Monash University

Faculty of Information Technology

FIT5128 Minor Thesis

Fusion of Infrastructure-Based 3D Pose Estimation and Mobile Visual-Inertial Odometry for Drift-Resilient Indoor Augmented Reality

---

Author:
William Tioe

Student ID:
30153808

Supervisors:
Prof. Le Hai Vu
Jiazhou 'Joe' Liu

Word count (Part 2, excl. references):
7399

Semester 1, May 2026

MONASH University

Contents

Acknowledgement **3**

Part 1: General Literature Review **4**

1 Introduction **1**

2 Substantive Literature Review **3**
2.1 Indoor Navigation 3
2.1.1 Device-Only Approaches 3
2.1.2 Hybrid Approaches 4
2.2 Markerless Approaches 5
2.2.1 Location-bound AR 5
2.2.2 Localisation and Tracking 6
2.3 Infrastructure-Assisted Localisation 9
2.3.1 Wireless Signal-Based Approaches 9
2.3.2 Vision-Based Systems 11
2.3.3 Hybrid Wireless Vision Systems 12

3 Summary of the State of the Art **13**

4 Research Project Plan **15**
4.1 Research Question and Aims 15
4.2 Research Design and Method 16
4.2.1 Research Design 16
4.2.2 Research Method 17
4.3 Data Collection 18
4.4 Experimental Goals 19
4.5 Ethics and Data Privacy 20

5 Conclusion **21**

References (Literature Review) **23**

Part 2: The Research Paper **28**

Abstract **28**

1 Introduction **29**

2 Background **30**
2.1 The Mechanics and Failure Modes of Mobile Visual-Inertial Odometry . . . 30
2.2 The Computational Overhead of Map-Based Mitigation 31
2.3 Infrastructure-Assisted Tracking and the Checkpoint Paradigm 32

3 Methodology and System Architecture **33**
3.1 System Architecture . . . 33
3.1.1 The Mobile Client . . . 33
3.1.2 The Infrastructure, Edge Nodes, and Central Hub . . . 34
3.1.3 Network and Communication Pipeline . . . 34
3.2 The Checkpoint Correction Mechanism . . . 35
3.2.1 The Coordinate Pipeline . . . 35
3.2.2 Pedestrian Detection and Target Selection . . . 36
3.2.3 The Inverse Translational Logic . . . 38
3.2.4 Preservation of the VIO Pipeline . . . 38
3.3 Ethical Considerations and Data Privacy . . . 39

4 Experimental Setup **39**
4.1 The Testing Environment . . . 40
4.2 Apparatus and Hardware Deployment . . . 40
4.3 Experimental Tasks and Route Definitions . . . 41
4.3.1 The Navigation Task . . . 41
4.3.2 Definition of Navigation Routes . . . 43
4.4 Experimental Procedure . . . 43
4.4.1 Drift and Failure Trials (Paths 1–3) . . . 44
4.4.2 Trajectory and Hardware Trials (Path 4) . . . 44
4.5 Evaluation Metrics . . . 44
4.5.1 Measurement 1: Final Drift Error (FDE) . . . 45
4.5.2 Measurement 2: Critical Spatial Failure Rate . . . 45
4.5.3 Measurement 3: Trajectory Deviation . . . 46
4.5.4 Measurement 4: Hardware Resource Consumption . . . 46
4.5.5 Statistical Analysis Methodology . . . 46

5 Results **48**
5.1 Final Drift Error (FDE) and Variance . . . 49
5.2 Trajectory Deviation and Critical Failures . . . 50
5.3 Hardware Resource Overhead . . . 52

6 Discussion **53**
6.1 Contextualising Baseline FDE and Survivorship Bias 53
6.2 Spatial Stabilisation and Catastrophic Drift Recovery 54
6.3 Architectural Trade-offs: The Networking Tax and Battery Paradox 55
6.4 Addressing Architectural Scalability 56

7 Conclusion **56**

References **58**

Part 3: Appendices **61**

A Empirical Justification for Torso Tracking Target Selection **61**

B Mathematical Formulation of the Central Hub Tracking Logic **63**

C Infrastructure Hardware and Extrinsic Calibration Methodology **65**

D Visual Documentation of Critical Spatial Failures **67**

E Outlier Analysis and Data Pre-processing Impact **70**

F Expanded Continuous Trajectory Deviation Analysis **71**

Acknowledgement

I would like to express my sincere gratitude to my supervisors, Prof. Le Hai Vu and Jiazhou 'Joe' Liu, for their invaluable guidance, support, and expertise throughout the development of this research. I am also immensely grateful to Matthew Willaton and Yide Tao for their significant contributions and insightful feedback. Furthermore, I would like to extend my thanks to the MCAV bachelor students, whose practical assistance and dedication were instrumental in the successful execution of this project.

Part 1: General Literature Review

Monash University

Faculty of Information Technology

Master of Artificial Intelligence

Literature Review

Fusion of Infrastructure-Based 3D Scene Reconstructions and Mobile Visual-Inertial Odometry for Drift-Resilient Indoor Augmented Reality

---

Author:
William Tioe

Student ID:
30153808

Supervisors:
Prof. Le Hai Vu
Jiazhou 'Joe' Liu

Semester 2, 2025

MONASH University

1 Introduction

Augmented Reality (AR) is an emerging technology that overlays digital information onto the physical world, allowing users to interact with aligned virtual content through smartphones, head-mounted displays, or other devices. In the context of navigation, AR provides intuitive guidance by embedding directions directly within a user's visual field, thereby reducing reliance on abstract maps and enhancing spatial awareness [1]. In outdoor environments, AR navigation benefits from well-established positioning systems, such as *Global Positioning System* (**GPS**). However, extending these capabilities indoors remains challenging, as GPS offers little to no reliable performance for accurate localisation in enclosed environments [1, 2].

Consequently, the development of indoor AR navigation relies on alternative approaches to localisation, which must address both the absence of GPS and the inherent limitations of smartphone-based localisation. While visual-inertial odometry (**VIO**) provides device-centred pose estimates, it suffers from drift accumulation and latency, especially in extended indoor use [3]. These issues hinder the reliability of AR navigation apps that depend on accurate, real-time alignment with the physical environment. Recent advances in infrastructure sensing offer opportunities to augment on-device localisation. In particular, fixed cameras capable of reconstructing 3D scenes and aligning them to abstract floor models present a promising pathway for achieving robust, drift-resilient, markerless indoor localisation.

This literature review aims to examine the current state of research relevant to indoor augmented reality navigation. Specifically, it will explore three key areas: existing implementations and methods for indoor navigation in the absence of GPS, markerless AR and infrastructure-assisted localisation approaches leveraging fixed sensors and 3D reconstructions. By evaluating the strengths, limitations, and gaps within these domains, the review seeks to establish a foundation for understanding how infrastructure-based sensing, specifically fixed cameras, can be integrated with smartphone localisation to support and enable drift-resilient, low-latency indoor AR.

The first section explores indoor navigation methods more broadly, with emphasis on techniques that operate effectively in GPS-denied settings. This includes device-only approaches, hybrid methods, and systems that integrate wireless or visual signals to support navigation. The second section focuses on markerless AR (**MAR**), highlighting techniques that estimate device pose without artificial markers. Here, the discussion considers visual-inertial

odometry and its extensions, addressing trade-offs between real-time performance and long-term drift. The third section turns to infrastructure-assisted localisation, where fixed cameras and wireless anchors augment device-based tracking. This section evaluates how external references improve robustness, while also noting challenges of deployment and scalability. Taken together, these sections situate the proposed research within the wider landscape of indoor AR localisation.

Following this literature review, the research project will outline a structured approach to investigating infrastructure-assisted localisation for indoor AR. The aim is to enhance smartphone-based tracking by fusing phone VIO with infrastructure camera observations, using an indoor location's existing 3D floor model as a shared reference frame. This approach will explore markerless feature mapping between the phone and fixed cameras to reduce drift and latency. The research will compare a baseline system relying solely on the 3D floor model with a hybrid system incorporating infrastructure cameras as a periodic "checkpoint" to correct accumulated drift. Experiments will assess scenarios to test tracking accuracy. The outcome will demonstrate whether infrastructure cameras can act as a supporting component that improves the robustness of indoor AR.

2 Substantive Literature Review

2.1 Indoor Navigation

2.1.1 Device-Only Approaches

Indoor navigation encompasses a range of methods designed to enable reliable positioning and movement guidance in enclosed environments where traditional GPS signals degrade or fail. Such systems integrate diverse sensing modalities and algorithms to address challenges such as multipath interference, occlusions and complex floor layouts. Within this domain, device-only approaches represent a key class of solutions, leveraging solely on-board sensors and computation to achieve localisation without reliance on external infrastructure.

One such example is NavARNode by Putra et al. [1], where the system employs ARCore with user-placed nodes during setup and utilises A* pathfinding to navigate within the building. This setup provides flexible runtime mapping as users place nodes and supports efficient A* pathfinding. Lu et al.'s [4] system works similarly but utilises image-based feature tracking for periodic calibration, as well as using geofencing anchors for location checkpoints instead of QR codes. This system is stated to have achieved sub-meter accuracy with calibration ($\leq 0.3m$).

In comparison, some implementations use a 3D map instead to assist with localisation. An example is EINS_AR by Jeny et al. [5], which utilises the Immersal SDK and Unity NavMesh to create a map, enabling dynamic path updates and occlusion handling. Shewail et al. [6] also achieves the same, utilising computer vision techniques such as ORB for feature mapping, Brute Force Match, K-Nearest Neighbours matching, and A* to achieve centimetre-level accuracy (7-10 cm) with AR arrows. Dong et al.'s ViNav [7] works similarly by building sensor-enriched 3D models via crowd-sourced photos and Structure-from-Motion mapping, enabling infrastructure-free multi-floor navigation with less than 1 m error. Hořejší et al. [8] implemented a similar system for warehouse navigation with a 3D model created in Unity and ARCore. This system calibrates itself via QR codes to initialise and guides workers within the warehouse with navigation lines, while achieving sub-meter accuracy (0.48 m).

However, these implementations have drawbacks. For Putra et al.'s [1] implementation to initialise navigation, scanning QR codes is still necessary

to provide the start location, and this method still requires extensive physical work, especially during the scanning and mapping phase. Lu et al.'s implementation [4] has a high dependence on visible reference markers to recalibrate itself, establishing a strict line of sight requirement as well as a large number of markers required. Performance for this system also varies due to lighting and the user's walking speed. Jeny et al.'s EINS_AR [5] is also computationally demanding and is sensitive to environmental conditions such as lighting. Shewail et al.'s system [6] also fails to work well under poor lighting and requires careful scanning of the environment for the model. ViNav by Dong et al. [7] is heavily dependent on dense image datasets and also suffers from lighting issues, as well as Hořejší et al.'s system [8], alongside its issues of battery drain and user distractions.

In short, the device-only indoor navigation systems discussed all have detrimental drawbacks for device-only indoor navigation, such as poor lighting, some kind of strenuous setup effort involving manual mapping, and scanning QR codes for positional initialisation. These issues even extend to requiring an extensive and comprehensive dataset or heavy computation. While several optimisations and advancements can improve device-only approaches, hybrid systems represent one of the key solutions to overcoming their inherent limitations. By combining on-device sensing with external infrastructure, these approaches provide a pathway to enhanced robustness, accuracy, and usability, particularly in environments where device-only methods continue to face persistent challenges.

2.1.2 Hybrid Approaches

Hybrid indoor navigation systems take various forms, ranging from wireless anchor-based methods such as UWB or Bluetooth beacons, physical markers such as QR codes, or LiDAR mapping. One such system is a campus navigation app designed by Patel et al. [2], where indoor navigation relies on QR codes and Unity NavMesh, while outdoor routes use Mapbox for voice-guided directions. This seamless indoor-outdoor integration enhances versatility for users. A similar system by Marian-Vladut et al. [9] achieves the same by relying on SLAM (Simultaneous Localisation And Mapping), VPS (Visual Positioning Systems), and QR codes for initialisation, navigation guidance and drift correction. This setup allows the system to be lightweight without relying on external infrastructure or even 3D models, as the system relies on QR codes to approximate itself. Alluhaidan et al.'s ESINS_AR [10] employs

a grid-based path-finding model with QR code initialisation and recalibration and an enhanced A* algorithm, which proved to be fast and accurate. Parab et al. built a LiDAR-enhanced AR system for a university campus [11], combining dense 3D mapping from LiDAR with QR anchors and SLAM for tracking. This approach ensures highly accurate navigation for users on campus. Finally, LEADOW, a system developed by Odeh et al. [12], utilises BLE beacons for localisation and Dijkstra's algorithm for path planning to provide step-by-step audio instructions for the visually impaired. This implementation allows for accurate tracking of the user, which is crucial as the target user-base has visual impairments, and is also cost-effective as BLE beacons are relatively inexpensive.

Nevertheless, the strengths of hybrid approaches come with trade-offs, particularly in terms so setup complexity, cost and scalability. Three systems discussed so far (Patel et al. [2], Marian-Vladut et al. [9], Alluhaidan et al. [10]) have a heavy reliance on consistent QR deployment and scanning, further substantiating the extensive physical effort drawback mentioned by Putra et al. [1]. Parab et al.'s implementation [11] also has issues regarding its cost of expensive mapping equipment as well as high processing demands. Last but not least, LEADOW by Odeh et al. [12] requires a dense beacon infrastructure, overturning the cost-effective benefits of BLE beacons, and does not detect physical obstacles, which is a crucial requirement for visually impaired individuals. In summary, a consistent issue amongst the hybrid approaches discussed is the reliance on QR codes or markers for calibration, while certain systems require heavy computation requirements locally on-device or a dense beacon infrastructure. Thus, recent research has increasingly explored markerless solutions as a promising alternative, while infrastructure-assisted systems remain viable in contexts where their deployment is justified. However, both approaches are shaped by trade-offs in scalability, deployment effort, and cost, which continue to influence their suitability across different environments.

2.2 Markerless Approaches

2.2.1 Location-bound AR

Within markerless approaches, one notable subcategory is location-based augmented reality. Location-based Augmented Reality (LAR) is a broad category with the primary goal of anchoring AR content to physical location

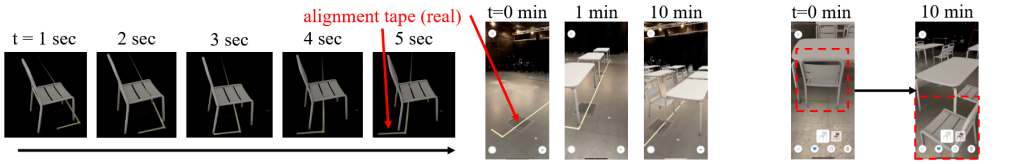
coordinates. This is commonly done using traditional sensor approaches, such as GPS and IMU, which are commonly available in smartphones to determine device locations and orientations to achieve immersive LAR experiences [13]. Yet, GPS has its shortcomings outdoors as well, having limited accuracy [13] and a range between 1 m and 30 m accuracy depending on environmental conditions [14].

However, there are a few implementations that solve this issue, a notable one by Brata et al. [13] being the fusion of Visual Simultaneous Localisation and Mapping (VSLAM) and the extensive visual reference database of Google Street View. Most of the existing VSLAM-based methods are implemented for indoor scenarios. Yet, it alone has several limitations, such as tracking loss and accumulating errors for building and localising trajectories. Therefore, Brata et al. propose a solution that integrates VSLAM and Google Street View with their LAR application, where VSLAM refines the user's localisation and Street View assists by aligning the device's camera view to estimate position and orientation. This results in an overall Mean Error of 0.796 m and Root Mean Square Error of 0.806 m as compared to conventional LAR's Mean Error of 7.328 and 7.379.

