



BOSCH



LUNDS
UNIVERSITET

Low-Cost Webcam-Based Motion Capture System

A Bachelor's Thesis in collaboration with Robert Bosch AB

Written By

Oskar Hessler and Ebbe Persson

June 2026

Department of Computer Science
LTH, Faculty of Engineering, Lund University
Postal address: Box 118, 221 00 Lund, Sweden

Abstract

The department at Robert Bosch AB where this thesis was conducted collaborates with Bosch Sensortec, which develops Inertial Measurement Unit (IMU) sensors for navigation in areas with poor satellite coverage and for motion-tracking applications. These sensors combine accelerometers and gyroscopes to estimate position and orientation, but accumulate drift errors over time. Benchmarking this drift currently requires expensive commercial motion capture systems, which motivated this thesis.

The goal of this work was to design and implement a low-cost, easy-to-deploy software system capable of recording the three-dimensional trajectory and rotation of a chosen object using consumer-grade webcams. The system uses ArUco markers detected via OpenCV to identify the object in each camera frame. A two-stage calibration process consisting of intrinsic and extrinsic calibration, based on a chequerboard pattern, establishes a shared world coordinate system across all cameras. Stereo triangulation is then used to reconstruct three-dimensional positions, while a software timestamp-based synchronization mechanism keeps all camera streams aligned. The recording pipeline separates time-critical frame capture from computationally heavier post-processing to support real-time operation.

The system was evaluated across several key performance areas, achieving an object detection rate of 93%, trajectory coverage of 92.9%, a synchronization mean time spread of approximately 13.3 ms, a re-projection error of 2-4 pixels, and an extrinsic baseline accuracy of ± 30 mm. All key performance targets defined in the requirement specification were met. A subsequent calibration improvement using a ChArUco board, implemented as an extension after the primary evaluation phase, reduced the extrinsic baseline error to ± 10 mm and the re-projection error to 0.8-1.8 pixels, as detailed in Appendix A. The remaining limitations, primarily camera resolution and marker occlusion, are well understood and can be addressed through targeted hardware and algorithmic improvements.

This thesis demonstrates that a functional low-cost motion capture system is achievable using consumer-grade webcams. The resulting software provides the department at Robert Bosch AB with a tool for evaluating the accuracy of Bosch Sensortec's IMU-based motion-tracking algorithms and establishes a solid foundation for future development toward higher precision.

Key Words

Motion tracking, Cameras, Low-Cost , Camera Calibration, Computer Vision, ArUco Markers.

Sammanfattning

Avdelningen på Robert Bosch AB där detta examensarbete genomfördes samarbetar med Bosch Sensortec, som utvecklar sensorer av typen Inertial Measurement Unit (IMU) för navigation i områden med dålig satellittäckning och för rörelsespårningsapplikationer. Dessa sensorer kombinerar accelerometrar och gyroskop för att uppskatta position och orientering, men ackumulerar driftfel över tid. Att utvärdera denna drift kräver i dagsläget dyra kommersiella motion capture-system, vilket motiverade detta examensarbete.

Målet med arbetet var att designa och implementera ett lågkostnads- och lättimplementerat mjukvarusystem som kan registrera ett valt objekts tredimensionella rörelsebana och rotation med hjälp av konsumentklassade webbkameror. Systemet använder ArUco-markörer detekterade via OpenCV för att identifiera objektet i varje kamerabild. En tvåstegskalibreringsprocess bestående av intrinsisk och extrinsisk kalibrering, baserad på ett schackbrädemönster, etablerar ett gemensamt världskoordinatsystem över alla kameror. Stereo-triangulering används sedan för att rekonstruera tredimensionella positioner, medan en mjukvarubaserad tidsstämpelsynkroniseringsmekanism håller alla kameraströmmar synkroniserade. Inspelningspipelinen separerar tidskritisk bildinsamling från mer beräkningsintensiv efterbearbetning för att stödja realtidsdrift.

Systemet utvärderades inom flera viktiga prestandaområden och uppnådde en objekt-detekteringsgrad på 93%, baninspelningstäckning på 92,9%, en synkroniseringens medelspännvidd på cirka 13,3 ms, ett reprojektionsfel på 2–4 pixlar samt en extrinsisk baslinjenoggrannhet på ± 30 mm. Alla nyckelprestandamål som definierats i kravspecifikationen uppfylldes. En efterföljande kalibreringsförbättring med ett ChArUco-bräde, implementerad som en utökning efter den primära utvärderingsfasen, reducerade det extrinsiska baslinjefelet till ± 10 mm och reprojektionsfelet till 0,8–1,8 pixlar, vilket beskrivs i detalj i Appendix A. De kvarvarande begränsningarna, främst kameraupplösning och markörblockering, är väl förstådda och kan åtgärdas genom riktade förbättringar av hårdvara och algoritmer.

Detta examensarbete visar att ett funktionellt lågkostnadssystem för rörelsefångst är möjligt att uppnå med konsumentwebbkameror. Den framtagna mjukvaran ger avdelningen på Robert Bosch AB ett verktyg för att utvärdera noggrannheten hos Bosch Sensortecs IMU-baserade rörelsespårningsalgoritmer och etablerar en stabil grund för framtida utveckling mot högre precision.

Nyckelord

Rörelseinspelning, Kameror. Lågkostnad, Kamerakalibrering, Datorseende, ArUco-markörer.

Preface

To aid the reader's understanding, it is worth knowing that the two experimental chapters differ in scope. The analysis chapter 4 was conducted without three-dimensional object reconstruction. The system tracks ArUco-markers placed in a plane, making it possible to assess calibration performance, synchronisation effectiveness, and data stream behaviour in isolation. The result chapter 5 was conducted with full three-dimensional object reconstruction. A physical object equipped with ArUco-markers is moved freely in space, rigid body template reconstruction is used to reconstruct the object in its trajectory and orientation.

Acknowledgements

We are sincerely grateful to Robert Bosch AB for making this bachelor's thesis possible. A special thanks goes to our Bosch supervisor, Stefan Noll, who gave us this opportunity and provided guidance throughout the project. We would also like to thank his colleague, Sebastian Farås, for his helpful feedback and technical support.

This thesis would also not have been possible without the academic supervision of Christian Nyberg at the Lund University, Faculty of Engineering, LTH. We are thankful for the opportunity from Stefan, and for the essential guidance from our supervisors, which, combined with Sebastian's support, made this work possible.

Oskar Hessler & Ebbe Persson

Terminology

This chapter introduces the foundational terminology essential for understanding motion capture and its underlying principles. The following glossary provides concise definitions for key concepts mentioned multiple times in this thesis.

Table 1: Glossary of Key Technical Terms.

Concepts	Explanation
ArUco markers	A type of fiducial marker from a predefined dictionary used in computer vision. Each marker encodes a unique ID and enables detection of both position and orientation from a single camera image.
Camera calibration	The process of determining a camera's <i>intrinsic</i> and <i>extrinsic parameters</i> to establish the mathematical relationship between three-dimensional world coordinates and two-dimensional image coordinates.
Cauchy noise	Appears as occasional, very bright or very dark outlier pixels (spikes) distributed randomly across an image. It's characterized by a "heavy-tailed" distribution, making large intensity deviations probable.
Edge detection	An image processing technique that identifies boundaries where brightness or color changes sharply, used to detect object outlines and structural features.
Extrinsic matrix	A mathematical representation of a camera's position and orientation in three-dimensional space relative to a world coordinate system.
Extrinsic parameters	The <i>rotation matrix</i> $\mathbf{R}$ and <i>translation vector</i> $\mathbf{t}$ that define a camera's spatial orientation and position in the world coordinate system.
Focal length	The distance between the camera's optical center and its image sensor, measured in millimetres. It determines the field of view and magnification of the captured image.
FPS (Frames Per Second)	The number of individual images or frames captured, displayed, or processed per second.

Concepts	Explanation
Gaussian noise	Appears as random variations in pixel intensity, following a bell-shaped probability distribution. It's common in images due to sensor noise or low-light conditions, often causing a "grainy" appearance.
Gradient	In an image, the gradient at a pixel is a vector indicating the direction and magnitude of the most rapid intensity change. Its primary application in image processing is <i>edge detection</i> and boundary identification.
Homogeneous coordinates	A coordinate system that uses an additional dimension to represent geometric transformations (like translation, rotation, and scaling) as unified matrix multiplications. This simplifies calculations, particularly for handling perspective projection.
Image plane	The two-dimensional surface within a camera where the projected image of a three-dimensional scene is captured.
IMU (Inertial Measurement Unit)	A device that measures an object's specific acceleration and angular velocity, typically using accelerometers and gyroscopes. It often aids in tracking an object's orientation and motion.
Intrinsic matrix	A matrix representing a camera's internal properties, used to project three-dimensional points onto the two-dimensional <i>image plane</i> .
Intrinsic parameters	The internal camera properties such as <i>focal length</i> and <i>principal point</i> that define the <i>intrinsic matrix</i> .
Lens distortion	An optical anomaly that causes straight lines in a scene to appear bent or curved in the captured image. It refers to the inaccuracies between the geometry of a scene and its photographic representation, typically arising from the lens's design.
Motion capture	The process of digitally recording the position and orientation of objects or people as they move over time.
Occlusion	The condition in which a marker or object is blocked in a way that prevents detection.
Optical center	The point within a camera lens where light rays from a scene converge before reaching the image sensor, often considered the origin of the camera's coordinate system.
Pixel coordinates	A two-dimensional coordinate system representing the location of each pixel in an image.
Pose estimation	The process of determining the position and rotation of an object relative to a coordinate system.

Concepts	Explanation
Principal point	The point on the <i>image plane</i> where the camera's optical axis intersects it. It represents the center of projection in the image and is crucial for <i>camera calibration</i> , often denoted as (c_x, c_y) .
Projection matrix	The combination of the <i>extrinsic</i> and <i>intrinsic matrices</i> that transforms three-dimensional world coordinates into two-dimensional pixel coordinates.
Real-time processing	The ability to process data inputs and generate outputs within strict, time-critical deadlines, responding to events as they happen rather than after they conclude.
Re-projection error	The pixel distance between an observed image point and its projected three-dimensional model point. It measures the accuracy of a <i>three-dimensional reconstruction</i> .
Rigid body	An object or part of an object whose parts maintain fixed spatial relationships to each other.
Rotation matrix	A 3×3 mathematical matrix used to describe the three-dimensional rotation of an object or coordinate system.
Three-dimensional reconstruction	The process of generating a three-dimensional model or representation of a real-world object, scene, or motion from two-dimensional images or sensor data.
Trajectory	In motion analysis, it refers to the path or curve that an object follows as it moves through three-dimensional space. It is typically represented as a sequence of discrete positions recorded over time.
Translation vector	A vector specifying the direction and distance an object or coordinate system is moved, without rotation or scaling.
Triangulation	A geometric method used to determine the three-dimensional position of a point by observing it from two or more known camera positions, by intersecting projection rays from multiple views.

Contents

1	Introduction	11
1.1	Background	11
1.2	Purpose	12
1.3	Targets	12
1.4	Problem Statement	12
1.5	Motivation for the Thesis	13
1.6	Limitations	13
1.7	Division of Labour	14
1.8	Source Evaluation	14
2	Technical Background	17
2.1	Geometry of Image Formation	17
2.2	Lens Distortion	20
2.3	Camera Calibration	21
2.4	Three-Dimensional Reconstruction	23
2.5	Image Processing	24
2.6	Edge Detection	26
2.7	Glyph Detection	27
3	Method	29
3.1	Requirements and Scope	29
3.1.1	Problem Definition and Objectives	29
3.1.2	Key System Requirements	30
3.2	System Overview and Resources	30
3.2.1	Hardware Components	30
3.2.2	Software Environments	31
3.2.3	High-Level System Architecture	32
3.3	System Workflow	33
3.3.1	Overall Workflow	33
3.3.2	Data Acquisition	34
3.4	Implementation Details	34
3.4.1	Camera Calibration	34
3.4.2	Data Synchronization	34
3.4.3	Object Detection	35
3.4.4	Three-Dimensional Reconstruction	35
3.4.5	Trajectory Tracking	36
3.5	Validation Methodology	36
3.5.1	Experimental Setup	36
3.5.2	Performance Metrics	38
3.5.3	Validation Procedure	40

4	Initial System Performance and Analysis	47
4.1	Calibration Evaluation	47
4.1.1	Calibration Accuracy	47
4.1.2	Methodological Implications	48
4.2	Data Synchronization Effectiveness	48
4.2.1	Temporal Accuracy	48
4.2.2	Methodological Implications	50
4.3	Object Detection Performance	51
4.4	Three Dimensional Reconstruction	52
4.5	Data Stream Performance	53
4.5.1	Temporal Continuity and Stability	53
4.5.2	Data Stream Reliability	54
4.6	Trajectory Tracking Performance	55
4.6.1	Trajectory Coverage and Precision	55
4.6.2	Methodological Implications	56
4.7	Overall System Performance and Limitations	56
4.7.1	Summary of System Achievements	56
4.7.2	Identification of System Limitations and Bottlenecks	56
4.7.3	Synthesis and Fitness for Purpose	57
5	System Refinement and Final Results	59
5.1	System Improvements	59
5.1.1	Calibration	59
5.1.2	Rigid Body Template	60
5.2	Overall System Performance	60
5.2.1	Core System Capabilities	60
5.2.2	Key Performance Metrics	61
5.3	Visual Representation of Achieved Results	62
5.4	Critical Component Performance Indicators	63
5.4.1	Camera Calibration Summary	63
5.4.2	Object Detection Success Rate	64
6	Discussion	65
6.1	System Performance in Relation to Purpose	65
6.2	Key Design Decisions and Their Consequences	65
6.2.1	ArUco Markers Strategy Evaluation	66
6.2.2	Synchronization Strategy Evaluation	67
6.2.3	Rigid Body Template Strategy Evaluation	68
6.2.4	Camera Calibration Strategy Evaluation	68
6.3	Limitations and Sources of Error	69
6.3.1	Three-Camera Triangulation	69
6.3.2	Camera Drivers	70
6.3.3	Causes and Implications of Signal Faults and Missing Data	71
6.4	Societal Benefits	74
7	Conclusion	75
7.1	Summary of Key Findings	75
7.2	Answers to problem statements	75
7.3	Contribution to Purpose	77
7.4	Future Development Opportunities	77
7.4.1	Applications of System	77
7.4.2	Possible Improvements of System	77

Appendix A	Improvements of System	81
A.1	Introduction	83
A.2	Quantitative Performance Results	83
A.3	Conclusion of System Improvements	86
Appendix B	Requirement Specification	87
B.1	Version History	89
B.2	Background	89
B.3	Priority Definition	89
B.4	Workgroup	90
B.5	Concept List	91
B.6	Requirements	92
B.6.1	Functional Requirements	92
B.6.2	Quality Requirements	94

Chapter 1

Introduction

This chapter introduces the context, objectives, and scope of the project. The thesis starts by explaining why this study is important and what problem it aims to solve. Then, clearly states the goals and objectives of this bachelor thesis. It also mentions any possible limitations to expect during the project and how the work was divided. This introduction provides a roadmap for what you can expect to learn in the following chapters about this low-cost motion tracking system.

1.1 Background

Robert Bosch GmbH is a German-based company with over 400,000 employees worldwide. While its roots stem from the automotive industry, the company has branched out to cover broader fields such as mobility, consumer goods, and building technology. This bachelor's thesis is conducted at Robert Bosch AB, a research and development centre located in Lund, Sweden.

The team at Robert Bosch AB works closely with Bosch Sensortec, a subsidiary of Bosch, focusing on embedded sensor development in areas of acoustics, wearables, and motion tracking. Bosch Sensortec develops Inertial Measurement Unit (IMU) sensors which, based on a combination of an accelerometer and a gyroscope, can determine the position and rotation of itself.

The usage of these sensors is intended as a complement to GPS navigation in locations that have little to no satellite connection. One example use case for the sensor could be in cars navigating underground in tunnels or when trying to navigate indoors in a building with poor GPS reception.

The problem with these sensors is that they have a small margin of error that grows with continued use. As a result of these real-world sensor errors, the calculated position and orientation gradually deviate from reality. This can be benchmarked using motion capture systems, but those are usually expensive.

A possible solution for this is to develop software that has the ability to record a trajectory with rotation of a chosen object. To achieve this solution, the chosen shape of the object is supposed to be beneficial for the determination of its position and rotation, since reading from an angled surface using a camera is not optimal for stable results. This would also create a need for camera calibration to validate the computer vision against the real world. But mainly an algorithm to determine the trajectory and rotation of the moving object.

1.2 Purpose

The purpose of the thesis is to develop a software to motion-track a specific object using webcams, and investigate use cases for this software. The expected outcome is that, based on this thesis, Bosch Sensortec will be able to use the developed software to evaluate sensor tracking algorithms.

1.3 Targets

The primary goal of this thesis is to create a low-cost motion-tracking software system that can be set up and used quickly to benchmark and measure sensor deviation, providing useful input for other tools developed by Bosch. To achieve this, the thesis will fulfil the following objectives:

1. Develop three-dimensional motion capture system based on object recognition and webcams.
2. Conduct interviews with domain experts to optimise specifications and needs of the software.
3. Find the most suitable image processing and recognition library for this project.
4. Find out the software's capability and accuracy of extracting an object's position and rotation from camera images.
5. Implement data recording synchronisation for multiple cameras with mutual reference time.
6. Make the software independent of fixed camera positions by implementing a software calibration initialisation step.
7. Develop a filter that removes positional and rotational anomalies.

1.4 Problem Statement

This thesis aims to answer the following questions:

1. What method can be developed to enable low-cost and low-setup-time recordings of three-dimensional trajectories in space?
2. How can current image recognition methods be used to generally detect a pre-defined object in space?
3. How can a trajectory be calculated using a multi-camera setup?
4. What calibration method can be introduced to effectively mitigate the challenges of camera setup and camera projections for three-dimensional trajectory recordings?
5. What synchronisation method can be used to synchronise separate camera recordings?
6. What method can enable real-time operation of three-dimensional trajectory recording?

1.5 Motivation for the Thesis

This thesis provides the opportunity to investigate the various applications of motion tracking software. This is considered an advantage from a labour-market perspective, based on the fields of work of nearby companies.

Bosch Sensortec expects a software system for motion capture which could assist in evaluating sensor algorithm accuracy. This software could be a helpful tool in parts of their sensor algorithm development.

More accurate and well-characterised sensors are also a prerequisite for more robust systems in areas such as:

- General activity recognition
- Driver assistance systems and autonomous vehicles
- Navigation for drones and robots
- Indoor positioning where GPS is not available

Better sensors and a better understanding of their limitations could lead to:

- Increased traffic safety
- Reduced energy consumption through more precise control systems
- More efficient automation and logistics

1.6 Limitations

Possible limitations that have been identified:

1. Keeping the cameras synchronised, meaning that hardware limitations may prevent each frame from being captured at the exact same moment.
2. The camera resolution and FPS could be limited by the computation power of the computers used.
3. Testing will be conducted with controlled conditions, and results may differ when the system is applied in real-world environments with varying lighting, backgrounds, or image quality.
4. Finally, the project will be implemented and tested on a single hardware environment, which may limit reproducibility or performance comparisons across different systems.

1.7 Division of Labour

Both authors contributed equally to the thesis which is presented in table 1.1. Table 1.1 represents an approximation of the division of labour across the thesis in percentage.

Table 1.1: Estimated division of labour in percentage.

Task	Oskar Hessler	Ebbe Persson
Information gathering	50%	50%
Requirement specification	50%	50%
Camera-based calibration implementation	60%	40%
Camera-based tracking implementation	40%	60%
Calibration of system	50%	50%
Testing and evaluation of the system	50%	50%
Report writing	60%	40%
Preparation for presentation	50%	50%
Poster	50%	50%
Presentation	50%	50%

1.8 Source Evaluation

This subchapter is to evaluate the trustworthiness of the sources referenced or cited in this report.

Satya Mallick [1] runs the blog learnopencv.com from which the post was retrieved. He is also CEO of the website opencv.org and founder of Big Vision AI, a consulting company specialising in OpenCV among other areas which is collaborating with well-known companies.

Kenji Hata [2] is an M.S. in Computer Science at Stanford University, co-author Silvio Savarese is an Associate Professor of Computer Science at Stanford University and is instructor for the course: "Computer Vision, From 3D Perception to 3D Reconstruction and beyond", the course from which this material was sourced.

K. Sadekar [3] is a Ph.D student at Portland State University where his research includes 3D computer vision, computational imaging and the design of synthetic dataset generation pipelines. This post was also posted on the blog learnopencv.com.

The book "An Invitation to 3-D Vision"[4] is written by the four professors, Yi Ma (visiting professor at University of California, Berkeley), Jana Košecká (George Mason University), Stefano Soatto (University of California, Berkeley), S. Shankar Sastry (University of California, Berkeley).

The article "A Survey of 3D Reconstruction" [5] is written by four authors. Shuai Liu (Ph.D student at Nanyang Technological University, Singapore), Tingyan Xing, Mengmeng Yang (Ph.D student at Tsinghua University, China) and Ran Yang. The article was edited by Professor Kenneth

J Loh (University of California, San Diego).

"Triangulation" are power-point slides[6] originating from the course "Computer Vision" at Carnegie Mellon University, School of Computer Science. Kris Kitani is an associate research professor of the Robotics Institute at Carnegie Mellon University and a former instructor of the course.

The article "Triangulation" from which the image [7] was retrieved, was written by the artist-teacher Darren R. Rousar who teaches the Sight-Size method of cast drawing.

The book "Foundations of Computer Vision"[8] is a book published by the MIT Press. Authored by the two professors William T. Freeman, Antonio Torralba and associate professor Philip Isola. All affiliated with MIT(Massachusetts Institute of Technology).

"Image Noise Models"[9] in "The essential Guide to Image Processing" is written by Professor Charles Bonchelet at Delaware University.

The article "Group Sparse Representation for Restoring Blurred Images with Cauchy Noise"[10] is written by Sangwon Lee and Professor Myungjoo Kang at Seoul National University.

The book "Digital Image processing"[11] is written by the researchers Mark Burge (Senior Scientist at MITRE) and Willhelm Burger (Ph.D. at the university of Utah) with research experience at prominent universities)

The article "Understanding the gamma adjustment of images"[12] was written by Lecturer Jakkarin Singnoo Chulalongkorn University, Thailand, and Professor Graham D. Finlayson at University of East Anglia, England.

The article "An improved image enhancement algorithm: Radial contrast- limited adaptive histogram equalization" [13] is written by: Professor Chunsheng Hu (University of Chinese Academy of Sciences), Professor Hao Li (Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates), Cailian Zeng, Professor Xiaoli Ji (Nanjing University) and Ph.D Teng Ma (Tsinghua University, China).

The post "Understanding edge detection in image processing"[14] posted on ultralytics.com by Aburami Vina. The post "Extended kalman filter" was also posted on the same website. Ultralytics is a company specializing in AI research but also in fields as Computer Vision. Working with companies such as Intel, Sony, and Raspberry Pi. The companies expertise and collaborations lend authority to the content.

