De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Faculteit Wetenschappen

School voor Informatietechnologie

master in de informatica

Masterthesis

Vision-Based Localisation and Mapping for Marker-Guided Handheld CNC Systems

Berne Sannen
Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Lode JORISSEN

COPROMOTOR :
Prof. dr. Nick MICHIELS

BEGELEIDER :
De heer Kristof OVERDULVE

2024
2025

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie
master in de informatica

Masterthesis

Vision-Based Localisation and Mapping for Marker-Guided Handheld CNC Systems

Berne Sannen
Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Lode JORISSEN

COPROMOTOR :
Prof. dr. Nick MICHIELS

BEGELEIDER :
De heer Kristof OVERDULVE

Universiteit Hasselt

Masterproef voorgedragen tot het behalen van de graad van master in de informatica

Vision-Based Localization and Mapping for Marker-Guided Handheld CNC Systems

Auteur:

Berne Sannen

Promotor:

Prof. dr. Lode Jorissen

Co-promotor:

Prof. dr. Nick Michiels

Begeleider(s):

Kristof Overdulve

Academiejaar 2024-2025

Abstract

This thesis presents a vision-based localization system for guiding handheld CNC tools over a workspace using ArUcoE fiducial markers. The system operates in two phases: first, a global marker map is constructed using camera observations. Second, the real-time pose of the tool is estimated relative to this map for path correction during motion. The approach relies on well-established computer vision techniques including marker detection, camera calibration, and pose estimation. ArUcoE markers are used to enhance detection accuracy, and the mapping process can optionally be refined through Bundle Adjustment.

The system is evaluated through both Unity-based simulations and real-world tests using a physical prototype. Experiments examine the effects of marker density, layout, and camera quality on mapping and tracking performance. Results show that medium-density, randomly distributed marker layouts offer a strong balance between accuracy and setup effort. These findings confirm the viability of using marker-based visual tracking for handheld CNC tools, with reliable performance demonstrated in both simulated and physical environments. The system offers a foundation for further work in vision-guided fabrication, including SLAM integration and support for other tool types.

Acknowledgments

I would like to express my sincere gratitude to my promotor, Prof. dr. Lode Jorissen, and co-promotor, Prof. dr. Nick Michiels, for their academic guidance and constructive feedback throughout the course of this thesis. Their comments and questions helped me critically evaluate the direction and assumptions of my work and ensured that the results were held to high academic standards. I greatly appreciate their availability and support.

I also want to thank my supervisor, Kristof Overdulve, for their continued support and technical insights during the development of this project. Their feedback on both implementation details and experimental methodology was instrumental in shaping the system into its final form. Their involvement was key in translating abstract ideas into a working and validated system.

I am especially grateful for the opportunity to pursue a thesis project based on my own ideas. Being able to define and develop a system that aligns so closely with my personal interests has been both motivating and rewarding. I appreciate the trust and flexibility I was given to explore this direction independently.

I would also like to thank Kirsten Mulkers for contributing to the mechanical side of the project. Even though this collaboration was outside the academic context, his input and enthusiasm played a valuable role in shaping the prototype.

Finally, I want to thank my family and close friends for their support and encouragement throughout my studies. Your presence behind the scenes helped make this thesis possible.

Berne, May 2025

Summary

This thesis presents the design, implementation, and evaluation of a vision-based localization system for a handheld CNC tool, inspired by commercial systems like the Shaper Origin. The system is designed to accurately track the position of a handheld tool over a workspace by detecting ArUco fiducial markers printed on the work surface. The ultimate goal is to enable the tool to follow a predefined path while automatically correcting small deviations caused by human operation. Unlike traditional CNC machines, which rely on rigid setups and fixed reference frames, this handheld system leverages real-time visual feedback to maintain precision during motion.

The project is divided into two main phases: (1) **map construction**, where the system creates a global reference map by registering the positions of all visible markers; and (2) **real-rime localization**, where the tool's real-time position is continuously estimated relative to this previously constructed map. This vision-based feedback loop allows the handheld tool to follow a precise trajectory, compensating for small errors in user guidance.

The core of the system is built on established computer vision algorithms, particularly those involving marker detection, camera calibration, and pose estimation. The detection of ArUco and ArUcoE markers provides reference points in each image frame, which are used to compute the camera's 3-DOF pose relative to the known map. The camera's intrinsic parameters are determined using Zhang's calibration method, and its extrinsic transformation with respect to the tool is manually measured and assumed fixed. Marker detection and pose estimation are implemented using OpenCV, while global map refinement optionally employs Bundle Adjustment (BA) to minimize reprojection error across all marker observations.

The system is evaluated in both a **Unity-based simulation** and through **real-world experiments** and a **physical prototype**. The simulation provides a noise-free environment ideal for benchmarking algorithmic performance, while the physical setup tests the system's robustness against real-world imperfections such as lens distortion, vibrations, and mechanical flex.

The experiments focus on three key performance indicators: **marker squarness**, **map accuracy** and **pose accuracy**, under a variety of marker layout conditions. Four different layouts were tested: a structured grid, and three random configurations with varying marker densities (dense, medium, sparse). Each layout was tested using enhanced ArUco markers (ArUcoE), with and without Bundle Adjustment, and using different cameras (webcam, Intel RealSense, and Unity virtual camera).

The results reveal several insights:

- ArUcoE markers significantly improve geometric detection accuracy compared to standard ArUco, especially when printed and mounted carefully.
- Medium-density random layouts often performed just as well or better than denser structured grids, especially in terms of rotational accuracy. This was attributed to increased viewpoint diversity aiding triangulation.
- Bundle Adjustment yielded structurally different maps but provided little to no numerical improvement in error metrics, likely due to implementation imperfections or sensor noise

overshadowing its potential benefit.

- Sparse layouts resulted in significant error accumulation, particularly when the number of overlapping markers per frame was insufficient to correct for previous pose drift.

Pose tracking in the simulation showed excellent results with near-perfect alignment to ground truth, confirming the validity of the algorithmic pipeline. In physical tests, although performance degraded due to real-world noise, the system still maintained reliable tracking. A final drawing experiment, where the tool was guided over a printed shape while drawing a line, showed that the system could closely follow the intended path with only minor deviations caused by mechanical limitations.

Several avenues for future improvement are proposed. Mechanically, the prototype could be upgraded with stiffer materials, better mounting, alowing for more accurate test results. Algorithmically, adaptive marker selection, and SLAM-based approaches could improve robustness. A particularly promising direction involves removing the explicit map-building step entirely, instead integrating mapping and tracking at the same time, as in traditional SLAM systems. Additionally, adapting the system to work with other types of cutting, such as a router or an handheld plasma cutter could further demonstrate its versatility in practical fabrication settings.

This thesis demonstrates that real-time, marker-based localization is a viable approach for guiding handheld CNC tools. The developed system, based on off-the-shelf components and open-source libraries, achieved reliable pose tracking in both simulation and physical tests. While challenges remain—particularly in scaling up the system for more demanding or less controlled environments—the results establish a solid foundation for further research and development in vision-guided fabrication.

Samenvatting

Deze thesis beschrijft het ontwerp, implementatie en evaluatie van een visueel lokalisatiesysteem voor een handmatig bediende CNC-machine, geïnspireerd op commerciële systemen zoals de Shaper Origin. Het systeem is ontworpen om nauwkeurig de positie van een tool te volgen binnen een werkoppervlak, door gebruik te maken van ArUco-markers die op het werkvlak zijn aangebracht. Het uiteindelijke doel is om het gereedschap een vooraf gedefinieerd pad te laten volgen, terwijl kleine afwijkingen door handmatige bediening automatisch worden gecorrigeerd. In tegenstelling tot traditionele CNC-machines, die afhankelijk zijn van een vaste opstelling en referentieframe, maakt dit handmatige systeem gebruik van realtime visuele feedback om precisie tijdens de beweging te behouden.

Het project is opgedeeld in twee hoofdfases: (1) **mapconstructie**, waarbij het systeem een globale referentiekaart opstelt door de posities van alle zichtbare markers te registreren; en (2) **realtime lokalisatie**, waarbij de positie van het gereedschap continu wordt geschat ten opzichte van deze eerder geconstrueerde kaart. Deze visuele terugkoppeling maakt het mogelijk om een nauwkeurig pad te volgen en kleine handmatige afwijkingen te compenseren.

De kern van het systeem is gebaseerd op gekende computer vision-algoritmen, met name voor markerdetectie, camerakalibratie en pose-estimation. De detectie van ArUco- en ArUcoE-markers levert referentiepunten in elk beeld, waarmee de positie van de camera ten opzichte van de kaart wordt berekend. De intrinsieke cameragegevens worden bepaald met Zhangs kalibratiemethode, terwijl de extrinsieke transformatie ten opzichte van het gereedschap handmatig wordt gemeten en als vast verondersteld. Markerherkenning en pose-estimation zijn geïmplementeerd met behulp van OpenCV, en de globale kaart kan optioneel worden verfijnd via Bundle Adjustment (BA), waarmee de reprojection error over alle markerwaarnemingen wordt geminimaliseerd.

Het systeem is geëvalueerd in zowel een **Unity-gebaseerde simulatie** als via **praktische experimenten** met een fysiek prototype. De simulatie biedt een ruisvrije omgeving die ideaal is voor het benchmarken van algoritmen, terwijl de fysieke opstelling de robuustheid test onder realistische omstandigheden zoals lensvervorming, trillingen en mechanische speling.

De experimenten richten zich op drie kernindicatoren: **marker squareness**, **kaartnauwkeurigheid** en **positie-nauwkeurigheid**, onder verschillende markerconfiguraties. Er werden vier opstellingen getest: een gestructureerd raster en drie willekeurige layouts met verschillende dichtheden (dense, medium, en sparse). Elke layout werd getest met verbeterde ArUco-markers (ArUcoE), met en zonder Bundle Adjustment, en met verschillende camera's (webcam, Intel RealSense, en een virtuele Unity-camera).

De resultaten leveren verschillende belangrijke inzichten op:

- ArUcoE-markers verbeteren de geometrische detectienauwkeurigheid aanzienlijk ten opzichte van standaard ArUco, vooral bij zorgvuldig afdrukken en monteren.
- Medium-densisty willekeurige layouts presteerden vaak net zo goed of beter dan gestructureerde rasters, vooral qua rotatienauwkeurigheid. Dit wordt toegeschreven aan meer diversiteit in kijkhoeken, wat helpt bij triangulatie.

- Bundle Adjustment leverde structureel andere kaarten op, maar gaf weinig tot geen meetbare verbetering in foutmetrics, waarschijnlijk door implementatiebeperkingen of door ruis in de sensors.
- sparse layouts veroorzaakten significante foutophoping, vooral wanneer te weinig markers per frame zichtbaar waren om voorgaande schattingsfouten te corrigeren.

De positietracking in de simulatie leverde uitstekende resultaten op met vrijwel perfecte overeenstemming met de grondwaarheid, wat de geldigheid van de algoritmische aanpak bevestigt. In fysieke tests verslechterde de nauwkeurigheid door reële ruisbronnen, maar het systeem bleef betrouwbaar volgen. In een laatste tekentest, waarbij het gereedschap een vorm op papier volgde terwijl het een lijn trok, bleek dat het systeem in staat was het gewenste pad nauwkeurig te volgen met slechts minimale afwijkingen veroorzaakt door mechanische beperkingen.

Er zijn verschillende richtingen voor toekomstig werk geïdentificeerd. Mechanisch kan het prototype verbeterd worden met stijvere materialen, betere montage en nauwkeurigere mechanica om de meetnauwkeurigheid te verhogen. Algoritmisch kunnen aanpassingen zoals adaptieve markerselectie en SLAM-gebaseerde benaderingen zorgen voor meer robuustheid. Een veelbelovende richting is het volledig schrappen van de aparte kaartconstructiefase, door mapping en tracking simultaan uit te voeren, zoals in klassieke SLAM-systemen. Daarnaast zou een uitbreiding naar andere toepassingen zoals frezen of plasmasnijden de veelzijdigheid van het systeem verder aantonen.

Deze thesis toont aan dat realtime, marker-gebaseerde lokalisatie een haalbare aanpak is voor het aansturen van handmatige CNC-gereedschappen. Het ontwikkelde systeem, opgebouwd met standaardcomponenten en open-source bibliotheken, behaalde betrouwbare positietracking in zowel simulatie als fysieke tests. Hoewel er nog uitdagingen zijn—vooral bij opschaling naar veeleisendere of minder gecontroleerde omgevingen—vormen de resultaten een stevige basis voor verdere ontwikkeling binnen visueel geleide productie.

Contents

Chapter 1

Introduction

The increase of digital fabrication has created demand for precision tools that combine the accuracy of industrial CNC systems with the flexibility of handheld operation. While stationary CNC machines achieve sub-millimeter precision through rigid mechanical structures, their fixed nature limits practical applications where portability is essential. Commercial solutions like the *Shaper Origin* demonstrate that vision-guided handheld tools can bridge this gap by augmenting human operation with real-time positional correction. However, the proprietary nature of such systems obscures critical technical details about their performance characteristics and implementation constraints.

This thesis presents the design, implementation, and validation of a custom vision-based localization system for a marker-guided handheld CNC tool, inspired by systems like the Shaper Origin. This project involves the full development of a working prototype—both hardware and software—capable of real-time pose tracking using ArUco fiducial markers. The system is then rigorously evaluated to assess its accuracy, reliability, and scalability.

The study investigates the impact of marker density, layout, and camera quality on pose estimation accuracy, as well as the trade-offs between setup complexity and tracking performance. A comprehensive validation framework combines controlled Unity-based simulations with physical experiments, allowing direct comparison between idealized and real-world conditions. The results highlight the effectiveness of the approach while also exposing its limitations in terms of marker visibility, calibration sensitivity, and mechanical stability.

The study makes three primary contributions to the field of vision-guided fabrication. First, it provides quantitative measurements of pose estimation accuracy using 2 types of markers. The standard ArUco markers —a widely used type of square fiducial marker— and an enhanced version of ArUco, the ArUcoE markers. Second, it offers an open-source implementation of the core tracking pipeline, serving as a reference for researchers and makers developing custom solutions. Third, it identifies specific optimization pathways for adapting these systems to specialized applications such as plasma cutting, where existing commercial solutions may not be suitable.

The remainder of this document is organized to systematically present the methodology and findings. Chapter 2 reviews the technical foundations of marker-based tracking and prior work in handheld CNC systems. Chapter 3 details the experimental framework and evaluation metrics. Chapter 4 presents the quantitative outcomes of the validation studies, while Chapter 5 discusses the broader implications of these findings and suggests directions for future research.

The development of a handheld CNC tool with real-time correction requires a combination of techniques from multiple fields, including robotics, computer vision, and related algorithms like bundle adjustment (BA). Unlike stationary CNC machines, which operate with pre-defined tool paths and rigid control over positioning, a handheld CNC tool introduces significant variability in movement due to human interaction. Therefore, an effective system must continuously track the tool's position relative to the workpiece and compensate for deviations in real-time.

To achieve this, several key challenges must be addressed. First, accurate camera calibration is essential to ensure that the intrinsic and extrinsic parameters of the vision system are well-defined, minimizing distortion and projection errors. Next, pose estimation techniques are used to determine the position and orientation of the camera relative to a known reference frame. However, standard pose estimation methods often suffer from noise and limited accuracy, necessitating the integration of sub pixel accuracy algorithms like bundle adjustment to increase the accuracy with each addition of a new camera angle of the same object. Furthermore, the choice of fiducial markers, such as ArUco, plays a crucial role in providing reliable feature points for tracking, with recent research focusing on improving detection robustness and subpixel accuracy.

Since this thesis falls within the field of computer science, the primary focus will be on the image processing aspects of the project. This includes the techniques used for camera calibration, pose estimation, marker detection, and the integration of these methods into a real-time correction system. While the mechanical and hardware components are essential to the overall functionality, the core contribution of this work lies in developing and optimizing the vision-based tracking and mapping algorithms.

Chapter 2

Related Work

This chapter presents a systematic examination of the multidisciplinary research foundations underlying vision-guided handheld CNC systems. The discussion progresses logically from fundamental engineering principles to advanced computer vision techniques, reflecting the developmental hierarchy of such systems. The analysis begins with mechanical implementations of position-correcting tools, which establish critical design constraints for visual tracking integration. This foundation leads to an evaluation of camera calibration methods, comparing traditional approaches like Zhang's method with contemporary alternatives designed to address real-world operational challenges.

The literature study then investigates pose estimation techniques, with particular emphasis on homography-based solutions suitable for planar workspaces. This examination reveals the inherent trade-offs between accuracy and computational efficiency in handheld implementations. Subsequent sections analyze marker-based tracking strategies, focusing on fiducial markers like ArUco and their enhancements for sub-pixel precision. These developments connect naturally to broader SLAM methodologies and bundle adjustment, demonstrating how visual odometry principles can be adapted for constrained CNC applications.

Throughout these interconnected domains, the review maintains focus on three critical evaluation criteria: theoretical foundations, implementation challenges, and performance characteristics relevant to real-time operation. Each section not only documents key contributions but also identifies specific limitations—including marker occlusion sensitivity and computational bottlenecks—that inform the experimental methodology developed in later chapters. The organizational structure intentionally mirrors the pipeline of an operational system, progressing from physical hardware considerations to algorithmic optimization, thereby providing both technical depth and practical implementation insights.

2.1 Engineering Approaches

The integration of vision-based tracking into precision machining has been an ongoing pursuit in engineering, particularly in robotics and CNC automation. Traditional CNC machines achieve their precision through rigid mechanical structures and carefully calibrated motors, eliminating most sources of error. This can be improved upon using computer vision, like the new Bambulab Vision Encoder [Bam25]. However, as systems become more flexible and adaptable, new challenges emerge in maintaining accuracy, particularly when human interaction is involved.

One notable example of a vision-assisted CNC system is presented by Rivers et al. in *Position-Correcting Tools for 2D Digital Fabrication* [RMD12]. This work introduces a handheld router and vinyl cutter that dynamically corrects user movements in real time to achieve high-precision fabrication. The system leverages a hybrid approach to positioning: the user provides coarse manual positioning by moving the tool frame along an approximate path, while the device fine-tunes the tool's position within the frame using a small, low-cost actuation system. This method combines the unlimited range of human motion with the precision of automated control, eliminating the need for expensive, large-scale multi-axis stages typically found in traditional CNC machines.