Although this approach is less directly applicable to indoor navigation, the underlying concept of leveraging an extensive database or model to align a device's localisation with the environment remains highly relevant. Nevertheless, such methods pose particular challenges indoors. Unlike outdoor settings, where resources like Google Street View provide vast image databases, indoor environments, especially private or restricted buildings, lack comparable coverage. This gap underscores the need for either comprehensive collections of indoor imagery or the generation of detailed indoor maps to enable accurate localisation.

2.2.2 Localisation and Tracking

Localisation refers to estimating a device's position and orientation using on-board sensors (e.g., GPS or IMU) or by referencing environmental landmarks [3]. Tracking, by contrast, is typically performed through visual-inertial odometry (VIO), which fuses data from vision-based sensors (such as cameras or LiDAR) with inertial measurements (IMU) to provide continuous updates of user movement and orientation [3]. Together, localisation and tracking are essential for enabling immersive, markerless AR, where accurate positioning must be achieved without physical markers or external beacons.



(a) Short term drift of virtual elements (b) Global drift of elements (c) Relative mis-alignment of elements

Figure 1: Instances of AR failure: (a) Over a period of a few frames, virtual objects drift in the virtual world due to tracking failure. (b, c) Virtual objects (tables and chairs) shift globally due to localisation failure. In (b), the table is misaligned to the tape, and in (c), the highlighted chair is misaligned to the table. [3]

Still, device-only MAR faces fundamental challenges. Yamaguchi et al. [3] conducted extensive experiments on smartphone-based AR localisation and tracking, identifying dynamic lighting conditions, low visual-feature density, and extreme movement speeds as critical limitations. In particular, IMU-based localisation accumulates drift at speeds above 2 m/s and even below 0.2 m/s. This means drift is unavoidable in most common scenarios, even when the device is stationary, as shown in Figure 1. Thus, highlighting the need for more robust localisation through robust algorithms or complementary sensing methods.

Several works aim to mitigate drift and improve accuracy without external infrastructure. Swamy et al. [15] propose an online method that combines SIFT feature extraction, FLANN-KNN matching, and ARCore depth maps for 3D positioning, achieving 2.5-3.5 cm accuracy without markers. They further introduced ADAPT, an adaptive SLAM system that switches between RGB-D SLAM when near AR content and Monocular SLAM when farther away. This hybrid switching balances resource efficiency with accuracy, outperforming Monocular SLAM when pose estimation uses the same number of frames.

Building on learning-based approaches, Liu et al. [16] presented MobileARLoc, a fully on-device framework that integrates VIO with Absolute Pose Regressors (APR). VIO validates APR predictions: reliable APR outputs correct drift, while unreliable ones are replaced by transformed VIO poses. The framework improved positional accuracy by up to 50% and rotational accuracy by 55% compared with PoseNet and MS-Transformer, particularly in complex indoor spaces. This demonstrates the potential of combining traditional tracking with neural pose regressors to enhance device-only

localisation.

Deep learning has also been applied to multi-floor environments. Kim and Shin [17] present a deep learning–based multi-floor indoor localisation system that fuses smartphone IMU, magnetometer, and barometer data to overcome drift and initialisation challenges in traditional PDR. Their method integrates magnetic field sequences with Long Short-Term Memory (LSTM) for floor detection, KNN matching for 2D positioning, and Seq2Seq modelling of barometer data for floor transition tracking. Correction nodes recalibrate vertical drift. This approach achieved 0.64 m RMSE and 99.8% reliability within 2 m, significantly outperforming conventional Pedestrian Dead Reckoning (PDR). However, it remains sensitive to device orientation and user pose variations.

In vehicle contexts, Shin et al. [18] introduce a deep learning-based framework for speed estimation in GNSS-denied environments. Using only smartphone accelerometer and gyroscope data, their stacked LSTM with attention mechanism achieved 0.38 m/s RMSE, outperforming DNN and Kalman filter baselines. While it does not provide full position updates, the system delivers reliable velocity inputs that are critical for dead reckoning pipelines in indoor or underground navigation. The same idea can potentially benefit smartphones, where accurate speed estimates help reduce drift in VIO and PDR, making localisation more stable in challenging environments.

Having reviewed device-only methods, we now turn to map-based and hybrid approaches, which integrate pre-constructed models or collaborative systems to further enhance localisation robustness. Messi et al. [19] developed a markerless, infrastructure-free localisation system for 6-DoF AR in GPS-denied indoor spaces. Their system constructs a Structure-from-Motion 3D reference map and uses the Hierarchical Feature Network (HF-Net) to match device query images. Position and orientation are then estimated using the Perspective-n-Point algorithm, supported by a Building Information Model. Reported accuracy reached 4-9 cm in position and 0.3-1.3° in orientation, validating the value of 3D reference maps. However, their evaluation was restricted to position and yaw, leaving questions around multi-floor and long-term drift.

Extending this line, Li et al. [20] introduced ColSLAM, a collaborative SLAM system for mobile phones that enables multi-user AR and large-scale mapping. Contributions include map-caching with asynchronous optimisation, point-line VIO with NetVLAD-based loop detection, and a Perspective-n-Point-and-Line method for pose refinement. ColSLAM achieved 2-7 cm av-

erage trajectory errors and scaled to more than five users while maintaining under 30 ms latency for local corrections. This demonstrates how collaborative mapping and distributed optimisation can reduce drift while supporting scalability.

Hybrid designs have also proven effective for drift suppression. Kuang et al. [21] propose PVM, a smartphone-based system that integrates pedestrian dead reckoning, image-based relocalisation, and map-constrained particle filtering. By combining inertial step tracking with visual corrections enhanced by Hidden Markov Models, the system reached 0.44 m RMSE in indoor environments while operating in real time on mobile hardware. While effective under occlusion and low-texture conditions, PVM remains limited by dependence on pre-constructed maps, sensitivity to initialisation, and incomplete multi-floor support.

Complementary to map-based systems, Higa et al. [22] demonstrated how environment-specific visual features can serve as localisation cues similar to Lu et al.'s implementation [4]. In their study of a tourist aquarium, smartphone images of fish tanks were classified with deep learning (ResNet18, MaxViT, EfficientNet, MobileNetV3, among others) to infer a visitor's location by region. While precise localisation was not achieved, and retraining was required when environments changed, this work underscores the role of domain-specific features in localisation, paralleling how 3D models could enrich AR navigation.

In summary, the reviewed literature illustrates both the progress and persistent limitations of current approaches. Device-only systems have become more accurate with adaptive SLAM and deep learning fusion, yet drift and sensitivity to orientation remain unresolved. 3D map-based and hybrid methods achieve centimetre-level precision, but at the cost of requiring pre-constructed maps, retraining, or infrastructure support. Multi-user and collaborative systems such as ColSLAM highlight promising scalability, but computational trade-offs and latency constraints remain barriers.

2.3 Infrastructure-Assisted Localisation

2.3.1 Wireless Signal-Based Approaches

Infrastructure-assisted localisation leverages fixed environmental sensors, such as cameras or wireless anchors, to augment device-based tracking. Unlike purely markerless methods, these systems use external references to reduce

drift and improve accuracy, enabling more reliable indoor AR positioning. In this section, we will cover wireless signal-based systems. Wireless methods exploit fixed transmitters or anchors to estimate device position. These systems are common due to the ubiquity of wireless infrastructure, though accuracy and stability remain challenges.

The BiodivAR tool, developed by Mercier et al. [14], is a location-based AR authoring system for teachers, students, and citizen scientists. It anchors text, images, 3D models and sounds to outdoor points of interest for biodiversity education and field trips. A key limitation is GPS inaccuracy, typically 1-30 m on mobile devices, which causes unstable anchoring. To address this, the authors proposed Real-Time Kinematic (RTK) positioning with an Ardusimple RTK surveyor module, capable of centimetre-level accuracy. While no numerical results were reported, RTK is expected to improve localisation by providing a fixed reference station. However, it is costly and less effective indoors. Even so, it shows how infrastructure can enhance localisation when context-appropriate.

Gong et al. [23] evaluated Bluetooth Low Energy (BLE) beacon tracking for construction workers. The position is calculated via trilateration, requiring signals from at least three sensors. Two testbeds, an indoor room and a corridor, were used to study challenges such as RSSI variability, multipath fading, beacon placement, and geometry. Results showed that a calibrated model achieved localisation within 1 m. In contrast, a pre-calibrated manufacturer model had an error of up to 4 m. Key deployment strategies included placing beacons at edges or corners, as central placements were less reliable, and accuracy declined beyond 1-6 m. BLE offers potential for indoor localisation but requires large-scale deployment in big buildings, even if individual units are inexpensive.

Akram et al. [24] surveyed indoor localisation systems in Internet of Things (IoT) contexts, emphasising machine learning (ML) for improved accuracy, scalability, and robustness. They compared wireless technologies (Wi-Fi, ZigBee, RFID, Bluetooth, UWB) and signal metrics (RSSI, Angle-of-Arrival, Time-of-Arrival, Time-Difference-of-Arrival). ML methods can significantly reduce errors, be adapted to dynamic environments and enhance fingerprinting approaches as showcased by Figure 2.

ML particularly improved Wi-Fi and BLE accuracy, while raw accuracy is poor. UWB remained the most precise, but ML-enhanced fingerprinting enabled Wi-Fi and BLE to reach 1-2 m accuracy. This makes common wireless technologies viable for scalable IoT deployments and promising for AR

tracking.

Ref/year	Technology	Technique	Algorithm	ML	Accuracy
[56]/2016	WiFi	CSI	Fingerprinting	DL	1.08m 2.0134m
[57]/2017	WiFi	RSSI	Fingerprinting	SVM	97.31%
[58]/2017	WiFi	RSSI	Fingerprinting	SRL-KNN	0.66m
[59]/2018	WiFi	AoA	Triangulation and Trilateration	N/A	2.5 m
[60]/2018	Bluetooth	RSSI	Fingerprinting	PSO-ANN	1.61 m
[61]/2022	WiFi	RSSI	Fingerprinting	SPSO	2.0817 m
[53]/2022	WiFi	CSI	Fingerprinting	BPNN-AGA	4m
[62]/2017	RFID	RSSI	Triangulation	KNN	1.3 m
[63]/2016	UWB	ToA, AoA, AoD	Triangulation	N/A	0.42 m, 0.59 m, 1.22 m
[64]/2022	UWB and RFID	RSSI	trilateration	N/A	0.5 m for 2D 1 m for 3D
[65]/2020	Zigbee	RSSI	Fingerprinting	KNN SVM Logistic Regression	87.30% 84.92% 84.12%

Figure 2: Comparisons between different combinations of wireless technologies and ML techniques [24]

2.3.2 Vision-Based Systems

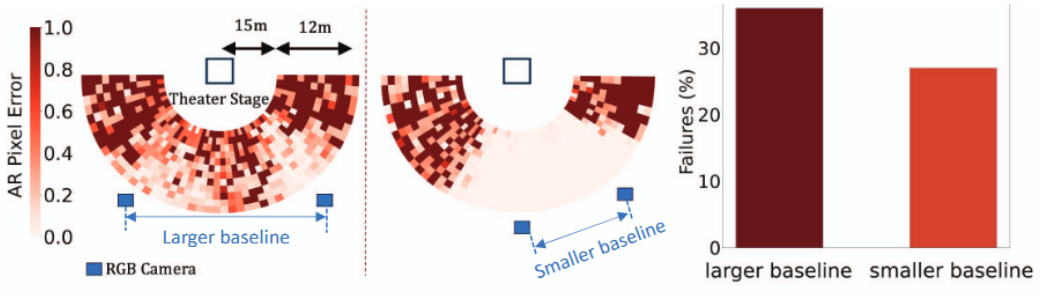
Vision-based localisation systems rely on cameras, either fixed or on-device, to estimate device pose. A notable system is StageAR, introduced by Jin et al. [25], which targets markerless localisation for mobile phones in live event venues such as theatres and concerts. Traditional AR approaches often fail in such settings due to dynamic lighting, moving sets, and the need to avoid intrusive fiducial markers.

StageAR addresses these challenges using sparse fixed infrastructure, such as cameras or LiDAR, to filter and update feature maps in real time. These maps are broadcast to audience devices, enabling instant-on 6-DoF localisation across large crowds. The system supports three configurations of increasing complexity: single-camera filtering of unreliable features, stereo camera depth estimation, and LiDAR-camera fusion. While the LiDAR approach achieved the highest accuracy, the stereo setup provides a useful reference for this review.

In controlled tests, the stereo system achieved centimetre-level accuracy under careful calibration, with an average baseline error of 2.5 cm. However, it was highly sensitive to misalignment: a 10 cm calibration error increased translation and rotation error more than sixfold. Stereo setups also showed

a 40% localisation failure rate under dynamic lighting, reflecting the limited robustness of two cameras alone. Time dilation further introduced drift, with errors rising from 3.5% at 2 s delay to nearly 10% at 4 s. Smaller baselines performed better than wider ones, as shown by Figure 3, highlighting the importance of deployment choices.

Despite these shortcomings, StageAR demonstrates how vision-based infrastructure can enable robust, markerless localisation by sharing feature maps between devices and fixed sensors. In indoor navigation contexts, where lighting is more stable, such methods offer a strong foundation for accurate and scalable localisation.

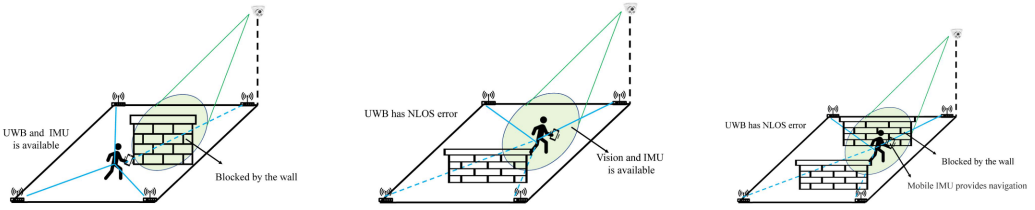


(b) Different configurations of stereo camera pair

Figure 3: Heat-map shows AR Pixel Error using two stereo cameras with different stereo baselines [25]

2.3.3 Hybrid Wireless Vision Systems

Hybrid systems integrate multiple modalities to overcome single-technology limitations. UVtrack, developed by Xu et al. [26], fuses UWB ranging, ambient camera vision, and IMU-based pedestrian dead reckoning (PDR) as shown by Figure 4. Inputs are combined using an adaptive weighted least squares method and a UWB-Vision Based Particle Filter, with geometric constraints preventing unrealistic movement. Across laboratory, office, and hall environments, UVtrack achieved 7 cm error, maintaining accuracy across phone placements (hand, pocket, clothing) and under occlusion (9 cm). Multi-user tests showed 97% success for two users and 90% for eight. Compared to the state-of-the-art, UVtrack improved accuracy by 55% over Wi-Fi-Vision-PDR fusion. Even when UWB or vision failed, fallback to PDR sustained accuracy within 20 cm over 6 m travel.



