FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multicamera System for Automatic Positioning of Objects in Game Sports

Miguel Soares Moreira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Luís Corte-Real, PhD

Co-supervisor: Teresa Terroso, PhD

July 27, 2016

Resumo

O desporto apresenta uma elevada importância e um alto impacto socioeconómico. A rápida evolução de plataformas multimédia potenciaram oportunidades para que uma enorme quantidade de recursos tecnológicos fosse aplicada no desporto. Rapidamente, tornou-se num campo de aplicação desejável para que equipas com elevados recursos financeiros recorressem a técnicas de monitorização e sensorização para melhorar o desempenho desportivo. No entanto, para a maioria das organizações desportivas só é possível utilizar tecnologias similares através de soluções mais flexíveis e não-invasivas. A aplicação de técnicas de Visão Computacional para estes casos ainda é limitada e requer uma investigação aprofundada.

A presente dissertação foca-se na localização da bola num espaço tridimensional, com particular aplicação em desportos coletivos *indoor*. Várias técnicas de calibração da câmaras e métodos de reconstrução 3D foram estudados e testados com o objetivo de propor uma abordagem que utilize um número reduzido de câmaras convencionais. O trabalho inclui a captura e preparação de um conjunto de sequências representativas para desenvolvimento e validação da proposta, assim como a respetiva informação de referência.

Os resultados obtidos são promissores e demonstram que uma estimação da posição 3D de uma bola pode ser alcançada com erros aceitáveis, tendo em conta os objetivos propostos. Não obstante, a influência do procedimento de calibração de câmaras é elevada e comprometeu todo o processo de extração de informação 3D. Por isso, é necessário mais trabalho de investigação relativo à sua precisão.

Este documento descreve não só as soluções que foram testadas e todas as estimativas quantitativas do objeto 3D, mas também as avaliações e análises de métodos de calibração de câmaras e reconstrução 3D usados para este propósito. São, também, referidas diferentes possibilidades passíveis de trabalho futuro relativamente a melhorias e desafios exigentes que não foram superados com o presente trabalho.

Abstract

Sports have a great importance and a high socioeconomic impact. The rapid evolution of multimedia platforms created an opportunity for a compelling quantity of technological resources and applications to be conveyed into sports. Rapidly, it turned into a desired application field where high-profile teams resort to monitoring and sensing techniques to improve their capabilities. However, for most sporting organizations an opportunity to use similar technologies can only be achieved with more flexible and non-invasive solutions. The application of Computer Vision techniques for such cases is still limited and requires further investigation.

The present dissertation focus on the location of a ball in a 3D space field, with particular application on indoor team sports. Several camera calibration techniques and 3D reconstruction methods were studied and tested with the objective of proposing an approach that uses a reduced number of conventional cameras. The work includes the capture and preparation of a representative sequences set to develop and validate the proposal as well as the respective reference information.

The results obtained are promising and demonstrated that an estimation of the 3D ball position can be achieved with acceptable errors, considering the proposed objectives. Nevertheless, the influence caused by the camera calibration procedure is great and constrained the process of extracting 3D information as a whole. Thus, it is necessary more investigation work concerning its accuracy.

This document describes not only the solutions that were tested and all of the quantitative estimations regarding the 3D object, but also an evaluation and analysis of both camera calibration and 3D reconstruction methods used for this purpose. Possible future work regarding improvements and demanding challenges that were not overcome are also discussed.

Acknowledgements

First, I would like to offer my sincerest gratitude to my supervisors, Dr Teresa Terroso and Dr Luís Corte-Real, and also to Dr Pedro Carvalho, who have supported me throughout my thesis with their patience, knowledge and "getting their hands dirty" when most don't. The level of my thesis dissertation is attributed to them for their encouragement and support and without them, it would have not been completed or written. I could not have thought of more friendly and helpful supervisors than them.

I would also like to thank the INESC TEC for providing the tools and support to the achievements on this work. Also, I would like to thank to the Pelouro do Desporto da Câmara Municipal da Maia for the availability in providing a location for the acquisition of the dataset, essential for this dissertation. I would also like to acknowledge Américo and Gil for always helping me out finding solutions for my problems and would like to thank them for all the lunches we had during this almost 6 months. I will never forget them.

I would also like to thank all the friends that I love and have been supporting me since high school years, and friends that I have met on this course and have a special place in my heart. For never letting me down and for being a constant source of joy in my life, I could never pay you enough.

Finally, I must express my very profound gratitude to my parents and to my brother for providing me with unceasing encouragement and support throughout my years of study and the process of researching and writing this dissertation. I love you all.

Miguel Moreira

"The best way to make your dreams come true is to wake up."

Paul Valéry

Contents

List of Figures

List of Tables

Glossary

0D	Zero-Dimensions
1D	One-Dimensions
2D	Two-Dimensions
3D	Three-Dimensions
CV	Computer Vision
DLT	Direct Linear Transformation
DVS	Desktop Vision System
EPNP	Efficient Perspective-n-Point
FIFA	Fédération Internationale de Football Association
FoV	Field-of-View
fps	Frames per second
GLT	Goal-Line Technology
HD	High-Definition
INRIA	Institut National de Recherche en Informatique et en Automatique
IP	Internet Protocol
KLT	Kanade-Lucas-Tomasi
LM	Levenberg-Marquardt
LS	Least-Squares
MSER	Maximally Stable Extremal Regions
PC	Personal Computer
PnP	Perspective-n-Point
PTZ	Pan-Tilt-Zoom
RAC	Radial Alignment Constraint
RANSAC	RANdom SAmple Consensus
RHS	Right Hand Coordinate System
RTSP	Real-Time Videos Streaming Protocol
SfM	Structure-from-Motion
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
TV	Television

Chapter 1

Introduction

1.1 Context

Sports has an important role in contemporary society and, with globalization, its popularity has expanded worldwide independently of the social class. Moreover, the ascent and rapid evolution of mass media and multimedia platforms have since conveyed a compelling quantity of technological resources into sports and it has become a multimillion-dollar industry [1].

Rapidly, sports games turned into a desired application field to implement Computer Vision (CV) techniques where sports organizations and television (TV) stations can resort to this solution to improve their capabilities.

> "Computer Vision's great trick is extracting descriptions of the world from pictures or sequences of pictures." [2]

Accordingly, distinctive and specialized tasks that rely on image understanding are solved by computers capable of using those different techniques and knowledge for vision perception and, more importantly, for geometric modeling. Therefore, CV can be considered a relevant area of research and development to gather three-dimensional (3D) data of an object inserted in a certain environment through a sequence of images obtained from a set of cameras. Considering a sports event and its elements, it can become an essential support as an appealing tool for TV viewers and, even, for tactical evaluation by a sports team staff.

During most sports games, elements such as the players and the ball conjugate on a determined space field. The former was considered a challenging area for the application of CV technology because it involved complex human motion [3]. Although players can be successfully detected and tracked [4], identical methods cannot be widespread to ball detection due to the ball's size, velocity and trajectory happening above the ground at most of the times. Therefore, estimating the 3D position of a ball becomes of uttermost importance in a context of a sports game.

1.2 Motivation

The nature of most sports don't invariably allow to successfully monitor objects through sensors or other devices fixed to them [5]. Necessarily, a non-invasive system can be a more practical solution to obtain 3D information about any active element in a sports game. This can be achieved by a vision-based approach.

Nowadays, commercially available technologies for visually tracking the trajectory of the ball in sports games are only achievable and feasible for high-profile professional sports due to the difficulty of installation and high-cost equipment [6]. For most sports organizations, an opportunity to enjoy and use similar technologies can only be achieved with more flexible and non-invasive solutions. Through advances in camera technology a legitimate quality of visual capture with an affordable availability can be guaranteed and will bestow a favorable circumstance for these organizations to implant a flexible camera system at their local sports venue and provide an easy way for coaches and mentors to transmit an effective feedback to their athletes [7]. Not only that, the application of CV techniques for low-profile sports organizations is still limited and requires further developments and investigation.

Generally, detecting the 3D position of the ball in sports is essential not only to determine final results but also to help understand some specific actions that may occur during these sporting events. Hence, its position respecting the playing field is becoming relevant and critical for broadcast media and sports organizations considering that their intentions are, respectively, to produce a more alluring TV viewer experience. Not only that, coaches and athletes are becoming more aware of how important technology can be in tactical and technical assessment of their performance, so much that their interest has been increasing in statistical analysis in sports sciences [5].

Cameras can capture an object's 3D information using both active or passive techniques. Respectively, an active technique uses its own source of light to measure reflected energy by the object's surface while a passive technique uses light emitted from the sun or from the actual scene to measure the radiance reflected by the object's surface. Considering sports games, mishaps and failures to extract this information are susceptible of occurring due to the ball's high-velocity, occlusion during games, abrupt variation of movements and reduced dimensions compared to the playing area, consequently creating a limited spatial and temporal resolution. Accordingly, in the presence of a small object with sudden changes of movement, relying on projected light to interpret 3D information will not help outlast those challenges so a passive technique is considered more capable for this approach.

The sequential process of recovering 3D information from a set of cameras that produce two-dimensional (2D) images require the use of an accurate camera calibration technique [8]. Since most of these techniques are directed to small-scale scenarios, there are challenging aspects about their application for large-scale scenarios, such as a playing fields, where the architecture of the required acquisition area may prove to be a big benefit to overcome those difficulties. Therefore, challenges relative to the computation of the ball's 3D position will be its small size, high velocity at which it travels, occlusion and different heights it assumes during sports events. Concerning

the use of cameras, reaching a flexible and easy-to-install solution that utilizes correct calibration techniques for a large space field will appear as the predominant scientific challenges.

1.3 Objectives

The main objective was to propose a flexible approach capable of extracting the 3D data position of a ball using standard (surveillance) cameras. Concerning that, different camera calibration and 3D reconstruction techniques were analysed and tested using some specific movement sequences from a ball and combining it with an evaluation of the number of cameras involved and their location relatively to the playing area. In addition, another milestone important to achieve the final goal was to analyze restrictions and advantages of a sports field's spatial characteristics and to gather all that information as to produce a versatile solution, highly accessible and with lesser complexion.

1.4 Contributions

The main contributions of the following dissertation are described bellow:

- A methodology was proposed for the extraction of the 3D position of an object in a context of a sports game using off-the-shelf cameras, considering measurements from the playing field in question.
- A camera calibration method was used and improved to extract the intrinsic parameters from several cameras disposed on the side of an indoor sports field.
- Application of several perspective techniques to estimate cameras' locations and orientations concerning a sports field environment and using a marker based system to produce ground-truth data.
- Analysis of different 3D reconstruction techniques to provide estimations of the 3D data position of a ball, considering its static position and several defined movements that are specific of some sports games.

1.5 Document Structure

This document is organized in five chapters. Chapter 2 presents the state of the art emphasizing important related work of camera calibration and 3D reconstruction methods, and a brief discussion about some important market solutions for sports analysis. Chapter 3 describes the proposed methodology, detailing the preparation of the dataset and describing the camera calibration and 3D reconstruction methods used. Chapter 4 explains decisions made throughout the experiment testings followed by a discussion of the results obtained. Chapter 5 presents a final discussion of the present work, followed by a suggestion of possible future work and improvements.

Chapter 2

Literature Review

This chapter is composed by a review of related work about methods and techniques to capture and extract 3D information using cameras and is organized as follows. Section 2.1 analyze existent camera calibration algorithms focusing on large-scale scenarios. Section 2.2 revises possible techniques to recover or extract 3D information with views from the scene in question. Section 2.3 exhibits a market survey about existent commercial solutions that solve the problem in question. Lastly, section 2.4 discusses all the conclusions attained from the given review, mainly addressing challenges and possible solutions related to the problem presented.

2.1 Camera Calibration

The retrieval of 3D information from a set of 2D images requires the use of an accurate camera calibration technique and the right choice of this geometric procedure will determine the achievable accuracy, which is a relevant aspect for recovering the 3D points afterwards [8].

To reconstruct a scene from a set of images, the relation between the coordinates of a set of points in 3D space with the coordinates of their corresponding 2D image points must be settled [8]. Camera's characteristics are generally grouped into intrinsic, or internal, and extrinsic, or external, parameters and the estimation of these is called *camera calibration*. The intrinsic parameters describe how the camera forms an image and they can be defined as the set of parameters that characterizes the optical, geometric and digital characteristics of the camera. These are important for linking the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame [8]. Essentially, intrinsic parameters conclude how light is projected over the camera lens onto the image plane. On the other hand, the extrinsic parameters define the camera's position and orientation. They can be defined as any set of geometric parameters that uniquely identify the transformation between the unknown reference frame and the world reference frame [8].

Optical lens systems are normally described by the standard pinhole camera model as a representation of the image projection process [9, 10, 11, 12, 13]. The pinhole camera model is considered an elementary imaging device that can capture flawlessly the geometry of perspective

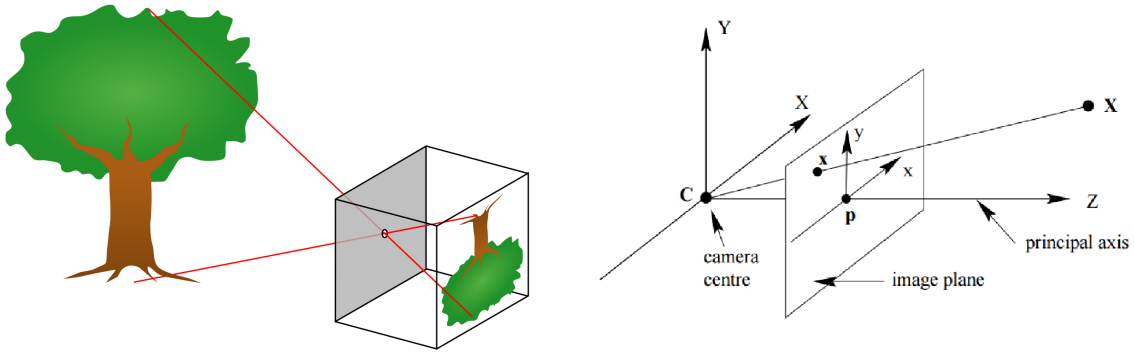


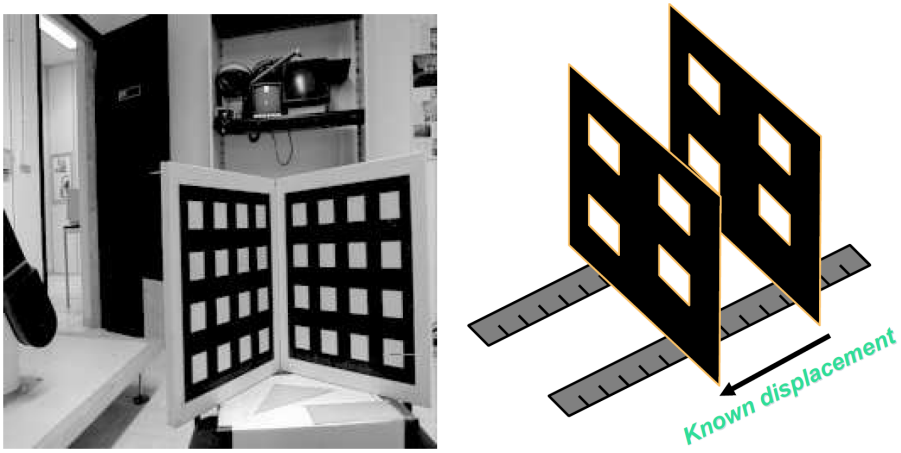
Figure 2.1: Principle of a pinhole camera: light rays from an object pass through a small hole and meet at the projection plane where a reverse image is formed. From [15].

projection. Any ray of light that is emitted to or reflected from that scene has to travel through the pinhole camera before reaching the imaging screen. The correspondence between the area on the imaging screen and the area in the 3D scene is determined by a certain number of rays that go through the pinhole and assemble at the projection plane where a reverse image of the object in question is produced, as seen in Figure 2.1. Therefore, an image is built up or projected from the reference scene to the imaging space [14].

Since many entertainment, scientific and engineering applications rely on the mapping between 3D scenes and their corresponding 2D camera images, finding distinctive applicable camera calibration methods still progresses as an active research issue in CV.

Calibration methods can be subdivided into two different topics: single camera calibration and multiple camera calibration. The first can be overcome with specific calibration methods applied to that exact camera, while the latter needs a crucial process of knowing the location of each camera used besides utilizing the same methods to calibrate one of them individually. The existing calibration methods, as Zhang defines in [16], can also be split into four different categories according to the dimension of the calibration object used: a 3D reference object based, a 2D plane based calibration, a one-dimension (1D) line based calibration and a zero-dimension (0D) approach.

In a 3D reference object based calibration, the camera calibration can be performed by observing an object or pattern whose geometry in 3D space is recognized with high precision [16]. One of the most recognized and most important classic 3D reference object based explicit method was introduced by Tsai [17]. He proposed a two-step calibration method that calibrates a camera with a single image of a 3D or planar calibration pattern, typically black and white chessboards, having recourse to the property of radial alignment constraint (RAC). This means that, assuming radial distortion as the prominent cause of image distortion, the location vectors of a scene point and its distorted image point should be radially aligned about the optic axis of the lens with their cross product equal to zero [18]. If a planar calibration pattern is used, the motion will need to be known and a precise translation of the object will provide the needed 3D reference points. Ex-



(a) (b)

Figure 2.2: 3D apparatus for calibrating cameras. In (a), the calibration apparatus used at Institut National de Recherche en Informatique et en Automatique (INRIA) [19] is shown consisting of two orthogonal planes, each with a checker pattern printed on. A 3D coordinate system is attached to this apparatus, and the coordinates of the checker corners are known very accurately in this coordinate system since three faces will be visible to the camera. In (b) an illustration of the apparatus used in Tsai's technique, which only uses one plane with a checker pattern displaced at least once with known motion, making it equivalent as knowing the 3D coordinates of the checker corners. From [16].

amples of 3D based apparatus for camera calibration can bee seen in Figure 2.2. Although its groundbreaking achievement, this approach requires an expensive calibration apparatus and an elaborate setup [16].

In a 2D plane based approach, the calibration can be done by observing a planar pattern set out at a few (at least two) different orientations. Many approaches have been developed to use planar patterns in camera calibration, due to their ease of manufacture, storage, and use [20]. Another notorious and important 2D plane based method is the one introduced by Zhang [21]. He proposed a flexible calibration technique for desktop vision systems (DVS) that calibrates a camera using a planar calibration pattern with at least nine for better accuracy, as depicted in Figure 2.3.The principle of developing a DVS was to target a general public with no expertise on CV, thus creating an easy to perform single camera calibration. In terms of methodology, the analysis of the relation between each feature point on the plane and the corresponding image point, which represents the homography matrix of each image, constrains the internal and external parameters of the camera. So, a nonlinear optimization of the results is required to get an accurate calculation of the value of all parameters [22]. However, Zhang describes is method as an approach that lies between photogrammetric and self-calibration because it uses an object to calibrate a camera whose geometry in 3D space is known and, at the same time, can use the motion of the camera in a static scene to retrieve its internal parameters by solely using image information.

As a result, Tsai's and Zhang's calibration methods have advanced CV and close range photogrammetry one important step to the real world applications [23]. Comparing these two, Zhang's

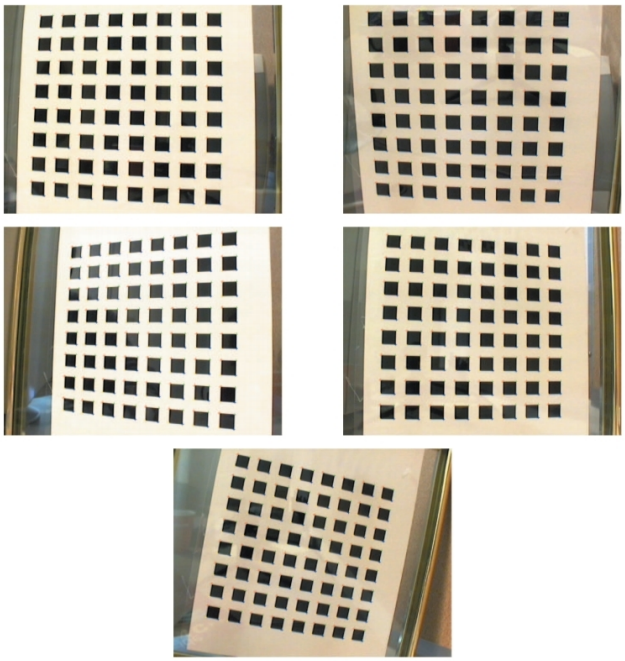


Figure 2.3: Five images of a model plane, together with the extracted corners. From [21].

algorithm does not need any knowledge of the object's motion and is more flexible as opposed to Tsai's one. Sun et al. [24] compared these two calibration algorithms. The results were that Tsai's method achieves high accuracy when trained on data of low measurement error. However, this requires an accurate 3D measurement that typically involves hundreds of samples directed to a fixed reference, which becomes prone to noise as Sun's real-data experiments confirmed. In contrast, Zhang's method does not require a laborious measuring task nor specialized equipments. In fact, the sensitivity of his algorithm to pixel coordinate noise can be overcome by printing another checkerboard pattern containing a higher number of grid corners confirming the flexibility and suitability of Zhang's planar approach for calibration in dynamic environments.

Following up Zhang's method and to provide an easy-to-use refined solution, Bouguet [25] developed a camera calibration toolbox where the nucleus calibration is fully automatic, except for points correspondence on the planar calibration object corners for corner extraction requiring manual input by the user. Zhang and Bouget's work, although widely used in many academic researches or applications, for being plane-based techniques require feature points on a plane calibration object to appear in different views which is not feasible for large-scale scenarios because it would need a big calibration object to get an acceptable depth of calibration.

In a 1D line based concept, the camera calibration can be executed using collinear points as calibration objects. One example is that of Zhang [27], who proposed calibration with a 1D object consisting of three or more collinear points with known distances translated into several markers on a straight line segment rotating around a fixed point. Through co-linearity and measurements of the marker's distance on the 1D line, an estimate camera parameters can be provided. The main

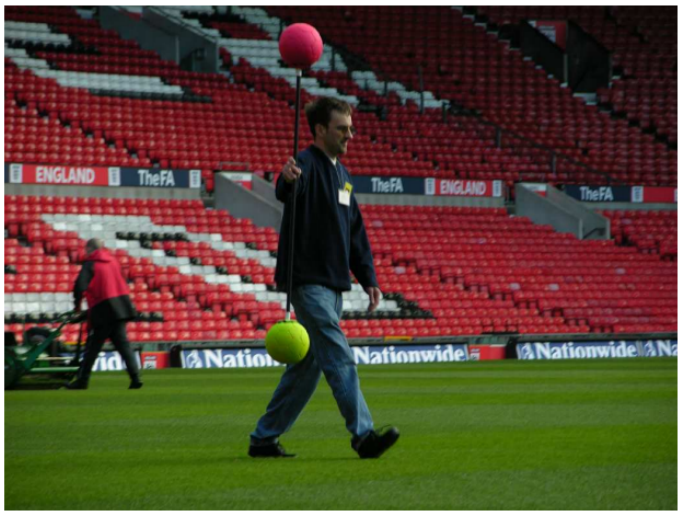


Figure 2.4: A wand-like calibration object with two balls of different colours. From [26].

advantage of this kind of calibration is unreferenced knowledge of the coordinates of space points simplifying the construction of calibration objects. Wu et al. [28] showed that the 1D calibration object can have a new geometric interpretation if rotates around a fixed point, functioning in essence as a 2D calibration object. Wang et al. [29] proved that 1D objects can be used for multicamera calibration with an algorithm of a 1D object undergoing free motion. Synchronously, all cameras must observe the 1D object undergoing at least 6 times general motions. Then, Wang et al. [30] presented an improvement to the accuracy of the 1D calibration method because, at that time, it was much lower than that of 2D or 3D calibration, essentially in the presence of image noise. However, using 1D objects to calibrate cameras has great advantages, such as easiness to construct 1D objects with known dimension, for example by marking three points on a stick; and, in a multicamera setup, all cameras can observe the whole calibration object simultaneously, which is essential and difficult to obtain with 3D or 2D calibration objects. The important drawback is that the 1D object should be supervised to engage in some especial motions, such as rotations around a fixed point and planar motions. Using sticks with markers is probably the standard approach for calibrating commercial multicamera systems for motion capture [31, 20]. Diverting attention to sport scenes, some systems use calibration objects [26, 32], such as wand-like objects depicted in Figure 2.4. However, this approach presents some practical problems in these events, such as the need to capture a specific calibration sequence and the difference of the reference frame between the camera system and the pitch.