At the core of the system is a robust localization mechanism based on computer vision. The device tracks its position by detecting high-contrast fiducial markers placed on the workpiece. These markers are printed on adhesive tape for easy application in arbitrary patterns across the material. The system initially used rectangular Micro QR codes (rMQR) for this purpose, but in later stages of development, these were replaced by a custom-designed fiducial marker. Each of these rMQR markers consists of a unique box-within-box "anchor" pattern and a 2D barcode, enabling reliable identification and spatial referencing. The camera mounted on the tool captures these markers, and the system stitches together a 2D map of the workspace during an initial scanning phase. This map-building process involves image registration techniques to align overlapping frames and create a cohesive representation of the material. Once the map is generated, the device localizes itself in real time by matching detected markers to the pre-scanned map, achieving an accuracy of within $0.22mm$ on average.

The actuation system is another critical innovation. Instead of traditional linear stages, the device employs eccentrics (rotating disks with off-center shafts) to convert rotational motion into precise linear adjustments. This design provides low-backlash, high-accuracy positioning over a small range (up to $1.5cm$ diameter), sufficient to compensate for typical human positioning errors. The linkage is driven by stepper motors, which are powerful enough to resist backdriving even under significant material forces, such as those encountered during routing.

The system supports two motion strategies tailored to different applications:

- **Constant-speed motion**: Used for tasks like routing, where maintaining a consistent feed rate is crucial to avoid material damage. The tool follows the path autonomously, while the user adjusts the frame to keep the tool within its correction range.
- **Freeform motion**: Ideal for applications like vinyl cutting or plotting, where the user moves the frame freely, and the device dynamically selects the most appropriate path to stay on the plan, avoiding interior material and prioritizing feature fidelity.

A user interface with a live display guides the operator, showing the tool's position relative to the plan and the correction range. This feedback ensures that even complex shapes can be fabricated accurately, as the user can adjust their movements in real time.

Since its introduction, this research concept has evolved into the commercially available *Shaper Origin*, demonstrating the viability of vision-assisted handheld CNC machining. The success of the system lies in its clever integration of computer vision, mechanical design, and user interaction, making high-precision fabrication accessible without the constraints of traditional CNC setups.

In robotics and digital fabrication, achieving high precision often hinges on robust calibration methods, especially for systems operating in large or dynamic workspaces. Traditional approaches rely on fixed reference points or expensive external metrology tools like laser trackers, but recent advances leverage vision-based techniques to calibrate robotic systems directly. For example, Yuanhao Yin et al. [Yin+24] propose a method where a monocular camera mounted on a robot's end-effector captures ArUco markers distributed across the workspace. By constructing an optimized ArUco map through closed-loop detection and global bundle adjustment, their system calibrates the robot's kinematic parameters, reducing absolute positioning errors from 1.359 mm to 0.472 mm. Crucially, this approach corrects the robot's pose by refining its Denavit-Hartenberg model, demonstrating how fiducial markers can expand calibration ranges while maintaining sub-millimeter accuracy.

A similar principle applies to consumer-grade systems like Bambu Lab's 3D printers, which use a Charuco-like pattern to calibrate the printer's mechanical alignment and compensate for wear over time. Here, the camera scans the print bed to detect deviations in nozzle height or belt tension, dynamically adjusting the printer's motion system rather than recalibrating the camera. This mirrors the industrial need for self-correcting systems that adapt to mechanical degradation without manual intervention. Both examples highlight a broader trend: fiducial markers (ArUco, Charuco) enable scalable, vision-assisted calibration of mechanical systems by treating the camera as a passive observer and the robot or printer as the active target of correction.

These challenges highlight the need for an approach that blends the adaptability of vision-based tracking with the reliability of precise calibration techniques. Unlike rigid industrial setups, this approach can function in a less controlled environment, requiring continuous recalibration and real-time adjustments.

2.2 Camera Calibration

Achieving high-precision position tracking in a vision-based system starts with accurate camera calibration. Any errors in the camera model—whether from lens distortion, misalignment, or inaccurate focal length estimation—propagate through every stage of the pipeline, leading to significant deviations in the final position estimation. In traditional machine vision applications, such as industrial inspection or robotic automation, cameras are often rigidly mounted in fixed positions, allowing for one-time calibration with minimal drift. However, in this project, where the camera moves relative to the workpiece, calibration must be both precise and adaptable.

One of the most widely used methods for intrinsic camera calibration is Zhang's Method [Zha00], which estimates camera parameters by observing a known planar pattern, such as a checkerboard, from multiple angles. This technique provides a robust solution for modelling lens distortions and computing the intrinsic matrix, making it a natural choice for this project. While Zhang's Method is well-established and widely applied, it assumes that the calibration pattern is perfectly flat and that image noise is minimal. Assumptions that may not always hold in real-world conditions.

To address some of these limitations, alternative approaches have been proposed. Model-based methods, such as those explored in mrcal [Kog], provide enhanced accuracy by incorporating more sophisticated distortion models and improving stability in low-quality image conditions.

In addition to intrinsic calibration, extrinsic calibration—determining the camera's position and orientation relative to the workspace—is equally crucial. In this project, the camera is mounted at a fixed height and angle relative to the work surface, but its exact position must still be calculated to ensure precise tracking. The approach by Yuanhao Yin et al. [Yin+24] addresses similar challenges, refining extrinsic parameters to align a robotic arm with its actual physical workspace.

A core challenge in extrinsic calibration lies in ensuring that the estimated homography matrix (H) accurately maps the camera's view onto a predefined workspace, particularly when multiple fiducial markers are present but their relative positions are unknown. Ondrašovič and Tarábek [OT21] address this by proposing a homography ranking method that systematically selects the optimal H from multiple candidate matrices derived from distinct markers on the same plane. Their approach operates under two key assumptions:

- **Geometric similarity:** All markers differ only by translation, rotation, and uniform scaling (similarity transformations) in the world frame.
- **Planar constraint:** All markers lie on the same planar surface.

The method works as a post-processing step for existing homography estimation techniques (e.g., RANSAC-based methods in OpenCV). For each marker, the algorithm computes a score function that quantifies the reprojection error across the entire image. This function rectifies all markers using a candidate H, then applies optimal similarity transformations to align auxiliary markers with the reference template. The Frobenius norm of the residual errors between transformed and target keypoints serves as the ranking criterion.

Crucially, the approach does not require knowledge of the markers' relative positions—a limitation of traditional methods. Instead, it exploits the geometric consistency of similarity transformations under perspective rectification. Experimental results demonstrate a 60% relative improvement in reprojection accuracy compared to random homography selection, with sub-pixel errors and robustness to noisy keypoint correspondences. A key limitation is the algorithm's quadratic complexity to the number of markers. Making it a very computationally expensive algorithm when using a high number of markers.

2.3 Pose Estimation and Camera Positioning

In recent years, marker-based systems have gained significant attention for camera pose estimation, particularly in indoor environments where traditional feature-based methods may struggle due to limited environmental features or sparse overlapping views. Two notable works in this domain are the papers by García-Ruiz et al. [Gar+24; Gar+25]. Both systems leverage fiducial markers, such as ArUco markers, to estimate the pose of fixed cameras, but they differ in their approach to handling connecting views and their optimization strategies. The former uses a dense network of cameras with overlapping views, while the latter employs a sparse network of cameras and an auxiliary camera to determine the spatial relationships between them. These works provide valuable insights into the challenges and solutions for marker-based camera positioning.

Another significant contribution to this field is the work by Chunhui Zhao et al. [Zha+17], which presents a multi-camera system designed to estimate the pose of a camera cluster in real-time. Their approach leverages overlapping fields of view and rigid constraints between cameras to achieve high accuracy in dynamic environments. The system employs a combination of multi-camera calibration, feature tracking using optical flow, and local bundle adjustment to optimize camera poses and 3D point clouds. The experimental results demonstrate that the system can estimate camera poses with high accuracy. This work highlights the importance of overlapping views and rigid constraints in multi-camera systems, which can be adapted to enhance the accuracy and robustness of pose estimation in single moving camera systems.

Although these systems use multiple cameras to observe the same markers simultaneously, the underlying principles are directly applicable to scenarios where a single moving camera observes a static marker field over time, as is the case in this thesis.

2.3.1 Marker-Based Systems and Overlapping Views

Both systems rely on fiducial markers to establish correspondences between cameras and estimate their poses. The dense camera network position estimation by Pablo García-Ruiz et al. is a method for large-scale indoor camera positioning using a small subset of markers that are strategically placed in areas visible to multiple cameras. The markers are moved iteratively to connect all cameras, creating a global graph of camera poses. This approach assumes that cameras have at least some overlapping views, allowing the markers to serve as common reference points for pairwise camera pose estimation. The system then performs a full optimization to minimize reprojection errors and enforce physical constraints, such as coplanarity of cameras and markers.

In contrast, their sparse camera network approach addresses the more challenging scenario of positioning cameras with no overlapping views. The authors introduce a novel methodology that uses an auxiliary mobile camera to progressively connect fixed cameras by strategically placing and relocating markers. This approach does not require overlapping views between fixed cameras, making it suitable for large and complex environments. The system employs a comprehensive optimization process that minimizes reprojection errors while applying constraints such as camera and marker coplanarity and the use of control points.

The presence or absence of overlapping views has a significant impact on the accuracy of camera pose estimation. In the dense approach, the reliance on overlapping views allows the system to establish direct correspondences between cameras, leading to more accurate pairwise pose estimation. However, this approach is limited to environments where cameras can be positioned to share at least some overlapping fields of view.

On the other hand, the sparse method demonstrates that accurate camera pose estimation is still possible even in the absence of overlapping views. By using a mobile camera to capture markers from different perspectives, the system can indirectly connect fixed cameras without requiring them to share a common field of view. This approach is particularly useful in large-scale environments where overlapping views are impractical. However, the lack of direct correspondences between fixed cameras introduces additional complexity, which is addressed through a robust optimization process that minimizes reprojection errors and enforces physical constraints.

Both systems employ graph-based optimization techniques to refine camera poses. With the overlapping cameras, the optimization process begins with local pose graphs for groups of cameras with overlapping views. These local graphs are then merged into a global graph, and a full optimization is performed to minimize reprojection errors and enforce constraints such as coplanarity of cameras and markers. The use of control points further enhances the accuracy of the reconstruction by aligning the camera poses with a known reference system.

Similarly, the sparse approach uses a meta-graph to fuse all visual information into a single optimization process. The system first estimates the poses of markers and cameras within local groups and then connects these groups through common markers. The global optimization process minimizes reprojection errors while applying constraints such as coplanarity and control points. This approach ensures that the system can handle large-scale environments with no overlapping views while maintaining high accuracy.

The works by García-Ruiz et al. [Gar+24; Gar+25] provide valuable insights into the use of fiducial markers and graph optimization for camera pose estimation. While the fixed camera systems address different challenges related to pose estimation of cameras, the principles of marker-based positioning and optimization can be applied to a mono camera system. The key difference lies in the ability of the camera to capture markers from multiple perspectives at different times. The principles of marker-based positioning are advantageous for mono-camera systems when the camera is controllably moved to capture overlapping views. By ensuring sufficient overlap between successive frames (typically 60-80% as recommended in the dense approach)

2.3.2 Multi-Camera Pose Estimation

Beyond single-camera positioning, integrating multiple viewpoints can significantly enhance the robustness of pose estimation by reducing ambiguities and improving depth perception. Chunhui Zhao et al. [Zha+17] present a multi-camera system designed to estimate the pose of a camera cluster in real-time, leveraging overlapping fields of view and rigid constraints between cameras. Their approach is particularly relevant to applications requiring high accuracy in dynamic environments, such as autonomous navigation and 3D reconstruction. While this project currently relies on a single moving camera, the core principles of multi-camera systems can be adapted to enhance the accuracy and reliability of pose estimation in a single-camera setup.

They propose a multi-camera platform consisting of three cameras: a main camera and two auxiliary cameras (left and right) with overlapping fields of view. The system is designed to estimate the pose of the camera cluster by leveraging the rigid constraints between the cameras and the overlapping views. The system begins with a calibration phase, where the intrinsic parameters and relative poses of the cameras are determined using a checkerboard pattern. This calibration provides a priori information about the rigid constraints between the cameras, which is used to refine the pose estimation during tracking. The initialization phase extracts feature points and generates an initial 3D point cloud, with the main camera's coordinate system serving as the world reference. The pose of the camera cluster is then estimated using the Perspective-n-Point (PnP) algorithm.

During the tracking phase, feature points are tracked across frames using optical flow, and the pose of the camera cluster is estimated using the PnP algorithm. The rigid constraints between the cameras are enforced to reduce errors caused by inaccurate 3D information. Newly generated 3D points are managed using a Kalman filter to reduce tracking errors, and key frames are selected based on the density of 3D points. Local bundle adjustment (BA) is performed every five key frames to optimize the camera poses and 3D point cloud. The system uses the g2o library to perform local BA, minimizing the reprojection error across the three cameras. This optimization ensures that the camera poses and 3D points are consistent with the observed features. The experimental results demonstrate that the system can estimate camera poses with high accuracy, achieving an average translation error of 0.26 cm in the x and z directions for a baseline length of 45 cm. The system's robustness is attributed to the rigid constraints between cameras and the use of overlapping views, which provide additional geometric constraints for pose estimation.

While their approach focuses on a multi-camera system with overlapping views, the principles can be adapted to a single moving camera system. In a single-camera setup, the camera's movement through the environment effectively simulates a multi-camera system, as each new position provides a different perspective on the scene. By leveraging similar principles—such as combining observations from different viewpoints to refine pose estimation—this approach can enhance tracking reliability and improve accuracy across the entire workspace. In a single moving camera system, the camera's trajectory can be treated as a sequence of virtual camera positions. By enforcing constraints between these virtual positions, such as assuming smooth motion or known motion models, the system can reduce pose estimation errors, similar to how rigid constraints are used in their multi-camera system.

The use of optical flow for feature tracking and bundle adjustment for pose optimization can be directly applied to a single moving camera. By tracking features across frames and performing local BA, the system can maintain accurate pose estimates over time, even in the absence of overlapping views. One of the challenges in single-camera systems is scale ambiguity, which arises because a single camera cannot directly measure depth. They address this issue by using overlapping views and rigid constraints to maintain scale consistency. In a single moving camera system, scale ambiguity can be mitigated by incorporating additional sensors, such as IMUs, or by using known dimensions of the environment, such as the size of fiducial markers.

The key frame policy and 3D point management techniques used in Zhao et al.'s system can also be applied to a single moving camera. By selecting key frames based on the density of 3D points and optimizing the point cloud using techniques like Kalman filtering, the system can maintain a consistent and accurate map of the environment. This approach ensures that the system can handle dynamic environments and maintain robust pose estimation over extended periods.

2.4 Fiducial Markers and ArUco Extensions

Fiducial markers are engineered visual patterns used for reliable detection and pose estimation in computer vision applications. These markers are designed to be robust to perspective distortion, partial occlusion, and lighting changes, making them especially valuable in robotics, augmented reality, and calibration tasks. Among the most commonly used families are ArUco, AprilTag, and ChArUco, each offering different trade-offs in terms of detection speed, precision, and encoding capacity.

The ArUco marker system was introduced by Garrido-Jurado et al. [Gar+14], who proposed a method for the automatic generation and robust detection of binary square fiducial markers, even under conditions of partial occlusion. Their work focused on maximizing inter-marker distance in Hamming space to reduce false positives and improve detection reliability. A key contribution of the paper is the use of error correction capabilities within the marker dictionary, allowing the system to correctly identify markers even when parts of the pattern are obscured or degraded. The authors also provide a fast detection algorithm based on contour extraction and bit decoding, which enables real-time performance. This balance of efficiency and robustness has made ArUco one of the most widely used fiducial marker systems in the computer vision and robotics communities.

ArUco markers are widely adopted due to their simple black-and-white square design, high detection speed, and support in popular vision libraries like OpenCV. Each marker encodes a binary ID using a 2D barcode pattern and includes a black border to facilitate segmentation. ArUco markers have been used extensively in real-time camera tracking, robot localization, and augmented reality systems.

[Ked+21] introduces an extended version of ArUco, called ArUcoE, which incorporates a checkerboard-like pattern around the original ArUco marker. This enhancement is designed to improve the accuracy of 6DoF (6 Degrees of Freedom) pose estimation by providing additional features for corner detection. The key innovation of ArUcoE lies in its rectangular enhancement patterns, which are placed around the original ArUco marker but leave the corners white, creating a chessboard-like structure. This design allows for more precise localization of the marker's corners, which are critical for accurate pose estimation.

This checkerboard like pattern allows for a subpixel algorithm (OpenCV's `cornerSubPix`) to refine the corner locations within a pixel. This technique uses the gradient of neighboring pixel values to estimate the exact position of the corner, overcoming the limitation of pixel-level accuracy. Additionally, a deep learning-based super-resolution method is applied to artificially increase the resolution of the marker's corner regions. The super-resolution model is an 11-layer convolutional neural network (CNN) trained on a dataset of 44,832 corner images. The model uses mean square error (MSE) as the loss function and is applied only to regions of interest, making it computationally efficient.

The authors evaluate ArUcoE in both simulated and real-world environments. In the simulated environment, created using Blender, ArUcoE with subpixel refinement achieves remarkable accuracy, with translation errors in the order of 10 µm or 1/100 mm. Rotation errors are also significantly reduced, to the order of 0.1°. These results demonstrate the effectiveness of the checkerboard-like enhancement and subpixel refinement in improving pose estimation accuracy.

In the real-world environment, the authors use an industrial robot arm equipped with ArUcoE markers and a high-precision laser tracker for ground truth measurements. The results confirm the superiority of ArUcoE over the standard ArUco marker, with translation errors in the order of 100 µm for stereo images and approximately 2-5 times as much for mono images. Rotation errors are similarly low, in the order of 0.10° for stereo and again approximately 2-5 times as much for a mono camera setup. The use of a stereo camera setup further enhances accuracy, as it provides additional depth information, making ArUcoE particularly suitable for high-precision applications such as neutron diffractometers and medical robotics.

The authors also compare the performance of the subpixel algorithm with the deep learning-based super-resolution method. While both techniques improve accuracy, the subpixel algorithm is found to be slightly more accurate and computationally cheaper. However, the authors suggest that further research into super-resolution methods is warranted, as they could enable the detection of smaller markers with high accuracy.