(a) UWB and IMU are available (b) Vision and IMU are available (c) Only IMU is available

Figure 4: UVtrack's dynamic adjustment in different scenarios [26]

Deshmukh et al. [27] reviews indoor positioning systems for mobile robotics, comparing non-RF, RF, and hybrid solutions by accuracy, latency, scalability, and robustness. LiDAR and Visible Light Communication achieved centimetre-level accuracy, UWB sub-20 cm, while Wi-Fi and Bluetooth were limited to meter-level accuracy. Hybrid systems, such as UWB combined with visual SLAM, reduced drift by over 50%. Systems with ¡100 ms were best suited to real-time navigation. Although robotics-focused, these findings extend to mobile devices, where sensor fusion similarly boosts localisation robustness.

Zhou et al. [28] propose a GNSS-aided indoor positioning framework that synchronises in-building systems (Wi-Fi, Bluetooth, UWB, cellular small cells) using GNSS assistance data to reduce clock offsets and improve ranging. Wi-Fi accuracy improved from 3-5 m to 1.5-2 m, and UWB achieved sub-0.5 m precision. The method improved robustness in multipath-rich environments and enabled smoother outdoor-indoor handovers. Advantages include scalability and cost-effectiveness, as it builds on existing infrastructure. However, it depends on reliable GNSS signals, which limits deep-indoor performance and raises privacy concerns. Overall, GNSS-aided synchronisation provides a practical bridge between outdoor GNSS and indoor wireless positioning.

3 Summary of the State of the Art

AR localisation has advanced across device-only, hybrid, and infrastructure-assisted methods, yet achieving drift-resilient, low-latency pose estimation remains unresolved. Device-only approaches rely on visual–inertial odometry (VIO), visual SLAM, or 3D map reconstructions. Systems such as NavARN-

ode [1] and ViNav [7] show that smartphones can support navigation with sub-meter accuracy through feature tracking, floor models, or crowd-sourced imagery. Adaptive frameworks like ADAPT [15] and learning-based regressors such as MobileARLoc [16] improve drift correction, while collaborative systems like ColSLAM [20] enhance scalability. Nonetheless, these methods remain limited by accumulated sensor error, with drift increasing over time and performance sensitive to lighting, feature density, and user motion.

Hybrid solutions introduce anchors such as QR codes, BLE beacons, or LiDAR mapping to improve robustness [2, 10, 11, 12]. For example, campus navigation systems combine QR-based initialisation with SLAM or grid-based pathfinding, while LiDAR-enhanced approaches achieve high precision for large indoor spaces. These methods reduce drift but rely on costly infrastructure, dense marker deployments, or intensive computation, limiting scalability.

Markerless AR approaches highlight parallel efforts to achieve infrastructure-free tracking. Location-bound AR fuses GPS/IMU with extensive databases such as Google Street View [13], while indoor methods combine VIO with neural pose regressors, adaptive SLAM, or map-based feature matching [16, 19, 22]. Reported accuracy ranges from centimetre-level pose estimates to multi-user collaborative mapping [20]. Yet persistent issues of drift, retraining, and limited indoor image databases restrict long-term reliability.

Infrastructure-assisted systems offer the most direct route to drift suppression. Wireless localisation using BLE, Wi-Fi, or UWB can achieve 0.5–2 m accuracy, with machine learning improving stability [23, 28, 24]. However, these require dense deployments and still lack the fine-grained pose estimation needed for AR alignment. Vision-based approaches, such as StageAR [25], deliver centimetre-level accuracy by broadcasting feature maps from fixed stereo cameras or LiDAR, though performance is highly sensitive to calibration, lighting, and baseline configuration. Hybrid wireless-vision systems like UVtrack [26] achieve strong robustness by fusing UWB, camera vision, and pedestrian dead reckoning, sustaining accuracy within centimetres even under occlusion. These results reinforce a recurring pattern: infrastructure improves accuracy and stability but at the cost of complexity and deployment effort.

Taken together, the literature reveals a clear gap. Device-only methods are accessible but lack long-term robustness. Infrastructure-assisted approaches improve precision but demand dense beacons, expensive sensors, or extensive calibration. While some studies examine infrastructure cameras,

there is limited exploration of how sparse cameras could be fused with device VIO as periodic checkpoints rather than continuous anchors. Likewise, few works consider using an abstract 3D floor model as a shared reference frame, leaving scope for further investigation.

This motivates the proposed research: to evaluate whether fixed infrastructure cameras, fused with device VIO and aligned through a 3D floor model, can support drift-resilient, low-latency indoor AR localisation. Rather than treating infrastructure cameras as primary anchors, the study will test their role as periodic checkpoints to correct drift. The intention is to balance the accessibility of device-only methods with the robustness of infrastructure-supported systems, contributing evidence on how such hybrid designs can enhance the scalability and reliability of markerless indoor AR.

4 Research Project Plan

4.1 Research Question and Aims

This research investigates whether fixed infrastructure cameras, when aligned with an abstract 3D floor model, can be fused with smartphone-based visual-inertial odometry (VIO) to improve drift resilience and reduce latency in indoor AR localisation. The guiding question is:

- How can 3D scene reconstructions from sparse infrastructure cameras, aligned with an abstract 3D floor model, be fused with smartphone VIO to enable lower-latency, drift-resilient, markerless pose estimation for indoor AR?

The study pursues three aims:

1. **Baseline evaluation**: establish the performance of an AR navigation system relying solely on smartphone VIO and a 3D floor model
2. **Fusion assessment**: examine how periodic "checkpoint" corrections from sparse infrastructure cameras affect drift, latency, and robustness in cluttered indoor environments.
3. **Practical feasibility**: evaluate whether a hybrid system can balance accessibility (device-only methods) with stability (infrastructure support), without the costs of dense markers or wireless networks.

4.2 Research Design and Method

4.2.1 Research Design

This project adopts a comparative experimental design, evaluating two systems: a baseline device-only AR navigation prototype and a hybrid system incorporating sparse infrastructure cameras. The baseline establishes how well smartphone VIO performs when aligned with a 3D floor model, while the hybrid system tests whether periodic corrections from infrastructure cameras can mitigate drift and reduce latency.

The research will progress through two milestones. The first milestone is an engineering phase, where the existing Monash campus navigation application (built on the Mapbox API) will be extended into an AR navigation prototype. This involves adapting the app to visualise navigation in AR and integrating 3D floor models, as it currently provides only 2D indoor maps. If full integration with Mapbox proves impractical, an alternative workflow, such as a standalone Unity prototype, will be considered. This milestone also evaluates a phone-only configuration, where the AR system relies solely on VIO and the 3D model, demonstrating feasibility without infrastructure support.

The second milestone is a research phase, where the AR prototype will be fused with the stereo camera infrastructure in the Monash Innovation Labs. These cameras will generate 3D reconstructions of the environment, providing periodic "checkpoint" corrections for the AR system. Here, infrastructure cameras are treated as supporting actors rather than primary localisation providers. They will be used selectively at crucial navigation points to minimise drift and assess the minimally required setup for robust indoor AR.

This phased design ensures a stepwise transition from engineering implementation to experimental evaluation, while keeping the scope achievable within the project timeline. Controlled experiments in the Monash Innovation Labs will provide a consistent, well-instrumented environment for measuring performance under varied conditions such as long trajectories, low-texture areas, and dynamic lighting.

This phased design ensures a progressive transition from system development to experimental evaluation, while keeping the scope achievable within the project timeline. Controlled experiments in the Monash Innovation Labs will provide a consistent, well-instrumented environment for measuring per-

formance under varied conditions such as long trajectories, low-texture areas, and dynamic lighting.

4.2.2 Research Method

The research method builds on the phased design in Section 4.2.1, combining engineering implementation with experimental evaluation. The approach integrates smartphone-based AR tracking with infrastructure camera reconstructions, aligned via a shared 3D floor model.

Phase 1 - AR Navigation Prototype (Baseline)

- **Implementation**: Adapt the Monash campus navigation app (Mapbox API) into an AR prototype that overlays routes in the user's field of view. This version will rely solely on smartphone VIO for localisation.
- **3D Floor Models**: Use the existing 3D model of the Monash Innovation Labs as the global reference frame for pose estimation. Since experiments are restricted to the lab, this model is sufficient without generating new large-scale maps.
- **Baseline testing**: Conduct navigation trials to measure drift, latency, and stability. These trials establish the reference for comparison with hybrid methods.

Phase 2 - Hybrid System with Infrastructure Integration

- **Camera Fusion**: Employ the Monash Innovation Labs' stereo camera network to generate 3D reconstructions. These will act as periodic checkpoints for pose correction, rather than continuous anchors. Checkpoints will be placed at key navigation points to assess both drift reduction and the minimally required infrastructure setup.
- **Fusion Pipeline**: Align device VIO estimates with infrastructure reconstructions through the shared 3D floor model. Corrections will be lightweight to reduce computation and latency.
- **Hybrid testing**: Repeat navigation trials from Phase 1 under identical conditions, measuring the degree to which checkpoints reduce drift and improve stability. Results will also be compared against reported benchmarks from existing device-only and hybrid systems in the literature, situating performance within the wider state of the art.

Evaluation Metrics The system will be compared using three key performance indicators:

1. **Drift**: mean positional error per trajectory length (cm per metre).
2. **Latency**: average delay between physical movement and AR visual alignment (ms).
3. **Robustness**: stability under varying conditions (e.g., lighting changes, occlusion, clutter).

Tools and Frameworks

- **ARCore/ARKit**: device-side VIO tracking.
- **Unity with AR Foundation**: AR navigation app development and cross-platform deployment.
- **OpenCV/SLAM libraries**: camera-based reconstruction, feature detection, pose alignment.
- **Ground truth validation**: fiducial markers, where feasible, used only for evaluation.

By structuring the method into baseline and hybrid phases, with defined metrics and tools, the research ensures a replicable and systematic comparison. Including comparisons with prior systems further clarifies the contribution of this work within the AR localisation landscape.

4.3 Data Collection

Data will be collected in the Monash Innovation Labs using the existing stereo camera infrastructure and the 3D floor model of the facility. Constraining experiments to this environment ensures conditions are consistent, replicable, and feasible within the project timeline.

For the baseline, data will be collected solely from the smartphone's onboard sensors, including the camera, accelerometer, and gyroscope. The AR prototype will rely on VIO and the 3D floor model to estimate pose. Logs will be recorded during repeated navigation trials covering varied paths such as straight corridors, turns, and loops. These datasets will quantify drift, latency, and stability.

For the hybrid system, the same trials will be conducted while enabling the stereo camera network. The cameras will generate periodic 3D reconstructions, which will be aligned with the floor model and fused with the device's VIO stream. Logs will capture device-side estimates, infrastructure corrections, and fusion outputs, enabling direct comparison with the baseline.

Where feasible, ground truth will be obtained using fiducial markers placed in the lab. These references will not be part of the operational pipeline but will benchmark positional error and drift. All measurements will be synchronised with device and infrastructure logs.
To ensure robustness, each experiment will be repeated under varied conditions:

- **Lighting** (bright vs dim)
- **Environmental dynamics** (static vs presence of moving people)
- **Trajectory length** (short vs extended)
- **Texture density** (feature-rich vs feature-poor)

All data will be securely stored on university-approved servers. Raw images from infrastructure cameras will be anonymised or converted into feature maps to minimise privacy risks. Logs will be timestamped and documented for reproducibility.

4.4 Experimental Goals

The experiments are designed to evaluate whether sparse infrastructure checkpoints can improve the drift resilience and latency of smartphone-based AR localisation. They are organised into three goals aligned with the research aims.
Goal 1 – Baseline Performance Characterisation

- Establish the performance of the AR prototype operating solely on smartphone VIO and the 3D floor model.
- Quantify drift, latency, and robustness under controlled conditions.
- Provide a dataset against which the hybrid configuration can be compared.

Goal 2 – Impact of Infrastructure Checkpoints

- Introduce sparse infrastructure cameras as periodic checkpoints in the navigation pipeline.
- Evaluate their effect on reducing drift and improving pose alignment.
- Determine whether checkpoint corrections lower or increase latency.
- Compare outcomes against the baseline and reported benchmarks to identify measurable improvements.

Goal 3 – Feasibility and Minimal Setup Analysis

- Investigate the role of infrastructure as a supporting actor rather than a primary localisation method.
- Test different checkpoint placements to assess the minimal infrastructure required for effective drift correction.
- Examine whether the hybrid system achieves a balance between accessibility and robustness.

4.5 Ethics and Data Privacy

This research involves data collection in the Monash Innovation Labs using smartphones and a network of stereo cameras. Ethical considerations focus on privacy protection, data security, and broader issues linked to AR and infrastructure-based localisation.

The stereo camera infrastructure is already governed by Monash University ethics protocols. As outlined in the explanatory statement [29], cameras use edge processing to reduce raw imagery to skeletal or feature-based representations, ensuring no identifiable personal information is stored. Other sensors, including LiDAR and people counters, capture only motion or occupancy data. Data is securely stored on Monash servers and anonymised for research use.

Recent studies emphasise privacy and ethics in positioning systems. Deshmukh et al. [27] note that IPS research must balance accuracy and scalability with privacy protection. Lynam et al. [30] highlight transparency in AR navigation, stressing that localisation pipelines may expose sensitive movement

data. Werner et al. [31] identify privacy, autonomy, fairness, and well-being as key ethical domains, warning that pervasive sensing may capture bystanders, creating surveillance risks if safeguards are not enforced.

For this project, smartphone sensor logs and infrastructure-based reconstructions will be used strictly for evaluating localisation performance. Measures include:

- **De-identification**: infrastructure data reduced to skeletal maps or point clouds; no facial recognition performed.
- **Controlled environment**: trials restricted to the Monash Innovation Labs under existing approvals.
- **Informed awareness**: signage will inform lab users when experiments are underway.
- **Data security**: logs stored on secure servers with access limited to the research team.

5 Conclusion

This review analysed device-only, hybrid, markerless, and infrastructure-assisted methods for indoor AR localisation. While progress has been significant, each approach remains limited in ways that restrict its long-term feasibility. Device-only methods such as VIO and SLAM are lightweight and accessible but consistently accumulate drift and are highly vulnerable to lighting conditions, feature sparsity, and rapid motion. Hybrid systems that employ QR codes, BLE, or LiDAR mitigate some of these issues, yet demand costly deployments, dense markers, or high computational resources, raising serious scalability concerns. Markerless AR, though promising for infrastructure-free tracking, suffers from instability over time, with line-of-sight requirements, machine learning retraining requirements and limited indoor imagery databases further undermining robustness. Infrastructure-assisted localisation provides the most reliable drift suppression, yet its reliance on dense beacons, precise calibration, or complex camera networks highlights major barriers to practical deployment.

Collectively, the literature demonstrates that current methods offer partial solutions but fall short of delivering scalable, drift-resilient indoor AR. In

particular, little attention has been given to how sparse infrastructure cameras could serve as occasional checkpoints to correct device drift. Similarly, few studies consider abstract 3D floor models as a unifying reference across devices and infrastructure.

This project will critically examine whether integrating VIO with sparse infrastructure corrections can balance accessibility with robustness. Through comparative evaluation of baseline and hybrid systems, it aims to provide evidence on the feasibility of achieving reliable, minimally intrusive AR localisation.

References (Literature Review)

[1] Bagas Samuel Christiananta Putra, I. Kadek Dendy Senapartha, Jyun-Cheng Wang, Matahari Bhakti Nendya, Dan Daniel Pandapotan, Felix Nathanael Tjahjono, and Halim Budi Santoso. Adaptive AR Navigation: Real-Time Mapping for Indoor Environment Using Node Placement and Marker Localization. *Information*, 16(6):478, June 2025. Publisher: Multidisciplinary Digital Publishing Institute.

