is summarised in Table 7.1.

Table 7.1: The control points that were randomly sampled for each of the control point vector conditions. If a control point was not sampled it was set to its mean clicked value.

Control Point Vector Condition	Which control points were randomly sampled?
1	None – Used the mean clicked vector
2	**A**
3	**B**
4	**C**
5	**D**
6	**E**
7	**F**
8	**G**
9	**H**
10	**I**
11	**J**
12	**K**
13	**L**
14	**M**
15	**N**
16	**A, B, C, D, E, F, G, H, I, J, K, L, M, N**

For each control point condition:

1. The following was performed 10,000 times:
    a. The control point vector was set to the mean clicked vector.
    b. Each control point that was to be sampled was updated with a value drawn from the relevant Gaussian distribution.
    c. The extrinsic parameters were calculated using the method introduced in Chapter 6. The intrinsic parameters were those calculated in Chapter 5.
    d. The circle centres extracted in Chapter 5 were reconstructed on the calibrated plane and the median reconstruction error was calculated.

2. The mean and standard deviation of the median reconstruction error was calculated.

For condition 1 the process was not repeated 10,000 times as it used the mean clicked control points.

7.3 Results and Analysis

Figure 7.3 visualises, for each control point, the two-dimensional probability density function (pdf) generated from the clicked data. The control points were sampled randomly from these probability density functions. In general the pdfs have more variance in the u dimension than the v dimension. This is probably due to the higher resolution in the u dimension than the v dimension yet is counter to the findings in Chapter 5. Control points **B** and **M** had no variance in one dimension and as such are displayed as lines.

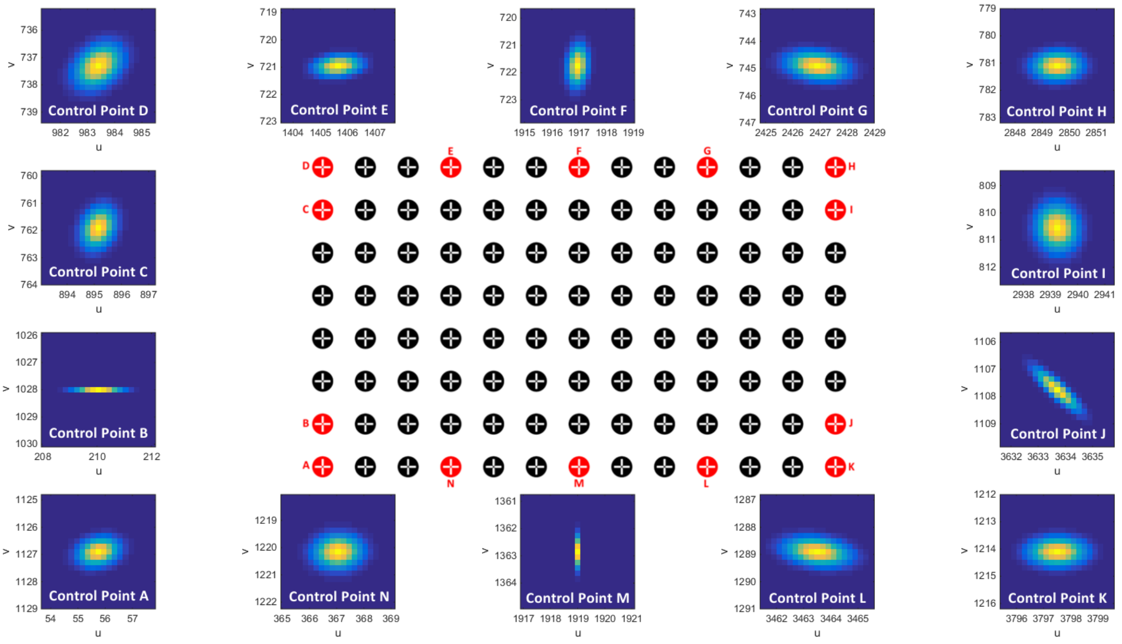


Figure 7.3: The probability density functions (pdf) for each control point. The pdfs are positioned relative to the corresponding control point.

Figure 7.4 plots the reconstruction error for condition 1, the mean clicked control points. The box plot follows the convention of (Tukey 1977). The median value is indicated by the red line and the interquartile range (IQR) by the blue box. The extent of the whiskers denotes the highest datum within 1.5 IQR of the 3rd quartile and lowest datum within 1.5 IQR of the 1st quartile. Any datum outside the range of the whiskers is classed as an outlier and marked with a red cross. In this case there were no outliers.

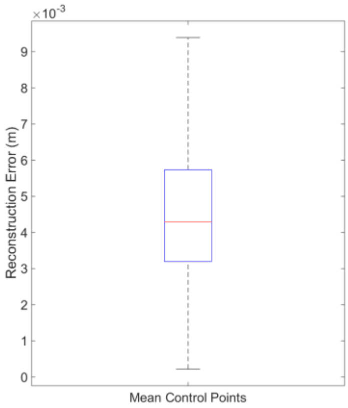


Figure 7.4 The reconstruction error for the set of mean control points. The box plot follows the convention of (Tukey 1977).

The median reconstruction error when using the mean control points was 0.0043 m with an interquartile range of 0.0025 m (1st Quartile = 0.0032 m. 3rd Quartile = 0.0057 m). (McInerney 2017) suggested that ±0.5 m is an acceptable reconstruction error for field hockey. To make this error applicable here it must be reduced by the same scaling factor as the model; subsequently a reconstruction error of 0.005 m is deemed acceptable. In this thesis the percentage of known world points that are below this value will be designated the 'Acceptable Error Rate'. The Acceptable Error Rate for the mean control points was 63.5%.

As conditions 2-16 were performed using a Monte Carlo simulation, rather than a single median reconstruction error there is a median reconstruction error for each simulation. Consequently a median reconstruction error probability distribution can be constructed for each condition. Figure 7.5 displays the median reconstruction error probability distribution for the control point vector conditions 2-15. The mean median reconstruction error was consistent across the control point vector conditions (0.0043 m), the same as the median reconstruction error when using the mean control point vector. This is to be expected as the non-mean control points were sampled from a Gaussian distribution. However, the standard deviation varied across the conditions. The median reconstruction error exhibited little variance for the conditions where the sampled control point was on the near side of the plane. In contrast, the median reconstruction error displayed more variance for the conditions that sampled control points on the far side of the plane. This pattern can be attributed to the planar resolution and is particularly evident when comparing conditions 7 and 10. Condition 7 randomly sampled control point **G** and condition 10 randomly sampled control point **L**. It can be observed in Figure 7.3 that these two control points have very similar sampling distributions, yet **G** is an area of low planar resolution and **L** in an area of high planar resolution. The result of this, which can be observed in Figure 7.5, is that despite the mean of the median reconstruction error being similar the standard deviation of the median reconstruction error for condition 7 is an order of magnitude larger than condition 10 (Condition 7 = 0.000069 m vs Condition 10 = 0.000008 m).

This result can be explained by considering the process of extrinsic calibration. The extrinsic calibration finds the rigid homography that minimises the distance between the world control points and the image control points projected onto the world plane. In areas of low planar resolution the same variance in the image point results in larger variance in the world point. Larger variance in the world point results in larger variance in the extrinsic parameters and as such larger variance in median reconstruction error.

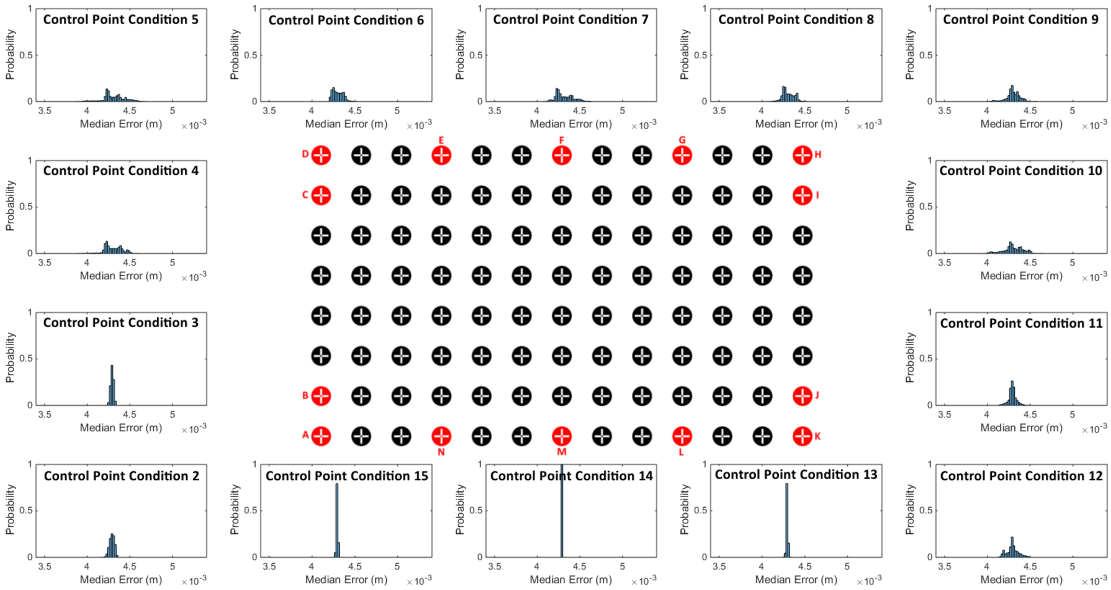


Figure 7.5: The median reconstruction error probability distribution for control point vector conditions 2-15. The error probability histograms are positioned relative to the control point that they randomly sampled.

Control point vector condition 16 considered the more realistic case where all control points were subject to some random error. Figure 7.6 displays the probability distribution for the median reconstruction errors. Note that the probability scale is different to that used in Figure 7.5. The mean median reconstruction error for condition 16 was 0.0043 m and had a standard deviation of 0.00028 m; therefore 95% of the time the median reconstruction error will be between 0.0037 m and 0.0049 m. As noted previously the mean median reconstruction error is the same as the mean control point’s median reconstruction error because the control points were sampled from a Gaussian distribution.

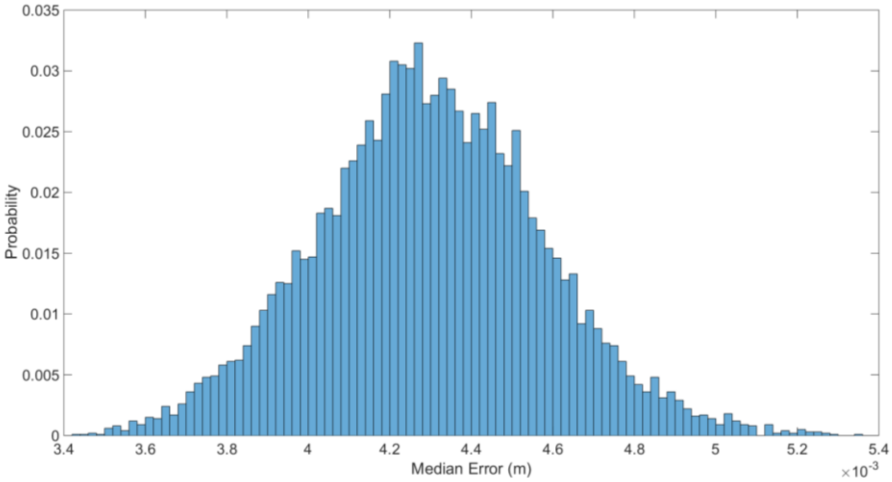


Figure 7.6: The median reconstruction error probability for the control point vector condition 16. Note that the probability scale is different to the plots in Figure 7.5.

The minimum median reconstruction error was 0.0034 m. While it can be claimed that the control point vector that gave this result is optimal, it cannot be inferred that this control point vector is closest to the true control point vector. The distortion model is not error free and as such residual distortion remains in the control points. The control point vector with minimal median reconstruction error is the one that best compensates for this residual distortion.

The mean Acceptable Error Rate was 63.5% and had a standard deviation of 1.1%. Again the fact that this value is the same as for the mean control point vector can be explained by the Gaussian nature of the control point sampling. The standard deviation of 1.1%, which equates to 1.15 points, suggests that random noise in the control points causes little variation in the number of the points that are below 0.005 m.

7.4 Summary

This chapter addressed the objective: *"Assess the effect of control point errors on the reconstruction error".* Experimental data was used to estimate the expected two-dimensional distributions for each of 14 control points. Given the distributions for the

14 control points, 16 different control point vector conditions were defined. Monte Carlo simulation was used to estimate the median reconstruction error distribution for each control point vector condition.

The first control point vector condition used the mean control points and had a median reconstruction error of 0.0043 m and interquartile range of 0.0025 m. The Acceptable Error Rate was defined as the percentage of the known world points that had a reconstruction error less than 0.005 m. The Acceptable Error Rate for the mean control points was 63.5%.

The next 14 control point vector conditions each randomly sampled a single control point. The resulting distributions indicate that variance in the control points in areas of low planar resolution cause more variance in the median reconstruction error.

The final control point condition randomly sampled all 14 control points. The resulting median reconstruction error distribution had mean 0.0043 m and a standard deviation of 0.00028m. The Acceptable Error Rate had mean 63.5% and a standard deviation of 1.1%. These statistics are an indication of the expected median reconstruction error for the camera pose used for the rest of the footage in this thesis. They suggest that over half of the reconstructed points were within 0.005 m of their expected location, however equally this means that for 36.5% of the points there is an error of greater than 0.005.

8 Camera Assembly

8.1 Introduction

As noted in the literature review, 4K footage has four times as many pixels per unit area than HD. This means that the image representation of an object in the scene is comprised of more pixels and as is such more detailed. An increase in detail is useful in the computer vision tasks presented in Chapters 3 and 4. However, it must be ensured that this increase in resolution does not increase the reconstruction error.

The previous three chapters have all used a FDR-AX33 4K camera when considering camera calibration. However, it is unknown if this camera assembly gives typical performance. This chapter investigates this by comparing the reconstruction error for three different camera assemblies. It reports both an inter camera and intra camera comparison to investigate if the observed effect is hardware specific.

This chapter addresses the objective: "*Assess the effect of camera assembly on the reconstruction error*". The contribution to knowledge for this chapter is demonstrating that the specific camera assembly is vital when trying to minimise the reconstruction error. It also demonstrates that using a 1:100 scale model has limitations due to the difficulty in replicating camera poses.

8.2 Method

The study considered three camera assemblies: 1) 4K, 2) HD-AX33, and 3) HD-PJ260VE. These were chosen, as they are representative of current consumer hardware. The 4K assembly used a Sony FDR-AX33 with a 0.28x Raynox HDP-2800ES lens converter and captured at an image resolution of 3840 pixels x 2160 pixels. The HD-AX33 assembly used the same camera but the image capture resolution was set to 1920 pixels x 1080 pixels. This assembly was included to ensure that the identified effect is due to the resolution rather than a difference in hardware. The HD-PJ260VE assembly used a Sony HDR-PJ260VE with a 0.3x Opteka Platimum Series lens converter and captured at 1920 pixels x 1080 pixels. These assembly details are summarised in Table 8.1.

Table 8.1: Details of the camera assemblies included in the study.

Assembly	Imaging Device	Resolution
4K	Sony FDR-AX33 + 0.28x Raynox HDP-2800ES	3840 pixels x 2160 pixels
HD-AX33	Sony FDR-AX33 + 0.28x Raynox HDP-2800ES	1920 pixels x 1080 pixels
HD-PJ260VE	Sony HDR-PJ260VE + 0.3x Opteka Platinum Series	1920 pixels x 1080 pixels

Each camera assembly captured an image of the model plane from a camera pose similar to that of the EuroHockey footage in Chapters 3 and 4. The camera pose was identical for the 4K and HD-AX33 assemblies, however due to the difference in the physical properties of the cameras, an identical pose for the HD-PJ260VE assembly could not be guaranteed. The model plane was similar to the model plane used in Chapter 5 however filled circles were used. The fourteen control points were those used in Chapter 7.

For each camera assembly, the data collection procedure and the circle centre extraction procedure were the same as that in Chapter 5. This gave a set of intrinsic parameters and a set of image known points for each camera assembly. The fourteen image control points were extracted from the set of image known points and the extrinsic parameters were estimated as in Chapter 6. The known points were reconstructed and the reconstruction errors calculated.

Two Wilcoxon signed-rank tests (Woolson 2008) were performed to compare 4K vs HD-AX33 and HD-AX33 vs HD-PJ260VE. The Wilcoxon signed-rank test is a non-parametric test that uses paired samples to determine if two populations have the same distribution. It is used here because the reconstruction error forms a folded distribution and as such parametric tests are unsuitable.