AprilTag markers were introduced by Olson et al. [Ols11] as a robust fiducial marker system designed for accurate and reliable camera pose estimation in challenging environments. Unlike simpler systems such as ArUco, AprilTags use a family of binary square markers with strong error correction based on a Hamming code structure, enabling reliable detection even under partial occlusion, motion blur, or low contrast. The paper presents a highly efficient detection algorithm that extracts candidate quads through edge detection and line fitting, followed by decoding and pose estimation using homography. One of the key innovations is AprilTag's use of a unique inner coding region surrounded by a thick black border, which simplifies detection while minimizing false positives. The authors demonstrate that AprilTags outperform other marker systems like ARToolKit and reacTIVision in terms of detection accuracy and robustness, especially at longer distances or under poor lighting. These properties have made AprilTags a popular choice in applications such as robotics, SLAM, and augmented reality, particularly in outdoor or high-speed environments.

A comprehensive comparison between fiducial marker systems is presented in the work of Wang et al. [WO16], where the authors evaluate the performance of AprilTag, ArUco, and other systems based on detection speed, accuracy, and robustness to noise, occlusion, and blur. AprilTags consistently outperform ArUco markers in terms of robustness under challenging imaging conditions, such as motion blur, lighting variation, and partial occlusion. This is primarily due to their use of more advanced error correction codes and a more distinctive internal structure. AprilTags are also known for lower false positive rates, making them preferable in critical or safety-sensitive applications. However, these benefits come at the cost of higher computational overhead, especially in dense or high-resolution image streams.

In contrast, ArUco markers offer faster detection and simpler integration via OpenCV, making them a practical choice for lightweight, real-time applications such as augmented reality or tool tracking. While less robust in degraded conditions, ArUco's speed and ease of use make it well-suited for prototyping and embedded systems. ChArUco boards, a hybrid of chessboard and ArUco markers, provide highly accurate corner detection with identifiable marker IDs, making them especially useful for camera calibration. However, they are more complex to print and detect in cluttered scenes and are less commonly used for dynamic tracking tasks.

The marker system proposed by Jorissen et al. [Jor+14] replaces conventional multiple-marker setups with a single structured grid of spacially coded fiducial dots, enabling robust tracking in occlusion-prone environments. The approach first detects individual dots via adaptive thresholding, then reconstructs grid lines through nearest-neighbor clustering. Then the lines can be identified using the spatial coding. Once identified these lines can be used to identify each of the dots. This grid topology allows persistent identification of dots even when partially or significantly occluded. Once initialized, optical flow tracking maintains pose estimation during close-range camera movements, where traditional marker systems typically fail. By combining this unified grid representation with frame-to-frame motion estimation, the method achieves superior stability compared to discrete marker arrays, particularly in scenarios requiring close-proximity operation or facing intermittent obstructions.

Ondrašovič and Tarábek [OT21] propose a method for ranking homographies derived from multiple markers placed on the same plane. This approach is particularly useful in scenarios where the relative positions of the markers are unknown. The authors introduce a score function to evaluate the reprojection accuracy of each homography, allowing for the systematic selection of the best homography for image rectification. The method demonstrates a 60% relative improvement in reprojection accuracy compared to random selection, making it a robust extension to existing homography estimation techniques. This approach is especially beneficial in applications like bird's-eye view generation and road surveillance, where multiple small markers are used.

2.5 Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping (SLAM) enables autonomous systems to navigate unknown environments through an iterative process of sensor data fusion, pose estimation, and environmental reconstruction. The SLAM pipeline typically begins with front-end processing where sensors such as LiDAR or cameras extract features from the environment, which are then matched across consecutive frames to estimate relative motion through techniques like Iterative Closest Point (ICP) or feature-based methods. The back-end optimization refines these estimates by constructing a pose graph that minimizes cumulative errors, often incorporating loop closure detection to correct long-term drift. Modern implementations leverage probabilistic formulations to handle sensor noise and environmental uncertainties, with recent advances incorporating deep learning for improved feature extraction and data association.

SLAM provides robotic systems with situational awareness, it creates a map of an unknown environment while navigating through said environment. This is particularly benefitial if there is no GPS available. This capability stems from fusing diverse sensor inputs - such as LiDAR point clouds, visual features from cameras, inertial measurements (IMU), or wheel odometry - through probabilistic frameworks that estimate both robot trajectory and environmental features or landmarks. However, SLAM systems fundamentally assume a static environment, as any movement of mapped landmarks (e.g., relocated furniture or passing pedestrians) introduces persistent localization errors that propagate through the entire map estimation process. The technology additionally struggles in feature-deficient environments where sensor data provides insufficient distinctive cues: LiDAR-based systems fail in geometrically uniform spaces like long empty corridors, while visual SLAM degrades when facing textureless surfaces or repetitive patterns. Though recent approaches show promise in minimizing these limitations.

The work of Prentice et al. [RB05] serves as a highly accessible yet thorough introduction to the foundational principles of SLAM. The paper systematically explains core SLAM components, including sensor and motion models, probabilistic filtering techniques, and landmark extraction, making it particularly valuable for readers new to the field. Its strength lies in its clear, step-by-step breakdown of key concepts such as state estimation, feature association, and basic mapping, ensuring a strong conceptual foundation before advancing to more complex topics. However, the paper primarily focuses on the essential building blocks of SLAM and doesn't cover some more advanced aspects such as loop closure detection, graph optimization, or modern scalable SLAM systems. Despite these limitations, its structured and intuitive presentation makes it an excellent starting point for understanding SLAM's fundamental mechanics. Future enhancements could extend its scope to incorporate contemporary advancements, but as an introductory resource, it remains widely recommended for its clarity and didactic effectiveness.

2.5.1 LiDAR SLAM Approaches

For LiDAR-based systems, Li et al. [Li+24] propose an optimized 2D SLAM pipeline leveraging Weighted Signed Distance Function (WSDF) maps to enhance scan matching accuracy in geometrically sparse environments. Their method improves the Cartographer algorithm by integrating adaptive feature weighting and Levenberg-Marquardt (LM) optimization, demonstrating significant error reductions both indoor and outdoor compared to conventional approaches.

The WSDF map representation addresses noise in occupancy grids by weighting updates based on proximity to observed features, drawing inspiration from Truncated Signed Distance Function (TSDF) techniques in 3D reconstruction. This approach mitigates Gaussian sensor noise while preserving map consistency. The LM optimization, applied to scan matching, refines pose estimation by dynamically adjusting trust-region parameters, reducing convergence issues inherent in Gauss-Newton methods.

This work is particularly relevant for implementations constrained to 2D environments, as it avoids computational overhead from 3D point cloud processing. The LM optimization's iterative refinement shares conceptual parallels with bundle adjustment, though it operates on LiDAR scan residuals rather than visual feature reprojections. The adaptive feature weighting strategy also provides insights for handling degenerate geometries (e.g., long corridors) by prioritizing stable features during matching. However, the method does not explicitly address loop closure, relying instead on Cartographer's existing graph-based backend.

2.5.2 Visual SLAM Approaches

Visual SLAM systems depend on distinctive visual features for reliable camera tracking and map reconstruction. The ORB (Oriented FAST and Rotated BRIEF) feature detector and descriptor [MMT15] has emerged as a particularly effective choice for SLAM applications due to its computational efficiency and robustness to viewpoint changes. By combining the FAST keypoint detector with a rotation-aware BRIEF descriptor, ORB features achieve rotation invariance while maintaining real-time performance, making them well-suited for SLAM systems operating under varying camera motions and lighting conditions.

The ORB-SLAM family of systems [MMT15; MT17; Cam+21] demonstrates the effectiveness of ORB features in visual SLAM. These systems employ a parallel architecture with three key components: tracking for frame-to-frame motion estimation, local mapping for bundle adjustment and keyframe management, and loop closing for global consistency. The efficiency of ORB features enables real-time operation on standard CPUs, while their distinctiveness supports robust place recognition through bag-of-words approaches like DBoW2 [MT17]. This combination of efficiency and recognition capability has made ORB-based SLAM systems particularly versatile for various applications.

The primary advantages of ORB-based SLAM systems include computational efficiency suitable for real-time operation, rotation invariance that improves robustness to camera motions, binary descriptors that enable fast feature matching, and scalability through the bag-of-words approach for loop closure detection. However, these systems face several limitations, including degraded performance in low-texture environments where few features can be extracted, sensitivity to motion blur due to reliance on feature tracking, and scale ambiguity in monocular configurations.

Building upon this foundation, Mur-Atal et al. [MT17] systematically addressed these limitations in ORB-SLAM2. They introduced support for stereo and RGB-D cameras, effectively resolving the scale ambiguity problem in monocular SLAM while maintaining the computational advantages of ORB features. Further improvements in ORB-SLAM3 [Cam+21] incorporated visual-inertial odometry to enhance robustness in challenging conditions and introduced an innovative atlas system for multi-session mapping, significantly extending the system's applicability to long-term operations. Comparative studies [MM21] have demonstrated that while direct methods may offer advantages in textureless environments, feature-based approaches like ORB-SLAM maintain superior performance in most general scenarios due to their robust relocalization capabilities and efficient map reuse.

Munoz-Salinas *et al.* [PH19] proposed UcoSLAM, which combines these natural features (keypoints) with artificial fiducial markers such as ArUco to mitigate the limitations of purely visual SLAM in textureless environments. Their hybrid approach achieves scale-aware mapping when markers are detected, while maintaining functionality in marker-free areas through keypoint tracking. While requiring significant environmental preparation, this approach mitigates some of the cons of visual SLAM and where the textureless areas should regain its detectability by adding some markers to those textureless areas. The markers also help in detecting the scale of the features whereas only natural features struggle with this. The technology works best in stable environments where markers can remain in fixed positions over time.

2.6 Bundle Adjustment

Bundle adjustment (BA) is a non-linear optimization technique widely used in *Structure from Motion* (SfM), *Simultaneous Localization and Mapping* (SLAM), and photogrammetry to refine 3D scene structure and camera parameters by minimizing the reprojection error between observed and predicted image points. Given multiple images of the same scene, BA optimizes 3D point positions, camera poses, and intrinsic parameters by looking for the minimum error between the current "known" 3D world coordinate of a keypoint which gets projected into the camera plane, and the position of the observed keypoint as observed by the camera. This happens iteratively, constantly improving the world coordinates of said keypoints and the position of the camera untill this reaches a minimum. Its strengths include high accuracy (via iterative methods like Levenberg-Marquardt), flexibility across camera models, and robustness to outliers when combined with techniques like RANSAC or robust loss functions. BA is fundamental in applications such as 3D reconstruction (e.g., COLMAP, OpenMVG), SLAM systems (e.g., ORB-SLAM), and aerial mapping (e.g., Pix4D). However, BA has limitations, including high computational cost (scaling cubically ($O(n^3)$) with point density), sensitivity to initial estimates, and thus requires an already good estimate to start off. And it is very susceptible to local minima.

2.6.1 Bundle Adjustment: Foundations and Scalability

Bundle adjustment, as formalized by Triggs et al. [Tri+00], represents the gold standard for jointly optimizing 3D structure and camera parameters through nonlinear least-squares minimization of reprojection errors. Their work established BA as a maximum likelihood estimation problem, employing the Levenberg-Marquardt algorithm with sparse Schur complement techniques to handle large-scale systems efficiently. A key innovation was the integration of robust kernels (e.g., Huber loss) to mitigate outlier influence. However, Triggs et al. identified critical limitations: the computational complexity scales cubically with the number of parameters, and convergence relies heavily on accurate initial estimates, making BA unsuitable for large-scale reconstructions. Addressing these challenges, Agarwal et al. [Aga+09] pioneered distributed BA, introducing a hierarchical approach that partitions images into subgraphs, solves localized BA problems independently, and merges results via similarity transforms. Their system, leveraging parallelization on GPUs and visibility-based sparsification, achieved unprecedented scalability—reconstructing 150,000 images of Rome within a day, while maintaining centimeter-level accuracy. Yet, Agarwal et al.'s method inherited fundamental BA limitations: the partitioning strategy risks accumulating alignment errors across subgraphs, and the approach remains sensitive to poorly conditioned configurations. Together, these papers laid the theoretical and practical groundwork for modern BA, though their trade-offs between accuracy, scalability, and robustness continue to motivate research in incremental and hybrid optimization techniques.

2.6.2 Applications and Innovations in Bundle Adjustment

The Rome in a Day project by Agarwal et al. [Aga+09] addressed the challenge of reconstructing large-scale 3D structures from unstructured photo collections sourced from online platforms. Since such images lack predefined camera parameters or scene constraints, the authors employed structure-from-motion (SfM) techniques, relying heavily on BA to optimize camera poses and 3D point locations. However, due to the sheer volume of available images—and consequently, an extremely large number of reconstructed points—traditional BA methods would have been computationally infeasible within the desired time constraints. The optimization previously discussed was crucial in handling the massive, unstructured dataset efficiently.

Mur-Atal et al. Mur-Artal, Montiel, and Tardós [MMT15] leverage bundle adjustment as a core component of their ORB-SLAM pipeline to achieve real-time, high-precision camera tracking and mapping. Their system employs a hierarchical BA strategy: (1) *motion-only BA* for lightweight camera pose refinement during tracking (optimizing only the current camera pose while keeping 3D points fixed), and (2) *local BA* for jointly refining a window of recent camera poses and observed map points, significantly reducing drift. To mitigate BA's computational cost, they introduce a *covisibility graph* that limits optimization to keyframes sharing observed points, reducing the problem size by 60–80% compared to global BA. However, this approach inherits BA's sensitivity to incorrect feature matches; ORB-SLAM addresses this by integrating RANSAC at multiple stages and maintaining a redundancy-based outlier rejection mechanism. Despite these optimizations, the system struggles in textureless environments where feature extraction fails—a limitation later addressed by hybrid approaches combining geometric and learning-based features.

Collectively, these works demonstrate how BA's flexibility allows integration with complementary techniques (covisibility graphs, fiducial markers, hybrid optimization) to address its core limitations: computational complexity, initialization sensitivity, and environmental dependencies. However, they also reveal enduring trade-offs—between accuracy and preparation effort (marker-based systems), or between real-time performance and robustness (SLAM systems)—that continue to motivate research in learned BA and sensor fusion approaches.

Chapter 3

Methodology

This chapter presents the methodology for developing a vision-based localization and mapping system designed to accurately track the position of a handheld tool (e.g., router bit, pen, engraver, ...) relative to a workspace. The goal is to enable the tool to perform small, automated corrective movements in real time, compensating for deviations between the user's rough input and the desired path.

In typical use, the human operator guides the tool along a predefined figure—such as a cut path or drawing—by roughly tracing its shape. However, it is difficult to do this both quickly and accurately by hand alone. To address this, the system decouples the human's hand position from the precise movement of the tool head. While the user provides approximate positioning, the tool makes fine corrections internally, allowing it to stay aligned with the target path. This approach improves both the speed and accuracy of the operation, while keeping the user in control of the general movement.

To achieve this, the system uses a downward-facing camera rigidly mounted to the tool's frame, which observes a planar workspace populated with fiducial markers (specifically, ArUcoE tags). The methodology is divided into two main phases. The first is the mapping phase (Section 3.4), during which the system constructs a global map of all visible marker positions using geometric pose estimation and bundle adjustment. The second is the tracking phase (Section 3.5), where the system localizes the camera in real time by matching observed markers to their known positions in the precomputed map. This localization allows to calculate the distance the user deviated from the desired path, then a small corrective movement can compensate for this deviation. Core components of this methodology include intrinsic and extrinsic camera calibration (Section 3.2), subpixel-refined marker detection, homography-based perspective correction to obtain a top-down view, and geometric validation of detected markers.

3.1 System Overview and Assumptions

The proposed system utilizes a downward-facing camera mounted on top of a handheld tool, the tool also contains a movable tip, a tool-specific insert (e.g., a cutter, pen, or engraving bit) mounted inside the device. This tip can be actively adjusted within the housing of the tool to compensate for unintended user motion. The camera is mounted at a fixed angle above a work surface. While the camera is elevated and angled to maximize visibility of the area in front of it, the motion of the handheld tool is constrained to planar translation and rotation, specifically, movement in the X and Y directions and rotation about the vertical axis (θ). To put it simply, the camera is mounted on the tool while that tool can not be lifted off of the ground and thus stays perfectly flat on the ground. The system operates in two distinct phases: (1) a mapping phase, in which the tool moves across the workspace to build a globally consistent map of fiducial markers using pose estimation and bundle adjustment; and (2) a tracking phase, during which the tool localizes itself and adjusts its movable tip in real time relative to the previously constructed marker map for precise cutting operations. The system compensates for perspective distortion using a pre-calibrated homography matrix **H**. This rectifies the camera view into a top-down perspective, enabling accurate 2D pose estimation even with the angled camera setup.

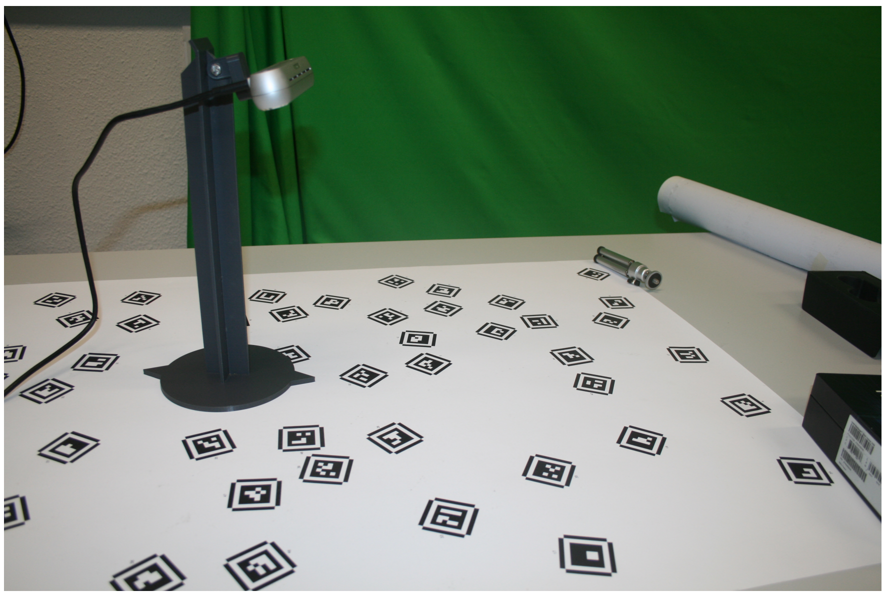


Figure 3.1: An example setup with the realsense camera and some markers on the workfield.