[2] Riya Patel, Richa Patel, and Jaiprakash Narain Dwivedi. Campus Navigation and Augmented Reality Guided Mobile Application. In *2025 3rd International Conference on Communication, Security, and Artificial Intelligence (ICCSAI)*, volume 3, pages 1802–1807, April 2025.

[3] Shunpei Yamaguchi, Aditya Arun, Takuya Fujiwara, Misaki Sakuta, Ryotaro Hada, Takuya Fujihashi, Takashi Watanabe, Dinesh Bharadia, and Shunsuke Saruwatari. Experience: Practical Challenges for Indoor AR Applications. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1030–1044, Washington D.C. DC USA, December 2024. ACM.

[4] Min Lu, Masatoshi Arikawa, Kohei Oba, Keiichi Ishikawa, Yuhan Jin, Tomihiro Utsumi, and Ryo Sato. Indoor AR Navigation Framework Based on Geofencing and Image-Tracking with Accumulated Error Correction. *Applied Sciences*, 14(10):4262, May 2024.

[5] J. R. V. Jeny, B. Sai Mahitha, M. Shashank, and N. Vamsi Krishna. EINS_ar: Enhanced Indoor Navigation System using Augmented Reality with Dynamic Path Adjustement. In *2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, pages 416–422, June 2025.

[6] Ashraf Saad Shewail, Hala H. Zayed, and Neven A. M. Elsayed. Real-time indoor tracking for augmented reality using computer vision technique. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 13(2):1845, June 2024.

[7] Jiang Dong, Marius Noreikis, Yu Xiao, and Antti Ylä-Jääski. ViNav: A Vision-Based Indoor Navigation System for Smartphones. *IEEE Transactions on Mobile Computing*, 18(6):1461–1475, 2019. Publisher: Institute of Electrical and Electronics Engineers Inc.

[8] Petr Hořejší, Tomáš Macháč, and Michal Šimon. Reliability and Accuracy of Indoor Warehouse Navigation Using Augmented Reality. *IEEE Access*, 12:94506–94519, 2024.

[9] Toma Marian-Vladut, Turcu Corneliu Octavian, and Pascu Paul. Evaluating User Acceptance and Usability of AR-Based Indoor Navigation in a University Setting: An Empirical Study. *International Journal of Advanced Computer Science and Applications (ijacsa)*, 16(4), May 2025. Publisher: The Science and Information (SAI) Organization Limited.

[10] Ala Saleh Alluhaidan, Latifa Abdullah Almusfar, Ahmed Younes Shdefat, Ahmed Elsayed Mansour, Diaa Salama Abdelminaam, and Yasmin Alkady. From GPS to AR: Leveraging Augmented Reality and Grid-Based Systems for Improved Indoor Navigation. *IEEE Access*, 13:55211–55230, 2025.

[11] Devang Kishor Parab, Pranav Prasanna Deshpande, Raj Manoj Thakur, Vedant Atul Warke, and Shushma Khanvilkar. Indoor Navigation System using Augmented Reality. In *2024 International Conference on Inventive Computation Technologies (ICICT)*, pages 720–725, Lalitpur, Nepal, April 2024. IEEE.

[12] Suhail Odeh, Murad Al Rajab, Jannat Natsheh, Isra' Zahran, Khader Ballout, and Mahmoud Obaid. LEADOW: An Empowering Indoor Navigation System for Individuals with Visual Impairments Using Bluetooth Beacons and Audio Guidance. In A. Mirzazadeh, Zohreh Molamohamadi, Efran Babaee Tirkolaee, Gerhard-Wilhelm Weber, and Janny Leung, editors, *Optimization and Data Science in Industrial Engineering*, pages 52–66, Cham, 2025. Springer Nature Switzerland.

[13] Komang Candra Brata, Nobuo Funabiki, Yohanes Yohanie Fridelin Panduman, and Evianita Dewi Fajrianti. An Enhancement of Outdoor Location-Based Augmented Reality Anchor Precision through VSLAM and Google Street View. *Sensors*, 24(4):1161, January 2024. Publisher: Multidisciplinary Digital Publishing Institute.

[14] Julien Mercier, Nicolas Chabloz, Gregory Dozot, Olivier Ertz, Erwan Bocher, and Daniel Rappo. BiodivAR: A Cartographic Authoring Tool for the Visualization of Geolocated Media in Augmented Reality. *ISPRS International Journal of Geo-Information*, 12(2):61, February 2023. Publisher: Multidisciplinary Digital Publishing Institute.

[15] Shneka Muthu Kumara Swamy and Qi Han. Online Mitigation of Spatial Drift of Virtual Objects in Mobile Augmented Reality. In *Proceedings of the 2024 SIGCOMM Workshop on Emerging Multimedia Systems*, EMS '24, pages 33–38, New York, NY, USA, August 2024. Association for Computing Machinery.

[16] Changkun Liu, Yukun Zhao, and Tristan Braud. MobileARLoc: On-device Robust Absolute Localisation for Pervasive Markerless Mobile AR. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 544–549, March 2024. ISSN: 2766-8576.

[17] Jin-Woo Kim and Yoan Shin. Deep Learning-Based Multi-Floor Indoor Localization Using Smartphone IMU Sensors With 3D Location Initialization. *IEEE Access*, 13:101532–101544, 2025.

[18] Beomju Shin, Shiyi Li, and Boseong Kim. Deep Learning-Based Vehicle Speed Estimation Using Smartphone Sensors in GNSS-Denied Environment. *Applied Sciences*, 15(16):8824, August 2025.

[19] Leonardo Messi, Francesco Spegni, Massimo Vaccarini, Alessandra Corneli, and Leoanrdo Binni. Infrastructure-Free Localization System for Augmented Reality Registration in Indoor Environments: A First Accuracy Assessment. In *2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, pages 110–115, June 2024.

[20] Wanting Li, Yongcai Wang, Yongyu Guo, Shuo Wang, Yu Shao, Xuewei Bai, Xudong Cai, Qiang Ye, and Deying Li. ColSLAM: A Versatile Collaborative SLAM System for Mobile Phones Using Point-Line Features and Map Caching. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 9032–9041, New York, NY, USA, October 2023. Association for Computing Machinery.

[21] Yujin Kuang, Yafei Liu, Yuan Yang, Chenchen Zhou, and Xiaoguo Zhang. A Novel Indoor Positioning System Utilizing Image-Based Relocalization and Pedestrian Dead Reckoning. *IEEE Transactions on Instrumentation and Measurement*, 74:1–15, 2025.

[22] Gabriel Toshio Hirokawa Higa, Rodrigo Stuqui Monzani, Jorge Fernando da Silva Cecatto, Maria Fernanda Balestieri Mariano de Souza, Vanessa Aparecida de Moraes Weber, Hemerson Pistori, and Edson Takashi Matsubara. Smartphone region-wise image indoor localization using deep learning for indoor tourist attraction. *PLOS ONE*, 19(9):e0307569, September 2024. Publisher: Public Library of Science.

[23] Yue Gong, JoonOh Seo, Tae Wan Kim, Seungjun Ahn, and Yanfang Luo. Field validation of beacon-based indoor tracking and localization system for construction workers. *KSCE Journal of Civil Engineering*, 29(2):100017, February 2025.

[24] Rawaa Akram, Alireza Norouzi, Aseel H. Al-Nakkash, and Negar Majma. Machine learning for indoor localization based IoT: A review. *AIP Conference Proceedings*, 3211(1):030023, May 2025.

[25] Tao Jin, Shengxi Wu, Mallesham Dasari, Kittipat Apicharttrisorn, and Anthony Rowe. StageAR: Markerless Mobile Phone Localization for AR in Live Events. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 1000–1010, March 2024. ISSN: 2642-5254.

[26] Yi Xu, Zhigang Chen, Ming Zhao, Fengxiao Tang, Yangfan Li, Jiaqi Liu, and Nei Kato. UVtrack: Multi-Modal Indoor Seamless Localization Using Ultra-Wideband Communication and Vision Sensors. *IEEE Open Journal of the Computer Society*, 6:272–281, 2025.

[27] Rushikesh A. Deshmukh, Meghana A. Hasamnis, Madhusudan B. Kulkarni, and Manish Bhaiyya. Advancing indoor positioning systems: innovations, challenges, and applications in mobile robotics. *Robotica*, 43(7):2710–2750, July 2025.

[28] Shuya Zhou, Xinghe Chu, and Zhaoming Lu. Enhancing Indoor Positioning with GNSS-Aided In-Building Wireless Systems. *Electronics*, 14(10):2079, May 2025.

[29] Monash University. Explanatory statement: Smart manufacturing hub digital twin (smh-dt): Mobility project full deployment. Report 36215, Faculty of Engineering, Monash University, Clayton, VIC, Australia, 2025.

[30] Hudson Lynam, Sergiu Dascalu, and Eelke Folmer. Augmented Reality Navigation: A Survey. *International Journal of Human–Computer Interaction*, 41(16):10190–10206, August 2025. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2024.2431757.

[31] Leonardo Werner, Philip Brey, and Adam Henschke. Augmented reality and ethics: key issues. *Virtual Reality*, 29(3):122, July 2025.

Part 2: The Research Paper

Abstract

Standard mobile Augmented Reality (AR) frameworks, such as Google ARCore, rely fundamentally on Visual-Inertial Odometry (VIO) for indoor localisation. However, in GNSS-denied, building-scale environments lacking dense visual features, native monocular VIO inevitably succumbs to perceptual aliasing and accumulates catastrophic translational drift, rendering continuous pedestrian navigation unsafe. While existing literature often mitigates this drift via continuous on-device spatial mapping, this approach introduces severe computational and thermal bottlenecks that restrict AR accessibility to flagship hardware. To resolve this, this research proposes and empirically evaluates a novel edge-assisted spatial architecture. The system leverages a sparse infrastructure network of OAK-D stereo-vision cameras and Intel NUC edge nodes to perform lightweight YOLO pedestrian inference. By processing absolute spatial coordinates at the edge and transmitting them to the mobile client via an asynchronous MQTT payload, the architecture performs an inverse environmental translation, seamlessly correcting accumulated drift without disrupting the device's native VIO loop.

Experimental trials conducted across diverse spatial geometries demonstrated significant improvements in navigational safety and accuracy. On extended linear trajectories prone to severe perceptual aliasing, the edge-assisted architecture achieved an 80.5% reduction in continuous spatial error (measured via Dynamic Time Warping) and reduced critical, barrier-colliding navigational failures by 50%. Crucially, hardware profiling confirmed that by offloading the heavy visual inference, the mobile client maintained a stable baseline memory footprint (307 MB) and a consistent 30 FPS rendering target. However, the evaluation also identified a systemic "networking tax," in which continuous asynchronous MQTT communication increased frame-time volatility and per-run battery drain by 80%. Ultimately, this research demonstrates that sparse, infrastructure-assisted localisation successfully bridges the gap between local device limitations and reliable, building-scale indoor AR navigation, thereby sacrificing mobile power efficiency to ensure fundamental spatial safety.

1 Introduction

Augmented Reality (AR) is a technology that overlays digital information onto the physical world, allowing users to interact with virtual content anchored to their real-world environment via devices such as smartphones or head-mounted displays. When applied to navigation, AR fundamentally enhances spatial awareness by projecting intuitive directional cues directly into the user's field of view. While AR navigation systems operate with high reliability in outdoor environments, largely due to the robust positioning capabilities of the Global Positioning System (GPS), extending these systems indoors presents a significant technical hurdle. Because building structures severely attenuate satellite signals, GPS is rendered effectively obsolete for the precise localisation required to maintain accurate AR overlays in enclosed spaces [1, 2].

To compensate for the absence of GPS, existing indoor localisation systems have traditionally utilised various wireless modalities, such as Bluetooth Low Energy (BLE) beacons or Ultra-Wideband (UWB) ranging, to estimate a user's physical position [3, 4, 5]. While these technologies can significantly improve general indoor tracking, Augmented Reality navigation requires much more than simple coordinate positioning; it demands precise, centimetre-level 6-DoF (Degrees of Freedom) tracking to seamlessly anchor virtual elements to the physical world [6]. Modern mobile devices can maintain rotational accuracy relatively well; however, they suffer from severe drift over extended distances, causing virtual navigation paths to slowly diverge from physical corridors [7]. As a result, a common approach in contemporary AR research is to supplement inertial tracking with vision-based methods to help mitigate this accumulated error. [8, 9].

However, requiring continuous visual capture and on-device environmental mapping introduces profound deployment constraints. Relying solely on a smartphone to perform complex absolute localisation against dense image datasets [10], exhaustive 3D point clouds [2], or heavy 3D maps [11] is architecturally prohibitive for standard mobile devices. While some frameworks attempt to mitigate this overhead by dynamically switching tracking modes [12], fundamental hardware limitations persist, inevitably degrading tracking performance over extended distances [13, 14]. Without offloading this spatial processing, reliable AR navigation remains restricted to high-end flagship smartphones equipped with advanced neural processing units. This creates a significant accessibility barrier, effectively excluding users with mid-range or

older devices and undermining the viability of a universal public navigation system [15].

To address the dual challenges of long-distance translational drift and on-device computational bottlenecks, this research proposes a novel edge-assisted AR navigation architecture that leverages Internet of Things (IoT) concepts [14]. Rather than forcing the mobile device to perform continuous environmental mapping, this system integrates sparse, smart-infrastructure stereo cameras to serve as a supporting tracking network [16]. By utilising an abstract 3D floor model as a shared reference frame, these infrastructure cameras function as periodic "checkpoints" to calculate position and correct the smartphone's accumulated drift. Crucially, the heavy visual processing is offloaded to local IoT edge computing nodes. This design ensures privacy by processing visual data locally at the edge, without storing bystanders' identifying information, while significantly reducing the computational load on the user's phone to ensure low-latency performance.

Therefore, this minor thesis aims to detail the architectural exploration and comprehensively evaluate the feasibility and performance of an edge-assisted, infrastructure-supported indoor AR navigation system. Specifically, the research pursues three primary objectives:

1. To significantly mitigate long-distance AR navigation drift and eliminate critical spatial routing failures in complex indoor environments.

2. To design and implement a sparse, infrastructure-assisted checkpoint methodology that reliably corrects mobile tracking drift without relying on continuous, on-device spatial mapping.

3. To demonstrate that offloading visual processing to local IoT edge nodes successfully eliminates on-device computational bottlenecks, ensuring the navigation architecture remains accessible to standard mobile hardware.

2 Background

2.1 The Mechanics and Failure Modes of Mobile Visual-Inertial Odometry

Modern markerless AR frameworks fundamentally rely on VIO to maintain 6-DoF device localisation in enclosed, GNSS-denied environments. VIO sys-

tems continuously fuse visual feature tracking from the device's camera with high-frequency inertial data from the smartphone's IMU to provide odometry, i.e., estimates of the change in the user's position over time relative to a start point. While these pipelines provide highly responsive short-term tracking, even state-of-the-art, off-the-shelf proprietary VIO frameworks inevitably suffer from unavoidable accumulation of spatial error over time [17]. The literature broadly refers to this degradation as **IMU drift**. To isolate the specific failure modes of long-distance AR navigation, it is necessary to decompose the general drift into its fundamental kinematic components: rotational and translational errors.

Rotational error occurs when small angular measurement inaccuracies accumulate in the gyroscope. Because the accelerometer continuously measures the absolute gravity vector, the VIO system can continuously self-correct pitch and roll with high accuracy; however, gravity provides no reference for yaw. Consequently, the system relies on the magnetometer, which is highly susceptible to interference from indoor metal structures, often leading to a gradual misalignment of the user's heading [18].