8.3 Results and Discussion

Figure 8.1 displays a boxplot of the results for each of the camera assemblies. As in Chapter 7 the boxplot follows the convention of (Tukey 1977).

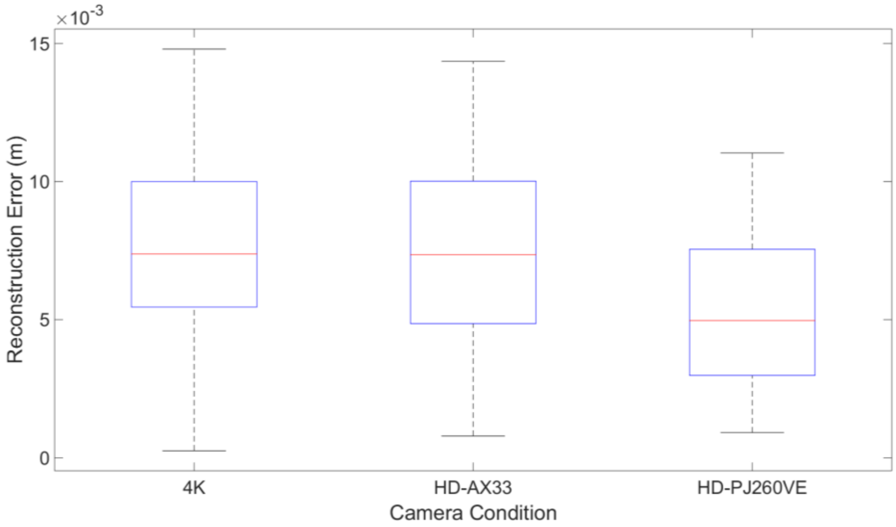


Figure 8.1: The reconstruction error in metres for each of the camera assemblies. The box plot follows the convention of (Tukey 1977)**. The median value is indicated by the red line and the interquartile range (IQR) by the blue box. The extent of the whiskers denotes the highest datum within 1.5 IQR of the 3rd quartile and lowest datum within 1.5 IQR of the 1st quartile. Any datum outside the range of the whiskers is classed as an outlier and marked with a red cross.**

An important thing to highlight is the difference between the 4K result here and that presented in Figure 7.4. Despite these boxplots purporting to measure the same thing, the results here (median reconstruction error = 0.00073) are worse than in Chapter 7 (median reconstruction error = 0.0043 m). This difference is attributed to an inconsistence in camera pose. Camera pose is a combination of camera location and camera orientation. As displayed in Figure 8.2, care was taken to ensure the camera location in the repeated experiment was similar; however the same care was not afforded to the camera orientation. This highlights a fundamental limitation of using a scale model to assess the reconstruction error; a small absolute change to the pose has an exaggerated effect on the scale model. For example a 10 cm camera translation at true scale, equates to a 1 mm translation on the scale model. It is therefore difficult to ensure an identical pose when an experiment is repeated on the scale model. It can be supposed that the assessment of the reconstruction on an actual pitch would be more repeatable as the camera can be placed with more accuracy; however this has practical limitations and other errors must be considered, i.e. errors in the placement of the world known points.

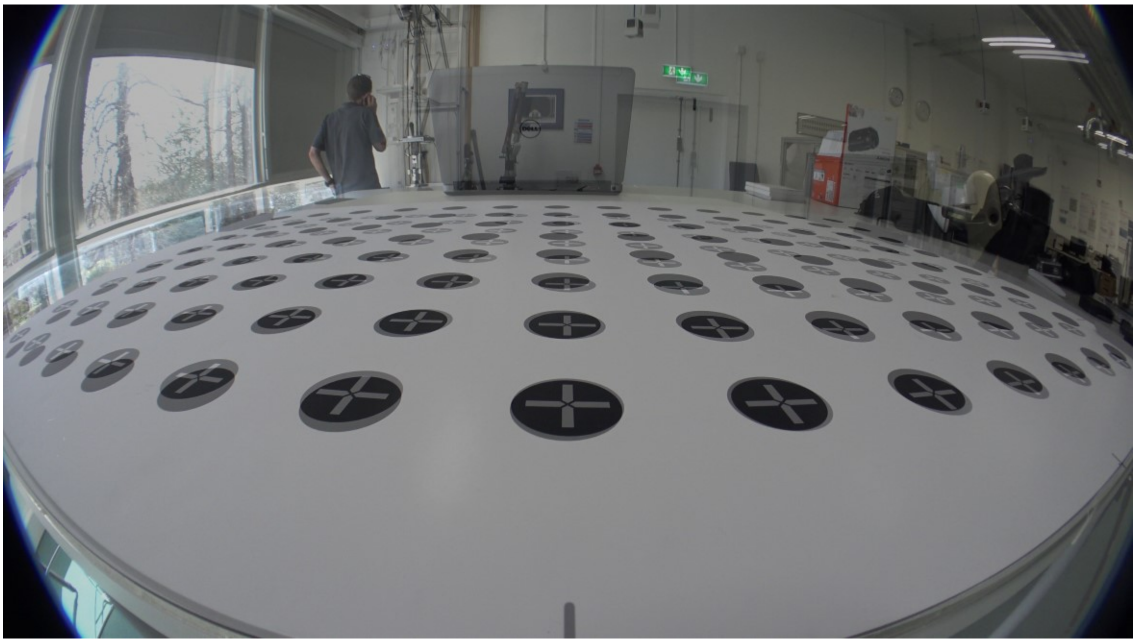


Figure 8.2: The plane frame from Chapter 7 overlaid with the 4K assembly. Despite the two camera locations being similar, the camera orientations are inconsistent.

In this chapter care was taken to ensure each camera assembly was placed at the same position and orientation and as such results between the different camera assemblies are comparable. Therefore the investigation into the effect of the different camera assemblies on the reconstruction is valid. A more extensive investigation into the effect of pose on the reconstruction accuracy is included in Chapter 9.

The next section will compare the 4K assembly with the HD-AX33 assembly, i.e. it will consider the effect of resolution keeping the hardware consistent. The following section will compare the HD-AX33 assembly with the HD-PJ260VE assembly. This comparison is important to determine if existing 4K hardware is limiting the reconstruction accuracy.

8.3.1 4K vs HD-AX33

It is clear in Figure 8.1 that 4K and HD-AX33 assemblies had similar reconstruction error distributions. This is supported by the Wilcoxon signed-rank test which indicated no statistical significant difference between the two assemblies ($z = 1.50$, $p < 0.134$). The image is a projection of the world scene onto the image plane; however the image is formed from discrete pixels. If it is assumed that a point can only be identified to the nearest pixel it can be supposed that a point in the HD scene cannot be extracted as accurately as a point in the 4K scene. It follows, that in a theoretical world where there are no errors other than those due to the limits of the image resolution, the rigid homography should fit the 4K points better and therefore the reconstruction error should be lower. However, here the control points are formed from the world known points which are extracted to subpixel accuracy. As such this error is eliminated and the 4K world known points are equivalent to the HD world known points but scaled to the larger resolution. This is a major limitation of the study and could have been addressed by manually identifying the control points. The control points were not manually identified because it was deemed the automatic extraction was more accurate. On reflection this was a mistake in the method as the accuracy with which the control points can be identified is a key component when comparing the two resolutions.

Figure 8.3 illustrates that the planar resolution is approximately halved when using the 4K assembly rather than the HD-AX33 assembly. This is the expected behaviour as the number of pixels is doubled while the reconstructed area remains constant. As noted in this Chapter’s introduction, this increase in planar resolution means smaller objects can be resolved, increasing the accuracy of the image representation of the world and aiding the computer vision operations.

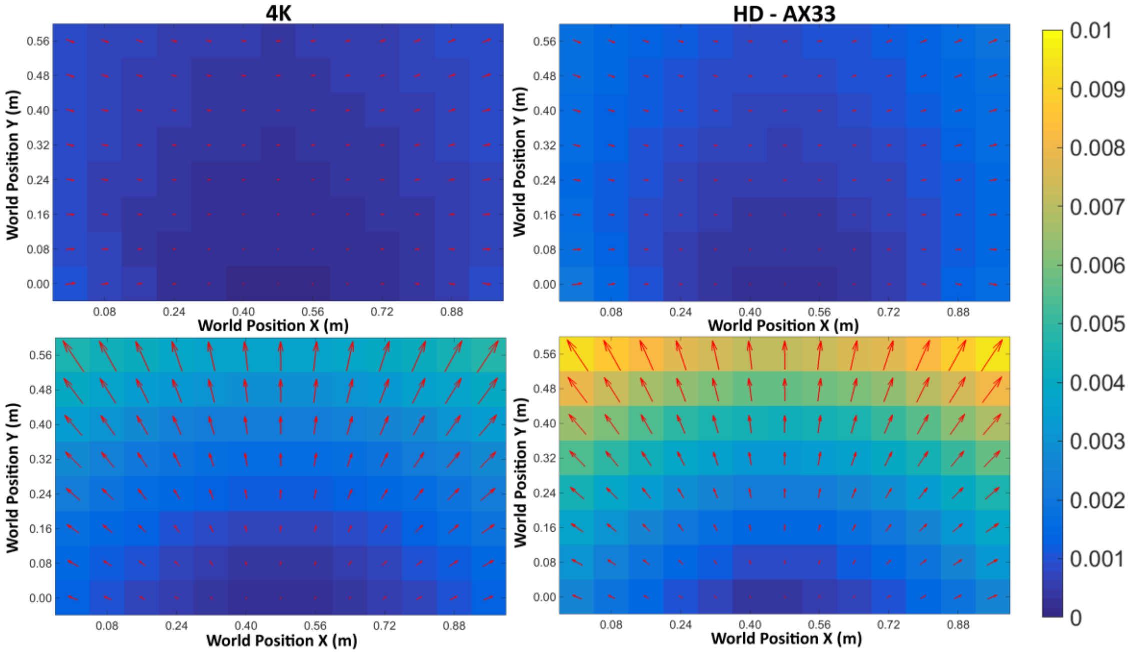


Figure 8.3: For each of the world known points, indicated by their (X, Y) position on the calibrated plane, the planar resolution for the 4K and HD-AX33 camera assemblies. Top Row – The effect of a one pixel change in the u dimension on the reconstructed coordinates of each circle centre. Bottom Row – The effect of a one pixel change in the v dimension on the reconstructed coordinates of each circle centre. The colours indicate the absolute magnitude of the difference. The arrows indicate the direction and the relative magnitude of the difference within a camera assembly.

8.3.2 HD-AX33 vs HD-PJ260VE

It is clear from the boxplots in Figure 8.1 that the reconstruction errors for the HD-AX33 assembly and the HD-PJ260VE assembly come from different distributions. This is supported by the Wilcoxon signed-rank test which indicated a statistical significant difference between the two assemblies ($z = 8.68$, $p < 0.000$). This is due to the different hardware and can be attributed to residual distortion because of ill-fitting intrinsic parameters.

The residual distortion due to the intrinsic parameters can be assessed using the mean projection error of the calibration points. For the HD-AX33 assembly the mean projection error was 0.95 pixels and for the HD-PJ260VE assembly it was 0.57 pixels. These values indicate that the intrinsic parameters were able to better model the HD-PJ260VE data than the HD-AX33 data. This is supported by the re-projection error maps in Figure 8.4. For the HD-PJ260VE, the re-projection errors are quite uniform across the frame. They tend to increase towards the periphery of the circular image but this is minor when compared to the HD-AX33. The re-projection errors for the HD-AX33 exhibit both more variance across the entire frame and higher residual distortion in the points on the periphery of the image. The following paragraphs explain how this residual distortion impacts on the reconstruction error.

Figure 8.5 displays the reconstruction error heat maps for the HD-AX33 and HD-PJ260VE camera assemblies. The colours indicate the absolute magnitude of the error. The arrows indicate the direction and the relative magnitude of the vector required to transform from the reconstructed world points to the actual world points. Despite a larger magnitude in the HD-AX33 assembly, the pattern of the errors is similar in the

HD-AX33 assembly and the HD-PJ260VE assembly. This pattern is due to residual distortion in the undistorted image. More specifically it is because the most extreme control points, those at (0.00, 0.00) and (0.96, 0.00) fall in the areas that have a large amount of residual distortion.

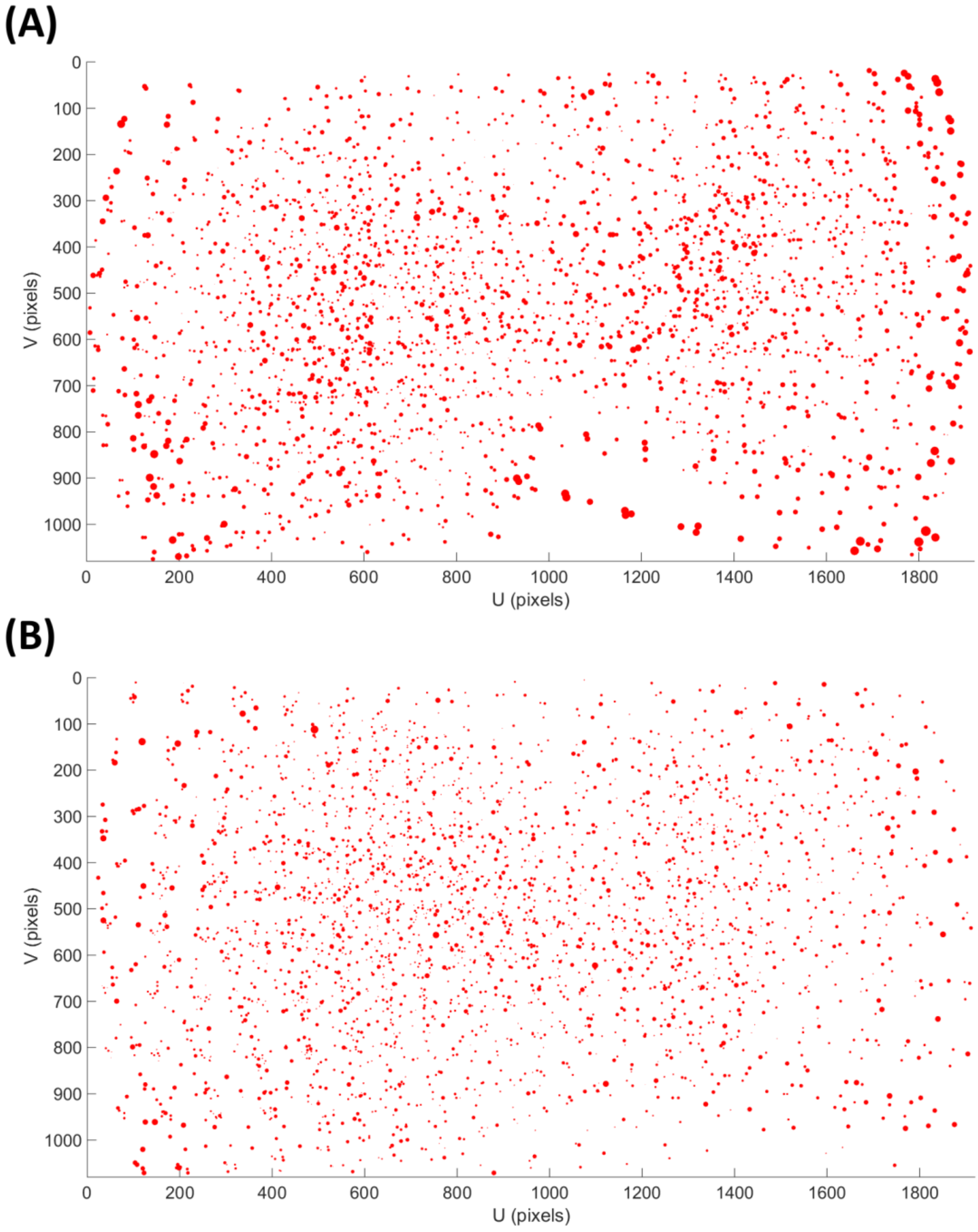


Figure 8.4: A map of the re-projection error for each of the HD intrinsic calibrations. Each dot represents one data point used in the calibration. The size of the dot is the associated relative re-projection error. (A) HD-AX33. (B) HD-PJ260VE

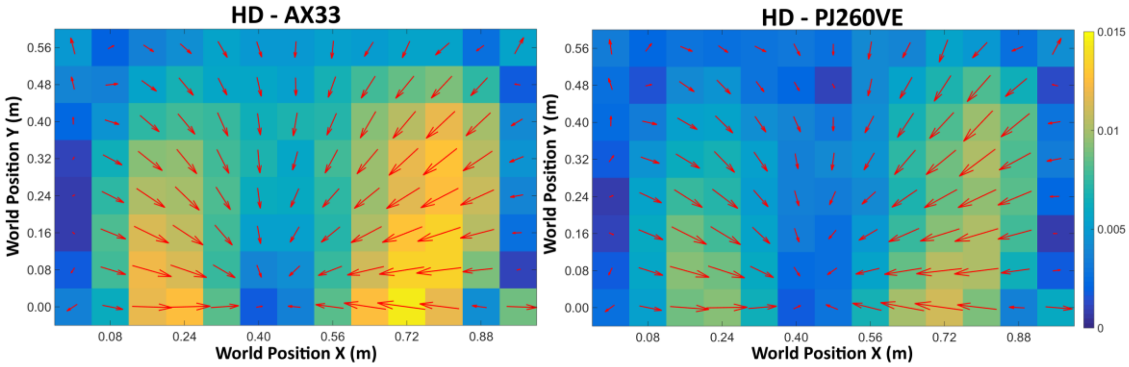


Figure 8.5: For each of the world known points, indicated by their *(X, Y)* position on the calibrated plane, the reconstruction error in metres for the HD-AX33 and HD-PJ260VE camera assemblies. The colours indicate the absolute magnitude of the error. The arrows indicate the direction and the relative magnitude of the vector transformation required to move from the reconstructed world points to the actual world points.