Key Assumptions

The methodology relies on the following assumptions:

1. **Planar Motion**: The tool moves only in 2D space (X, Y, θ)
2. **Planar Markers**: all markers lie on a static, flat plane parallel to the motion of the tool.
3. **Stable Homography**: The homography **H** remains valid throughout operation (i.e., no lens distortion changes, tilt of the tool or camera refocusing).
4. **Marker Rigidity**: Detected ArucoE tags are affixed to the environment without relative motion (although markers can disappear).

3.2 Setup and Calibration

The system's accuracy relies on careful workspace configuration and precise calibration procedures. The workspace consists of a flat planar surface populated with an unknown distribution of fiducial markers. These markers are arranged to ensure continuous visibility during operation: at least one marker must be detectable for tracking, while two or more are required during the initial map creation phase to resolve positional ambiguity. Although the more markers in the overlapping view between the map and the current frame, the more accurate the result will be. Thus the strategic placement and density of the markers is crucial for robust pose estimation.

The camera is rigidly mounted on top of the handheld tool, maintaining a fixed height and inclination angle relative to the work surface. The tool features an physical indicator that marks its nominal position during system initialization, this clearly visible marker denotes the tool's reference point within the workspace. These constraints align with the assumptions defined in Section 3.1 and ensure consistent relative positioning between the camera and the movable tip for reliable 2D pose estimation.

Intrinsic Calibration

Calibration occurs in two stages. The first stage involves intrinsic camera calibration, which is done using Zhang's method [Zha00]. This is performed using a printed checkerboard pattern placed on a flat surface. This step determines the camera's internal parameters, including focal length, optical center, and lens distortion coefficients. To ensure high accuracy, the checkerboard was printed at high resolution and carefully glued onto a wooden board. The board itself was sanded to be as flat as possible, minimizing warping or unevenness that could introduce errors during calibration. Multiple images of the checkerboard were captured from various angles and distances, and then Zhang's method can be applied to compute the intrinsic parameters. This calibration procedure is conducted only once for the specific camera and corrects for manufacturing tolerances and lens imperfections, ensuring accurate undistortion and consistent geometric interpretation in all subsequent image processing steps.

To evaluate the quality of the calibration, reprojection error was calculated, which measures the difference between the projected 2D locations of known 3D points (based on the estimated camera parameters) and their actual observed positions in the image. A low reprojection error indicates high calibration accuracy. In this setup, the average reprojection error (0.756 pixels) was found to be within acceptable limits, confirming the reliability of the intrinsic parameters for precise localization tasks.

Extrinsic Calibration

The second stage is extrinsic calibration, which determines the angular offset between the camera's optical axis and the plane of the workfield. This step accounts for the fixed, angled mounting of the camera and must be repeated if the camera is repositioned or its alignment is disturbed. Unlike intrinsic calibration, which characterizes the internal properties of the camera, extrinsic calibration focuses on the camera's orientation relative to the workspace.

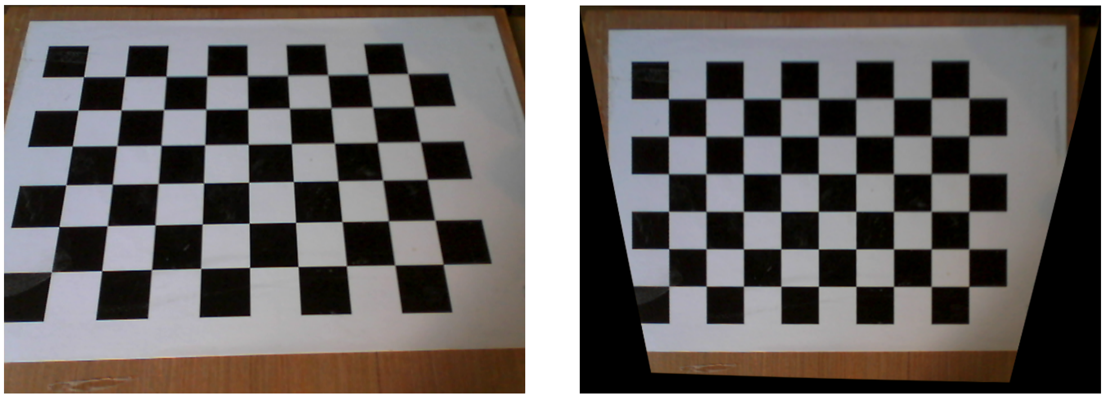


Figure 3.2: the frame before perspective transform (left) the frame after perspective transform (right)

To perform this calibration, the same high-quality checkerboard pattern used during the intrinsic calibration is placed flat on the work surface. Since the intrinsic parameters are already known, they are preserved during this step. A perspective transformation is computed to map the distorted, angled view of the checkerboard to a top-down, rectified image where the lines of the checkerboard are parallel and evenly spaced. This transformation effectively "undoes" the perspective distortion introduced by the oblique mounting angle of the camera, allowing the system to interpret the workspace as if it were being viewed from directly above. The resulting homography matrix is then used throughout the system to project the current frame to its top down view.

Marker Placement

Proper marker placement is essential for both accurate mapping and reliable localization. Unlike previous setups where the checkerboard was predefined, this system assumes that marker placement will be done by the user. As such, understanding the spatial requirements and constraints is crucial.

During the mapping phase, at least two visible markers are required to be able to add new markers to the map, assuming one of them is already part of the existing map. This means that markers must be placed with sufficient overlap in the camera's view to ensure that multiple markers can be seen simultaneously. The exact spacing depends on the camera's field of view and mounting angle, but in general, markers should be placed densely enough that the system can consistently detect two or more within the same frame. Notably, continuous localization is not required during mapping; temporary loss of pose estimation is acceptable as long as mapping conditions are met.

In contrast, the tracking phase demands continuous localization. For this, the system requires that at least one marker be visible at all times. Losing all visible markers results in the loss of positional tracking, making real-time corrections impossible. Therefore, marker placement must ensure uninterrupted visibility throughout tool movement.

The optimal density and placement of markers depend heavily on the camera's viewing characteristics. A wider FOV and shallower viewing angle provide a larger visible area, allowing for lower marker density. However, this comes at the cost of decreased detection accuracy, as markers appear smaller and more distorted at greater distances. Conversely, a narrower FOV and steeper angle result in a already more top-down view before extrinsic calibration, thus improving marker detection accuracy but reducing the visible area and thus requiring higher marker density.

The trade-offs between marker density, and detection reliability will be explored in more detail in Sections 4.4 and 4.5.

3.3 Marker Detection

Reliable and accurate marker detection is a fundamental component of the system, as it directly determines the quality of pose estimation and, by extension, the overall performance of the tool. This section outlines the detection pipeline and supporting techniques used to extract geometric and positional information from fiducial markers placed in the workspace. It begins by describing the marker specifications and detection enhancements, followed by geometric validation procedures, pose extraction, and scale factor estimation. These steps together ensure robust marker tracking under varying conditions and enable precise localization throughout the system.

3.3.1 Marker Specifications

The system utilizes standard ArUco markers, chosen for their optimal balance between detection reliability and information density. These fiducial markers consist of a binary pattern enclosed within a black border, providing robust detection even under variable lighting conditions. Marker size is maintained constant across the workspace to ensure uniform detection characteristics. This consistency enables automatic scale calculation during pose estimation, as the physical dimensions serve as a known reference.

In theory, the workspace covered by the system could be arbitrarily large, as there are no inherent limitations on the area over which the tool can move. However, in practice, the maximum usable workspace is constrained by both the number of unique fiducial markers available and the physical size of the board, paper, material, or floor that the system is placed upon. Since each marker must be uniquely identifiable—repeated markers would lead to ambiguity in localization—the choice of ArUco dictionary becomes a critical factor. For example, the commonly used ArUco dictionaries support a maximum of 1000 unique markers, imposing a hard upper bound on the number of distinguishable landmarks that can be placed in the environment.

To mitigate this limitation, it is possible to generate custom marker dictionaries with a greater number of unique IDs, or reuse identical markers at known, fixed locations within a well-structured map. However, these approaches increase system complexity and reduce flexibility. As such, they fall outside the scope of this thesis, where the focus is more on solutions that rely solely on standard ArUco dictionaries and uniquely identifiable markers.

To ensure the system remains scalable, the required marker density should be low enough so that the physical dimensions of the workspace become the primary limitation, rather than the finite number of unique marker IDs. In other words, the number of markers required per square meter must be limited to enable sufficiently large workspaces within the constraints of the chosen ArUco dictionary. However, this requirement is in direct conflict with the accuracy-driven requirements just described in Subsection 3.2, which suggest that higher marker density improves localization accuracy. Therefore, a balance must be found, dense enough to maintain reliable tracking performance, yet sparse enough to keep the system scalable. Identifying this trade-off is a key focus of the evaluation in later sections.

3.3.2 Enhanced Detection Using ArUcoE

The system incorporates the ArUcoE marker enhancement as proposed by Kedilioglu et al. [Ked+21] to improve pose estimation accuracy beyond conventional ArUco markers. This modified fiducial design surrounds the standard ArUco pattern with a 2×2 checkerboard border (this counts the internal corners so its 3x3 squares with the ArUco code in the central white square). This creates distinct rectangular regions that facilitate subpixel corner localization.

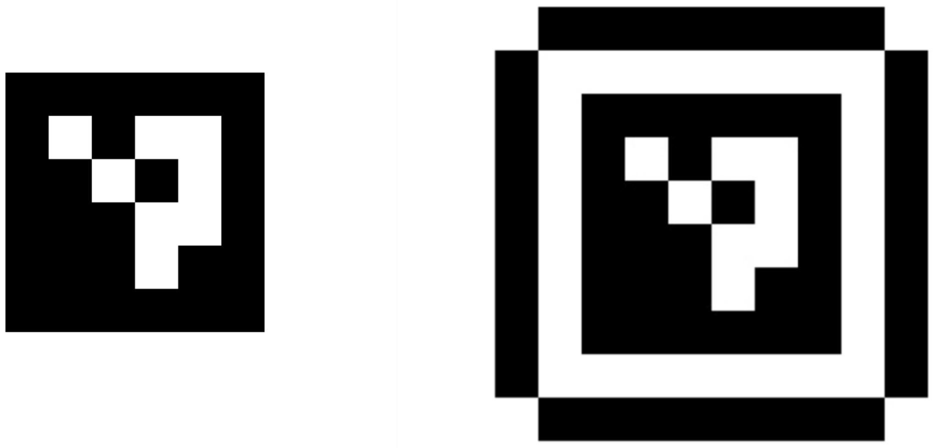


Figure 3.3: ArUco (left) marker with id 0 and the same marker of ArUcoE (right)

During detection, the inner ArUco marker is first identified using standard marker detection techniques. Once detected, the system uses the known geometry of the surrounding checkerboard to define expected corner locations in the image. To refine these corners, a region of interest (ROI) is defined around each expected checkerboard corner. The initial locations of the corners within these ROIs are estimated using OpenCV's implementation of the *goodFeaturesToTrack* algorithm. This method detects strong corners by analyzing the eigenvalues of the autocorrelation matrix of image gradients. In essence, it identifies points in the image where intensity changes significantly in multiple directions—i.e., where a corner is likely to be.

These initial corner positions are then refined to subpixel accuracy using OpenCV's *cornerSubPix* function [FG87], which is based on an iterative refinement method. The algorithm works by fitting a local intensity model to a small window around the initial corner estimate. It computes the gradient of pixel intensities within this window and determines how the estimated position of the corner should be shifted to better align with the underlying intensity distribution. This process is repeated iteratively: on each iteration, the estimated position is updated based on the minimization of a cost function that represents the difference between the modeled and observed gradients.

Convergence is reached when either the corner position changes by less than a specified epsilon threshold or a maximum number of iterations is completed. The output is a set of 2D corner coordinates with subpixel precision—typically accurate to within a fraction of a pixel.

This level of precision is crucial in the context of this system. Since the pose of the tool is inferred from the spatial configuration of known points (the marker and checkerboard corners), even small errors in corner localization can lead to significant inaccuracies in pose estimation. By utilizing this enhancement to the ArUco marker, the system maximizes the reliability of the extracted geometry and ensures consistent, accurate pose tracking across frames.

3.3.3 Geometric Validation

As previously discussed the system transforms the frame into a fixed top-down perspective, which ensures key geometric properties in the image. In this view, lines that are parallel in the real world will be parallel in the image, and relative distances between points are maintained. As a result, markers—which are designed as perfect squares—should also appear as perfect squares in the frame. This geometric consistency allows the system to evaluate the quality of marker detections on the fly. Specifically, it can enforce strict shape constraints by validating how closely each detected marker resembles an ideal square. Two key metrics are used for this purpose: diagonal consistency and side length consistency.

Both of these validations ensure that the detected ArUco markers closely approximate a perfect square, allowing the effectiveness of the detection and corner refinement algorithms to be quantitatively assessed. This validation step is performed after subpixel corner refinement but before pose estimation, serving as a geometric consistency check. Markers that significantly deviate from the expected square shape—based on side lengths and corner angles—are excluded from further processing. In such cases, a warning can be issued, indicating potential issues with calibration or marker quality.

Diagonal consistency. The system validates marker integrity by checking the difference in length between the two diagonals of the detected quadrilateral. This diagonal-based squareness check is used to assess whether the marker's corners form approximate right angles. If the diagonals are equal in length, the shape has all 90° internal angles

$$\delta = \frac{\left| \|\mathbf{p}_3 - \mathbf{p}_1\| - \|\mathbf{p}_4 - \mathbf{p}_2\| \right|}{0.5(\|\mathbf{p}_3 - \mathbf{p}_1\| + \|\mathbf{p}_4 - \mathbf{p}_2\|)} \tag{3.1}$$

where

- $\mathbf{p}_i$ is the position of the i^{th} corner
- $\| \cdot \|$ denotes Euclidean distance

Side length consistency. While diagonal consistency verifies the presence of right angles, it does not guarantee that all four sides are equal, i.e. the shape could still be a rectangle. To address this, the system performs a secondary check by measuring the variation between the side lengths. A low variation confirms that the detected marker is not only square in angle but also uniform in size,

$$\delta = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(d_i - \bar{d})^2} \tag{3.2}$$

where

- d_i is the length of the i^{th} side of the square
- $\bar{d}$ is the average side length

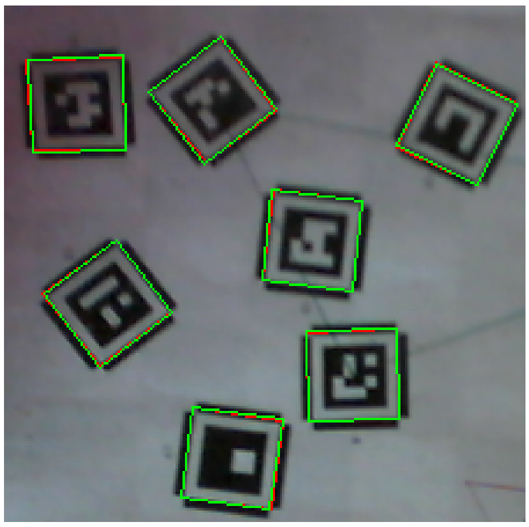


Figure 3.4: Detected markers are shown in red, with an ideal square projection overlaid in green. The accuracy of marker detection is visually indicated by the amount of visible red—less visible red implies a better alignment between the detected and ideal marker shapes.

However, this approach has certain limitations. It implicitly assumes that the physical markers are perfectly square and lie flat on the surface, which is not always the case in practice. Minor warping of the paper or imperfections in the printing process can cause subtle deviations from the ideal marker geometry, potentially affecting the validation outcome even if the detection algorithm is functioning correctly. Furthermore, since ArUco marker detection is pixel-aligned, the marker's alignment relative to the image sensor grid does influence the perceived distortion. In particular, when a marker happens to align well with the pixel grid, geometric inaccuracies may be artificially minimized. This effect can mask small physical imperfections and may introduce a bias in favor of standard ArUco markers during validation, especially when comparing against more complex alternatives.

Despite these limitations, the square validation step remains a valuable filter for maintaining the accuracy and reliability of pose estimation, especially when used in controlled environments with well-prepared markers.

3.3.4 Pose Calculation

To simplify further calculations, each detected marker is converted into a compact pose representation. Since all markers lie on a flat plane, their full geometric description can be captured using just three parameters: x, y, and θ, representing the marker's position and orientation in the workspace. This conversion preserves all essential information about the marker's placement while also providing a more efficient format for downstream processing. Additionally, deriving the pose from all four refined corner points improves robustness and accuracy. The position is calculated as the average of the corner coordinates, while the orientation is estimated by averaging the angles of the marker's sides, accounting for the correct rotational offset. This averaging process reduces the influence of local noise or distortion in individual corner detections, leading to more stable and precise pose estimates.

This approach proves particularly efficient for real-time processing, requiring only basic vector arithmetic to maintain pose updates at the camera's full frame rate. It is crucial that the markers are perfectly square, as discussed in the previous subsection 3.3.3, since the geometric assumptions underlying this method rely on ideal right angles and uniform side lengths.

The marker's **position** (x, y) is computed as the average of the four corner coordinates:

$$x = \frac{1}{4}\sum_{i=1}^{4} x_i, \quad y = \frac{1}{4}\sum_{i=1}^{4} y_i \tag{3.3}$$

The **orientation** θ is computed from the average direction of all four marker edges. First, directional vectors for each edge are calculated by subtracting consecutive corner coordinates. Then, the angle of each vector with respect to the x-axis is computed using the `atan2` function. To ensure consistency and avoid discontinuities due to wrapping around $\pm\pi$, the angles are unwrapped before averaging.

Finally, since a square marker's edges span 360°, the average orientation of all four edges naturally aligns with the diagonal bisector of the marker. To recover the marker's actual heading (e.g., aligned with its top edge), a fixed angular offset of $\frac{3\pi}{4}$ is subtracted from the average:

$$\theta = \text{mean}(\text{unwrap}(\text{atan2}(dy, dx))) - \frac{3\pi}{4} \tag{3.4}$$

This method preserves all spatial information while potentially improving accuracy, as it integrates information from all corners and sides. The resulting pose is compact, more robust against local distortions, and well-suited for use in geometric transformations and pose estimation pipelines.