The tutorials "Detection of ArUco Markers"[15] and "Detection of ChArUco Boards"[16] is retrieved from the openCV documentation as well as "Camera Calibration and 3D Reconstruction"[17]. OpenCV(Open Computer Vision) is the largest computer vision library and is a fundamental tool for computer vision widely used in the industry of computer vision.

The paper "Means and averaging in the group of rotations"[18] is written by Professor Maher Moakher at University of Tunis El Manar.

The book "Introduction to Kalman Filter and Its Applications" is written by Ph.D Youngjoo Kim(Korea Advanced Institute of Science and Technology, South Korea) and Professor Hyochoong Bang(Korea Advanced Institute of Science and Technology, South Korea).

All of the discussed sources above are authored by professors, researchers or experts affiliated with well-known universities, reputable academic publishers of established organisations within the field of computer vision. These affiliations indicate that the sources have undergone rigorous review processes, thereby establishing their trustworthiness for the context of this report.

Chapter 2

Technical Background

This chapter establishes the technical foundation for understanding motion capture. The core technologies and computational techniques that convert physical movement into digital data, providing essential context for the analyses presented in this thesis are explored.

2.1 Geometry of Image Formation

Having some understanding of image formation and the geometry behind it is a fundamental part of the knowledge needed to create a motion-tracking software system. In the post Geometry of Image Formation[1], Mallick explains the problem of a three-dimensional point being projected onto a two-dimensional image plane.

To understand this problem Mallick[1] starts with the example of trying to find the pixel coordinates (u, v) of a three-dimensional point $\mathbf{P}$ in an image taken by a camera in a three-dimensional room. For this to be possible there are three coordinate systems in play.

First, there is the World coordinate system, this is the coordinate system of the actual room. Any point in this room is given by, (X_W, Y_W, Z_W) and this point is found by measuring the distance along the three axes.

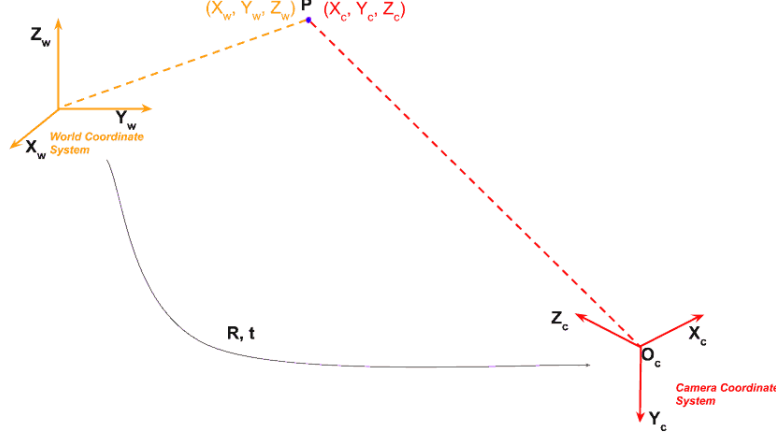


Figure 2.1: Real World Coordinates and Camera Coordinates. Source: Adapted from [1]

Figure 2.1 illustrates that if there is a camera put in a room, one finds that the perspective of what this camera captures of the room can also be put in a coordinate system (X_C, Y_C, Z_C) . If the camera's position would be the origin of the room and its axes are aligned with the real world axes so that:

$$(X_C, Y_C, Z_C) = (X_W, Y_W, Z_W)$$

In this case image formation would not be a problem in the same way. But that would be an impractical constraint, one wants to be able to put the camera anywhere in a room and point in any direction. This means that the camera can be positioned at an arbitrary location $\mathbf{t}$ and rotated by an arbitrary rotation $\mathbf{R}$ described as a 3x3 matrix. Using $\mathbf{R}$ and $\mathbf{t}$ as the relationship between the camera and the real world, the perspective of the camera can be translated into the real world. This can be done using matrix multiplication:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t} \quad (2.1)$$

Mallick[1] also explains that sometimes the expression above is written in a more compact form called the Extrinsic Matrix. To obtain the 3x4 Extrinsic Matrix the 3x1 translation matrix $\mathbf{t}$ is appended to the right of the 3x3 rotation matrix $\mathbf{R}$:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = [R|\mathbf{t}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.2)$$

It is also explained by Mallick[1] that once the camera coordinates are acquired using the rotation and translation of the world coordinates we are able to obtain the pixel coordinates (u, v) . This is done by projecting the point onto the image plane.

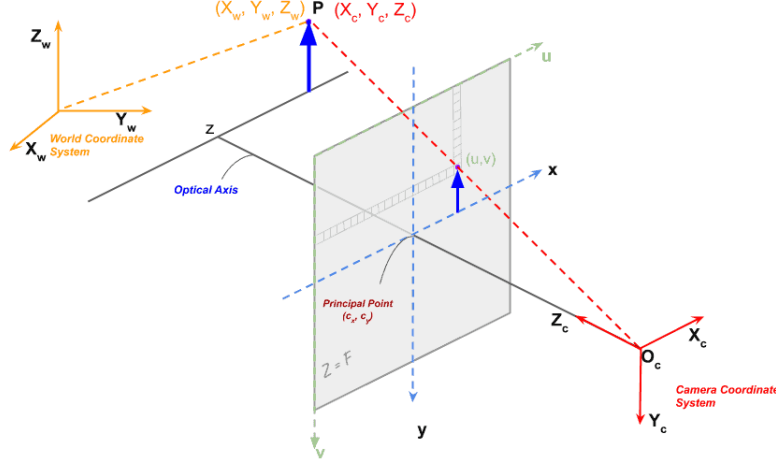


Figure 2.2: Projection of point P onto the image plane is shown. Source: [1]

Figure 2.2 shows the camera projection in case of a pinhole camera represented by O_C (Using a pinhole camera is for mathematical convenience). Given that the image plane (x, y) is placed at a distance f (focal length) away from the optical centre, the image plane coordinates of the three-dimensional point (X_C, Y_C, Z_C) are given by:

$$x = f \frac{X_C}{Z_C} \quad y = f \frac{Y_C}{Z_C} \quad (2.3)$$

These two equations can be rewritten in matrix form as follows.[1]

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \quad (2.4)$$

The matrix K shown below is called the intrinsic matrix and contains the intrinsic parameters of the camera. But this matrix only contains f besides 0:s and 1:s.

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

But this is usually not enough and a few different parameters have to be taken into account.[1]

- f_x, f_y since the pixel aspect ratio may not be 1:1, it may have two different focal lengths.
- c_x, c_y since the optical centre and the centre of the image coordinate system may not coincide.
- γ since there may be a skew between the x and y axes of the camera.

This means that the intrinsic matrix K can be rewritten as:

$$K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

In the above equation the x and y pixel coordinates are with respect to the centre of the image. It is now possible to calculate the pixel coordinates (u, v) .

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \quad (2.7)$$

where,

$$u = \frac{u'}{w'} \quad (2.8)$$

$$v = \frac{v'}{w'} \quad (2.9)$$

2.2 Lens Distortion

The pinhole model [2] is a fundamental mathematical technique to convert three-dimensional space to a two-dimensional plane disregarding distortion and lens blur. The model builds on three-dimensional points in the real world projecting on the image plane as straight lines preserving geometric relationship. However, real lenses have physical deviations in their construction. These variances, commonly known as lens distortions, produce anomalies in lenses and bend light unevenly. This leads to a systematic, warped, rearrangement of image points. Correcting these imperfections is essential in computer vision to improve accuracy.

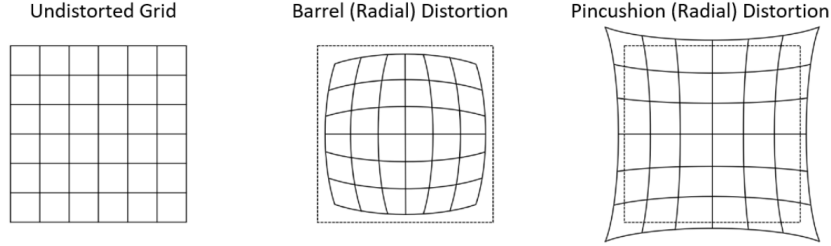


Figure 2.3: Comparison of pincushion and barrel radial distortion.

Radial distortion is one type of distortion and can be grouped into two categories. Barrel distortion and pincushion distortion [3]. Both types of distortion are bending light rays in the edges of the image and make straight lines appear curved. The difference between barrel and pincushion distortion is the curvature of the line. Barrel distortion has negative radial displacement and bows outward while pincushion distortion has positive radial displacement and bends inward as seen in Figure 2.3. Distortion is generally more pronounced towards the edges of the image and is particularly noticeable in wide-angle lenses.

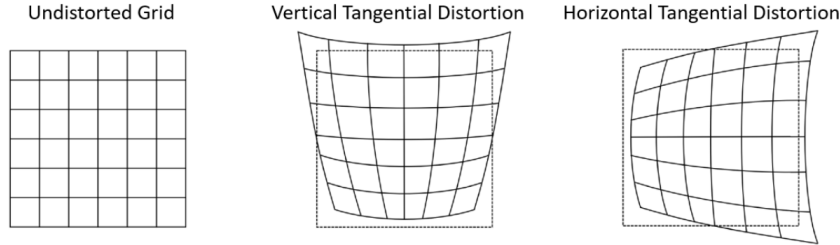


Figure 2.4: Comparison of vertical and horizontal tangential distortion.

Another type of distortion is tangential distortion. This type of distortion is caused by lens-sensor misalignment, where the lens and sensor are not parallel to each other [3]. This makes the image points appear shifted or stretched and results in a tilted appearance as seen in Figure 2.4.

These distortions in lenses displace feature points from their true positions and cause computer vision to differ from the real world. This is a problem when translating between two-dimensional planes and three-dimensional space which make positional assumptions skew from reality. This can be fixed by calibration, connecting the plane and the space, and as a result undistort the image.

2.3 Camera Calibration

Camera calibration is the step of the image processing pipeline that ties computer vision to the real world. Its purpose is to establish the mathematical relationship between three-dimensional world points and their two-dimensional image projections, enabling accurate spatial measurements and supporting downstream tasks such as three-dimensional reconstruction and triangulation. This is done by undistorting the image plane using algorithms and connecting reference coordinates from three-dimensional space to the processed two-dimensional plane.

In order to understand the calibration procedure, it is necessary to understand projection matrices, the relationship between the intrinsic and extrinsic matrices, and what each matrix represents, as these are the key outputs obtained from the calibration process.

The intrinsic matrix $\mathbf{K}$, as mentioned in section 2.1, is responsible for scaling focal length and pixel size and therefore converting real-world distances to pixel units, as well as translating the principal reference point. While not directly part of the intrinsic matrix, calibration also determines lens distortion coefficients, which are outputs used to correct lens anomalies and improve three-dimensional reconstruction. The intrinsic matrix encapsulates the camera’s internal geometry, mapping three-dimensional points on the camera’s coordinate system to two-dimensional coordinates on the image plane.

The extrinsic matrix is a combined matrix, as mentioned in section 2.1, and is responsible for describing the camera’s position and orientation in three-dimensional space. The extrinsic matrix consists of a rotation matrix $\mathbf{R}$, which specifies the camera’s orientation, and a translation vector $\mathbf{t}$, which specifies the camera’s position in three-dimensional space.

The projection matrix $\mathbf{P}$ is the product of the intrinsic and extrinsic matrix factors, as seen in equation 2.10. This combined matrix allows a three-dimensional point from the world coordinate system to be projected onto its corresponding two-dimensional pixel coordinates in the image plane. The extrinsic matrix first transforms the three-dimensional point from world coordinates to the camera’s coordinate system, after which the intrinsic matrix projects the point in camera coordinates onto the two-dimensional image plane. Accurate estimation of the projection matrix is

critical because errors in these parameters propagate into reconstructed three-dimensional points, affecting the precision of motion capture and spatial measurements.

$$\underbrace{P}_{\text{Projection}} = \underbrace{K}_{\text{Intrinsic}} \times \underbrace{[R|t]}_{\text{Extrinsic}} \quad (2.10)$$

There are three major calibration methods used to extract intrinsic and extrinsic data: geometric clues/auto-calibration, deep learning, and calibration patterns. Geometric clues (also known as auto-calibration) extract features from the image such as straight lines, light sources, or colours and generate coordinates based on known geometric properties. Deep learning handles limited-data cases based on training data. Calibration patterns are used when the image capture environment can be controlled. Each method aims to estimate intrinsic and extrinsic parameters as accurately as possible, since these directly impact the quality of three-dimensional reconstruction.

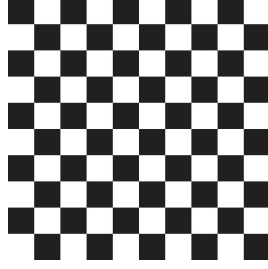


Figure 2.5: Chequerboard-pattern used for camera calibration.

Calibration patterns use known references to generate positional data. An example of a calibration pattern is the chequerboard pattern seen in Figure 2.5. The pattern has known square dimensions and known overall dimensions. Using these references, in combination with known lens distortion, it is possible to determine the intrinsic and extrinsic values and create the matrices used to determine positional data in three-dimensional space from an image plane, which are then used in triangulation and motion capture applications.

In order to fully calibrate a camera using patterns, a known calibration pattern is used to estimate the intrinsic and extrinsic data. Once these parameters (intrinsic matrix, extrinsic matrix, and distortion coefficients) are estimated, it becomes possible to accurately translate between the two-dimensional image plane and three-dimensional space, enabling accurate three-dimensional reconstruction, measurement, and motion tracking.

2.4 Three-Dimensional Reconstruction

Three-dimensional reconstruction is the process of estimating the spatial position of real-world points using their projections onto multiple two-dimensional image planes [4]. Since the image-formation process maps three-dimensional points to two-dimensional coordinates, depth information is inherently lost during projection. Recovering this missing depth is essential in motion capture systems, where accurate spatial positions are required to represent motion in real-world coordinates [5].

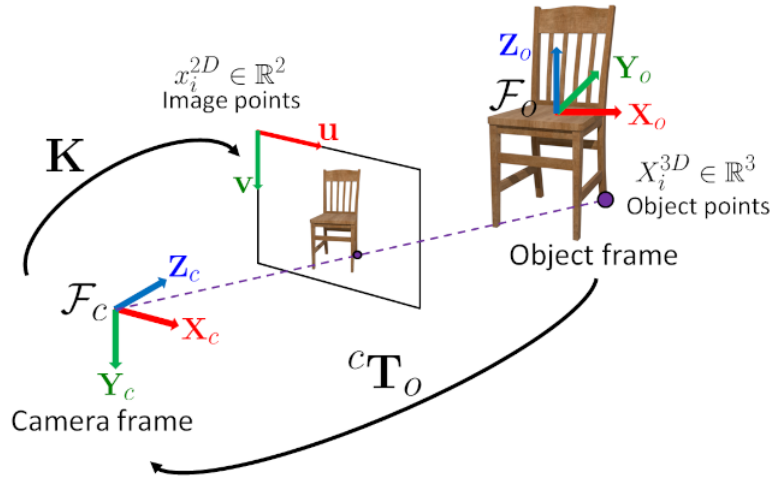


Figure 2.6: Visual representation of projecting three dimensional coordinates onto a two dimensional plane. Source [17]

Due to the loss of depth information during perspective projection, a single camera view is insufficient to uniquely determine three-dimensional point locations [4]. An image observation corresponds to a ray extending from the camera's optical centre into the scene, and any point along this ray projects to the same pixel coordinates, as shown by the purple dotted line in Figure 2.6. Without additional constraints, the true depth of the point cannot be determined. By observing the same point from multiple viewpoints, the corresponding projection rays intersect in three-dimensional space, allowing the original position to be estimated [5]. This principle forms the foundation of stereo vision and multi-view reconstruction systems. In motion capture pipelines, multiple cameras are typically used to ensure reliable reconstruction of moving markers or objects.

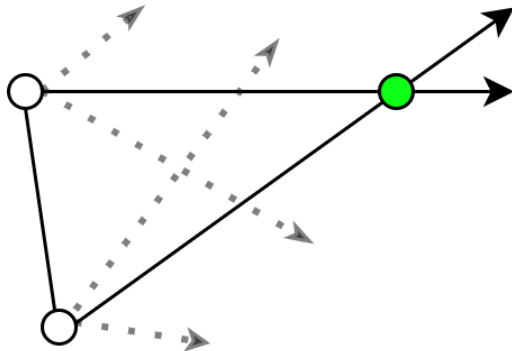


Figure 2.7: Example of triangulation in two dimensions.

Triangulation is the geometric method used to recover three-dimensional point positions by combining corresponding observations from multiple calibrated camera views [6]. Each detected image point defines a projection ray in space based on the camera’s intrinsic and extrinsic parameters. When a feature is observed in at least two views, triangulation estimates the three-dimensional point that best satisfies all projection constraints, as shown in Figure 2.7. In practice, noise in image measurements and calibration errors prevent perfect ray intersection, so the reconstructed position is estimated by minimizing the re-projection error across views. Using more than two views improves accuracy, reduces noise, and handles obstructed points more robustly.

The accuracy of three-dimensional reconstruction is fundamentally dependent on the precision of the camera calibration parameters used in the projection model [5]. Intrinsic calibration errors, such as inaccurate focal length or uncorrected lens distortion, lead to incorrect ray directions, while extrinsic calibration errors cause misalignment between rays from different cameras. These errors propagate into the reconstructed three-dimensional points, resulting in spatial bias or increased reconstruction noise. Accurate and consistent calibration is therefore a prerequisite for reliable three-dimensional reconstruction in multi-camera motion capture systems [8].

2.5 Image Processing

High-end motion capture setups are expensive because they are designed to reduce noise and distortion while maintaining optimal spatial conditions. If high-end setups are replaced with a cheaper alternative such as webcams, similar results can still be achieved by manipulating the raw image feed. This is accomplished through image processing.

The purpose of image processing is to apply filters and adjustments to captured images in order to create optimal image conditions. These improved images are then processed by three-dimensional reconstruction algorithms to improve accuracy. In order to understand image processing, it is first necessary to understand why it is needed.

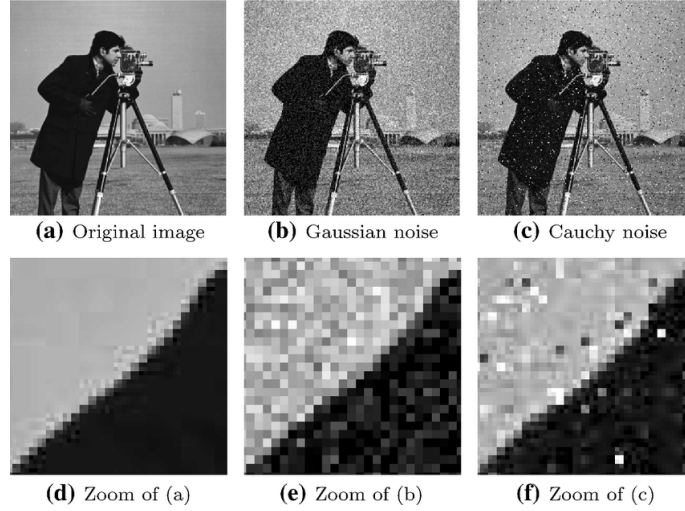


Figure 2.8: Comparison of Gaussian and Cauchy noise. Source [10].

When an image is captured under sub-optimal conditions, flaws commonly become visible in the resulting image. These flaws originate from the photographic sensor producing faulty readings and generating random variations in brightness or colour information. These faulty pixel values can create spurious features that interfere with the triangulation algorithm and result in incorrect positional outputs. These faults, referred to as noise, are commonly grouped into Gaussian and Cauchy noise [9] and can be seen in Figure 2.8.

In order to reduce random noise in images, filters can be applied. Common filters used for noise reduction include Gaussian blur, median filtering, and bilateral filtering [11]. Each filter comes with advantages and disadvantages, and certain conditions make some filters more suitable than others. Gaussian blur applies a weighted average to neighbouring pixels, resulting in effective reduction of Gaussian noise at the expense of image sharpness and detail. Median filtering replaces each pixel value with the median value of neighbouring pixels and is therefore effective at removing salt-and-pepper (Cauchy) noise, also at the expense of sharpness and detail. Bilateral filtering, unlike Gaussian blur and median filtering, preserves image details more effectively by considering both spatial proximity and intensity differences in a non-linear manner. This results in improved noise reduction while preserving sharpness, although at the cost of increased computational complexity and processing time.

Another problem affecting images is illumination level. When illumination is either too bright or too dark, the photographic sensor becomes less accurate at separating differences within the image [12]. This results in detail loss, such as reduced distinction between core features and edges. Similar to noise, this can negatively affect accuracy later in the three-dimensional reconstruction process.

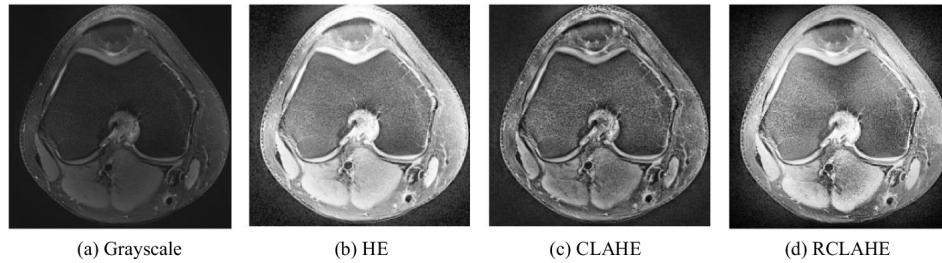


Figure 2.9: Comparison of different types of Histogram Equalisations. Source [13]

There are several solutions to illumination-related problems. The most direct solution, when possible, is to adjust the lighting conditions in the environment where image capture takes place. If this is not possible, image-processing adjustments can be applied instead. The two main methods used to address illumination problems are Gamma Correction and Adaptive Histogram Equalisation. Although both methods aim to improve image visibility, they achieve this through different approaches. Gamma Correction adjusts the overall brightness and tonal range of the entire image [12], while Adaptive Histogram Equalisation enhances local contrast, making features more visible despite varying illumination across image regions, as shown in Figure 2.9 [13].

There are both advantages and disadvantages associated with image processing. While filters and adjustments can improve image quality and thereby improve three-dimensional object pose estimation and stability, image manipulation also introduces trade-offs. Loss of fine detail is a significant issue and should be considered when applying noise-reduction and smoothing filters. Some filters, when applied aggressively, can also introduce new artefacts such as halo effects or patchiness that negatively affect image quality. The processing time required to apply filters to every frame must also be considered, since image processing can be computationally intensive. Consequently, extensive parameter tuning may be required to achieve the desired performance. A balance between the advantages and disadvantages of image processing must therefore be carefully weighed.

2.6 Edge Detection

Edge detection is an image processing technique that identifies points where brightness and colours change sharply, in order to outline objects, create boundaries and add structure[14]. This enables a computer to separate shapes, measure dimensions and determine how parts of a scene connect.

The process typically begins by converting the image to greyscale, making it easier for the computer to focus on the brightness of each pixel rather than colour[14]. This is done by converting the image to greyscale, which means that the pixels in the image only show brightness.