As previously mentioned, some calibration methods do not use or depend on a calibration object and can be classified as a 0D approach to the problem of calibrating cameras. This is the case of self-calibration. The task of taking pictures of the calibration object may be difficult in large-scale scenarios and self-calibration avoids it [33]. It calibrates a camera without computing its physical parameters explicitly [34] with those parameters being replaced by a set of non-physical implicit parameters that are used to interpolate between some known tie-points [35]. Although no

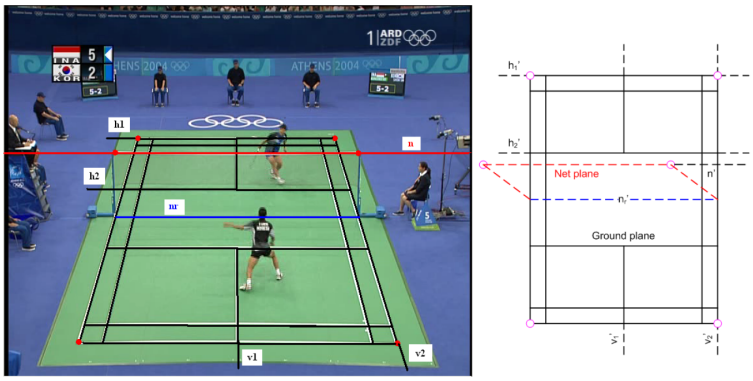


Figure 2.5: The planes, lines and points are selected in the image and the correspondences in the standard model are determined. Six intersection points are used for calibration. From [37, 38].

calibration objects are necessary, a large number of parameters need to be estimated resulting in an exhausting mathematical problem with the risk of singularities greater compared to calibration from planes with known metric structure [36]. One of the most recognized self-calibration approaches is Structure-from-Motion (SfM). SfM is a strategy that uses movement of the camera in a static scene, where the harshness of that scene provides two constraints on the cameras' internal parameters from individual camera dislocation by using image information alone. Accordingly, if images are taken by the same camera with fixed internal parameters, correspondences between multiple images of scene lines, points or other geometric entities across multiple views will be sufficient to recover both the internal and external parameters establishing the first step of this algorithm, called image matching. SfM is an attractive calibration method because internal and external camera parameters are extracted from images of the unmodified scene itself [12].

In terms of the importance of single or multiple views usage, a single camera is the most used in existing CV 3D reconstruction methods. However, this inflicts a restraint on the results obtained, since the 3D scene can only be reconstructed up to a certain extent [39]. In framework applications that demand spatial data accumulation in Euclidean space like sports games, this restriction imposes an enormous relevance. This problem can be solved using calibrated stereo cameras, but with additional steps for camera calibration. Nonetheless, the development of nearly all multiple camera systems until this date have been for small-scale applications with controlled environments. Sports events present a number of additional challenges for 3D acquisition and extraction of objects.

For some time, a number of existing camera calibration methods available for sport scenes used the lines of the playing field to determine camera locations, either using ad-hoc information about the field [41, 42], vanishing points [43] or field/court models. For example, Farin et al. [44] proposed an algorithm that uses a combinatorial search to establish correspondences between lines that were detected with a Hough transform [45] and a court model. First, it starts by identifying white pixels after several tests and classified them as court line candidates. Then, a

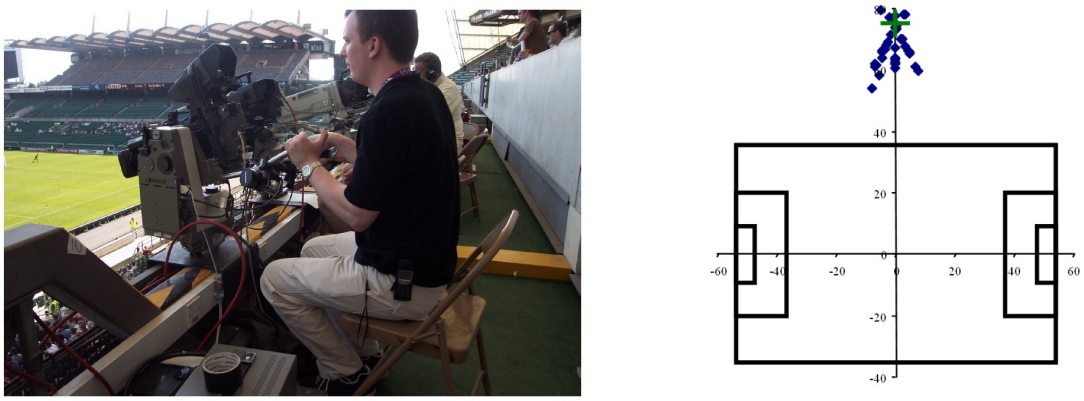


Figure 2.6: On the left, a camera mounted on a pan/tilt head. On the right, camera positions estimated from pitch lines, using individual images. From [40].

Hough transform is applied to form a court model. At the end, a combinatorial search establishes correspondences through a ground-plane homography estimation by assigning line candidates to lines in the court model and making possible to determine the corresponding geometric transformation of the court. However, the Hough line detection algorithm may find lines that are not part of the court and omit some lines that do belong to the court. Besides that, lines can also be too vulnerable to be discovered or not appearing on the camera view. Claiming that the initialization speed of this calibration achieves about one second on a standard computer, Farin et al. [46] presented an enhancement to increase the speed to real-time by maintaining the same white line detector but performing, afterwards, an extraction of the line parameters by applying a Random Sample Consensus (RANSAC)-based [9] line detector. These two algorithms presented are fully automatic, robust against large occlusions and flexible to different sports.

Nonetheless, the ground-plane homography limits to the mapping between the image plane and the standard court plane which is a transform from one 2D plane to another 2D plane, making it impossible to provide any 3D height information of the object. Therefore, Han et al. [37, 38] proposed a 3D modeling consisting on handling both the ground plane and the net plane to generate six intersect points and upgrade to a full camera matrix, and a two-step algorithm to recognize a set of features at positions within these two planes to estimate high quality camera calibration parameters. Figure 2.5 shows an example of the points marked for calibration. At the end, they present an estimation error less than 5% of estimated average players' height on the image sequences tested. However, while the ground-plane homography only establishes 2D to 2D mapping, the full camera matrix allowed to compute the height of objects if their ground position is known.

In sports broadcasts, it is common to use a 3D line model of a standardized playing field to calibrate cameras. Relating image lines and model lines leads to a set of 2D-to-3D line correspondences with existing approaches dealing with each camera separately, depending on the known appearance and geometry of the playing field [47, 48]. Thomas et al. [40] discuss TV broadcast camera calibration approaches for different sports, where a broadcast camera with a zoom lens is mounted on a pan/tilt head, as seen left on Figure 2.6, by presenting two different scenarios:

tracking camera movement using pitch lines and tracking of using areas of rich texture. On the first scenario, it is considered that cameras covering sports events remain in fixed positions during a match. Therefore, it makes sense to use this prior knowledge to compute an accurate camera position, after storing multiple images (see right on Figure 2.6) that indicates a common camera position value. The challenge of the initialisation process, described in [49], is to locate pitch lines in the image and deduce which lines on the pitch they correspond to, only knowing the camera position and pitch dimensions. The Hough transform can be used as a mean to allow establishing a measure of how well the image matches the set of lines that would be expected to be visible from a given pose. On the second scenario, it is considered that there are significant changes in camera focal length, but can make use of the constraint that the camera is generally mounted on a fixed point. The challenge is to be able to estimate the camera pose for these kind of applications from the content of the image where no lines exist at all, such as athletics. The approach on [50] is pointed out, where lines are used to initialize and track the calibration. In the case of pan-tilt-zoom (PTZ) cameras, once the center of projection of the camera is known, pan, tilt and zoom can be determined more by comparing image lines with a database of projected model lines for different values of pan, tilt and zoom. Here, a Kanade-Lucas-Tomasi (KLT)-based tracker [51] is used to identify and track features, using a combination of fixed reference features; the camera pose is computed from the observed feature points, along with a RANSAC process to remove outliers. More recently, Sankoh et al. [52] proposed a method that estimates an homography matrix for the first frame and identifies reliable corresponding feature points between consecutive frames using Speeded Up Robust Features (SURF) [53]. Then, estimates homography for next frame and extracts objects' texture regions in that same frame, completing that course for the remainder of frames of a single moving camera. In order to evaluate and observe the estimation accuracy of the homography matrix in each consecutive frame, the authors compared it with the RANSAC-based line detection method in [38] under the same manually calculated conditions. The results demonstrated that the proposed method can estimate the homography matrix more accurately by calculating a lower average distance between the projected point and the real feature point. The estimation accuracy of the homography matrix has a tremendous effect on the quality of object extraction, thus producing more reliable results.

Resorting to broadcast images, Wen et al. [54] introduce a technique of calibrating basketball videos by generating the panoramic image from a whole court view shot video. Then, a homography transformation is estimated using the tracked KLT features (court lines and advertising images on the court) between consecutive frames and each frame is transformed to an identical coordinate system. After that, court region in the panoramic image is detected based on the dominant color. Considering that panoramic court may be distorted due to accumulated transformation errors, the court is warped to a quadrangle. Finally, by employing the obtained corner correspondence, the quadrangular basketball court is rectified to a standard one using a homography to remove the perspective effect. His robustness permits that one frame with missing court features can obtain those lost features from other frames.

Without employing homography calculations, Zhang et al. [55] introduce a method to obtain

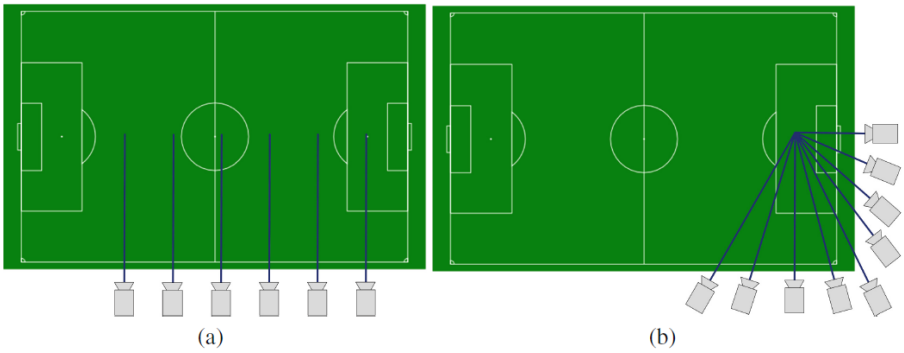


Figure 2.7: Two possible camera arrangements for soccer scenes: a linear arrangement and a curved arrangement with piecewise linear properties over large scales. Both arrangements have different properties. (a) In the linear arrangement, all cameras are placed on a line next to the long side of the pitch and have the same look-at angle. (b) In the curved arrangement, all cameras are placed around a corner of the pitch and point to a spot in the scene. From [58].

Figure 2.8: Multicamera feature matches, considered as inliers. Most outliers are removed using the angle-based filtering. Only a subset of the multicamera matches are shown. From [58].

an adequate calibration information of a single moving camera based on a football field model with marked line pick-up algorithms. Using *a priori* knowledge about an unchanging camera's physical location, it develops a two-stage camera calibration method. First, it uses a Direct Linear Transformation (DLT) [56] method to calculate the coordinates of the camera location. Second, it employs a pan-tilt model to simplify camera projection equations and takes advantage of the Levenberg-Marquardt (LM) [57] nonlinear algorithm to get camera parameters, instead of computing the homography matrix. The author names it Pan-tilt Camera Calibration Algorithm and compares it to the DLT method, by examining their robustness. Shen et al. [10] also use the DLT method along with a non-linear optimization [9] to get the projection matrix in a monocular 3D reconstruction of a ball in a table tennis game. Since the DLT method needs at least six court-to-image point correspondences and the table tennis court dimensions are known, four corners of the table and two upper points of net are used.

Focusing on multiple cameras, Goorts et al. [59] present a method to calibrate large scale camera networks for 3D CV applications in sport scenes using (two) plane sweepings [60] and a depth filtering step using 8 cameras, placed in a linear arrangement as shown in Figure 2.7. In the initialization phase, i.e. when cameras are calibrated, Scale-Invariant Feature Transform (SIFT) [61] local feature points, such as playing field lines and players, are detected in the scene to generate

camera correspondences. Then, these are paired between cameras and those pairwise correspondences are tracked across multiple cameras (see Figure 2.8) using a graph-based search algorithm. Those matching correspondences will be used to calculate the camera's position, orientation and intrinsic parameters through Svoboda et al. [62] calibration method, that requires at least 3 cameras and a moving laser pointer used inside camera's field-of-view (FoV) in order to find image correspondence points. Nevertheless, segments in the virtual image can only have one depth and will result in players disappearing if they are overlapping in the image. Furthermore, the method is only suitable for a smaller baseline setup (about 1 meter). Hence, Goorts et al. [63] extends the plane sweep approach, allowing the generation of higher quality results by employing an initial plane sweep to generate a crude depth map, filter it and use it for a second, depth-selective plane sweep. The experiment consists on 7 cameras aimed at the pitch with a static location and orientation and placed around one-quarter of the field with a wide baseline setup, i.e. around 10 meters between each camera. As shown in Figure 2.7, these methods presented can have two possible camera arrangements: a linear arrangement and a curved arrangement. Goorts et al. [64] also used the same method but with a different variation. Once the multiple camera matches are determined, they perform an angle-based filtering which further enhances the correctness of the final result of the calibration by eliminating possible mismatches. Pairwise feature matches are propagated over all camera views using a confident-based voting method However, it does not permit that the cameras are too much rotated relative to each other because it is assumed that lines connecting matching features are more or less parallel and if the cameras are placed far apart, results would not be correct. To complement this work, in [58] he also presented some alternatives for feature-based point correspondence determination. For the first alternative, they moved an easily-detected orange ball in the scene. The ball can be detected by selecting the largest orange object in the scene, and the point can be determined by choosing the center of the circle enclosing the orange pixel blob. Results showed that the method is reliable for calibration, resulting in only a few outliers. However, they claim some issues regarding the practical availability in sports scenes. First, there must be a calibration recording, which costs time and effort. Second, it is forbidden to enter the pitch before or right after the game. This will limit the possible locations of the ball to the side of the pitch and the spectator area, resulting in less correspondences useful. The second alternative uses a stroboscope to generate light flashes. The stroboscope is attached to the synchronization signal of the cameras, determining that a flash occurs only when cameras take a frame allowing the stroboscope to provide a light flash with high intensity for a short time. If the shutter time of the cameras is reduced, the only thing visible in the image will be the flash.

Puwein et al. [65] also presented a framework to calibrate multicamera sequences but separated by wide baselines. By providing the calibration of a single frame, a method to extract and update a feature-based representation of the scene being captured is bootstrapped, taking advantage of different kinds of features. In the feature matching stage, the large tolerance which Maximally Stable Extremal Regions (MSER) [66] features provide towards changes in viewpoint is leveraged and combined with SIFT [8] in a two-step matching procedure. The special structure of homographies mapping scene planes to images leads to a very efficient 2-point RANSAC when matching MSER

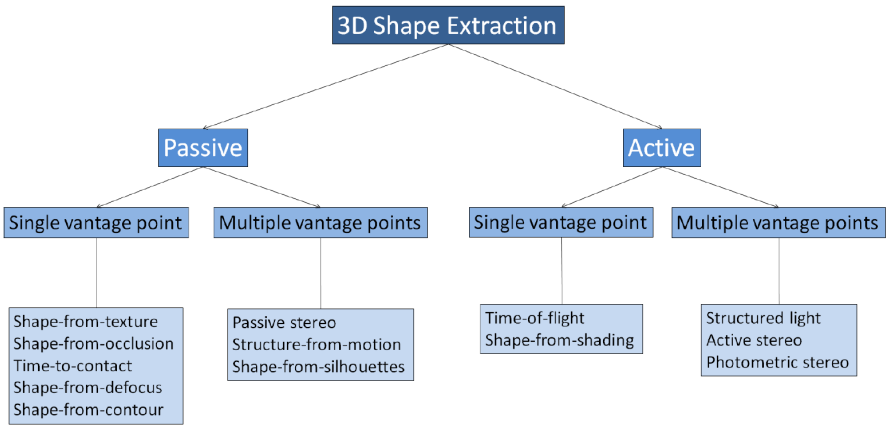


Figure 2.9: Taxonomy of methods for the extraction of information on 3D shape. From [67].

features corresponding to a virtual, larger FoV and leading to more features being extracted and matched. A feature based representation of the scene is created and updated using a very simple yet effective strategy based on the visibility and the uniqueness of features, Figure.

2.2 3D Data Extraction

The extraction of moving targets' 3D information in sport events is essential to acquire some technical and tactical information. To obtain that data, besides of an accurate camera calibration, an adequate choice of 3D scene reconstruction from a set of multiple images must also be scrutinized. The 3D data extraction methods can be represented by the taxonomy [67] presented in Figure 2.9.

The first division is between active and passive methods. In the active form, suitable light sources are used as the internal vector of information and are controlled, as part of the strategy to retrieve 3D information. Active lighting incorporates some form of temporal or spatial modulation of the illumination. From a computational point of view, active methods tend to be less demanding, since active illumination simplifies some of the steps in the 3D acquiring process. On the other hand, in passive methods, the reflectance of the object and the illumination of the scene are used to derive the shape information, thus no active lighting device is necessary since light is not controlled or only with respect to image quality. Normally, passive techniques work with any ambient light available.

The second distinction is between the number of observed viewpoints we have from the scene, defining them as single-view or multi-view systems. On the former, the system works only from a single viewpoint. On the latter, several viewpoints and/or controlled illumination source positions are involved. For this type of systems to work well, every different components, such as cameras, often have to be positioned far enough from each other, resulting in a wide baseline (distance between two or more lenses).

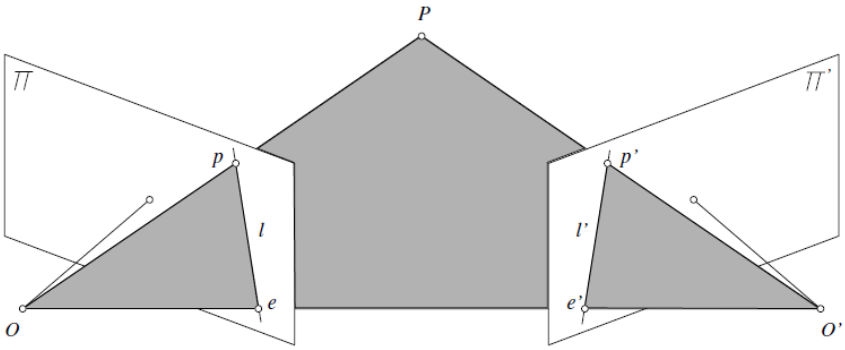


Figure 2.10: Epipolar geometry: The point P, the optical centers O and O' of the two cameras, and the two images p and p' of P all lie in the same plane. Here, as in the other figures of this chapter, cameras are represented by their pinholes and a virtual image plane located in front of the pinhole. This is to simplify the drawings; the geometric and algebraic arguments presented in the rest of this chapter hold just as well for physical image planes located behind the corresponding pinholes. From [68].

As previously said in section 1.2, considering that the main objective is to infer the 3D position of a small-size moving object that suffers from sudden changes of movement and occlusions, an active method would need a controlled light source to point to the object continuously during the 3D extraction, which is not considered feasible. Thus, only passive methods will be accounted for this analysis. The differences between number of viewpoints on passive methods will now be discussed with more detail.

Focusing on multiple views, the most common in CV is to use two different views to compare images and acquire relative depth about a scene. Such framework is mentioned as stereo vision. The principle behind stereo-based 3D reconstruction is that, given the two projections of the same point in the world onto the two images, its 3D position is found as the intersection of the two projection rays [67]. Stereo vision involves two processes: the fusion of features observed by two eyes and the reconstruction of their 3D inverse image. The inverse image of matching points may be found at the intersection of the rays passing through these points and the associated pupil centers [68]. Repeating such process for several points yields the 3D shape and configuration of the objects in the scene.

The matching pairs of correspondence points are constrained to lie on corresponding epipolar lines in the two images, thus this geometric epipolar constraint associated with a pair of cameras will be essential to control the demanding stereo fusion process. Considering that images p and p' of a point P observed by two cameras with optical centers O and O', those five points will all belong to the epipolar plane defined by the two intersecting rays OP and $O'P$, as shown in Figure 2.10. In particular, the point p' lies on the line l' where this plane and the retina π' of the second camera intersect. The line l' is the epipolar line associated with the point p, and it passes through the point e' where the baseline joining the optical centers O and O' intersects π'. Likewise, the point p lies on the epipolar line l associated with the point p', and this line passes

through the intersection e of the baseline with the plane π.

Given a calibrated stereo rig and two matching image points p and p', in an ideal scenario it is straightforward to reconstruct the corresponding scene point by intersecting the two rays $R = Op$ and $R' = O'p'$. However, the rays R and R' practically never actually intersect due to calibration and feature localization errors. This method, referred as triangulation, requires the equations of the rays and, hence, complete knowledge of the cameras: their relative positions and orientations, but also internal settings like the focal length [67].

Usually, passive stereo uses two synchronized cameras. Two images of a static scene could also be taken in sequence by placing the same camera in two different positions. This method is named structure-from-motion or SfM. If images are taken over short time intervals, it will be easier to find correspondences by tracking feature points over time. Once such strategy is considered, more than two images can be taken while moving the camera and, thus, estimating 3D structures from the sequence of 2D images. The goal of stereo passive methods is to recover the 3D geometrical structure of a scene from two images of it. A 3D Euclidean reconstruction can be achieved if the intrinsic and extrinsic camera parameters are known, i.e. are fully calibrated. Although if the baseline distance is unknown, the reconstruction degenerates to a 3D metric reconstruction. If the calibration matrices of the cameras are unknown, the scene structure can only be recovered up to a 3D affine transformation. Lastly, if no information about the camera setup is available, then only a projective 3D reconstruction is possible.

The shape-from-silhouettes approach is another multiple view passive method that uses foreground-background segmentation to reconstruct 3D objects by carving out voxels in space [69, 70], or by reconstructing 3D meshes directly [71].

As stated before, the existing 3D reconstruction methods of sports scenes and objects can be divided into single camera and multiple camera usage. Regarding single camera, Kim et al. [72] estimate the ball's height with reference to human's height which is not reasonable in many sports. In their algorithm, they manually determine two objects perpendicular to the ground and calibrate the camera's projection on the ground. Furthermore, these two objects need to have similar scene depth. By exploiting triangle relationship, the ball height is computed. Reid et al. [73] evaluate the 3D position of a ball by a geometric constraint, which makes use of shadows on the known ground plane. However, it has difficulty to automatically detect the shadow in image.