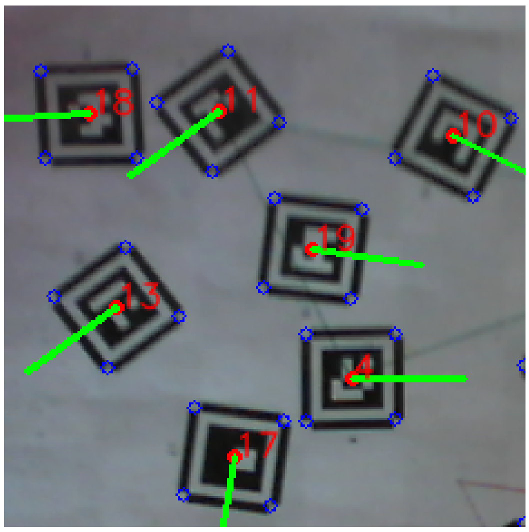


Figure 3.5: Detected corners of the ArUcoE markers in blue, with each marker's calculated position shown as a red dot and its orientation indicated by a green line.

Since all markers are of known and consistent physical dimensions, the system can compute a global scale factor that converts pixel measurements into real-world units (e.g., millimeters). For each detected marker, the average distance between adjacent corners in pixel space is calculated and compared to the predefined physical side length of the marker. This yields a local pixel-to-millimeter scale, and by aggregating such measurements across the workspace, the system establishes a globally consistent scale factor.

This scale allows the entire map to be interpreted in real-world units, enabling accurate measurements and alignment with physical components. Additionally, because this scaling is derived dynamically from marker detections, it can adapt to slight variations in lens distortion across the work area. Continuous validation of marker size against expected dimensions also provides a form of system health monitoring: significant deviations may indicate camera calibration drift, marker deformation, or print quality issues.

3.4 Workspace Mapping

The system will later use the ArUcoE markers to estimate its pose relative to the workspace. However, since the poses of these markers are initially unknown, localization cannot yet be performed. As a result, the system must first construct a global map of all visible markers. This mapping process begins without any prior knowledge of marker positions and relies on observing them from different viewpoints as the tool moves. Each observation provides spatial constraints that allow the system to infer the relative position and orientation of the markers. By combining multiple overlapping observations, a consistent global map is incrementally built, which then serves as the reference frame for all subsequent pose estimation.

In the intended use case, the user typically wishes to perform a cutting or drawing operation at a specific location on the physical workspace—for example, aligning a design with the corner of a material or centering it on a pre-defined area. To enable this, a global map must be created so that the digital design can be positioned accurately relative to real-world coordinates. As a result, the mapping phase is not only a technical requirement, but also a practical benefit. Since the user must already interact with the system to define where the operation should occur, the additional setup step of generating the marker map does not present a significant overhead.

This mapping procedure assumes that the camera maintains a strictly top-down view of the workfield and that its motion is constrained to a plane parallel to the surface. As a result, each frame-to-map transformation consists of only three degrees of freedom (DOF): two translational components in the x and y directions, and a rotation about the z-axis. These simplified constraints allow for efficient 2D pose estimation and reduce computational complexity.

As the tool traverses the workfield, it incrementally aligns its current observations with the existing map, adding new markers as they are detected. To counteract cumulative errors from individual frame alignments, the system optionally applies bundle adjustment to refine all marker poses jointly. This optimization minimizes the reprojection error between observed and predicted marker positions across all frames using a least-squares approach.

3.4.1 Map Matching

Before a global map can be constructed from individual local observations, it is important to first consider the basic operations involved in combining multiple maps. To merge two maps, they must first be brought into a common coordinate system. The initial requirement is that both maps share the same scale, which can be achieved by applying the global pixel-to-millimeter scale factor as discussed in 3.3.4. Once the scale is consistent, the maps must also be aligned in position and orientation. This involves estimating a rigid transformation that best aligns the marker positions in one map with those in the other. Only after this alignment can the individual maps be meaningfully combined or compared.

As discussed earlier, only a single marker is required to determine the tool's pose. This is due to the fact that the system operates in a planar environment with only three DOF. Since each marker also encodes three variables—position and orientation on the plane—a single known marker provides sufficient information to fully localize the tool. However, if the goal is to expand the map by adding new markers, at least two known markers must be visible in the current frame so one can be used to compute the transformation and the other can be added.

To compute this alignment, a custom optimization-based approach is employed. Given a set of marker poses observed in the current frame and their corresponding known poses in the map, a transformation is estimated that minimizes both the **positional** and **angular** errors between matched markers. This is implemented as a cost function that takes a rotation angle ϕ and translation parameters t_x, t_y as input. The source marker positions are rotated and translated, and the resulting positions are compared to the known target positions in the global map. The **positional error** is computed as the sum of squared differences in x, y coordinates, while the **angular error** is computed as the sum of squared differences in orientation angles, taking into account angle wrapping.

The total cost is a weighted combination of these two errors:

$$\text{cost} = \sum \left\| R(\phi) \cdot P_{\text{source}} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} - P_{\text{target}} \right\|^2 + \text{angle\_weight} \cdot \sum \left(\theta_{\text{target}} - (\theta_{\text{source}} + \phi)\right)^2 \quad (3.5)$$

where

- $[t_x, t_y, \phi]$ is the current estimated transformation to be scored
- $P, \theta_{\text{source}}$ is the pose of an observed marker and $P, \theta_{\text{target}}$ is the pose of a mapped marker
- $R(\phi)$ is the rotation matrix for the given ϕ
- angle_weight is some high value so the angular error is taken into account first by the minimize method

This minimization is performed using the Powell optimization method, as implemented in the `scipy.optimize.minimize` function in Python. Powell's method is a derivative-free algorithm that does not require gradient information, making it particularly well-suited for this application, where the cost function is geometric and non-differentiable. Moreover, the method is efficient for low-dimensional problems such as the present case, which involves only three parameters (translation in x and y, and rotation about the z-axis). Its robustness and stability in smooth, unconstrained optimization scenarios make it a practical and effective choice.

3.4.2 Map Construction

The initial step in the mapping process involves constructing a global map of the ArUco markers distributed across the workfield. Since no prior information is available regarding the positions or orientations of the markers, the system incrementally builds the map by observing markers from multiple viewpoints as the tool is moved.

To ensure full coverage of the workspace, the tool is manually moved in such a way that all fiducial markers are observed at least once. For each captured frame, the system detects the visible markers and estimates their poses relative to the camera. This process results in a locally constructed map of marker positions, which may differ in origin and orientation from the predefined global map. To enable meaningful integration, the observed map must be aligned with the global map using the method described in Subsection 3.4.1. This alignment procedure computes a transformation that best matches the observed markers to their global counterparts, minimizing positional and rotational discrepancies.

Accurate transformation estimation is critical, as it directly affects the placement of new markers and the consistency of the overall map. Errors introduced at this stage may propagate through subsequent updates and must be corrected later through global optimization procedures.

Since all pose estimations and marker detections are initially expressed in pixels, the global scale factor that was previously discussed in Subsection 3.3.4 is applied to convert these into real-world units (millimeters). This scale is derived from the known physical size of the ArUco markers, which allows consistent translation between image-space and real-space. Applying this scaling ensures that all integrated marker positions are expressed in a unified, metric coordinate system that directly corresponds to the physical workspace.

Once the alignment between the current frame and the existing map has been established, any newly observed markers are transformed into the global coordinate frame, scaled appropriately, and appended to the map along with their estimated poses. This iterative process continues until all visible markers in the workspace have been incorporated. Overlapping observations across different frames ensure redundancy in the dataset, which is later leveraged for global optimization. The result of this step is a preliminary map that serves as the foundation for further refinement and accuracy improvement.

A key limitation of this incremental mapping approach is the potential for error accumulation. When a marker is integrated into the map with a slight pose estimation error this inaccuracy can propagate especially if that marker is subsequently used as the sole reference to localize new frames. In such cases, all newly added markers inherit the original marker's positional and angular errors, potentially distorting the global map over time. Angular errors are particularly impactful, as even a small rotation discrepancy can lead to significant spatial deviation over longer distances. This compounding effect underscores the importance of having multiple overlapping markers in view during mapping. By relying on redundant observations and averaging across several reference markers, the system can suppress the influence of individual pose errors and maintain geometric consistency throughout the map. These overlapping constraints are especially beneficial for downstream global optimization steps, where accumulated drift can be corrected more effectively.

3.4.3 Global Optimization via Bundle Adjustment

To address that key limitation of accumulation of local pose errors during incremental mapping 3.4.2, the system performs a global optimization step using bundle adjustment. This technique jointly refines all marker poses and frame transformations by minimizing the overall reprojection error across the entire dataset. Rather than treating each frame-to-frame transformation independently, bundle adjustment leverages all available marker-frame observations simultaneously, enabling it to reconcile conflicting measurements and reduce drift introduced during earlier stages of mapping.

Each observation contributes a constraint that relates the estimated pose of a frame to the known pose of a marker. These constraints are formulated as residuals, typically in terms of positional and angular discrepancies between expected and observed relative poses. The optimization problem is then defined as a nonlinear least-squares minimization over all frame and marker poses, subject to these constraints. This process aligns all observations globally, ensuring that the resulting map is both internally consistent and as geometrically accurate as possible.

The presence of overlapping markers across frames is particularly beneficial in this context, as it introduces redundant observations that anchor the optimization and help resolve local ambiguities. Frames that observe multiple markers simultaneously create cross-links within the map structure, which act as soft constraints to counteract drift and outlier influence. In sparse regions, where only one or two markers are visible, the optimization is less effective—highlighting the importance of good marker layout during the mapping phase.

Bundle Adjustment Implementation

To perform bundle adjustment, the system first collects all available observations collected during the mapping phase, including the estimated camera poses, the marker poses detected in each corresponding frame and the estimated map created as described above. These form the basis of the optimization process. Each observation encodes a relative pose between a camera pose and one or more observed markers, while the initial map and camera trajectory provide a starting point for refinement.

The core idea of the optimization is to project the current estimate of the global marker map into each camera frame, simulating what the camera should have observed given its estimated pose and the current map geometry. This simulated observation is then compared to the actual observed marker pose in that frame, and the difference forms the residual. The specific residual metrics—based on positional and angular discrepancies—are described in detail in Section 4.1.

Formally, for each observation, the system backprojects the corresponding global marker pose into the local coordinate frame of the observing camera. The projected pose is then compared to the observed marker pose in that frame, and the residuals are computed as a combination of Euclidean distance and wrapped angular difference. This process is repeated for all markers in all frames, producing a full residual vector for the least-squares solver.

The optimization is implemented using the `scipy.optimize.least_squares` function, which applies the Levenberg-Marquardt algorithm to iteratively refine the parameters. The parameter vector includes both the global marker poses and the estimated camera poses for each frame, with each represented as a 3-element vector (x, y, θ). In each iteration, the solver slightly adjusts the marker and camera pose estimates and re-evaluates the residuals. This process continues until convergence is reached—i.e., when further adjustments do not significantly reduce the total error.

By jointly optimizing all poses with respect to all observations, this global adjustment step significantly reduces accumulated drift and corrects local inconsistencies introduced during incremental mapping. It ensures that the final map is geometrically consistent and that camera pose estimates are better aligned with the true structure of the workspace.

3.5 Real-time Tracking

With the marker map established as described in Section 3.4, the system enters its operational phase. The tool continues to operate under the same constraints as during mapping: fixed in orientation and height, observing the workfield from a top-down perspective. The geometric principles originally used to construct the marker map are now applied in a live setting to estimate the tool's pose, with an emphasis on speed and consistency.

Using the known marker map, the system localizes the tool in real time by matching detected markers in each frame to their corresponding poses in the global map. This provides a continuous global pose estimate, which serves as the basis for controlling the motion of the attached tool.

The primary goal in this phase is to trace out a predefined closed path with the cutting tool. This path must be followed in a specific direction, depending on the application requirements. Accurate tracking of the tool's position along the path is essential for ensuring the quality of the final result.

3.5.1 Tool Localization

Real-time tool localization relies on the same geometric principles established during map construction (Section 3.4.2). At each frame, the system detects all visible markers and computes their poses relative to the camera. These detected marker poses are then matched to their counterparts in the global map. Provided that at least one known marker is visible, a transformation can be calculated that determines the camera's pose in the global coordinate frame. Since the camera is rigidly mounted on the handheld tool, the tool's pose can be directly inferred from the camera's pose through a fixed transformation. This allows the system to continuously track the position and orientation of the tool relative to the workspace in real time.

This transformation is again constrained to three degrees of freedom: two for planar translation and one for in-plane rotation. These constraints are valid under the assumption that the camera maintains a fixed height and orientation perpendicular to the workfield, as introduced in Section 3.4. An important consequence of this restriction is that a single marker which is also defined by three parameters (position and orientation in the plane) is sufficient to compute a unique camera pose. By solving a pose alignment problem using the known marker positions, the system obtains a live estimate of the camera's position and orientation in global coordinates.

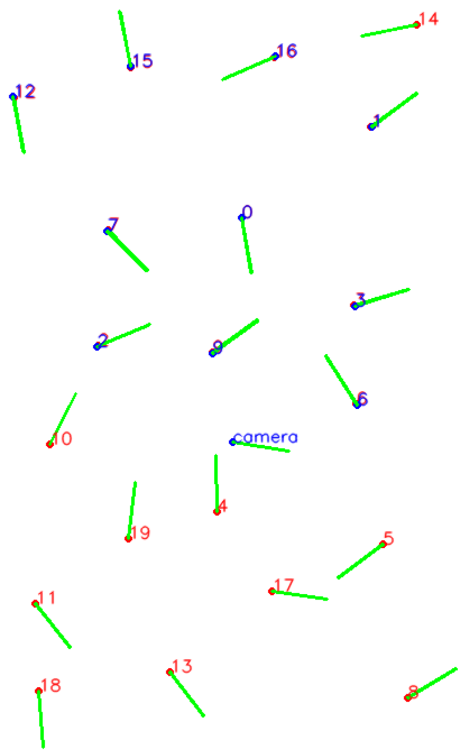


Figure 3.6: Visualization of the mapping and localization process. Red markers represent the stored poses from the global map, while blue markers show the corresponding poses currently observed by the camera. The camera's estimated pose is also shown in blue.

An important consideration during localization is the handling of scale. Marker detections are initially expressed in pixel-based coordinates, meaning that their positions are relative to the image resolution rather than the physical workspace. In contrast, the global map is defined in real-world units, based on the known physical size of the markers. To enable meaningful comparison and alignment, the observed marker poses are first scaled to match the map's coordinate system. This can be done using the global scale factor calculated in the previous Section 3.4

To accurately infer the tool's pose from the camera's pose, an additional calibration step is required to determine the fixed transformation between the two. This transformation accounts for the spatial offset between the camera and the tool's physical indicator—the point used to define the tool's effective location. The calibration is performed by placing the tool such that its indicator is aligned with a predefined reference marker whose pose is already known in the global map. The system then records the camera's pose at this configuration, allowing it to compute the relative transformation from the camera to the tool. This transformation is subsequently applied during operation to convert the estimated camera pose into the corresponding tool pose.

To perform this calibration step, the tool is first manually positioned such that its physical indicator aligns precisely with a known reference marker in the global map. The system then observes the visible markers within the camera frame and computes the transformation from global coordinates to the current camera frame. This transformation describes the camera's pose relative to the map.

Using the known global pose of the reference marker, the system applies the inverse of the computed transformation to project the reference marker into the camera's image plane. Although the indicator itself typically lies outside the camera's field of view, this projection allows the system to determine where the indicator *would* appear within the image if it were visible. Since the camera is rigidly mounted to the tool, the spatial relationship between the camera and the indicator remains fixed, making this one-time projection a reliable proxy for the tool's position.

In subsequent frames, the system continues to compute the camera's pose based on visible markers. It then applies the current transformation to the previously computed image-space indicator coordinates to obtain the corresponding world-space position of the indicator. This enables accurate localization of the tool's reference point at every frame, even when the indicator itself remains outside the camera's field of view.

3.5.2 Path Representation and Tracking

The tip used for cutting or tracing is not rigidly fixed to the tool but instead has a limited range of motion within the tool. This relative movement allows the system to perform small corrective adjustments to correct when the user does not follow the path perfectly. As a result, the tip's position in world coordinates is computed as the sum of the globally localized tool position as described in the previous subsection 3.5.1, and the tip's position relative to the tool's indicator, which is estimated in real-time based on internal measurements and calibration of the stepper motors used to move the tip. This decoupling between tool and tip is essential for maintaining accurate path-following under manual guidance.

The trajectory to be followed by the tip is typically the result of a user-defined design process in a CAD program or vector graphics environment. These designs may be exported in various file formats, depending on the application domain such as G-code for CNC machines or SVG for laser cutters. While the internal representation of these formats can differ significantly, they often reduce to one or more closed polygons when viewed in two-dimensional space. For the purposes of this thesis, it is assumed that the input path is simplified and expressed as only one such polygon, without further exploring the parsing or interpretation of complex CAD formats.

With the global map established and the tool in place, the system proceeds to guide the tip along a predefined path. It is first assumed that the user has positioned the tool outside the target polygon. The path-tracking procedure then begins by identifying the closest point on the polygon, guiding the tip toward it. This approach ensures that motion starts from a natural entry point, avoiding abrupt direction changes or unintentional traversal through the interior of the shape.

Once this initial point is reached, the system continues to follow the polygon by sequentially advancing through its corners. These waypoints are traversed in a consistent direction —clockwise or counter-clockwise— based on application-specific requirements. Directionality can be critical in certain cutting tasks, where material properties or tool rotation may dictate a preferred orientation.

Throughout this process, the tip is moved at a constant linear speed, carefully chosen by the

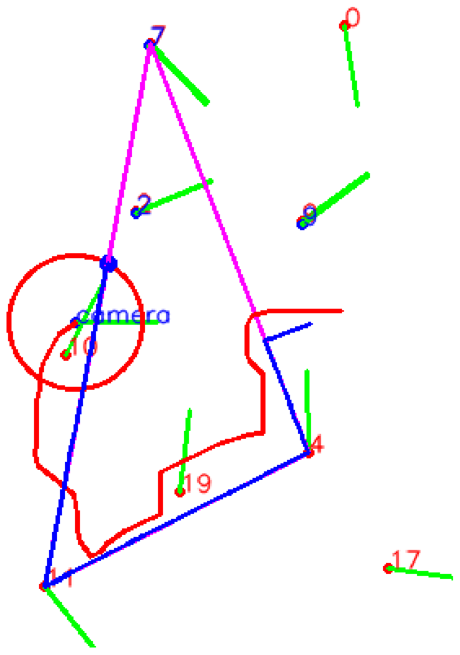


Figure 3.7: Visualization of a toolpath-following operation. The red line indicates the estimated tool's movement trajectory, while the blue line shows the actual path followed by the tip. The intended shape to be cut is marked in magenta. The current positions of both the tool and the tip (blue dot) are displayed at the time of capture. The red circle represents the tip's limited range of motion relative to the tool.

user based on the material being processed, depth of cut, ... Maintaining a consistent feed rate is essential: too fast and it can damage the tip or cause poor-quality edges; too slow and the material might overheat or burn, and the cutting operation will take longer than necessary. The selected speed thus balances precision, safety, and finish quality, adapting to the demands of different materials and operating conditions.