Abirami Vina[14] also explains that the next filter scans to find places within the image with a high gradient, which are nearby places within the image with a drastic change in brightness. The algorithm continues to refine the image by removing small details and only keeping the significant shapes and lines, resulting in a clear outline and image.

There are two main edge detection methods[14]: gradient-based edge detection, which is described above, and Gaussian-based edge detection, which adds a blurring step to eliminate false edges. The Gaussian-based method is supposed to work better with low-quality images with a lot of noise.

2.7 Glyph Detection

Glyphs are used in computer vision to identify objects or replace them with, for example, pictures.

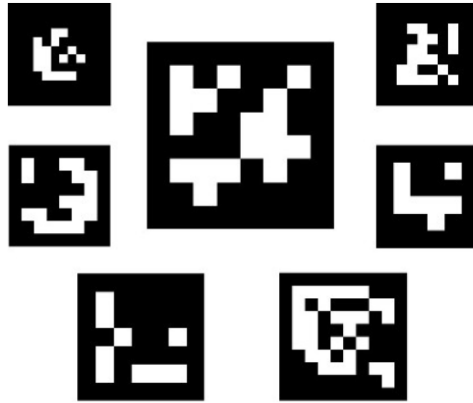


Figure 2.10: Example of ArUco Markers. Source: [15]

Glyphs have various use cases such as object recognition, augmented reality, and pose estimation (X, Y and Z positions as well as yaw, pitch and roll rotations). There are many different kinds of glyphs in computer vision and ArUco Markers[15] are one kind shown in Figure 2.10.

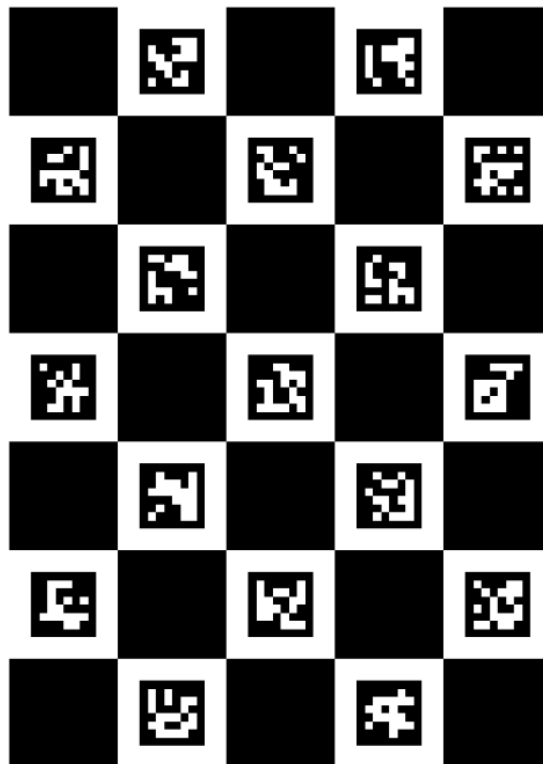


Figure 2.11: Example of ChArUco Board Source: [16]

These markers consist of a black border and a grid of $N \times N$ bits that can encode a large number of unique markers, with the exact dictionary size determined by the chosen ArUco library, which can be used for identifying objects or estimating their position and rotation. However there are more precise ways to determine the rotation of these markers. This is done by combining ArUco Markers with a chessboard and is called ChArUco, an example of this is shown in Figure 2.11. Since the ArUco markers are fitted inside a chessboard this requires the ChArUco board to be larger than if using a ArUco marker. The ChArUco board is an effective tool for the intrinsic calibration of a camera since the ArUco markers in the board are used as reference points. This enables the possibility of not having the whole board detected in the frame making it easier to getting the outlining edges of cameras vision to be undistorted.

Chapter 3

Method

This chapter outlines the methodology used to develop and validate the motion capture system. The process began with the foundational literature study (detailed in Chapter 2) to establish the system’s core requirements and design, followed by an iterative development cycle. In this cycle, the system was implemented, its performance was analysed, and the findings were used for targeted improvements. The following subsections detail the methods used in each stage of this workflow.

3.1 Requirements and Scope

This section briefly reiterates the core problem definition and project objectives, and then details the key functional and quality requirements that precisely define the capabilities and performance expectations of the three-dimensional trajectory tracking system.

3.1.1 Problem Definition and Objectives

As detailed in Section 1.4, the primary challenge addressed by this project is the development of a low-cost and low-setup-time method for recording three-dimensional trajectories in space, utilizing current image recognition techniques, multi-camera setups, and robust calibration and synchronization.

The overarching objective of this work, derived from the targets outlined in Section 1.3, was to develop three-dimensional motion capture software based on object recognition, capable of accurately extracting an object’s position and rotation from camera images, implementing robust data recording synchronization for multiple cameras, and enabling independence from fixed camera positions through software calibration. These foundational problem definitions and objectives directly guided the functional and quality requirements discussed in Section 3.1.2, and subsequently informed all design and implementation decisions within the methodology.

3.1.2 Key System Requirements

The development of the three-dimensional trajectory tracking system was guided by a defined set of functional and quality requirements. A comprehensive detailing of these requirements can be found in the *Requirements Specification* in Appendix A, chapter B.6.

The key requirements for this project, that were critical for outlining its scope, include:

1. **Core Functional Operations (A.7.1.3):** These define the fundamental user interactions, such as initiating and terminating motion recordings, exporting recorded data, and entering calibration mode.
2. **Synchronization Accuracy (B.6.2.3):** This ensures more reliable data by mandating that frame groups consist only of frames captured within a 0.02-second temporal variance.
3. **Standardised Output Data Format (B.6.1.4):** This ensures that the output data is formatted for seamless integration with other company in-house projects.
4. **Real-World Correlation and Accuracy (B.6.2.4):** This requirement specifies that the developed software must demonstrate accuracy in correlation with real-world movement.
5. **User Interface and Workflow (B.6.1.1, B.6.1.2):** These define the overall operational scope and the specific tasks supported by the project's interface.
6. **Motion Capture Object (B.6.1.5):** This specifies the particular object this motion capture system is designed to track.

These requirements formed the fundamental basis for the hardware and software selections, the design architecture, and the validation methodology discussed within this chapter.

3.2 System Overview and Resources

This section outlines the system on a hardware and software level as well as introducing what resources have been used.

3.2.1 Hardware Components

The system relies on a combination of standard computing hardware and specialized imaging equipment.

3.2.1.1 Computing Hardware

The computational backbone of the system, responsible for executing all image processing and algorithmic tasks, was provided by a laptop. This machine featured an Intel i5-1145G7 2.6 GHz (quad-core) CPU, integrated Intel Iris Xe Graphics GPU, and 16 GB of RAM, providing sufficient processing power for the targeted real-time operations.

3.2.1.2 Camera System

Image acquisition, a critical component for three-dimensional reconstruction, was enabled by three Logitech MX BRIO webcams. These cameras were selected for their ability to provide 1080p resolution at 60 frames per second, alternatively 4k resolution at 30 frames per second, both with image enhancement algorithms, which directly addresses the system's requirement for high-

frequency motion data (see Section 3.1.2). Their optimal focus range of 0.3 to 2.5 meters further supported the intended operational workspace.

3.2.1.3 Motion Capture Object

The motion capture object used in this project consists of an A4 sheet of paper with six ArUco markers from the DICT 6×6 250 library arranged in a 3×2 grid, each marker printed at 53 mm per side. The sheet provides a rigid, flat surface with multiple uniquely identifiable markers to support both position and rotation estimation.

3.2.2 Software Environments

The project’s software infrastructure is built upon a Python-centric development environment, utilizing key libraries for image processing and data management.

3.2.2.1 Operation System and Development Environment

The entire software stack was developed and executed within Windows 11 Enterprise, providing a stable and feature-rich platform. Visual Studio Code was utilized as the integrated development environment (IDE), offering powerful debugging tools and seamless integration with Python.

3.2.2.2 Programming Languages and Core Libraries

The core logic of the system is implemented in Python 3.12. The Python environment is managed using Anaconda to ensure consistent package versions and reproducibility across development iterations. Essential libraries include OpenCV for image processing and computer vision tasks, NumPy for numerical operations, and Pandas for data manipulation and analysis.

3.2.2.3 Version Control

Version control for the project’s codebase was managed using Git, with Bitbucket serving as the remote repository platform.

3.2.3 High-Level System Architecture

The system is structured as a modular pipeline consisting of three primary components: the input module, the calibration module and the process module. This architecture was designed to decouple data acquisition, processing and evaluation, thereby improving maintainability, scalability and flexibility.

Figure 3.1 illustrates the overall system architecture and interaction between the modules.

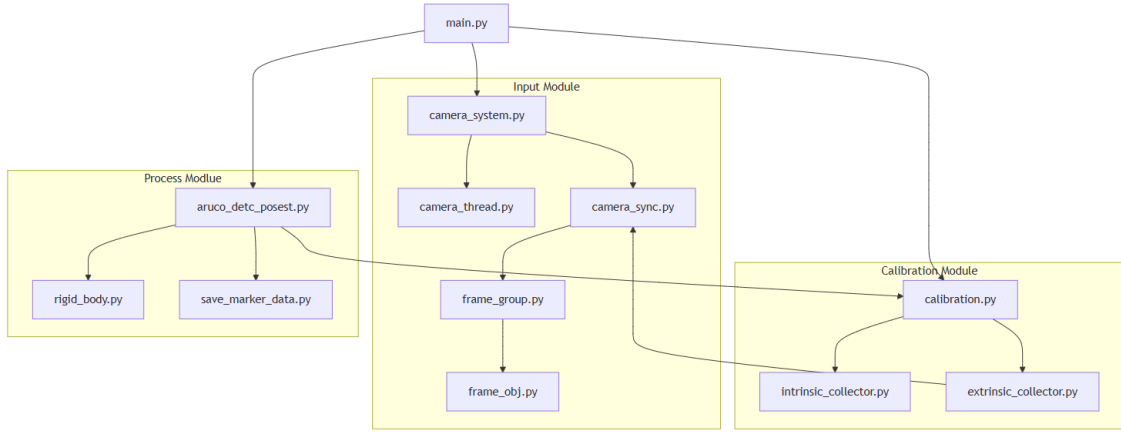


Figure 3.1: Illustration of high level system architecture.

Explanation of entities illustrated in figure 3.1:

main.py - Pipeline orchestrator, asks user which phases to run and runs all phases in order.

camera_system.py - Starts and stops both camera and sync threads. Remaps OS camera indices to positional indices 0, 1, 2.

camera_thread.py - One thread per physical camera, reads frames from hardware and pushes them to the sync manager.

camera_sync.py - Matches frames from all cameras within a preconfigured time window into synchronized frame groups.

frame_group.py - Data container holding one frame from each camera and a shared timestamp.

frame_obj.py - Data container for a single frame: image, frame ID, camera index, timestamp.

calibration.py - Runs intrinsic and extrinsic camera calibration. Auto-retries cameras that fail the quality threshold. Saves results to .npy files.

intrinsic_collector.py - Collects chequerboard frames per camera for intrinsic calibration. Only saves frames where board is detected.

extrinsic_collector.py - Collects synchronized frame groups where all cameras see the chequer-board simultaneously.

aruco_detc_posest.py - Detects ArUco markers in each camera, triangulates three-dimensional positions, estimates world-frame rotations, and drives the pose capture loop.

rigid_body.py - Calibrates rigid body templates, reconstructs occluded markers and computes rigid body pose per frame.

save_marker_data.py - Writes marker positions, rotations, and rigid body poses to the output file.

3.2.3.1 Input Module

The input module is responsible for acquiring data from multiple cameras and preparing it for further processing. It manages camera sources and organizes captured data into structured units suitable for multi-view analysis.

3.2.3.2 Calibration Module

The calibration module performs both the intrinsic and extrinsic calibration of the cameras, providing parameters to accurately perform functionality of the procession module.

3.2.3.3 Processing Module

The processing module performs the core functionality of the system. It takes the prepared input data and applies object detection, three-dimensional reconstruction, three-dimensional object reconstruction and trajectory tracking to estimate the motion of the observed object in three-dimensional space.

3.3 System Workflow

This section describes the operational workflow of the system, outlining how data moves through the implemented modules during execution. The workflow illustrates the sequence of steps from data acquisition to processing and the generation of results. By presenting the system flow in a structured manner, this section clarifies how the individual components interact to achieve the overall functionality of the system.

3.3.1 Overall Workflow

The system is designed as a modular architecture in which data flows sequentially from the input module to the processing module during normal operation. The input module acquires visual data from multiple cameras and organizes the captured frames into synchronized frame groups. These frame groups represent temporally aligned multi-view observations and serve as input to the processing module.

The processing module performs object detection, three-dimensional reconstruction, and trajectory tracking based on the synchronized data. This stage produces spatial estimates that describe the motion of the observed object in three-dimensional space.

In addition to the operational pipeline, a separate test module is implemented to evaluate system performance. This module can be executed independently and is used to verify correctness, stability, and compliance with the defined requirements. By separating validation from the core processing workflow, the system maintains a clear distinction between functional execution and evaluation.

3.3.2 Data Acquisition

The data acquisition process is responsible for capturing and structuring visual data from multiple camera sources. Each camera is represented by a software abstraction that encapsulates its properties and configuration, enabling flexible handling of different devices within the system.

Captured data is organized into individual frames, where each frame represents a discrete time instance from a specific camera. These frames form the fundamental units of data used throughout the system and serve as the basis for subsequent multi-view processing.

3.4 Implementation Details

This section details the technical implementation of the system’s core components, translating the theoretical principles outlined in Chapter 2 into a functional multi-camera motion capture pipeline. Each subsection addresses a distinct stage of the pipeline, from establishing the geometric properties of the cameras and aligning their recordings in time, to detecting markers in each frame and reconstructing their positions in three-dimensional space.

3.4.1 Camera Calibration

Following the principles of camera calibration outlined in section 2.3, a two-stage calibration process was conducted for each camera within the system. This ensured that the internal properties of each camera and their precise real-world relationships to one another were accurately established.

For every individual camera, its unique internal properties, including distortions and focal properties, were determined. This was done by capturing multiple images of a chequerboard with a predefined size and number of squares, taking pictures at different angles and depths. Using these images, the relationship between two-dimensional pixel points in each frame and their known three-dimensional real-world coordinates on the chequerboard allowed for the calculation of the camera’s intrinsic parameters.

With the intrinsic properties of each camera understood, the next step was to define their positions and orientations within a shared three-dimensional coordinate system. This was done by capturing a series of time-synchronized frames between the cameras, all containing the same chequerboard as before, to establish camera relations. This series of relations was then averaged out over the amount of synchronized frames used and then used to calculate the extrinsic camera values connecting all the cameras to the real world coordinate system.

This comprehensive calibration procedure, utilizing the calibration pattern method discussed in Section 2.3, yielded a precise mathematical model for each camera, which is critical for accurate three-dimensional reconstruction and motion capture.

3.4.2 Data Synchronization

To enable accurate multi-view reconstruction, frames from different cameras must be temporally aligned. This is achieved through a synchronization mechanism that aligns all camera streams to a common system time.

Frames are grouped into synchronized sets, referred to as frame groups, where each group contains one frame from each camera corresponding to approximately the same time instance. The grouping process is based on frame timestamps, ensuring that all frames within a group fall within a tolerance

of 0.02 seconds. This tolerance is chosen to balance synchronization precision with robustness to minor timing discrepancies between cameras. The resulting frame groups provide consistent multi-view observations and serve as the primary input to the processing module.

For evaluation and to address varying operational requirements, two distinct synchronization approaches were implemented:

- **Real-time Synchronization:** This method continuously processes incoming camera frames, forming synchronized groups as the data arrives. It utilizes a buffered approach, temporarily storing a limited number of frames to facilitate optimal temporal matching within the specified tolerance. The primary objective is to provide synchronized data with minimal latency, making it suitable for live feedback and interactive applications.
- **Post-processing Synchronization:** In contrast, this approach operates on the entire recorded dataset after all frames have been acquired. It performs a global optimization across the full capture duration to form synchronized frame groups. The algorithm employed for post-processing was a greedy approach, prioritizing the closest temporal matches within the 0.02-second tolerance. While potentially allowing for a more exhaustive search, this method inherently introduces latency as it requires the complete dataset before outputting synchronized data.

The suitability and comparative performance of these two synchronization strategies are further discussed and analysed in Section 4.2.

3.4.3 Object Detection

With synchronized and calibrated camera feeds established, the next crucial step was to identify the objects of interest. ArUco markers, as introduced in Section 2.7, were attached to the object of interest to serve as robust visual tags. For each incoming frame, the image was first converted to greyscale to simplify processing, aligning with practices discussed in Section 2.6 for efficient feature extraction. Then, OpenCV's `detectMarkers()` function was used to find all visible ArUco markers. This function efficiently extracts the two-dimensional corner locations of each marker and its unique ID. By combining these two-dimensional observations with the camera's internal characteristics, the system could quickly determine the position of the marker's four corners and the rotation of the marker. This vital positional information was then ready for the next stage of motion analysis.

3.4.4 Three-Dimensional Reconstruction

As mentioned in Section 2.4, a single camera cannot accurately determine the depth of an object in an image. Based on this limitation, using positional data from a single camera was not considered a viable solution.

By instead using three cameras in the setup, it is possible to triangulate the object's position using the method described in Section 2.4. This allows for a position estimate that better corresponds to the real-world position, using the OpenCV function `triangulatePoints()`.

The use of three cameras also provides the advantage of estimating the marker's rotation using the camera with the best view. The best view is defined as the camera in which the marker's projected area is maximized, as this view is least affected by perspective distortion. This approach ensures that the obtained rotation values are as close as possible to the true real-world values achievable with this camera setup.

3.4.5 Trajectory Tracking

Using the ArUco marker ID described in Subchapter 2.7, it is possible to determine the position of each marker. By combining multiple detections of the markers over time, a trajectory can be constructed that describes the marker’s movement throughout the observation period.

3.5 Validation Methodology

The validation of the multi-camera motion capture system was conducted using a controlled experimental setup configured to evaluate its performance against the defined requirements in section 3.1.2. This section details the specific configuration used for validation, the metrics employed, and the step-by-step procedure followed.

3.5.1 Experimental Setup

The system’s core hardware components, as described in Section 3.2.1, were utilized, with specific configurations for the validation process:

3.5.1.1 Hardware Configuration

This section outlines the specific hardware components and their configurations employed for the system’s validation, covering cameras, computing platform, connectivity, markers, calibration, and physical setup. The following hardware was used:

- **Cameras:** Three Logitech MX BRIO webcams were deployed. For validation purposes, they were configured to capture at a resolution of 640x480 pixels and approximately 30 frames per second. This due to the lack if appropriate drivers which is revisited in section 6.3.2.
- **Computer:** The computational backbone for all image processing and algorithmic tasks was provided by the laptop specified in Section 3.2.1.1 (Intel i5-1145G7 2.6 GHz CPU, integrated Intel Iris Xe Graphics GPU, and 16 GB of RAM).
- **Data Cables:** Two high-speed USB 3.0 C to A cables, as well as one USB C 3.0 cable, were used to connect each camera to the computer. These cables ensured reliable, low-latency data transfer, minimizing potential bottlenecks during simultaneous multi-camera streaming.
- **ArUco Markers:** ArUco markers from the DICT_6X6_250 library, serving as the Motion Capture Object (Section 3.1.2, B.6.1.5), were printed to a physical size of 53 mm per side. This specific size was selected to ensure robust detection across the typical working distances of the camera setup.
- **Calibration Pattern:** A 7x6 checkerboard pattern with 28 mm squares was used for all camera calibration procedures, following the methodology detailed in Section 3.4.1.
- **Physical Arrangement:** The three cameras were rigidly mounted on a stable frame, forming a fixed triangular configuration. This arrangement created a common capture volume approximately 40 cm wide, 40 cm deep and 40 cm high, within which the test object was moved. This fixed setup ensured consistent extrinsic parameters throughout the validation.

3.5.1.2 Software Environment

The validation utilized the system’s software infrastructure, running within the Python 3.12 environment on Windows 11 Enterprise, as outlined in Section 3.2.2. The core libraries used included:

- **OpenCV:** Crucial for camera interfacing, ArUco marker detection (Section 3.4.3), intrinsic and extrinsic camera calibration (Section 3.4.1), image undistortion, and three-dimensional point triangulation (Section 3.4.4). The validation leverages OpenCV’s robust and widely-validated implementations for these standard computer vision tasks.
- **NumPy:** Employed for efficient numerical computations and array manipulations, particularly in processing image data and mathematical transformations.
- **Pandas:** Used for structured data management, analysis of timestamps, and organization of tracking information from multiple cameras and frames.
- **Matplotlib:** Utilized for generating static visualizations of performance metrics, such as histograms and time-series plots.
- **Seaborn:** Provided enhanced statistical data visualization capabilities, complementing Matplotlib for clearer presentation of results.
- **VSCode:** Used as the main Integrated Development Environment (IDE) for coding, debugging, and project management.
- **SciPy:** Employed for advanced scientific computing, including optimization, signal processing, and spatial algorithms.

3.5.1.3 Test Parameters

To ensure a consistent and repeatable evaluation reflecting operational conditions and addressing core requirements, the following parameters were strictly adhered to:

- **Total Frame Groups for Analysis:** The validation was conducted using two distinct recordings. For the synchronization analysis in Section 4.2, a dedicated dataset of 2000 synchronized frame groups was collected. For all subsequent pipeline evaluations, including object detection, reconstruction, and trajectory tracking, a separate continuous recording session was performed, yielding 1615 frames. This large sample size ensured statistical significance for performance metrics, particularly for the synchronization analysis.
- **Recording Duration:** The duration of the recording sessions was tailored to their specific dataset requirements. The synchronization test, designed to collect 2000 frame groups at 30 fps, lasted approximately 70 seconds. The separate pipeline evaluation recording, which yielded 1615 frames, had a duration of approximately 54 seconds. Both durations were deemed sufficient to capture representative motion dynamics.
- **Synchronization Threshold:** The critical requirement for synchronization accuracy requirement B.6.2.3 in section 3.1.2 of a maximum 20 ms time spread between frames in a group served as the primary performance target.
- **Calibration Data Quantity:** For intrinsic calibration (Section 3.4.1), 30 distinct chequerboard images per camera were captured. For extrinsic calibration, a total of 50 synchronized multi-camera views of the chequerboard were recorded. To ensure that the images captured a broader range of movement and perspective, every tenth frame was selected from a continuous sequence containing the chequerboard for both intrinsic and extrinsic calibration. This quantity and selection strategy aligns with best practices for achieving robust and sub-pixel re-projection accuracy with OpenCV.

3.5.2 Performance Metrics

To thoroughly evaluate the multi-camera three dimensional tracking system, a series of quantitative metrics were established, targeting key operational aspects. This section details the specific performance indicators used to assess the system’s fundamental components, including the temporal alignment of camera feeds, the reliability of object detection, the accuracy of camera calibration, the precision of three dimensional reconstruction and the precision of the trajectory tracking. These metrics collectively validate the system’s capacity to deliver consistent, accurate, and reliable spatial data.