Conversely, **translational error (or positional drift)** is an inherent limitation of inertial navigation. To calculate the device's physical displacement on a floor plan, a tracking system must integrate the device's acceleration over time. Because of this integration process, even microscopic sensor noise compounds, creating a physical offset. Yamaguchi et al. emphasise that this IMU drift is a "well-established fact" in AR technologies, demonstrating empirically that these errors continuously widen over time and severely degrade location-tracking accuracy [7]. As a result of this translational drift, the virtual navigation path slowly diverges from the physical corridors. Since failed long-distance indoor routing is primarily by accumulated positional error rather than minor rotational misalignment, implementing robust error correction for this translational displacement is critical for functional AR frameworks [8, 9].

2.2 The Computational Overhead of Map-Based Mitigation

To correct this translational drift, prior research has largely focused on continuous absolute localisation, forcing the mobile device to match its live camera feed against pre-existing environmental maps. Systems such as ViNav

rely heavily on dense image datasets [10], while others utilise exhaustive 3D SFM point clouds [2] or process real-time LiDAR scans against heavy spatial maps [11].

While these methods theoretically mitigate drift, they introduce a severe computational bottleneck. Continuously performing complex feature matching on massive spatial maps is highly computationally intensive on standard mobile devices [13]. As highlighted in recent reviews of indoor localisation, sustaining this level of continuous on-device processing introduces significant limitations regarding the cost of continuous battery use [14]. Furthermore, this strict hardware constraint effectively limits reliable AR navigation to expensive flagship smartphones, creating a major barrier to universal accessibility by excluding users with mid-range or older mobile devices [15].

2.3 Infrastructure-Assisted Tracking and the Checkpoint Paradigm

To bypass the on-device computational bottleneck, recent approaches have explored offloading tracking responsibilities to external infrastructure. Wireless modalities such as Bluetooth Low Energy (BLE) [3, 5] or Ultra-Wideband (UWB) [4] are highly resource-efficient; however, they generally provide only coarse, meter-level coordinate positioning. This fails to meet the strict user requirements for indoor smartphone localisation, which demands GNSS-like functionality and centimetre-level precision to accurately anchor 6-DoF AR assets [6].

Alternatively, utilising networks of external cameras and multi-modal infrastructure to assist tracking provides high accuracy by leveraging fixed infrastructure rather than forcing the mobile device to scan the environment. For instance, the StageAR framework utilises sparse external cameras for markerless localisation [16], while systems such as UVtrack fuse external vision sensors with UWB communications to achieve the centimetre-level precision required for seamless indoor tracking [4]. However, requiring continuous camera coverage or dense multi-sensor infrastructure to track a user imposes prohibitive deployment costs and immense data-processing overhead. This reveals a critical gap in the literature: the lack of a computationally lightweight, infrastructure-assisted hybrid approach. As Liu et al. note, the antagonistic features of VIO (smooth but drifting) and absolute localisation (drift-free but computationally heavy) must be optimally fused to ensure ro-

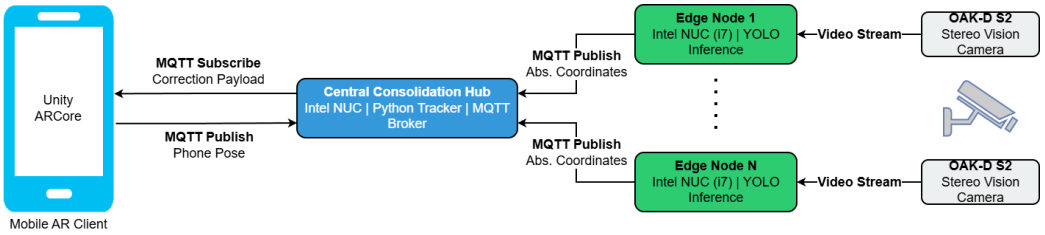


Figure 5: Architectural Diagram of the System

bust tracking [9]. However, rather than forcing this computationally heavy fusion to occur on-device, offloading the intensive visual processing to local edge nodes offers a superior hardware solution. By utilising an abstract geometric 3D floor model as a shared reference frame, sparse infrastructure cameras can act purely as intermittent positional "checkpoints," correcting the translational drift without triggering on-device thermal throttling or requiring continuous camera coverage.

3 Methodology and System Architecture

To address the computational bottlenecks and accumulated translational drift identified in the literature, this research implements an edge-assisted, infrastructure-supported AR navigation system. The architecture is designed to completely decouple the heavy visual processing required for absolute positioning from the mobile client, utilising a local IoT network to handle spatial calculations.

3.1 System Architecture

The proposed system in Figure 5 operates through a distributed architecture comprising three primary tiers: a lightweight mobile AR client, a distributed edge-computing vision network, and a central consolidation hub linked via a low-latency communication protocol.

3.1.1 The Mobile Client

The user-facing application is developed in Unity (version 6.3) for deployment on standard Android-based commercial mobile hardware. To manage

continuous, short-term 6-DoF tracking, the application relies on the standard Visual-Inertial Odometry (VIO) pipeline provided by ARFoundation. Crucially, the mobile device is completely relieved of any spatial mapping or external feature-matching responsibilities. Its sole computational tasks are rendering the abstract 3D navigational path, obtaining odometry via VIO, and listening for coordinate correction payloads from the external network.

3.1.2 The Infrastructure, Edge Nodes, and Central Hub

The external tracking infrastructure consists of a network of OAK-D S2 stereo cameras serving as sparse positional checkpoints. Rather than connecting directly to the cloud, these cameras feed visual data to local edge computing nodes—Intel NUCs (10th Gen i7) utilising integrated GPUs. These local edge nodes perform the computationally intensive machine learning inference (via YOLO26) required to detect the user within each camera's isolated field of view.

To coordinate this distributed system, a Central Hub consolidates the telemetry from all active edge nodes. When a user passes through a checkpoint, the edge node sends the localised spatial data to the central hub, which then calculates the user's definitive absolute pose across the global 3D Unity floor plan. By processing video feeds locally at the edge and strictly passing abstract coordinate data to the central hub, the system completely mitigates the privacy risks associated with transmitting or storing raw, continuous camera footage.

3.1.3 Network and Communication Pipeline

To facilitate communication between the mobile client, the edge nodes, and the central hub, the architecture relies exclusively on the Message Queuing Telemetry Transport (MQTT) protocol over a local wireless network. While User Datagram Protocol (UDP) is theoretically favoured for extremely low-latency telemetry due to its lack of connection overhead, implementing a custom UDP pipeline introduces significant development overhead and potential network vulnerabilities.

For this architecture, MQTT was selected as the optimal pragmatic solution. First, it provides a robust, built-in security and authentication layer—essential for protecting telemetry across a distributed edge network—without requiring the development of custom encryption protocols. Second, leverag-

ing a centralised MQTT broker simplifies the network topology, allowing the central hub to seamlessly route absolute pose data to the mobile client via a publish-subscribe model. Operating within the high-bandwidth environment of a localised university network, this pipeline ensures that mission-critical coordinate payloads are delivered reliably, securely, and within the required sub-second latency thresholds.

3.2 The Checkpoint Correction Mechanism

The core function of the proposed architecture is to seamlessly correct accumulated translational drift without disrupting the mobile device's internal tracking loop. Standard AR tracking engines, specifically Google's ARCore (accessed via Unity's ARFoundation), operate as a closed "black box." The engine tightly couples the virtual AR Camera to the device's physical hardware sensors to maintain its Visual-Inertial Odometry (VIO) pipeline. Attempting to force or teleport the AR Camera to a corrected absolute coordinate natively disrupts this pipeline; ARCore will immediately attempt to override the manual change, snapping the camera back to its internally calculated position and causing severe visual glitching and tracking failure. To bypass this limitation without interfering with ARCore's inner workings, this research implements a "nudge", or an inverse environmental translation, effectively sliding the abstract 3D map beneath the user.

3.2.1 The Coordinate Pipeline

The correction sequence is triggered strictly when a user enters the physical field of view of a sparse infrastructure camera. The local edge node processes the video frame, utilising a YOLO26 inference pipeline to identify the user's pedestrian bounding box. By mapping the user's pixel coordinates to a pre-calibrated homography matrix of the physical floor plan, the edge node calculates the user's absolute, real-world positional coordinates (X_{abs}, Z_{abs}). This data is instantly published by the central hub via MQTT as a lightweight JSON payload. Because the scope of this research is strictly limited to 2D translational drift across a floor plan, the vertical Y-axis (height) and rotational data (yaw, pitch, roll) are intentionally omitted from the correction payload, allowing the smartphone to continue managing these metrics natively via gravity and gyroscope tracking.

3.2.2 Pedestrian Detection and Target Selection

Building on the coordinate pipeline described above, each infrastructure checkpoint executes a three-stage computer vision and mapping pipeline on the local Intel NUC edge node:

1. **Inference and Target Selection:** The edge node runs a lightweight YOLO26 object detection model. While the theoretical ideal would be to track the smartphone directly to calculate its exact spatial offset, empirical testing demonstrated that this approach is highly unreliable at extended distances and highly susceptible to geometric occlusion by the user's body (see Appendix A for empirical justification). To ensure robust reliability, the system designates the user's torso, specifically the region spanning the shoulders to the waist, as the primary tracking target, as visualised in Figure 6. By bounding this larger, morphologically stable feature, the model achieves consistently high detection confidence. To approximate the device's location and represent the user's physical footprint on the floor plan, the system extracts the bottom-centre pixel coordinate of this torso bounding box.

2. **Spatial Transformation:** Because the YOLO26 model outputs 2D pixel coordinates, this data must be translated into the physical 3D space. The edge node utilises a pre-calibrated homography matrix specific to that camera's physical installation angle and height. By applying a perspective transformation, the system maps the bottom-centre 2D pixel coordinate to an absolute real-world coordinate (X_{abs}, Z_{abs}) relative to the fixed global origin of the physical floor plan.

3. **Coordinate Frame Alignment and Payload Generation:** A critical challenge in integrating external spatial data with Unity is the discrepancy between reference frames. The infrastructure network calculates positions relative to its fixed physical origin, whereas Unity's ARCore dynamically initialises its local origin $(0, 0, 0)$ wherever the application is launched. Furthermore, inherent discrepancies in coordinate handedness between the computer vision models and Unity's left-handed spatial system require explicit transformation. To resolve this, the X and Z axes are strictly aligned and swapped during the consolidation phase. Before transmission, these raw spatial coordinates are processed through a modified, 6-state kinematic Kalman filter to

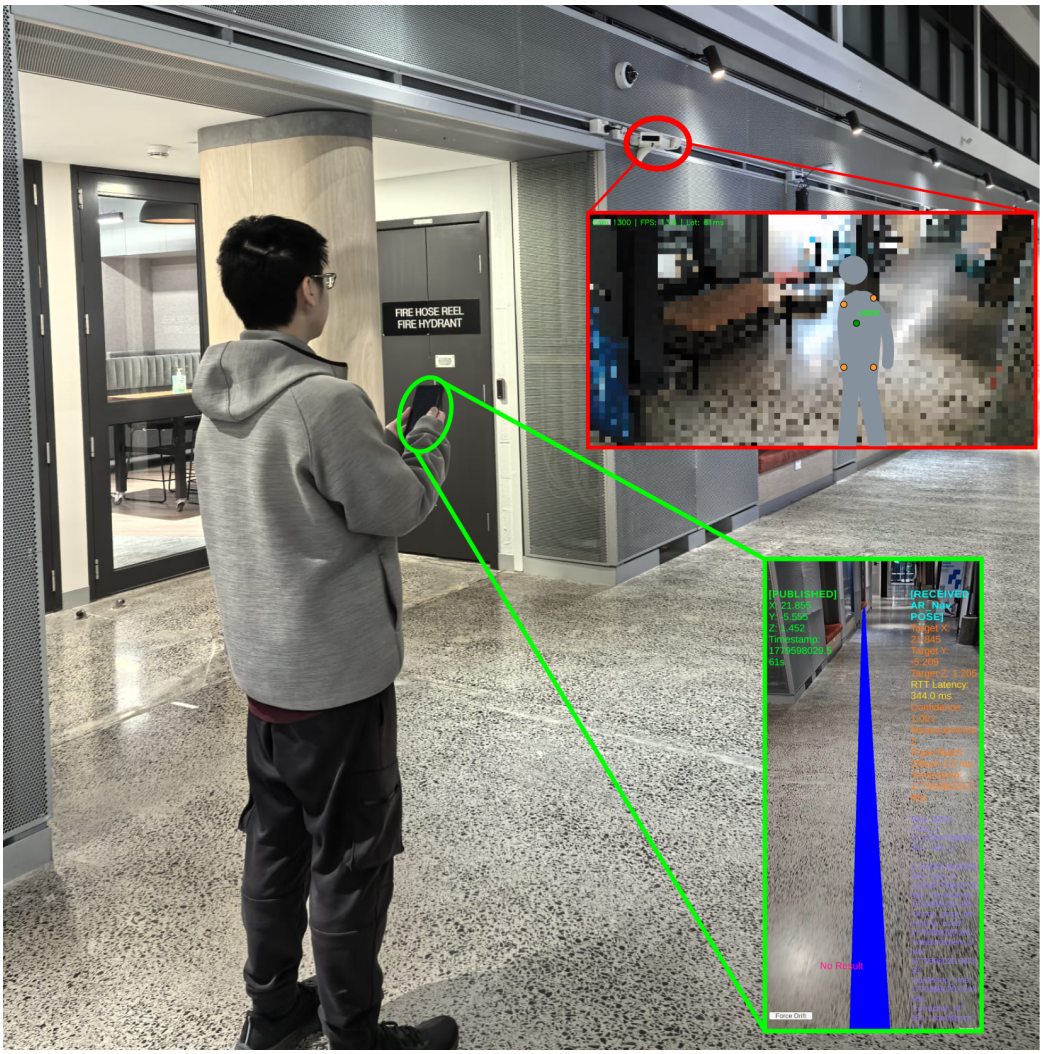


Figure 6: Concurrent visual perspectives of the edge-assisted architecture. The mobile AR client renders the local navigational trajectory, while the external infrastructure camera simultaneously performs YOLO26 torso detection to extract absolute spatial coordinates.

eliminate high-frequency tracking jitter and safely bridge infrastructure blind spots using a custom velocity-decay mechanism (the complete mathematical formulation is provided in Appendix B). Finally, the central hub formats this smoothed coordinate data into a lightweight JSON payload (e.g., "device_id": "client_01", "x": 7.5, "z": -2.3).

3.2.3 The Inverse Translational Logic

Upon receiving the MQTT correction payload, the Unity mobile client executes a C# script to adjust coordinates. Let the user's current drifted position in the virtual space, as tracked by ARCore, be denoted as (X_{vio}, Z_{vio}). When the absolute coordinates (X_{abs}, Z_{abs}) are received, the system calculates the accumulated translational drift offset:

$$\Delta X = X_{abs} - X_{vio}$$
$$\Delta Z = Z_{abs} - Z_{vio}$$

Rather than attempting to override ARCore's native camera control, the Unity client applies an inverse transformation to the root GameObject of the entire AR session. The environment is mathematically translated by a vector of $(\Delta X, 0, \Delta Z)$. To ensure this correction does not jar the user or induce simulator sickness, the C# script utilises a Vector3.Lerp (Linear Interpolation) function to smoothly interpolate the slide over a short duration until the virtual corridors and physical walls are perfectly aligned.