Throughout execution, the system continuously updates the position of the tip based upon the estimated tool's pose and tracks its progress along the polygon. The result is a responsive and precise control loop that adapts in real-time to user movements, compensates for small deviations, and ensures that the tip closely adheres to the intended path. Upon completion, the system returns to the initial entry point, completing the closed-loop trajectory as defined by the input path.

To ensure both the integrity of the resulting shape and the safety of the tool, the system incorporates simple yet effective constraints on path execution. At all times, the tip is prevented from crossing into the interior of the polygonal path, thus avoiding unintended cuts that would compromise the final output. This is particularly important in finishing passes where high accuracy is required. Furthermore, the system monitors the relative velocity between the user-driven tool and the tip's internal position. If the user moves too quickly and the tool lags behind, the system automatically retracts the tool to prevent excessive lateral force that could damage the cutting bit. Conversely, if the tool attempts to advance beyond what the user is guiding, its movement is limited to remain close to the path boundary, preventing tool strain and maintaining consistent material interaction.

Chapter 4

Experiments & Results

To quantify the performance and accuracy of the proposed system, both simulated and real-world validation experiments were conducted. These experiments aim to evaluate the marker detection pipeline, the reliability of map construction, and the overall system's ability to track and follow a given path with minimal deviation.

To enable the experiments, two additional validation systems were developed. First, a simulation environment was created using Unity, allowing full control over marker placement, camera motion, and ground-truth pose information. This simulation assumes an ideal pinhole camera model, free from lens distortion, sensor noise or lighting issues, thus serving as a baseline to assess the geometric correctness of the localization and tracking algorithms. Second, a physical prototype was built using a pen-based drawing system with retractable mechanics, driven by stepper motors and controlled via the same localization infrastructure. This setup enables visual inspection and measurement of real-world drawing accuracy on printed marker fields.

Together, these experiments aim to demonstrate the system's practical viability and help characterize both the achievable accuracy and the limitations under ideal and non-ideal conditions.

4.1 Marker Map Generation

To support flexible and consistent experimentation across both simulated and real-world setups, a map generation system was developed that produces 2D marker layouts with configurable parameters. These include the overall dimensions of the map, the number of ArUco markers used in that map and thus the density of markers, the size of each marker, and whether the markers are placed in a grid or distributed randomly. The output of this process is a JSON file containing a full specification of the marker map, including the map, and marker size, and each marker's ID, position in millimeters, and rotation.

These generated maps serve as a source of ground-truth for the experiments. In the simulation environment, it is used to instantiate the markers in Unity, placing each one precisely at its designated coordinates within the virtual scene. This ensures the simulation reflects the intended spatial arrangement exactly, without manual intervention or potential placement errors. For physical experiments, the same map is used to generate an render of said marker map. All markers are drawn onto a white background and placed according to their specified dimensions, after which the image is printed as a single sheet. This guarantees that the relative positions between markers match the digital layout exactly, allowing the printed map to act as a physical counterpart to the virtual scene.

Map Alignment

Both the simulation and printed maps use a shared coordinate system in which the origin is located at the top-left corner of the layout. However, the system's reconstructed map does not necessarily share this same origin or orientation. The estimated map may be translated, or rotated depending on the camera's starting pose and the initial markers detected. To meaningfully compare estimated poses to ground-truth data, an alignment step has to be performed. This is done using the same method as described in 3.4.1. This transformation is then applied to all estimated camera poses before computing error metrics.

Map Error

Once the observed marker map has been aligned with the ground-truth map, the accuracy of the mapping process can be quantitatively assessed by comparing the position and orientation of each corresponding marker. The translational error for a single marker is defined as the Euclidean distance between its estimated position and the ground-truth position:

$$e_t = \sqrt{(x_{est} - x_{gt})^2 + (y_{est} - y_{gt})^2} \tag{4.1}$$

Here, x_{est}, y_{est} represent the estimated position of the marker as computed by the system, while x_{gt}, y_{gt} denote the known ground-truth position. This measure captures how far the estimated marker lies from its true location in the workspace.

For the rotational error, care is taken to handle angular wrapping correctly, such that differences like $1°$ and $359°$ are interpreted as a minimal discrepancy. The angular error is calculated as:

$$e_r = |((\theta_{est} - \theta_{gt} + \pi) \bmod 2\pi) - \pi| \tag{4.2}$$

In this formula, θ_{est} and θ_{gt} represent the estimated and ground-truth orientations of the marker, respectively, expressed in radians. The modular arithmetic ensures that the error remains within the range $[0, \pi]$, accounting for the circular nature of angles.

These error metrics are computed for each individual marker in the aligned map. By computing the mean translation and rotation error across all markers, it becomes possible to evaluate the overall quality of the reconstructed map and assess the system's spatial accuracy.

4.2 Simulation-Based Evaluation

To establish a controlled baseline for assessing system performance, an evaluation was conducted in a simulated environment. This simulation facilitates the isolation of algorithmic behavior from real-world noise sources such as lens distortion, mechanical imprecision, and sensor noise. By modeling the virtual workfield in Unity, complete control was retained over both the camera's pose and the placement of ArUco markers, enabling precise and repeatable validation.

The virtual camera was positioned at a fixed height and angle above the planar workspace, replicating the assumptions used throughout the mapping and tracking pipeline. Marker positions were generated according to a set of predefined layouts, identical to the ones used in the real-world experiments. As the camera moved across the scene, synthetic images can be streamed directly into the localization pipeline, or recorded to be used as video for the system. The ground-truth pose provided directly by Unity is then recorded frame by frame so it can later be compared to the systems estimated pose.

This environment offers the opportunity to rigorously quantify the pose estimation accuracy without interference from uncontrolled physical variables. The results serve both to validate the core geometric assumptions of the system and to identify any systematic deviation introduced by the estimation procedure itself.

4.2.1 Simulation Setup

The simulation was developed in Unity to serve as a deterministic and noise-free testbed for validating the system's localization accuracy. In this environment, a flat virtual plane was populated with ArUco markers at fixed, known positions, mimicking the physical workspace. The markers were placed according to the previously mentioned predefined map that matches the ones used in later real-world experiments, with their positions stored in millimeters to be consistent with the real-world coordinate system.

The camera model in Unity was configured to closely resemble the physical setup. It was positioned exactly 150 mm above the plane and angled downward at precisely 60 degrees. This choice replicates the fixed, slightly oblique top-down view used by the real system, allowing the same pose estimation pipeline to be reused without modification. No synthetic lens distortion or sensor noise was introduced, thereby isolating algorithmic performance from physical artifacts.

During the simulation, the camera can be manually translated across the workspace to ensure that each marker became visible in multiple frames from different viewpoints. Two modes of operation were available for this process. In the first mode, the simulation streamed each rendered image frame directly to the pose estimation system without logging the corresponding ground-truth pose. This streaming mode proved useful during the setup phase, allowing for real-time testing and visual inspection while manually positioning the camera.

In the second mode, the simulation recorded a video sequence in which both the rendered image and the exact ground-truth pose of the camera were logged for every frame. This recording mode enabled precise validation of the pose estimation pipeline, as it provided a one-to-one association between the frame and ground-truth camera pose. Regardless of the mode used, the image frames were processed using the same detection and localization routines as in the real-world system, including marker detection, frame-to-map alignment, and pose estimation.

4.2.2 Limitations

While the Unity simulation provides a controlled environment for evaluating localization accuracy, it inevitably simplifies several aspects of the real-world system. Most notably, the virtual camera is assumed to have a perfect pinhole model, free of any lens distortion, motion blur, or sensor noise. In contrast, real-world cameras exhibit nonlinear distortions and imperfections that can significantly impact marker detection and pose estimation. Additionally, lighting conditions, marker print quality, and surface reflections—all absent in the simulation—can introduce variability in real deployments.

Beyond optical considerations, the simulation also omits mechanical imperfections inherent to physical setups. Real systems may suffer from slight misalignments, mechanical backlash, or vibrations introduced by the tool or environment. These deviations, though difficult to precisely model or predict, can significantly influence the system's accuracy and must be taken into account when interpreting experimental results obtained from physical hardware.

Another significant limitation of the simulation is its assumption of perfect marker quality and detection. In Unity, all markers are rendered with ideal sharpness, contrast, and flatness, ensuring that they are always fully visible and easily detectable. However, in real-world conditions, marker quality can vary substantially due to imperfections in printing, wear and tear, uneven surfaces, or reflections caused by glossy finishes. Such imperfections may lead to failed detections, false positives, or inaccuracies in pose estimation. Moreover, the detection process in simulation assumes that markers are always fully in frame, and not occluded. In practice, markers may be partially visible, or momentarily obscured by the operator's hand. These factors can lead to unreliable detections or fluctuations in estimated poses, which are not captured in the simplified simulation environment. As a result, performance observed in simulation may overestimate the system's robustness under realistic conditions. This can be seen in the following set of tests where Unity outperforms the other camera's in marker detection quality. The only imperfections introduced here are because of the frame resolution.

4.3 Squareness Evaluation of Detected Markers

A crucial part of the system's accuracy depends on how well the camera captures and processes the geometry of the scene. Since the ArUco markers serve as fixed reference points for both mapping and pose estimation, any distortion in their detected shape directly impacts the precision of the system. In particular, the squareness of the detected markers is a key indicator of geometric accuracy. Especially since the following step in the pipeline explained in subsection 3.3.4 heavily relies on the assumption of the markers being a perfect square.

This squareness is influenced by several factors. First, it depends on the accuracy of the perspective transformation used to generate a top-down view of the scene. If this transformation is poorly estimated, it introduces skewing and stretching that degrade the marker's shape. Second, it depends on the quality of the marker detection algorithm itself, which must localize corners as precisely as possible, ideally down to the sub-pixel level. Finally, the actual physical squareness and flatness of the printed marker also plays a role: a wrinkled or slightly curved paper can produce perfectly valid but visually distorted detections, misleading further processing steps.

Because later stages in the system rely on the assumption that the markers are square and planar, it is essential to validate this assumption experimentally. Any deviation from squareness at this early stage can propagate through the pipeline, reducing the overall fidelity of the spatial model. Therefore, in this test, the squareness of the detected markers is measured and compared under different conditions, including variations in camera type, marker quality, and marker type. The results allow us to quantify how much each of these factors contributes to geometric distortion and help identify which configurations are best suited for reliable operation.

4.3.1 Experimental Setup

To evaluate the squareness of detected markers, an experiment was designed to isolate the effects of camera quality, physical marker quality, and marker type. For this purpose, two different cameras were used: a Logitech 960-000348 webcam, representing a low-cost, consumer-grade imaging device, and an Intel RealSense camera, representing a higher-quality camera commonly used in robotics and computer vision applications. The RealSense camera also has depth imaging capabilities wich will not be utilized in these tests.

In this experiment, only ArUcoE markers were used, which incorporate both a standard ArUco marker in the center and an extended border that allows for enhanced detection accuracy. To evaluate the impact of the detection method itself, two steps in the detection pipeline were considered separately. First, markers were detected using the standard ArUco detector, which provides pixel-level corner positions. Then, in a second stage, the same markers were passed through the ArUcoE refinement step, which enhances the initial detection to achieve subpixel accuracy. This separation allows for a direct comparison between the base ArUco detection and the refined ArUcoE results using the exact same physical markers.

To assess the influence of physical marker quality, the ArUcoE markers were printed in two different ways. In one case, they were printed on standard office paper, which is susceptible to slight warping and curling. In the other, the same markers were printed on high-quality paper and glued to a flat wooden board to ensure maximum flatness and physical squareness.

For each combination of camera, detection method, and marker material, a series of top-down views was generated using the perspective transformation pipeline described in Section 3.2. The squareness of each detected marker in these views was then evaluated using the method outlined in Subsection 3.3.3. In short this metric is based on a geometric consistency check of the detected marker corners: it compares the lengths of the two diagonals, as well as the four individual edges, of the quadrilateral. In the case of a perfect square, the diagonals should be equal in length, and all four edges should be of equal length as well. Deviations from these ideal conditions indicate geometric distortion, which can result from inaccuracies in the perspective transformation, imprecise corner detection, or physical deformation of the marker.

Each configuration was tested multiple times to ensure statistical robustness. This setup allows for a direct comparison of the impact of image quality and marker flatness on detection accuracy, as well as the performance difference between ArUco and ArUcoE detection algorithms.

4.3.2 Marker Squareness Results and Analysis

Figure 4.1 shows the results of the marker squareness validation test, comparing standard ArUco markers (in blue) with ArUcoE markers (in orange). At first glance, the squareness scores for standard ArUco markers may appear acceptable. However, these results are somewhat misleading due to a fundamental artifact of how the top-down view is constructed.

Standard ArUco markers are detected as pixel-aligned squares, meaning their apparent geometry is heavily influenced by the resolution of the input image and, more critically, by the homography used to transform the camera frame into a top-down perspective. This transformation tends to "snap" detected markers to the pixel grid, often aligning them in such a way that even imperfect detections appear geometrically sound. As a result, the error is often bounded to within half a pixel, giving the illusion of high accuracy. However, the actual shape consistency of the underlying detection is not well captured, and the resulting scores are unreliable. In essence, these results reflect properties of the image transformation rather than the quality of the marker detection itself.

This effect can also be observed in the results from the Unity simulation. Despite the idealized environment and perfect camera model, the squareness scores for standard ArUco markers are not significantly better than those from physical systems—and in some cases are worse. This is because even in simulation, the final frame is rendered at 1080p resolution, causing the markers to become pixel-aligned just as they would in real-world images. As a result, their apparent shape is artificially altered. In contrast, the ArUcoE markers in Unity demonstrate exceptionally high geometric fidelity, as the subpixel refinement process can take full advantage of the perfect simulated world. This stark difference reinforces the conclusion that standard ArUco squareness scores are not a reliable indicator of detection quality.

In contrast, the ArUcoE markers demonstrate much more meaningful results, especially when printed at high quality and observed using a higher-resolution camera. The addition of the checkerboard border allows for true subpixel corner refinement, revealing actual improvements in geometric fidelity. In these cases, the squareness scores not only become more consistent but also significantly better on average. As can be seen in the figure, the performance of ArUcoE under optimal print and camera conditions approaches the idealized performance observed in the Unity simulation, confirming its effectiveness in improving marker-based pose estimation.

Since the results show a clear improvement when using higher-quality printed markers, it is evident that print quality and surface flatness play a significant role in the accuracy of marker detection. However, it remains unclear whether the current "good" print-produced to the highest feasible standard within the constraints of this thesis—fully eliminates all physical imperfections. It is possible that further improvements in print resolution, material stiffness, or mounting precision could yield even better results. If we could assume a perfectly printed and flat marker sheet, then this squareness validation would provide a reliable metric for evaluating the quality of the overall detection and calibration pipeline. As it stands, the test gives useful comparative insights, but its results are still partially influenced by residual imperfections in the physical setup.

The results validate the hypothesis that extending the marker with additional high-contrast

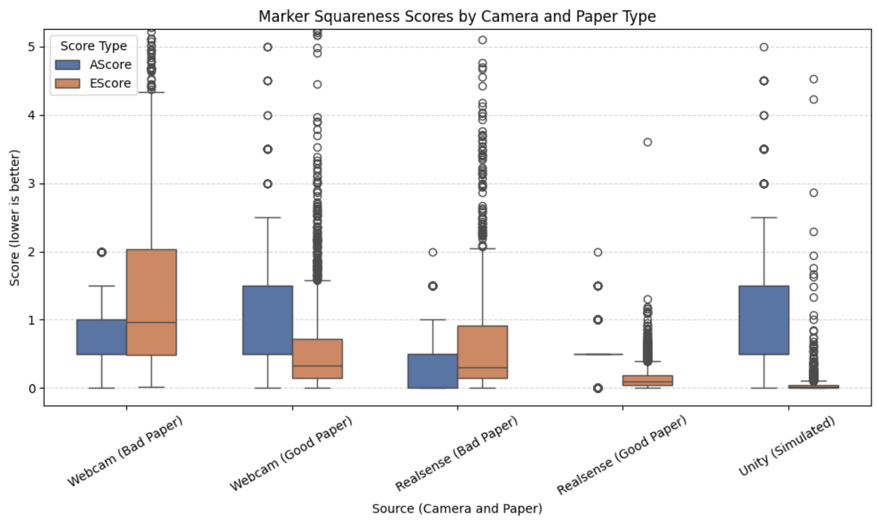


Figure 4.1: The results of the squareness test. ArUco in blue, ArUcoE in orange

features significantly improves the accuracy of the detected shape. In practical terms, this means that pose estimation based on ArUcoE markers is likely to be more robust, particularly in applications where high precision is required. As such, ArUcoE markers were selected as the default fiducial system for the remainder of the experiments in this thesis.

4.4 Map Accuracy

Accurate map construction is the foundation upon which the entire localization and tracking pipeline depends. Any errors introduced during the setup phase will directly degrade real-time pose estimation and, consequently, the precision of toolpath execution. The goal of this section is to quantify the system's ability to generate a globally consistent map of fiducial markers under a variety of conditions, and to identify which mapping strategies and marker layouts yield the highest accuracy.

To this end, we evaluate three successive stages of the mapping pipeline: (1) a baseline map generated using only standard ArUco markers, (2) an enhanced map generated with ArUcoE markers that provide improved corner detection, and (3) a refined map produced by applying Bundle Adjustment (BA) to the ArUcoE result. By scoring the reconstructed map at each stage against the ground truth, we can assess the incremental benefits of marker enhancement and global optimization.

In addition to mapping methodology, we investigate how marker layout and density affect map accuracy. Five layout configurations are tested: a highly dense, structured grid serving as an ideal baseline; three randomly distributed layouts with decreasing densities (dense, medium, and sparse); and a worst-case arrangement in which no more than two markers are visible in any single frame.