Liu et al. [74] propose an algorithm to predict the ball's 3D information from monocular view with an improved method that predict the ball's flying plane through virtual shadow computing. Virtual shadow can be defined as objects' projective points on the playing field plane. It categorizes two classes for ball's positions. The first is considered as ball on the ground, and it is trivial to compute their positions on the playing field from image through the transform between image plane and the field plane. The second is considered as the ball flying through the air, where it will have a jumping-off point and a falling point to be estimated through an optimal parabola.

By introducing some dynamic equations, Ohno et al. [4] and Yamada et al. [75] evaluate the ball position fitting a physical model of ball movement in the 3D space. They assume that the ball motion is determined by the gravity and the air friction. These equations depend of an unsolved

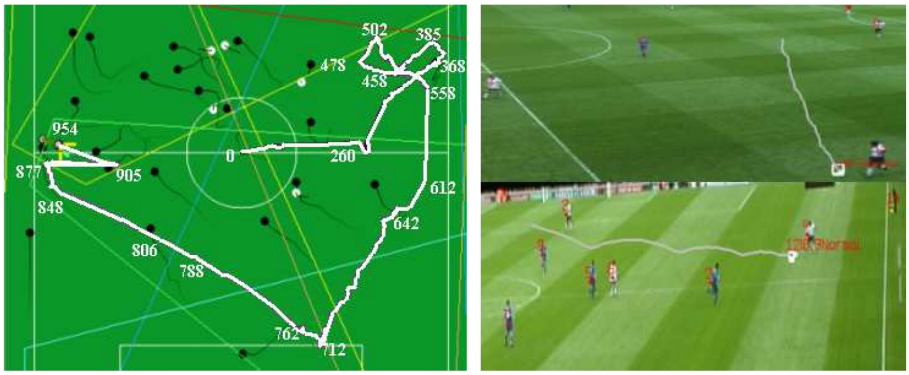


Figure 2.11: Estimated 3D ball trajectory compared with ground-truth and two 2D trajectories. From [77].

initial ball's speed, which are difficult to acquire solely from images. Ribnick et al. [76] developed a theoretical analysis of a linear solution and established the minimum conditions for the existence of a unique solution. They asserted the need of at least three image samples to solve the ball motion problem, since there are six unknown and independent parameters, 3D position and velocity, as each image measurement provides at most two constraints on the system. Shen et al. [10] continue the work in [4, 76] to propose a monocular 3D reconstruction method based on those two methods. However, they address that those cannot be directly used in the ball trajectory estimation in sports due to the model drifting problem and solve it by mixing the physical and the geometrical model of the ball. The model drifting problem can be defined as the drift of the estimated ball trajectory from the trajectory along the direction perpendicular to the camera image plane.

Considering multiple cameras, in [78] 2D ball is detected by template matching and tracked using epipolar line constraints between four cameras. They consider the virtual viewpoint image from the upper direction of the soccer field. The ball position from the virtual viewpoint can be estimated by finding the intersection of epipolar lines corresponding to where the ball positions between the cameras. According to the ball position from the virtual viewpoint, the number of the candidate cameras is limited. When the ball is close to one side in the virtual viewpoint image, candidate cameras are limited to two cameras on the same side as the ball's position. Kim et al. [79] generate a feature matching of the lines tracked on the playing field and, by building small mosaics combined with several frames of a broadcast video stream, acquire the ball's 3D position. When two synchronized videos are available, the ball's position can be computed through the multiple view geometry relationship.

Ren et al. [80, 81, 77] (see Figure 2.11) employ eight stationary cameras placed along the stand to cover the field in a soccer match. In their system, these cameras are calibrated to the ground-plane coordinate system in advance. Thus the players' positions can be determined through the homography between image and the playing field in monocular view. Two methods are described for detecting that the ball has left the ground plane, and is following a 3D trajectory. The first uses triangulation of multiple sources of data to estimate a 3D position; the second uses an analysis of

the trajectory from one or more data source to infer that the ball has left the ground. To estimate the flying ball's position in the air, it needs at least two cameras to use the epipolar geometry constraint. The use of more cameras allow to never miss an object in case of occlusion in some views. In case the ball is only detected from a single view, hints about its height are extracted from the shape of its trajectory. In case of frames where no observations are available, ball's position is estimated by polynomial interpolation to generate a continuous 3D ball trajectory in all frames.

Ishii et al. [82] claim that 3D position can be easily calculated by a wide baseline stereo method, after ball's 2D position from both cameras is estimated. The ball is represented by a 3D model and a deterministic method is used to estimate the ball's 3D position. Also, they resort to a Kalman filter to compensate the ball-missing frames.

Aksay et al. [6] and Kumar et al. [83] use the dataset from the ACM Challenge 2010 [84]. This dataset includes nine stream videos of a tennis match scenario with eight of the nine cameras placed in different positions around a tennis court and one placed above the court. Along with that, some chessboard images and 3D locations of few known objects in the scene were included for camera calibration. Both authors calculate 3D ball's position using a simple triangulation method of the 2D points in each camera. Also using the same dataset but resorting to a different approach, Kelly et al. [85] used a volumetric intersection technique to compute the visual hull of the 3D object (players and ball), which is based on the shape-from-silhouette method [86]. The visual hull is obtained by the intersection of planes from all cameras. If a 3D point is simultaneously captured by all cameras of a section for a specific plane, then that point belongs to the visual hull. The 3D mesh is subsequently calculated using the marching cubes algorithm [87]. Although not being able to verify ground-truth data about some reprojection errors that could occur, the authors claimed that good results were achieved.

2.3 Market Solutions

One of the first commercially available multicamera systems based on 3D information on CV was developed for tracking cricket balls in 3D. Its name is Hawk-Eye [88] and was first used in 2001, subsequently applied to tennis officiating. The system is now deployed with a number of high-speed synchronized cameras viewing the tennis court and mounted on the underside of the roof, Figure 2.12. Because the cameras are static, they are easier to be accurately calibrated in advance resulting in short shutter times and higher frame rates used. The system first identifies possible balls in each camera image and then candidates for balls are linked with tracks across multiple frames. Afterwards, plausible tracks are then matched between multiple cameras to generate a trajectory in 3D. The Hawk-eye technology has gaining much reputation and has since been used in official refereeing for several important competitions, such as the 2015 Rugby World Cup.

The GoalControl system [90] was used for the 2014 World Cup in Brazil [5] and, likewise Hawk-Eye, it is a certified Goal-Line Technology (GLT) technology installation from Fédération Internationale de Football Association (FIFA). This system works with 14 high-speed cameras (Figure 2.13), 7 per goal, around the pitch and positioned at the stadium roof. The cameras are

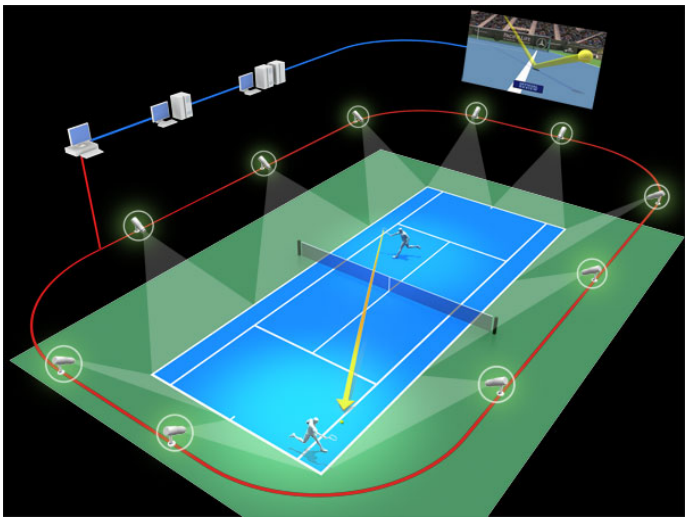


Figure 2.12: The Hawk-Eye technology. From [89].

connected to an image processing computer system which tracks the movement of all objects on the pitch and filters out the players, referees and all disturbing objects. The remaining object will be the ball and the system learns its 3D position with a precision of a few millimeters. When the ball passes the goal line, the system sends a vibration and optical signal to the officials' watch. Besides, all camera images of goal and near-goal events are stored and can be used for TV replay anytime.

ChyronHego [92] has a broadcast graphics solutions for real-time 3D tracking of objects in sports, including players, referees and the ball. The image tracking system uses two super high-definition (HD) cameras and image processing technology to deliver live tracking of all moving objects, achieving a maximum delay of just three frames. The camera units use stereo methods to ensure that the entire playing surface is filmed from several angles. Their software analyses every image to extract 3D positions for each object achieving tracking in real-time.

The Playfulvision system [93] is a automatic video analysis technology. It can provide video-based analytics and statistics for different sports, including volleyball, tennis and equestrian. In its professional version, six cameras are placed around the field or court and are connected to a computer. It can detect the position of the players and the 3D position of the ball at real-time, recognizing different players based on the number of the shirts (volleyball scenario). It was developed at the CV Laboratory in Swiss Federal Institute of Technology in Lausanne.

Developed by the Fraunhofer Institute for Integrated Circuits IIS, the GoalRef system [94] is also licensed by FIFA and differs from the previous ones because it is a radio-based sensing system that uses low-frequency magnetic fields to detect if a goal has been scored. Two low-frequency magnetic fields are used, with one magnetic field created in the goal area and the other one created in and around the ball, whenever it approaches the goal. Then, the software monitors the condition of the magnetic field in the goal and detects if there is a change due to the passage of the coils of the ball over the goal line. At the end, the result is transmitted via wireless to the referee's

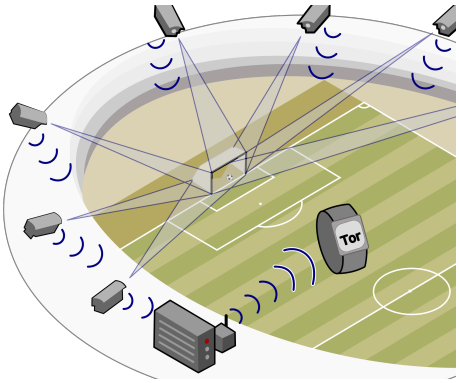


Figure 2.13: Example of a high-speed camera used in GoalControl system. From [91].

wristwatch and a message is displayed in real time.

2.4 Summary

In recent years, 3D acquisition systems are attaining irrefutable interest in 3D reconstruction of sport scenes, not only for technical and tactical assessment but also for TV visual graphics. Two major issues dominate the problem of acquiring and extracting 3D information about objects in a dynamic scene: camera calibration and 3D extraction.

Once there is an enormous interest in 3D reconstruction of sport events, a number of different methods to calibrate single and multiple cameras are proposed by many authors. One can use intersecting points on lines of the playing field, vanishing points on the scene, use feature detection in texture or homogeneous planar scenes and, even, use calibration objects to get the results needed. With this process completed, many authors claimed that, considering an ideal scenario, using geometric constraints from the scene, one can achieve 3D data from a point from that scene and, thus, reconstruct in 3D an object given.

In spite of many achieved academic work regarding these issues, they present many challenges. In multicamera setups, cameras can be separated by wide baselines. These provides a hard issue in order to extract multiple view correspondences. Wide baselines are used to provide fewer cameras but to guarantee that every part of the scene is captured. When calibrating in these setups, features in the scene can change their appearance significantly between viewpoints, making trustworthy feature matching between cameras difficult. To establish correspondences between views, the wide baselines can be tackled in a two-step procedure that benefits both from the invariance of MSER features and the descriptiveness of SIFT features. In other cases of multicamera systems, no information regarding the calibration methods were given since they were provisioned with calibrated datasets in advance, also as considering problems of synchronized images which were considered solved from the beginning.

Regarding mono-vision systems, many authors utilize broadcast images of sport events, a situation where only one camera stays in an initial fixed position but suffers from image translations

throughout coverage. It is common, in these situations, to use a 3D line model of a standard playing field to calibrate cameras. Relating image lines and model lines leads to a set of 2D-to-3D line correspondences with existing approaches dealing with each camera separately, depending on the known appearance and geometry of the playing field. Additionally, single camera usage lack the multi-view constraints, which reduce the relative error between cameras.

Chapter 3

Methodology

The development of a camera solution capable of identifying and reconstructing the 3D position of an object in a context of a sports game was the main objective of the present thesis. A set of multiple cameras was used to capture a ball in a basketball field, comprising several camera locations and diverse ball movement sequences inside a 3D playing field frame.

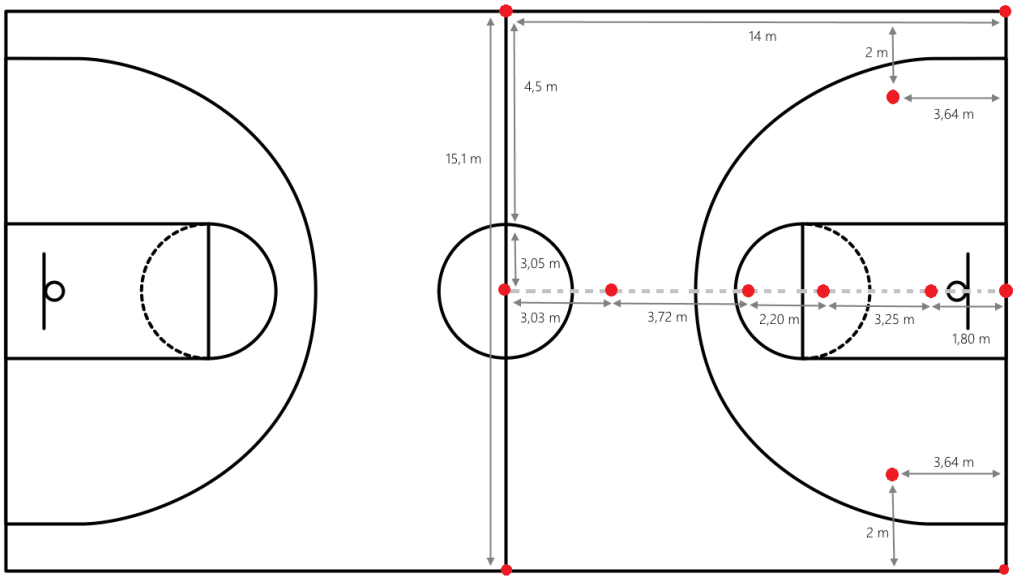
3.1 Dataset Preparing

To prepare the dataset, an image acquisition protocol was developed considering all stages of camera calibration and different static and dynamic ball scenarios. To acquire the dataset, four cameras were defined as the adequate number to provide good and differentiated test results. They were denominated as IP1, IP2, IP3 and IP4 for differentiation issues.

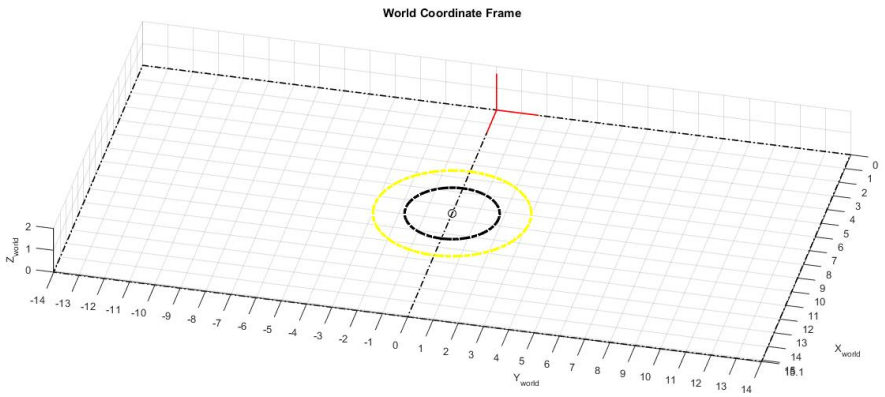
A Personal Computer (PC) was linked to a network switch with all cameras connected via networking cables and with correctly defined Internet Protocol (IP) addresses. The cameras were in the same network as the PC by establishing the first three octets of the IP addresses as the same. For the image acquisition, the Real-Time Videos Streaming Protocol (RTSP) were used to stream the video data and record it on the PC data storage.

The playing field where the image acquisition was performed, had the dimensions of a basketball field, with a total of 15.1×14 meters (Figure 3.1a). To allow a full capture of the chosen world frame, only the right half of the basketball field was considered (Figure 3.1b). The field was illuminated with a natural and consistent light providing good information from all of the captured field. The walls and the floor of the pavilion were white contributing for a good manual detection of the markers' points and the ball, specially when they were far away from the cameras.

All four cameras used on the image acquirement process were placed on the stands on one side of the basketball field to ensure the capture of the right half of it, as seen in Figure 3.2. Cameras IP1 and IP4 were placed facing the bottom right and bottom left corners of the chosen world reference frame respectively, and with a sufficient focus to capture the entire half court. Cameras IP2 and IP3 were placed in the middle of the other two cameras and close to one another, so that a stereo-vision system could be portrayed. On Figure 3.3, the view from all of the four



(a)



(b)

Figure 3.1: (a) Court field measurements (in meters). The choosing field reference was the right half of a basketball field. Red dots indicate chosen ground-truth positions. (b) World reference frame (in meters). The center origin of the world coordinate system is the upper left corner of the right side of the field, illustrated by the red lines. The *xy*-plane orientation constrained a *z*-axis positive value for positive heights.

cameras is detailed. The definition of the camera's image resolution depended on their personal features and were chosen to obtain the maximum potential of the image acquirement. Not only that, the designation of the frame rate and video codec was equal to guarantee conformity between cameras. The parameters are defined on the list below:

- **Resolution** (in pixels): 1920×1080 (IP2 and IP3) and 1280×720 (IP1 and IP4);
- **Video Codec**: H.264;
- **Frame Rate** (fps): 30 frames per second;

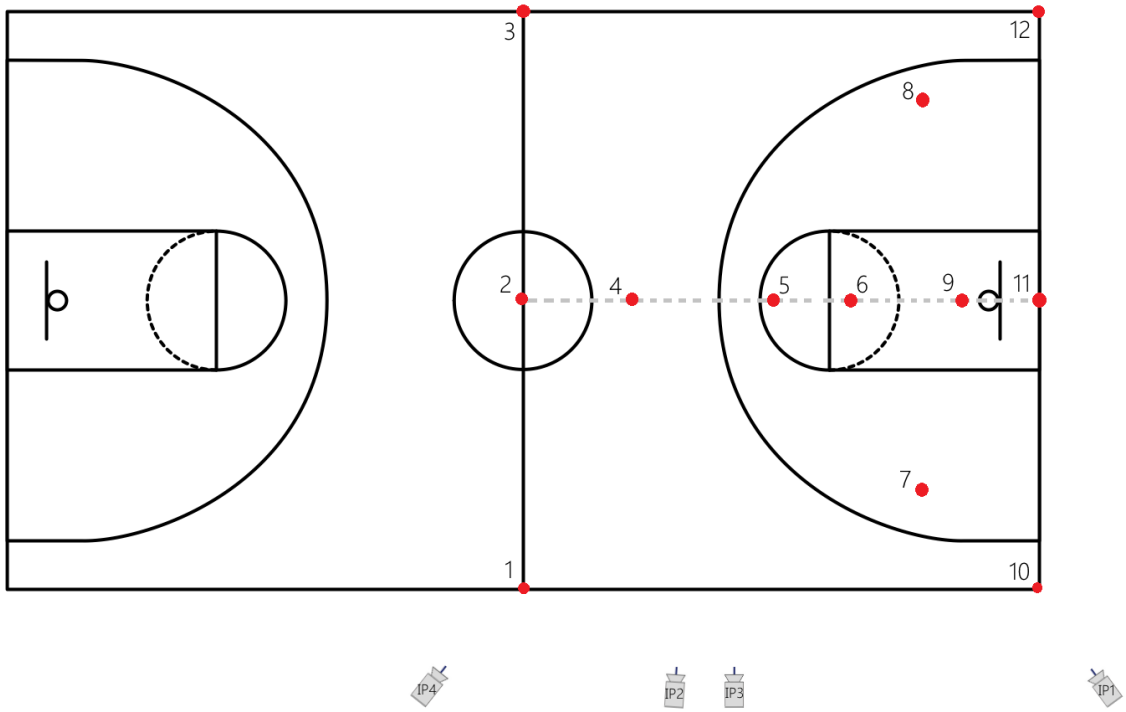


Figure 3.2: Placement of the 4 cameras around the field by the order IP4, IP2, IP3 and IP1 from left to right. Red dots' numbers represent the order followed during the acquisition protocol for the positioning of the ball.

Each camera had the auto-focus disabled in order to maintain the same calibration parameters throughout the image acquiring, since a change on the image focus would change the values of the intrinsic parameters. The 3D object used for the image acquisition was a basketball (ball) with a recognizable size and color in contrast with the playing field and was not occluded to allow for an easy manual detection on all image frames.

Initially before any image acquisition, the measurements of the field lines and ground-truth points were taken. Considering the number of cameras used and to guarantee a good flexibility on their placement, only the right half of the basketball court was considered as the world reference frame. As seen in Figure 3.1a, 12 points on the ground court were chosen to allow a construction of a real world coordinate system totally covering the right half of the field. The upper left corner of the considered 3D field was acknowledge as the center origin of the world coordinate system, as illustrated on Figure 3.1b. The choice of considering that xy-plane orientation constrained a positive z-axis as the height of the field due to the Right Hand Coordinate System (RHS) which guarantees positive values for ball positions above the ground ($z = 0$m).

A detailed description of the different steps that were followed during the image acquisition can be found on Table 3.1.

On step 1, the calibration was performed on each camera individually. On step 2, camera pose and orientation were determined using markers placed on positions 1, 3, 4, 6, 10 and 12 (see

(a)

(b)

(c)

(d)

Figure 3.3: View of the cameras located on the stands of the pavilion. They were pointed to the basketball court in such a way as to guarantee full capture of the world reference system. (a) View of camera IP1. (b) View of camera IP4. (c) View of camera IP2. (d) View of camera IP3.

Table 3.1: Steps performed on image acquirement, for each camera

Steps	Description
1	Capture calibration data by moving a planar black and white chessboard pattern.
2	Capture calibration data using markers placed on different positions of the field.
3	Record a static ball in different positions for different heights.
	3.1 Static ball for $z = 0$m.
	3.2 Static ball for $z = 1$m.
	3.3 Static ball for $z = 2$m.
	3.4 Static ball for $z = 3$m.
4	Record different continuous movements.
	4.1 Record a ball in continuous movement for $z = 0$m.
	4.2 Capture a dribbling ball movement.
	4.3 Record a ball in a free continuous movement.

Figure 3.2) so that it would mark down the frontiers of the field as the world coordinate frame and provide depth as a calibration object.

To ensure strong ground-truth data on steps 3.2, 3.3, and 3.4, i.e. when the ball is not positioned on the ground, a thread with a mass attached in one end and the ball attached to the other end was used. The mass is placed on the ground on each marked position and the ball's height was measured on the thread. Accordingly, a stepladder was used and the ball was held with both hands with the thread perpendicular to the field plane (see Figure 3.4).

On steps 4.1, 4.2 and 4.3, when the ball was moving continuously, camera synchronization was performed using an electronic flash device, as shown in Figure 3.5. This procedure provided for the initial instant $t = 0$s, for all cameras since all four cameras had an equal frame rate, subsequently frames after the initial instant were therefore synchronized.

3.2 Camera Calibration

As mentioned in chapter 2, camera systems are usually described by the standard pinhole camera model as a representation of the image projection process [9], as depicted in Figure 3.6. An image 2D point can be defined as $m = [u, v]^T$ and a 3D world point as $M = [X, Y, Z]^T$. Considering homogeneous vectors, they can be represented as $m = [u, v, 1]^T$ and $M = [X, Y, Z, 1]^T$, respectively. In the pinhole model, the 2D image projected point into the image plane, m, of the 3D real world point, M, can be determined as

$$m = K \left[R \middle| t \right] M \tag{3.1}$$

Figure 3.4: One view of camera IP4. A line thread was used to validate the measurements of the ball coordinates in 3D space, for heights above the ground plane.