3.5.2.1 Camera Synchronization Metrics

Accurate camera synchronization is vital for multi-camera systems, especially in applications requiring precise three dimensional reconstruction and motion capture. As detailed in the calibration process (Section 2.3) establishing the precise, spatial relationships between cameras relies heavily on time-synchronized frames. Therefore, assessing the temporal alignment of frames captured by each camera is critical for validating the system’s ability to provide reliable data for downstream tasks.

For this validation, camera synchronization performance is quantitatively evaluated using the following metrics:

- **Time Spread:** Quantifies the temporal difference between the earliest and latest timestamp among all frames within a designated synchronized frame group, measured in milliseconds. This directly measures the instantaneous synchronization error.
- **Mean Time Spread:** Represents the average Time Spread calculated across all 2000 recorded frame groups. This serves as the primary indicator of the system’s overall synchronization quality.
- **Maximum Time Spread:** Identifies the largest Time Spread observed during the entire recording session. This worst-case scenario is directly compared against the stringent 20 ms threshold requirement (B.6.2.3) to ascertain critical failure points.
- **Standard Deviation:** Measures the variability and consistency of the Time Spread across the recorded frame groups, indicating the stability of the synchronization performance over time.
- **Threshold Compliance Rate:** Calculates the percentage of frame groups that successfully meet the specified 0.02-second synchronization requirement. This metric provides a clear indication of how often the system achieves acceptable synchronization, computed as (compliant groups / total groups).
- **Temporal Consistency:** Evaluated by applying a moving average analysis with a 50-group window to the Time Spread data. This assesses the stability and reliability of the synchronization performance throughout the entire recording duration, identifying any drift or intermittent issues.

3.5.2.2 Camera Calibration Metrics

The two-stage camera calibration (Section 3.4.1) establishes intrinsic camera properties and their precise extrinsic relationships, critical for three dimensional reconstruction. Its accuracy is validated using the following metrics:

- **Intrinsic Re-projection Error:** This metric quantifies intrinsic calibration accuracy as the mean pixel distance between detected and reprojected chequerboard corners. A value of less than 1.0 pixel is the target threshold, indicating effective correction for lens distortions and accurate focal properties.
- **Extrinsic Baseline Error:** This metric assesses the accuracy of the extrinsic calibration by measuring the discrepancy between the calculated Euclidean distances between camera optical centres and their physically measured counterparts. Agreement within a ± 30 mm tolerance confirms the accurate representation of the camera rig's geometry.

3.5.2.3 Object Tracking Metrics

Following the establishment of synchronized and calibrated camera feeds, the system's ability to reliably detect and track objects using ArUco markers, as detailed in Section 3.4.3, is critical for accurate three dimensional reconstruction and motion analysis. The object detection process, involving greyscale conversion and the `detectMarkers()` function from OpenCV, aims to efficiently identify marker corners and IDs. To validate the effectiveness of this detection mechanism, the following metrics are employed:

- **Detection Rate:** This metric quantifies the reliability of ArUco marker detection across the camera system. It is calculated as the percentage of total frames where at least one ArUco marker was successfully identified by at least two cameras, enabling three-dimensional reconstruction: (frames where at least two cameras detected a marker / total frames). A high detection rate indicates robust performance of the detection algorithm under various viewing conditions for the camera system.
- **Detection Robustness:** This metric evaluates the reliability of ArUco marker detection across the system by characterizing its performance limitations. It quantifies the percentage of successful detections under controlled variations of marker-to-camera distance, orientation, and physical marker size. This provides insight into the system's capacity for continuous tracking within its operational volume.

3.5.2.4 Three-Dimensional Reconstruction Metrics

The system's multi-camera triangulation (in section 3.4.4) for three dimensional position determination and optimized rotation estimation is critical. Its performance is validated using the following metrics:

- **Triangulation Consistency:** This metric quantifies the precision of three dimensional position estimates. It measures the standard deviation of a marker's three dimensional position when triangulated independently from all valid camera pairs. Lower values signify higher precision and reliability.
- **Re-projection Error:** This metric evaluates the end-to-end accuracy of the entire calibration and reconstruction pipeline. It is the mean pixel error when reconstructed three dimensional points are projected back onto the two dimensional image planes of all detecting cameras. A target threshold of less than 2 pixels confirms consistency between three dimensional estimates and original two dimensional observations.

3.5.2.5 Data Stream Performance Metrics

These metrics evaluate the temporal performance of the foundational data stream provided by the pipeline. They quantify the consistency and stability of the incoming synchronized frame groups, which is a critical prerequisite for generating smooth and reliable final trajectories.

- **Temporal Continuity:** This metric quantifies how consistently the system delivers data around its expected frame rate. It is used to identify the frequency and magnitude of significant data gaps or processing delays.
- **Performance Stability Over Time:** This metric assesses whether the system’s data throughput degrades or experiences drift during a continuous recording session. It is used to detect accumulating latency or other systematic timing issues.

3.5.2.6 Trajectory Tracking Metrics

Despite the absence of external ground truth, these metrics evaluate the internal precision and continuity of the final three-dimensional trajectories, reflecting the collective performance of prior system components:

- **Trajectory Smoothness:** This metric quantifies the short-term, high-frequency fluctuations in reconstructed three-dimensional trajectories by analysing the standard deviation of frame-to-frame positional changes. Only frame pairs where both frames contained a valid detection were included. Gaps caused by temporary marker occlusion were excluded to avoid artificially inflating the metric. Lower jitter indicates a more precise and internally consistent motion path.
- **Trajectory Coverage:** This metric measures the percentage of frames where a successful three-dimensional position was obtained for a specific marker, relative to the total number of frames in the recording.

3.5.3 Validation Procedure

The validation of the multi-camera motion capture system was conducted through a structured execution plan, encompassing data acquisition, pre-processing, and systematic evaluation of each performance metric detailed in Section 3.5.2.

To ensure a consistent and statistically significant dataset for evaluating the system’s performance, the following preparatory steps and data collection procedures were performed:

1. **Test object preparation:** ArUco markers from the DICT 6x6 250 library, each 53 mm per side, were affixed to a flat A4 sheet of paper in a 3×2 grid arrangement, yielding six uniquely identifiable markers. The sheet was held rigidly by hand and moved freely through the capture volume, ensuring continuous visibility of multiple markers by at least two cameras during typical motions.
2. **Motion Trajectory:** The marked test object was moved within the predefined capture volume of approximately $40 \times 40 \times 40$ cm following a repeatable figure-of-eight trajectory. This motion was specifically designed to traverse the entire volume and induce diverse marker orientations and distances relative to the cameras.
3. **Data Recording:** Two distinct recordings were performed to create datasets tailored for different validation purposes.
 - For the specific evaluation of the synchronization algorithms in Section 4.2, a dedicated recording of 2000 synchronized frame groups was collected.

- For the end-to-end pipeline evaluation, covering object detection, reconstruction, and trajectory analysis in Sections 4.3 through 4.6, a separate, continuous recording of the moving marker board was performed, yielding a dataset of 1615 raw frames.

3.5.3.1 Camera Synchronization Metrics

Synchronization performance for both Real-time and Post-processing approaches explained in section 3.4.2 was evaluated using an identical dataset of 2000 synchronized frame groups from three cameras, obtained from the general data collection. The evaluation procedure for each specific metric was as follows:

1. Time Spread:

- **Procedure:** For each of the 2000 frame groups processed by both synchronization methods, the precise capture timestamp for each of the three camera frames was extracted. The Time Spread was then computed as the difference between the maximum and minimum timestamps among these three synchronized frames within that group.
- **Metrics Derived:** This computation directly yielded the individual **Time Spread** for each frame group, serving as the basis for subsequent aggregate metrics.

2. Mean Time Spread, Maximum Time Spread, and Standard Deviation:

- **Procedure:** The individual Time Spread values computed for all 2000 frame groups, independently for both Real-time and Post-processing Synchronization methods, were collected and statistically analysed.
- **Metrics Derived:** The **Mean Time Spread** and **Maximum Time Spread** were calculated as the arithmetic average and the single largest observed value, respectively. The **Standard Deviation** of the Time Spread values was also computed to quantify the variability and consistency of the synchronization performance.

3. Threshold Compliance Rate:

- **Procedure:** The Time Spread values for all 2000 frame groups were compared against the specified 20 ms synchronisation tolerance, 0.02 seconds as per Section 3.5.1.3.
- **Metrics Derived:** The **Threshold Compliance Rate** was determined by counting the percentage of frame groups whose individual Time Spread was less than, or equal to, 0.02 seconds, expressed as a percentage of the total 2000 frame groups. This also informed the distribution analysis by identifying groups within/outside the threshold.

4. Temporal Consistency:

- **Procedure:** A 50-frame group moving average filter was applied to the sequence of individual Time Spread values over the entire recording period for both synchronisation methods.
- **Metrics Derived:** The resulting smoothed curve provided insight into **Temporal Consistency**, allowing for visual and quantitative assessment of stability, systematic drift, or intermittent performance degradation over time.

5. Overall Comparative Evaluation:

- **Procedure:** All calculated metrics for Real-time and Post-processing Synchronization methods were directly compared.
- **Metrics Derived:** This facilitated a comprehensive comparative analysis of their achieved temporal precision, consistency, and adherence to the 20 ms tolerance, leading to conclusions about their relative effectiveness.

3.5.3.2 Camera Calibration Metrics

The accuracy of the two-stage camera calibration process (Section 2.3) was validated through the following procedures, assessing both intrinsic camera properties and their extrinsic relationships:

1. Intrinsic Re-projection Error:

- **Data Acquisition:** For each of the three Logitech MX BRIO cameras, a distinct set of 30 images of the 7×6 chequerboard pattern with 28 mm side length squares was captured. The chequerboard was presented at various positions, orientations, and depths within the camera’s field of view to ensure comprehensive coverage of the image plane and diverse perspective.
- **Procedure:** OpenCV’s calibration functions were then applied independently to each camera’s acquired image set to compute its unique intrinsic parameters (focal lengths f_x, f_y , principal point c_x, c_y , and distortion coefficients k_1, k_2, k_3, p_1, p_2).
- **Metrics Derived:** The **Intrinsic Re-projection Error** for each camera was calculated and recorded as the mean pixel distance between the detected chequerboard corners and their reprojected positions using the calibrated intrinsic model, serving as a primary indicator of intrinsic calibration accuracy.

2. Extrinsic Baseline Error:

- **Data Acquisition:** A separate continuous video sequence was recorded by all three cameras simultaneously, capturing the 7×6 chequerboard pattern as it was moved through the shared capture volume. From this sequence, a set of 50 time-synchronized multi-camera frame groups were used for extrinsic parameter determination.
- **Procedure:** These synchronized multi-view observations of the chequerboard were used to determine the extrinsic parameters of each camera relative to a common world coordinate system, following the methodology described in Section 2.3.
- **Metrics Derived:** The **Extrinsic Baseline Error** was calculated by measuring the Euclidean distances between the optical centres of the calibrated cameras and comparing them against precise physical measurements taken directly from the rigidly-mounted camera frame. Agreement within a ± 30 mm tolerance was considered acceptable.

3.5.3.3 Object Tracking Metrics

The system’s object tracking capabilities were evaluated by assessing both the general detection reliability for each camera and the system-wide robustness of ArUco marker detection within the operational volume.

1. Detection Rate per Camera:

- **Procedure:** For each frame within the 1615-frame test recording, ArUco markers were detected using OpenCV’s `detectMarkers()` function as detailed in Section 3.4.3.

The two-dimensional corner coordinates and unique IDs for each detected marker were recorded.

- **Metrics Derived:** For each individual camera, the total number of frames where at least one ArUco marker was successfully identified was counted. The **Detection Rate per Camera** was then calculated as (frames with detections / total frames).

2. Detection Robustness:

- **Procedure:** The frequency with which each individual ArUco marker ID was successfully detected by at least one camera was quantified across all 1615 frames of the test recording. This measured the continuity of each marker’s presence within the system’s combined field of view during the entire motion sequence.
- **Metrics Derived:** The **Detection Robustness** was expressed as the average percentage of frames each marker ID was visible to at least one camera. The minimum observed visibility percentage for any single marker ID provided an indication of worst-case detection continuity within the operational volume. This highlights the system’s capacity to maintain awareness of individual markers under dynamic conditions.

3.5.3.4 Three-Dimensional Reconstruction Metrics

The system’s three-dimensional reconstruction performance was evaluated by assessing the precision of marker position estimates, the reliability of successful reconstructions, and the overall end-to-end accuracy of the pipeline.

1. Triangulation Consistency:

- **Procedure:** For each individual marker visible in all three cameras within a single frame group, three independent three-dimensional position estimates were generated. This was achieved by triangulating the marker’s position from all possible unique pairs of cameras.
- **Metrics Derived:** The standard deviation of these three three-dimensional estimates was calculated for each such marker in each relevant frame group. The overall mean of these standard deviations, averaged across all relevant markers and frames, served as the **Triangulation Consistency** metric, quantifying the precision of the three-dimensional position estimates.

2. Re-projection Error:

- **Procedure:** For every successfully reconstructed three-dimensional point, it was projected back onto the two-dimensional image plane of each camera that originally detected that marker.
- **Metrics Derived:** The Euclidean distance in pixels between each reprojected two-dimensional point and its corresponding original two-dimensional detected corner position was calculated. The mean of these errors across all re-projections served as the **Re-projection Error**, validating the end-to-end accuracy of the calibration and reconstruction pipeline.

3.5.3.5 Data Stream Performance Metrics

The performance of the foundational data stream, which provides the synchronized frame groups for all subsequent processing, was evaluated to quantify its temporal consistency and stability. The analysis was conducted on the successfully reconstructed frame groups derived from the 1615-frame test recording.

1. Temporal Continuity:

- **Procedure:** The time difference between the timestamps of every consecutive frame group was calculated across the entire recording session. A histogram was then generated from this set of time differences to visualize the distribution of inter-frame intervals.
- **Metrics Derived:** The **Temporal Continuity** was quantified by calculating the percentage of frame groups falling within key time interval brackets. This provides a statistical measure of how consistently the system delivers data around the expected frame rate and identifies the frequency of significant delays or frame drops.

2. Performance Stability Over Time:

- **Procedure:** The individual time difference for each successfully reconstructed frame group was plotted sequentially against its Group ID. A 50-frame group moving average filter was then applied to this sequence to smooth short-term fluctuations and reveal the underlying performance trend over the duration of the recording.
- **Metrics Derived:** The resulting smoothed curve provides a visual assessment of **Performance Stability**. A flat trend indicates high stability with no systematic drift or performance degradation. The overall average of the time differences was also calculated to confirm the effective frame rate of the system during the recording.

3.5.3.6 Trajectory Tracking Metrics

The final output quality of the system is assessed by analysing the generated three-dimensional trajectories of the tracked ArUco markers. These metrics evaluate the internal precision and continuity of the motion paths, reflecting the collective performance of all preceding system modules. The trajectories are formed by linking the successfully reconstructed three-dimensional positions of individual markers across consecutive frames using their unique IDs, as described in Section 3.4.5.

1. Trajectory Smoothness:

- **Procedure:** For each individual trajectory, the frame-to-frame positional change is calculated by measuring the Euclidean distance between the three-dimensional position in consecutive frames. Only frame pairs where both frames contained a valid detection were included. Gaps caused by temporary marker occlusion were excluded to avoid artificially inflating the metric.
- **Metrics Derived:** The standard deviation of these positional changes is computed per marker trajectory. Per-marker values are reported individually, and an overall mean across all markers serves as the final **Trajectory Smoothness** metric. A lower value indicates a more precise and internally consistent motion path.

2. Trajectory Coverage:

- **Procedure:** For each unique ArUco marker, the total number of frames in which a successful three-dimensional position was reconstructed is counted and compared against the total number of frames in the recording.

- **Metrics Derived:** The **Trajectory Coverage** is calculated as the percentage of the total recording of frames where a successful three-dimensional position was obtained for a specific marker. Per-marker values are reported individually, with the mean across all markers providing an overall system-level completeness metric.

Chapter 4

Initial System Performance and Analysis

This chapter evaluates the system’s performance using data from two distinct recordings. The first, a 2000-frame-group dataset, is used exclusively to validate the synchronization algorithms in Section 4.2. A second recording, consisting of 1615 raw frames, is used for all subsequent analyses of the core motion capture pipeline, from object detection through trajectory tracking.

4.1 Calibration Evaluation

As established in Section 3.4.1, a two-stage process was used to calibrate the intrinsic and extrinsic parameters of the webcams. The initial intrinsic stage involved capturing 30 images of a chequerboard from diverse viewpoints to model lens distortions, while the subsequent extrinsic stage utilized 50 images. The chequerboard’s known square size was used throughout both stages to gather depth and position estimations.

Intrinsic calibration was performed first, enabling the use of the now-calibrated cameras for the subsequent extrinsic parameter estimation. This sequential approach is considered best practice for effectively calibrating cameras and mitigating the propagation of errors between stages.

4.1.1 Calibration Accuracy

Re-projection error serves as the primary metric for intrinsic calibration quality. As detailed in Section 3.5.3.2, this error represents the pixel distance between the detected and reprojected positions of the chequerboard corners. Ideally, according to the OpenCV documentation, this value should be sub-1 pixel, with smaller values representing higher quality calibration. The results show consistently stable intrinsic parameters, with re-projection errors clustering around the 0.9-pixel mark. As expected, this value degraded with poorly executed calibrations and improved when optimal chequerboard coverage was achieved.

The extrinsic baseline error serves as the primary quality metric for evaluating extrinsic calibration performance. As detailed in Section 3.5.3.2, this metric is derived by comparing the calculated Euclidean distances between camera optical centres with their physically measured counterparts. Through multiple runs on a standardised test rig with predetermined measurements, it was con-

cluded that, while not perfect, the calculated camera positions were consistently accurate to within approximately ± 30 mm of their true locations. This consistency was confirmed by rerunning the same test multiple times and observing that the results remained stable.

4.1.2 Methodological Implications

The validation confirms the calibration pipeline is fit for purpose, with the achieved extrinsic spatial accuracy of approximately ± 30 mm meeting the precision requirements of this thesis. This result establishes a critical performance baseline. It follows that improved calibration would directly translate to higher fidelity in the final motion capture data.

The primary limitation in achieving superior calibration was the operator-dependent quality of the image acquisition. This suggests that future work could focus on a more guided capture process to ensure more consistent data.

Ultimately, the calibration approach is validated as appropriate for the project’s scope, proving that standard computer vision methods can be successfully applied within a low-cost motion capture system.

4.2 Data Synchronization Effectiveness

This section analyses the performance of two distinct synchronization methods implemented to align frames from multiple cameras in time: Real-time, and Post-processing Synchronization. Both methods were designed to group frames from multiple cameras such that all frames within a synchronized set fall within a threshold of 20 ms maximum time spread tolerance between the earliest and latest image. An identical dataset consisting of a recording of 2000 synchronized frame groups from three cameras was used for evaluation, ensuring a consistent benchmark.

4.2.1 Temporal Accuracy

Quantitative analysis showed a strong similarity in the temporal accuracy achieved by both synchronization methods. Real-time Synchronization processed 2000 groups, achieving a mean time spread of 13.40 ms and a maximum of 19.99 ms. Similarly, Post-processing Synchronization yielded 2000 groups, with a mean time spread of 13.28 ms and a maximum of 19.89 ms. This minimal 0.12 ms (0.9%) difference in mean time spread indicates comparable temporal precision between the two methods. Both approaches consistently maintained an average mean spread of approximately 13.3 ms, well within the 20 ms threshold.

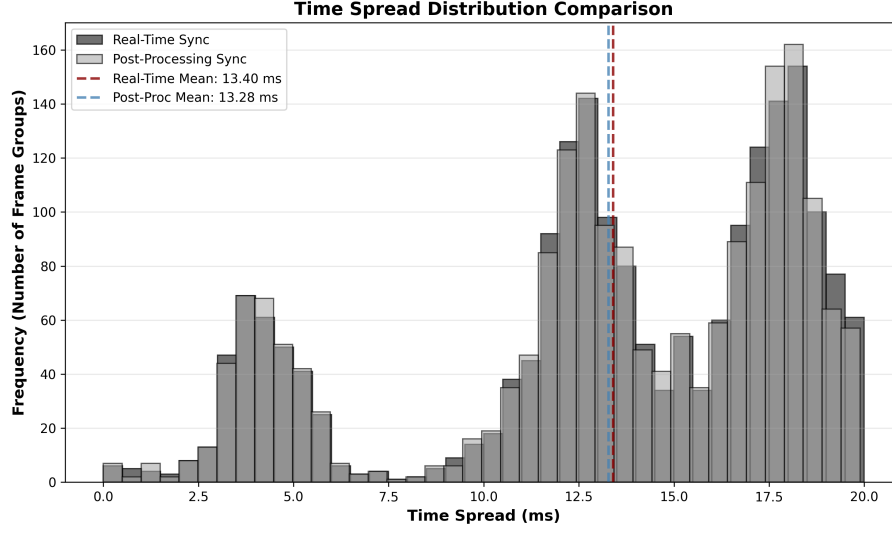


Figure 4.1: Histogram comparing the time spread distribution of Real-time vs. Post-processing Synchronization. Both methods show nearly identical performance, with distributions centred around 13-14 ms and all groups falling under the 20 ms threshold. Dashed lines indicate the highly similar mean values.

Figure 4.1, the Time Spread Distribution Histogram, visually supports these quantitative findings. It shows both synchronization approaches achieving similar quality. Both Real-time and Post-processing produce nearly identical distributions, with most frame groups clustering around 13-14 ms spread. All observed groups remain below the 20 ms threshold. The overlapping bars and close mean values confirm the methods perform more or less equivalently.