The known world points form a grid which can be defined by twenty one straight lines. As the extrinsic calibration is a rigid transformation from image points to world points, it follows that in a perfectly undistorted image the corresponding image points should also be incident to a grid of straight lines. Any residual distortion in the image will result in deviation from this grid; however, this grid is unknown. Instead we can consider the inverse operation; use linear regression to fit a straight line to the expected points and then examine the residuals of this fit. Figure 8.6 visualises this operation. Here the red crosses denote the normalised, undistorted known image points for the HD-AX33 assembly. The blue grid is formed of the twenty one straight lines fitted using linear regression.

Examining Figure 8.6 it is evident that points further from the principal point have more residual distortion. This is particularly apparent in the points that form the lowest of the horizontal lines. At both ends of the line the points have not been undistorted enough which means the points form a curve rather than a straight line.

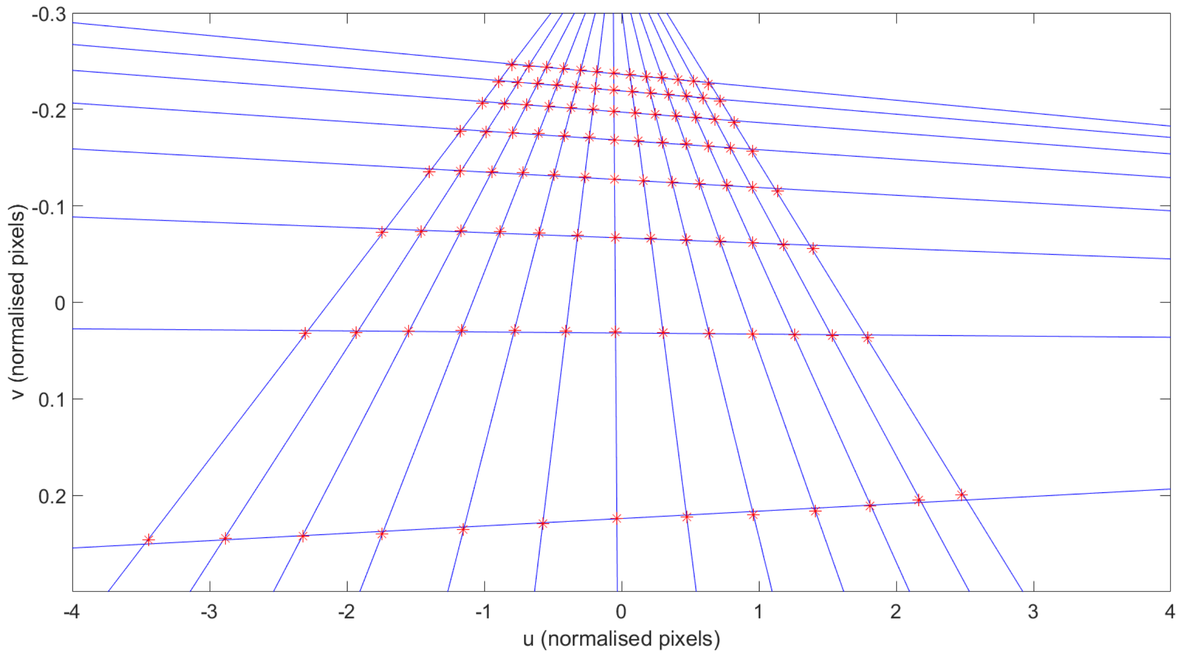


Figure 8.6: For the HD-AX33 assembly the normalised, undistorted known image points are marked with red stars. A blue line is fit to the points that form each of the twenty one expected straight lines. If the image was perfectly undistorted the image points would be incident with the line. Instead the points form a curve.

Given the fact the extrinsic calibration assumes a rigid transformation and the fact the known image points form a curve it is easier to explain the pattern of the reconstruction errors in Figure 8.5. The two points (0.00, 0.00) and (0.96, 0.00) are not fully undistorted, therefore the distance in the image between these points and the others is less than it should be. As these two points are used as control points the extrinsic model is constrained to fit them. This has the effect of pulling the reconstruction of the other points outwards towards the point, giving the pattern observed in Figure 8.5.

In hockey, only a very small percentage of the match is played in the extreme corners of the pitch. This is even more apparent with the abolition of long corners. Subsequently it may be preferred to achieve higher reconstruction across the rest of the pitch at the expense of the corners of the pitch. Further work could investigate this, specifically the effect of not using control points **A** and **K** in the estimation of the extrinsic parameters.

The results of this chapter suggest that if the automatically extracted circle centres are used as the control points then there is no difference in the reconstruction error for the HD-AX33 assembly and the 4K assembly. Despite this, it also found that the HD-PJ260VE hardware gave superior results to the 4K hardware used in the prior chapters. Subsequently the next chapter will use the HD-PJ260VE hardware assembly, to investigate the effect of camera pose on the reconstruction error. Using this hardware assembly minimises the error due to an ill-fitting intrinsic model so we can be more confident that the differences in reconstruction error are due to the camera pose.

8.4 Summary

This chapter addressed the objective: *"Assess the effect of camera assembly on the reconstruction error*". Three different camera assemblies were considered: 4K, HD-AX33 and HD-PJ260VE. The 4K and HD-AX33 camera assemblies were compared to assess the effect of resolution on the reconstruction error. The HD-AX33 and HD-PJ260VE camera assemblies were compared to assess if the reconstruction error was limited by the hardware assembly.

The 4K and HD-AX33 comparison showed that given control points automatically extracted as circle centres the two reconstruction errors are very similar. This is to be expected as if the control points are extracted on a continuous scale then HD is just a 0.5x scaling of 4K. Yet it highlights a limitation in the comparison method of the Chapter. In practice the control points are clicked pixels which are discrete in nature.

The study was unable to assess the impact of the difference in discretisation, the key difference between 4K and HD.

The second comparison between the two HD assemblies found that the HD-PJ260VE hardware assembly gave superior reconstruction error to the hardware assembly used in the prior three chapters. This was because the intrinsic model was better able to fit to the lens distortion. Subsequently the next chapter will use the HD-PJ260VE hardware assembly, to investigate the effect of camera pose on the reconstruction error.

The results, when compared to those in Chapter 7, highlight a fundamental limitation of using a 1:100 scale model to assess the reconstruction error. At this scale it is very difficult to ensure an identical pose when repeating an experiment; the 1:100 scale is such that a small absolute difference in pose can result in a large relative difference. These small absolute differences in the pose are sufficient to give significantly different results. More repeatable results may be achieved by calculating the reconstruction error on a full scale hockey pitch; however this has other practical considerations, i.e. what is the error in the placement of the world known points?

9 Camera Pose

9.1 Introduction

Camera pose is the location and orientation of the camera relative to the world coordinate system. While at many tournaments a performance analyst is restricted to a specific camera pose, at some tournaments they may have a choice. Each camera pose will have a different view of the pitch and therefore captures a unique projection from world coordinates to image coordinates. It follows that the inverse transformation, the reconstruction from image coordinates to world coordinates, is also unique. Each of these unique transformations has a different associated reconstruction error.

(Brewin & Kerwin 2003; Hinrichs et al. 2005) investigated the effect of camera pose on reconstruction error, however both only considered 2D-DLT. Here (Kannala & Brandt 2006)'s camera model is applied, therefore this chapter investigates how the camera pose affects the reconstruction error.

This chapter addresses the objective: *"Assess the effect of camera pose on the reconstruction error".* The contribution to knowledge for this chapter is the effect of the camera pose on the reconstruction error. This knowledge could be used by a performance analyst to assess the reconstruction error given a camera pose, or to choose a camera pose from those available.

9.2 Method

In both dimensions, at intervals of 0.08 m, circles of diameter 0.04 m formed a 13 x 8 grid of 104 known points. This 0.96 m x 0.56 m scale model was printed onto a flat, rigid board. The study used the same control points as in Chapter 7. These control points are marked in red in Figure 9.1.

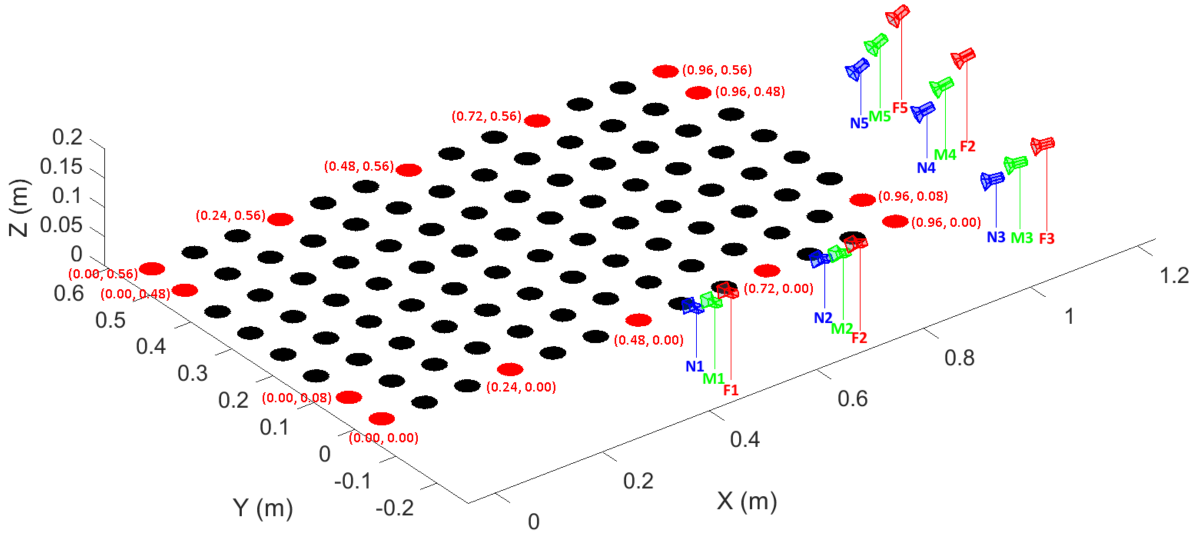


Figure 9.1: The camera poses considered in the study. The camera poses are colour coded by their designation: Near – Blue, Mid – Green or Far – Red. The world known points used as control points are indicated as red dots and labelled with their world coordinates.

The study considered the fifteen different camera poses illustrated in Figure 9.1. Due to the scale model’s two lines of symmetry, it was deemed unnecessary to consider any camera pose beyond the midpoint of either dimension. The camera poses were determined as a combination of two factors:

- The camera’s distance from the scale model. This is a combination of the camera’s distance set back from the scale model and the camera’s elevation. The camera poses available at a stadium are dependent upon its bowl profile. As stadia are built to meet certain bowl profile guidelines, there is a relationship between a point’s set back distance and its elevation. This relationship can be used to calculate reasonable camera poses. No stadia guidelines could be found for the International Hockey Federation, therefore the three distances were determined based upon the guidelines of FIFA (FIFA-Fédération Internationale de Football Association 2011). These distances are denoted in the camera pose label by a letter: **N** – near, **M** – mid or **F** – far.

- The camera's pose relative to the scale model. Five camera poses were chosen relative to specific markings on the field hockey pitch. These poses, denoted by a number were as follows:

    1. The midpoint of the X dimension – in line with the intersection of the half-way line and the side line.
    2. Three quarters of the way along the X dimension – in line with the intersection of the quarter line and the side line.
    3. The corner of the scale model – set back diagonally from the corner of the pitch.
    4. One fifth of the way along the Y dimension – in line with the intersection of the shooting circle and the goal line.
    5. The midpoint of the Y dimension – directly behind the goal.

The effect of each of these factors was investigated. Table 9.1 lists the coordinates of each camera location considered. Pose **N1** is similar to the camera pose used to capture the footage used in other chapters of this thesis.

Table 9.1. The location of the camera poses in the world coordinate system.

	N			M			F		
	X	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
1	0.48	-0.14	0.09	0.48	-0.19	0.12	0.48	-0.23	0.15
2	0.72	-0.14	0.09	0.72	-0.19	0.12	0.72	-0.23	0.15
3	1.07	-0.11	0.09	1.09	-0.13	0.12	1.12	-0.16	0.15
4	1.11	0.12	0.09	1.15	0.12	0.12	1.19	0.12	0.15
5	1.11	0.28	0.09	1.15	0.28	0.12	1.19	0.28	0.15

The data collection and analysis method were the same as presented in Chapter 8.

9.3 Results and Discussion

The box plot in Figure 9.2 displays the reconstruction error for each of the camera poses. Again the box plot follows the convention of (Tukey 1977).

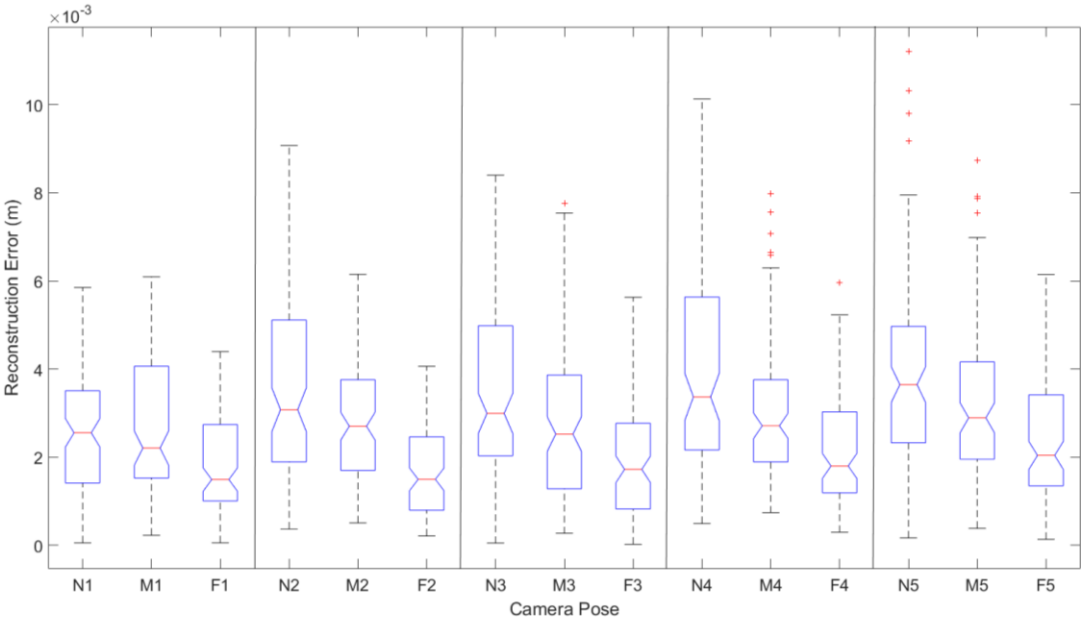


Figure 9.2: The reconstruction error for each of the camera poses. The box plots follow the convention of (Tukey 1977)**. The median value is indicated by the red line and the interquartile range (IQR) by the blue box. The extent of the whiskers denotes the highest datum within 1.5 IQR of the 3rd quartile and lowest datum within 1.5 IQR of the 1st quartile. Any datum outside the range of the whiskers is classed as an outlier and marked with a red cross.**

The results in Figure 9.2 indicate that for all camera poses the expected median reconstruction error is less than (McInerney 2017)'s acceptable error, adjusted for scale of the model. Yet for only poses **F1** and **F2** are all points below the acceptable value. Chapter 7 introduced the Acceptable Error Rate (AER), the percentage of points that were reconstructed with less than 0.005 m error. Figure 9.3 presents the AER for each of the camera poses. For all camera poses the AER is at least 70%.