3.2.4 Preservation of the VIO Pipeline

This inverse translation method represents a highly efficient, computationally lightweight correction paradigm. By shifting the environment rather than the camera, the system lets ARCore remain the undisputed master of local spatial tracking and floor plane estimation, while relying on the external edge network strictly as the absolute ground-truth reference. The user experiences a seamless correction of their physical position on the map without compass misalignment, tracking loss, or the visual stuttering associated with on-device spatial recalculations.

3.3 Ethical Considerations and Data Privacy

The experimental methodology strictly adhered to data privacy protocols regarding spatial computing and edge-assisted research. The primary sensory dataset collected during the experimental navigational runs consisted exclusively of anonymised spatial telemetry, specifically continuous coordinate mapping (X, Y translation vectors) and hardware resource utilisation logs.

Crucially, while the system relies on vision-based tracking, the camera feeds utilised by the ARCore pipeline and the external OAK-D sensors were processed strictly for real-time spatial feature extraction. To facilitate qualitative system debugging and trajectory verification, temporary first-person video recordings were locally captured from the mobile device's perspective during select experimental runs. Strict data minimisation protocols were enforced during this process; the researcher actively navigated to avoid the incidental capture of bystanders or Personally Identifiable Information, such as facial features.

Furthermore, a strict data retention and destruction policy was applied to these visual records. The qualitative debugging recordings were utilised solely for internal trajectory verification and are scheduled for permanent deletion immediately following the conclusion of the data analysis phase. At no point were these video feeds transmitted to the MQTT edge infrastructure, nor are they included in any permanent public datasets. Because the finalised analytical data is completely devoid of PII and limited entirely to geometric spatial metrics, the collection process maintains strict user anonymity and poses no ongoing privacy risk.

4 Experimental Setup

To empirically evaluate the efficacy of the edge-assisted correction architecture, real-world trials were conducted within a controlled, GNSS-denied indoor facility. The experimental setup was designed to replicate the challenging spatial conditions that typically induce Visual-Inertial Odometry (VIO) drift in standard smartphone applications.

4.1 The Testing Environment

The system was deployed and evaluated within the Monash Innovation Lab. The facility is characterised by an offset T-shaped architectural layout, featuring long linear corridors with a central open foyer located midway along the primary vertical axis. While the overall geometry of the building is relatively straightforward, the environment naturally introduces the magnetometer and accelerometer noise commonly encountered in real-world indoor deployments, such as varying artificial lighting and dense electronic infrastructure, thereby providing a realistic baseline for evaluating translational drift correction.

The specific navigation routes were mapped along these structural axes to incorporate visual features known to challenge monocular tracking systems. All three experimental paths commenced in a shared initial corridor characterised by an asymmetric visual profile: an interrupted planar boundary on the user's left, opposite a highly detailed laboratory environment on the right. The left-hand boundary is intermittently broken by large doorway entrances.

4.2 Apparatus and Hardware Deployment

For all real-world experimental trials, the mobile AR client was executed on a Samsung Galaxy S23+ smartphone (released in 2023) running Android 16. To validate the hypothesis that a computationally lightweight, infrastructure-assisted AR system can operate without continuous camera tracking, dense spatial coverage was intentionally avoided. Instead, a sparse network of OAK-D S2 stereo cameras was deployed (a physical visual of the camera hardware is provided in Appendix C). To establish the distributed edge network, seven separate Intel NUC (10th Gen i7) edge nodes were distributed across the Monash Innovation Lab, as detailed in the deployment map in Figure 7. This supporting edge infrastructure was designed by Matthew Willaton and funded by Monash University, with each node tasked with managing two dedicated cameras.

During initial deployment, empirical testing revealed that the cameras' legacy factory calibration (relying on pre-2024 spatial models) lacked the precision required for accurate coordinate mapping. Due to the study's temporal constraints, rigorous manual calibration and homography matrix generation were prioritised for five specific NUC-camera sets. Crucially, these five highly

Figure 7: Top-Down Map of Infrastructure Cameras in Experimental Setup

calibrated nodes were not chosen at random; they were strategically selected for their proximity to the designated navigational end-markers. While all seven nodes remained active within the infrastructure network, prioritising precise calibration of checkpoints near the route's terminal points ensured maximum spatial accuracy at the locations where the user's Final Drift Error (FDE) would be recorded.

4.3 Experimental Tasks and Route Definitions

4.3.1 The Navigation Task

The primary experimental task required the user to physically traverse a set of predefined indoor environments while guided exclusively by the mobile AR application. For each trial, the user positioned the device at a designated starting coordinate, initialised the ARCore tracking session, and followed the rendered 3D navigational line at a standard pedestrian walking pace. During the task, the user was required to keep the device consistently oriented forward to ensure the camera continuously captured environmental features for the VIO pipeline. Crucially, the user was tasked with strictly following the virtual trajectory, regardless of its spatial drift, executing the "forced completion" protocol (manually bypassing obstacles) only if the system triggered a critical failure by routing into a physical barrier.

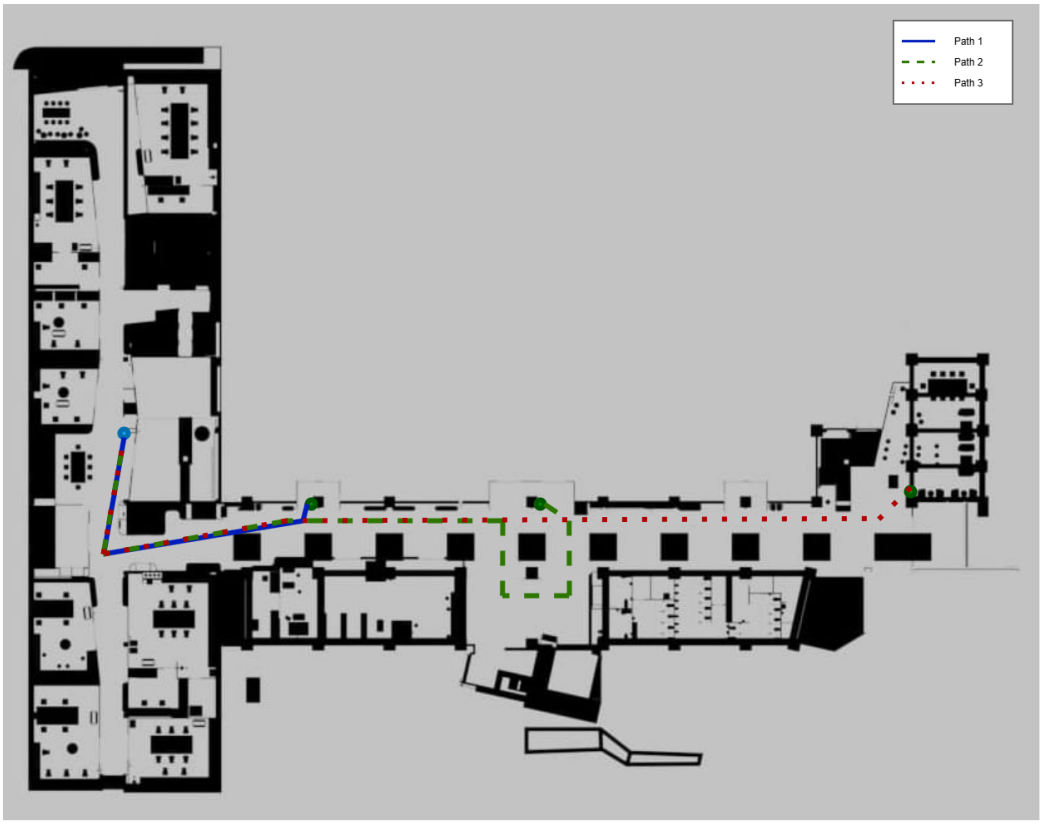


Figure 8: Top-down floor plan of the Monash Innovation Lab detailing the three experimental routes. All trajectories share an initial asymmetric corridor featuring depth discontinuities (doorways on the left). Path 1 (Blue/Solid) terminates after a short distance. Path 2 (Green/Dashed) executes a continuous rotational loop within the central depth-expansion foyer. Path 3 (Red/Dotted) traverses the maximum building length, exposing the tracking pipeline to severe perceptual aliasing induced by the repetitive structural support beams along the primary corridor.

4.3.2 Definition of Navigation Routes

To thoroughly evaluate the system's resilience across various spatial complexities, the navigation task was executed across four distinct structural routes within the Monash Innovation Lab (as visualised in the top-down trajectory map in Figure 8):

- **Path 1 (Short Traverse):** A concise, direct route designed to establish a control measurement for standard VIO tracking performance. This path assesses the system's foundational accuracy over a minimal distance, providing a reference value for Final Drift Error (FDE) before severe accumulated spatial drift begins.
- **Path 2 (Medium Traverse with Loop Detour):** A moderately extended route that incorporates a complex, near-complete loop detour near the terminal point of the trajectory. This specific geometry was designed to severely stress the system's rotational tracking and global orientation estimation, exposing the VIO pipeline's vulnerability to uncorrected angular drift (yaw error) during intricate navigational manoeuvres.
- **Path 3 (Full-Length Traverse):** An extended, building-scale route traversing the entire length of the Innovation Lab. This maximum-distance trial was explicitly designed to stress-test the absolute limits of the tracking pipeline by accelerating the accumulation of catastrophic translational drift, thereby reliably triggering and measuring critical spatial failures (e.g., barrier collisions).
- **Trajectory and Resource Path (Path 4):** A single, strictly predefined route equipped with physical ground-truth waypoints that replicates the exact long-distance straightaway geometry of Path 3. This specific trajectory was utilised exclusively to capture continuous pose data (trajectory mapping) and to benchmark the smartphone's hardware resource consumption during the identical spatial conditions that induce severe perceptual aliasing.

4.4 Experimental Procedure

To quantify the efficacy of the edge-assisted correction architecture against standard VIO tracking, all experimental trials were evaluated under two

distinct operational conditions: **Baseline Mode** (standard ARCore VIO with no external assistance) and **Assisted Mode** (infrastructure-supported map translation active).

4.4.1 Drift and Failure Trials (Paths 1–3)

For each of the three primary navigation routes (Paths 1, 2, and 3), 15 trials were conducted in Baseline Mode and 15 in Assisted Mode, resulting in a total of 90 evaluation walks. Each individual trial commenced with the user scanning a physical ground-truth start marker, which instantiated the InnovationLabMap v5 prefab and initialised the virtual navigation trajectory. The user then strictly followed the rendered AR directional line.

If the user successfully reached the end of the virtual path, they scanned a physical stop marker to record their terminal position. However, if the accumulated spatial drift caused the virtual trajectory to route the user into a physical barrier, the researcher executed a "forced completion" protocol. The researcher manually bypassed the physical obstacle while maintaining the active ARCore camera feed, forcing the device to the terminal destination coordinate. This ensured that the terminal degradation of the unassisted VIO pipeline could still be recorded even after a catastrophic navigational failure.

4.4.2 Trajectory and Hardware Trials (Path 4)

To capture continuous pose data and hardware benchmarks, strictly controlled resource runs were conducted on Path 4 (3 runs in Baseline Mode and 3 in Assisted Mode). To ensure consistent physical traversal across all runs, this predefined route was manually demarcated on the laboratory floor using masking tape. These physical tape markings served as explicit ground-truth waypoints for the user to follow. During these trials, the user walked the route by adhering as closely as possible to these visual markers, while the smartphone's estimated pose and native hardware resource metrics were continuously logged via custom Unity telemetry scripts.

4.5 Evaluation Metrics

The data extracted from the experimental procedures were evaluated against four primary metrics to quantify the system's spatial accuracy, safety, and efficiency.

4.5.1 Measurement 1: Final Drift Error (FDE)

Final Drift Error (FDE) measures the absolute terminal positional accuracy of the tracking system. As established by Kim et al., FDE is a critical quantitative metric for evaluating Visual-Inertial Odometry (VIO) systems because it calculates the absolute end-point position error without requiring a continuous external motion capture setup [17].

In the context of this research, FDE is the most vital evaluation metric because it directly quantifies the cumulative translational drift—the exact failure mode that causes long-distance AR navigation to route users into physical barriers. It is defined here as the absolute spatial deviation (measured in centimetres) between the user's true physical standing position and the intended physical destination when the virtual navigation system indicates the route is complete. An ideal FDE value is zero, where any larger value directly denotes an inaccurate position estimate resulting from uncorrected accumulated tracking drift [17].

4.5.2 Measurement 2: Critical Spatial Failure Rate

To assess the practical safety and reliability of the navigation system, critical failures were evaluated strictly through post hoc video coding of the recorded runs to eliminate observer bias. The primary motivation for this metric is fundamental to spatial computing: an AR navigational path must never lead a user into impassable physical locations.

Therefore, a critical spatial failure was defined strictly by an objective environmental collision. Specifically, this occurs whenever any portion of the rendered 3D navigational line clips into, intersects, or directs the user directly into a permanent physical architectural barrier, such as a structural wall or pillar. Dynamically changing obstacles, such as movable furniture, were explicitly excluded from this failure condition as they do not represent fixed environmental boundaries.

To ensure maximum methodological rigour, a zero-tolerance boundary heuristic was established. Rather than applying a subjective allowance threshold (e.g., logging a failure only if a third of the line is occluded), any degree of architectural clipping was recorded as a catastrophic failure. Routing a user into a solid structure fundamentally breaks both the utility and the safety of the navigation system (see Appendix D for visual documentation of these failure states).

4.5.3 Measurement 3: Trajectory Deviation

While FDE measures terminal error, Trajectory Deviation evaluates the continuous accuracy of the estimated path. By overlaying the device's continuously logged estimated pose data onto the established ground-truth waypoints of the physical floor plan (captured during Path 4), this metric explicitly evaluates the distance error. Specifically, it visualises exactly how the continuous travel distance progressively degrades VIO spatial accuracy and demonstrates the spatial efficacy of the infrastructure "checkpoints" in correcting this trajectory deviation.

4.5.4 Measurement 4: Hardware Resource Consumption

To validate the hypothesis that offloading visual processing to edge nodes maintains universal device accessibility (Objective 3), the system's hardware footprint was benchmarked. To accurately capture resource consumption without incurring the performance overhead of external profiling software, telemetry scripts were integrated directly into the Unity application. This approach ensured that the computational load was logged natively during the active AR session. The system's hardware footprint was quantified using three primary metrics:

1. **Average Frame Time:** Logged internally via Unity to evaluate rendering stability and identify processing bottlenecks.
2. **Memory Usage:** Tracked programmatically to measure the active RAM footprint required to maintain the navigation state.
3. **Battery Discharge:** Recorded over the duration of the runs via native OS API calls to evaluate overall power efficiency.