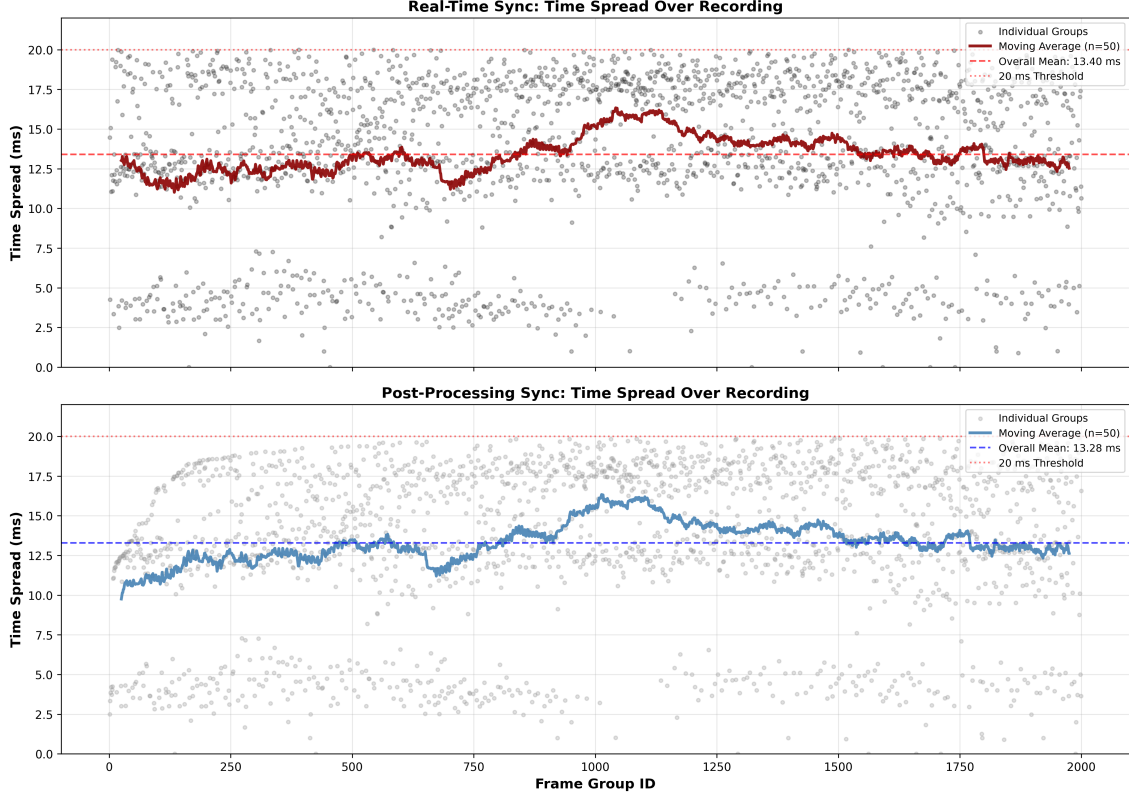


Figure 4.2: Time spread plotted sequentially for Real-time and Post-processing synchronization, with their respective 50-group moving averages shown as bold lines. Both methods remain consistently stable throughout the recording, staying well below the 20 ms threshold (dotted red line) and showing no signs of performance drift.

Figure 4.2 further investigates temporal consistency over the recording period, comparing both methods side-by-side. This plot, which displays the individual data points and a bold 50-group moving average for each method, reveals that synchronization quality remains stable throughout the 2000 frame groups. Neither method shows signs of degrading performance or significant drift in time spread over the duration of the recording. Both moving average lines closely track the overall mean line and consistently stay below the 20 ms threshold. This visual confirmation indicates that both Real-time and Post-processing synchronization maintain a consistent performance level, adapting effectively to any minor temporal variations that may occur during the recording.

4.2.2 Methodological Implications

The remarkable accuracy equivalence within the 20ms threshold, supported by quantitative data and Figures 4.1 and 4.2, indicates that system limitations, specifically webcams, USB bandwidth, internal USB buses, and the operating system, constitute the primary bottleneck, rather than the chosen algorithm. Both methods reliably achieve robust grouping, operating close to the practical software synchronization limit.

Real-time Synchronization’s buffered approach, which employs a 30-frame buffer, demonstrates sufficient capability to consistently match global optimization performance over time, as illustrated

in Figure 4.2.

Practically, the zero-second latency of Real-time Synchronization renders it superior to Post-processing, which introduces a growing overhead. This advantage is particularly significant given the minimal 0.12 ms accuracy difference between the two methods. Real-time Synchronization’s benefits for live feedback, interactive applications, and efficient workflows are thus clear. Post-processing offers limited additional advantage in these scenarios. Therefore, Real-time Synchronization presents a pragmatic solution that effectively balances high accuracy, complete input processing, and zero latency for most motion capture applications.

4.3 Object Detection Performance

As mentioned in Section 3.4.3, an OpenCV function is used for object detection. The detection of a marker in each frame is crucial to the system’s functionality, as described in Section 3.5.3.3. A test was performed to evaluate the Detection Rate, as defined in Section 3.5.3.3.

For this test, the system was configured with three cameras, each positioned 40 cm from the centre along three mutually perpendicular axes. The cameras were angled slightly to ensure a shared field of view. A board containing six ArUco markers was moved in a figure-eight pattern during a separate recording of 1615 frames, distinct from the 2000 frame group dataset used for synchronization analysis. The analysis showed that of the 1615 frames in the recording, 113 frames had no markers that could be detected by at least two cameras simultaneously. This prevents three-dimensional reconstruction for those instances, resulting in a dataset of 1502 viable frame groups for the subsequent trajectory analysis. This represents a successful reconstruction rate of 93.0% at the frame-group level.

Three probable causes were identified for the 113 frames with failed or incomplete observations:

1. **Glare:** Glare on the ArUco board can obscure a marker, preventing it from being recognised.
2. **Motion Blur:** If the object moves at too high a velocity, motion blur can distort the marker’s appearance, making it undetectable.
3. **Marker Position:** The marker’s position, both within a single camera’s frame and within the overall camera system, is a critical factor.
 - (a) **Position within the frame:** Imperfections in the intrinsic camera calibration can leave residual lens distortion, especially near the edges of a frame. A distorted marker in these regions may not be recognised.
 - (b) **Position within the system:** A reconstruction-level failure occurs if a marker is detected by only one camera. Since triangulation requires at least two views, the marker’s three-dimensional position cannot be reconstructed, and it is effectively lost for that frame.

4.4 Three Dimensional Reconstruction

By measuring the triangulation consistency, reconstruction success rate and the re-projection error outlined in subsection 3.5.3.4 over 1615 frames.

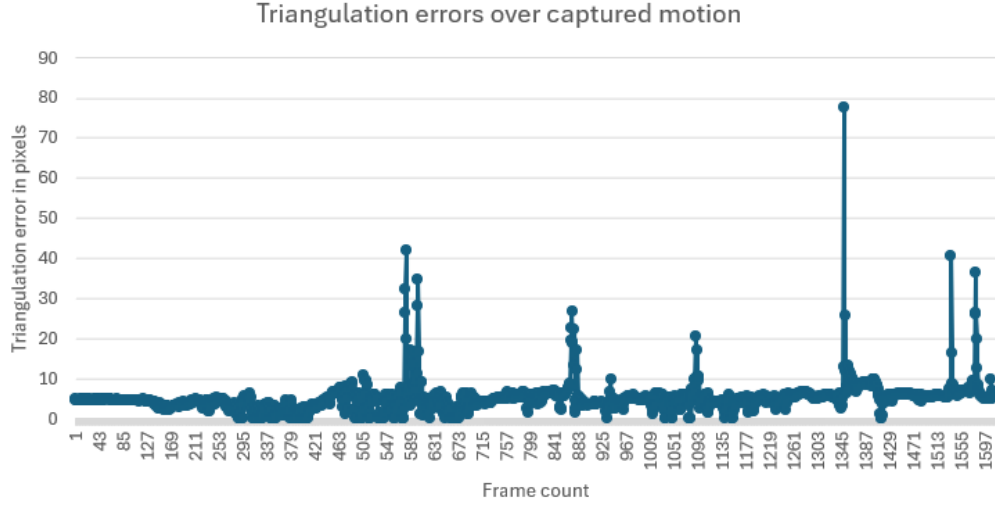


Figure 4.3: Diagram of the pixel error of triangulations made between the three camera pairs.

In Figure 4.3, a relatively stable triangulation error of approximately three to four pixels is observed. By averaging the triangulated positions from the camera pairs, a more consistent estimate of the marker position is obtained.

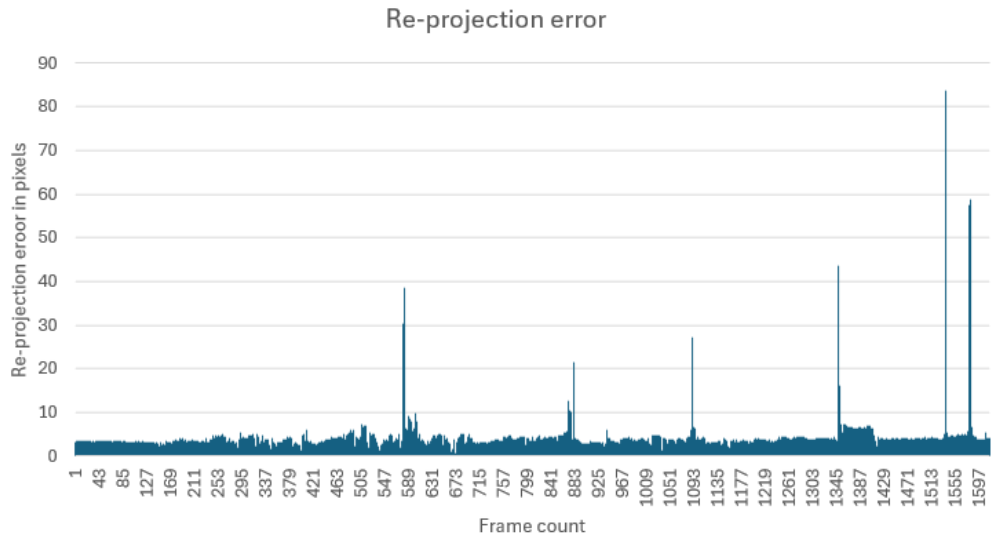


Figure 4.4: Diagram of the mean re-projection error across the three cameras for a series of frames.

In figure 4.4 it is possible to see that the re-projection error stays around two to four pixels, which

also can be corrected by using a mean value. In both figure 4.3 and figure 4.4 it is possible to identify that six data points deviate from the others. Because of the reoccurrence of these six data points, it is likely that these deviations arose because of an error while extracting the real world pose of the markers from the frames. Factors such as illumination, motion-blur and camera angle, could possible alter the camera’s perception of the marker resulting in the data point anomalies visible in both figures.

4.5 Data Stream Performance

This section evaluates the performance of the motion capture pipeline in terms of trajectory smoothness and temporal coverage. Smoothness reflects the consistency of incoming frame group intervals over time, while coverage describes the proportion of frames in which markers were successfully detected and tracked.

4.5.1 Temporal Continuity and Stability

A primary indicator of trajectory tracking performance is data coverage over time.

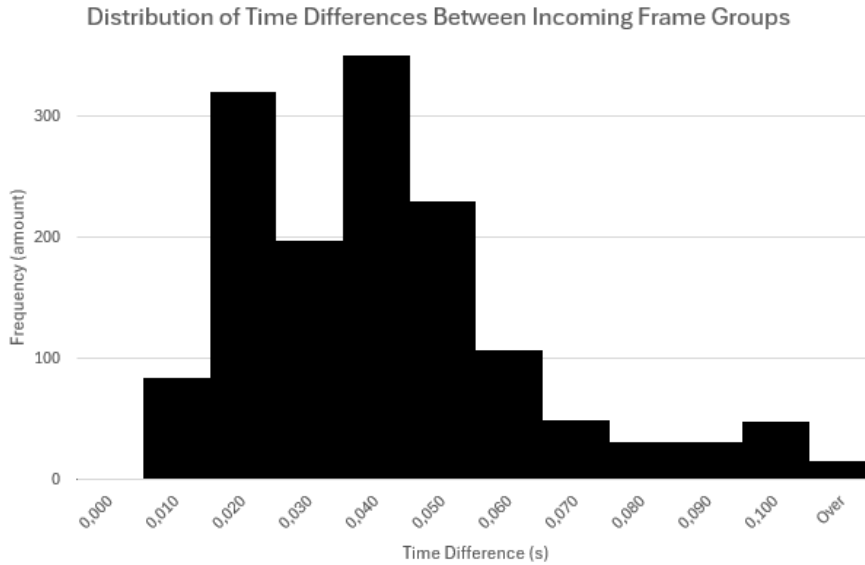


Figure 4.5: Histogram of time differences between consecutive incoming frame groups, showing that 65% of intervals fall within 0.015–0.04 s and 99% fall below 0.1 s.

To evaluate this, the temporal continuity of the data stream was analysed by plotting a histogram of the time differences from the 1502 successfully reconstructed frame groups derived from the test recording between consecutively received frame groups, as shown in Figure 4.5.

The histogram of time differences between consecutive frame groups reveals a highly consistent data throughput. The majority of intervals are tightly clustered, with 65% arriving within a 0.015–0.04 s window. Collectively, 83% of all frame groups are received within the 0.05 s mark, confirming that the data stream aligns with the expected camera output rate. The tail of the distribution, representing 17% of intervals that exceed 0.05 s, corresponds to occasional frame drops

or processing delays, possibly attributable to shared bus bandwidth constraints. Critically, with 99% of all intervals falling below the 0.1 s threshold, the system demonstrates robust performance and reliable data acquisition.

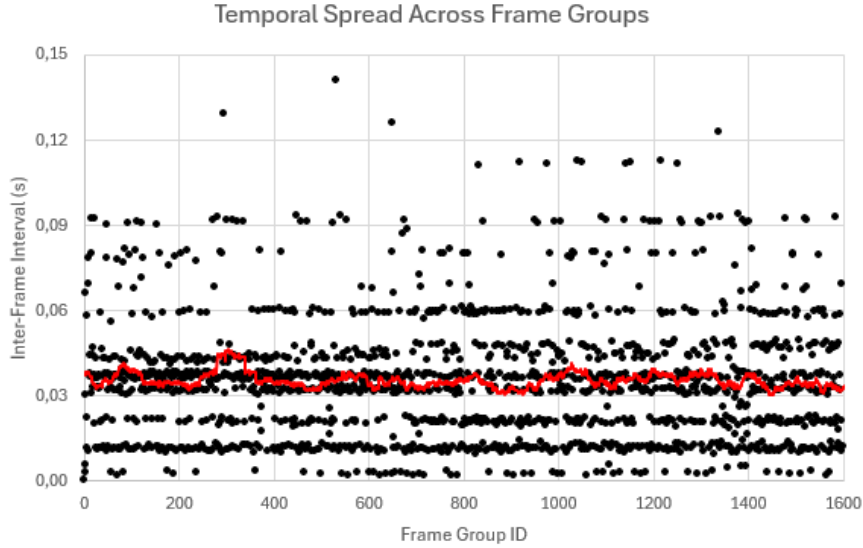


Figure 4.6: Inter-frame interval for each frame group, plotted sequentially by ID. The black dots represent the time difference for individual frame groups, while the red line indicates the 50-group moving average, highlighting the central trend.

In addition to analysing the overall distribution of time intervals, it is crucial to assess the system’s performance stability over the duration of a recording. Figure 4.6 plots the temporal spread for each frame group sequentially to identify any potential degradation or systematic timing issues over time.

The scatter plot, overlaid with a 50-group moving average, demonstrates a highly stable tracking performance. The moving average remains consistently flat, confirming the absence of any drift or performance degradation throughout the recording session. While individual data points exhibit minor deviations, there are no sustained periods of high spread, which indicates that systematic timing issues did not accumulate.

This stability is further validated by the system’s average inter-frame interval, which is approximately 0.035 s. This value corresponds to a data rate of around 28.6 frames per second, aligning with the expected output of a 30 fps camera system experiencing minor, predictable throughput variations.

4.5.2 Data Stream Reliability

The flat moving average sustained throughout the recording confirms that the pipeline introduces no latency accumulation or performance degradation over time. The 83% of intervals falling within 0-0.05 s demonstrates sufficient frame density for smooth trajectory reconstruction, while the outliers could reflect inherent USB hardware constraints rather than pipeline limitations. Overall, the results confirm the system delivers a reliable and consistent data stream suited for the motion capture system.

4.6 Trajectory Tracking Performance

The final part of the motion capture analysis evaluates the actual system output, the reconstructed three-dimensional trajectories. In the absence of an external ground truth, trajectory coverage and smoothness are used as the primary evaluation metrics, assessing the completeness and internal precision of the system’s output respectively.

4.6.1 Trajectory Coverage and Precision

Using the same dataset as the data stream evaluations, the metrics presented in Table 4.1 were derived. Markers 0 to 5 are ArUco markers placed on a flat A4 sheet of paper in a 3×2 grid, forming a single rigid body. All markers therefore undergo identical physical motion throughout the recording.

Table 4.1: Trajectory Tracking Performance Data.

Marker	Total Frames	Valid Frames	Coverage (%)	Mean Step (mm)	Jitter Std (mm)
Marker_0	1615	1525	94.43	7.0227	17.5459
Marker_1	1615	1548	95.85	5.7925	13.0809
Marker_2	1615	1456	90.15	5.0914	13.4114
Marker_3	1615	1465	90.71	6.3142	16.5218
Marker_4	1615	1425	88.23	4.7281	4.8225
Marker_5	1615	1583	98.02	5.1709	6.4934

As seen in Table 4.1, trajectory coverage is high, ranging from 88.23% to 98.02% with a mean of 92.90%. This confirms the pipeline reconstructed a three-dimensional position for most frames. The missing frames are attributed to temporary marker occlusions rather than a systematic pipeline error.

The mean Euclidean step distance is consistent across all markers (4.73 to 7.02 mm), as expected for a rigid body. This consistency validates the measurement, making the standard deviation of positional changes (the jitter metric) a more informative indicator of tracking precision.

Jitter values reveal a clear split: Markers 4 and 5 exhibit significantly lower jitter (4.82 and 6.49 mm) than Markers 0 to 3 (13.08 to 17.55 mm). Since the underlying motion is identical for all markers, this difference is a measurement artefact, not a result of physical movement. The cause is the spatial position of each marker relative to the cameras’ optical centres. Markers closer to the image centre benefit from lower residual lens distortion and better-conditioned triangulation, reducing positional noise.

Ultimately, this variation in jitter reflects the quality of the underlying camera calibration. Residual inaccuracies in the intrinsic camera models propagate into the triangulation, producing inconsistent three-dimensional position estimates, particularly for markers far from the optical centre. The elevated jitter in Markers 0 to 3 is therefore a direct consequence of calibration quality and would be expected to decrease with an improved calibration.

4.6.2 Methodological Implications

The results demonstrate that the pipeline reliably produces trajectory data under real operational conditions, with coverage consistently above 88% across all markers. Losses are attributable to temporary occlusion rather than systemic pipeline failure.

The similarity in mean step distances across all markers confirms internal physical consistency, validating the measurement approach. The jitter split between marker groups indicates that precision is spatially non-uniform, an expected characteristic of camera-based systems, where accuracy naturally decreases with distance from the optical centre due to lens distortion.

Smoothness performance is therefore calibration-limited rather than a pipeline limitation and would improve directly with better intrinsic calibration. While absolute accuracy remains unverified, the system's fitness for its intended purpose as a benchmarking tool can be assessed. The key distinction is that the system's jitter represents a non-cumulative positional error. This provides a stable baseline against which cumulative errors, such as the common drift in inertial sensors, can be effectively measured. The best-case jitter of 4.8-6.5 mm is therefore suitable for this purpose in applications involving gross motor movements, confirming the system's utility.

4.7 Overall System Performance and Limitations

The preceding sections have each examined an individual subsystem in isolation. This section consolidates those findings into a unified view of overall system performance, identifies the principal limitations and bottlenecks that emerged from the evaluation, and assesses the pipeline's fitness for its intended purpose as a basis for the chapters that follow.

4.7.1 Summary of System Achievements

The analyses presented in this chapter demonstrate that all primary subsystems met their defined performance targets. Intrinsic calibration achieved re-projection errors consistently around 0.9 pixels, within the sub-1.0 pixel target, and extrinsic calibration yielded baseline accuracy within ± 30 mm across repeated runs. Both synchronisation methods maintained a mean time spread of approximately 13.3 ms across 2000 frame groups, well within the 20 ms threshold and with no observable drift. Object detection achieved a 93.0% successful detection rate over 1615 frames. Three-dimensional reconstruction produced re-projection errors of three to four pixels, which did not meet the sub-2-pixel target defined in Section 3.5.3.4; this is attributed primarily to residual calibration inaccuracies and the outlier spikes visible in Figure 4.3 and 4.4, though the practical impact is partially reduced through averaging of the triangulated position estimates from multiple camera pairs, which helps suppress outlier-induced variance. Trajectory tracking confirmed a mean coverage of 92.90% across all markers, with coverage losses attributable to temporary occlusion rather than pipeline failure.

4.7.2 Identification of System Limitations and Bottlenecks

The evaluation also identified a set of clear, data-supported limitations. The most significant is spatially non-uniform tracking precision. The jitter disparity between Markers 0-3 (13.08-17.55 mm) and Markers 4-5 (4.82-6.49 mm) demonstrates that positional noise increases for markers located further away from the optical centre, a direct consequence of residual lens distortion from the intrinsic calibration. Synchronisation performance is bounded not by the choice of algorithm

but by the underlying USB hardware and operating system scheduling, as evidenced by the near-identical results of both synchronisation methods. Trajectory coverage is structurally limited by the requirement for simultaneous visibility from at least two cameras, making occlusion an inherent rather than correctable constraint. Lastly, the absence of an external ground truth means that only internal consistency can be evaluated; absolute positional accuracy remains unverified.

While no external ground truth was available to measure absolute positional accuracy directly, an indirect estimate can be derived from existing metrics. The extrinsic baseline error of ± 30 mm bounds the systematic spatial error introduced by calibration, while the jitter standard deviation of 4.82-6.49 mm for the best-positioned markers and 13.08-17.55 mm for the worst-positioned markers bounds the random positional noise. As these two error sources compound additively, the estimated positional accuracy ranges from approximately 5 mm in the best case to approximately 37-48 mm in the worst case, depending on marker position within the capture volume and calibration quality. This range is consistent with what a low-cost webcam-based system would be expected to achieve, though direct validation against a reference instrument remains a recommended step before operational deployment as an IMU benchmarking tool. The dominant contributor to the upper bound is the extrinsic calibration error; the improved calibration pipeline introduced in Section 5.1.1 is expected to reduce this component, with a further reduction possible through the ChArUco calibration strategy discussed in Section 6.2.4.

4.7.3 Synthesis and Fitness for Purpose

Considered as a whole, the results of the analysis confirm that the pipeline functions correctly and consistently across all subsystems. The performance chain connecting calibration quality, triangulation fidelity, and trajectory smoothness is coherent and well-understood. Each limitation has a clearly identified cause, and the primary bottleneck, the calibration quality, is tractable. The system therefore constitutes a validated foundation for the low-cost motion capture approach developed in this thesis.

Chapter 5

System Refinement and Final Results

Building upon the analysis presented in Chapter 4, this chapter details the improvements implemented within the motion tracking system. Key advancements include modifications to the camera setup and the integration of a rigid body template. The chapter then presents the final outcomes of the refined system, supported by data that quantifies its effectiveness and discusses the implications for future development.

5.1 System Improvements

This section describes the key improvements of the system after the findings from the analysis chapter.

5.1.1 Calibration

As mentioned in Section 4.1.2, the calibration performance directly correlates with system performance. To improve the calibration sequence, calibration steps have been reworked. Previously, the calibration was performed linearly, starting with individual recordings for each camera, followed by the extrinsic recording. After this, the calibration was finalized, regardless of its quality.

The new pipeline begins identically to the previous one, but after the individual recordings, it directly performs the intrinsic calibration. If any camera is not within the threshold value of a sub-1 pixel re-projection error, that specific camera is looped back into a recalibration. This is done until all cameras are within the predefined limits or until a specified amount of failed attempts has been reached, at which point the system proceeds with the best available result in order to prevent an infinite recalibration loop.