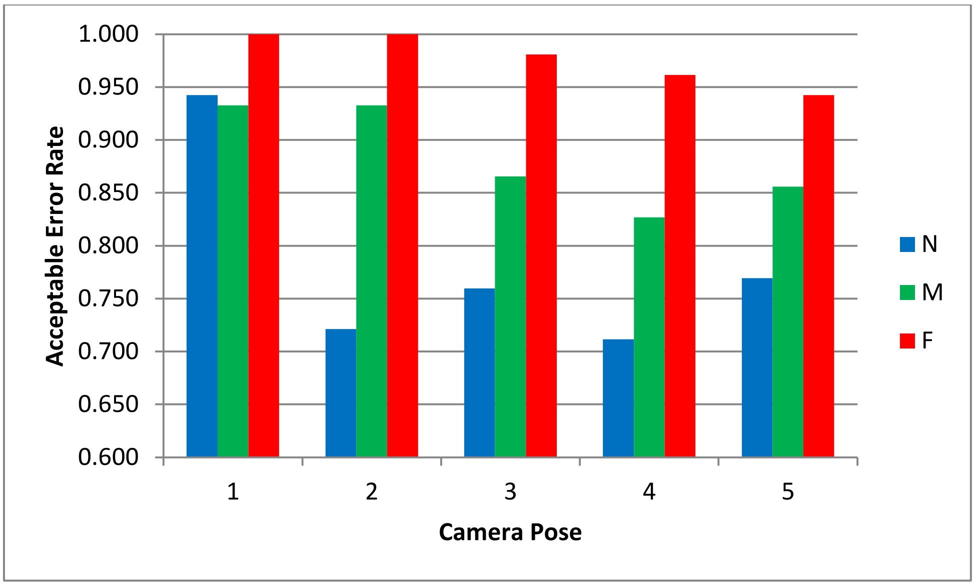


Figure 9.3: For each camera pose, the Acceptable Error Rate, the percentage of points that were reconstructed with less than 0.005 m of error.

There are two patterns in the median reconstruction errors presented in Figure 9.2:

1. The median reconstruction error decreases as the camera is moved further from the pitch.
2. The median reconstruction error increases as the camera is moved from the midpoint of the long dimension to the midpoint of the short dimension.

Given these patterns a performance analyst should be advised to locate the camera as far from the pitch and as close to the halfway line as possible. The remainder of this section will explore why the different camera poses have different reconstruction errors.

As noted previously, as the camera is moved further from the pitch the median reconstruction error decreases. One possible explanation for this is the reduced required angle of view. A wide-angle lens is designed to project an entire hemisphere as a finite circle (Shah & Aggarwal 1996) with the resulting projection subject to barrel distortion. (Kannala & Brandt 2006) model this distortion as a function of the angle between the principal axis and the incoming ray, the angle of incidence. An incoming ray with a higher angle of incidence will be more distorted. As the required angle of view is reduced, the maximum angle of incidence will be reduced and as such the maximum distortion will also be reduced. If the maximum distortion is reduced the complexity of the distortion is decreased and it follows that the distortion model can fit more accurately. This is supported by the projection error for the intrinsic calibration for each of the camera poses. Figure 9.4 displays the effect of focal length (fc) on the projection error. Focal length is used here as an approximate measure of the angle of view. A lower fc indicates a wider angle of view. The camera model allowed for different focal lengths in the u and v dimension. Here fc is the focal length in the u dimension. The estimated focal lengths for each camera pose are listed in Table 9.2.

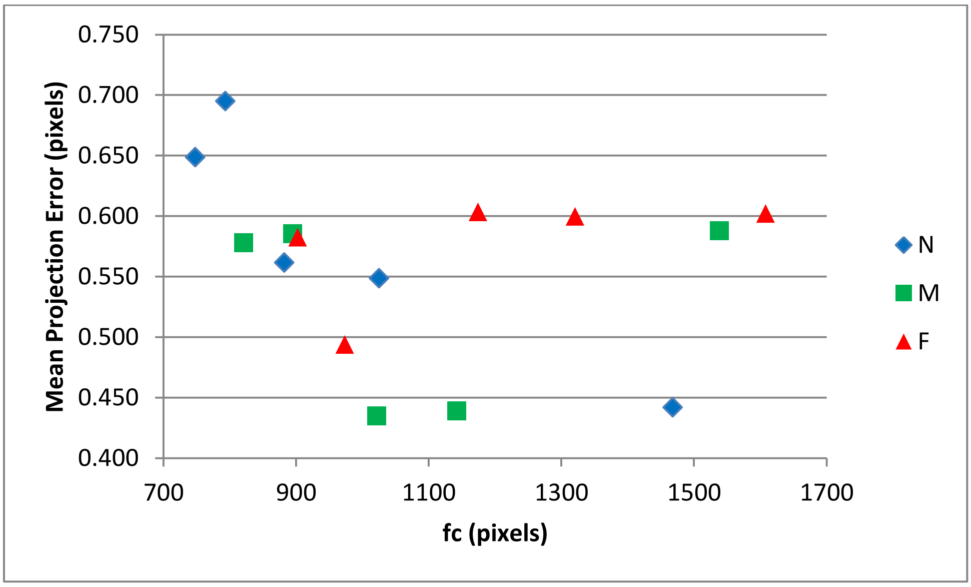


Figure 9.4: The relationship between the focal length (fc) in the u dimension and the mean projection error.

Table 9.2: The estimated focal length in pixels for each of the camera poses.

	N	M	F
1	748.03	821.01	902.43
2	793.37	895.14	973.74
3	1468.06	1538.67	1608.17
4	1025.07	1142.31	1320.81
5	882.34	1021.73	1174.36

As expected the mean projection error decreases as fc increases. This relationship appears to hold until $fc \approx 1200$ pixels. At this point the mean projection error jumps up to around 0.6 pixels. This is due to the use of a wide-angle lens convertor. The wide-angle lens convertor supplements the existing lens system. The existing lens system has a zoom range that is suitable for use with the convertor, beyond this range there are optical effects that the model is unable to account for.

Despite the evidence that the angle of view has an effect on the projection error, Figure 9.5 shows that the angle of view does not have a similar effect on the reconstruction error. This suggests it is not the angle of view but another factor that is causing the difference in median reconstruction error. I propose this factor is the range of the planar resolution of the circles in the model image. This will be discussed in the following sections.

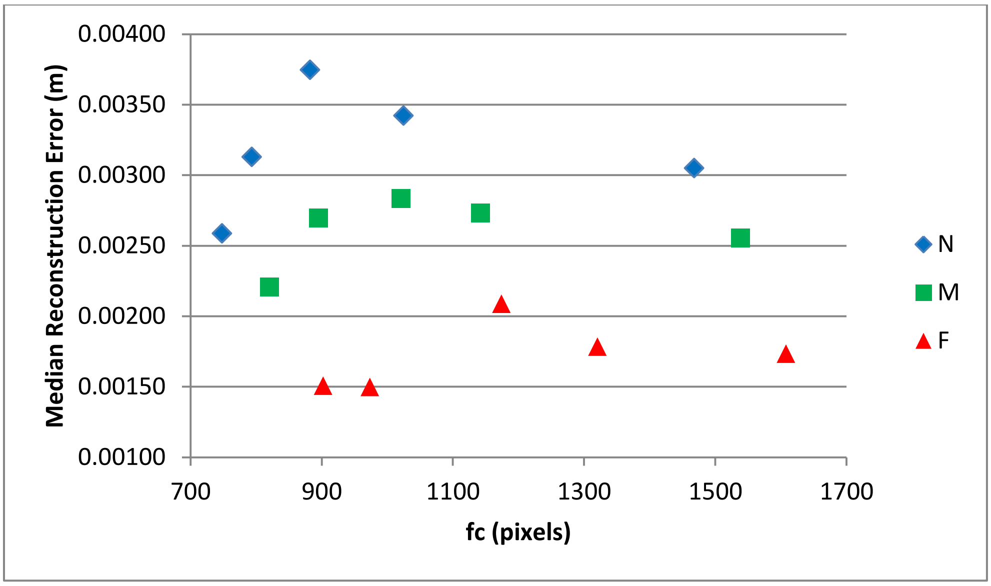


Figure 9.5: The relationship between focal length (*fc*) in the u dimension and the median reconstruction error.

Figure 9.6 displays the model images for the poses **N1**, **M1** and **F1**, while the magnified regions in the cut-outs focus on the 40 pixel by 20 pixel region around the control point at (0, 0.56). It is clear the range of the planar resolution decreases as the camera is moved further from the model. The planar resolution has two possible effects on the reconstruction error:

1. Error in the Control Point Extraction – The calculation of the reconstruction error relies on accurate control points. Any error in the control point extraction process will introduce error to the reconstruction error. In this study the control points were extracted from the set of circle centres. The circle centre extraction process fits an ellipse to the representation of each of the circles in the undistorted model image. It is the statistics of these ellipses that are used to estimate the circle centres. A circle with a lower planar resolution will have a smaller image representation. An ellipse fitted to a circle with a smaller image representation is more sensitive to each pixel; the removal of a single pixel has

a greater effect on the ellipse statistics. If it is assumed a larger image representation allows for more image detail, it follows that the fitted ellipse is more representative of the true circle and the extracted circle centre is more accurate. Still it does not follow that the extracted circle centres of progressively larger circles will be more accurate; it is more likely the accuracy increases rapidly but then soon plateaus. Given this, a smaller range in planar resolution implies that the planar resolution is more consistent across the whole calibrated plane, which will have the effect of reducing the mean circle centre extraction error.

2. Error in the Control Point Correspondences – Fourteen of the circle centres were used as control points. The control points are correspondences between image coordinates and world coordinates. To achieve an accurate extrinsic calibration a control point's image coordinate must be the true location of the known world point in the image. If it is assumed that an extracted circle centre contains error, then the image coordinate does not correspond exactly to the known world point (WP) but instead to the hypothetical extracted world point (EWP). The difference between the WP and the EWP introduces error to the extrinsic calibration. If a constant circle centre extraction error is assumed, a lower planar resolution will result in a larger difference between the WC and the EWC.

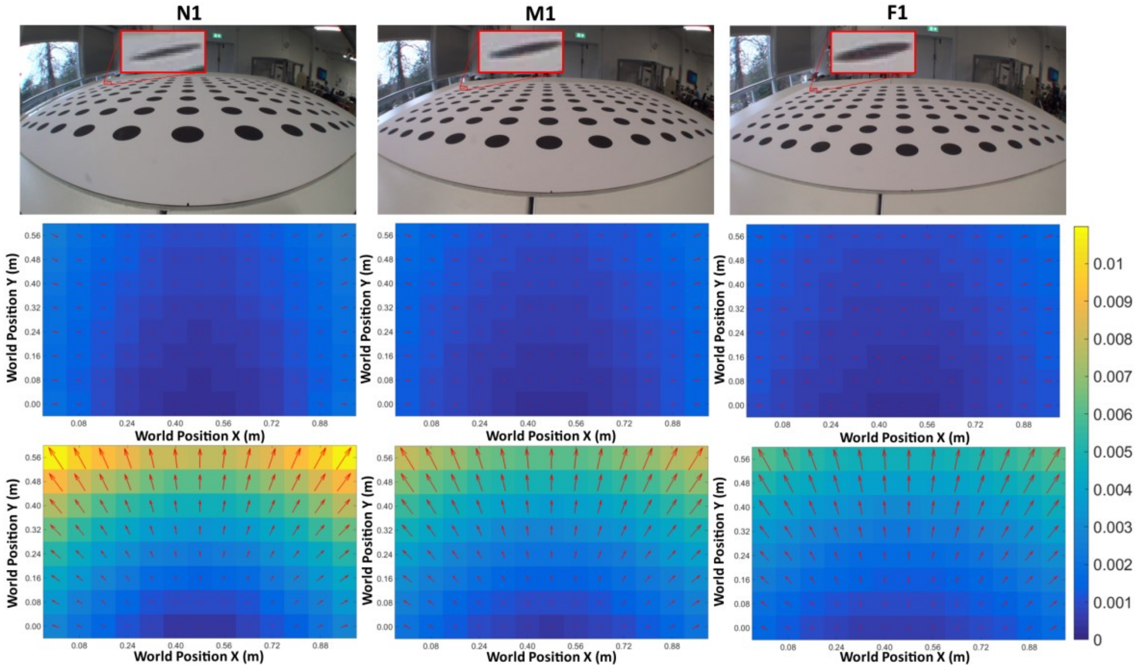


Figure 9.6: The effect of camera pose on the planar resolution. For each of camera poses N1, M1 and F1: Top Row – The model frame with a 40 pixels x 20 pixels cut-out that magnifies the control point at (0, 0.56). Middle Row – The effect of a one pixel change in the u dimension on the reconstructed coordinates of each circle centre. Bottom Row – The effect of a one pixel change in the v dimension on the reconstructed coordinates of each circle centre. The colours indicate the absolute magnitude of the difference. The arrows indicate the direction and the relative magnitude of the difference within a camera pose.

These two effects may explain why the reconstruction error decreases as the camera is moved further from the model and the range of planar resolution is decreased. More specifically it is an increase in the angle between the camera's principal axis and the calibrated plane that increases the planar resolution of the highlighted control point. This is a similar finding to the work in (Hinrichs et al. 2005), albeit this work uses Kannala and Brandt's camera model rather than 2D-DLT. An angle of incidence of 0°, an aerial view, will minimise the range of planar resolution across all the control points, therefore it could be hypothesised that this camera pose would minimise the reconstruction error. Further work could determine if the camera's angle of incidence is a key factor when determining the reconstruction error.

Figure 9.7 displays the model image for camera poses **N1-N5**. Again the cut-out displays the 40 pixels x 20 pixels around the control point at (0, 0.56). In **N3-N5** the highlighted circle appears almost as a horizontal line segment. This makes it hard to fit an accurate ellipse and as such extract accurate circle centres. The increase in reconstruction error is explained by effect number 2 above, error in the control point correspondences. Further work could investigate how the size and shape of the circle representation effects the circle centre extraction.

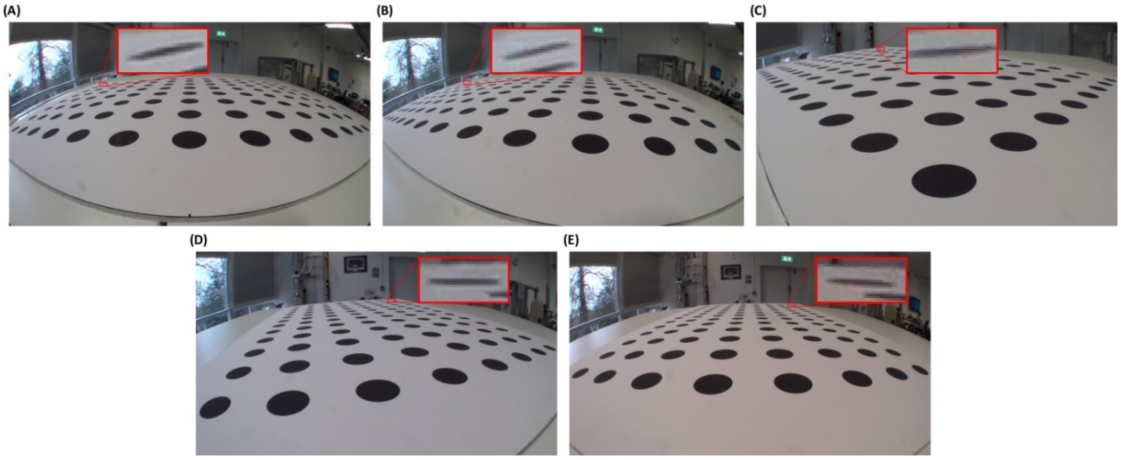


Figure 9.7: The model images for poses: (A) N1, (B) N2, (C) N3, (D) N4 and (E) N5. The cut-out in each image is 40 pixels x 20 pixels and magnifies the control point at (0, 0.56).

It could be argued that error in the control point extraction is an error in the assessment method and as such these results are not transferable to the real world. Yet the calibration in the real world also requires accurate control point correspondences and the resolution with which the image control points can be identified is dependent upon the planar resolution.

9.4 Summary

This chapter addressed the objective: *"Assess the effect of camera pose on the reconstruction error*." To do so, it investigated the reconstruction error for fifteen different camera poses. The camera poses were chosen to be representative of those a performance analyst may have at an international field hockey tournament. Therefore this chapter provides a novel resource for a performance analyst to determine the expected reconstruction error given a camera pose.