Figure 3.5: One view of camera IP4. An electronic flash device produced a bright light allowing to synchronize all cameras for the ball in continuous motion sequences.

or

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

where R and t are the 3×3 rotation matrix and the 3×1 translation vector, which relate the world coordinate system to the camera reference frame (extrinsic parameters), and K is the 3×3

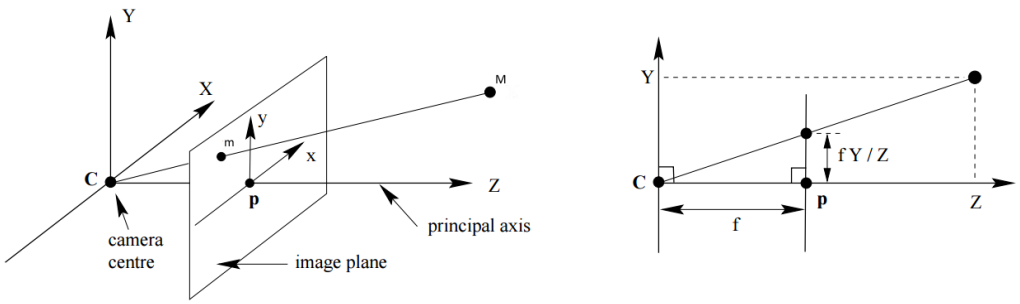


Figure 3.6: Pinhole camera geometry. The camera center C is the origin of a Euclidean coordinate system, the principal point p is the intersection between the image plane and the camera axis, and the focal length f is the distance from the pinhole camera to the image plane. From [9].

intrinsic camera matrix. The intrinsic camera matrix K is responsible for transforming 3D camera coordinates to 2D homogeneous image coordinates and contains the intrinsic parameters focal length (f_x and f_y) and principal point (p_x and p_y). Each one of these parameters represent a geometric feature of the pinhole camera, with the focal length symbolizing the distance from the camera to the image plane and the principal point representing the center point of the image plane. Once the internal camera matrix is estimated, it remains the same as long as the focal length and the image resolution does not change. Considering that K transforms the image coordinate system into a camera coordinate system, a new expression can be constructed by

$$m = K \left[I \middle| 0 \right] m_{cam} \tag{3.3}$$

where m_{cam} is defined to emphasize the projected point in camera coordinates, with the camera assumed as the origin of this coordinate system [9].

However, points in 3D space are expressed in terms of a world coordinate frame. As expressed before, R and t represent the extrinsic parameters which designate the coordinate system transformations from 3D world coordinates to 3D camera coordinates. The rotation-translation matrix $[R|t]$ is called the matrix of extrinsic parameters and translate world coordinates of a point M to the camera coordinate system, where t defines the position of the world coordinate system origin expressed in the camera coordinate system. They define

$$P = K \left[R \middle| t \right] \tag{3.4}$$

where P is the 3×4 camera projection matrix and represents the mapping from the image coordinate frame to the world coordinate system by

$$m = PM. \tag{3.5}$$

Considering equation 3.3, the camera coordinate frame and the world coordinate frame are

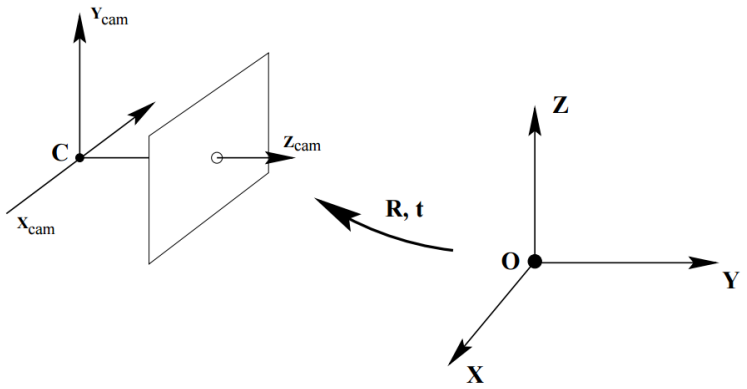


Figure 3.7: The Euclidean transformation between the world and camera coordinate frames. From [9].

related via a rotation and a translation [9], as illustrated in Figure 3.7. As a result, $m_{cam} = R(M - C)$, with C representing the coordinates of the camera centre in the world coordinate frame. Then, this equation may be written as

$$m_{cam} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = KR \left[I \middle| -C \right] M \tag{3.6}$$

where M is in a world coordinate frame. If $t = -RC$, the transformation of $m_{cam} = RM + t$ can be defined as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{3.7}$$

Expressing $x' = x/z$ and $y' = y/z$, the image coordinates are given as

$$\begin{cases} u = f_x x' + p_x \\ v = f_y y' + p_y \end{cases}. \tag{3.8}$$

As a consequence of some restrictions in the lens manufacturing process, straight lines in the world imaged through real lenses normally become curved on the image plane [95]. This is called the lens distortion, which can be radial and tangential. The radial distortion happens because the lens of the camera projects the points in the scene according to their distance from the origin of the image plane. For radial distortions, the distortion is practically zero on the center of the image and increases as it approaches the frontiers of the image plane [96], as depicted in Figure 3.8. Since

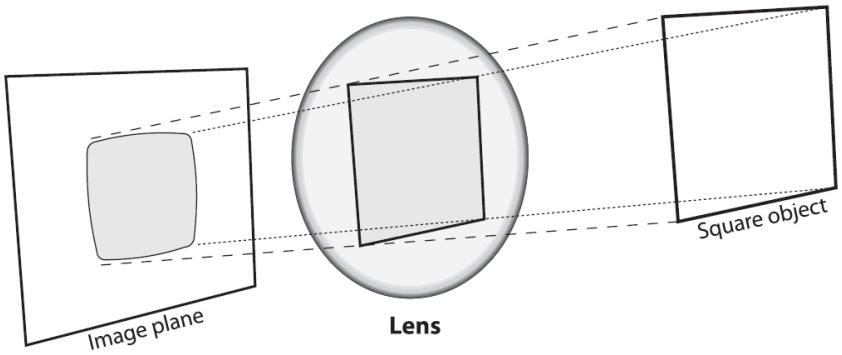


Figure 3.8: Radial distortion. A camera lens distorts the location of pixels near the edges of the image plane. Image rays farther from the center of the lens are bent compared to rays that pass closer to the center, accordingly curving the sides of a projected square object, for example. From [96].

each lens element is radially symmetric, almost all of the lens distortion is radially symmetric [95]. The tangential distortion occurs due to manufacturing defects, from the lens not being exactly parallel to the imaging screen [96].

The lens distortion should be taken into account as a nonlinear intrinsic parameter, however it cannot be included in the linear camera model described by the intrinsic camera matrix. Therefore, the above model can be extended as

$$x_d = x' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + 2p_1x'y' + p_2(r^2+2x'^2) \tag{3.9}$$

$$y_d = y' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + p_1(r^2+2y'^2) + 2p_2x'y' \tag{3.10}$$

where $r^2 = x'^2 + y'^2$, u_d and v_d are the undistorted image points in pixel units, k_1, k_2, k_3, k_4, k_5, and k_6 are the radial distortion coefficients and, p_1 and p_2 are the tangential distortion coefficients. Considering the relationship between the distorted and the undistorted image points [21], then

$$\begin{cases} u_d = f_x x_d + p_x \\ v_d = f_y y_d + p_y \end{cases}. \tag{3.11}$$

The distortion coefficients depend solely on the lens of the camera, so they can be used for different image resolutions on the same camera.

3.2.1 Intrinsic Calibration

The camera calibration process followed in this dissertation uses multiple views of a planar black and white chessboard pattern and was first introduced by Zhang [21]. This method calibrates cameras by solving a homogeneous linear system that creates homographic relationships between

several perspective views of the calibration pattern, assuming that it lies on the plane $Z = 0$. Considered a variant of the auto-calibration, it computes the camera parameters using the pattern's motion relative to a planar calibration object. This causes the calibration method to be more flexible, once the camera and the pattern can be moved freely and many images can be taken without measurements of the pattern's position. As said before in section 2.1, the responsiveness of the calibration algorithm to measurement errors can be improved by expanding the number of black and white squares on the chessboard, thus increasing the number of corners to be extracted from the pattern.

Zhang's camera calibration method is a two step process where a linear algebraic approximation is succeeded by a non linear searching [21]. The multiple view approach from it becomes more practical since it is simpler to capture numerous views of a calibration pattern than to use the motion of the camera to capture a static calibration scene e.g SfM, the difficulty to construct a 3D object to use as the calibration rig e.g. Tsai's method, or resorting to a 1D object that needs to be synchronously captured by all cameras.

Through Zhang [97, 21], m and M are related by a 3×3 matrix $H = [h_1\ h_2\ h_3] = K[r_1\ r_2\ t]$ denominated homography, where

$$\begin{bmatrix} u \\ v \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3.12}$$

considering $Z = 0$. Knowing that $r1$ and $r2$ are orthonormal,

$$h_1^T K^{-T} K^{-1} h_2 = 0 \tag{3.13}$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2, \tag{3.14}$$

which are the two fundamental constraints on the intrinsic parameters, given one homography. Through the closed-form solution called Singular Value Decomposition (SVD) [98], let

$$B = K^T K^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \tag{3.15}$$

that can be defined by a six-dimension vector

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{12} & B_{23} & B_{33} \end{bmatrix}^T. \tag{3.16}$$

Assuming ith column vector of H to be $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$, then

$$h_i^T B h_j = v_{ij}^T b. \tag{3.17}$$

For each homography and assuming the two basic constraints 3.13 and 3.14, two homogeneous

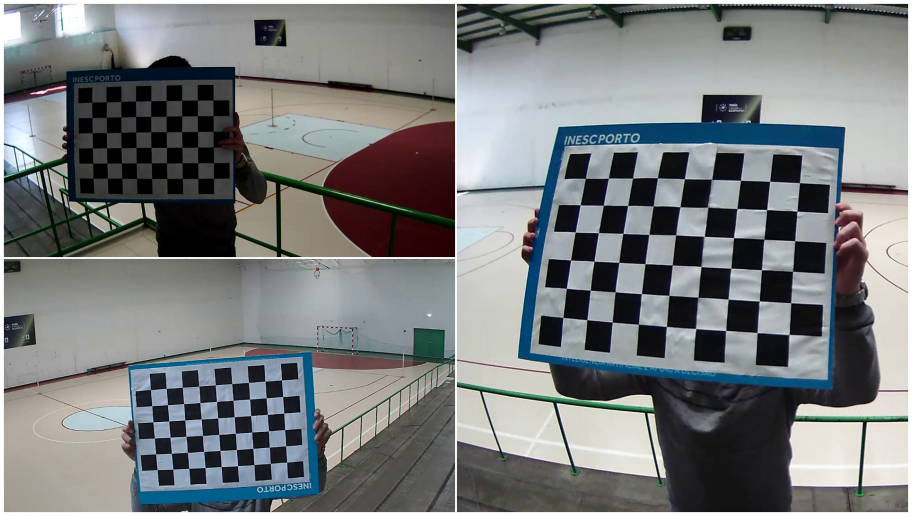


Figure 3.9: One frame example from three of the four installed cameras, displaying the used calibration pattern. To produce better results, the pattern was rotated and moved with several orientations in front of the cameras so that several frames with different pattern positions could be identified, thus detecting the corners from the black and white chessboard with differentiated poses.

equations proceed as

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0, \tag{3.18}$$

with $v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$. If n images of the model plane are observed, by piling up n equations such as the previous, then $Vb = 0$, where V is a $2n \times 6$ matrix. In order to obtain a single solution, at least three images are necessary ($n >= 3$). The closed-form solution is given by the eigenvector [99] of $V^T V$ associated with the smallest eigenvalue. Once b is estimated, all intrinsic camera parameters can be computed.

However, the solution was obtained by minimizing a not physically meaningful algebraic distance. Through a maximum likelihood inference such as the LM algorithm [57], a refinement of the intrinsic parameters was done. Assuming n images of the planar pattern and m image points corrupted by independent and identically distributed noise on the pattern plane, a maximum likelihood estimate was obtained by minimizing the distance:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \left\| m_{ij} - m(K, R_i, t_i, M_j) \right\|^2 \tag{3.19}$$

where $m(K, R_i, t_i, M_j)$ is the projection of point+ M_j in a calibration image i, according to equation 3.12.

The calibration procedure started by moving a checkerboard pattern in front of each camera individually, as seen in Figure 3.9. Only the images where the corners of the chessboard pattern

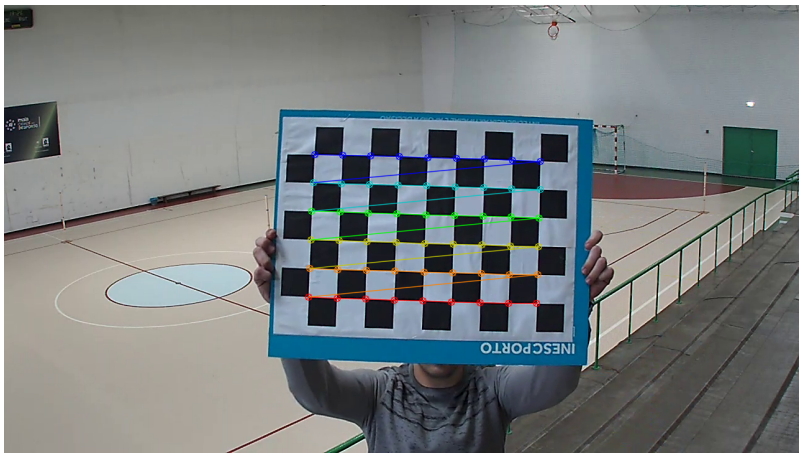
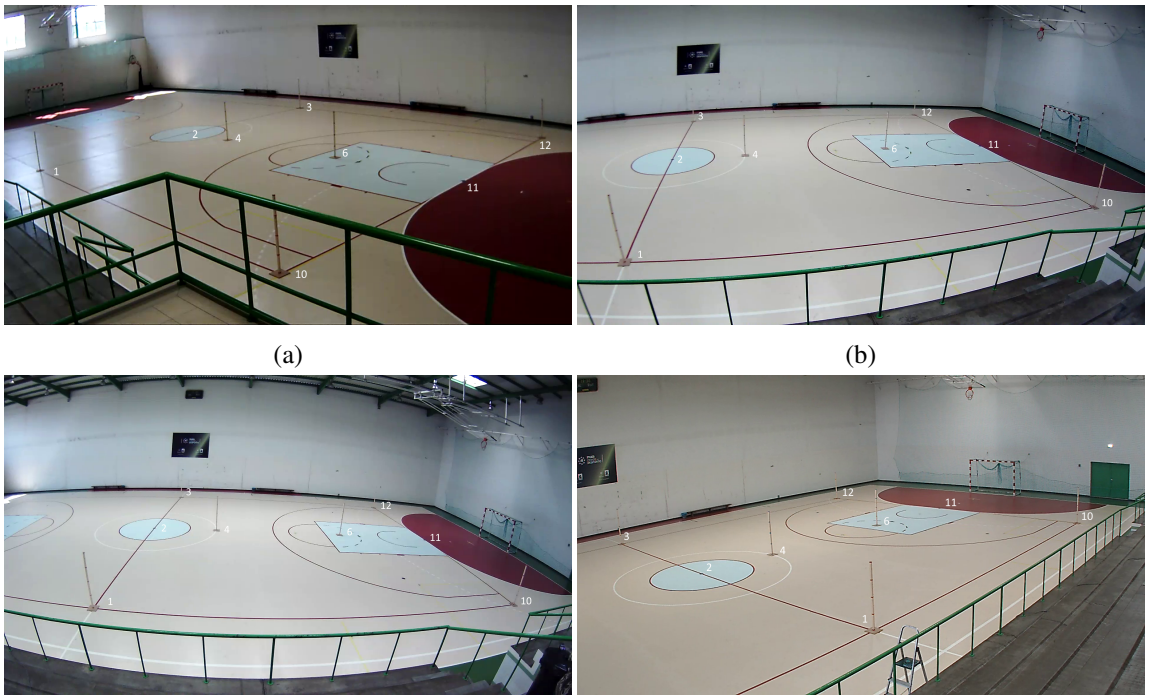


Figure 3.10: One frame example from camera IP4, displaying a colored grid covering the planar pattern found corners.

were all correctly identifiable were considered as valid. The process of identifying and validating the calibration images followed a defined frame skip value to allow a sufficient interval for the calibration pattern to change its orientation and avoid redundancies. In planar patterns with shapes in a rectangular form such as chessboards, an accurate calculation of the coordinates of corners is required. For that, a geometry-based corner detection algorithm was used to measure differential geometry features of corners and find an estimate of corners' coordinates [96]. In order to make the process more visual, the output of the corner detection algorithm was displayed by overlaying a colored visible grid to notify that all the chessboard corners were found, as illustrated in Figure 3.10.

To improve the accuracy of the coordinates from the black and white chessboard corners, it was used an iterative sub-pixel refinement method [96] to extract more exact corner positions since a more real-valued resolution was needed instead of the approximations to integer pixel values previously obtained. The calculation of the sub-pixel location used the mathematical fact that a dot-product between a vector and a orthogonal vector is equal to 0, a circumstance that happens on corner locations. The method iterated from a starting point value and continues until the termination criterion were reached. It was decided empirically to compute a maximum number of iterations of 30 and the desired accuracy of the sub-pixel values at which the iterative algorithm stopped was 0.1 pixels.

To provide a qualitative measure of accuracy, reprojection errors for each pattern view were calculated, i.e. the distance between a pattern point detected in a calibration image frame, and the corresponding projection into the same image. This provided a useful assertion of the average reprojection error in each calibration chessboard image. Using this information, image frames with a reprojection error above a certain value were excluded from the calibration procedure and a recalibration of the camera was processed considering only the calibration image frames with a reprojection error below the defined maximum acceptable reprojection error.



(a) (b)

(c) (d)

Figure 3.11: Six markers were placed around right-half of the field to allow for a correct estimation of the camera's locations and orientations to the same world reference. In each marker, points were labeled with colored duct tape and spaced with 0.4 meters in height. (a) View of camera IP1. (b) View of camera IP2. (c) View of camera IP3. (d) View of camera IP4.

The calibration process returned 8 distortion coefficients following a rational model of the lens distortion computation [100], equations 3.9 and 3.10.

3.2.2 Extrinsic Calibration

The extrinsic parameters obtained during the intrinsic parameters calibration step were related to each i-th view of the pattern. This means that all cameras had multiple values of R and t regarding the calibration object motion during the calibration process and these values only represent a transformation from the camera coordinate system to the world coordinate frame defined on the pattern calibration.

So, to ensure that the R and t values were computed for the same world coordinates defined on the court field, six markers with a height of 1.6 meters were placed on positions 1, 3, 4, 6, 10 and 12 of the field, considered the world reference frame. Points were labeled with colored duct tape in each marker and spaced with 0.4 meters in height, thus establishing five points for each one to be used as a calibration object point with a total of 30 points, as illustrated in Figure 3.11. To improve the accuracy, the midpoints of the center line (position 2) and the base line (position 11) of the basketball field, representing the width lines of the world frame, were also considered, thus increasing the total points used to 32. Hence, the field can be considered a calibration object because the extrinsic parameters were obtained from known 3D world points existent on the markers

and were viewed in the same 3D location by every camera, thus guaranteeing that every position of camera centers and camera's headings are all according to the same world reference frame.

This problem of obtaining the position and orientation of a camera given its intrinsic parameters and a set of n correspondences between 3D points and their 2D projections is called the Perspective-n-Point (PnP) [101]. The assumption made in the PnP is that the camera is already calibrated, which defines that all of the intrinsic parameters are already estimated. The formulation of the PnP problem can be explained as follows: given a set of correspondences between 3D points in a world reference frame and their 2D projections onto the image plane, PnP attempts to estimate the pose (R and t) of the camera with respect to the world and the focal length f and to transfer the world coordinates to the camera coordinates frame. Four PnP solver methods were studied: the Iterative PnP, the EPNP [102], the P3P [103] and the RANSAC PnP method [101]. However, P3P was not considered because it needs exactly four point correspondences to solve the PnP problem. Though four correspondences can be acceptable to estimate the pose, it was advantageous to analyze larger number of point sets so that some redundancy could be introduced and the sensitivity to noise diminished. Since it were considered n points, the remaining three approaches that solve the problem of computing camera pose were analyzed and compared.

The PnP methods received as parameters both the camera matrix and the distortion coefficients that were already computed, the real 3D object points from the markers and the pixel positions of each point projected on the image of the camera. Therefore, correspondences between 3D and 2D points granted the camera pose from each different PnP solver. Then, using the previously estimated intrinsic parameters and distortion coefficients and the estimated rotation matrix and translation vector from the PnP solvers implemented, a projection of the 3D world points onto the image plane was done (see equation 3.2). To prove that the estimated extrinsic parameters were accurate, an intersection must occur or, at least, the points projected must appear closer to the correspondent image points given before.

The PnP problem solvers can be classified as non-iterative or iterative methods. Most of the non-iterative methods start with an estimation of the 3D points position regarding the camera coordinate system by solving them for the points depth [102]. After, it will be easy to retrieve camera's position and orientation as the Euclidean motion aligns these positions on the given coordinates in the world coordinate system [102]. Non-iterative methods can be efficient, but their limitation lie on instability in the presence of noise, especially when $n \leq 5$. As said, the stability of the non-iterative methods can be improved by introducing redundant points as supplementary information. If redundant points are unavailable, accurate results can be achieved by introducing iterative schemes based on the minimization of non-linear cost functions.

3.2.2.1 Efficient PNP Method

The Efficient PNP method (EPNP) [102] consists of a non-iterative solution and is considered by its authors to have lower computational cost than other non-iterative methods, and as fast as iterative ones with only a slight loss of accuracy. Differently than most approaches, that attempt to

solve for the depths of the reference points in the camera coordinate system, the EPNP coordinates are given as a weighted sum of four non-coplanar virtual control points.

Defining n points whose 3D coordinates are known in the world coordinate system, p_i with $i = 1, ..., n$; and 4 control points, c_j with $j = 1, 2, 3, 4$; the weighted sum of the control points is

$$p_i = \sum_{j=1}^{4} \alpha_{ij} c_j, \tag{3.20}$$

with α_{ij} as the homogeneous barycentric coordinates.

In theory, the control points could be chosen arbitrarily but, in practice, the accuracy increases if one is selected as the centroid of the reference points and the rest to form a basis aligned with the principle directions of the data. This can be considered as conditioning the linear system of equations by normalizing the point coordinates in a similar way as the classic DLT algorithm [9]. Given this formulation, the coordinates of the control points in the camera reference frame are the unknowns of the PnP problem.

Let K be the camera internal calibration matrix and u_i the the 2D projections of the p_i reference points, then

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \sum_{j=1}^{4} \alpha_{ij} c_j K p_i = K \tag{3.21}$$

with c_j in a camera coordinate system. The unknown parameters of this linear system are the 12 control point coordinates, and two linear equations for each reference point will appear defining the coordinates of the control points in the camera reference frame as the unknowns of the PnP problem.

3.2.2.2 Iterative PnP Method

Considering an iterative approach, the Iterative PnP method used was based on the DLT method followed by an iterative LM optimization since it estimates the camera pose through the sum of the squared distances between the observed 2D projections and the projected 3D points that minimize the reprojection error.