After the intrinsic calibration is complete, the improved extrinsic calibration begins. Previously, a mean was used to calculate a relation across all camera positions. This resulted in a non-predictive world coordinate system. The improved system is now calculating each camera's relation to camera 0 and therefore making camera 0 the origin of the world coordinate system. This improvement makes each recording more stable and predictable than the previous implementation. By enforcing a consistent world coordinate origin and a minimum calibration quality threshold, this improvement

is expected to reduce the systematic spatial error identified in Section 4.7.2, thereby tightening the upper bound of the positional accuracy estimate.

5.1.2 Rigid Body Template

A rigid body is an object whose sides and surfaces maintain fixed spatial relationships to one another, containing no axles or moving parts. If an object contains a moving part, that part is treated as a separate rigid body.

The rigid body template is established by recording all markers of the object across multiple frame groups, requiring at least two markers to be simultaneously visible to the cameras in each frame. This allows the spatial relationships between all marker positions to be calculated and stored as a fixed template.

During motion tracking, each visible marker's detected pose is combined with its stored template entry to recover the full rigid body pose in world space. When multiple markers are visible their estimates are averaged, with rotations fused using the spherical mean as defined by Moakher [18]. This allows full reconstruction as long as at least one marker is visible. Unlike the previous system discussed in Chapter 4, which was evaluated using a planar marker setup, this rigid body template approach enables robust tracking of arbitrary three-dimensional objects, significantly expanding the system's application scope.

5.2 Overall System Performance

This section is a summary of the system performance and limitations with key functional and performance aspects as well as data to back the performance statements.

5.2.1 Core System Capabilities

This section describes the systems functionalities and limitations.

5.2.1.1 Degrees of Freedom

The system is capable of capturing a rigid body's translational degrees of freedom, representing its position in three-dimensional space, as well as its rotational degrees of freedom, expressed as yaw, pitch, and roll.

5.2.1.2 Recording Duration

The system's recording capacity is approximately 3-4 minutes at 30 fps, limited by memory. Recording for longer durations requires more RAM to temporarily store the captured frames.

5.2.1.3 Tracking Range

The tracking range of the camera system depends on two factors: camera resolution and ArUco marker size. A higher camera resolution provides more detailed frames, allowing smaller markers to be detected. This relationship also explains why larger markers are easier to detect.

5.2.1.4 Frame Rate

The system is not dependent on a specific frame rate. However, a high frame rate provides smoother motion captures at the cost of reducing the total recording time, since the number of frames captured per second increases and therefore also memory usage.

5.2.1.5 Processing

The system leverages both real-time processing and post-processing to optimize efficiency. Real-time processing is used during recording to synchronize frames, which directly contributes to a shorter total processing time for the user. Marker position calculations, along with calibration and rigid body template computations, are all handled in post-processing. These tasks begin only after the user stops the recording and the complete dataset for their respective phases is available.

5.2.1.6 Rigid Body Templates

During the initial template calibration phase, at least two markers must be simultaneously visible to establish spatial relationships. However, for continuous motion tracking, the system requires visibility of at least two of its markers across different cameras, or a single marker visible simultaneously on multiple cameras. This robust tracking mechanism enables the reconstruction of the complete pose of the object, even under conditions where certain markers may be obstructed.

5.2.1.7 Scalability

The system is not dependent on the number of markers that are used, which enables the possibility of using as many markers as fits the capture volume. This makes the system capable of motion tracking of more complex scenarios, e.g., a person performing a task or a moving robot.

5.2.2 Key Performance Metrics

The results shown in Table 5.1 demonstrate that the key functions of the system met their defined performance target. It is important to note that, with the exception of synchronization, these metrics are inherently influenced by user input and recording conditions, and therefore represent expected ranges rather than fixed values.

Table 5.1: Key Performance Metrics from Chapter 4.

Sub System	Metric	Result
Camera Calibration	Extrinsic Baseline Error	± 30 mm
Synchronization	Mean Time Spread	$\simeq 13.3$ ms
Object Detection	Detection Rate	93%
3D Reconstruction	Re-projection Error	2 - 4 px
Trajectory Tracking	Mean Coverage	$\simeq 92.90\%$

** An improved calibration technique, discussed in Section 6.2.4, was tested after the primary evaluation phase. Results are presented in Appendix A.*

5.3 Visual Representation of Achieved Results

The following figures present a visual representation of the system's achieved results:

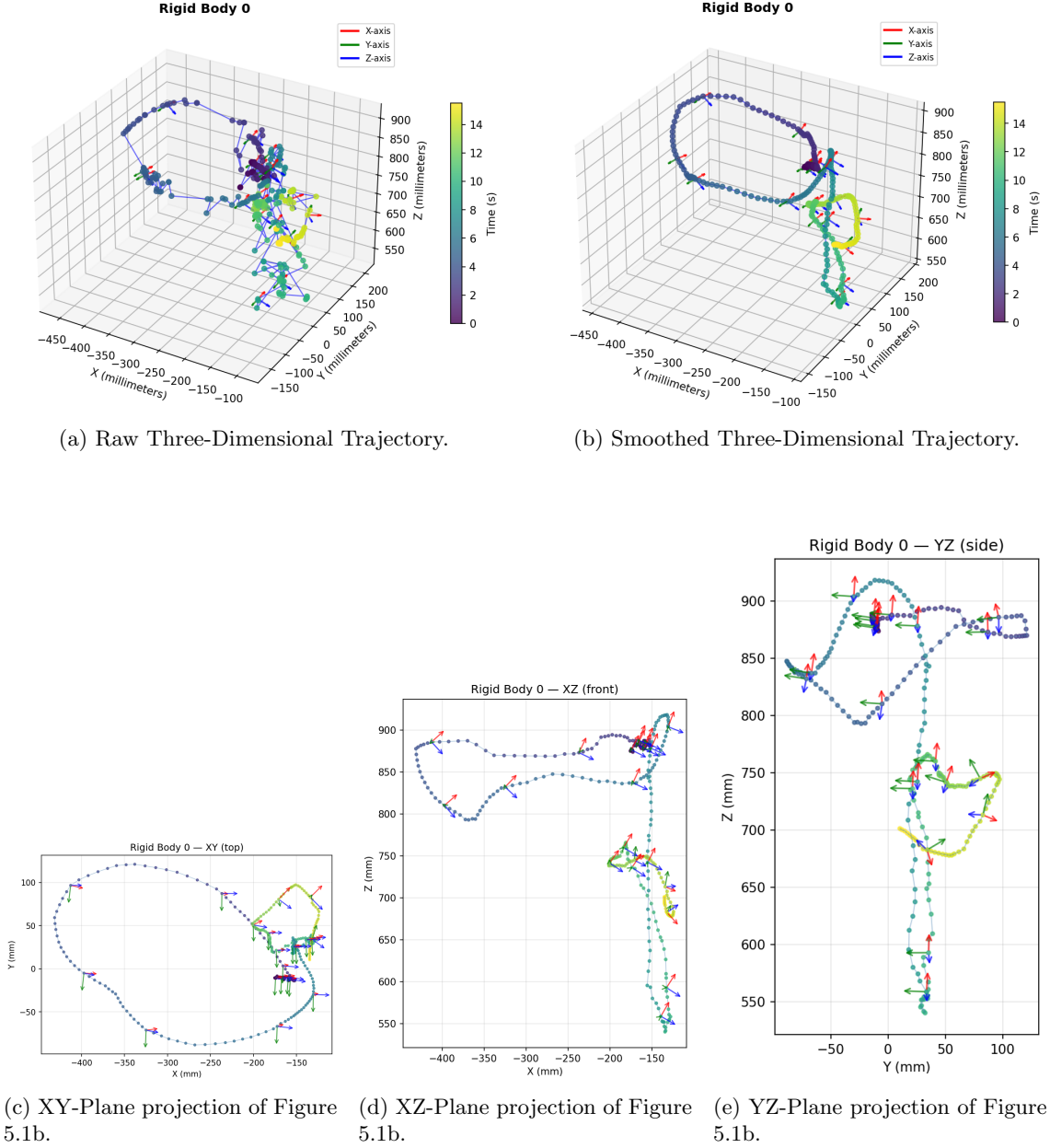


Figure 5.1: Visual representation of the rigid body's tracked trajectory. The top row shows the three-dimensional trajectory in its raw (a) and smoothed (b) forms. The bottom row provides two-dimensional projections of the smoothed trajectory onto the XY (c), XZ (d), and YZ (e) planes for detailed analysis of planar movements.

Figure 5.1 offers a visual overview of a rigid body's tracked motion. It presents both the raw

and processed three-dimensional trajectories, alongside their two-dimensional planar projections, to enable detailed analysis.

Figure 5.1a displays the raw three-dimensional trajectory. This initial data contains inherent measurement noise, which can obscure clear motion patterns. To improve clarity, a rolling mean filter was applied. The smoothed trajectory, shown in Figure 5.1b, reveals the rigid body’s path with reduced noise, enhancing interpretability.

The smoothed three-dimensional trajectory in Figure 5.1b illustrates three movement phases performed by the operator. The sequence begins with an initial circular motion (approximately 0-5 seconds), followed by a linear forward and backward movement (5-10 seconds), and concludes with a rotational manoeuvre (10-15 seconds). This visually aligns with the intended physical actions, as manually controlled during the experiment, demonstrating successful capture of the user’s input.

For a more in-depth examination of specific planar characteristics, Figures 5.1c, 5.1d, and 5.1e provide two-dimensional projections of the smoothed trajectory from Figure 5.1b onto the XY, XZ, and YZ planes, respectively. These projections are useful for assessing movement within each plane and for highlighting subtle details. For example, the circular motion is most distinct in Figure 5.1c, while the linear translation and the rotational movement are better understood by their appearance in Figure 5.1d and 5.1e.

This visual assessment confirms the system’s capability to track and represent rigid body movements, translating user inputs into quantifiable trajectory data.

5.4 Critical Component Performance Indicators

The following sections identify the two critical components that most directly dictate the quality of the system’s output. Rather than summarising all performance metrics, this section highlights the key areas where improvements have been made and where the remaining quality ceiling is determined.

5.4.1 Camera Calibration Summary

Camera calibration is the foundational performance bottleneck of the system. As demonstrated in Section 4.6.1, residual inaccuracies in the intrinsic camera models propagate directly into triangulation precision, causing the jitter disparity observed between marker groups. This makes calibration quality the primary determinant of trajectory smoothness and accuracy.

The improved calibration pipeline introduced in Section 5.1.1 addresses the most significant weakness identified in the analysis, operator-dependent inconsistency. By enforcing a recalibration loop until a sub-1 pixel re-projection error is achieved, the system aims for a minimum intrinsic calibration standard and, after a specified number of retries, accepts the best quality achieved, rather than accepting whatever quality a single capture session produces. Furthermore, the refined extrinsic calibration process, which establishes Camera 0 as the consistent world coordinate origin, enhances the stability and predictability of the overall system’s pose estimations. The practical impact is a raised and enforced quality floor.

However, calibration quality remains input-dependent by nature. While the pipeline now rejects poor calibrations, the upper bound of achievable quality is still determined by the diversity and coverage of the chequerboard images provided, and precision diminishes towards the edges of the camera’s field of view. The reasoning behind this limitation and the case for an alternative calibration approach are discussed in Section 6.2.4, and the results of that improvement are presented

in Appendix A.

5.4.2 Object Detection Success Rate

The primary improvement to object detection robustness is the introduction of the rigid body template described in Section 5.1.2. In the approach evaluated in the first development iteration, described in Chapter 3 and 4, which used a planar object, each marker's three-dimensional position was treated independently, meaning the loss of any individual marker directly resulted in lost positional data for that frame, reflected in the coverage range of 88.23% - 98.02% observed across all markers.

The rigid body template addresses this by establishing the spatial relationships between all markers during an initial recording phase. By applying the template-based pose recovery described in Section 5.1.2, the full rigid body can be reconstructed in three-dimensional space as long as at least one marker remains visible to the cameras, meaning individual marker occlusions no longer necessarily result in a failed reconstruction.

Considering the 93% system-wide detection rate observed in the analysis, the rigid body approach represents a significant robustness improvement since it now supports three dimensional rotations of the rigid body better. A frame that would previously have failed due to a single occluded marker can now still yield a valid reconstruction. Combined with the guaranteed sub-1 pixel calibration introduced in Section 5.1.1, the quality of each individual reconstruction is also expected to improve, further reducing the positional noise that contributed to jitter disparities identified in Section 4.6.1. The practical ceiling of the system is therefore no longer individual marker detection continuity, but whether the rigid body as a whole remains sufficiently visible within the capture volume.

Chapter 6

Discussion

This chapter reflects on the system developed throughout this thesis, examining how well it performs in relation to its intended purpose and the design decisions that shaped it. Each major design choice is evaluated in terms of its trade-offs and practical consequences, followed by an assessment of the system’s limitations and the sources of error encountered during development and validation.

6.1 System Performance in Relation to Purpose

This thesis set out to develop software capable of motion-tracking a specific object, with the end goal of providing Bosch Sensortec a tool to evaluate sensor tracking algorithms. The system developed in this thesis fulfils that implementation goal, specifically it:

- captures position and rotation of a marked rigid body across multiple cameras
- synchronizes data streams
- reconstructs occluded markers
- exports the results in a structured format ready for downstream analysis

The discussion that follows reflects on two aspects of this outcome: how well the implementation performs in practice, and to what degree the output is genuinely fit for its intended purpose as an IMU benchmarking tool.

6.2 Key Design Decisions and Their Consequences

Throughout the development of the motion capture system, several design decisions were made that significantly influenced the system’s capabilities, constraints, and overall performance. This section evaluates the most consequential of these choices, the selection of ArUco markers for object detection, the synchronization strategy, the rigid body template approach, and the camera calibration method, discussing both the reasoning behind each decision and the practical implications that emerged.

6.2.1 ArUco Markers Strategy Evaluation

The motion capture system developed in this thesis relies on the detection and tracking of ArUco markers. This approach was selected due to several factors: their robust detection capabilities, seamless integration with the OpenCV library, their reliable performance in controlled environments, and their inherently low cost.

During the system’s development, several known limitations of ArUco markers became apparent. These include the necessity for their physical presence on the tracked objects, the requirement that the marker be visible and identifiable within the camera’s frame at an appropriate scale, the impact of varying lighting conditions on detection rates, and the occasional problem of glare. Specifically, reflected light from light sources in a room, particularly off the paper on which markers are printed, can sometimes render a marker undetectable.

While evaluating the use of ArUco markers, other object detection and tracking strategies were also researched. Two main alternatives were considered: traditional feature-based object recognition and deep learning-based object detection.

The traditional object recognition approach operates on a similar principle to ArUco markers in that it tracks pre-defined visual patterns. However, instead of relying on a reference marker, it tracks inherent shapes, colours, sharp edges, or other distinct features present in the frames, as mentioned in 2.6. This method would have been viable if the system’s requirements had prevented the application of physical markers. Nevertheless, ArUco markers were deemed superior for our application as they perform the same function with greater robustness and lower processing demands compared to a general feature-based approach.

Deep learning-based object detection utilizes large datasets to train models for identifying pre-defined objects within an image and their spatial relationship. While this is a highly capable approach, capable of generating excellent results, its feasibility for this thesis was constrained by several factors. It is highly dependent on large-scale labelled datasets, significant processing power, and the availability of a robust ground truth. Given that this thesis aims to research a low-cost motion capture system using webcams and lacked an established ground truth, deep learning was concluded to be an unsuitable solution due to its demanding data and computational requirements.

Finally, an evaluation was made against industry-standard motion capture systems, even though such solutions typically fall outside the scope of a webcam-based system. Possible alternatives using different sensor types than standard cameras include depth sensors, infrared sensors, and LiDAR systems.

All these alternative sensors offer the potential for significantly higher accuracy than webcam-based systems, especially under optimal conditions. This superior performance is primarily attributed to two factors: these approaches typically involve more complex setups with multiple sensors capturing richer data, and they inherently bypass this system’s largest bottleneck, camera calibration, as they provide direct three-dimensional measurements. While such alternative approaches would likely result in a more sophisticated motion capture system, they were excluded from the scope of this thesis due to their higher cost, increased setup complexity, and their non-reliance on webcams.

Following this comprehensive evaluation, it was concluded that ArUco markers serve as the most fit-for-purpose solution for this software, given its intended objectives and constraints.

6.2.2 Synchronization Strategy Evaluation

Accurate multi-sensor fusion and state estimation necessitate that all data originate from the same physical instant in time. Without proper synchronization, sensor measurements from different devices, or even different parts of the same device, will not correspond temporally, leading to inaccurate state estimation which can snowball into larger problems.

The root causes of asynchrony can be linked to three main areas:

- **Independent Hardware Clocks:** Each hardware component used in this system possesses its own unique internal clock with an inherent offset and varying drift rates. This results in components operating at slightly different speeds and inconsistent relative timing.
- **Processing Delays:** Data may undergo varying amounts of pre-processing or buffering before becoming available for system-level processing, introducing unpredictable delays.
- **Varying Communication Latencies:**
 - **External Transmission:** Any data transferred across different physical mediums introduces variable and often unpredictable delays. Factors include cable length, type, and hub processing.
 - **Internal Bus Delays:** Even within a single processing unit, different data streams might traverse various internal communication buses with differing speeds and priorities, leading to skewed arrival times. Furthermore, multiple external devices can be internally linked to the same bus, worsening priority and queueing issues.

As stated in Chapter 4.2, an evaluation was made between real-time and post-processing approaches for frame synchronization. In the comparison shown in Figure 4.2, it was concluded that both methods performed equally under the specific conditions encountered. However, real-time processing offered significant advantages in speed and operational immediacy, leading to its selection for the system.

While the chosen synchronization strategy proved effective under current operating conditions, several factors could impact its performance with different system configurations:

- **Hardware Dependencies:** The synchronization's efficacy is highly dependent on the specific hardware components used.
- **Data Volume Increases:** A primary untested vulnerability is the system's ability to handle significantly increased data volume. For instance, if camera frequencies double to 60 FPS per camera, and resolution increases to 1080p, a scenario not fully explored due to the non-optimal camera quality that will be discussed in Chapter 6.3.2, the current real-time synchronization algorithm could be overloaded. The implemented memory-saver, which discards frames when buffers become too large, could lead to data loss if computation cannot keep pace. Such an increase in frame input and frequency could also pose memory management challenges.
- **Internal Bus Configuration:** Improper configuration or heavy loading of internal data buses could lead to prioritization conflicts, preventing certain camera frames from arriving in time. This would result in incomplete or incorrect frame group creations.

Environmental factors also warrant consideration. Temperature is a known variable influencing hardware clock stability, potentially altering timing processes. Similarly, electromagnetic interference could affect data transmission integrity and timing.

Nevertheless, the resulting frequency of synchronized frame groups and the rate of discarded frames were deemed sufficient for the intended use case.

6.2.3 Rigid Body Template Strategy Evaluation

Rigid body templates are used to preserve the spatial relationships between markers in a fixed positional and rotational configuration relative to one another. This is achieved by calibrating one or more objects before motion tracking. In theory, this enables the estimation of the position and orientation of related markers even when some markers are not visible. In turn, this enables the computation of an object's centroid even when all related markers cannot be observed.

The template creates the ability to track any three-dimensional object when the cameras are not seeing all sides of it, with a calibration of the object done prior to motion tracking. This is a central ability for this system since the ability to see all sides of an object is limited.

The flaw with finding relations and reusing them to reconstruct not visible parts of an object during motion tracking. Is the possibility of the relation template not being truthful to the real world relations of the markers. This would create a flaw in every frame during a motion track where the template is needed to reconstruct not visible markers. Thereby the system design relies on the user for how well the system preforms. Which could be prevented in a system where calibration of rigid body templates where not needed. Instead a larger number of cameras could be used to completely eliminate the need of relation templates. Eliminating the possibility of calibrating your own objects would also make the systems performance less user dependent, but at the same time limiting the user to certain objects calibrated during development could possibly make the system disadvantageous for some use cases. Solutions for this could be to implement both pre defined objects and the possibility to calibrate your own templates. Careful user manuals of how to use the system highlighting importances that results in layers of errors could also be written as a solution.

6.2.4 Camera Calibration Strategy Evaluation

This system initially used a chequerboard for intrinsic and extrinsic camera calibration, prioritizing ease of use and perceived point density. The reasoning was that chequerboards offer more corner detections per area, as square size is less critical than for ChArUco boards, which require larger squares to embed detectable ArUco markers. This higher point count was expected to yield better calibration.

However, a key limitation emerged: reliable chequerboard detection requires the entire board to be visible in every image. This constrains stable calibration to areas near the optical centre. While motion capture is accurate centrally, precision diminishes significantly at the camera's edges.

ChArUco boards would likely mitigate this limitation. They do not require full visibility; their detection mechanism allows pose estimation and corner interpolation even from partially visible boards. This robustness would be expected to extend the effective calibration volume, enabling more accurate motion tracking across a wider field of view. Whether this theoretical advantage translates into a meaningful practical improvement for this specific system is explored in Appendix A.

6.3 Limitations and Sources of Error

No system is without its limitations, and the one developed in this thesis is no exception. This section addresses the two most notable sources of error that affected system performance. The first concerns the geometric trade-offs that arise from using only three cameras for triangulation, and the second relates to the driver restrictions that prevented the cameras from operating at their full potential.

6.3.1 Three-Camera Triangulation

The system uses a three camera approach to determine position and rotation of a three-dimensional object in space.

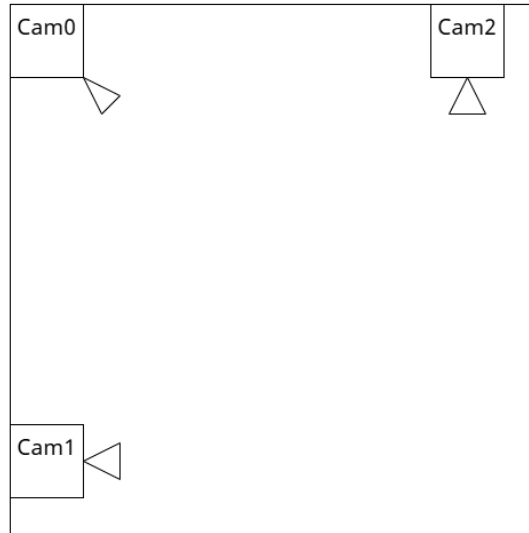


Figure 6.1: Top-down view of the camera setup.

This gives the system the benefit of being deployable i.e., easy to set up in an office environment, when set up as shown in Figure 6.1. The deployability of the system is a main purpose of the system, using as few as three cameras contributes to that purpose. It is worth mentioning that this approach results in flaws of depth perception, as described in Section 2.4, thus one angle is not covered by a camera. Given that three cameras are used, the camera positioning shown in Figure 6.1 is chosen for reducing depth perception error.

A triangulation becomes more accurate if the same markers are seen by two different cameras simultaneously, rather than two cameras seeing different markers and using the rigid body template to create a triangulated position. This was tried but resulted in larger deviations since the rigid body calibration errors from Section 6.2.3 enlarged the triangulation error. The solution used was to squeeze the cameras together, making it more likely for a marker to be detected by at least two cameras.