4.5.5 Statistical Analysis Methodology

To rigorously evaluate the performance of the edge-assisted architecture against the unassisted Baseline VIO, a formal statistical pipeline was established. Following standard hypothesis testing frameworks, the analysis was conducted on the data collected from the 90 experimental trials (Final Drift Error datasets) to determine if the observed spatial improvements were statistically significant. The pipeline was structured via the following procedures:

- **Outlier Exclusion (IQR Filter):** First, an Interquartile Range (IQR) filter ($1.5 \times IQR$) was applied to the collected FDE datasets.
  - *Motivation:* The purpose of this filter is to isolate systemic algorithmic performance from extreme, random experimental anomalies (e.g., momentary hardware stalls or physical user tripping) that could artificially skew the dataset variance.
- **Bootstrap Analysis (Mean Trajectory Improvement):**
  - *Motivation:* VIO drift data is fundamentally bounded at zero and often highly right-skewed, meaning it violates the strict normality assumptions required for standard parametric tests (such as a Student's t-test). Because of this failure of the distribution assumption, a simulation-based method was explicitly chosen to evaluate the mean.
  - *Methodology:* A $20,000$-iteration bootstrap analysis was conducted on the data collected. Bootstrapping is a simulation-based method that works by repeatedly resampling the collected dataset with replacement to construct an empirical sampling distribution. From these $20,000$ simulated samples, a 95% Confidence Interval (CI) was constructed to estimate the true mean improvement in the trajectory.
- **Mann-Whitney U Test (Systematic Shift in Errors):**
  - *Motivation:* Because the FDE data distribution violates parametric normality assumptions, this non-parametric test was selected to evaluate whether the edge-assisted system produced systematically smaller positional errors than the baseline.
  - *Hypotheses:* Because the study specifically tests whether the assisted architecture lowers the tracking error, this is framed as a one-tailed test.
    - **Null Hypothesis (H_0):** There is no difference in the distribution of Final Drift Errors between the Baseline VIO and the Edge-Assisted architecture ($H_0 : \text{Median}_{Baseline} = \text{Median}_{Assisted}$).
    - **Alternative Hypothesis (H_A):** The Edge-Assisted architecture produces systematically smaller errors than the Baseline VIO ($H_A : \text{Median}_{Baseline} > \text{Median}_{Assisted}$).

- *Methodology:* The test is performed by combining all data points from both groups, ranking them numerically from smallest to largest, and then comparing the sum of the ranks for each group to determine if one system consistently ranks lower (possesses smaller error) than the other.

- **Brown-Forsythe Test (Reduction in Spatial Variance):**
  - *Motivation:* In AR navigation, consistency is as critical as average accuracy; a system that is highly accurate on average but wildly unpredictable is fundamentally unsafe. The Brown-Forsythe test was selected to evaluate variance because it centres the data using the median rather than the mean, making it significantly more robust to non-normal distributions and outliers than a standard F-test.
  - *Hypotheses:* This is a two-tailed test focused on variance equality.
    - **Null Hypothesis (H_0):** The spatial variance (consistency) of the Final Drift Errors is identical between both tracking systems ($H_0 : \sigma^2_{Baseline} = \sigma^2_{Assisted}$).
    - **Alternative Hypothesis (H_A):** The spatial variance is significantly different between the two systems ($H_A : \sigma^2_{Baseline} \neq \sigma^2_{Assisted}$).
  - *Methodology:* The test is performed by transforming the original FDE data into absolute deviations from each group's respective median. An independent statistical comparison is then conducted on these transformed absolute deviations to test for equality of variances between the two tracking systems.

5 Results

To evaluate the efficacy of the infrastructure-assisted AR navigation system, empirical data were gathered across 90 designated evaluation walks. The evaluation compares the performance of the proposed architecture (Assisted Mode) against the unassisted Visual-Inertial Odometry pipeline of Google ARCore (Baseline Mode).

5.1 Final Drift Error (FDE) and Variance

As established in Section 4.4.1, the Baseline runs on Path 3 frequently resulted in catastrophic spatial failures, requiring the "forced completion" protocol to manually bypass physical obstacles and record a terminal coordinate. Therefore, the unassisted Baseline FDE data strictly represent terminal error, including these forced completions.

Table 1: Statistical Summary of Final Drift Error (FDE) and Variance Reduction

Navigation Route	**Mean Improvement** (Meters)	**Bootstrap 95% CI** (Meters)	**Variance Reduction** (Brown-Forsythe p-value)	**Systematic Shift** (Mann-Whitney p-value)
Path 1 (Short Traverse)	0.043	$[-0.06, 0.15]$	$< \mathbf{0.01}^{**}$	0.253
Path 2 (Loop Detour)	0.008	$[-0.18, 0.22]$	$< \mathbf{0.01}^{**}$	0.901
Path 3 (Full-Length)	1.300	$[0.82, 1.84]$	$< \mathbf{0.01}^{**}$	$< \mathbf{0.001}^{***}$

$^{**}p < 0.01$, $^{***}p < 0.001$. Outliers removed via $1.5 \times$ IQR threshold prior to analysis. see the comparative outlier analysis in Appendix E

For Path 1, after removing extreme outliers, we observed a negligible improvement in the mean trajectory of 0.043m. The subsequent bootstrap analysis supported the lack of a significant shift, generating a 95% Confidence Interval (CI) that crossed zero ([-0.059m, 0.147m]), aligning with the non-significant Mann-Whitney result ($p = 0.253$). However, the Brown-Forsythe test demonstrated a highly significant reduction in tracking variance when the edge-assisted pipeline was active ($F = 8.97, p < 0.01$).

Similarly, for Path 2, we observed a minimal mean improvement of 0.008m after outlier exclusion. The bootstrap analysis generated a 95% CI of [-0.183m, 0.215m], confirming no significant shift in the central mean. Yet, the statistical analysis again revealed an extremely significant reduction in spatial tracking variance when the edge checkpoints were utilised ($F = 12.40, p < 0.01$).

For the extended trajectory of Path 3, the filtered data revealed a substantial observed reduction in the mean absolute error of 1.301 meters. The bootstrap analysis strongly validated this improvement; an aggressively calculated 99% Confidence Interval remained strictly positive ([0.694m, 2.016m]), proving a definitive spatial correction even at the highest thresholds of statistical stringency. Furthermore, the Mann-Whitney U test indicated that the baseline tracking errors were systematically larger than the assisted errors ($U = 180.0, p < 0.001$), and the Brown-Forsythe test confirmed a significant reduction in overall spatial variance ($F = 9.28, p < 0.01$).

5.2 Trajectory Deviation and Critical Failures

To evaluate continuous spatial adherence, the recorded spatial coordinates of the mobile client were plotted against the true physical path, and Dynamic Time Warping (DTW) was applied to measure the continuous spatial deviation. Furthermore, critical failures (defined as the AR trajectory clipping through a permanent environmental barrier) were tallied.

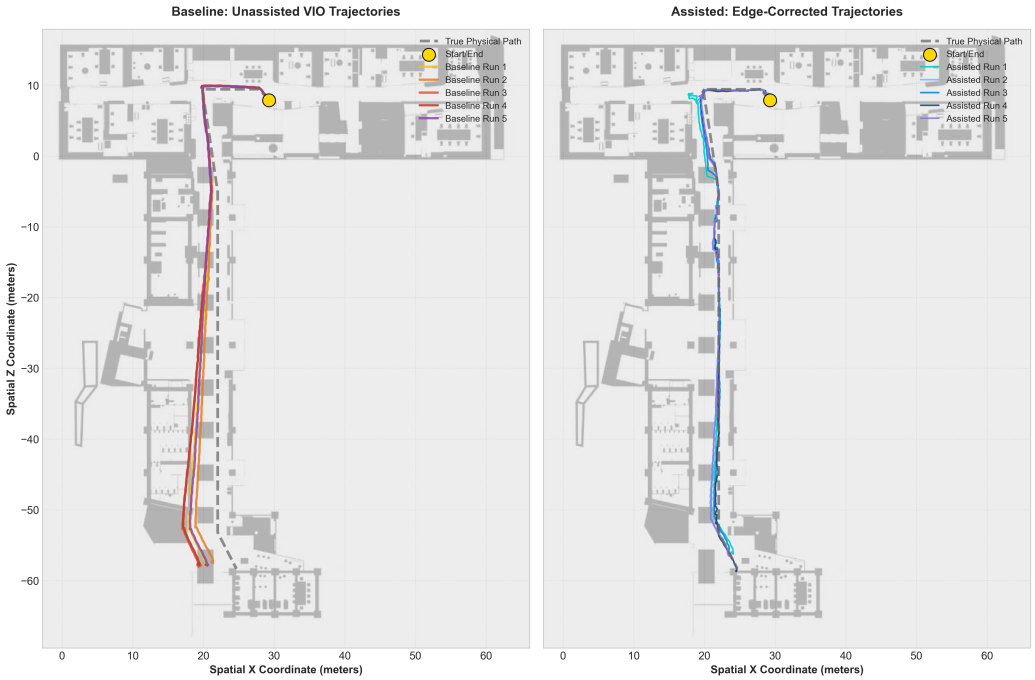


Figure 9: Visual Trajectory Analysis for Path 3 plotting the continuous spatial coordinates of the Baseline (left) and Assisted (right) experimental runs.

The visual trajectory overlay (Figure 9) demonstrates the spatial inflection point of the tracking divergence on Path 3. To further quantify this divergence across the entire route, Figure 10 illustrates the average continuous Dynamic Time Warping (DTW) deviation over the trajectory lifecycle. As visualised, the unassisted baseline runs suffered from compounding translational drift, resulting in an average continuous DTW trajectory deviation of 1.88 meters, with individual runs exceeding 2.19 meters. Conversely, the assisted runs yielded an average DTW deviation of just 0.37 meters. This

represents an 80.5% reduction in continuous spatial error, as the external checkpoints effectively bounded the variance of error. For a granular, time-series breakdown of this spatial correction mechanism across individual experimental runs, refer to Appendix F.

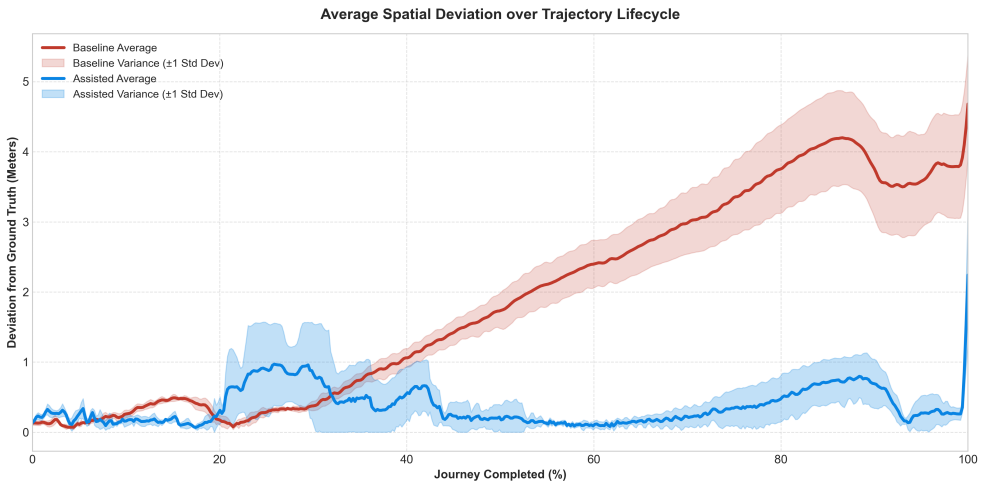


Figure 10: Average spatial deviation over the trajectory lifecycle for Path 3 (measured via DTW). The solid lines represent the mean deviation from the physical ground truth, while the shaded regions denote the ± 1 standard deviation's variance. The edge-assisted architecture (blue) effectively suppresses the compounding drift inherently observed in the unassisted baseline (red).

Regarding critical safety, the compounding translational drift observed in the unassisted Baseline runs consistently caused the virtual trajectory to route into impassable physical geometry. Across the 15 trials per route, the unassisted Baseline triggered 24 critical collisions on Path 1, 21 on Path 2, and reached a peak of 36 catastrophic spatial failures on the extended corridor of Path 3. The implementation of the edge-assisted architecture successfully intercepted and corrected these lateral drifts, reducing critical collisions to 10 on Path 1, 7 on Path 2, and 18 on Path 3. This represents a precise 50% reduction in critical navigational failures during the system's most rigorous spatial stress test.

Table 2: Dynamic Time Warping (DTW) Continuous Spatial Deviation for Path 3

Experimental Run	Baseline VIO (m)	Assisted Architecture (m)
Run 1	1.86	0.57
Run 2	1.42	0.39
Run 3	2.16	0.29
Run 4	2.19	0.22
Run 5	1.75	0.37
Mean Average	**1.88**	**0.37**
Overall System Improvement: 80.5% Error Reduction		

Table 3: Critical Spatial Failures (Environmental Collisions) per Navigation Route

Navigation Route	Baseline	Assisted	Failure Reduction
Path 1 (Short Traverse)	24	10	58.3%
Path 2 (Loop Detour)	21	7	66.7%
Path 3 (Full-Length)	36	18	50.0%
Total Across All Trials	**81**	**35**	**56.8%**

Note: Data represents total critical collisions aggregated across 15 trials per route.

5.3 Hardware Resource Overhead

Runtime performance metrics were logged continuously via custom Unity scripts during the Path 4 runs.

Both the baseline and assisted modes exhibited near-identical RAM usage, peaking at 307 MB. The mean rendering time remained stable across both modes (34 ms). However, the assisted mode recorded a maximum frame time spike of 1,501.93 ms (compared to the baseline's 517.23 ms). The standard deviation of the continuous frame times increased from 15.14 ms under the baseline to 37.45 ms under the assisted architecture. Finally, average battery drain per run increased from 1.0% in the baseline mode to 1.8% in the assisted mode.

Table 4: Resource Utilisation and System Overhead Summary (Path 4)

Performance Metric	**Baseline**	**Assisted**
Mean Frame Time (ms)	34.29	34.78
Frame Time Std. Dev (ms)	15.14	37.45
Max Frame Time Spike (ms)	517.23	1501.93
Peak RAM Usage (MB)	307	307
Avg Battery Drain/Run (%)	1.0%	1.8%

6 Discussion

The synthesis of the objective data presents a cohesive architectural narrative: standard mobile VIO degrades predictably based on spatial geometry, and the proposed external infrastructure dynamically serves as both a spatial stabiliser and a critical fail-safe recovery mechanism.

6.1 Contextualising Baseline FDE and Survivorship Bias

When interpreting the Final Drift Error (FDE) recorded in Section 5, it is vital to contextualise the true viability of the unassisted Baseline runs. Because the "forced completion" protocol was necessary to manually bypass physical obstacles and drag the device to the final coordinate, the Baseline FDE data technically represents a best-case "survivorship" scenario. The true navigational safety of the unassisted baseline is significantly worse than the terminal 1.88m error implies, as the native pipeline routinely failed to maintain a human-navigable path long before reaching the destination.

Key Insight: The fundamental takeaway from this observation is that evaluating indoor AR navigation systems solely on terminal metrics (like FDE) is methodologically dangerous. A tracking pipeline can technically arrive at a destination with a "moderate" terminal error while having completely violated spatial boundaries along the route. Therefore, this research demonstrates that future spatial computing literature must prioritise continuous spatial deviation (DTW) and critical environmental collision rates as the primary indicators of true system viability.

6.2 Spatial Stabilisation and Catastrophic Drift Recovery

The varying geometries of Paths 1, 2, and 3 successfully isolated specific failure modes within the ARCore tracking pipeline. In environments with frequent directional changes (Paths 1 and 2), standard mobile VIO maintains an acceptable mean accuracy but suffers from high-frequency spatial jitter, as computing accumulated angular velocity over sequential turns is notoriously difficult for smartphone IMUs. By yielding highly significant variance reductions ($p < 0.01$) without altering the central mean, the proposed system proved that it successfully serves as an algorithmic low-pass filter. The external OAK-D camera feeds acted as a rigid spatial anchor, effectively clamping down on rotational fluctuations.

Key Insight: In short, complex environments, the edge-assisted system functions purely as an algorithmic low-pass filter. This proves that infrastructure assistance is highly effective for variance stabilisation (clamping down on rotational fluctuations) even before catastrophic translational drift occurs.