This method can be divided into planar and non-planar structure cases. On non-planar structure cases, it uses the DLT method to compute the extrinsic parameters. On planar structure cases, where all the points lie on the same plane, a homography (see equation 3.12) is used to get the extrinsic parameters. On both cases, it is used an LM optimization to refine the estimations that were referenced. In this dissertation, it was followed the non-planar structure. First, it determined the projection matrix P by solving a linear equations system defined by n 2D projection points, m_i,

from n 3D world points, M_i:

$$\begin{cases} \frac{P_{11}X_i+P_{12}Y_i+P_{13}Z_i+P_{14}}{P_{31}X_i+P_{32}Y_i+P_{33}Z_i+P_{34}} = u_i \\ \frac{P_{21}X_i+P_{22}Y_i+P_{23}Z_i+P_{24}}{P_{31}X_i+P_{32}Y_i+P_{33}Z_i+P_{34}} = v_i \end{cases} \tag{3.22}$$

where P_{ij} are the elements of the projective matrix P. Through the SVD, a solution was found using a linear system in the form of $Ap = 0$, with p as a vector composed by the coefficients P_{ij}.

Only then, it extracted the camera pose using the equation 3.2. Considering that pixel locations suffer from an independent and identically distributed noise, a refinement was done using an iterative LM optimization, already described in section 3.2.1. Empirically, the maximum number of iterations defined was 20.

3.2.2.3 RANSAC PnP Method

The RANSAC method [101] is a non-deterministic iterative method that estimates parameters from observed data producing an approximate result as the number of iterations increase. It can be used along with a PnP method to produce a solution robust to outliers in a set of n point correspondences, considering incorrect matches (outliers) are eliminated to estimate the camera pose with a certain probability of obtaining a good solution. The PnP method implemented along with the RANSAC algorithm was the EPNP, described in section 3.2.2.1.

First, the RANSAC PnP method chose five random 2D-3D correspondences differently for each iteration. Then, it calculated cameras' poses using these five correspondences through an EPNP solution and the reprojection error for each reprojection of the 3D point and their original 2D projection. Considering an initial specified threshold, an estimation would be considered an inlier if the reprojection error was less than the threshold value. Alternatively, it would be considered an outlier. The threshold value represented the maximum allowed distance between the 2D observed and computed point projections in pixel units. As mentioned, five different sets of 2D-3D correspondences are randomly selected for each iteration, so this process of reducing the influence of outliers resulting from inaccurate correspondences was repeated for each iteration of the RANSAC method until it reached a maximum number of iterations or a maximum number of inliers. The maximum number of iterations and inliers were defined as 100 and the inlier threshold value used by the RANSAC method was defined as 8 pixels. All values were chosen empirically.

3.3 Triangulation

After the calibration stage and retrieval of the intrinsic and extrinsic parameters of the all the cameras involved, a 3D reconstruction of the target object - the ball - was the next step. This was possible through triangulation by calculating a point in 3D space given its projections onto two views and resorting to the camera matrices of those views.

The problem of determining a point's 3D position from a set of corresponding image locations and known camera positions is defined as triangulation. Assuming that both extrinsic and intrinsic camera parameters are known exactly, i.e. both camera matrices; or at least with great accuracy compared with a pair of matching points in two images or views [9], two rays can be easily reconstructed from the corresponding projected points with the intersection of those rays assuring the knowledge of their 3D position [9].

Considering that a 3D world point M is visible in two different views from the scene, P and P' are two known camera projection matrices and, m and m' define projections of the point M in the two images, the two rays in space corresponding to the two image points can be computed [104]. As stated before, the triangulation problem is to encounter the intersection of these two rays in space. Apparently, this may arise as an irrelevant issue once we consider that the intersection of these two lines in space will always be perfect. Unfortunately, due to the presence of noise and errors in the measured image points m and m', the projected rays can not be guaranteed to cross each other.

This implies that there will not exist a point M which accurately meets $m = PM$, $m' = P'M$; causing the image points to not satisfy the epipolar constraint $m'^T Fm = 0$, as seen in Figure 3.12, with F as the Fundamental matrix [9]. As mentioned before in section 2.2, the epipolar geometry is defined as the ideal geometry relation between the 3D points and their projections onto the images from two cameras pointed to a 3D scene in different positions. This is a problem that triangulation can solve by estimating M using the projected points m and m', and the camera matrices P and P'. Assuming τ as a triangulation method used to compute a 3D space point M, we can write

$$M = \tau(m, m', P, P'). \tag{3.23}$$

Thus, there have been proposed a few different techniques and methods to achieve better solutions for 3D reconstruction [104]. Three of those triangulation methods were evaluated to solve this 3D reconstruction problem: the Midpoint Method, the Linear Least-Square (LS) Method and the Iterative Linear-LS Method.

3.3.1 Midpoint Method

The Midpoint Method [105, 106] is the most intuitive method to achieve an acceptable 3D reconstruction and is based on a simple linear geometry problem. Its objective is to achieve the midpoint of the perpendicular between the two back-projected rays, as seen in Figure 3.13.

Knowing both 3×4 projective camera matrices, P and P', from the two cameras, we can decompose camera's internal parameters, K and K'; the rotation matrices, R and R'; and the translation vectors, t and t'. Afterwards, the back-projected rays need to be determined. These will be vectors with initial point on the camera centers, p and p', and crossing the projected points on each image plane, m and m', with end point on the infinity.

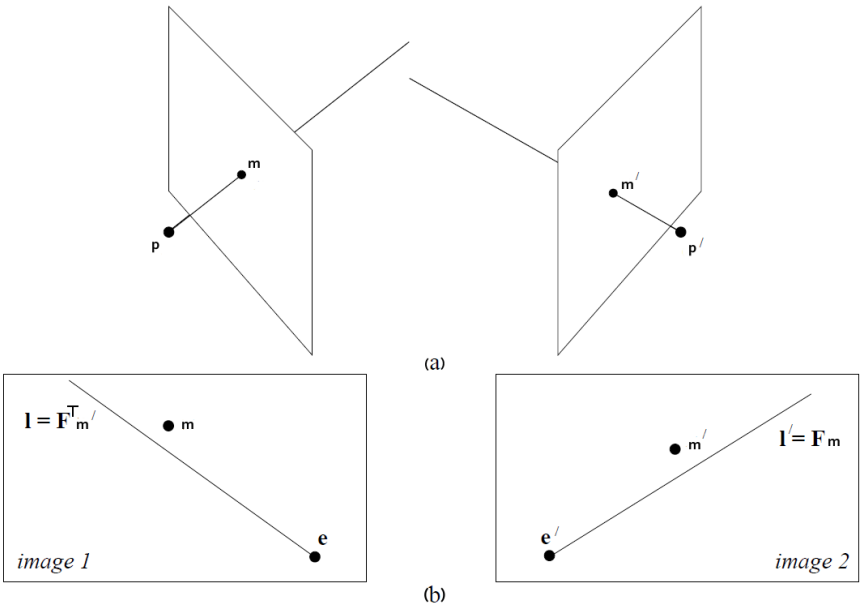


Figure 3.12: (a) Rays back-projected from measured points m and m' are skew in 3D space. (b) The epipolar geometry for m, m'. The measured points do not satisfy the epipolar constraint. The epipolar line $l' = Fm$ is the image of the ray through m, and $l = F^T m'$ is the image of the ray through m'. Considering rays can not intersect, m' does not lie on l', and m does not lie on l. From [9].

Considering equation 3.2, multiplying the image point m by the inverse of the intrinsic matrix K will partially transform the point from image coordinates to camera coordinates

$$K^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix}. \tag{3.24}$$

This conveys that the 3D coordinates of the object cannot be retrieved because the value z is not known yet. However, the coordinates must be situated on a line from the object to the camera origin that passes through the imaging screen. By knowing that the image plane is at distance f, where f is the focal length of the camera, a ray that passes through the object can be defined as

$$r = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} f\alpha \\ f\beta \\ f \end{bmatrix}. \tag{3.25}$$

After the back-projected ray is calculated, a line between the nearest points of each ray is estimated and its midpoint will be the estimated 3D point M. Defining a and a' as the points that are

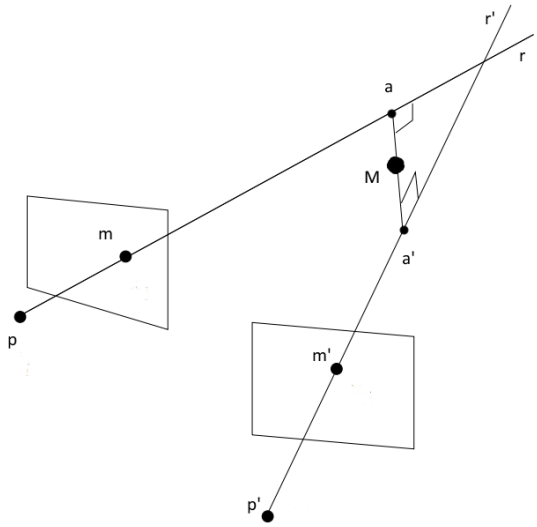


Figure 3.13: Midpoint method triangulation. In a Euclidean frame, the midpoint M of the perpendicular defined by a and a' and between rays r and r' can be used to give an estimate of the 3D point. Each ray is defined by the camera center p and p' and the projected image point m and m' of the same 3D point. From [105].

connected by the line, its midpoint can be defined as

$$M = p + 1/2(a - a'). \tag{3.26}$$

Although this approach seems practical and attractive, it only applies in a Euclidean coordinate frame where distance and perpendicularity can be measured [106].

3.3.2 Linear Least-Squares Method

The Linear Least-Squares Method [104] is the most common method for 3D reconstruction. Considering $m = s(u,v,1)^T$ in equation 3.5, where (u,v) are the observed point coordinates and s is an unknown scale factor. Designating p_i^T as the i-th row of the matrix P, the equation may be written as

$$\begin{cases} su = p_1 M \\ sv = p_2 M \\ s = p_3 M \end{cases} \tag{3.27}$$

Eliminating s using the third equation,

$$\begin{cases} up_3^T M = p_1^T M \\ vp_3^T M = p_2^T M \end{cases} \tag{3.28}$$

From two views, we obtain a total of 4 linear equations in the coordinates of the m, which may be written in the form $Am = 0$ for a suitable 4×4 matrix, A. This can be composed in

$$A = \begin{bmatrix} up_3^T - p_1^T \\ vp_3^T - p_2^T \\ u'p_3'^T - p_1'^T \\ v'p_3'^T - p_2'^T \end{bmatrix}. \tag{3.29}$$

The solution can be determined only up to scale and a non-zero solution for M is needed [9]. The Linear Least-Squares method assumes that the solution point M is not at infinity, otherwise $M = (X,Y,Z,1)$ could not be assumed [104]. By designating that set of homogeneous equations, $Am = 0$ is reduced to a set of four inhomogeneous equations in three unknowns. The least-squares solution to these inhomogeneous equations can be found using two different methods: the method of pseudo-inverses [9] or the SVD [98] with the latter as the one chosen in this dissertation. Two different implementations of the Linear LS method were used and a comparison on the 3D estimation accuracy was done.

3.3.3 Iterative Linear Least-Squares Method

The Iterative Linear Least-Squares Method [104] has the purpose to strive against some inaccuracies presented by the Linear Least-Squares method since the value being minimized $||Am||$ has no geometric meaning. Taking this into account, the idea of the iterative method is to change the weights of matrix A flexibly so that the weighted equations can give an image coordinate measure of a geometric error function.

Considering that the point M will not exactly satisfy the system in equation 3.28, an error

$$\varepsilon = up_3^T M - p_1^T M \tag{3.30}$$

will exist. However, the value that needs to be minimized is the difference between the projection of M and the measured image coordinate u corresponding to p_1^T/p_3^T, determining a new error minimization

$$\varepsilon' = \varepsilon/p_3^T M = u - p_1^T M/p_3^T M. \tag{3.31}$$

If the equations 3.28 are weighted by $1/p_3^T M$, the resulting error will minimized. For the second camera, the weight will be the same $1/p_3^T M$. Since the M value is an undetermined value that will only be known after the equations are solved, an iterative procedure will happen to adapt the weights by setting $w_0 = w_0' = 1$ and solving the system of equations to find a solution M_0. The result of M_0 will be precisely the same obtained from the previous Linear Least-Square method and, having found it, the weights can be computed.

The iteration was repeated various times, multiplying the equations 3.28 in each i-th step by $1/p_3 M_{i-1}$ or $1/p_3' M_{i-1}$, where M_{i-1} represents the solution found in the preceding iteration. The

errors, for each equation in 3.28, in image measurements are:

$$\varepsilon_i = u - p_1^T M_i / p_3^T M_i \tag{3.32}$$

and

$$\varepsilon_i' = u' - p_1'^T M_i / p_3'^T M_i. \tag{3.33}$$

The iterations converge to a specific M value after the changes in the weight become insignificant. Empirically, it was defined $\varepsilon = 0.005$ as the threshold value of the weight that would not modify the estimations from the iterative process. Not only that, the number of maximum iterations was defined as 10 at most [104].

Chapter 4

Results and Discussion

In this Chapter, the results of the methodologies previously described in Chapter 3 are evaluated and discussed. Moreover, the whole testing sequence is described with emphasis on the choices made throughout. As mentioned before, the proposed solution was divided into three phases. The first consisted on computing the intrinsic and distortion calibration parameters for a setting of cameras. The second summed up as a 3D-2D correspondence of points to estimate camera poses. The third and final one had the purpose of obtaining 3D information from an object by using and intersecting information from two cameras with relation to the same viewed scene.

The testing sequences were performed on a pavilion with a basketball field, where several image sequences were captured using four different cameras with different positions to allow full feasibility for the testing and evaluation process.

4.1 Intrinsic Calibration

As previously mentioned in Section 3.2.1, Zhang's calibration method was used for each camera individually to estimate its intrinsic parameters and distortion coefficients. For that, several calibration images of the chessboard planar pattern were taken under different orientations by moving it closer to the camera and inside its field of view. The process of taking such images followed up a skip frame value of 20. The chessboard used for this procedure was a black and white pattern with a size of 9×6 squares of 5.4cm, as presented on Figure 4.1. The values obtained for cameras IP1 and IP4 (see Figure 3.2) are expressed on Tables 4.1 and 4.2.

As mentioned before, the principal point describes the value of the image center and is represented by the center width (p_x) and the center height (p_y). Concerning that the image resolution defines the maximum number of pixels for each axis of the image coordinate frame, the values of the center width and height must be approximately half of these. As cameras IP1 and IP4 have the same image resolution, 1280×720, the values of their image center width and height will be roughly similar too. Accordingly, $1280/2 = 640 \simeq 639.03 \simeq 640.14$ pixels and $720/2 = 360 \simeq 356.85 \simeq 361.99$ pixels which are an indication of coherence on the values estimated. Not only that, the values estimated for focal width and focal length must also denote one

Figure 4.1: The calibration pattern used was a 9×6 chessboard pattern, with a square size of 5.4 centimeters. The calibration pattern was moved around the camera's field of view, exploring as much as different orientations to improve the intrinsic parameters estimation.

Table 4.1: Camera IP1 intrinsic parameters obtained using 41 frames.

	Value	Unit
Image Width	1280	pixels
Image Height	720	pixels
Focal Width (f_x)	1394.75	pixels
Focal Height (f_y)	1437.88	pixels
Center Width (p_x)	639.03	pixels
Center Height (p_y)	356.85	pixels
Avg. Reprojection Error	0.491	pixels

Table 4.2: Camera IP4 intrinsic parameters obtained using 100 frames.

	Value	Unit
Image Width	1280	pixels
Image Height	720	pixels
Focal Width (f_x)	1176.09	pixels
Focal Height (f_y)	1175.60	pixels
Center Width (p_x)	640.14	pixels
Center Height (p_y)	361.99	pixels
Avg. Reprojection Error	0.445	pixels

important property called aspect ratio, f_x/f_y. Verifying for both cameras, $f_x/f_y = 0.97$ for IP1 and $f_x/f_y = 1.00042$ for IP4, the values are nearly 1 which dictate that the pixels are square [97] and validate the results obtained.

However, there were some issues regarding the calibration of the other two cameras, IP2 and IP3. The former had a problem during the video capture of the chessboard pattern and it was elim-

inated from any experiments regarding the image acquiring that was carried out. The latter was considered successful until the stage of calibrating camera's extrinsic parameters. As mentioned before, a calibration image was considered valid when the chessboard was correctly found and the average reprojection error of the pattern points was below a certain value (0.5 pixels). Despite its average error manifested values similar to the ones presented in cameras IP1 and IP4, after a full analysis of the images considered as valid frames, it was verified that the chessboard was solely moved around a portion of the image and it did not reach the edges of the camera's image. This had implications because this camera presented greater radial distortion than the others, which increase on the edges of the image.

Not only that, but since there was a calculation of the reprojection error for each frame and an elimination of the images with a reprojection error below 0.5 pixels, some of the valid frames that remained for the recalibration process were virtually similar, since the chessboard's orientation and position appeared to be practically the same. Thus, this resulted in some redundancy in the pattern's pose estimation which did not grant variability on the calibration of the camera's focal length and lens distortion. To overcome this challenge, two other calibration procedures were done adjusting the skip frame values. For the 2nd calibration, the skip frame value was altered to 5 frames. For the 3rd calibration, the skip frame value was defined as 1 frame, allowing the method to utilize every frame of the calibration video capture and make the maximum use of it. Also, the value of the highest reprojection error accepted for each frame was changed to 0.55 pixels to prevent that a large number of frames could be eliminated and increased the number of frames with an adequate error. In Table 4.3, the estimated values for the three calibration procedures on camera IP3 are displayed.

Table 4.3: Camera IP3 intrinsic parameters obtained with 38, 92 and 203 frames (from left to right).

	1st	**2nd**	**3rd**	**Unit**
Image Width	1920	1920	1920	pixels
Image Height	1080	1080	1080	pixels
Focal Width (f_x)	1195.05	3516.18	1178.76	pixels
Focal Height (f_y)	1187.95	3356.99	1174.55	pixels
Center Width(p_x)	943.02	986.49	955.20	pixels
Center Height (p_y)	526.97	530.83	542.18	pixels
Avg. Reprojection Error	0.4521	0.4516	0.5035	pixels

Looking into the obtained results, a conclusion regarding the best estimation of the intrinsic parameter of camera IP3 could not be formulated since the average reprojection error was influenced by a change on the maximum acceptable error for each calibration procedure and a low average value does not suggest better estimations produced. Not only that, the focal length and the principal point estimations seemed reasonable for each calibration, except for the exaggerate focal length on the 2nd one which can indicate that this procedure has the worst estimations. Nonetheless, all camera calibration procedures were considered for the next phase where a decision about which one achieved the most accurate intrinsic parameters estimation of camera IP3.

The distortion coefficients values, presented on Table 4.4, had the important purpose of image distortion removal, which was a critical step before the triangulation process. To verify if the tabulated estimations were acceptable, a distortion elimination of the image plane for each camera was performed and the two images were compared using the field lines as references. Carrying out a comparison on these two images, the lines slightly curved on the images with distortion should appear rectilinear on the undistorted images. However, as shown in Figure 4.2, image distortion from camera IP3 could not be successfully removed which could constrain some estimations of this particular camera on the following steps, as opposed to camera IP4 that successfully corrects its lens distortion (Figure 4.3).

Table 4.4: Obtained distortion coefficients for the three cameras IP1, IP3 and IP4. For the latter, the coefficients regard the three implemented calibration procedures. $k1$, $k2$, $k3$ and $k4$ are the radial distortion parameters; $p1$ and $p2$ are the tangential distortion parameters (equations 3.9 and 3.10).

	IP1	**IP4**	**IP3 (1st)**	**IP3 (2nd)**	**IP3 (3rd)**
k1	-0.2212	-0.1418	-0.1534	0.0163	-0.0211
k2	3.3711	-1.1966	0.6430	-11.8167	-0.6699
k3	32.0545	1.1676	0.0142	59.9473	13.2406
k4	0.2420	0.2321	0.2443	-0.0491	0.4789
k5	2.5907	-1.2092	0.6055	8.7850	-1.7970
k6	38.3855	0.4841	0.1505	19.3059	17.1218
p1	-0.0091	0.0024	0.0108	0.1151	0.0067
p2	0.0065	-0.0023	0.0011	-0.0025	0.0002

Although an explicit interpretation of the estimated distortion coefficients can be done by removing the lens distortion from images and checking if curved lines are corrected, some conclusions can be attained by observing the values listed on Table 4.4. However, some conclusions about the estimated values can be drawn. Although it is normal for some cameras to present some distortion values slightly higher than others, specially considering the greater difference of lens distortion between camera IP3 and cameras IP1 and IP4, it is not common to have much higher values on particular distortion coefficient parameters since they absorb calibration errors. Some of these values created an alarm concerning the accuracy from the calibration process, such as $k3$ and $k6$ of camera IP1 and the two latter calibration procedures from camera IP3.

4.2 Extrinsic Calibration

After the assessment of the intrinsic and distortion coefficients for all cameras, an estimation of the pose of the calibrated cameras was performed through a PnP solution, as referenced in Chapter 3.3.

Altogether, 32 points were defined in the basketball field as the value n (Section 3.2.2) for the 3D-2D point correspondences and retrieved the extrinsic parameters for all cameras. Some points were not noticeable in all cameras thus defining lesser n points for cases of cameras IP1 and IP3. After the estimation of the rotation matrix (R) and translation vector (t) of each camera,

(a)

(b)

Figure 4.2: One view from camera IP3. (a) Original image (with distortion). (b) The same image after lens distortion removal. It is noticeable that the handrail continues to appear curve.

those values were used to project 3D points to each camera's image plane (see equation 3.14) and allowed a viable error comparison. The error was calculated using the 2D Euclidean distance between the image points manually extracted $x = (x_x, x_y)$ and their image projection $xp = (x_{px}, x_{py})$

(a)

(b)

Figure 4.3: One view from camera IP4. (a) Original image (with distortion). (b) The same image after lens distortion removal. It is noticeable that the stands and the midcourt line do not appear curve any longer.

calculated considering the extrinsic parameters, R and t, obtained through PnP,

$$d(x, x_p) = \sqrt{(x_x - x_{px})^2 + (x_y - x_{py})^2}. \tag{4.1}$$

Tables 4.5 and 4.6 show the error calculation of the extrinsic parameters for cameras IP1 and IP4

obtained with the three PnP solvers considered in this dissertation, and described in section 3.2.2. The data suggests that the PnP results on camera IP4 are slightly better than the ones on camera

(a) (b)

Figure 4.4: One view of camera IP4. (a) 6 markers, with 5 different heights labelled with red duct tape, distributed around the basketball field. Those points, plus two others from the baseline and the center line, formed the 32 3D world points used to estimate the pose of the camera, a value that depended on the number of points observable from each camera. (b) White dots represent the projected points estimated using the Iterative solution; they must coincide with the image of the 3D world points defined on the markers and on the field.

Table 4.5: Error estimation for the obtained extrinsic parameters of camera IP1, considering the Iterative, EPNP and RANSAC PnP solutions, and $n = 24$ image points.

	Iterative PnP	**Efficient PNP**	**RANSAC PnP**	**Unit**
Avg. Reprojection Error (d)	5.9	7	5.9	pixels
Std. Deviation (σ)	4	4.6	7.8	pixels

Table 4.6: Error estimation for the obtained extrinsic parameters of camera IP4, considering the Iterative, EPNP and RANSAC PnP solutions, and $n = 32$ image points.

	Iterative PnP	**Efficient PNP**	**RANSAC PnP**	**Unit**
Avg. Reprojection Error (d)	1.2	1.7	1.2	pixels
Std. Deviation (σ)	0.7	1	0.7	pixels

IP1. A closer look indicates that the PnP solution estimated with the iterative and RANSAC methods achieved similar results on both cameras, with the EPNP achieving the worst average errors.