For all fifteen camera poses the median reconstruction error was below the 0.005 m deemed acceptable, however there was variation in the median reconstruction error. Therefore a performance analyst who wishes to minimise the reconstruction error should locate the camera:

1. As close to in line with the half way line as possible.
2. As far back from the pitch as possible.

The difference in the reconstruction error can be explained by the range in the planar resolution of the calibrated plane. Control points at a lower planar resolution have more potential for error due to:

1. Less accurate extraction.
2. Errors in the extraction introducing errors into the control point correspondences.

10 Player Extraction Accuracy

10.1 Introduction

Chapters 3 and 4 investigated parts of the algorithm necessary to extract player image coordinates from the scene. Chapter 3 investigated a method to extract player blobs from a wide-angle hockey video. Chapter 4 trained a Convolutional Neural Network to classify extracted blobs as a player or not a player. This chapter will combine these two chapters. First it introduces a method to reform over-segmented players. This method uses the Convolutional Neural Network class scores to assess the likelihood that different combinations of blobs are players. The combinations with the highest likelihood are considered for further analysis. Subsequently it assesses how accurately a player's image coordinates can be extracted from a wide-angle field hockey video. Here an accurate extraction is defined as an extracted point within some distance T of its corresponding expected point. The chapter will not consider the accuracy of the appearance of the player.

The work of this chapter could be combined with the world point reconstruction presented in Chapters 6-9 to assess the accuracy with which a player's world coordinates can be extracted. However, such a study requires accurate known world player coordinates. These are unavailable for the dataset used throughout this thesis. As the true world positions are not available, it is beyond the scope of this thesis to assess the accuracy of the complete algorithm. A future data collection could be completed that collects both video and player coordinates concurrently, however the researcher must consider: 1. the accuracy of the player coordinate collection procedure, and 2. how the player coordinates can be synced to the video.

This chapter addresses the objective: "*Investigate how accurately the player's coordinates can be extracted from wide-angle field hockey footage*". The novel contribution to knowledge for this chapter is an algorithm to extract the coordinates for all the players on a hockey field, using a single wide angle camera. This includes the introduction of a method to reform over-segmented player blobs. The chapter also assesses the accuracy of this algorithm given the camera pose at a recent international hockey tournament.

10.2 Method

The algorithm for the assessment of player coordinates extraction can be decomposed into three sub-algorithms: 1. ground truth dataset creation, 2. player extraction, and 3. analysis. In the first sub-algorithm the ground truth player coordinates are manually identified in a video. The second sub-algorithm is the workflow that extracts the player coordinates. The final sub-algorithm analyses the difference between the ground truth player coordinates and the extracted player coordinates. Each of the following subsections is dedicated to one of these sub-algorithms.

10.2.1 Ground Truth Dataset

Creation Procedure

Each ground truth dataset is formed from the player coordinates from one or more videos. For each video a set of frames of interest and a set of players of interest are defined. The coordinates of each player of interest in each of the frames of interest were identified. In this work, a player's coordinates is defined as the projection of their centre of mass onto the ground plane. If the player is stationary then it was assumed that this was the point midway between the player's feet. If the player is moving then this assumption does not hold and the point on the ground plane was estimated from the player’s pose. If the player’s feet were occluded by another player then no point was identified for this player in this frame and this datum is not included in the analysis. Therefore the procedure for the extraction of a video’s ground truth was as follows:

- For each video in the dataset
    - For each frame in the frames of interest
        - For each player in the players of interest
            - If the players coordinate is visible in the frame
                - Click the coordinate of the player on the ground plane

Datasets

Two datasets were considered in the study. Both datasets were created from videos in the EuroHockey Dataset (Section 3.2). The videos were captured in 4K (3840 pixels x 2160 pixels) from an approximate camera position relative to the pitch of X= 46 m, Y= -13 m and Z= 8 m. This is similar to camera position **N1** in the previous chapter. Despite the findings of Chapter 8, the 4K resolution was chosen to ensure that the player’s pixel representations were large enough for accurate segmentation. The camera was calibrated following the method outlined in Chapter 6.

The first dataset, the *Four Quarter Dataset*, considered the algorithm’s ability to extract players under different environmental conditions. These four quarters were chosen because of their diversity in weather and the teams represented. The weather has an impact on the colours in the scene, which may affect the ability to extract players. It can also cause shadows which can have an adverse effect on segmentation or player localisation, both of which will decrease extraction accuracy. Each team has a different playing uniform, each formed from different colours. The extraction accuracy may not be consistent across these different uniforms.

Each video was captured at 25 Hz. The frames of interest were the 100th to 349th frame. Starting the analysis at the 100th frame allowed the background subtraction algorithm to converge. Analysing 250 frames allowed adequate variance in the player's positions, while maintaining a reasonable manual identification time. All 22 players and the 2 umpires were included in the players of interest. Over the four quarters this gave an approximate total of 24000 ground truth coordinates.

This dataset was formed from the first 10 seconds of each quarter; subsequently, as observed in Figure 10.1, the players tend towards the centre of the pitch and there are very few coordinates in the more extreme corner regions. The high sampling frequency means player movement is low between frames. This results in the player coordinates that are globally sparse and locally dense. Therefore, while the dataset allowed comparison between different playing kits and weather, it is not ideal for assessing the overall accuracy of the algorithm.

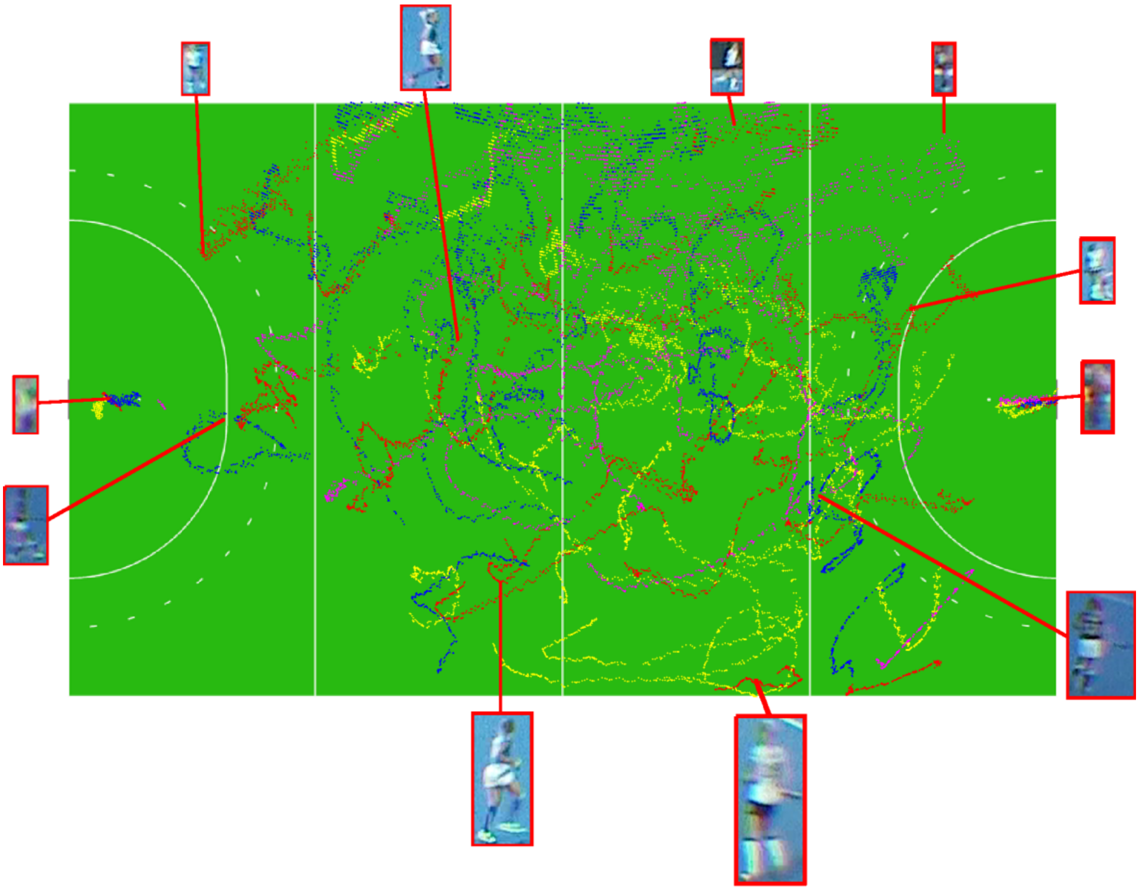


Figure 10.1: The player coordinates for the *Four Quarter Dataset. The cut-outs indicate the relative size of a player at different locations on the pitch.*

The second dataset, the *Whole Quarter Dataset*, tried to address the limitations of the first by extracting more diverse player coordinates. For the entirety of a single quarter, the image coordinates of four players from a single team were manually extracted at 5 Hz. Two defenders and two forwards were chosen to ensure full pitch coverage. The quarter was 16 minutes and 42 seconds or 25050 frames. The first 100 frames were disposed of to allow the background subtraction algorithm time to converge; therefore there was a total of 4990 analysis frames. During the quarter the players left and re-entered the field via rolling substitutions; subsequently there were a total of 14779 player coordinates.

It took approximately 10 hours to perform coordinate identification for the entire quarter for each player. Assuming a full set of players for both teams, it is clear that manually identifying the coordinates of all the players is not viable for the performance analysts.

Figure 10.2 displays the player coordinates for the *Whole Quarter Dataset*. The coordinates are sparser than in the Four Quarter Dataset, yet it gives better coverage in the left most and right most quarters.

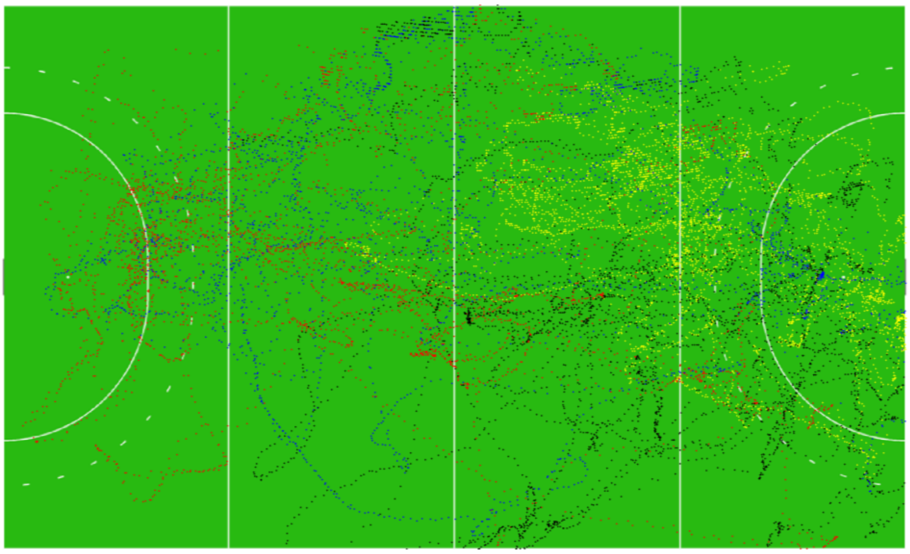


Figure 10.2: The player coordinates for the *Whole Quarter Dataset*

10.2.2 Automatic Player Position Extraction

This section highlights the algorithm for automatic player coordinate extraction for each frame. First the frame is segmented using the background subtraction method introduced in Chapter 3. This gives a set of candidate blobs that that can be categorised as: noise, correct player, over-segmented player, under-segmented player and multi-player.

The second step addresses the problem of over-segmented players. Extracted blobs that are spatially close together are grouped to form 'super'-blobs. The grouping algorithm is as follows:

1. For each blob:
    a. Calculate bounding box.
    b. Extract its image coordinate as the midpoint of the base of its bounding box.
    c. Use the procedure outlined in Chapter 6 to transform the image coordinate into a world coordinate.
    d. Calculate the world coordinate’s distance from the camera.
    e. Estimate the expected height and width of a player at this distance from the camera to create an expected bounding box. The functions to estimate height and width were calculated empirically and were found to be proportional to the reciprocal of the point’s distance from the camera.
    f. As illustrated in Figure 10.3, create three possible bounding boxes aligned with the base of the blob’s bounding box:
        i. One possible bounding box is centred on the midpoint of the blob’s bounding box.
        ii. One possible bounding box is aligned with the left edge of the blob’s bounding box.
        iii. One possible bounding box is aligned with the right edge of the blob’s bounding box.

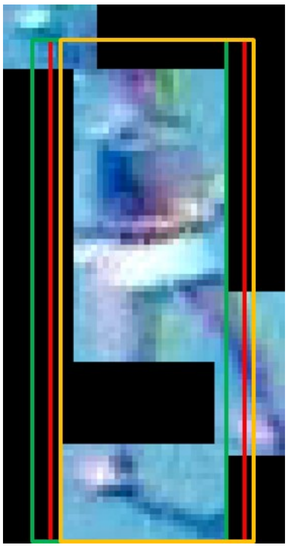


Figure 10.3: The three possible bounding boxes for the blob at the base of the image. Each possible bounding box is aligned with the base of the blob's bounding box. The red possible bounding box is centred at the midpoint of the blob's bounding box. The yellow possible bounding box is aligned with the left edge of the blob's bounding box. The green possible bounding box is aligned with right edge of the blob's bounding box.

   g. Add the three possible bounding boxes to a set of all possible bounding boxes.

2. For each possible bounding box create a binary vector of its intersection with each blob. For speed of computation this was implemented as the intersection with the blob's bounding boxes. Better results may be achieved by finding the intersection with the blob's actual boundary.
3. The initial blob may be well segmented, as such append an identity matrix with length that of the number of blobs to the set of possible binary vectors.
4. Suppress any identical binary vectors to create a set of candidate binary vectors. An identical binary vector means the possible bounding box overlaps with the same set of blobs; this is the case for all the possible bounding boxes in Figure 10.3.

5. For each candidate binary vector, create its corresponding candidate bounding box. The candidate bounding box is formed as the bounding box for all the blobs in the candidate binary vector.
6. As in Chapter 4, each bounding box was padded to make it square. The padding was always applied to retain the centre of the base of the bounding box.
7. Extract the set of candidate images bounded by each of the padded bounding boxes.
8. Resize each image to be 224 pixels x 224 pixels.
9. Use the Convolutional Neural Network (CNN) trained in Chapter 4 to infer the class scores for each of the images.
10. Each blob may be included in more than one candidate image but can only be assigned to one player, therefore the class scores and number of blobs that form the image were used to determine which candidate was superior:
    a. For each candidate image get the maximum score and the corresponding maximum class.
    b. Dispose of any candidate images where the maximum class is the non-player class.
    c. Sort the list of maximum classes by the maximum score and then the number of blobs in the candidate image. Candidate images with a high score and that are composed from multiple blobs are preferred.
    d. While the list of maximum classes is not empty:

i. Take the element at the top of the list and add it to the set of extracted images.

ii. Remove from the list any candidate images that share a blob with the element.

The final step in algorithm extracts the image coordinates as the midpoint of the base of each of the extracted images.

10.2.3 Analysis

In this work the algorithm was assessed using the recall given by Equation (32),

$$recall_m = \frac{num\ assigned\ with\ cost\ less\ than\ m}{num\ expected\ positions} \qquad (32)$$

Where m is the maximum distance between an expected coordinate and an extracted coordinate. This value can be presented as a graph for a range of m. It is therefore unnecessary to select a specific threshold before the calculation.

The precision is also reported for the *Four Quarters Dataset*, given by Equation (33).

$$precision_m = \frac{num\ assigned\ with\ cost\ less\ than\ m}{num\ extractions} \qquad (33)$$

The precision cannot be reported for the *Whole Quarter Dataset* as only a subset of the players was considered. Ergo many of the extractions were not assigned to ground truth positions despite the possibility that they were correct.

The per frame matching of I expected player coordinates with J extracted player coordinates was treated as an assignment problem and solved using the Hungarian algorithm (Kuhn 2010). The I x J cost matrix was formed using the Euclidean distance between the extracted player coordinates the expected player coordinates. The cost of non-assignment was set to 1000, a value larger than the maximum possible distance between extracted and expected coordinates; therefore, if possible all extracted player coordinates were assigned to expected player coordinates irrespective of the distance between the two.

Despite this chapter assessing the ability to accurately extract player coordinates from the image, the assessment itself must be performed in the metric world space. As demonstrated in Chapter 5, the metric area accounted for by one pixel is inconsistent across the pitch. Performing the assessment in image coordinates would bias the calculation to those players in positions with low metres per pixel. Therefore both the expected image coordinates and the extracted image coordinates were transformed to world coordinates before the calculation. This transformation has the benefit of meaning that m is metric and has a physical meaning. The transformation was performed using the method outlined in Chapter 6 and used the same intrinsic and extrinsic parameters for both sets of image coordinates. It was assumed that points were reconstructed without error.