By systematically varying both the mapping algorithms and the spatial arrangement of markers, this section aims to reveal the trade-offs between setup effort, marker density, and map fidelity. The results will inform guidelines for practical deployment, ensuring that the workfield can be mapped both quickly and accurately to support high-precision tool operations.

4.4.1 Experimental Setup

To evaluate the accuracy of the mapping process, a series of controlled experiments were conducted using different marker layouts and multiple camera configurations. First the different layouts are created as described in Section 4.1. Then for each trial, that map was generated using the method discussed in Section 3.4,and the corresponding map error was computed according to the metrics defined in Section 4.1. This process was repeated multiple times per layout and camera type to ensure statistical significance and reproducibility.

Each experiment followed the same procedure: the handheld tool was moved manually across the workspace, allowing the camera to observe and register all markers at least once. This camera feed was then processed by the mapping system to build a global map of marker positions. Depending on the specific configuration under test, this map was constructed using only enhanced ArUcoE detection, or further refined using bundle adjustment (BA). At each stage, the quality of the resulting map was scored independently to assess the incremental benefits of each technique.

The tests were repeated across multiple hardware setups, including the same standard webcam, the Intel RealSense camera, and synthetic data generated within the Unity simulation environment, as used in earlier experiments. For the physical tests, the same marker maps were printed at large scale on A0 sheets. This ensured that the spatial layout of the markers exactly matched the digital versions, while also allowing the tests to be conducted over a larger physical area. High-quality printing was used to minimize geometric distortions and maintain marker fidelity, consistent with the considerations discussed in Section 4.3. This setup enabled a fair comparison between different camera systems and the simulated baseline.

The worst-case scenario map was constructed using the same marker layout as the structured grid map. However, to simulate maximal error accumulation, all markers were deliberately covered except for two: the previously added marker and the new marker being integrated. This setup forces the system to rely solely on a single reference marker for each new addition, eliminating the benefits of overlapping observations. As a result, any pose estimation errors introduced in one step are fully propagated to subsequent markers without correction, effectively simulating the most unfavorable mapping conditions. This configuration serves to illustrate the upper bound of accumulated error in the absence of redundancy.

4.4.2 Results

Table 4.1: Average map error (in mm) for different mapping configurations across hardware setups and map types.

Camera	**Method**	**Grid**	**Random Layouts**			**Worst Case**
			Dense	Medium	Sparse	
	ArUcoE	4.39	3.51	3.60	3.86	11.38
Webcam	+BA	4.39	3.51	3.60	3.86	11.38
	ArUcoE	4.03	1.87	2.02	3.24	5.08
RealSense	+BA	4.03	1.87	2.02	3.24	5.08
	ArUcoE	0.43	0.35	0.39	0.53	1.43
Unity	+BA	0.43	0.35	0.39	0.53	1.43

Table 4.1 shows the average map reconstruction error in millimeters across different mapping methods, hardware setups, and marker layout configurations. As expected, the Unity simulation produced the lowest error across all layout types, with sub-millimeter accuracy in both dense and sparse configurations. The RealSense camera consistently outperformed the standard webcam, achieving lower average errors across all scenarios due to its superior optics and depth sensing capabilities.

For the webcam, map error remained relatively stable across all random layouts when using ArUcoE markers, ranging from 3.51mm to 3.86mm. However, under the worst-case layout, the error increased sharply to over 11 mm, highlighting the importance of marker overlap for achieving accurate global positioning. Surprisingly, applying bundle adjustment (BA) to both the webcam and RealSense data did not yield measurable improvements.

The RealSense camera showed greater sensitivity to marker layout density, with the error rising from 1.87mm in the dense random layout to 3.24mm in the sparse case. Nonetheless, even in the worst-case scenario, the RealSense maintained an error of only 5.08 mm—less than half that of the webcam under the same conditions.

In the Unity simulation, map accuracy was largely unaffected by layout sparsity, with only a minor increase in error from 0.35mm to 0.53mm between the dense and sparse configurations. This stability confirms the validity of the mapping pipeline under ideal conditions and serves as a lower bound reference for physical system performance.

4.4.3 Analysis and Discussion

The experimental results highlight several important trade-offs and insights relevant to real-world deployment of the mapping system. First, the choice of camera has a significant impact on mapping accuracy. The RealSense camera consistently achieved lower error than the webcam, particularly in sparse layouts where the advantage of improved image quality and depth perception is most pronounced. In contrast, the webcam suffered heavily with the worst-case scenario, with error accumulating rapidly.

Second, the mapping pipeline using ArUcoE markers already produces high-quality results, and applying bundle adjustment (BA) on top of it did not lead to measurable improvements in map accuracy, despite the fact that the map returned by BA was noticeably different from the initial amp. This suggests that while the optimization does modify the map geometry, it does not necessarily reduce the overall reconstruction error according to the chosen metric. The lack of improvement could indicate that the initial map is not yet truly optimal, but that the potential gains from BA are masked by factors such as sensor noise or limitations in the custom BA implementation. For instance, numerical instability, incorrect Jacobians, or suboptimal weighting of residuals may prevent the optimizer from converging to a better solution. Further investigation is needed to determine whether the issue lies in the quality of the input data or in the implementation details of the optimization process.

Third, marker layout and density have a clear effect on map fidelity. While dense, structured grids produced the most accurate results, the improvement from medium to dense layouts was relatively small compared to the more noticeable drop in accuracy from medium to sparse configurations. This suggests diminishing returns: adding more markers beyond a certain point yields only marginal accuracy gains, despite significantly increasing the setup effort. Since placing additional markers requires time, this trade-off becomes important in practical scenarios. The worst-case scenario, which limited visibility to only two markers per frame, resulted in the highest errors due to the compounding of uncorrected pose estimates. Overall, these findings highlight that while overlapping marker observations are crucial for maintaining global consistency, there is a balance to be struck between accuracy and setup complexity.

Third, marker layout and density have a clear effect on map fidelity. While dense, structured grids yielded the best results, the improvement over medium-density layouts was relatively small compared to the more noticeable drop seen with sparse configurations. This suggests diminishing returns, where increasing marker count adds setup effort without proportional accuracy gains. These results reinforce the trade-offs outlined in Subsections 3.2 and 3.3.1: although higher density supports both mapping accuracy and continuous tracking, it also limits scalability and increases setup complexity. The findings confirm that careful tuning of marker density is essential to balance tracking reliability, mapping performance, and practical deployment effort.

Finally, the Unity simulation results validate the expected theoretical performance of the system. The negligible error across all configurations confirms the correctness of the underlying algorithms in an idealized setting, and provides a valuable benchmark for interpreting deviations observed in the physical experiments.

4.5 Pose Accuracy

An essential component of the proposed system is its ability to accurately estimate the pose of the handheld tool during operation. While the previous section focused on evaluating the internal consistency and geometric fidelity of the generated marker maps, this section aims to assess the end-to-end performance of the complete pose estimation pipeline. Accurate localization is critical, as any correctional motion of the movable tip relies on the correctness of the estimated pose.

To rigorously evaluate pose accuracy, experiments are conducted within the simulated environment (Unity), where the ground truth pose of the camera is available at every frame. This allows for a direct and quantitative comparison between the system's estimated pose and the actual pose of the tool. The simulation is configured in recording mode, as described in Section 4.2.1, enabling the collection of synchronized image frames and precise pose annotations.

The recorded image frames are processed by the pose estimation pipeline, identical to the one used during live operation. By comparing the estimated poses to the known ground truth, both translational and rotational errors can be computed. This evaluation provides insight into the system's real-world tracking potential and highlights the influence of map quality, marker layout, and detection fidelity on localization performance.

4.5.1 Evaluation Metrics

To quantitatively assess the accuracy of the system's pose estimation, two primary metrics are used: translational error (e_t) and rotational error (e_r). These metrics enable a direct comparison between the estimated camera pose produced by the system and the ground truth pose obtained from the simulation environment.

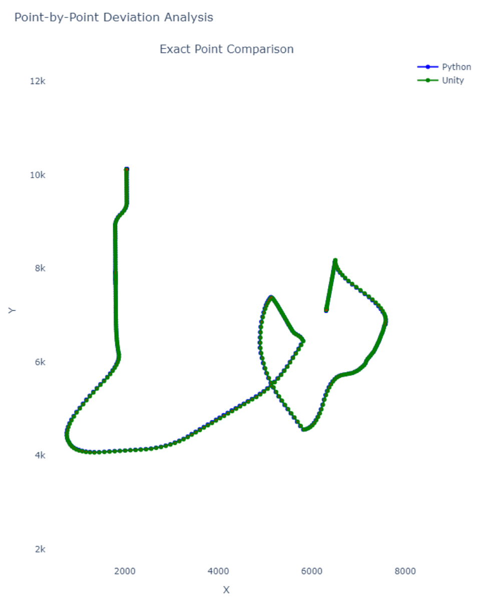
As in the map accuracy evaluation, the estimated and ground truth poses do not initially share a common coordinate frame. Therefore, a rigid transformation is first computed to align the estimated map to the ground truth map. This transformation is subsequently applied to the estimated camera poses, ensuring they are expressed in the same coordinate system as the ground truth. Once aligned, the translational and rotational errors are computed using the same formulations described in Section 4.1, but applied to the camera poses rather than the marker positions. The translational error measures the Euclidean distance between the aligned estimated and ground truth positions, while the rotational error captures the angular discrepancy, properly accounting for angle wrapping.

Map Setups This evaluation is conducted using a variety of marker maps, including those generated by the physical cameras as described in Section 4.4. These maps are loaded into the Unity simulation environment, allowing the system to estimate its pose relative to each map. The same map layouts will be used as in the previous experiment 4.4, the resulting maps of each of the cameras of this experiment can then be used here to estimate the position. This allows for a controlled comparison and provides insight into how the quality of the underlying map influences the final pose estimation accuracy, and consequently, the effectiveness of the system as a whole. In addition an idealized ground truth map—constructed directly from the marker generation process described in Section 4.1—is included as a baseline.

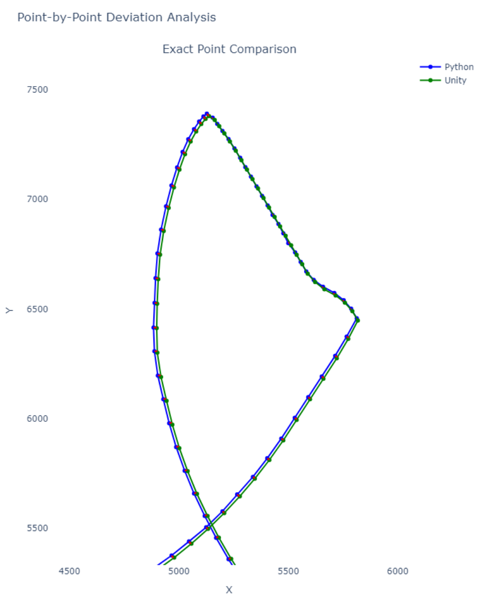
This comparison aims to provide a comprehensive understanding of how both marker layout and the quality of the underlying map impact the final pose estimation performance. It also helps to identify practical limitations of the system when used with lower-fidelity maps, such as those obtained from low-cost webcams or in sparse marker configurations.

4.5.2 Results

To quantify the influence of map quality and layout on pose accuracy, a systematic evaluation is conducted using four different marker distributions as used in Section 4.4 (structured grid, random dense, random medium and random sparse), each processed under three mapping conditions: a perfect map (ground truth), and maps constructed using image data from the Webcam, and the RealSense camera. The results are summarized in Tables 4.2 and 4.3, which report the average translational and rotational errors (e_t and e_r, respectively) computed over all frames for each configuration.



(a) Full path over the 1 m × 1 m workspace.



(b) Zoomed-in view highlighting pose deviations.

Figure 4.2: Comparison between the ground truth trajectory from Unity (green) and the system's estimated poses (blue). Subfigure (a) shows the complete traversal path using a 1 m × 1 m map, while subfigure (b) presents a zoomed-in view to visualize the deviations between estimated and true poses more clearly.

Figure 4.2 presents a qualitative visualization of the system's pose estimation accuracy when using a perfect map as generated directly from ground truth marker positions (cf. Section 4.1). The setup used here corresponds to a randomly generated medium-density layout, comprising 50 markers uniformly distributed over a 1 m × 1 m workspace. The green line indicates the ground truth camera path as recorded by Unity, while the blue line shows the corresponding pose estimates produced by the system. As shown in the zoomed-in view (Figure 4.2b), the deviation between the estimated and true trajectories is minimal, demonstrating the system's ability to maintain high pose accuracy when operating with an ideal reference map.

Table 4.2: Average pose estimation translation errors e_t (in mm) for different map types and mapping methods.

Map Type	Perfect Map	Webcam	RealSense
Structured Grid	0.730	1.911	1.308
Random Dense	0.649	1.753	1.188
Random Medium	0.706	1.529	1.012
Random Sparse	0.673	2.186	0.901

Table 4.3: Average pose estimation rotation errors e_r (in degrees) for different map types and mapping methods.

Map Type	Perfect Map	Webcam	RealSense
Structured Grid	1.84	1.92	2.04
Random Dense	1.93	2.01	2.81
Random Medium	1.59	1.55	1.62
Random Sparse	1.09	1.16	1.33

4.5.3 Analysis and Discussion

The results in Tables 4.2 and 4.3 reveal several important trends regarding the impact of map quality and layout on pose estimation accuracy.

First, as expected, using the perfect map consistently yields the lowest translation errors across all layout types, confirming the effectiveness of the pose estimation algorithm when decoupled from mapping inaccuracies. This is further illustrated in Figure 4.2, where the deviation from ground truth remains minimal throughout the entire path. However, even with perfect maps, the translational error does not drop below 0.6 mm–0.7 mm, indicating that other factors such as marker detection uncertainty and camera calibration imperfections also contribute to the overall error budget.

When comparing the Webcam and RealSense results, the RealSense generally achieves lower translation errors, especially in the dense and medium random layouts. This suggests that improved sensor quality can offer tangible benefits during map construction. However, the performance advantage diminishes in the sparse density, where limited marker visibility appears to become the dominant bottleneck regardless of camera quality.

Interestingly, higher marker density does not necessarily guarantee better results. The medium-density layout outperforms both the Structured Grid and the Dense configurations, most notably in rotational error. This may be due to the increased viewpoint diversity offered by the randomized placement, which results in more varied angles and positions from which markers are observed. Such variation can help reduce ambiguity during pose estimation and improve robustness, whereas structured or overly dense layouts may lead to more repetitive or less informative observations.

An important observation is that in the sparse layout, the camera temporarily lost sight of all markers during a few frames. However, since the camera was stationary and only rotating at the time, this had little impact on the reported results. While the sparse configuration still achieved reasonable accuracy overall, this incident highlights its reduced reliability compared to denser layouts.

Finally, rotation errors appear to be less sensitive to layout variation than translation errors. All configurations maintain sub-3° rotation accuracy, and even sparse layouts yield competitive results, suggesting that orientation estimates are more robust to mapping imperfections.

The visualization in figure 4.2 reveals a notable relationship between motion speed and pose accuracy. In regions where the camera moves more slowly—evident from closely spaced datapoints—the estimated poses remain tightly aligned with the ground truth. Conversely, in segments where the datapoints are further apart, indicating more displacement per frame and thus higher speed, a slight increase in estimation error is observed.

Overall, these results validate the design choices and trade-offs discussed earlier: moderate marker density with good coverage provides a strong balance between setup effort and system performance, while more expensive setups only yield marginal accuracy gains if not losses.

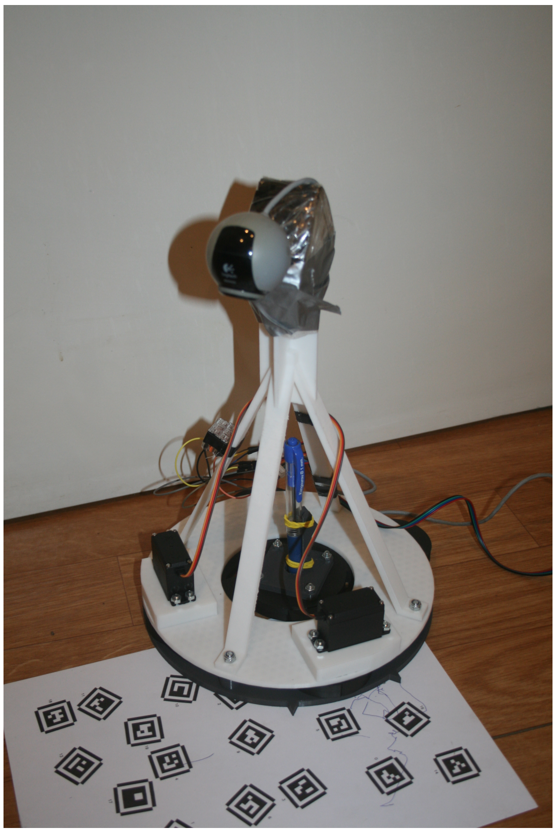


Figure 4.3: The prototype with a movable pen using the webcam

4.6 Physical Prototype

To complement the simulation-based results and evaluate the practical applicability of the system, a physical prototype was constructed.

The prototype uses two servo motors to control motion in the x and y directions, a cam is mounted off-center to these servos and these are linked with 2 free to rotate arms, where the pen is placed in the pivot-point between the 2 arms. This allows for a small but robust range of motion, using only the constrained 180°rotational range of these servo motors. The goal of this setup is to trace a predefined polygon on paper using only the system's localization and control logic, allowing visual inspection of the accuracy and reliability of the system.

This setup serves as an end-to-end test of the entire pipeline, from image acquisition and pose estimation to motion planning and physical actuation. Unlike the Unity simulation, this test introduces real-world variables such as marker detection noise, mechanical backlash, and imperfect alignment of the pen with the camera's coordinate system.

Although this experiment does not yield exact numerical accuracy metrics like in the simulation, it offers a valuable qualitative assessment. Deviations between the printed polygon and the drawn line can be directly observed, and the system's ability to maintain alignment throughout the path provides insight into both static accuracy and dynamic tracking performance. This demonstration validates that the implemented system is capable of performing closed-loop control of a tool in physical space based solely on camera input and precomputed marker geometry.