Extending the evaluation to a spatial issue, every projected points on the image plane of camera IP4 were almost equal to its 3D correspondence regardless of the field position. An example of the use of the estimated camera pose to project the 3D points on the image plane and compare it to the original image, is shown on Figure 4.4. They both illustrate the arrangement of the markers around the basketball field that allow for a considerable number of 3D-2D point correspondences to be used for the PnP solver with the latter exemplifying the projection of the calculated image points x_p, producing a visual comparison against the actual 3D points on the imaging screen. On camera IP1, the average Euclidean distance of the projected points is spatially conditioned since for each

method there were some points with a great projection while others appeared rather off of the original marked position, as seen in Figure 4.5. A comparison made between each PnP solution for camera IP1 and its projected points on the image, showed that the points from markers on positions 1, 3, 4 and 10 had a diminished error on all images; on the other hand, points from markers on positions 6 and 12 possessed a larger variation between PnP solutions. For instance, the projected points from the marker on position 6 were corrected on the RANSAC solution contrasting to the points from the marker on position 12 that suffered a wrongful alteration.

The difference between the two cameras suggest that the accuracy of the intrinsic parameters calibration on camera IP1 is lower. Since the projection of points on camera IP1 is erroneous to a greater extent on a specific space of the basketball field, this can be an evidence of an imprecise calibration of the distortion coefficients, particularly in the upper right area of the image plane. It could be solved using the motion of the chessboard correctly on that area of the imaging screen to retrieve more reliable values from the distortion coefficients. Since new image acquiring was not feasible, a recalibration of the intrinsic parameters was performed by resorting to every frame of the calibration video. However, the calibration procedure only found a few more valid frames, and an enhancement was impractical to reach.

As previously stated, the intrinsic parameters calibration for camera IP3 was considered successful on the first attempt. However, the average Euclidean distances for the 1st calibration displayed some discrepancies in certain field areas. It was then decided that the calibration procedure should become more error-free and new values for the intrinsic parameters and distortion coefficients were computed. Altogether, two more calibrations were performed. Those intrinsic parameters estimated were employed to evaluate the sharpness of the extrinsic parameters calibration.

On Table 4.7, values of the average Euclidean distances from the 3D-2D point correspondences to their projection considering the estimated R and t for each PnP solution and their standard deviations are presented.

Table 4.7: Error estimation for the obtained extrinsic parameters of camera IP3 1st, 2nd and 3rd calibration procedure, considering the Iterative, EPNP and RANSAC PnP solutions, and $n = 26$ image points.

1st calibration	**Iterative PnP**	**Efficient PNP**	**RANSAC PnP**	**Unit**
Avg. Reprojection Error (d)	7.4	21.6	10.8	pixels
Std. Deviation (σ)	3.5	6.7	14.3	pixels
2nd calibration	**Iterative PnP**	**Efficient PNP**	**RANSAC PnP**	**Unit**
Avg. Reprojection Error (d)	68.2	77.7	91.6	pixels
Std. Deviation (σ)	46.5	38.6	90.6	pixels
3rd calibration	**Iterative PnP**	**Efficient PNP**	**RANSAC PnP**	**Unit**
Avg. Reprojection Error (d)	8.9	16.6	11.1	pixels
Std. Deviation (σ)	3.5	6.4	14	pixels

Analyzing the average error of the three calibration procedures, the results suggest that the 2nd one obtained the most inaccurate parameters. By observing Figure 4.6 and the values obtained on the intrinsic parameters calibration, the estimation of the focal length is excessively high as

(a)

(b)

(c)

Figure 4.5: One view of camera IP1. White dots represent the projected points estimated using the estimated PnP solution for the camera pose. The projected points must coincide with the image of the 24 3D world points defined on the markers and on the field. (a) Iterative PnP solution. (b) Efficient PnP solution. (c) RANSAC PnP solution.

opposed to the other two calibrations of camera IP3. Considering that, these values were deemed as defective and removed from the experiments. On the other hand, both the 1st and the 3rd

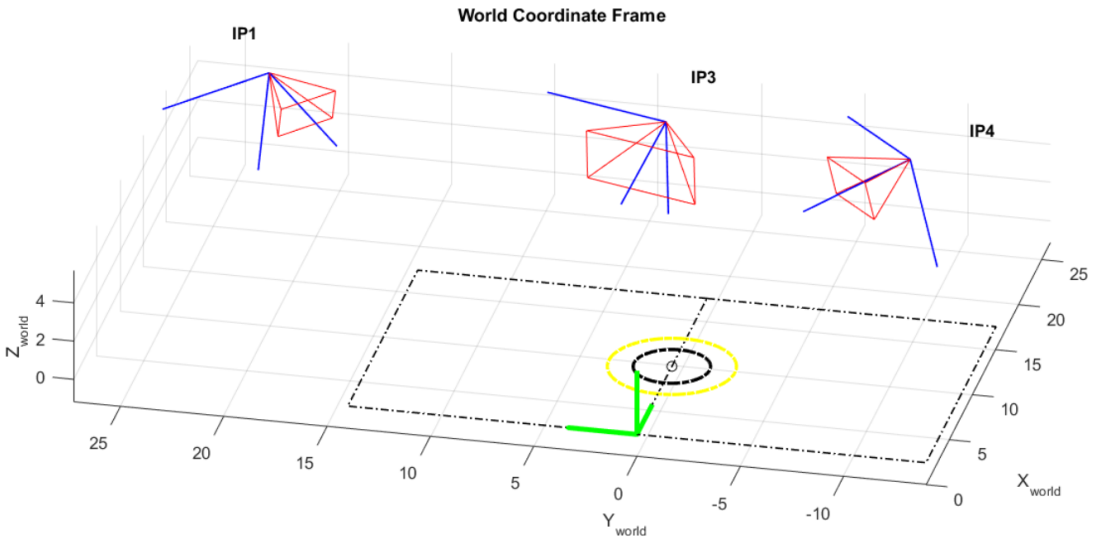
(a)

(b)

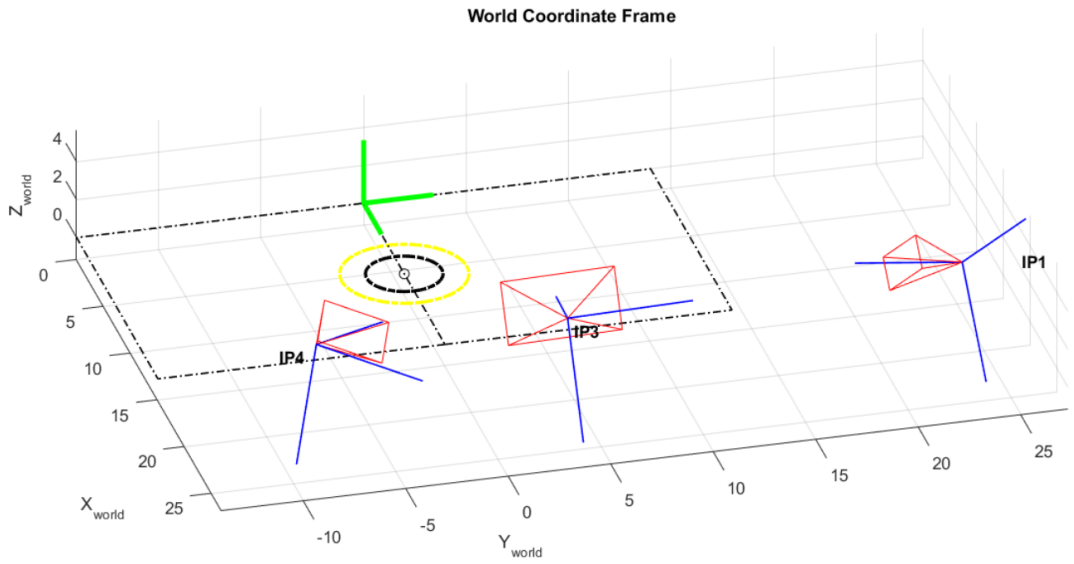
(c)

Figure 4.6: One view from camera IP3. (a) Iterative PnP solution for 1st calibration procedure. (b) Iterative PnP solution for 2nd calibration procedure, with estimated values for focal length deeming erratic projections of the 3D world points from the markers. (c) Iterative PnP solution for 3rd calibration procedure.

calibration presented very similar results. Since no major differences could be detected during this phase in terms of error on the estimation of the camera pose, both calibrations were considered for the remaining experiments. To understand the rotation and translation values of each camera, Figure 4.7 presents a 3D mathematical representation of the obtained poses for cameras IP1, IP4 and IP3 (3rd calibration procedure).



(a)



(b)

Figure 4.7: 3D graphic of the pose of the cameras using the estimated values for the extrinsic parameters. (a) Front perspective of the cameras' position and orientation. (b) Back perspective of the cameras' position and orientation. Blue lines represent cameras coordinate axis, red lines represent their image screen resolution, the black dashed line represents the basketball field and the green line represents the origin of the world coordinate frame. The graphic axes units are in meters.

4.3 Triangulation

At this point, different estimations were examined to reach a possible solution of determining the ball in a defined 3D space given its projection onto two camera images. Initially, the ball was considered as static to provide ground-truth data. By changing the ball from different positions on planes $z = (0,1,2,3)$m in the world coordinate system, a perspective depth of the designated 3D field was scrutinized. Also, a variation on the ball height would allow to evaluate if there would exist an error fluctuation on each dimension. The ball projection point onto each camera image was manually extracted. The 2D coordinate on the image frame was delineated as the center of the ball. Let the perimeter of the ball be $P = 2\pi r$, with r as the ray of the ball. As $P = 75$cm, its ray $r = 12$cm $= 0.12$m and the value defined as the static ball height, $z = (0,1,2,3) + r = (0,1,2,3) + 0.12$ m.

Three cameras were considered for the triangulation problem: IP1, IP4 and IP3 with two different calibrations, as mentioned in Section 4.2. Knowing that, five pairs of cameras were selected as testing solutions: IP1-IP4, IP1-IP3 with the 1st IP3 calibration procedure, IP1-IP3 with the 3rd IP3 calibration procedure, IP4-IP3 with the 1st IP3 calibration procedure, and IP4-IP3 with the 3rd IP3 calibration procedure.

4.3.1 Static Ball

4.3.1.1 Linear-LS Method

To avoid a considerable number of testing, a decision regarding the correct estimated calibration parameters was made. To that end, a first triangulation experiment using a Linear-LS triangulation method, described in Section 3.3.2, was performed using these five combination of camera pairs with different PNP solutions obtained for camera IP3 (Iterative PnP and RANSAC PnP). Values of the average 3D Euclidean distance for each estimated position of the static ball and its standard deviation are in Tables 4.8, 4.9, 4.10, 4.11 and 4.12. The 3D Euclidean distance is calculated considering the difference between the real 3D world point and the estimated 3D world point:

$$d(X,X_e) = \sqrt{(X_x - X_{ex})^2 + (X_y - X_{ey})^2 + (X_z - X_{ez})^2} \tag{4.2}$$

with X_e as the estimated 3D object world coordinates.

A closer look at the obtained results indicates that the 3rd calibration procedure of camera IP3 delivers higher accuracy on the estimated 3D point instead of the 1st one. For example, for $z = (1,2)$m and considering the estimated camera pose from the iterative PnP method, the 1st calibration procedure had an erratic behaviour. Exceptionally, the 3rd one presented slightly worst estimations on $z = 3$m. The reason supporting that occurrence is the number of positions evaluated for that height for all cameras, since only five positions were tested during image acquisition. As the number of positions was low, the average Euclidean distance does not converge to a value that could be considered an approximation of the error to the real 3D space location. Considering that, the 1st calibration procedure obtained worst calibration parameters for a generality of 3D points from the field, as it can be stated on estimations for $z = 0$m. It was assumed for the remainder of

Table 4.8: Average, Maximum and Minimum errors for the Linear-LS method performance on camera pair IP1-IP3, considering IP3 1st calibration procedure and the Iterative PnP and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	5.8 (1)	66.9 (11)	26.9	18.9	cm
z = 1m	61.6 (1)	1713 (12)	655.4	483.2	cm
z = 2m	44.1 (2)	1692.4 (12)	636.4	478	cm
z = 3m	28 (1)	166.1 (3)	82.4	53.5	cm
All points	5.8 (1)	1713 (12)	391.1	472.9	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	5.3 (8)	55.6 (12)	25.4	17.4	cm
z = 1m	17.9 (1)	1495.2 (12)	624.8	414.4	cm
z = 2m	77.8 (1)	1467.3 (12)	607.4	405.2	cm
z = 3m	31.2 (5)	211 (12)	109.7	67.2	cm
All points	5.3 (8)	1495.2 (12)	376.2	421.4	cm

Table 4.9: Average, Maximum and Minimum errors for the Linear-LS method performance on camera pair IP4-IP3, considering IP3 1st calibration procedure and the Iterative PnP and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	5.5 (1)	113.9 (12)	30.7	27.4	cm
z = 1m	63.3 (1)	821.8 (12)	399.2	220.6	cm
z = 2m	49.9 (1)	843.3 (12)	380.2	229.6	cm
z = 3m	18.7 (1)	140.8 (12)	63.9	48.6	cm
All points	5.5 (1)	843.3 (12)	242	244.8	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	3.8 (4)	58.3 (3)	15.4	16.2	cm
z = 1m	82.5 (1)	843.5 (3)	454.8	246	cm
z = 2m	75.2 (2)	834.9 (12)	414.9	218.7	cm
z = 3m	17.4 (12)	52.9 (5)	40.4	14.4	cm
All points	3.8 (4)	843.5 (3)	260.8	270.9	cm

the experiments, that the most accurate calibration procedure for camera IP3, and the one used for other triangulation testings, was the 3rd calibration procedure.

Results obtained for camera pair IP1-IP4 had better approximations to the real value of the 3D location of the world object. The extrinsic parameters used in this experiment were the ones obtained on the Iterative PnP solution, since it demonstrated to have better results. The testing of this camera pair was performed to check the validity of the triangulation method, and a graphic representation of all the estimations of the ball's 3D position was produced and is illustrated in Figure 4.8. However, the second Linear-LS method tested, described in Section 3.3.2, granted

Table 4.10: Average, Maximum and Minimum errors for the Linear-LS method performance on camera pair IP1-IP3, considering IP3 3rd calibration procedure and the Iterative PnP and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	6.3 (1)	65.9 (11)	27.3	19.9	cm
z = 1m	4.5 (1)	64.6 (11)	31.7	16.9	cm
z = 2m	7.7 (1)	88.1 (11)	47.6	27.6	cm
z = 3m	14.3 (1)	167.9 (3)	86.6	55.1	cm
All points	6.3 (1)	167.9 (3)	40.7	32.6	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	4.9 (8)	56.5 (12)	22.9	14.5	cm
z = 1m	10.8 (1)	95.9 (12)	34.6	23.2	cm
z = 2m	17.4 (2)	139.8 (12)	51.3	36.6	cm
z = 3m	52.4 (1)	249.1 (12)	129.6	78.4	cm
All points	4.9 (8)	249.1 (12)	45.8	46.9	cm

Table 4.11: Average, Maximum and Minimum errors for the Linear-LS method performance on camera pair IP4-IP3, considering IP3 3rd calibration procedure and the Iterative PnP and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	6.3 (9)	118.3 (12)	31	29.9	cm
z = 1m	3.9 (3)	126.8 (12)	37.4	34.3	cm
z = 2m	8.9 (3)	128.4 (12)	47.6	37.7	cm
z = 3m	15.1 (3)	126.2 (12)	65.6	47.7	cm
All points	6.3 (9)	128.4 (12)	41.3	37.2	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	5.3 (8)	55.6 (12)	25.4	17.4	cm
z = 1m	8.8 (1)	59.7 (11)	32.2	19.9	cm
z = 2m	14.9 (12)	94.3 (7)	49.4	31.6	cm
z = 3m	12.2 (3)	94.9 (5)	50.4	30	cm
All points	5.3 (8)	94.9 (5)	34.9	26.4	cm

better results (Table 4.13) and was considered the best implementation of the Linear-LS method. A graphic 3D representation was also produced to allow a good visual comparison between both Linear-LS methods, as shown in Figure 4.9.

The disparity on estimation values where camera IP3 is used can be explained due to the divergent PnP solutions obtained previously, producing some well-triangulated positions instead of others for the same pair of cameras. As explained on the preceding subsection, one camera pose for IP3 combined with the "stable" camera pose from IP1 or IP4 achieve better triangulation values for a specific subset of points but can also generate worst values for other subset. Yet, other

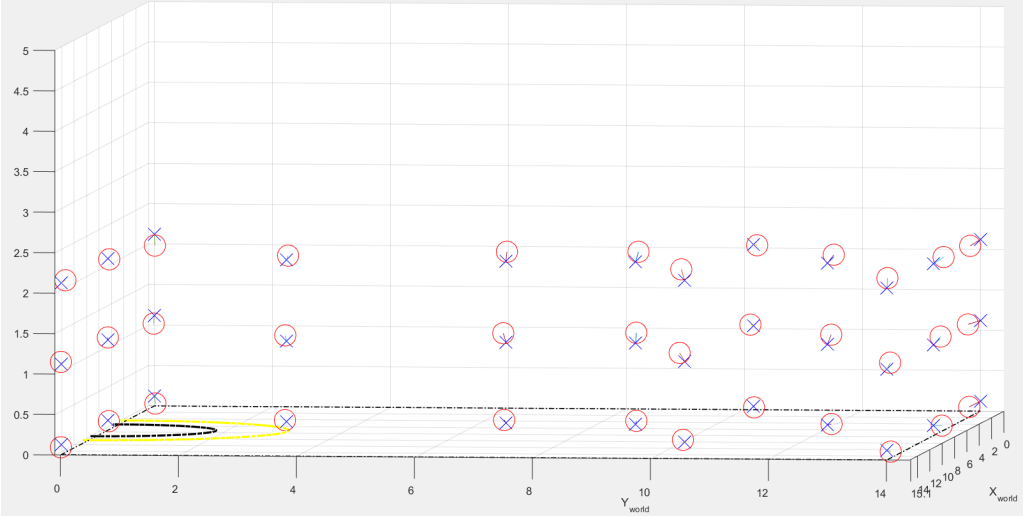


Figure 4.8: Graphic representation of the estimated ball 3D location for 36 different points on the defined 12 positions on the basketball field (heights of 0m, 1m and 2m). The estimation was performed with the 1st Linear-LS method on the camera pair IP1-IP4, considering the Iterative PnP solution for both cameras. Red dots represent the estimated ball, and blue crosses represent the real location of the ball's center.

Table 4.12: Average, Maximum and Minimum errors for the Linear-LS method performance on camera pair IP1-IP4, considering the Iterative PnP solution for both cameras.

	Minimum d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	1.8 (8)	34.1 (12)	12.6	9.4	cm
z = 1m	2.8 (1)	37.5 (12)	15.9	11.2	cm
z = 2m	4.8 (2)	57.9 (12)	21.2	13.7	cm
z = 3m	10.3 (1)	76.7 (12)	30.5	27.4	cm
All points	1.8 (8)	76.7 (12)	17.4	14.5	cm

Table 4.13: Average, Maximum and Minimum errors for the 2nd implementation of the Linear-LS method performance on camera pair IP1-IP4, considering the Iterative PnP solution for both cameras.

	Minimum d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	1.5 (4)	25.3 (12)	8.6	6.4	cm
z = 1m	1.7 (3)	26.1 (12)	11.9	7.1	cm
z = 2m	3.2 (2)	29.2 (12)	12.5	7.8	cm
z = 3m	5.9 (3)	32.2 (12)	13.2	11	cm
All points	1.5 (4)	32.2 (12)	11.14	7.83	cm

camera pose for IP3 matched with the same camera pose IP1 or IP4 achieve worst triangulation values for a subset that formerly delivered good results and vice versa. Since no conclusion could be attained respecting a best solution for camera IP3 pose estimation from different PnP solutions,

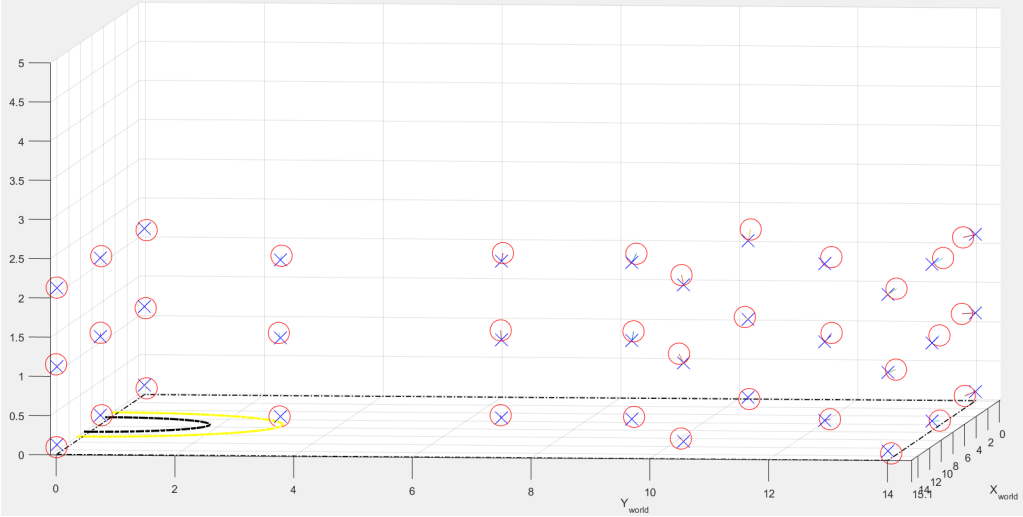


Figure 4.9: Graphic representation of the estimated ball 3D location for 36 different points on the defined 12 positions on the basketball field (heights of 0m, 1m and 2m). The estimation was performed with the 2nd Linear-LS method on the camera pair IP1-IP4, considering the Iterative PnP solution for both cameras. Red dots represent the estimated ball, and blue crosses represent the real location of the ball's center.

those were still examined on the following 3D reconstruction methods.

4.3.1.2 Midpoint Method

The implemented Midpoint method, described in Section 3.3.1, was tested and its results were compared with the values obtained with the Linear-LS method for the pair IP1-IP4, since they were producing the best results. As seen on Table 4.14, values for the average Euclidean distance of the estimated 3D point and its standard deviation were also computed. A graphic 3D representation was also produced to verify the average error of each estimated ball position for camera pair IP1-IP4, as depicted in Figure 4.10.

Table 4.14: Average, Maximum and Minimum errors for the Midpoint method performance on camera pair IP1-IP4, considering the Iterative PnP solution for both cameras.

	Minimum d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	1.5 (4)	25.2 (12)	8.6	6.5	cm
z = 1m	1.8 (3)	25.8 (12)	11.8	7	cm
z = 2m	3.2 (2)	28.8 (12)	12.5	7.7	cm
z = 3m	4.6 (3)	31.5 (12)	12.7	10.9	cm
All points	1.5 (4)	31.5 (12)	11.07	7.74	cm

The Midpoint method presented good estimations for the 3D position of the ball despite being considered an easy and simple geometric method compared to the Linear-Ls method that behaves poorly for projective or affine transformations [104]. However, its accuracy was slightly better

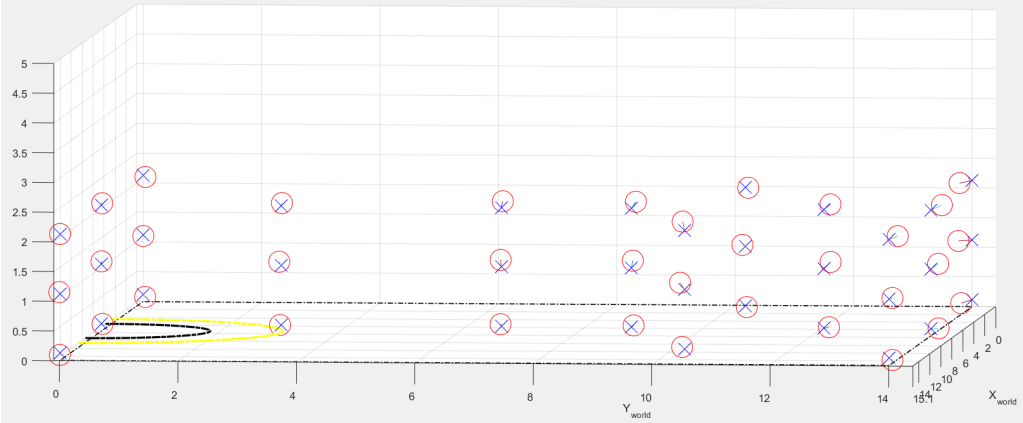


Figure 4.10: Graphic representation of the estimated ball 3D location for 36 different points on the defined 12 positions on the basketball field (heights of 0m, 1m and 2m). The estimation was performed with the Midpoint method on the camera pair IP1-IP4, considering the Midpoint PnP solution for both cameras. Red dots represent the estimated ball, and blue crosses represent the real location of the ball's center.

than the Linear-LS method which indicates that the perpendicular solution performs sufficiently well for perspective projections knowing an approximate calibration of the cameras.