Assessing in world coordinates gives the further option of assessing the extraction accuracy dependent upon the player's position on the pitch. Subsequently, as illustrated in Figure 10.4, the pitch was divided by an 8 x 5 grid. This gave 40 sectors, each of which was approximately 11 m^2. For the points in each sector, Equation (32) was applied with an m of 0.5 m. As earlier in this thesis, value of m was set at 0.5 m based on (McInerney 2017).

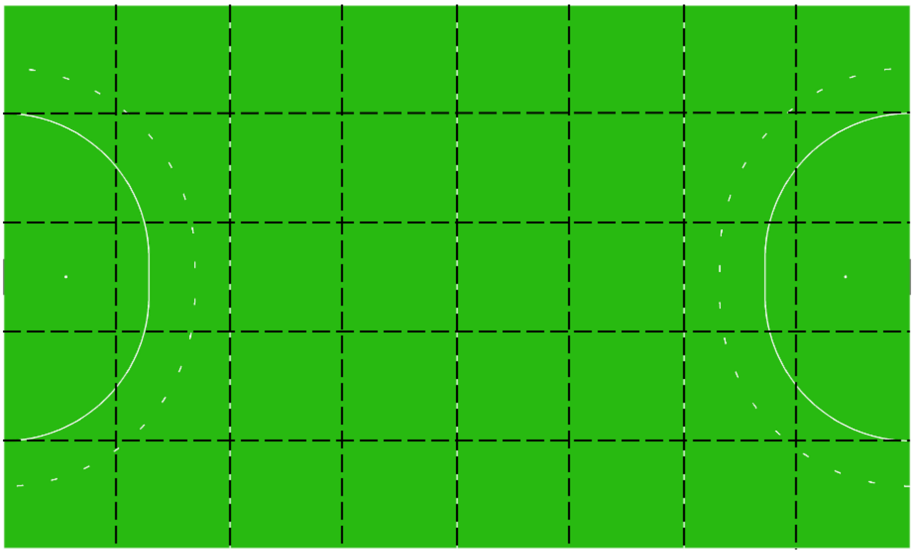


Figure 10.4: The pitch divided by an eight x 5 grid. Each sector is approximately 11 m^2.

10.3 Results and Discussion

Figure 10.5 displays the recall for the two datasets and the precision for the *Four Quarters Dataset*. Both datasets give a similar recall with about 50 % of the expected coordinates being extracted within 0.5 m. This increases to approximately 66% after 1 m after which both taper off. The precision on the Four Quarters Dataset is approximately 75% at 1 metre. This is a superior performance to (Carr et al. 2012), who report a precision of 55% and recall of 50% at a tolerance of 1 metre when using multiple cameras. This result suggests that the coordinate extraction algorithm is achieving state-of-the-art performance despite using a single camera.

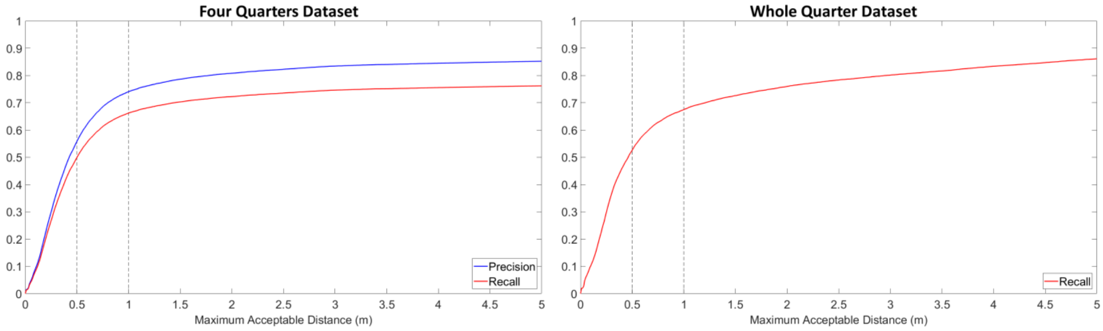


Figure 10.5: For each dataset, the recall of the expected image coordinates that are matched with a detected image coordinate given a maximum acceptable distance of m. For the four quarter dataset the precision is also included. m = 0.5 m and m = 1 m are marked with a dashed line.

The characteristics of the recall lines can be attributed to two main factors: incorrect extraction and non-extraction. Incorrect extraction describes the error when a detection has been made but the extracted coordinate is not in the correct position. It results in relatively small errors due to the midpoint of the bounding box not aligning with the true player position and can explain the rapid rise in the recall between 0 and 1. Non-extraction errors occur when no detection has been made and as such there can be no extraction. This may be due to non-segmentation or misclassification as a non-player by the CNN. It is this error that causes the tapering of the recall, which is in effect an artificial limit on the maximum performance. This artificial limit can be linked back to the results in Chapter 3 and 4. Chapter 3 reported that 61% of the blobs are well segmented and a further 15% are over-segmented. Assuming that all over-segmented players can be resolved, this would mean approximately 76% of blobs were well segmented. However in Chapter 4, the results on the test set suggested that 3.1% of player blobs were classified as non-players. This further reduces the number of players that have been extracted to approximately 73.5%, similar to the artificial limit suggested by the tapering.

Comparing the two datasets, the recall of the *Whole Quarter dataset* is slightly superior between zero and one, after which the two lines diverge. This is possibly due to incorrect matches. As noted by (Bernardin & Stiefelhagen 2008) the closest match is not necessarily the correct match. The *Whole quarter dataset* only expected four of the players in the quarter, yet all the players were extracted. Subsequently if the actual match of one of the expected players was not extracted, it may match with a spatially close but incorrect extraction. For low m it is unlikely that such a match will occur because two players cannot occupy the same space, however as m increases so does the likelihood that there will be another incorrect extraction within range.

Figure 10.5 also displays the precision for the *Four Quarter Dataset*. This is the ratio of extractions that are assigned to a ground truth given a maximum assignment distance. The graph shows that approximately 60% of extractions are within 0.5 metres of a ground truth position and as many as 75% of extractions are within 1 metre. Alternatively this means that 25% of extractions are noise assuming a maximum assignment distance of 1 m.

Figure 10.6 illustrates, for each dataset, the recall for each sector, given m = 0.5 m. If the cell is white then no coordinates were expected in this sector. Both datasets exhibit a pattern of the recall decreasing as the sector is further from the camera. In the sectors closest to the camera a recall of between 70% and 80% can be expected however in the most extreme regions this reduces to close to zero. This suggests that with the current hardware and approach a single camera vision based system is unsuitable for highly accurate whole hockey pitch tracking but could be used over a smaller area. The following paragraphs propose reasons the recall is poor in these sectors.

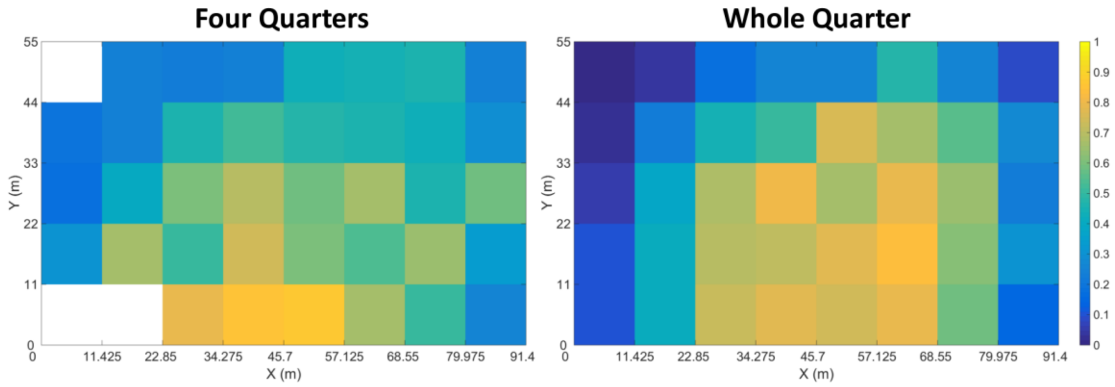


Figure 10.6: For each dataset, the recall (m = 0.5) for each of the sectors of the pitch. If the cell is white then no coordinates were expected in the sector.

One possible reason that the recall is lower in regions further from the camera is an increase in the uncertainty of the ground truth coordinates. The uncertainty of a ground truth coordinate is a result of the following factors: discretisation of the image space, the image resolution of the player and the projection of the player's centre of mass to the ground plane. The following paragraphs review these factors.

The ground truth image coordinate was the pitch pixel that was the downward projection of the player's centre of mass. The manual identification of these points was performed in the discretised pixel space. Assuming the true point can be perceived perfectly, the discretisation means it has a maximum discretisation error of ± 0.5 pixels. As shown in Chapter 5, a pixel further from the camera accounts for a larger distance than a pixel closer to the camera. Therefore, the potential error of the ground truth coordinates is larger the further from the camera. Now assume that the proposed algorithm extracts a player's coordinates without this same discretisation error; the recall will be lower for points further from the camera simply due to the larger potential error in the ground truth coordinates. The discretisation of the pitch is evident in Figure 10.1. In the lower parts of the pitch, those closest to the camera, the points form smooth trajectories. Whereas in the most extreme upper parts of the pitch, those furthest from the camera, the points form discrete, stratified lines.

The previous point assumed that the player’s ground truth coordinates can be identified perfectly; this is not the case. The coordinate must be estimated from the available image evidence. It is clear from Figure 10.1 that the resolution of the player’s image representation decreases as the player is moved further from the camera. The higher the resolution, the more detail from which to infer the player’s image coordinates and in theory the closer to the true point.

A player’s image coordinate was assumed as the projection of the player’s centre of mass onto the ground plane. As noted in the method section this point was estimated from their pose; however there is error in this estimation process. The oscillations in some of the trajectories in Figure 10.1 are evidence of this error. The oscillations are due to the step from one leg to the other and the amplitude increases as the player travels faster. The oscillations are less apparent in Figure 10.2 as the sampling frequency is reduced to 5 Hz. The previous two points are dependent upon distance from camera, whereas this is independent and affects all sectors of the pitch.

Each of the outlined factors introduces error to the ground truth coordinates of a player. Some of these errors may have been reduced by filtering the ground truth data; however that was not considered in this thesis.

Another possible reason that the recall is lower in regions further from the camera is the method for transforming from the segmentation to image coordinates. In the current implementation the player’s image coordinate is the midpoint of the base of the segmented blob's bounding box. The midpoint is found as the sum of the bounding boxes sides divided by two. One weakness of this implementation is that the space of possible image coordinates is discretised, with a difference between values of 0.5 pixels. As with the discretisation of the ground truth coordinates, this causes a discretisation error which when transformed to world coordinates is larger the further the point from the camera. One solution to reduce this error would be to increase the resolution of the camera. This would decrease the metres per pixel and as such reduce the effect of this discretisation. This is unfeasible with current consumer technology.

Estimating the player's coordinate from their bounding box also decreases the accuracy irrespective of the player's position. Consider a player holding their stick such that there is a long thin protrusion from the blob; this protrusion will be included in the bounding box and the image coordinate will be shifted to compensate for it (Figure 10.7). Instead better results may be given by using the image moments to calculate the centre of mass of the blob and projecting this point down onto the base of the bounding box. Yet projecting down onto the base of the bounding box implicitly assumes that the player is stationary or travelling perpendicular to the camera. If, as is more likely the case, the player is travelling in some other direction, i.e. one foot is further from the camera than the other, then the base of the bounding box is not a good estimate for the projection of the player's centre of mass. In this case, better results would be achieved by more accurately estimating the projection of the player's centre of mass.

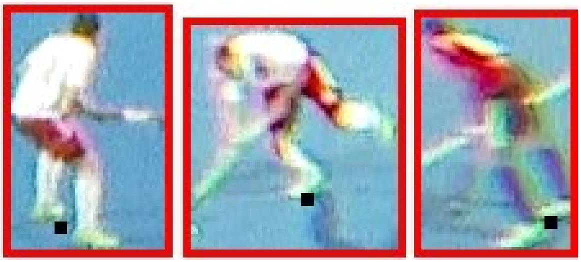


Figure 10.7: Examples of when the centre of the base of the bounding box does not coincide with the player's true coordinates. The black dot indicates the player's true coordinates.

A third reason that the recall is lower in regions further from the camera is the size of the player in the frame. As indicated in Figure 10.1, the area of a player in the frame decreases as they are positioned further from the camera. It follows that the detail of their image will decrease, which in turn decreases the likelihood of an accurate segmentation. The accuracy of the extraction relies upon having an accurate segmentation.

In addition to this the CNN used to classify image patches as players, requires input images to be 224 pixels by 224 pixels. For an image that has a small initial size the up-sampling procedure can lead to a blurry input image (Figure 10.8). A blurry image may lead to misclassification of the image as noise and its removal from the analysis. Further analysis is needed to determine how the up-sampling procedure affects the CNN's ability to correctly classify images.

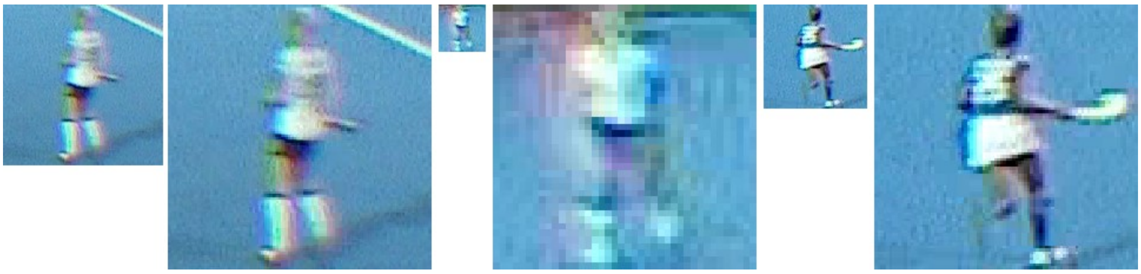


Figure 10.8: Examples of the up-sampling necessary for input to the CNN. Small Images: The relative size of the original image. Large Images: The original images all up-sampled to 224 pixels x 224 pixels.

The *Four Quarter Dataset* was designed to allow comparisons between environmental conditions and teams. Therefore, for each quarter of the *Four Quarter dataset*, each of the players was classified into one of the following groups: outfield Team A, outfield Team B, goalkeeper Team A, goalkeeper Team B and Umpires. Figure 10.9 compares the recall for the eight outfield groups across the four quarters. The goalkeeper and umpire classes are omitted because of the relatively low number of examples in each class. The maximum difference in recall between teams is 10% when m is 0.5 m and increases to 15% when m is 1 m. As there was normally 24 players in a frame this 10% difference in recall results in approximately 2.5 extra extractions per frame.

The difference in the recall could be explained by the distribution of players across pitch. As explained previously the recall is highly dependent upon the position of the players on the pitch. In the quarters that have a higher accuracy, the mean player position may be closer to the camera. In addition to this, the data is sampled from a time series at 25 Hz. At this high frequency, samples S_t and S_{t+1} are highly correlated. This means that if the coordinate cannot be extracted accurately for S_t, there is an increased likelihood that the coordinate cannot be extracted accurately for S_{t+1}.