The solutions to the two problems described above, the positioning shown in Figure 6.1 and the squeezing of the camera positioning were found to obstruct each other. Squeezing the cameras together results in the loss of depth perception and spreading them apart results in larger trian-

gulation errors. The decision to squeeze the cameras together was made since the ArUco marker size is known to the system and contributes to depth perception.

Adding more cameras to the system could be a theoretical extension to address both limitations described above. The current system is based on three cameras, which already enables triangulation between views; however, increasing the number of cameras would improve robustness by ensuring that each marker is observed from multiple viewpoints simultaneously.

With additional viewpoints, depth estimation becomes more reliable due to increased redundancy in triangulation, reducing sensitivity to occlusions and viewpoint-dependent errors. Furthermore, improved coverage of the object from multiple angles would reduce reliance on rigid body templates for pose estimation, thereby decreasing the impact of rigid body calibration errors.

6.3.2 Camera Drivers

A significant constraint throughout the development of the system was the lack of appropriate drivers for the cameras. As mentioned in Section 3.2.1.2, the Logitech MX Brio officially supports either 1080p resolution at 60 frames per second or 4K resolution at 30 frames per second. In addition to impressive resolution and frame rates, the MX Brio webcam also incorporates built-in image enhancement algorithms to, for example, reduce motion blur.

However, these advanced features were never accessible due to company policy prohibiting external, non-network-approved downloads. This limitation meant that the high-end MX Brio webcams were restricted to standard system generic drivers. To address this, research was conducted into the actual output capabilities of the webcams when Logitech-specific drivers were unavailable.

After research, it was found that if cameras lack their specific drivers, they typically default to the operating system's generic webcam driver, known as the USB Video Class (UVC) driver. This driver enables only the most basic functions, such as fundamental frame rate and resolution settings.

OpenCV utilizes two primary APIs to interact with webcam drivers. When the correct vendor-specific driver is detected, these APIs offer the following characteristics:

- **MSMF (Microsoft Media Foundation)**: Supports more features and is considered the modern and most capable API today.
- **DSHOW (DirectShow)**: Considered a legacy API and only supports basic features.

The challenge discovered with these two APIs arises from their interaction with the USB Video Class driver. When these APIs do not connect to a webcam's specific driver, they limit themselves to a "safe mode," which, after further research, was found to have the following impacts on code execution:

- **MSMF (Microsoft Media Foundation)**: Locks resolution to the most stable, often lowest, resolution and frame rate found to be supported reliably by the generic driver. Internal indexing of cameras works reliably. Compresses images.
- **DSHOW (DirectShow)**: Offers support for different resolutions and frame rates but exhibits silent failures, falling back to lower settings without warning, making it impossible to guarantee what the cameras are actually delivering at any moment. Memory usage is higher than MSMF at equivalent settings. The webcam connects to different internal computer indexes with each restart, with minimal accessible identification information.

After understanding the differences between the MSMF and DSHOW APIs and their performance with USB Video Class drivers, an assumption was made that the DSHOW API could potentially

be the better choice due to its theoretical support for higher resolutions and frame rates. However, two critical problems made this unviable. First, the random camera indexes with each restart made it impossible to reliably identify and control specific cameras in a multi-camera setup. Second, and more significantly, DSHOW’s silent fallback behavior meant that the cameras would change their resolution and frame rate mid-operation without any notification, producing inconsistent and unreliable data that could not be trusted. In practice, the higher resolution support was also found to be limited, with delivered frame rates often falling well below requested values. Additionally, DSHOW’s higher memory usage at equivalent settings further strained system resources. MSMF, while limited to 640×480 p at 30 fps, behaves deterministically: it either operates at a confirmed mode or rejects it outright, making its output predictable and verifiable. This deterministic behavior was prioritized over DSHOW’s theoretically higher ceiling, and DSHOW was subsequently discarded as a viable solution. Therefore, after careful consideration, the decision was made to stick with the stable, albeit low-resolution, 640×480 p, MSMF approach.

Since the system’s full potential cannot be tested under these driver limitations, only speculations can be made. However, it is reasonable to expect that the potential for a 6.75-fold increase in resolution (from 640×480 p to 1080×1080 p) or a 27-fold increase (to 4K), along with a potential doubling in frame rate, would significantly improve the accuracy of the system, especially when considering the webcams’ internal image enhancement algorithms.

Something interesting to explore, if the full camera potential were reached, would be to stress-test the camera synchronization algorithm, processing times, and memory usage. Memory-wise, the system would fill up faster, reducing recording time. Processing times would grow significantly due to more pixels to analyse, and this analysis is performed multiple times on the same frame group. The synchronization algorithm would probably hold up, but if the incoming frame rate were to exceed what the sync queue can absorb, individual frames would be silently dropped as they arrive. The queue has a fixed capacity and discards new frames when full rather than flushing. This would create scattered gaps in the recorded frame groups. In contrast, completed frame groups are never discarded and accumulate in memory until recording ends, meaning sustained overload would instead cause unbounded memory growth. This scenario has not been close to occurring during the current stress testing.

6.3.3 Causes and Implications of Signal Faults and Missing Data

No motion capture system is perfectly immune to data imperfections, and the low-cost webcam-based solution developed during this thesis is no exception. While previous sections have highlighted individual limitations related to system components, this section consolidates and elaborates on how these, along with observed environmental factors, collectively manifest as signal faults and missing data. Understanding these integrity challenges is critical for evaluating the system’s reliability and informing future improvements.

6.3.3.1 Factors Leading to Missing Data and Detection Failures

The absence of marker detections, which results in visible gaps within the reconstructed trajectory data, as shown in Figure 6.2, stems from a complex interplay of geometric limitations, environmental challenges, and the inherent constraints imposed by the camera hardware and software. The fundamental three-camera setup, while optimized for deployability as discussed in Section 6.3.1, inherently limits viewpoint diversity. Markers frequently become occluded by the tracked object itself, preventing them from being simultaneously visible to the minimum two cameras required for triangulation. The issue is particularly pronounced during dynamic movements. For instance, rapid object rotation or movement can lead to markers not being detected due to harsh angle conditions, high movement speeds, or the object leaving the field of view of two cameras, which

then results in a singular camera not being able to triangulate. A key observation is that certain rotations of the object can lead to a loss of information as markers may no longer be observable from a sufficient number of angles.

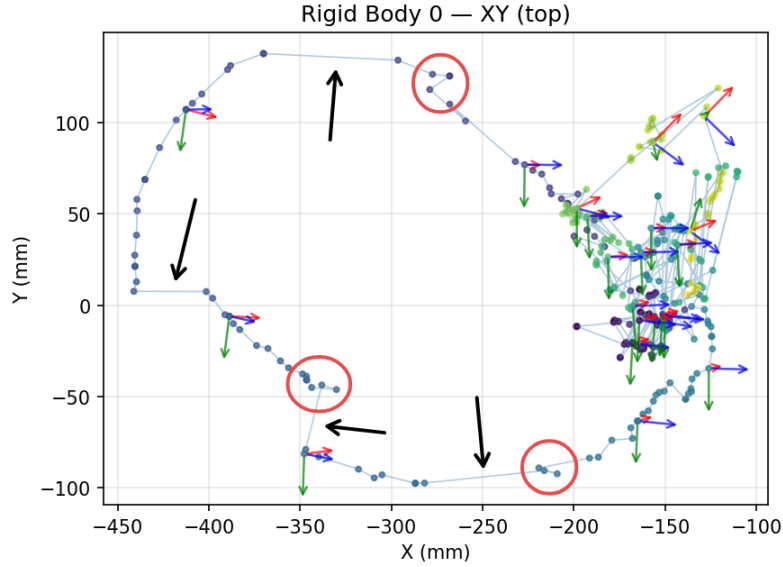


Figure 6.2: The XY-plane projection of a raw trajectory recording, corresponding to the data shown in Figure 5.1a. The black arrows indicate instances of missing data or "no reads", while red circles highlight sections exhibiting signal faults, such as localized inaccuracies and jitter.

Beyond these geometric challenges, the system's ability to consistently detect markers is heavily influenced by image quality. A significant contributor to detection failures is the restricted image quality enforced by the camera driver limitations detailed in Section 6.3.2. Operating at a resolution of 640×480 p and 30 frames per second, without access to the cameras' native image enhancement algorithms for capabilities such as motion blur reduction, means that fast-moving markers can appear blurred within the captured frames. This blurring effect makes it difficult for the ArUco detection algorithms to robustly identify the marker's pattern or to accurately calculate its position, leading to intermittent "no reads." Furthermore, the system's sensitivity to ambient lighting conditions directly impacts marker visibility. In dimmer environments, webcams automatically increase their exposure times to gather sufficient light. While this brightens the image, it simultaneously amplifies motion blur for any moving markers, creating a critical trade-off where improved illumination leads to degraded temporal clarity and an increased incidence of "no reads" due to blur. Conversely, highly uneven or excessively bright lighting can cause saturated pixels within the marker regions, hindering accurate detection. It was observed that the insufficient width of the white border surrounding the ArUco markers themselves might be another contributing factor to detection difficulties. When the greyscale filter was applied during image preprocessing, this narrow white border, particularly against a dark background or in areas of varying illumination, could potentially become indistinguishable from the marker's internal pattern or the background. This phenomenon could reduce the marker's perceptible area, making it more challenging for the detection algorithm to reliably segment the marker from its surroundings and thus increasing the likelihood of missed detections. A frequent and particularly challenging issue observed with both printed markers and calibration patterns was their susceptibility to glare. Reflections from ambient light sources, originating from overhead lighting or windows, and off the ink of the printed markers, often had a tendency to obscure ArUco patterns at certain angles. These specular highlights often

rendered markers partially or entirely undetectable, or caused erroneous pose estimations due to the algorithm’s inability to reliably extract the four corners. While not critically observed at the current operational parameters, the discussions in Section 6.3.2 regarding camera drivers and Section 7.4.2.4 concerning processing time highlight a potential future cause for data gaps. If higher resolutions or frame rates were achievable, the synchronized frame buffer could be overwhelmed, or processing times might exceed acquisition rates, leading to frame groups silently being dropped, thereby creating additional, scattered gaps in the recorded output.

6.3.3.2 Factors Contributing to Signal Faults

Beyond complete detection failures, the system also exhibits signal faults, where markers are detected but their reconstructed three-dimensional positions or orientations are inaccurate, as exemplified by the features shown in Figure 6.2. These errors often arise from systematic issues that propagate through the processing pipeline. The fidelity of the reconstructed three-dimensional data is fundamentally tied to the precision of both intrinsic and extrinsic camera calibrations. As noted in Section 6.2.4, the reliance on chequerboards led to less stable calibration in the periphery of the camera’s field of view. This inherent inaccuracy propagates, meaning that detected two-dimensional marker positions are mapped to incorrect three-dimensional coordinates, especially for markers tracked towards the edges of the active volume. Furthermore, the use of rigid body templates to infer the pose of undetected objects, as detailed in Section 6.2.3, introduces another potential source of signal fault. Any discrepancy between the calibrated template and the real-world spatial relationships of markers on the physical object directly translates into systematic errors in the three-dimensional reconstruction. These errors become more pronounced when the system relies heavily on the template to reconstruct a significant number of occluded markers. Even under optimal detection conditions, the process of triangulating two-dimensional marker projections from multiple cameras into a three-dimensional point is susceptible to numerical errors. The geometric configuration of the three cameras, with its inherent compromises between depth perception and triangulation accuracy, as mentioned in Section 6.3.1, implies that certain object positions and marker configurations will yield less reliable three-dimensional reconstructions. These inherent geometric compromises lead to small but persistent inaccuracies, manifesting as noise or jitter in the reconstructed trajectories. Related to the image quality challenges, factors such as slight motion blur, glare, or pixel saturation, as discussed in Section 6.3.3.1, can cause the detection algorithm to miscalculate the precise two-dimensional centroid of a marker. Even minor errors in two-dimensional centroid detection are significantly amplified during the triangulation process, resulting in noticeable inaccuracies in the final three-dimensional coordinates.

An observed pattern further compounding signal faults is their frequent occurrence immediately preceding or following instances of “no reads.” Two primary hypotheses explain this transitional instability. Firstly, it is plausible that during these periods, when markers are at the very boundary of detectability, the direct detection algorithm struggles to identify them robustly. While a detection might still occur, the underlying ArUco pattern analysis could be suboptimal, leading to less accurate corner or centroid estimations. Secondly, these transitional phases often involve a change in the specific ArUco markers being tracked or the set of visible markers used to determine the rigid body’s pose (Section 6.2.3). If the rigid body template itself contains calibration errors or an inherent offset for certain marker configurations, then relying on a newly available or re-detected set of markers could introduce a shift in the calculated pose. This shift, which manifests as apparent noise or a jump in the trajectory, would persist until the system re-establishes a stable and consistent set of marker detections for the rigid body. Both mechanisms could independently or synergistically contribute to localized signal faults and jitter until the marker detection and pose estimation conditions stabilize.

6.3.3.3 Implications for Data Analysis and System Reliability

The pervasive nature of these signal faults and missing data points significantly impacts the utility of the generated motion capture data. As depicted in Figure 6.2, the raw output often presents trajectories with numerous discontinuities and localized inaccuracies, making direct interpretation challenging. While techniques like data interpolation, which will be discussed in Section 7.4.2.5, can be employed to bridge these gaps, it is crucial to recognize that such interpolated values are estimations rather than directly measured data, thereby introducing a layer of uncertainty. This understanding of data integrity challenges underscores the inherent trade-offs in designing a low-cost motion capture system and establishes key priorities for future methodological and technological improvements.

6.4 Societal Benefits

As mentioned in Chapter 1.1, this thesis aimed to help Bosch Sensortec benchmark and measure the deviation of IMU sensors. However, what societal benefits can improved IMU sensors offer?

IMU sensors have a broad range of applications. They are, or can be, utilized in everything from navigation in vehicles and smartphones to robotics and biomechanics. The use cases are extensive, but the underlying principle is simple: by tracking the movement of an object, the system reacts and provides the user with some form of output. Referring back to the motivation of this thesis in Chapter 1.5, improved IMU sensors could enhance:

- Driver assistance systems and autonomous vehicles
- Navigation for drones and robots
- Indoor positioning where GPS is unavailable

All these areas of improvement directly benefit humans by enhancing either safety, automation, or potentially both. For example, driver assistance systems could become more precise without GPS signals in tunnels or high-rise building areas, leading to fewer delayed decisions in traffic and, consequently, an improvement in traffic safety. Navigation software could implement indoor navigation in shopping malls, resulting in easier world navigation. Furthermore, more precise robotics could improve accuracy in manufacturing, leading to fewer faults, less waste, and higher productivity.

Another potential application for IMU sensors lies in sports or medicine. They could be used to evaluate body movement, such as identifying potential strains based on abnormal body movement, and also for improving body movement efficiency.

All these, and numerous other use cases, represent a direct societal benefit of better IMU sensors. Since the software developed in this thesis is designed to serve as an evaluation benchmark against current sensors, it could potentially assist in achieving better accuracy and reducing sensor drift. If this goal is achieved, it would directly impact the improvement of the areas mentioned above. Therefore, by providing an easy-to-use and validated tool for benchmarking IMU sensor performance, this thesis offers a direct and practical contribution to the development of more accurate and reliable sensor technologies. The availability of such a tool can accelerate innovation in critical areas such as automotive safety, robotics, and medical diagnostics, ultimately translating into societal benefits.

Chapter 7

Conclusion

This chapter concludes the thesis by summarising the most important findings from the work carried out. The research questions defined in the problem statement are revisited and answered based on the results and discussion presented in previous chapters. The chapter then considers how the developed system contributes to the overall purpose of the thesis and identifies areas where the system could be taken further in future work.

7.1 Summary of Key Findings

This thesis successfully demonstrates the design, implementation, and validation of a low-cost optical motion capture system using consumer-grade webcams. The analysis confirms that the system is not only functional but also that its performance characteristics are well-understood and quantifiable. The principal limitations, as detailed in Section 6.3, originate from two primary sources: hardware constraints, such as camera resolution, and physical phenomena, such as marker occlusion. Crucially, the targeted solutions proposed in Section 7.4 directly address these identified bottlenecks, confirming that significant performance enhancements are achievable. This work therefore establishes a robust baseline and a clear, data-driven path toward developing a highly accurate, low-cost motion capture system. Calibration quality was identified as the primary lever for improving system accuracy, and an exploration of this is presented in Appendix A.

7.2 Answers to problem statements

This section directly addresses the research questions initially outlined in the Problem Statement in Section 1.4. Each question is presented, followed by a concise answer derived from the thesis's findings.

1. *What method can be developed to enable low-cost and low-setup-time recordings of three-dimensional trajectories in space?*

The method developed is a software system utilizing standard webcams in a three-camera setup. This approach is low-cost due to the use of readily available webcams and achieves low-setup-time through a flexible and efficient calibration process. It's designed to be easily deployable in an office environment without requiring specialized infrastructure.

2. *How can current image recognition methods be used to generally detect a pre-defined object in space?*

ArUco markers detected using OpenCV's `detectMarkers()` function are used as the primary image recognition method for identifying a predefined object in space. These markers are attached to a rigid body and provide robust visual identification through uniquely encoded patterns.

Each detected marker establishes known correspondences between two-dimensional image coordinates and three-dimensional world positions. These correspondences are then used to estimate the pose of the object.

3. *How can a trajectory be calculated using a multi-camera setup?*

Using a three-camera setup, the trajectory is calculated through three-dimensional reconstruction using triangulation based on the detected ArUco markers. The OpenCV `triangulatePoints()` method is employed to determine the object's position by combining positional data from multiple cameras. The camera with the maximized projected area for an ArUco marker is used to determine marker rotation to minimize perspective distortion.

4. *What calibration method can be introduced to effectively mitigate the challenges of camera setup and camera projections for three-dimensional trajectory recordings?*

A Two stage calibration process is used:

- (a) **Intrinsic Calibration:** Involves capturing multiple images of a chequerboard pattern from different angles and depths to determine a camera's unique internal distortions and focal properties. The improved pipeline from Section 5.1.1 enforces a recalibration loop until a sub-1 pixel re-projection error is achieved, improving consistency.
- (b) **Extrinsic Calibration:** After intrinsic calibration, time-synchronized frames of the chequerboard are used across cameras to define their positions and orientations within a shared three-dimensional coordinate system. The improved system calculates camera relations to Camera 0, making it the origin for a more stable and predictable world coordinate system.

5. *What synchronisation method can be used to synchronise separate camera recordings?*

Camera synchronisation is achieved through software timestamp matching. Each camera thread stamps every captured frame with the elapsed time since a shared system start time, with all cameras starting simultaneously via a threading barrier to ensure a common time reference. A dedicated sync thread continuously buffers incoming frames from all cameras and matches them by finding the closest frame across all cameras by timestamp. When all matches fall within a 20ms threshold, the frames are grouped into a synchronized frame group.

6. *What method can enable real-time operation of three-dimensional trajectory recording?*

Real-time operation is enabled through a pipeline that separates time-critical tasks from computationally heavy ones. Frame synchronisation runs continuously in a dedicated background thread during recording, grouping frames from all cameras within a 20 ms threshold to provide immediate feedback through the live camera display. Marker position calculations, calibration, and rigid body template computations are deferred to their respective post-recording phases, as each requires a complete dataset to produce valid results. This separation ensures the recording loop remains responsive throughout capture without being burdened by heavy processing.

7.3 Contribution to Purpose

The work from this thesis resulted in a successfully developed low-cost, low-setup-time motion tracking system using standard webcams in a three-camera setup. It uses ArUco markers for object recognition and calculates trajectories through three-dimensional reconstruction with triangulation. The system incorporates a two-stage calibration process and real-time operation through separated tasks, all of this using time-synchronized video feeds, fulfilling its core objectives as a tool for evaluating sensor tracking algorithms.

7.4 Future Development Opportunities

Section 7.2 confirmed that the core objectives of this thesis, developing a low-cost, low-setup-time motion tracking system with robust detection, triangulation, calibration, synchronization, and real-time capabilities, have been successfully met. These established methodologies form a solid foundation.

However, research inherently uncovers further possibilities for refinement and expansion. This section will explore future development opportunities and applications of the system by building directly on the insights gained from the problem statement answers.

7.4.1 Applications of System

The main application of the system is intended for office environments where more extensive motion capture systems, such as optical systems using infrared cameras, are not available. This may be due to requirements for dedicated space, permanent installation, or high cost.

By limiting the number of cameras to three and ensuring that the system can be recalibrated, the setup becomes more adaptable to different environments. Using standard webcams also reduces the overall cost of the system.

These benefits come at the cost of reduced accuracy, where more advanced motion capture systems would perform better. As a result, the system is not suitable for applications such as digital character creation in film or video games, where high-precision tracking is required and infrared-based systems are typically used.

However, the system may still be useful in scenarios where approximate ground truth or alternative motion tracking is required during development, particularly when optical tracking methods are preferred or more suitable than non-optical approaches.

7.4.2 Possible Improvements of System

Building on the system's established foundation, this section identifies the most impactful areas for future refinement. Each opportunity addresses a specific limitation encountered during development or evaluation, ranging from calibration precision and data reliability to hardware constraints and output completeness. Together, they outline a concrete roadmap for evolving the system beyond its current capabilities.

7.4.2.1 Camera Calibration

As discussed in Section 6.2.4, the initial choice of a chequerboard for camera calibration, while offering a high perceived point density, introduced a key limitation: reliable detection requires the entire board to be visible in every image, which constrained stable calibration primarily to the optical centre and led to diminished precision at the camera’s edges. ChArUco boards address this directly. Their detection mechanism allows for accurate pose estimation and corner interpolation even when the board is only partially visible, extending the effective calibration volume and enabling more accurate motion tracking across a wider field of view. As the extrinsic baseline error is the dominant term in the positional accuracy estimate derived in Section 4.7.2, improving calibration quality would be expected to reduce the worst-case accuracy bound of 37-48 mm established in that section. This improvement was explored as an extension after the primary evaluation phase and the results are presented in Appendix A.

7.4.2.2 Kalman Filter

The Kalman filter is an algorithm which can be used in various applications to provide estimates of some unknown variable over time in linear models[19]. However, a linear model is not the case in this system since every frame is processed independently. Therefore the Extended Kalman filter[20] could be applied in the system to filter marker positions and rotations, as it does not assume linearity. This could potentially reduce or eliminate positional and rotational outliers by estimating a new pose when the current pose seems to be unreasonable. Positional and rotational outliers could occur due to various factors mentioned in section 6.3.3.1.

7.4.2.3 Ground Truth

Currently, the lack of a ground truth dataset presents a significant challenge for objective system evaluation. The present assessment relies solely on visual inspection, providing only an estimated measure of performance.

The creation of a robust ground truth is a critical next step of improvement. Establishing precise real-world position, rotation, and movement data will enable direct comparison with system estimations. This foundational dataset will not only facilitate accurate calibration and ongoing development but will also provide a quantifiable measure of system accuracy, significantly enhancing the reliability and interpretability of our output data evaluation.