Uncertainties concerning camera IP3 led to exploit the Midpoint method as a deciding factor of which PnP solution could be the best approximation for the IP3 camera pose. An error evaluation using the Midpoint solution for this camera is shown in Tables 4.15 and 4.16.

Table 4.15: Average, Maximum and Minimum errors for the Midpoint method performance on camera pair IP1-IP3, considering IP3 3rd calibration procedure and the Iterative and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	6.9 (8)	29.1 (12)	16.4	10.2	cm
z = 1m	7.2 (1)	29.9(12)	20.7	12.4	cm
z = 2m	5.2 (8)	28 (12)	21.6	11.4	cm
z = 3m	5.4 (3)	30.8 (12)	20.8	9.62	cm
All points	5.2 (8)	30.8 (12)	19.6	11.4	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	1.9 (4)	33.9 (7)	15.9	9.8	cm
z = 1m	8.5 (8)	42.3 (1)	22.1	10.6	cm
z = 2m	10.4 (4)	41 (1)	23	10.2	cm
z = 3m	5.7 (3)	57.9 (1)	28.8	19.8	cm
All points	1.9 (4)	57.9 (1)	21.1	12.3	cm

Observing the average Euclidean distances of the estimated 3D locations of the ball to the real 3D point on both pairs IP1-IP3 and IP4-IP3, the RANSAC PnP solution presents better results for

Table 4.16: Average, Maximum and Minimum errors for the Midpoint method performance on camera pair IP4-IP3, considering IP3 3rd calibration procedure and the Iterative and RANSAC PnP solutions.

Iterative PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	4.6 (11)	69.5 (7)	24.1	20.2	cm
z = 1m	8.3 (3)	72.3 (7)	26.6	19.6	cm
z = 2m	5.7 (8)	78.7 (7)	30.2	20.9	cm
z = 3m	16.4 (3)	50.4 (12)	32.9	12.9	cm
All points	4.6 (11)	78.7 (7)	27.5	19.8	cm
RANSAC PnP	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	2.9 (2)	55.6 (7)	17.4	14.5	cm
z = 1m	8.7 (8)	57.1 (7)	22.5	13.9	cm
z = 2m	10.8 (4)	62.3 (7)	25.2	14.8	cm
z = 3m	14.8 (3)	41.4 (1)	23.6	10.5	cm
All points	2.9 (2)	62.3 (7)	21.8	14.4	cm

both combinations, while the iterative PnP solution has a similar result for pair IP1-IP3 but worst results for pair IP3-IP4. Therefore, the extrinsic parameters considered as the best approximation for the camera IP3 pose was the RANSAC PnP estimations and used for the iterative triangulation method.

4.3.1.3 Iterative Linear-LS Method

For the Iterative Linear-Ls method, two Linear-LS methods were previously compared as possible approaches to be used iteratively. The 2nd Linear-LS method proved to have a better behaviour than the 1st Linear-LS method. Thus, the 2nd implementation was used to obtain results through several iterations with error minimization procedures. The average error for the Euclidean distance estimation for each height of the ball for the possible camera pair combinations and using the RANSAC PnP solution for camera IP3 are depicted in table 4.17, and are graphically visible in Figure 4.11.

Concerning camera IP3 pose estimation, it was confirmed that this camera has inaccurate pose estimations and both pairs IP1-IP3 and IP4-IP3 could not reach great values, compared to camera pair IP1-IP4. Hence, camera IP3 was not considered for the next stage.

Focusing solely on camera pair IP1-IP4, the Iterative Linear-LS method had a slight increase in the average Euclidean distance error compared to the non-iterative Linear-LS method and to the Midpoint method. Generally, it was verified that the Midpoint method performed slightly better than all other 3D reconstruction techniques by reaching an average error of 11.07 cm. Nonetheless, both the iterative Linear-LS and non-iterative Linear-LS produce great results with 11.7 cm and 11.14 cm of average error, respectively.

Table 4.17: Average, Maximum and Minimum errors for the Iterative Linear-LS method performance on camera pairs IP1-IP3 and IP3-IP4, considering IP3 3rd calibration procedure and RANSAC PnP solution, and the Iterative PnP solution for cameras IP1 and IP4.

IP1-IP3	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	2.3 (4)	35.2 (1)	16.4	10.4	cm
z = 1m	8.3 (8)	42.5 (1)	22.2	10.7	cm
z = 2m	11.2 (8)	41.2 (1)	23.4	9.9	cm
z = 3m	2.9 (3)	61.9 (1)	29.9	22.1	cm
All points	2.3 (4)	61.9 (1)	21.5	12.8	cm
IP4-IP3	**Minimum** d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	2.9 (2)	56.3 (7)	17.7	14.8	cm
z = 1m	8.8 (2)	57.1 (7)	22.8	13.9	cm
z = 2m	11.6 (4)	62.9 (7)	25.8	14.8	cm
z = 3m	15.4 (3)	42.2 (1)	24.4	10.5	cm
All points	2.9 (2)	62.9 (7)	22.2	14.5	cm

Table 4.18: Average, Maximum and Minimum errors for the Iterative Linear-LS method performance on camera pair IP1-IP4, considering the Iterative PnP solution for both cameras.

	Minimum d (Position)	**Maximum** d (Position)	**Average** d	**Std. Deviation** σ	**Unit**
z = 0m	1.5 (4)	25.7 (12)	8.9	6.6	cm
z = 1m	0.6 (3)	26.4 (12)	12.4	7.4	cm
z = 2m	4.7 (2)	29.7 (12)	13.2	8.1	cm
z = 3m	5.5 (3)	33.1 (12)	14.3	10.9	cm
All points	1.5 (4)	33.1 (12)	11.7	8.1	cm

4.3.2 Ball in Motion

After evaluating the static ball sequences, testings for the ball in motion sequences were performed. As mentioned before, in Chapter 3, the ball was captured in three different motion scenarios. The first sequence consisted on movement along the field's plane $z = 0$m with starting point on the upper right corner of the basketball court. The second sequence resided on arbitrary dribbling of the ball, with up and down movements. On the third sequence, the ball was thrown against the court's ground with some initial velocity and left completely on its own trajectory throughout the rest of the movement, solely depending on the gravity force from that moment on.

To be able to triangulate a 3D point in Euclidean space it is necessary that two cameras could relate the projection of the same point on their image planes. The static ball sequences prevented this issue to be a problem once the ball was stable and the 3D point coincided in each camera, independently of any video synchronization. However, on sequences with ball motion it was mandatory some type of synchrony between cameras to allow accuracy on calculations of the triangulation methods. That challenge was surpassed with an electronic flash device. When the

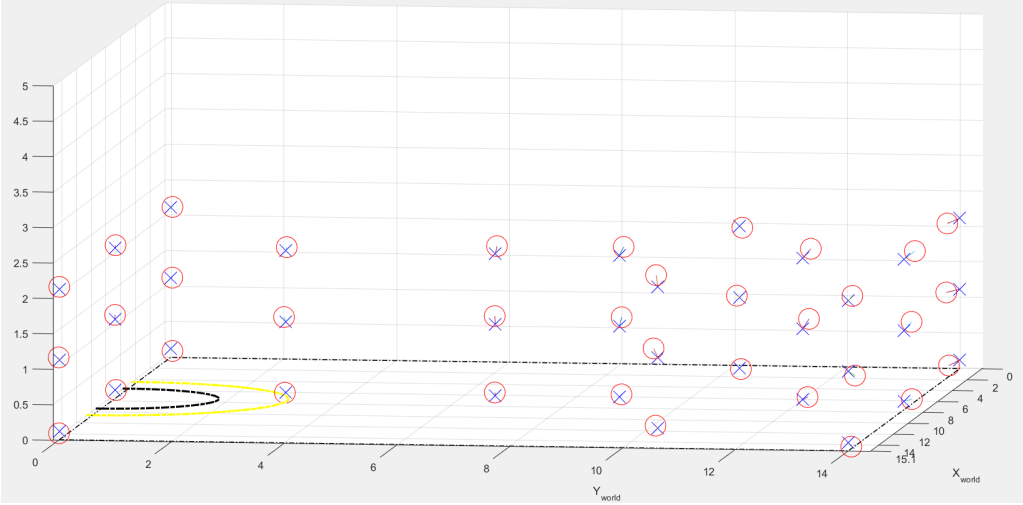


Figure 4.11: Graphic representation of the estimated ball 3D location for 36 different points on the defined 12 positions on the basketball field (heights of 0m, 1m and 2m). The estimation was performed with the Iterative Linear-LS method on the camera pair IP1-IP4, considering the Iterative PnP solution for both cameras. Red dots represent the estimated ball, and blue crosses represent the real location of the ball's center.

electronic flash was triggered, it achieved its peak intensity rapidly and then faded away at a slow rate. The intensity rise was far quickly than the slow decline, and the period of time the flash is at least 1/10 of its maximum intensity could be used to quantify its duration. Assuming that the electronic flash device has a typical duration of 3 to 4ms of at least 1/10 of maximum intensity and that videos had an approximate frame rate of 30 fps (3.3ms per frame), the flash was visible on one frame for each camera in case it was captured. Although frames from different cameras could not detect the flash at the same intensity value, this was still considered a good synchronization procedure because cameras would never have a time synchronization difference of more than 33.3ms. For each video the starting frame was determined as the following frame of the one where a flash was visible.

An example of the flash captured by each camera is presented on Figures 4.12a, 4.12b and 4.12c. Both IP1 and IP4 cameras can detect the flash with high intensity on one frame, while camera IP3 captures the moment when the flash intensity is rather low. Nonetheless, this frame can still be considered as the synchronization frame because it was the one that presented a smaller time distance to the synchronization frames of other cameras.

Concerning all sequences, only camera IP1-IP4 was considered for the testing experiments. Figure 4.14 shows six continuous frames of the ball's path on the ground plane and Figure 4.13 depicts a 3D graphic description of the estimated 3D position of the ball. Establishing a comparison between Figures 4.13 and 4.14, it is visible that the camera pair IP1-IP4 produces a great estimation of the trajectory of the ball.

On the second sequence, a flash was not detected for camera IP4. Testing this sequence was important to estimate the 3D position of the ball considering human intervention on ball's trajec-

(a)

(b)

(c)

Figure 4.12: Frames that successfully captured the bright light produced from the electronic flash device and allow to synchronize all three cameras. (a) Flash captured on camera IP1. (b) Flash captured on camera IP3, slightly visible on the ceiling from the pavilion. (c) Flash captured on camera IP4.

tory. In spite of that, it could be more useful to examine ball's trajectory on a free movement sequence where gravity does most of the work. Since the third sequence presents the previously detailed example, the second sequence was removed from testing experiments.

Regarding the third ball in motion sequence, a good example of an almost imperceptible flash was found and happened on camera IP1. As shown in Figures 4.15a and 4.15b, the flash was

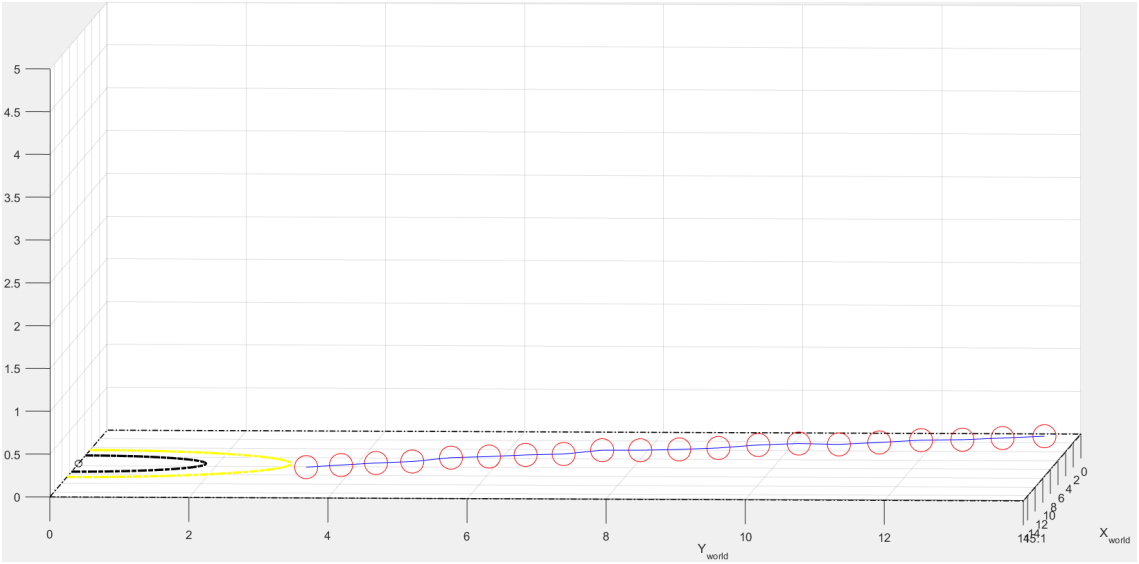
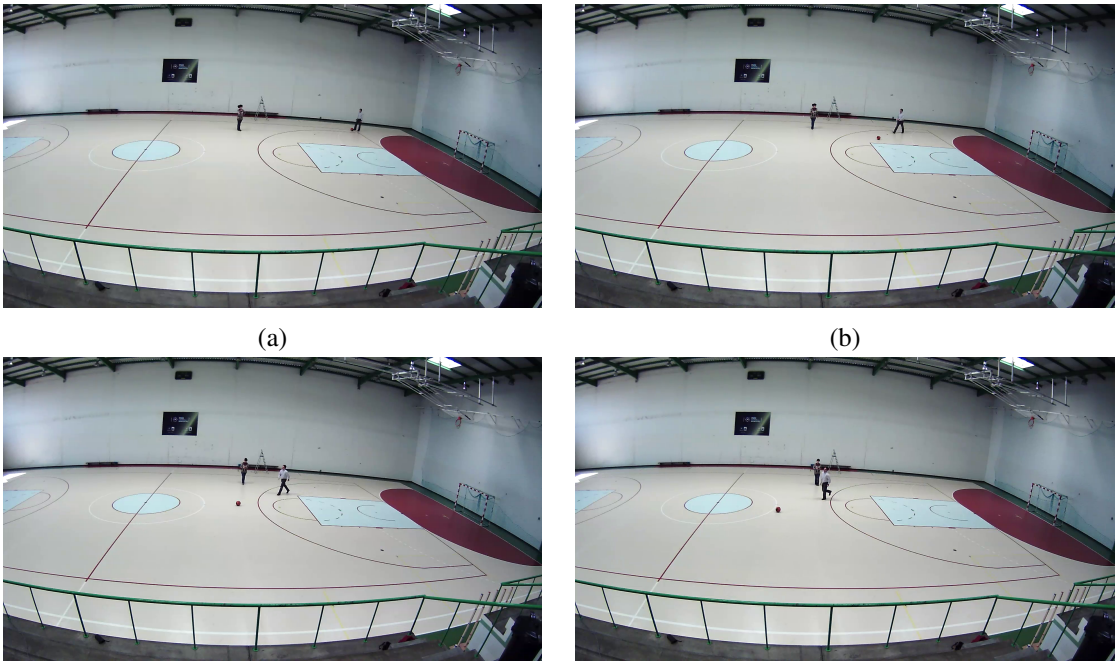


Figure 4.13: Graphic representation of the estimated ball 3D location for the first ball in motion sequence. The estimation was performed with the Midpoint method on the camera pair IP1-IP4, considering the Iterative PnP solution for both cameras. Red dots represent the estimated ball on each 20 frames. The trajectory was defined as starting in the upper right corner of half of the field and travelling on the ground to meet the midcourt line.

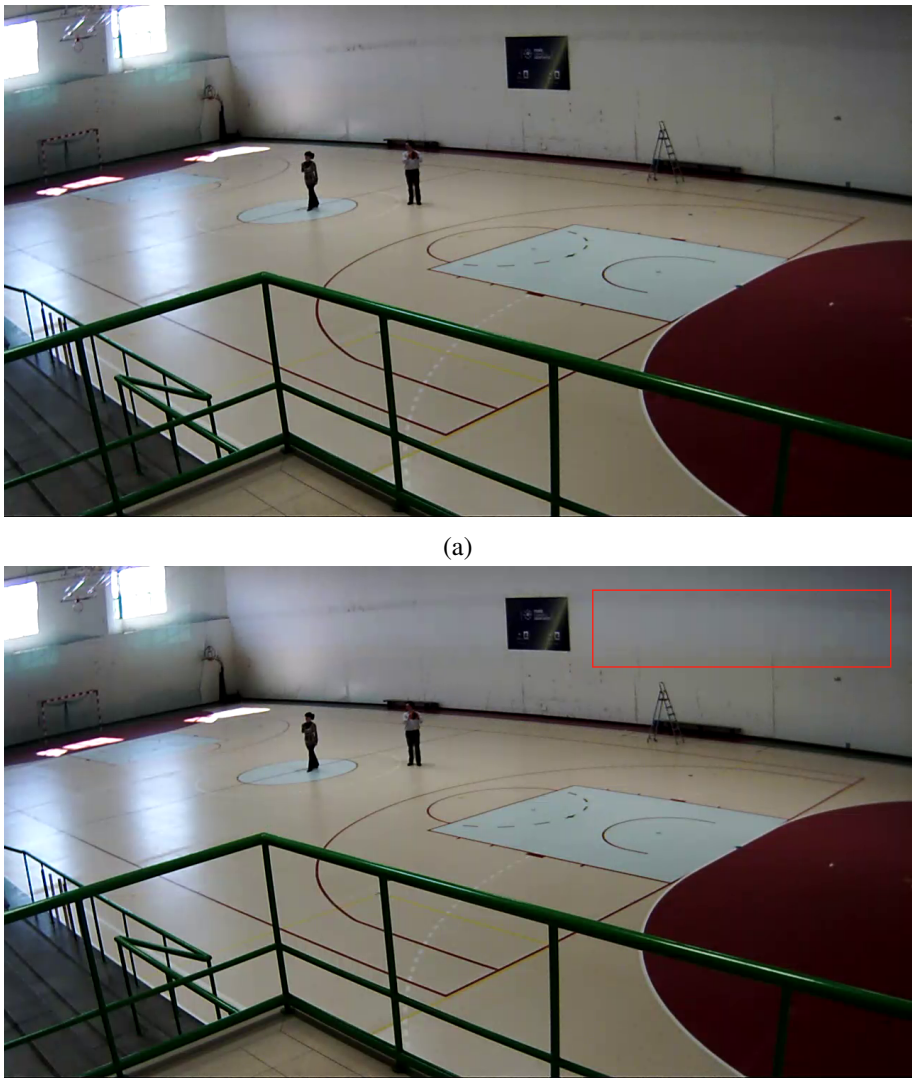


(a) (b)

(c) (d)

Figure 4.14: Four scenes of the same view from camera IP3 on the first ball in motion sequence. The real trajectory of the ball happened solely on the ground, with starting point on the upper right corner of the field and ending point on the midcourt line.

slightly detected on the top of the image and was only noticeable due to the skip between frames on the continuous video since the camera captured the flash when it had a very low light intensity.



(a)

(b)

Figure 4.15: Video capture from camera IP1 of consecutive frames where the electronic flash device is triggered (a) Preceding frame, no light is captured. (b) Following frame, end tail of the flash is captured on the top of the image (red box) and is used as the synchronization frame.

After synchronizing both cameras, an estimation of the ball's trajectory was produced and is illustrated on Figure 4.16. The initial trajectory of the ball was defined as starting on a person's hand and sent against the ground plane with some velocity. This allowed the ball to start a free movement that only depended on the gravity and producing a parabolic trajectory, as depicted in Figure 4.17. Comparing Figures 4.16 and 4.17, the trajectory estimation produced by camera IP1-IP4 had a behaviour that changed continuously with the ball's trajectory on the field. At first, the 3D location of the ball is correctly produced and the defined trajectory is well calculated. That can

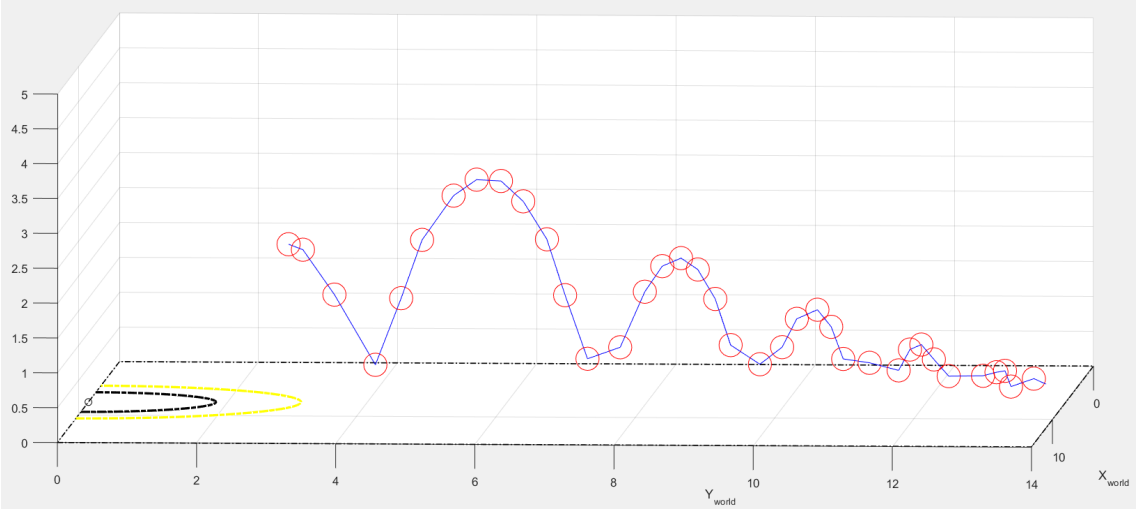
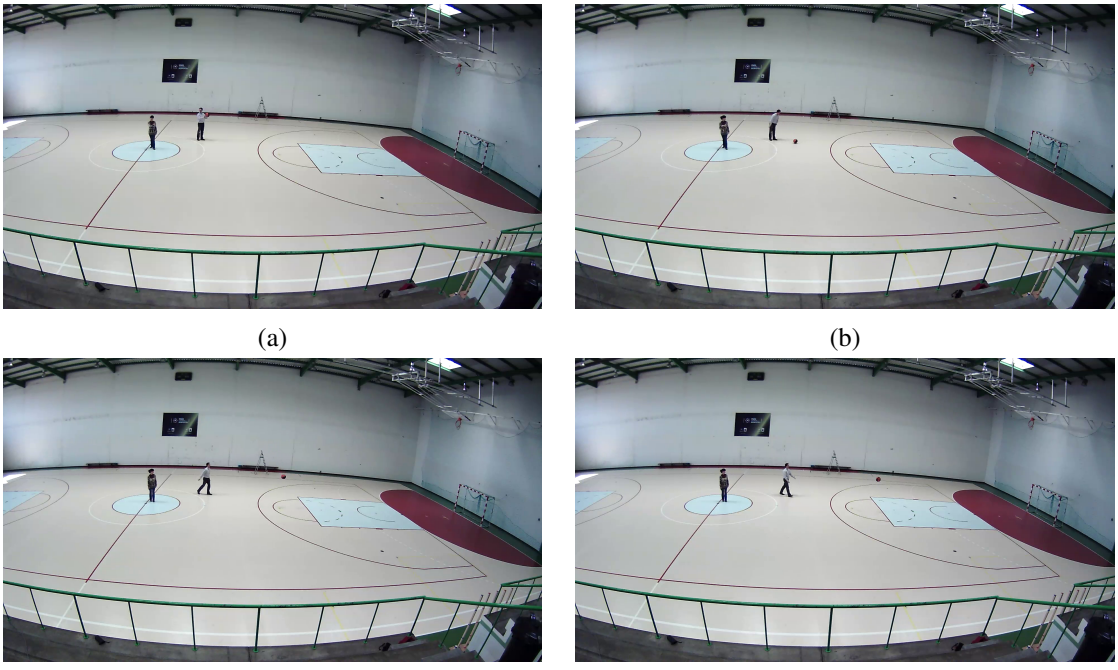


Figure 4.16: Graphic representation of the estimated ball 3D location for the third ball in motion sequence. The estimation was performed with the Midpoint method on the camera pair IP1-IP4, considering the Iterative PnP solution for both cameras. Red dots represent the estimated ball on each 5 frames. The ball started its motion by being thrown with some velocity into the ground.