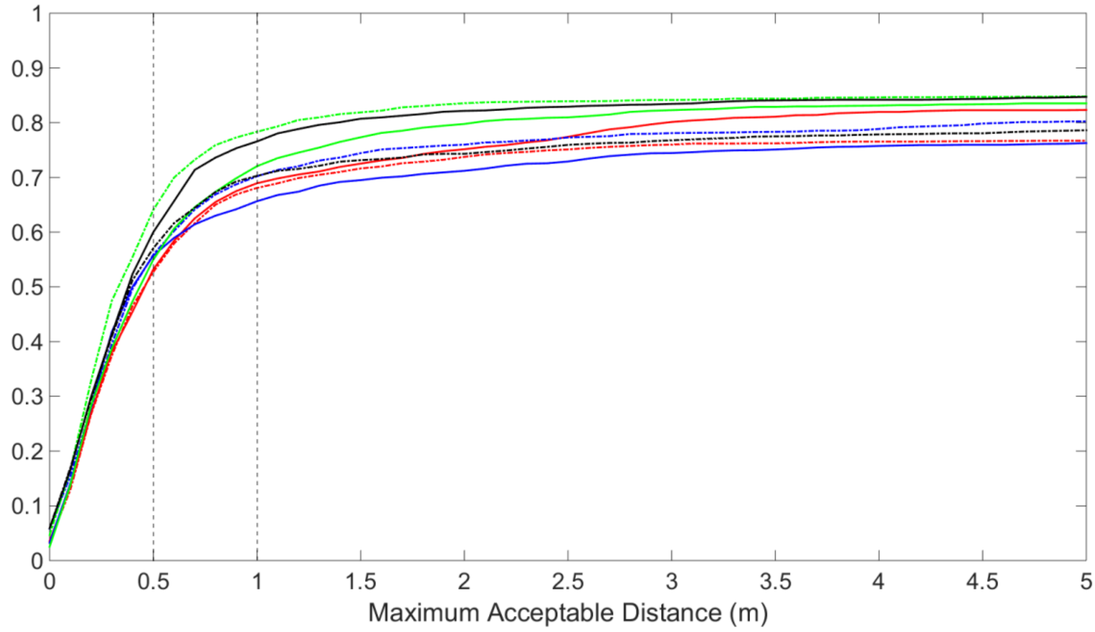


Figure 10.9: For each quarter in the Four Quarters dataset, for each outfield class, the recall of the expected image coordinates that are matched with a detected image coordinate given a maximum acceptable distance of m. Each quarter is illustrated in a different colour. For each quarter, one team is illustrated by a solid and the other by a dashed line. m = 0.5 m and m = 1 m are marked with a dashed line.

A second possible explanation for the differences in recall for the quarters is that the algorithm is less robust to some kit colours under some environmental conditions. The next paragraphs will attempt to explain some of the differences in recall given the different conditions.

The quarter indicated by the red line displays the worst recall. This quarter was played under floodlights located at the four corners of the pitch. Subsequently each player has four cast shadows, which if incorrectly segmented change the shape of the bounding box (Figure 10.10). As described above, these shadows can cause a protrusion which shifts the midpoint of the bounding box. An attempt was made to remove shadows using the expected Hue, Saturation and Value (HSV) of the background; however its description is beyond the scope of this thesis.

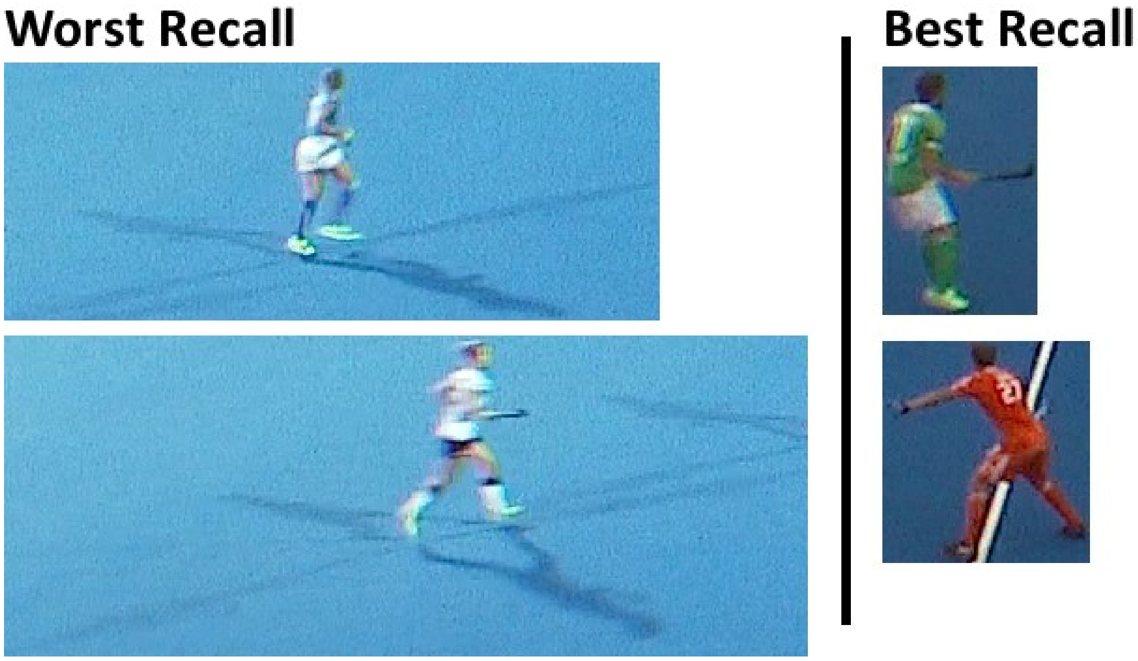


Figure 10.10: Example players from the quarters that gave the worst and best recalls in the *Four Quarter Dataset.* The quarter with the worst recall was played under floodlights, subsequently there is four cast shadows for each player. However, it is promising that even though one team played in blue the recalls for both teams were similar. The best recall occurred during a quarter where the sky was overcast. The teams were also wearing green and orange so contrasted well with the blue pitch.

A further observation from this quarter is related to the colours of the team's kits. One team wore a white kit and the other a blue kit. It is promising that the two teams give very similar recall as it suggests the algorithm is able to handle the blue kit despite the pitch colour also being blue.

Both teams in the green quarter exhibit good recall, suggesting that in general good segmentation was achieved. This could be explained by the conditions. The quarter was played under an overcast sky in the middle of the day, which meant there were no cast shadows (Figure 10.10). The team with dashed green line gave the best recall. This team wore a bright orange kit, which is very different from the blue pitch colour. Subsequently it is assumed that the players segmented very accurately which lead to accurate player coordinate extractions.

10.4 Summary

This chapter addressed the objective: *"Investigate how accurately the player's coordinates can be extracted from wide-angle field hockey footage."* To do so, it introduced a novel algorithm to automatically extract player coordinates. This included a novel method to reform over-segmented blobs into players. This method is based upon the class likelihood for 'super'-blobs. The accuracy of the algorithm was assessed by comparing the automatically extracted coordinates with manually identified player coordinates.

The manual coordinate identification for a single player at 5 Hz for a 15 minute quarter took approximately 10 hours. It is therefore unreasonable for a performance analyst to manually identify the coordinates for all the players in a tournament and an automatic solution is necessary.

The accuracy was assessed using the recall and precision of the extracted player coordinates that were within a distance m, of the expected player coordinates. The comparison was performed in world coordinates to give m a physical meaning.

An m of 0.5 metres resulted in a recall of approximately 50% and precision of 55%. An m of 1 metre resulted in a recall of approximately 66% and precision of 75%, a respective 16% and 20% improvement on the state-of-the-art.

The errors in recall can be attributed to: Incorrect extraction and non-extraction. Incorrect extraction occurs when the midpoint of the bounding box does not align with the expected player coordinates. Non-extraction occurs when no detection is made for a particular player.

Assuming a constant m of 0.5 metres, the recall decreases the further a point from the camera. This is attributed to both a decrease in the accuracy of the ground truth coordinates and a decrease in the accuracy with which a player's coordinate can be extracted.

Different quarters displayed different recall, suggesting that the environmental conditions and the playing kits of the teams may have an effect on the accuracy that can be achieved.

A limitation of this work, is the lack of a comparison between ground truth world known points and the extracted world points. This comparison relies on having accurate ground truth world known points, something which is unavailable for dataset used here. Instead this chapter compared the projection to the calibrated plane of the ground truth image coordinates and the extracted image coordinates. A future data collection could be completed that collects both video and player world coordinates concurrently, however the researcher must consider: 1. the accuracy of the player coordinate collection procedure, and 2. how the player coordinates and the video can be synchronised.

11 Conclusion

In elite level sport, coaches are always trying to develop tactics to better the opposition. In a team sport such as field hockey, a coach must consider both the strengths and weaknesses of both his own team and that of the opposition to develop an effective tactic. (Leser et al. 2011) state that spatiotemporal metrics are a key tool in the performance assessment of field sports. This is supported by the work of (McInerney 2017), who identified a set of performance metrics that can be used to predict team success in elite level field hockey.

Radio frequency systems can provide spatiotemporal metrics, however the necessity to wear a transponder means that often data for both teams is not available. Instead cameras provide an un-intrusive solution without the need for the players to wear transponders.

Given a video of a hockey match, a performance analyst could manually identify the player coordinates, yet this work has shown that to identify the coordinates for a single player at 5 Hz can take 10 hours per quarter. Subsequently it is unreasonable for a performance analyst to identify the coordinates for all the players involved in a match.

An alternative solution would be to automate the extraction of the players' coordinates. Existing commercial systems attempt to do this by using computer vision techniques; however these systems do not meet the constraints of an elite hockey tournament.   At an elite hockey tournament a performance analyst is restricted to a single camera position, the location of which is not consistent across tournaments. The performance analyst therefore needs the ability to extract the players' coordinates across the whole pitch but with the flexibility of the solution working from any reasonable camera position.

Consequently, the aim of this thesis was to:

Develop an algorithm to extract player coordinates from footage captured with a single wide-angle camera at a field hockey tournament.

To achieve this aim the algorithm was divided into two sub-algorithms: Player Feature Extraction and Reconstruct World Points. This is highlighted in Figure 11.1. Player coordinates form one of the inputs to the subsequent sub-algorithm, Trajectory Formulation. However as none of (McInerney 2017) performance metrics rely on trajectories; this was deemed beyond the scope of the thesis.

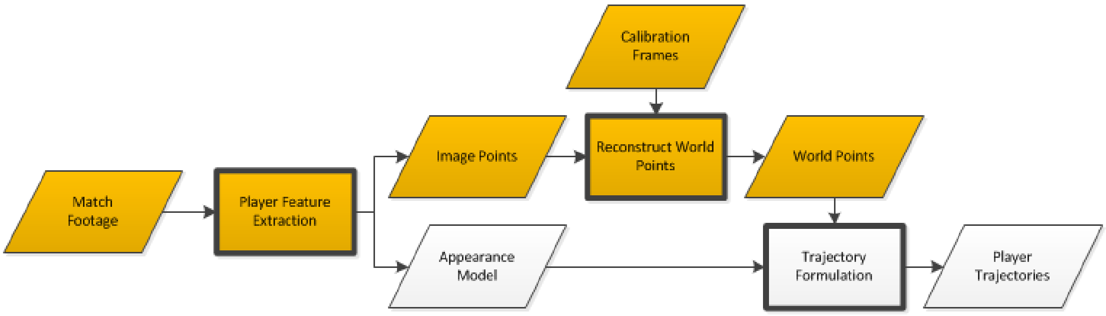


Figure 11.1: Orange: The scope of this thesis, the algorithm to automate the process of player coordinates extraction. Data is indicated by parallelograms. Sub-algorithms are indicated by emboldened rectangles. The algorithm is composed of two sub-algorithms: Player Feature Extraction and Reconstruct World Points. White: How coordinates extraction may fit into a Multi Object Tracking framework.

Chapters 3 and 4 were dedicated to the sub-algorithm that extracted image points from frames of a video. Chapter 3 developed a method to accurately segment accurate player regions in a frame. This method, the Temporal Median (Cucchiara et al. 2003), models a per pixel expected background by sampling the median over the preceding frames. The parameters of the model were optimised for a field hockey dataset in

(Higham et al. 2016). This method was able to correctly segment 61% of blobs in the dataset, while a further 15% were over-segmented. As the extraction of a player's coordinates requires segmentation of their blob in the scene, the practical findings of this chapter mean that the upper bound of correct player coordinates extraction is 76%.

Chapter 4 investigated if a Convolutional Neural Network (CNN) could be trained to classify the contents of an image for hockey player recognition. The four classes considered were: single player, single player with bottom poorly segmented, multiple players and non-player. Two different approaches were investigated. The first trained a Support Vector Machine on the output of a pre-trained CNN. The second fine-tuned an existing CNN for the task. Fine-tuning the network outperformed the SVM approach and had a classification error of 14.1% on the test set. Part of this error is accounted for by the similarity between the classes and may be resolved with more data.

While this work was carried out on field hockey dataset, the Temporal Median algorithm and the Convolution Neural Network do not exploit anything specific to field hockey; therefore these techniques could be applied in any field sport. The parameters listed were tuned for player segmentation on a blue field hockey pitch, so there may be inferior results given a different playing field. Better results may be achieved by tuning the parameters for the specific dataset.

The output of the first two chapters was a set of image coordinates marking the player's positions in a sequence of frames. The Reconstruct World Points sub-algorithm takes these image coordinates and transforms them into world coordinates.

Chapters 5-9 investigated this transformation and the effect of different conditions on it.

The assessment of reconstruction accuracy requires a set of known world points and their corresponding points in the image. Chapter 5 used a 1:100 scale model to show that from the expected camera position, the centre of a circle is an appropriate method to demarcate the known world points. A scale model was used as it meant world known points could be placed with machine precision and that the camera poses tested were not restricted by stadium access.

Due to lens distortion and the perspective transform, the centre of a projected circle is not the centre of the original circle. A method was proposed that used the grid structure of the known points to extract the circle centres. A limitation of this work is that the extracted circle centres cannot be assessed objectively because the true image points of the circle centres are unknown. This unaccounted for error adds uncertainty when computing the reconstruction error.

Instead a novel visual pattern was designed, which allowed a qualitative assessment to show the returned circle centres were aligned with the correct position in the image. This method was used in the subsequent chapters to extract the circle centres used to assess reconstruction accuracy.

Traditionally image coordinates are reconstructed to world coordinates using the inversion of the projection from world points to image points. Chapter 6 showed that for a camera pose with a high angle of incidence to the calibrated plane, like those a performance analyst experiences at international hockey tournaments, this method of reconstruction is sub-optimal due to the difference in planar resolution across the calibrated plane. Subsequently the chapter proposed estimating this transformation directly by minimising the control point reconstruction error. Results suggest that the proposed method results in a lower mean reconstruction error and less variance in the reconstruction error across the calibrated plane. For a performance analyst this means that on average the extracted coordinates are more accurate, and that the error associated with converting from image coordinates to world coordinates is more consistent wherever the player is on the pitch. In addition, the change is algorithmic so they do not need to adapt their current data collection procedure. This method was used to reconstruct points in the following chapters.

Chapter 7 explored the expected reconstruction accuracy for the camera pose granted to the performance analysts at a recent international field hockey tournament. For a 1:100 scale model the median reconstruction error was 0.0043 m and the distribution of errors had an interquartile range of 0.0025 m. The Acceptable Error Rate was defined as the percentage of points that were reconstructed with less than 0.005 m of error and was found to be 63.5 %. Alternatively this means that 36.5% of the points are not within the 0.005 m deemed to be acceptable.

Chapter 8 investigated the effect of camera resolution on the reconstruction accuracy. Two different comparisons were made. The first found that the reconstruction accuracy is similar for footage captured with the same camera setup but at HD and 4K resolutions. Due to the continuous nature of the circle centres, HD is just a 0.5x scaling of 4K and similar results should be expected. As the control points were not manually selected the study was unable to assess the effect of the discretisation due to the resolution. The second comparison found that an alternative camera assembly gave better reconstruction accuracy. As such this camera assembly was used in Chapter 9.

Chapter 8 also identified a limitation of using a 1:100 scale model when assessing the reconstruction error. Chapter 8 was a repeat of the experiment in Chapter 7; however the results were considerably worse. At this scale it is very difficult to ensure an identical pose when repeating an experiment; the 1:100 scale means that a small absolute difference in pose can result in a large relative difference. These small absolute differences in the pose are sufficient to give significantly different results. More repeatable results may be achieved by calculating the reconstruction error on a full scale hockey pitch; however this has other practical considerations, i.e. what is the error in the placement of the world known points?

Chapter 9 considered the reconstruction accuracy given a set of fifteen different camera poses. The camera poses were chosen to be representative of those a performance analyst may expect at an international hockey tournament. Therefore the results can be used to determine the expected reconstruction error for a given camera pose, or to choose the most accurate camera pose from those available.

For all the camera poses the median reconstruction error was below the 0.005 m deemed acceptable, however to minimise the reconstruction error the performance analyst should try to locate the camera as close to in line with the half way line as possible and as far back from the pitch as possible.

Finally Chapter 10 combined the work of Chapters 3-9. A novel algorithm to automatically extract player coordinates was described. This included a novel method to reform over-segmented blobs using the class likelihood of ‘super-blobs’. The chapter then assessed the accuracy of the algorithm by comparing the automatically extracted image coordinates to manually extracted image coordinates.

The accuracy of the algorithm was assessed by projecting a set of ground truth image coordinates and a set of extracted image coordinates onto the calibrated plane. It found that players could be extracted within 1 m of their ground truth coordinates with a precision of 75% and a recall of 66%. This is a respective improvement of 20% and 16% improvement on the state-of-the-art. The error in the recall can be attributed to incorrect extraction and non-extraction. It also found that the likelihood of extraction decreases the further a player is from the camera, reducing to close to zero in the most extreme parts of the pitch. This is due to both a lower accuracy in the ground truth coordinates and lower accuracy in the player extraction.