7.4.2.4 Drivers and Their Impacts

As mentioned in Section 6.3.2, the driver limitation was a significant setback for this thesis. However, if the cameras had been given access to the correct drivers, the system would actually have been stress-tested to its full potential. Based on the current input of three camera frames using BGR colour space at a resolution of 640×480 p, this results in 2.6 MB per frame group and 78 MB per second, assuming 30 frames per second. Considering the possibility of 4K resolution (3840×2160 p), the memory usage would jump up to 71 MB per frame group and 2.1 GB per second.

The current approach, since memory was not a significant problem at the low 640×480 p resolution, involves post-processing all frames picked from a buffer that is constantly loaded during recording. This means memory usage scales directly with recording duration, with the benefit of less processing required during recording. This design also makes it possible to display the output of all three cameras, allowing the user to ensure that the tracked rigid body remains inside the tracking volume.

An improvement to the system that would drastically change memory usage would be to remove the post-processing buffer and instead process every frame group in real-time. This approach could then discard used frames, thereby clearing up memory. This would greatly improve memory efficiency. The problems that could arise from this approach are that the display of the camera images would probably become laggy or even unusable. Another potential problem is that the processing speed of the calculations might be slower than the throughput of synchronized frame groups, resulting in frame group drops and gaps in the output data. This could possibly be solved using a multi-threaded queue for frames that arrive too quickly, which would result in growing memory usage again, but at a slower rate.

7.4.2.5 Data Interpolation

Currently, the system is dependent on the constant visibility of markers to perform triangulation and produce a positional estimate. When the system cannot detect a sufficient number of markers, frames are skipped, leading to gaps in the reconstruction data.

If these gaps in the output are small, an accurate estimation method called data interpolation could be adapted. Data interpolation works by filling in missing data points based on the data immediately before and after the gap. The smaller the missing gap, the more accurate the estimation can be assumed to be.

This technique could potentially be used in conjunction with the memory improvement proposal from Section 7.4.2.4 to fill in small gaps in the output when processing time is too slow and the buffer grows too large resulting in dropped frame groups. Data interpolation could also be used independently in scenarios where glare, motion blur, or other occurrences cause markers to be undetected.

Implementing this would result in smoother outputs and better interpretability. However, it must be noted that these interpolated values are estimations and not actual measured data. Nevertheless, an assumption that the smaller the gap, the more reliable the interpolated result will be can be made.

Disclosures

Generative AI has not been used to generate the content of this thesis. AI tools have been used only for proofreading, grammar correction, and language improvement. All intellectual content, including research, analysis, implementation, and conclusions, is the authors' original work.

Appendix A

Improvements of System

Improvements of Bosch Sensortec's Low-Cost Webcam-Based
Motion Capture System

Contents

A.1	Introduction	83
A.2	Quantitative Performance Results	83
A.3	Conclusion of System Improvements	86

A.1 Introduction

This appendix presents the results of the calibration improvement motivated by the limitation identified in Section 6.2.4, where a ChArUco-based approach was introduced as a replacement for the original chequerboard calibration described in Section 3.4.1. The goal of this improvement was to evaluate its impact on system accuracy, particularly in terms of extrinsic baseline error and re-projection error.

A.2 Quantitative Performance Results

The impact of the system improvement was quantified by comparing new measurements against the baseline data captured with the original system. Table A.1 provides a comparative summary of the key performance metrics. A visual comparison of the raw three-dimensional trajectories between the baseline and improved calibration can be seen in Figure A.1, followed by the full set of two-dimensional projections for the improved system in Figure A.2.

Table A.1: Comparison of Baseline and Improved Performance Metrics.

Sub-System	Metric	Baseline Result	Improved Result
Camera Calibration	Extrinsic Baseline Error	± 30 mm	± 10 mm
Camera Calibration	Intrinsic Re-Projection Error	$\simeq 0.9$ px	$\simeq 0.2$ px
Synchronization	Mean Time Spread	$\simeq 13.3$ ms	$\simeq 13.3$ ms
Object Detection	Detection Rate	93%	$\simeq 96\%$
3D Reconstruction	Re-Projection Error	2 - 4 px	0.8 - 1.8 px
Trajectory Tracking	Mean Coverage	92.90%	$\simeq 96\%$
Trajectory Tracking	Positional Jitter	4.82–17.55 mm	$\simeq 5$ mm

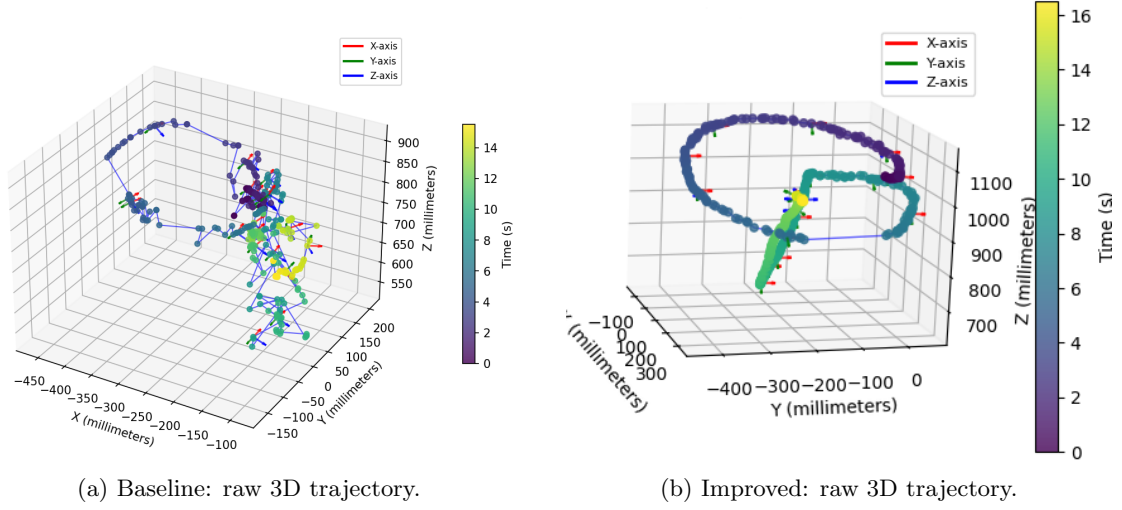
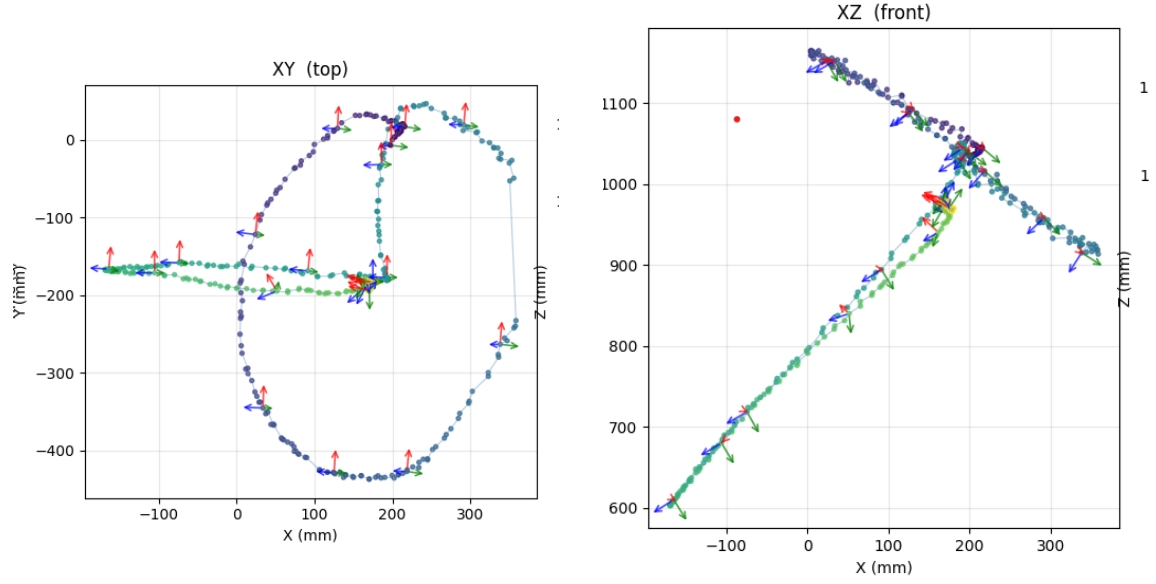
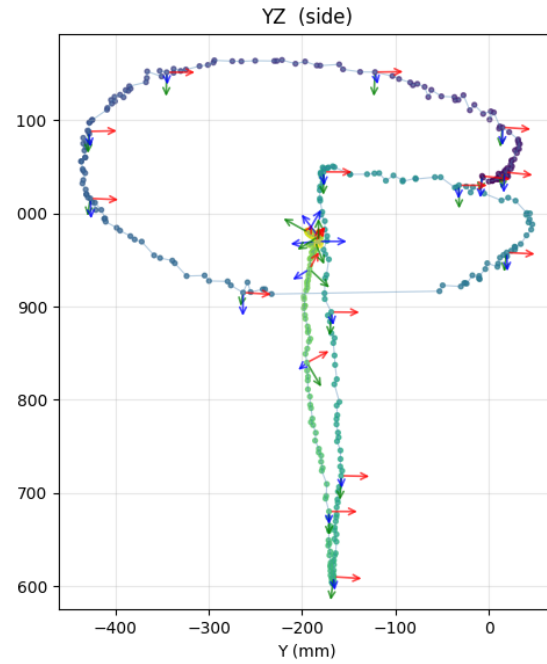


Figure A.1: Side-by-side comparison of raw three-dimensional trajectories recorded with the baseline chequerboard calibration (left) and the ChArUco-based calibration (right).



(a) XY 2D projection of the improved trajectory. (b) YZ 2D projection of the improved trajectory.



(c) XZ 2D projection of the improved trajectory.

Figure A.2: Two-dimensional projections of the improved system's trajectory onto the XY, YZ, and XZ planes.

A.3 Conclusion of System Improvements

The implementation of a ChArUco-based calibration routine resulted in a lower extrinsic baseline error, lower intrinsic re-projection error, and lower reconstruction re-projection error. These improvements in turn result in a more accurate system. Another benefit of using a ChArUco board rather than a chequerboard is that the system is able to orient itself within the board. This makes it possible to calibrate and undistort even the outer edges of the camera’s field of view.

As anticipated in Section 7.4.2.1, the reduction in extrinsic baseline error from ± 30 mm to ± 10 mm directly tightens the worst-case positional accuracy bound established in Section 4.7.2. Furthermore, the positional jitter, which ranged from 4.82-17.55 mm across marker groups in the baseline system, was reduced to approximately 5 mm across all marker groups. This indicates that the spatial non-uniformity observed in the baseline system, where markers further from the optical centre exhibited significantly higher jitter, was largely resolved by the extended calibration volume provided by the ChArUco-based approach.

The conclusion of this is that both a smaller extrinsic baseline error and a smaller re-projection error result in recorded visual representations that more closely resemble the actual movement performed.

Appendix B

Requirement Specification

System Requirements for Bosch Sensortec's Low-Cost
Webcam-Based Motion Capture System

Contents

B.1	Version History	89
B.2	Background	89
B.3	Priority Definition	89
B.4	Workgroup	90
B.5	Concept List	91
B.6	Requirements	92
	B.6.1 Functional Requirements	92
	B.6.2 Quality Requirements	94

B.1 Version History

This section represents the version history of the requirement specification in order to be fully transparent with the updates of requirements for the system during the thesis.

Table B.1: Version history of the requirement specification document.

Date	Version	Comment
2026-02-11	0.1	Initial version
2026-02-24	0.2	Updated data values
2026-05-07	0.3	Clarifications of system

B.2 Background

This document defines the requirements for the to-be-developed low-cost motion capture software for Bosch Sensortec. The goal for the development of the low-cost motion capture software is that it will assist the Bosch Sensortec Lund team in the evaluation process of their motion-sensor algorithm development. If successful, the expected outcome is less manual time in the evaluation of the algorithms and a shift of labour back to development and improvement. This will result in a more streamlined development process and reduce the total time spent on projects.

B.3 Priority Definition

With the creation of requirements for this document, the project group has decided to categorise the demands based on the necessity of each specific requirement. The scale best suited for this project, according to the project group, is the binary scale of high and low priority. High priority requirements are core functionality for the project, while low priority requirements are additional features that can be implemented if time and resources permit. Low priority requirements will be marked with **L**, whereas high priority requirements will not be marked.

B.4 Workgroup

The work group for the development of this project consists of the following people:

- Oskar Hessler
- Ebbe Persson

B.5 Concept List

This section defines and explains words and concepts used in this document.

Table B.2: Definition of document words and concepts.

Concepts	Explanation
Quaternion	A mathematical construct that represents three-dimensional rotations.
Motion capture software	The Python program that processes camera inputs and generates positional and rotational data of a motion-tracked <i>object</i> .
The system	The motion capture software together with cameras.
Motion-sensor algorithm	The algorithm in motion-tracking sensors that calculates the output.
Object	The object that is tracked in the motion capture software.
Motion recording	The capturing of motion by the motion capture software between start and stop.
Unique frame group identifier	A unique identifier of each <i>frame group</i> represented by an integer starting from 0.
Frame timestamp	The specific time a frame was captured in relation to the start time of the motion recording.
Frame group	A group consisting of the closest frame timestamps in proximity, with each input source only represented once per group.
Output data	See requirement B.6.1.4.
The project	The development of the <i>system</i> .
Rigid Body	An object or part of an object that is rigid.
Rigid body template	Pose relations between markers belonging to the same rigid body.
Rigid body centre	The mathematical centre point of the rigid body markers.
Rigid body marker	The markers to be tracked in a motion capture.
Intrinsic calibration	Models cameras internal geometry to enable precise three-dimensional reconstruction.
Excentric calibration	Determines the rotation and translation between the cameras.

B.6 Requirements

This section defines all the requirements for the system.

B.6.1 Functional Requirements

This subsection outlines the functional requirements for the system.

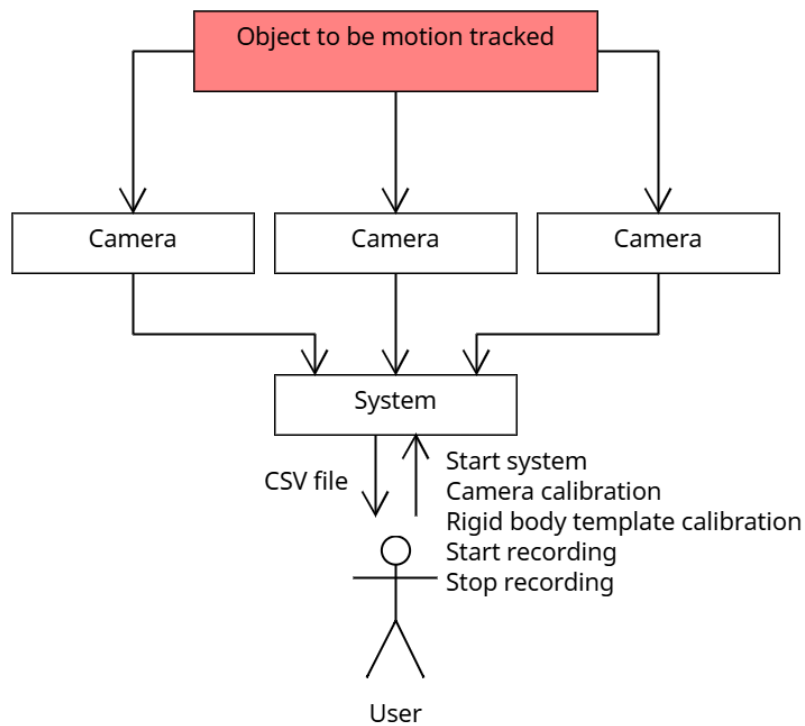


Figure B.1: Context diagram that describes how the system interacts with the surrounding domain.

Task 1: Record motion and run full calibration	
Purpose	Subtasks
The user is suppose to be able to record motion and recive output in a CSV-file.	1. Make sure the cameras are connected. 2. Start the system. 3. Choose option 2 which states: "Run full calibration (redo all phases, clears old data)". 4. Perform intrinsic camera calibration for each connected camera individually. 5. Perform extrinsic camera calibration for all connected cameras. 6. Perform rigid body template calibration. 7. Record motion. 8. Stop recording.
Prerequisites	
User have connected the cameras.	
Critical	
1. Room lighting quality that makes markers and chessboard detectable. 2. Chessboard used for calibration is put on a plane surface. 3. Markers is put on a plain surfase.	

Figure B.2: Task 1: Record motion

B.6.1.1 The system shall consist of an interface according to Figure B.1.

B.6.1.2 The system shall support the task shown in Figure B.2.

B.6.1.3 The system shall support the following actions:

- B.6.1.3.a Start motion recording.
- B.6.1.3.b Stop motion recording.
- B.6.1.3.c Camera calibration.
- B.6.1.3.d Rigid body template calibration.

B.6.1.4 The output file should be formatted as follows:

- B.6.1.4.a A CSV file with comma-separated values.
- B.6.1.4.b UTF-8 encoding.
- B.6.1.4.c A CSV file structure following:
 - i. Row 1: Metadata header.
 - ii. Row 3-7: Column headers.
 - iii. Row 8→: Data.
- B.6.1.4.d The data should be formatted as follows:
 - i. Column 1: Unique frame group identifier starting at 0 as an integer.

- ii. Column 2: Frame timestamp in relation to the recording start as a float in seconds.
- iii. Column 3 - 6: Containing quaternion data (X, Y, Z, W) representing three-dimensional orientation of the rigid body centre as floats.
- iv. Column 7-9: Containing positional data (X, Y, Z) of the rigid body centre as floats.
- v. The following columns will contain the same information as B.6.1.4(B.6.1.4.d)iii followed by B.6.1.4(B.6.1.4.d)iv for the number of rigid body markers that the system tracked during the recording.
- vi. The following columns will contain the same information as B.6.1.4(B.6.1.4.d)iii followed by B.6.1.4(B.6.1.4.d)iv for the number of markers that the system tracked during the recording.

B.6.1.5 The system should be able to capture the motion of markers placed on an object.

B.6.2 Quality Requirements

This subsection outlines the quality requirements for the system.

B.6.2.1 Frame timestamps should not vary by more than 0.02 seconds in a single frame group.

B.6.2.2 The intrinsic calibration of each webcam shall have a re-projection error smaller than 1 pixel.

B.6.2.3 Each camera shall have a extrinsic calibration with an accuracy of ± 30 mm between the computed position and the real-world counterpart.

B.6.2.4 The system should generate output data that correlates with the real-world movement of the tracked object.

References

- [1] S. Mallick. “Geometry of image formation,” Accessed: Jan. 28, 2026. [Online]. Available: <https://learnopencv.com/geometry-of-image-formation/>.
- [2] K. Hata and S. Savarese. “CS231A Course Notes 1: Camera Models,” Accessed: Jan. 28, 2026. [Online]. Available: https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf.
- [3] K. Sadekar. “Understanding Lens Distortion,” Accessed: Jan. 28, 2026. [Online]. Available: <https://learnopencv.com/understanding-lens-distortion/>.
- [4] Y. Ma, S. Soatto, J. Kořecká, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2001. Accessed: Feb. 9, 2026. [Online]. Available: https://www.eecis.udel.edu/~cer/arv/readings/old_mkss.pdf.
- [5] S. Liu, M. Yang, T. Xing, and R. Yang, *A Survey of 3D Reconstruction: The Evolution from Multi-View Geometry to NeRF and 3DGS*. MDPI, 2025, vol. 25, p. 5748. DOI: 10.3390/s25185748. Accessed: Feb. 9, 2026. [Online]. Available: <https://www.mdpi.com/1424-8220/25/18/5748>.
- [6] K. Kitani. “Triangulation [powerpoint slides].” [Online]. Available: https://www.cs.cmu.edu/~16385/s17/Slides/11.4_Triangulation.pdf.
- [7] D. R. Rousar. “Triangulation,” Accessed: Feb. 9, 2026. [Online]. Available: <https://www.sightsize.com/triangulation/>.
- [8] A. Torralba, P. Isola, and W. Freeman, *A Flexible New Technique for Camera Calibration*. Cambridge, Massachusetts: The MIT Press, 2024, ch. 44. Accessed: Feb. 10, 2026. [Online]. Available: <https://visionbook.mit.edu/multiview.html>.
- [9] C. Boncelet, “Chapter 7 - image noise models,” in *The Essential Guide to Image Processing*, A. Bovik, Ed., Boston: Academic Press, 2009, pp. 143–167, ISBN: 978-0-12-374457-9. DOI: <https://doi.org/10.1016/B978-0-12-374457-9.00007-X>. Accessed: Feb. 9, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978012374457900007X>.
- [10] S. Lee and M. Kang, “Group sparse representation for restoring blurred images with cauchy noise,” *Journal of Scientific Computing*, vol. 83, May 2020. DOI: 10.1007/s10915-020-01227-8. Accessed: Feb. 9, 2026.
- [11] M. J. B. Wilhelm Burger, “Filters,” in *Digital Image Processing: An Algorithmic Introduction using Java*. London: Springer London, 2008, pp. 87–116. Accessed: Feb. 9, 2026.
- [12] J. Singnoo and G. D. Finlayson, “Understanding the gamma adjustment of images,” in *Proceedings of the IS&T 18th Color and Imaging Conference (CIC)*, IS&T, 2010, pp. 134–139. DOI: 10.2352/CIC.2010.18.1.art00024. Accessed: Feb. 9, 2026.

- [13] C. Hu, H. Li, T. Ma, and et al., “An improved image enhancement algorithm: Radial contrast-limited adaptive histogram equalization,” *Multimedia Tools and Applications*, vol. 83, pp. 83 695–83 707, 2024. DOI: 10.1007/s11042-024-18922-5. Accessed: Feb. 9, 2026. [Online]. Available: <https://rdcu.be/e25pA>.
- [14] A. Vina. “Understanding edge detection in image processing. ultralytics. [online],” Accessed: Feb. 11, 2026. [Online]. Available: <https://www.ultralytics.com/blog/edge-detection-in-image-processing-explained>.
- [15] O. Documentation. “Detection of aruco markers. docs.opencv.org,” Accessed: Feb. 17, 2026. [Online]. Available: https://docs.opencv.org/3.4/d5/dae/tutorial_aruco_detection.html.
- [16] O. Documentation. “Detection of aruco boards,” Accessed: Feb. 17, 2026. [Online]. Available: https://docs.opencv.org/4.x/df/d4a/tutorial_charuco_detection.html.
- [17] O. Documentation. “Camera calibration and 3d reconstruction,” Accessed: May 8, 2026. [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html.
- [18] M. Moakher, “Means and averaging in the group of rotations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 5–7, 2002. DOI: 10.1137/S0895479801383877. Accessed: May 9, 2026.
- [19] Y. Kim and H. Bang, “Introduction to kalman filter and its applications,” in *Introduction and Implementations of the Kalman Filter*, F. Govaers, Ed., London: IntechOpen, 2018, ch. 2. DOI: 10.5772/intechopen.80600. Accessed: May 9, 2026. [Online]. Available: <https://doi.org/10.5772/intechopen.80600>.
- [20] ultralytics. “Extended kalman filter,” Accessed: May 9, 2026. [Online]. Available: <https://www.ultralytics.com/glossary/extended-kalman-filter-ekf>.