(a) (b)

(c) (d)

Figure 4.17: Four scenes of the same view from camera IP3 on the third ball in motion sequence. The real trajectory of the ball started by being thrown against the ground and described a parabola for each time it hit the ground.

be verified throughout the moment that the ball leaves the person's hands, the moment it touches the floor for the fist time and the first parabolic motion it generates. However, as soon as the ball gets closer to the camera IP1 and farther from the camera IP4, the trajectory estimation tends to de-

teriorate itself. This is another evidence that the camera IP1 was not correctly calibrated compared to camera IP4 and that resulted on worsening the estimations of the triangulation methods.

However, conclusions could not be assumed concerning the actual best location of cameras and its responsibility on the 3D position estimations for the ball due to the calibration quality of camera IP3 not being as accurate as the calibration of the other two cameras. This prevented a fair comparison between all camera pairs, allowing IP1-IP4 to outperform all.

Regardless of uncertainty on the values obtained for some of the intrinsic parameters for camera IP3, culminating on doubts on the accuracy of the extrinsic parameters estimated, results can be considered promising since the average error was around 20 to 30 cm. One of the main reasons for that distrust is the exaggerate image distortion that the IP3 camera lens introduces to its image, comparing it to the other two cameras. This could have led to erroneous image distortion removal, if the calibration parameters estimations were not precise enough.

4.4 Summary

The objective of the present dissertation was to develop a camera system capable of identifying and reconstructing the 3D position of an object in a context of a sports game. The first step was to retrieve cameras intrinsic parameters and lens distortion coefficients. This was obtained through a calibration procedure using multiple views of a planar black and white chessboard pattern. Since there were some disparities on the values obtained for one camera, a computation of the reprojection error for each calibration image followed by a recalibration procedure excluding frames above a certain error value was performed.

The next step was to estimate the extrinsic parameters from cameras, i.e., cameras pose. Different PnP solvers were studied and used to compute those parameters using six markers placed around the game field that served as sources of points from the 3D world frame. Not only that, the estimations of the camera intrinsic matrix and distortion coefficients previously obtained were essential to remove distortion from images and project the estimated 3D points on the image plane according to the estimations of the cameras locations and orientations.

The ultimate step was to reconstruct the 3D object using two cameras. For ground-truth data, the ball was captured on some defined positions around the field on a static motion. Then, through triangulation methods and testing several camera pairs combinations, the estimated 3D coordinates were compared to the real 3D ball position. Results showed that a considerably accurate estimation of the ball 3D position can be obtained if accurate estimations from the calibration procedures were retrieved.

Considering continuous movement, an electronic flash device was used to produce a bright light that allowed a synchrony between cameras, considering the flash only appeared for a single frame of the image acquirement which determined an instant $t = 0$ for each camera.

Chapter 5

Conclusions and Future Work

5.1 Final Discussion

This dissertation proposes a solution to compute the 3D data position of a ball during a sports event in a large space and using off-the-shelf cameras. At an image processing step, the solution incorporated two stages: camera calibration and 3D reconstruction to obtain 3D information of an object given its detection.

Dataset was acquired on a basketball court with static cameras placed on one side of the field, on a higher level and oriented to it. Measurements of the field were taken and a world reference frame was constructed based on them. Also, positions around the field were chosen accordingly to allow ground-truth data that were used as quantitative comparisons with the estimations obtained from the testing sequences using the ball. Initially, the ball was placed static on each position. Then, a change in height was made for the same positions to create depth on the dataset acquired. Lastly, some continuous ball movements highly usual in sports games were captured.

On recent literature, many methods propose calibration procedures that only calibrate for single cameras or for several cameras oriented to the same planar calibration object. However, since a sports field can be characterized as a large world frame and cameras need to be related to that world coordinate system, labeled markers were placed on the field to produce a rich set of 3D points of that reference world frame. For this purpose, the camera calibration process was divided in two different stages: intrinsic calibration and extrinsic calibration.

The intrinsic calibration stage required the use of a planar calibration pattern and was performed on each camera individually, producing estimations of the intrinsic parameters and the distortion coefficients. It was developed an iterative methodology, where an evaluation of the calibration process was made on each iteration through the reprojection error, excluding frames with great error. Also, it has been found important to assess the threshold value, as well as the number of frames considered for the calibration process and the variability of the calibration pattern motion. Therefore, it is advised that a prosperous set of calibration images are carefully taken before any calibration procedure can be done. Reprojection errors and image distortion removal processes were a good visual reference for the accuracy of the different estimations obtained for

the same camera. However, solely on the next stage was possible to create certainties regarding a comparison on the accuracy of estimations for the same camera.

The extrinsic calibration procedure was possible by creating 2D image correspondences of the produced set of 3D points from the labeled markers placed on the field to reach camera pose estimations. Three methods were evaluated and tested through a projection of points on camera images using the estimated poses for each camera. The computation of the extrinsic parameters proved to be directly influenced by the previously estimated intrinsic parameters because they define the image plane characteristics where the points were projected on. Thus, further evaluation concerning the accuracy of the intrinsic calibration stage was now achievable. Since it was not acquired a wide variety of 3D orientations of the pattern on a camera that had a larger lens distortion comparably to the others, it proved to be difficult to improve the results of the initial calibration stages and influenced the consequent estimations. The Iterative PnP was the method that displayed consistently lower errors on the projected points.

The final step was to obtain the 3D position of the ball through triangulation. Taking advantage on the sequences captured, the static ball granted ground-truth data while providing an equal location of the ball independently from the time instant for all cameras. Several triangulation methods were implemented and evaluated with all possible combinations of camera pairs. Camera pairs with more accurate calibration results obtained lower errors for all tested triangulation methods, with the Midpoint method presenting slightly better results than the others. Results were promising since the error measurement was on the order of 11 cm and considering a field space volume of 634 m^3 (height of 3m). Concerning the ball in motion sequences, the video synchronization step using an electronic flash device proved to be a simple method to guarantee an approximation in time instant for all cameras and allow a 3D reconstruction of the ball. However, the light produced by the device had a shorter period of time where it could be perceptible compared to the time instant between frames, resulting in some situations where the flash was not captured by a camera. This could be overcome by triggering several flashes evenly spaced in time.

Given these evaluations, the calibration procedure established itself as the most laborious step and susceptible to influence all of the following obtained estimations. It can also be considered has the foundation of the process of retrieving 3D information of the ball. To avoid inconsistencies, a carefully produced recalibration method proved to provide better results on the computation of an object location in a world reference frame. Concerning the use of the playing field as a calibration object, a marker-based extrinsic calibration proved to be useful when multiple and distinct cameras are placed wide apart from each other and a large 3D space field is used as the world frame. Finally, different variations of the triangulation method were tested, obtaining good estimations for the world 3D ball position.

5.2 Future Work

Despite the potential of the obtained results, the present work is still preliminary and a lot of developments and improvements are yet to be accomplished.

Considering the estimations obtained for the 3D object, a study of the projection errors on the 3D spatial field could be made. This would produce a process capable of improving a 3D object extraction by predicting which triangulation method or camera pair would suit better for the consequent position.

Regarding the ball motion in 3D space, some new captures could be made. This would allow for a better understanding of the process of synchrony between cameras and the effect that different frame rates could provide for situations where the ball's velocity is greater. Also, solutions could be obtained at a video processing sequence since an estimation of the 3D position of the ball would be achievable not only through triangulation processes but also using the available time information, through estimations of the probable 3D position, e.g. using Kalman filters or others, or through inference on the type of trajectory, e.g. linear, parabolic, etc.

Considering the influence of the camera calibration process, new methods to accurately estimate cameras parameters could be tested, e.g. using the field lines of the playing field, etc. Focusing on the placement of the cameras, a higher number of cameras could be used to assess the influence that the location and orientation of a camera has on extracting 3D information. For this purpose, different camera poses could be tested that would create a wider range of camera pair combinations and enable a comparison between several possibilities.

As triangulation methods is concerned, some different techniques should be tested as to evaluate if more accurate values could be obtained. Not only that, if a higher number of cameras were to be used, the possibility of integrating more cameras to triangulate the 3D object would also be a considerable improvement. This would help to determine if the number of cameras used to reconstruct a 3D object would increase the accuracy on the estimation computed or if the initial calibration step could be relaxed or simplified with the extra image information.

Finally, considering that the scenario tested was simplified and only a ball was used on the sequences produced, a real scenario could be applied to the experiments. In a legitimate game, it would be necessary to introduce automatic tracking techniques to detect the ball and interacting them with estimations of the ball's 3D position, considering cases of its occlusion (players and obstacles) or erratic detections (similar colors and shapes of other elements on the game and on the field).

References

[1] R. Hamid, R.K. Kumar, M. Grundmann, Kihwan Kim, I. Essa, and J. Hodgins. Player localization using multiple static cameras for sports visualization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 731–738, June 2010.

[2] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

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

[4] Y. Ohno, J. Miura, and Y. Shirai. Tracking players and estimation of the 3d position of a ball in soccer games. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 145–148. IEEE, 2000.

[5] T. Moeslund, G. Thomas, and A. Hilton. *Computer Vision in Sports*. Springer International Publishing, Switzerland, 2014.

[6] A. Aksay, V. Kitanovski, K. Vaiapury, E. Onasoglou, J. Agapito, P. Daras, and E. Izquierdo. Robust 3d tracking in tennis videos. *Engage Summer School*, 2010.

[7] ACM Multimedia Challenge 2011. Technicolor challenge. Available in http://www.utdallas.edu/~zxz082020/content-technicolor-challenge.html, accessed in 15th January of 2016.

[8] M. Villa-Uriol, G. Chaudhary, F. Kuester, T.C. Hutchinson, and N. Bagherzadeh. Extracting 3d from 2d: selection basis for camera calibration. *IASTED Comp. Graph. & Imag*, pages 315–321, 2004.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2nd edition, 2004.

[10] L. Shen, Q. Liu, L. Li, and H. Yue. 3d reconstruction of ball trajectory from a single camera in the ball game. In *Proceedings of the 10th International Symposium on Computer Science in Sports (ISCSS)*, pages 33–39. Springer, 2016.

[11] H. Fathi and I. Brilakis. Multistep explicit stereo camera calibration approach to improve euclidean accuracy of large-scale 3d reconstruction. *Journal of Computing in Civil Engineering*, page 04014120, 2014.

[12] H. Rastgar. *Robust Self-Calibration and Fundamental Matrix Estimation in 3D Computer Vision*. PhD thesis, University of Ottawa, 2013.

[13] F. Chen, X. Chen, X. Xie, X. Feng, and L. Yang. Full-field 3d measurement using multi-camera digital image correlation system. *Optics and Lasers in Engineering*, 51(9):1044–1052, 2013.

[14] B. Cyganek and J. P. Siebert. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2009.

[15] Wikipedia. Pinhole camera. Available in https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/Pinhole-camera.svg/2000px-Pinhole-camera.svg.png, accessed in 05th January of 2016.

[16] G. Medioni and S.B. Kang. *Emerging Topics in Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[17] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[18] A. Kumar and N. Ahuja. Generalized radial alignment constraint for camera calibration. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 184–189. IEEE, 2014.

[19] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[20] P. Sturm, S.r Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(1–2):1–183, 2011.

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

[22] L. Song, W. Wu, J. Guo, and X. Li. Survey on camera calibration technique. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on*, volume 2, pages 389–392. IEEE, 2013.

[23] F. Remondino and C. Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):266–272, 2006.

[24] Wei S. and J.R. Cooperstock. Requirements for camera calibration: Must accuracy come with a high price? In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 356–361, Jan 2005. doi:10.1109/ACVMOT.2005.102.

[25] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2015.

[26] O. Grau, M. Prior-Jones, and G. Thomas. 3d modelling and rendering of studio and sport scenes for tv applications. In *Proc. of WIAMIS*, volume 2, 2005.

[27] Z. Zhang. Camera calibration with one-dimensional objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):892–899, 2004.

[28] FC Wu, ZY Hu, and HJ Zhu. Camera calibration with moving one-dimensional objects. *Pattern Recognition*, 38(5):755–765, 2005.

[29] L. Wang, FC Wu, and ZY Hu. Multi-camera calibration with one-dimensional object under general motions. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE, 2007.

[30] L. Wang, F. Duan, and K. Lu. An adaptively weighted algorithm for camera calibration with 1d objects. *Neurocomputing*, 149:1552–1559, 2015.

[31] G. Adorni, M. Mordonini, S. Cagnoni, and A. Sgorbissa. Omnidirectional stereo systems for robot navigation. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 7, pages 79–79. IEEE, 2003.

[32] Y. Ohta, I. Kitahara, Y. Kameda, H. Ishikawa, and T. Koyama. Live 3d video in soccer stadium. *International Journal of Computer Vision*, 75(1):173–187, 2007.

[33] J. Civera, A. J. Davison, and J. M. M. Montiel. *Structure from motion using the extended Kalman filter*, volume 75. Springer Science & Business Media, 2011.

[34] G. Wei and S. De Ma. Implicit and explicit camera calibration: Theory and experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):469–480, 1994.

[35] J. Heikkila and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

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

[37] J. Han, D. Farin, et al. Generic 3-d modeling for content analysis of court-net sports sequences. In *Advances in Multimedia Modeling*, pages 279–288. Springer, 2007.

[38] J. Han, D. Farin, et al. Broadcast court-net sports video analysis using fast 3-d camera modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1628–1638, 2008.

[39] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[40] G. Thomas. Sports tv applications of computer vision. In T. Moeslund, A. Hilton, V. Krüger, and L. Sigal, editors, *Visual Analysis of Humans*, pages 563–579. Springer, 2011.

[41] F. Szenberg, P. Carvalho, and M. Gattass. Automatic camera calibration for image sequences of a football match. In *Advances in Pattern Recognition—ICAPR 2001*, pages 303–312. Springer, 2001.

[42] Q. Li and Y. Luo. Automatic camera calibration for images of soccer match. In *International Conference on Computational Intelligence*, pages 482–485, 2004.

[43] X. Ying and H. Zha. Camera calibration from a circle and a coplanar point at infinity with applications to sports scenes analyses. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 220–225. IEEE, 2007.

[44] D. Farin, S. Krabbe, W. Effelsberg, et al. Robust camera calibration for sport videos using court models. In *Electronic Imaging 2004*, pages 80–91. International Society for Optics and Photonics, 2003.

[45] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[46] D. Farin, J. Han, and P. de With. Fast camera calibration for the analysis of sport sequences. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[47] S. Gedikli, J. Bandouch, N. von Hoyningen-Huene, B. Kirchlechner, and M. Beetz. An adaptive vision system for tracking soccer players from variable camera settings. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS)*, 2007.

[48] G. Thomas. Real-time camera tracking using sports pitch markings. *Journal of Real-Time Image Processing*, 2(2-3):117–132, 2007.

[49] A. Hilton, J.-Y. Guillemaut, J. Kilner, O. Grau, and G. Thomas. 3d-tv production from conventional cameras for sports broadcast. *Broadcasting, IEEE Transactions on*, 57(2):462–476, 2011.

[50] R. Dawes, J. Chandaria, and G. Thomas. Image-based camera tracking for athletics. In *Broadband Multimedia Systems and Broadcasting, 2009. BMSB'09. IEEE International Symposium on*, pages 1–6. IEEE, 2009.

[51] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[52] H. Sankoh, S. Naito, M. Harada, T. Sakata, and M. Minoh. Free-viewpoint video synthesis for sport scenes captured with a single moving camera. *ITE Transactions on Media Technology and Applications*, 3(1):48–57, 2015.

[53] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[54] P.-C. Wen, W.-C. Cheng, Y.-S. Wang, H.-K. Chu, N. Tang, and H.-Y. M. Liao. Court reconstruction for camera calibration in broadcast basketball videos.

[55] S. Zhang. Research on camera calibration in football broadcast videos. *International Journal of u-and e-Service, Science and Technology*, 8(3):89–98, 2015.

[56] Y. Abdel-Aziz. Direct linear transformation from comparator coordinates in close-range photogrammetry. In *ASP Symposium on Close-Range Photogrammetry in Illinois, 1971*, 1971.

[57] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[58] P. Goorts, S. Maesen, Y. Liu, M. Dumont, P. Bekaert, and G. Lafruit. Automatic calibration of soccer scenes using feature detection. In *E-Business and Telecommunications*, pages 418–434. Springer, 2014.

[59] P. Goorts, C. Ancuti, M. Dumont, S. Rogmans, and P. Bekaert. Real-time video-based view interpolation of soccer events using depth-selective plane sweeping. 2013.

[60] R. Yang, M. Pollefeys, H. Yang, and G. Welch. A unified approach to real-time, multi-resolution, multi-baseline 2d view synthesis and 3d depth estimation using commodity graphics hardware. *International Journal of Image and Graphics*, 4(04):627–651, 2004.

[61] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[62] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence*, 14(4):407–422, 2005.

[63] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, and P. Bekaert. Free viewpoint video for soccer using histogram-based validity maps in plane sweeping. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, pages 378–386. IEEE, 2014.

[64] P. Goorts, S. Maesen, Y. Liu, M. Dumont, P. Bekaert, and G. Lafruit. Self-calibration of large scale camera networks. 2014.

[65] J. Puwein, R. Ziegler, J. Vogel, and M. Pollefeys. Robust multi-view camera calibration for wide-baseline camera networks. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 321–328. IEEE, 2011.

[66] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[67] T. Moons, Luc Van G., and M. Vergauwen. *3D reconstruction from multiple images, Part 1: Principles*. Now Publishers Inc, 2009.

[68] J. Ponce and D. Forsyth. *Computer vision: a modern approach*. Prentice Hall, 2nd edition, 2012.

[69] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[70] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[71] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.

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

[73] I. Reid and A. North. 3d trajectories from a single viewpoint using shadows. In *BMVC*, volume 50, pages 51–52. Citeseer, 1998.

[74] Y. Liu, D. Liang, Q. Huang, and W. Gao. Extracting 3d information from broadcast soccer video. *Image and Vision Computing*, 24(10):1146–1162, 2006.

[75] A. Yamada, Y. Shirai, and J. Miura. Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 303–306. IEEE, 2002.

[76] E. Ribnick, S. Atev, and N. P. Papanikolopoulos. Estimating 3d positions and velocities of projectiles from monocular views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):938–944, 2009.

[77] J. Ren, J. Orwell, G. A. Jones, and M. Xu. Tracking the soccer ball using multiple fixed cameras. *Computer Vision and Image Understanding*, 113(5):633–642, 2009.

[78] K. Matsumoto, S. Sudo, H. Saito, and S. Ozawa. Optimized camera viewpoint determination system for soccer game broadcasting. In *MVA*, pages 115–118, 2000.

[79] H. Kim and K. S. Hong. Robust image mosaicing of soccer videos using self-calibration and line tracking. *Pattern Analysis & Applications*, 4(1):9–19, 2001.

[80] J. Ren, J. Orwell, G. A. Jones, and M. Xu. A general framework for 3d soccer ball estimation and tracking. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 3, pages 1935–1938. IEEE, 2004.

[81] J. Ren, J. Orwell, G. Jones, and M. Xu. Real-time 3d soccer ball tracking from multiple cameras. In *Proc. of the British Machine Vision Conference (BMVC'04)*, pages 829–838, 2004.

[82] N. Ishii, I. Kitahara, Y. Kameda, and Y. Ohta. 3d tracking of a soccer ball using two synchronized cameras. In *Advances in Multimedia Information Processing–PCM 2007*, pages 196–205. Springer, 2007.

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

[84] Media Integration MICC and Communication Centre. Acm multimedia challenge 2010. Available in http://acmmm10.unifi.it/, accessed in 15th January of 2016.

[85] P. Kelly, C. Ó Conaire, D. Monaghan, J. Kuklyte, D. Connaghan, J. Agapito, and P. Daras. Performance analysis and visualisation in tennis using a low-cost camera network. 2010.

[86] S.-K. Chang and YR Wang. Three-dimensional object reconstruction from orthogonal projections. *Pattern Recognition*, 7(4):167–176, 1975.

[87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[88] Hawk-Eye Innovations. Available in http://www.hawkeyeinnovations.co.uk/, accessed in 29th December of 2015.

[89] My Zone. The hawk-eye technology. Available in http://www.studiosayers.com/storage/post-images/hawkeye_blog.jpg?__SQUARESPACE_CACHEVERSION=1295554981370, accessed in 29th December of 2015.

[90] GoalControl. Available in http://goalcontrol.de/en/, accessed in 29th December of 2015.

[91] Wikipedia. Goalcontrol. Available in https://upload.wikimedia.org/wikipedia/commons/thumb/4/44/Goalcontrol.svg/2000px-Goalcontrol.svg.png, accessed in 29th December of 2015.

[92] ChyronHego. The world's only true 3d real-time tracking system. Available in https://www.cev.ca/images/pdf/CYR-PLAYERTRACK.EN.PDF, accessed in 29th December of 2015.

[93] FIVB: Fédération Internationale de Volleyball. New volleyball tracking system trialled in montreux. Available in http://www.fivb.ch/viewPressRelease.asp?No=45951&Language=en#.VngM9fmLRD8, accessed in 29th December of 2015.

[94] Fraunhofer IIS. Goalref – goal-line technology. Available in http://www.iis.fraunhofer.de/content/dam/iis/en/doc/lv/ok/GoalRef_Flyer_en.pdf, accessed in 29th December of 2015.

[95] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM, 1996.

[96] Dr. G. R. Bradski and A. Kaehler. *Learning Opencv, 1st Edition*. O'Reilly Media, Inc., first edition, 2008.

[97] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE, 1999.

[98] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[99] E. W. Weisstein. Eigenvector, 2002. Available in http://mathworld.wolfram.com/Eigenvector.html, accessed in 05th April of 2016.

[100] C. B. Duane. Close-range camera calibration. *Photogramm. Eng*, 37(8):855–866, 1971.

[101] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[102] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[103] X.S. Gao, X.R. Hou, Ji. Tang, and H.F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[104] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[105] P. A. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure from motion. In *Computer Vision—ECCV'94*, pages 85–96. Springer, 1994.

[106] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *International journal of computer vision*, 23(3):235–259, 1997.