A limitation of this chapter is the lack of comparison of ground truth world known points and extracted world points. This would require a dataset with accurate ground truth world known points and could be considered for future work.

A natural extension of the work presented in this thesis would be to formulate trajectories from the extracted coordinates. This is an assignment problem over time; however it is non-trivial due to non-extractions and the possible close proximity of the players. A possible solution could use a set of Kalman Filters (Kalman 1960) to maintain a per player movement model. The difference between each Kalman Filter’s predicted coordinates and the extracted coordinates could then be used to assign detections to trajectories. A per player appearance model could be used to make this assignment more robust. Using an appearance model is complicated by teammates wearing identical uniforms; it is therefore the finer details such as hair and boot colour which distinguishes players. As noted earlier, using 4K footage gives larger players in the frames and as such easier to extract the finer details of the player's appearance.

Having access to trajectories allows a performance analyst to ask different tactical questions. For example they may ask 'How does player A react to their team losing the ball?'. This knowledge may allow a coach to develop a more in depth tactic to exploit the player's tendency. Trajectories can also be exploited to interpolate missing detections in each frame. This should improve the precision of the algorithm presented in Chapter 10.

In summary, the results of this thesis suggest that with the available single camera technology, field hockey player coordinates can be extracted with state-of-the-art accuracy in regions close to the camera; however the accuracy rapidly decreases in more extreme regions of the pitch. Subsequently the player coordinates cannot be extracted with sufficient accuracy across the whole extent of a field hockey pitch. While it is possible to calibrate a plane the size of a field hockey pitch, poor segmentation means that coordinates can only be extracted for approximately 70% of the players. It is particularly evident that the further a player is from the camera the less likely that their coordinate will be extracted. Better results may be achieved by increasing the resolution of the camera or improving the lens system, however using a collocated dual camera system may be more appropriate.

12 References

Axis Communications, 2017. AXIS P1428-E Network Camera. Available at: https://www.axis.com/gb/en/products/axis-p1428-e [Accessed February 22, 2017].

Azuma, R., 1997. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), pp.355–385.

Basler, 2017a. acA3800-14uc. Available at: http://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca3800-14uc [Accessed February 22, 2017].

Basler, 2017b. Basler Lens C125-0418-5M F1.8 f4mm. Available at: http://www.baslerweb.com/en/products/accessories/lenses/basler-lens-c125-0418-5m-f18-f4mm [Accessed February 22, 2017].

Beetz, M. et al., 2007. Visually tracking football games based on TV broadcasts. In *IJCAI International Joint Conference on Artificial Intelligence*. pp. 2066–2071.

Bergstra, J. & Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13, pp.281–305.

Bernardin, K. & Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Eurasip Journal on Image and Video Processing*, 2008.

Bouguet, J.-Y., 2015. Camera Calibration Toolbox for Matlab. Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/.

Bradski, G. & Kaehler, A., 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*, OReilly Media Inc.

Brewin, M.A. & Kerwin, D.G., 2003. Accuracy of scaling and DLT reconstruction techniques for planar motion analyses. *Journal of Applied Biomechanics*, 19(1), pp.79–88.

Brown, D.C., 1971. Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8), pp.855--866.

Carr, P., Sheikh, Y. & Matthews, I., 2012. Monocular object detection using 3d geometric primitives. In A. Fitzgibbon et al., eds. *Computer Vision–ECCV 2012. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 864–878.

Catapult, 2017. Catapult. Available at: http://www.catapultsports.com/uk/ [Accessed February 3, 2017].

Clarke, T. & Fryer, J., 1998. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91), pp.51–66.

Cortes, C. & Vapnik, V., 1995. Support-Vector Networks. *Machine Learning*, 20(3), pp.273–297.

Cucchiara, R. et al., 2003. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), pp.1337–1342.

Cybenko, G., 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2, pp.303–314.

Dalal, N. & Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. pp. 886–893.

Dictionary.com, 2015. Field Hockey. Available at: http://dictionary.reference.com/browse/Field Hockey?s=t [Accessed April 20, 2015].

Dietterich, T.G., 2000. Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, 1857, pp.1–15.

Dollár, P. et al., 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), pp.743–761.

Du, X. et al., 2017. Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. pp. 1–11.

Dunn, M. et al., 2012. Reconstructing 2D planar coordinates using linear and nonlinear techniques. In *30 International Conference of Biomechanics in Sports*. pp. 380–383.

Ekin, A. & Tekalp, A.M., 2003. Robust dominant color region detection and color-based applications for sports video. In *Proceedings 2003 International Conference on Image Processing*.

Engineering and Physical Sciences Research Council, 2017. EPSRC - Engineering and Physical Sciences Research Council. Available at: https://www.epsrc.ac.uk/

[Accessed February 1, 2017].

English Institute of Sport, 2017a. English Institute of Sport. Available at: http://www.eis2win.co.uk/ [Accessed February 1, 2017].

English Institute of Sport, 2017b. Performance Analysis. Available at: http://www.eis2win.co.uk/expertise/performance-analysis/ [Accessed February 1, 2017].

Erdmann, W.S., 1992. Gathering of Kinematic Data of Sport Event by Televising the Whole Pitch and Track. In *10 International Symposium on Biomechanics in Sports*. pp. 159–162.

Erik, M., Pedersen, H. & Pedersen, M.E.H., 2010. *Good parameters for particle swarm optimization*,

Farrell, J., Xiao, F. & Kavusi, S., 2006. Resolution and light sensitivity tradeoff with pixel size. In *Proc. SPIE 6069, Digital Photography II, 60690N*.

Felzenszwalb, P., McAllester, D. & Ramanan, D., 2008. A discriminatively trained, multiscale, deformable part model. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

FIFA-Fédération Internationale de Football Association, 2011. *Football Stadiums-Technical recommendations and requirements*, Available at: http://www.fifa.com/mm/document/tournament/competition/01/37/17/76/stadiumbook2010_buch.pdf.

Figueroa, P. et al., 2004. Tracking soccer players using the graph representation. In

Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. p. 787--790 Vol.4.

FIH, 2014. Hockey coaches figure it out. Available at: http://www.fih.ch/news/hockey-coaches-figure-it-out/ [Accessed February 3, 2017].

Fleuret, F. et al., 2008. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), pp.267–282.

Freund, Y. & Schapire, R.E., 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), pp.119–139.

GeoGebra, 2017. GeoGebra. Available at: https://www.geogebra.org/apps/ [Accessed February 23, 2017].

GoPro, 2017. GoPro HERO3+ Black Edition. Available at: https://gopro.com/support/hero3plus-black-support [Accessed February 22, 2017].

Great Britain Hockey, 2015. Business and Performance Framework Agreement. Available at: http://www.greatbritainhockey.co.uk/page.asp?section=1436&sectionTitle=Business+%26+Performance+Framework+Agreement [Accessed May 11, 2015].

Harris, C. & Stephens, M., 1988. A Combined Corner and Edge Detector. In *Procedings of the Alvey Vision Conference 1988*. pp. 147–151.

Hartley, R. & Zisserman, A., 2003. *Multiple View Geometry in Computer Vision.2nd*, Cambridge University Press.

He, K. et al., 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 171–180. Available at: http://arxiv.org/pdf/1512.03385v1.pdf.

He, K. et al., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV '15 Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pp. 1026–1034.

Heikkila, J. & Silven, O., 1997. A four-step camera calibration procedure with implicit image correction. In *Proceedings., 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. p. 1106.

Higham, D. et al., 2016. Finding the Optimal Background Subtraction Algorithm for EuroHockey 2015 Video. In *Procedia Engineering*. pp. 637–642.

Hinrichs, R.N. et al., 2005. Predicting out-of-plane point locations using the 2D-DLT. In *29th Annual meeting of the American Society of Biomechanics*. pp. 249–251.

Howard, A.G. et al., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arxiv*. Available at: http://arxiv.org/abs/1704.04861.

von Hoyningen-Huene, N., 2011. *Real-time Tracking of Player Identities in Team Sports*. PhD Thesis. Technische Universitat Munchen. Munich.

von Hoyningen-Huene, N. & Beetz, M., 2009. Rao-Blackwellized Resampling Particle Filter for Real-time Player Tracking in Sports. In *Computer Vision Theory and*

Applications. International Conference on, VISAPP 2009, Lisboa, Portugal. pp. 464–471.

Hudson, C., 2015. *Automated Tracking of Swimmers in the Clean Swimming Phase of a Race*. PhD Thesis. Sheffield Hallam University. Sheffield.

Hughes, C. et al., 2008. Review of geometric distortion compensation in fish-eye cameras. In *IET Irish Signals and Systems Conference (ISSC 2008)*. pp. 162–167.

Hughes, G.F., 1968. On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Transactions on Information Theory*, 14(1), pp.55–63.

International Hockey Federation, 2015. Hockey. Available at: http://www.fih.ch/ [Accessed April 20, 2015].

Intille, S.S. & Bobick, A.F., 1995. Closed-World Tracking Closed-worlds. In *Proceedings of the Fifth International Conference on Computer Vision*. pp. 672–678.

Jaakkola, T.S. & Haussler, D., 1999. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pp.487–493.

Kalman, R.E., 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1), p.35.

Kannala, J. & Brandt, S.S., 2006. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8), pp.1335–1340.

Karpathy, A., 2016. Neural Networks. *CS231n Convolutional Neural Networks for Visual*

Recognition. Available at: http://cs231n.github.io/neural-networks-1/ [Accessed April 3, 2017].

Kennedy, J. & Eberhart, R., 1995. Particle swarm optimization. In *IEEE International Conference on Particle swarm optimization*. pp. 1942–1948.

Kingma, D.P. & Ba, J.L., 2015. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations 2015*. pp. 1–15.

Kingslake, R., 1989. *A History of the Photographic Lens*, Academic Press.

Krevelen, D.W.F. van & Poelman, R., 2010. A Survey of Augmented Reality Technologies, Applications and Limittions. *The International Journal of Virtual Reality*, 9(2), pp.1–20.

Krizhevsky, A., Sutskever, I. & Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS’12 Proceedings of the 25th International Conference on Neural Information Processing Systems*. pp. 1097–1105.

Kuhn, H.W., 2010. The Hungarian method for the assignment problem. In M. Jünger et al., eds. *50 Years of Integer Programming 1958-2008*. Springer, Berlin, Heidelberg, pp. 29–47.

LeCun, Y. et al., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp.2278–2323.

Leser, R., Baca, A. & Ogris, G., 2011. Local positioning systems in (game) sports. *Sensors*, 11(10), pp.9778–9797.

Lin, M., Chen, Q. & Yan, S., 2013. Network In Network. *arXiv preprint*, p.10. Available at: http://arxiv.org/abs/1312.4400.

Liu, J. et al., 2009. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2), pp.103–113.

Liu, Y. et al., 2006. Extracting 3D information from broadcast soccer video. *Image and Vision Computing*, 24(10), pp.1146–1162.

Lu, W.L. et al., 2013. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7), pp.1704–1716.

Mallon, J. & Whelan, P.F., 2004. Precise radial un-distortion of images. In *Proceedings - International Conference on Pattern Recognition*. pp. 18–21.

Mateos, G.G. & Tsai, R., 2000. A Camera Calibration Technique Using Targets of Circular Features. In *Ibero-America Symposium on Pattern Recognition (SIARP)*.

Mazhurin, A. & Kharma, N., 2012. An Image Segmentation Assessment Tool. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP),*. pp. 436–443.

McInerney, C., 2017. *Determining spatio-temporal metrics that distinguish play outcomes in field hockey*. PhD Thesis. Sheffield Hallam University. Sheffield.

Metropolis, N. & Ulam, S., 1949. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247), p.335.

Perronnin, F., Sánchez, J. & Mensink, T., 2010. Improving the Fisher kernel for large-scale image classification. In K. Daniilidis, P. Maragos, & N. Paragios, eds. *Computer Vision - ECCV 2010. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 143–156.

Raynox, 2017. HDP-2800ES Diagonal Fisheye Conversion Lens 0.28x. Available at: http://www.raynox.co.jp/english/video/hdp2800es/index.html [Accessed February 22, 2017].

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), pp.386–408.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323(6088), pp.533–536.

Russakovsky, O. et al., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), pp.211–252.

Salvi, J., Armangué, X. & Batlle, J., 2002. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7), pp.1617–1635.

Seo, Y. et al., 1997. Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick. In *Image Analysis and Processing*. pp. 196–203.

Shah, S. & Aggarwal, J.K., 1996. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29(11), pp.1775–1788.

Simonyan, K. & Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*. pp. 1–14.

Sobel, I. & Feldman, G., 1973. A 3x3 isotropic gradient operator for image processing. *Pattern Classification and Scene Analysis*, pp.271–272.

Sobral, A. & Bouwmans, T., 2014. BGS Library: A Library Framework for Algorithm's Evaluation in Foreground/Background Segmentation. In *Background Modeling and Foreground Detection for Video Surveillance*. CRC Press, Taylor and Francis Group.

Sobral, A. & Vacavant, A., 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122, pp.4–21.

Sony, 2017. AX33 4K Handycam® with Exmore R® CMOS sensor. Available at: http://www.sony.com/electronics/handycam-camcorders/fdr-ax33 [Accessed February 22, 2017].

Sportstec, 2015. Sportscode. Available at: http://sportstec.com/products/sportscode-version-10 [Accessed May 12, 2015].

Srivastava, N. et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, pp.1929–1958.

STATS, 2017. STATS. Available at: https://www.stats.com/ [Accessed February 20, 2017].

Stauffer, C. & Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. In *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*. pp. 246–252.

Sturm, P.F. & Maybank, S.J., 1999. On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*. pp. 1432–1437.

Swain, M.J. & Ballard, D.H., 1991. Color indexing. *International Journal of Computer Vision*, 7(1), pp.11–32.

Szegedy, C. et al., 2015. Going deeper with convolutions. In *2015 IEEE Conference on COmputer Vision and Pattern Recognition (CVPR)*.

Tang, Y., 2013. Deep Learning using Linear Support Vector Machines. *Workshop on Challenges in Representation Learning, ICML*.

Tong, X. et al., 2011. Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video. *ACM Transactions on Intelligent Systems and Technology*, 2(2), pp.1–32.

Tsagris, M., Beneki, C. & Hassani, H., 2014. On the Folded Normal Distribution. *Mathematics*, 2, pp.12–28.

Tukey, J.W., 1977. Exploratory Data Analysis. *Analysis*, 2(1999), p.688.

Uijlings, J.R.R. et al., 2013. Selective search for object recognition. *International Journal of Computer Vision*, 104(2), pp.154–171.

UK Sport, 2017. World Class Performance Programme. Available at: http://www.uksport.gov.uk/our-work/world-class-programme [Accessed February 1, 2017].

Vedaldi, A. & Lenc, K., 2015. MatConvNet. In *Proceedings of the 23rd ACM international conference on Multimedia - MM '15*. pp. 689–692.

Viola, P. & Jones, M.J., 2001. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1, pp.511–518.

Viola, P. & Jones, M.J., 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2), pp.137–154.

Viola, P., Jones, M.J. & Snow, D., 2005. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2), pp.153–161.

Weng, J., Cohen, P. & Herniou, M., 1992. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10), pp.965–980.

White, B. & Shah, M., 2007. Automatically Tuning Background Subtraction Parameters Using Particle Swarm Optimization. In *Multimedia and Expo, 2007 IEEE International Conference on*. pp. 1826–1829.

Woolson, R.F., 2008. Wilcoxon signed-rank test. *Wiley Encyclopedia of Clinical Trials*, pp.1–3.

Yosinski, J. et al., 2014. How transferable are features in deep neural networks? In *NIPS'14 Proceedings of the 27th International Conference on Neural Information*

Processing Systems. pp. 3320–3328.

Zeiler, M.D. & Fergus, R., 2014. Visualizing and understanding convolutional networks. In D. Fleet et al., eds. *Computer Vision - ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*. Springer, Cham, pp. 818–833.

Zhang, M., 2016. Laowa 12mm f/2.8 to be the Widest Rectilinear f/2.8 Lens. *PetaPixel*. Available at: https://petapixel.com/2016/07/26/laowa-12mm-f2-8-widest-rectilinear-f2-8-lens/ [Accessed February 20, 2017].

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

Zivkovic, Z., 2004. Improved adaptive Gaussian mixture model for background subtraction. *ICPR "04 Proceedings of the Pattern Recognition, 17th International Conference on (ICPR"04)*, 2, pp.28–31.