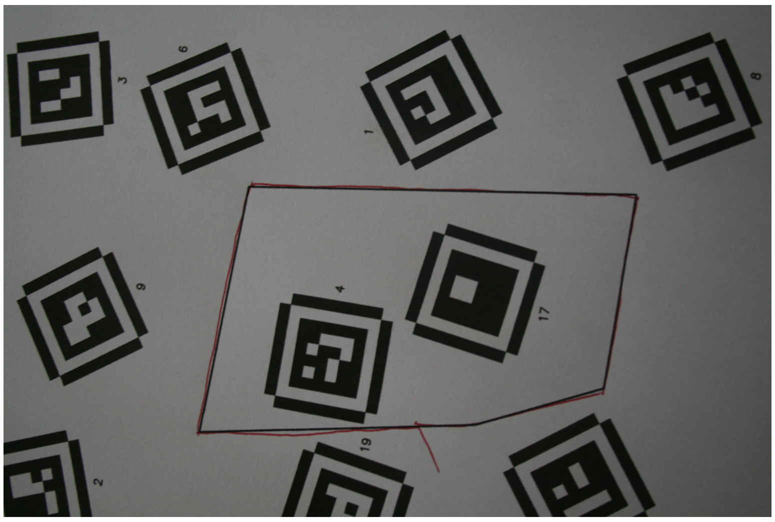


Figure 4.4: This is how the results of this test would look like if they woul have been finished. The printed polygon is the ground truth and the red line representing the path followed by the pen

Physical System Description

The physical prototype is designed to replicate the key functionalities of a handheld CNC tool under controlled, planar conditions. It consists of a custom-built 2D plotter mechanism driven by two servo motors, responsible for movement along the x and y axes. To start drawing the pen can simply be actuated to extend its tip. This configuration enables the system to draw paths on paper, effectively emulating the cutting or marking actions of a CNC head.

The camera used for localization is rigidly mounted on the moving carriage, preserving a fixed spatial relationship between the tool (the pen tip) and the camera frame. This ensures that any movement estimated by the visual tracking system directly translates to physical tool movement. The same webcam as in the previous tests is employed, and its intrinsic parameters are calibrated using Zhang's method to mitigate lens distortion during marker detection.

The marker map is printed on a single sheet of paper, which also includes the predefined polygonal path that the tool is intended to trace. Both the markers and the path are positioned in a shared coordinate system, ensuring that the printed path aligns precisely with the reference trajectory defined in software. 20 ArUco markers are distributed in a random layout across the sheet to provide robust and redundant localization throughout the workspace. This setup allows for a direct visual comparison between the commanded path and the actual trajectory drawn by the tool, enabling qualitative assessment of system accuracy and stability.

This setup allows for the full pipeline to be tested: the system detects markers, estimates the tool's position in real time, and drives the motors so the pen follows a virtual path that is mapped onto the physical workspace. This provides a robust framework for validating both the localization and control subsystems under realistic operating conditions.

Drawing Execution and Error Measurement

Once the system has initialized and constructed the marker-based workspace map, it proceeds to execute the drawing task. The user manually guides the tool along the printed path while the system continuously estimates the tool's position and actuates the pen to follow the corresponding virtual trajectory. Because the printed shape and the marker map are defined in the same coordinate system, the virtual path should ideally align perfectly with the printed one. As a result, the line drawn by the system should trace the printed shape exactly. Any deviations the drawn line has from the printed line serve as visual indicators of pose estimation errors and control inaccuracies. This method provides a straightforward yet effective means of evaluating the system's real-world performance, capturing both systematic biases and dynamic tracking limitations.

Discussion

At the time of writing, the physical drawing experiment—intended to validate the system's performance under real-world conditions—has not yet been fully completed, due to unexpected hardware issues. However, the setup is in its final stages, and preliminary testing indicates that the prototype is functioning as expected. All components required for the experiment are operational, and the drawing procedure is currently being finalized. The results and analysis, while not included in this thesis, are expected to closely align with the observations and expectations outlined earlier. Once completed, this experiment will provide valuable insights into the system's ability to generalize from simulation to physical deployment, particularly with respect to mechanical tolerances and sensor robustness. Figure 4.4 shows the expected results of this test although here it is drawn by hand using a red pen.

Chapter 5

Discussion & Conclusion

The primary objective of this thesis was to develop and evaluate a vision-based localization system for a handheld CNC tool, inspired by systems like the Shaper Origin. The system leverages a downward-facing camera rigidly mounted on the tool to observe a printed workspace containing known ArUco markers. By detecting these markers and matching them to a preconstructed global map, the system continuously estimates the camera's pose, and by extension the pose of the tool tip, with respect to the workspace.

To achieve this, the project was divided into two main phases: (1) map construction, during which a mosaic of the workspace is built and marker positions are registered in global coordinates, and (2) online localization, where real-time pose estimation is performed using visible markers in each frame. The accuracy and robustness of this pipeline were evaluated through rigorous tests, a Unity-based simulation and a physical validation prototype. These experiments served to assess whether such a system could reliably enable camera-guided control of a CNC tool in both virtual and real environments.

5.1 Key Findings

The evaluation revealed several key insights into the system's mapping and localization performance. First, reliable pose estimation can be achieved using camera-based detection of ArUcoE markers, even in real-time conditions. The initial mapping pipeline already produces accurate maps, with minimal improvement observed when applying bundle adjustment—likely due to imperfections in the implementation, as the resulting map structure differed but yielded similar accuracy metrics.

Second, marker layout plays a significant role in accuracy. Denser layouts generally improve results, but the difference between medium and dense configurations was relatively minor compared to the large drop-off in performance from medium to sparse. This suggests diminishing returns: increasing marker density beyond a certain point requires disproportionately more setup effort without substantial accuracy gains.

Interestingly, medium-density random layouts occasionally outperformed structured grids, particularly in rotational accuracy. This may be due to the increased diversity in viewing angles and marker arrangements, which help improve triangulation and reduce systematic biases.

Overall, the experiments validate the feasibility of accurate, marker-based localization under realistic constraints, while also highlighting important trade-offs between accuracy, setup complexity, and marker density.

5.2 Limitations

While the system demonstrates strong potential through both simulation and real-world validation, several limitations remain that affect its generalizability, scalability, and precision.

5.2.1 Limitations of the Unity Simulation

As discussed in Section 4.2.2, the Unity-based simulation offers a controlled environment ideal for algorithm validation and performance benchmarking. However, it inevitably abstracts away many complexities present in real-world scenarios. The virtual camera is modeled as an ideal pinhole device, free from lens distortion, motion blur, or image noise. Lighting is perfectly uniform, marker detection is deterministic, and all markers are treated as planar, perfectly printed, and fully visible.

Moreover, the simulation lacks mechanical constraints such as vibration, backlash, or drift, which are common in physical setups. It also assumes perfect marker placement without any physical deformations, curling, or misalignment. As a result, while useful for identifying baseline behavior, the simulation may overestimate the real-world performance of the system.

5.2.2 Uniform Marker Detection Quality

A further limitation lies in the assumption of uniform marker quality and detectability. In both the simulation and early prototype phases, all markers are considered equally sharp, well-printed, and free of degradation. In reality, variations in printer quality, paper flatness, surface reflectance, or ink bleeding can significantly impact the detection confidence and the precision of subpixel corner estimation.

This issue is particularly pronounced for real-world applications where markers may be reused, worn, or exposed to uneven lighting. The current pipeline does not yet incorporate per-marker quality assessment or adaptive weighting during optimization, which may lead to localized inaccuracies if poorly detected markers are treated with equal confidence.

5.2.3 Lack of Dynamic Error Handling

The system currently does not include mechanisms for dynamic error detection or recovery. For instance, if a faulty marker enters the field of view or if tracking is lost mid-operation due to insufficient marker visibility, the system cannot yet gracefully degrade or correct itself. Incorporating confidence metrics or fallback heuristics remains an important avenue for future work.

5.2.4 Calibration Sensitivity

Both intrinsic and extrinsic calibrations are performed manually and assumed to remain valid throughout operation. Any minor shift in camera alignment, marker scale, or surface geometry can invalidate these calibrations, leading to drift or incorrect pose estimates. A more robust calibration protocol—or runtime self-checking—would significantly enhance system stability in uncontrolled environments.

5.2.5 Limitations of the Physical Prototype

The current physical system represents an early-stage prototype, and as such, it includes several limitations that impact the accuracy and repeatability of the experimental results. The entire structure is 3D-printed and assembled from low-cost components, which introduces several sources of mechanical and structural error.

One of the most significant limitations is the mechanical rigidity of the camera mounting bracket. Although the camera is fixed to the handheld tool, there remains a small degree of flex and play between the camera and the tool body. Even slight shifts or vibrations can invalidate the extrinsic calibration, as the system assumes a fixed and rigid transformation between the camera and the tool's reference point. This can lead to persistent pose estimation errors, especially during rapid movements or when the system experiences mechanical stress.

Additionally, the stepper motors used for actuation are open-loop and of relatively low precision. They are prone to skipping steps under load or during fast direction changes, and there is no feedback mechanism in place to detect or correct for such errors. This directly affects the accuracy of drawn paths and introduces drift over time. Furthermore, mechanical backlash in the belt and pulley system and looseness in 3D-printed joints reduce the positional fidelity of the tool tip, especially during direction reversals.

Thermal expansion, surface friction, and the alignment of the pen mechanism also introduce variability in the tool's motion and drawing accuracy. Combined with the lack of consistent pen pressure and occasional vertical play in the servo mount, these factors make it difficult to isolate pose estimation errors from mechanical imprecision in the output.

While the prototype successfully demonstrates the feasibility of the system's core concepts, these hardware limitations must be addressed in future iterations to unlock the system's full potential and allow for meaningful high-precision evaluations.

5.3 Future Work

While the current prototype lays a solid foundation for vision-based localization in a handheld CNC tool, it has not yet been fully completed or experimentally validated. Finalizing and testing the physical system is therefore the immediate priority. Once this is achieved, several avenues for further development become possible, including system refinement, improved robustness, and functional extensions such as automatic path correction or adaptive control based on material feedback.

Another key focus lies in improving the mechanical quality of the prototype. This includes replacing the 3D-printed structural components with more rigid structures or materials, reinforcing the camera mount to eliminate flex, and ensuring a robust and repeatable alignment between the camera and the tool's reference point. Additionally, upgrading to closed-loop stepper motors or integrating encoder feedback can help by achieving a higher resolution than the used servos, leading to increased accuracy and improve motion fidelity. These changes would not only enhance accuracy but also enable more reliable long-term operation.

Beyond refinement, future work aims to extend the system from a proof-of-concept drawing tool into a functional handheld router. In collaboration with a colleague, the goal is to create a system similar in spirit to the Shaper Origin, where the user roughly traces a predefined path and the tool actively corrects deviations in real time. This will require integration with a compact, lightweight routing mechanism and improvements to both real-time control and safety handling.

Another potential direction for future work involves removing the need for an explicit mapping phase in scenarios where precise alignment of the operation with the physical workspace is not critical. In such cases, the user might simply specify a starting point and general direction (e.g., "start here and proceed along this orientation"), allowing the system to begin tracking and executing the operation immediately. In this mode, the system could simultaneously build the map and localize itself on the fly. While this would reduce setup time and streamline the user experience, it introduces new challenges. In particular, global optimization techniques like bundle adjustment rely on a complete set of observations and overlapping constraints across frames—something that may not be feasible during live operation. As a result, pose accuracy may degrade over time without the benefit of post-hoc correction. Exploring whether real-time SLAM-like approaches or delayed map refinement could offset this limitation would be an interesting avenue for future investigation.

Looking further ahead, a more ambitious goal is to adapt the system for use with a handheld plasma cutter. While this would demand substantial mechanical redesign—particularly in terms of mounting, thermal shielding, and cable management—the core localization and control software is expected to transfer with minimal changes. This transition would highlight the modularity and adaptability of the approach, enabling the same vision-based localization system to serve a variety of fabrication tools with only hardware-specific modifications.

These directions illustrate the broader potential of the system, not only as a technical exploration but as a flexible platform for low-cost, vision-guided fabrication tools.

5.4 Conclusion

This thesis explored a vision-based localization system for a handheld CNC tool using fixed ArUco markers. By leveraging detected marker positions, the system was able to generate a map of a 1 m *by* 1 m workspace, then it was able to locate itself to the generated map and readjust the bit to the required position, even under real-time constraints. This resulted in a drawn out figure with relatively good accuracy. The two-phase approach—first building a global marker map, then performing continuous pose tracking—proved effective in both simulation and physical validation.

Key experiments investigated the influence of marker layout, density, and mapping quality on localization accuracy. Results showed that although dense, structured grids offered reliable performance, medium-density random layouts performed surprisingly well and even outperformed other configurations in some cases. This highlights a trade-off between marker density and setup effort, and suggests that some degree of randomness may enhance robustness due to greater viewpoint diversity.

Physical validation using both a webcam and RealSense camera confirmed that the system performs reliably under real-world conditions, though quality varied depending on camera properties and map fidelity. The results also confirmed that the system is highly dependent on marker visibility and suffers when markers are occluded or out of view, especially in sparse configurations.

While the current implementation is limited by factors such as camera quality, mounting rigidity, and stepper resolution, it provides a strong foundation for further development. Future improvements could include incorporating SLAM techniques, adaptive marker layouts, or sensor fusion to improve robustness and scalability.

In summary, this work demonstrates that accurate, marker-based localization for handheld tools is feasible using relatively simple components, provided that marker placement and system calibration are carefully managed. The findings lay the groundwork for practical implementations in fabrication tools and similar applications.

Bibliography

[Aga+09] Sameer Agarwal et al. "Building Rome in a day". In: *2009 IEEE 12th International Conference on Computer Vision.* 2009, pp. 72–79. DOI: 10.1109/ICCV.2009.5459148.

[Bam25] Bambu Lab. *Bambu Lab Vision System.* Accessed: 2025-05-31. 2025. URL: https://eu.store.bambulab.com/nl/products/vision-encoder?srsltid=AfmBOop5u1XgqwCJVv2LY3G-CU3Xg7CuawGHOpERuGuU1xZPQAYUop-7.

[Cam+21] Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890. DOI: 10.1109/TRO.2021.3075644.

[FG87] Wolfgang Förstner and Eberhard Gülch. "A fast operator for detection and precise location of distinct points, corners and centres of circular features". In: *Proc. ISPRS Intercommission Workshop* (1987), pp. 281–305.

[Gar+14] S. Garrido-Jurado et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0031320314000235.

[Gar+24] Pablo García-Ruiz et al. "Large-Scale Indoor Camera Positioning Using Fiducial Markers". In: *Sensors* 24.13 (2024). ISSN: 1424-8220. DOI: 10.3390/s24134303. URL: https://www.mdpi.com/1424-8220/24/13/4303.

[Gar+25] Pablo García-Ruiz et al. "Sparse Indoor Camera Positioning with Fiducial Markers". In: *Applied Sciences* 15.4 (2025). ISSN: 2076-3417. DOI: 10.3390/app15041855. URL: https://www.mdpi.com/2076-3417/15/4/1855.

[Jor+14] Lode Jorissen et al. "Robust Global Tracking Using a Seamless Structured Pattern of Dots". In: *Augmented and Virtual Reality.* Ed. by Lucio Tommaso De Paolis and Antonio Mongelli. Cham: Springer International Publishing, 2014, pp. 210–231. ISBN: 978-3-319-13969-2.

[Ked+21] Oguz Kedilioglu et al. "ArUcoE: Enhanced ArUco Marker". In: *2021 21st International Conference on Control, Automation and Systems (ICCAS).* 2021, pp. 878–881. DOI: 10.23919/ICCAS52745.2021.9650050.

[Kog] Dima Kogan. *mrcal.* http://mrcal.secretsauce.net.

[Li+24] Zhuoran Li et al. "An Optimization on 2D-SLAM Map Construction Algorithm Based on LiDAR". English. In: *Journal of Intelligent & Robotic Systems* 110.4 (Sept. 2024). Publisher Copyright: © The Author(s) 2024. ISSN: 1573-0409. DOI: 10.1007/s10846-024-02123-1.

[MM21] Alexey Merzlyakov and Steve Macenski. "A Comparison of Modern General-Purpose Visual SLAM Approaches". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2021, pp. 9190–9197. DOI: 10.1109/IROS51168.2021.9636615.

[MMT15] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671.

[MT17] Raúl Mur-Artal and Juan D. Tardós. “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”. In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103.

[Ols11] Edwin Olson. “AprilTag: A robust and flexible visual fiducial system”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[OT21] Milan Ondrašovič and Peter Tarábek. “Homography Ranking Based on Multiple Groups of Point Correspondences”. In: *Sensors* 21.17 (2021). ISSN: 1424-8220. DOI: 10.3390/s21175752. URL: https://www.mdpi.com/1424-8220/21/17/5752.

[PH19] Alwin Poulose and Dong Seog Han. “Hybrid Indoor Localization Using IMU Sensors and Smartphone Camera”. In: *Sensors* 19.23 (2019). ISSN: 1424-8220. DOI: 10.3390/s19235084. URL: https://www.mdpi.com/1424-8220/19/23/5084.

[RB05] Søren Riisgaard and Morten Rufus Blas. *SLAM for Dummies: A Tutorial Approach to Simultaneous Localization and Mapping.* Tech. rep. 2005. URL: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf.

[RMD12] Alec Rivers, Ilan E. Moyer, and Frédo Durand. “Position-correcting tools for 2D digital fabrication”. In: *ACM Trans. Graph.* 31.4 (July 2012). ISSN: 0730-0301. DOI: 10.1145/2185520.2185584. URL: https://doi.org/10.1145/2185520.2185584.

[Tri+00] Bill Triggs et al. “Bundle adjustment—a modern synthesis”. In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings.* Springer. 2000, pp. 298–372.

[WO16] John Wang and Edwin Olson. “AprilTag 2: Efficient and robust fiducial detection”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2016, pp. 4193–4198. DOI: 10.1109/IROS.2016.7759617.

[Yin+24] Yuanhao Yin et al. “Vision-based autonomous robots calibration for large-size workspace using ArUco map and single camera systems”. In: *Precision Engineering* 90 (2024), pp. 191–204. ISSN: 0141-6359. DOI: https://doi.org/10.1016/j.precisioneng.2024.08.010. URL: https://www.sciencedirect.com/science/article/pii/S0141635924001855.

[Zha+17] Chunhui Zhao et al. “Pose estimation for multi-camera systems”. In: *2017 IEEE International Conference on Unmanned Systems (ICUS).* 2017, pp. 533–538. DOI: 10.1109/ICUS.2017.8278403.

[Zha00] Z. Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.