Conversely, Path 3 induced a classic VIO "hallway problem." Faced with a long corridor flanked by identical structural beams, the camera lacks the distinct depth cues needed to accurately estimate forward translation, thereby triggering perceptual aliasing. Unable to extract velocity, ARCore begins to diverge smoothly but confidently from the true path, resulting in the lateral cascading drift visualised in Figure 9. In this extreme scenario, the edge-assisted architecture seamlessly transitioned from a stabiliser to a critical rescue mechanism. By intercepting the lateral drift using absolute external camera data, the system forcibly pulled the AR coordinate frame back onto the true physical path. While the sparse nature of the camera network means the system cannot entirely eliminate intermediate collisions within the blind spots between nodes, the architecture successfully halved the catastrophic failure rate (reducing Path 3 collisions from 36 to 18). In the context of indoor pedestrian navigation, the unassisted lateral deviation of nearly two meters constitutes a catastrophic safety failure; suppressing this to under 40 centimetres validates the architecture's capacity to maintain a vastly safer spatial envelope than native VIO.

Key Insight: In extended, feature-poor environments, edge assistance transitions from a mere stabiliser to a critical rescue mechanism. Suppressing a catastrophic two-meter lateral deviation down to under 40 centimetres validates that sparse checkpoints can successfully "hard-reset" the VIO loop,

bounding the error before it results in a permanent navigational failure.

6.3 Architectural Trade-offs: The Networking Tax and Battery Paradox

While spatial fidelity has improved drastically, the hardware metrics (Table 4) reveal the inherent computational overhead of asynchronous edge correction.

Positively, the architecture did not introduce systemic memory bloat (maintaining a 307 MB RAM footprint) and preserved a stable mean render time of 30 FPS. This confirms that the MQTT listener operates efficiently in the background. However, the doubling of the standard deviation of the frame time (from 15.14 ms to 37.45 ms) highlights a systemic "networking tax." When the mobile client receives an MQTT payload, the Unity engine must deserialise the incoming JSON string. In C#, frequent string manipulation allocates temporary memory, steadily increasing the frequency of the Unity Garbage Collector (GC)—which introduces micro-stutters to the main thread. Furthermore, forcibly translating the global AR coordinate map to align with the injected edge coordinates triggers abrupt spatial recalculations, resulting in visual volatility.

Finally, the hardware data reveals a "battery paradox" specific to spatial computing. Standard IoT paradigms frequently cite power conservation as a benefit of offloading processing to the edge. However, because continuous local tracking is strictly required for the AR interface to function smoothly between checkpoints, the mobile client cannot halt its internal ARCore pipeline. Consequently, the device sustains the maximum baseline VIO computational load whilst simultaneously expending additional energy to keep the active Wi-Fi radio listening for MQTT packets. This confirms that edge-assisted localisation in AR functions not as an energy-saving offload mechanism, but as an energy-consuming fail-safe, sacrificing battery efficiency (an 80% increase in per-run drain) to guarantee spatial safety.

Key Insight: The critical takeaway is that spatial computing cannot leverage standard IoT power-saving paradigms. Because the local VIO loop must remain active, developers must knowingly sacrifice battery efficiency to guarantee spatial safety.

6.4 Addressing Architectural Scalability

Ultimately, the statistical mapping across these diverse geometries proves the necessity of infrastructure offloading for building-scale AR. While native inside-out tracking is sufficient for simple, localised tasks, it fundamentally fails to scale, suffering from severe rotational jitter and eventual catastrophic drift. By mitigating both failure modes under varying environmental stressors without overwhelming the device's native hardware, the proposed architecture bridges the gap between local device limitations and the need for reliable, large-scale indoor spatial tracking.

Key Insight: Monocular VIO frameworks are mathematically insufficient for building-scale pedestrian routing. To achieve universal AR accessibility without relying on flagship processors, the industry should shift toward hybrid edge-infrastructure architectures that treat mobile devices as rendering clients rather than solitary spatial mappers.

7 Conclusion

Standard Android-based Augmented Reality (AR) frameworks, specifically Google's ARCore, are fundamentally constrained by their reliance on inside-out monocular tracking and software-driven depth estimation. When deployed on mobile hardware lacking dedicated physical depth sensors, these systems inevitably succumb to depth discontinuities and perceptual aliasing in complex, building-scale environments. This results in cumulative spatial drift, rendering native Visual-Inertial Odometry (VIO) pipelines unsafe for continuous pedestrian navigation. This research proposed, engineered, and evaluated a novel edge-assisted spatial architecture designed to rescue failing ARCore tracking by dynamically offloading absolute spatial coordinates from an external network of OAK-D stereo-depth sensors.

Ultimately, these experiments prove that infrastructure-assisted localisation successfully transitions standard mobile AR from a fragile, localised tool into a highly reliable, large-scale navigational framework. By systematically intercepting catastrophic lateral drift, yielding an 80.5% reduction in continuous spatial error and a 50% decrease in critical navigational collisions, the proposed architecture ensures continuous spatial safety. Furthermore, by successfully offloading the heavy vision-based inference to local edge nodes utilising OpenVINO-optimised models on integrated GPUs, the sys-

tem proves that building-scale AR can remain hardware-accessible without requiring flagship smartphone processors.

However, this research is subject to specific methodological and architectural limitations. Methodologically, the external infrastructure deployment relied on manual, static extrinsic calibration of the OAK-D sensors. While rigorous, this static approach does not account for micro-vibrations or subtle thermal shifts that may have introduced slight variations in mapping during the trials. Furthermore, the architecture's spatial safety is intrinsically tied to the checkpoint density; as demonstrated by the residual collisions on Path 3, the system remains vulnerable to drift accumulation in the blind spots between sparse sensor nodes.

Despite these limitations, the results provide clear avenues for future research. Primarily, the reliance on a TCP-based MQTT broker introduces quantifiable computational overhead and contributes to the aforementioned spatial battery paradox. Future iterations must transition to a low-latency, stateless UDP broadcast pipeline and investigate power-efficient radio duty-cycling to eliminate the "networking tax" and C# garbage-collection stutters experienced by the mobile client. Additionally, the central hub currently experiences target ambiguity when occluded by crowds; integrating predictive kinematic modelling into the edge logic will be necessary to maintain a probabilistic spatial lock on the primary user in crowded environments. Finally, to eliminate reliance on manual static calibration, future work should develop automated, dynamic, on-the-fly recalibration protocols across the external camera network, ensuring uniform depth estimation without periodic manual intervention.

References

[1] Ala Saleh Alluhaidan, Latifa Abdullah Almusfar, Ahmed Younes Shdefat, Ahmed Elsayed Mansour, Diaa Salama Abdelminaam, and Yasmin Alkady. From GPS to AR: Leveraging Augmented Reality and Grid-Based Systems for Improved Indoor Navigation. *IEEE Access*, 13:55211–55230, 2025.

[2] Leonardo Messi, Francesco Spegni, Massimo Vaccarini, Alessandra Corneli, and Leoanrdo Binni. Infrastructure-Free Localization System for Augmented Reality Registration in Indoor Environments: A First Accuracy Assessment. In *2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, pages 110–115, June 2024.

[3] Yue Gong, JoonOh Seo, Tae Wan Kim, Seungjun Ahn, and Yanfang Luo. Field validation of beacon-based indoor tracking and localization system for construction workers. *KSCE Journal of Civil Engineering*, 29(2):100017, February 2025.

[4] Yi Xu, Zhigang Chen, Ming Zhao, Fengxiao Tang, Yangfan Li, Jiaqi Liu, and Nei Kato. UVtrack: Multi-Modal Indoor Seamless Localization Using Ultra-Wideband Communication and Vision Sensors. *IEEE Open Journal of the Computer Society*, 6:272–281, 2025.

[5] Suhail Odeh, Murad Al Rajab, Jannat Natsheh, Isra' Zahran, Khader Ballout, and Mahmoud Obaid. LEADOW: An Empowering Indoor Navigation System for Individuals with Visual Impairments Using Bluetooth Beacons and Audio Guidance. In A. Mirzazadeh, Zohreh Molamohamadi, Efran Babaee Tirkolaee, Gerhard-Wilhelm Weber, and Janny Leung, editors, *Optimization and Data Science in Industrial Engineering*, pages 52–66, Cham, 2025. Springer Nature Switzerland.

[6] Günther Retscher. Indoor navigation—user requirements, state-of-the-art and developments for smartphone localization. 3(1):1–46.

[7] Shunpei Yamaguchi, Aditya Arun, Takuya Fujiwara, Misaki Sakuta, Ryotaro Hada, Takuya Fujihashi, Takashi Watanabe, Dinesh Bharadia,

and Shunsuke Saruwatari. Experience: Practical Challenges for Indoor AR Applications. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1030–1044, Washington D.C. DC USA, December 2024. ACM.

[8] Min Lu, Masatoshi Arikawa, Kohei Oba, Keiichi Ishikawa, Yuhan Jin, Tomihiro Utsumi, and Ryo Sato. Indoor AR Navigation Framework Based on Geofencing and Image-Tracking with Accumulated Error Correction. *Applied Sciences*, 14(10):4262, May 2024.

[9] Changkun Liu, Yukun Zhao, and Tristan Braud. MobileARLoc: On-device Robust Absolute Localisation for Pervasive Markerless Mobile AR. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 544–549, March 2024. ISSN: 2766-8576.

[10] Jiang Dong, Marius Noreikis, Yu Xiao, and Antti Ylä-Jääski. ViNav: A Vision-Based Indoor Navigation System for Smartphones. *IEEE Transactions on Mobile Computing*, 18(6):1461–1475, 2019. Publisher: Institute of Electrical and Electronics Engineers Inc.

[11] Devang Kishor Parab, Pranav Prasanna Deshpande, Raj Manoj Thakur, Vedant Atul Warke, and Shushma Khanvilkar. Indoor Navigation System using Augmented Reality. In *2024 International Conference on Inventive Computation Technologies (ICICT)*, pages 720–725, Lalitpur, Nepal, April 2024. IEEE.

[12] Shneka Muthu Kumara Swamy and Qi Han. Online Mitigation of Spatial Drift of Virtual Objects in Mobile Augmented Reality. In *Proceedings of the 2024 SIGCOMM Workshop on Emerging Multimedia Systems*, EMS '24, pages 33–38, New York, NY, USA, August 2024. Association for Computing Machinery.

[13] Petr Hořejší, Tomáš Macháč, and Michal Šimon. Reliability and Accuracy of Indoor Warehouse Navigation Using Augmented Reality. *IEEE Access*, 12:94506–94519, 2024.

[14] Rawaa Akram, Alireza Norouzi, Aseel H. Al-Nakkash, and Negar Majma. Machine learning for indoor localization based IoT: A review. *AIP Conference Proceedings*, 3211(1):030023, May 2025.

[15] J. R. V. Jeny, B. Sai Mahitha, M. Shashank, and N. Vamsi Krishna. EINS_ar: Enhanced Indoor Navigation System using Augmented Reality with Dynamic Path Adjustement. In *2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, pages 416–422, June 2025.

[16] Tao Jin, Shengxi Wu, Mallesham Dasari, Kittipat Apicharttrisorn, and Anthony Rowe. StageAR: Markerless Mobile Phone Localization for AR in Live Events. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 1000–1010, March 2024. ISSN: 2642-5254.

[17] Pyojin Kim, Jungha Kim, Minkyeong Song, Yeoeun Lee, Moonkyeong Jung, and Hyeong-Geun Kim. A benchmark comparison of four off-the-shelf proprietary visual–inertial odometry systems. 22(24):9873.

[18] Beomju Shin, Shiyi Li, and Boseong Kim. Deep Learning-Based Vehicle Speed Estimation Using Smartphone Sensors in GNSS-Denied Environment. *Applied Sciences*, 15(16):8824, August 2025.

Part 3: Appendices

A Empirical Justification for Torso Tracking Target Selection

As established in Section 3.2.2, the initial approach for the edge-assisted architecture was to track the mobile device directly. I attempted to have YOLO26 detect the development phone at varying distances within the room. However, empirical testing showed this was unreliable: the model failed to detect the phone when the user moved too far away or when the device was blocked from view by the user's body. Consequently, I resorted to tracking the user's torso, which proved to be significantly more consistent and reliable.

Limitation 1: Distance Constraints

Figure 11 documents the initial trials across varying distances using the infrastructure OAK-D S2 cameras. While YOLO26 successfully bound the smartphone at close and medium distances, the detection completely failed as the user stepped farther back into the corridor. At standard indoor navigation distances, the phone simply becomes too small within the camera's frame for the model to reliably extract a bounding box.

Figure 11: Visual documentation of smartphone tracking constraints. (Left and Centre) At close and medium ranges, YOLO26 successfully detects the mobile device. (Right) At extended distances, the model completely fails to detect the phone.

Limitation 2: Geometric Occlusion and Frame Cropping

Furthermore, relying on a smartphone introduces severe occlusion issues. Users frequently walk past and away from the camera nodes. As documented

in Figure 12, when a user navigates away from a checkpoint, the mobile device is entirely obstructed by their own body. Conversely, the torso remains fully visible, allowing the system to maintain a confident spatial lock ("USER") even from behind.

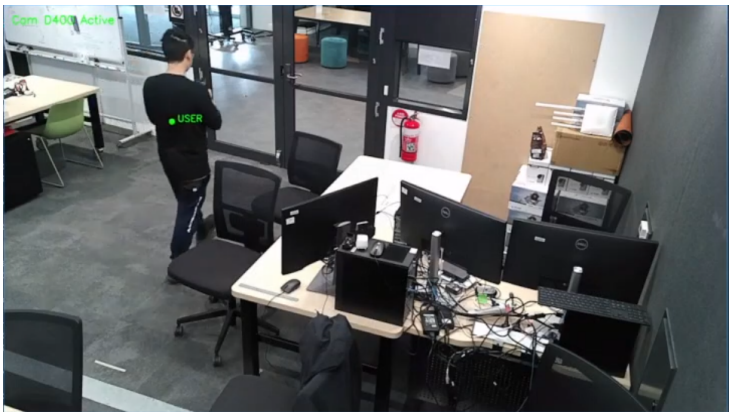


Figure 12: Demonstration of torso tracking. The system maintains a successful spatial lock ("USER") even when the camera is positioned behind the user, a scenario that completely occludes the smartphone.

Additionally, Figure 13 demonstrates the robustness of the torso's visual mass at extreme close ranges. Even when the user is very close to the camera, and their full profile is clipped by the frame boundaries, the torso provides sufficient features for the system to reliably extract a central coordinate (identified here as a detected but logically "IGNORED" coordinate because the phone is not active).

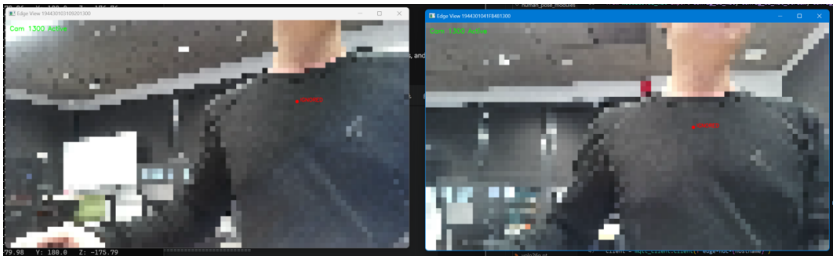


Figure 13: Demonstration of torso tracking robustness under close-range constraints. The system successfully identifies the torso centroid even when partially out of frame.

Because of these empirical observations, the system was pivoted to torso tracking. By bounding the region from the shoulders to the waist, the system tracks a significantly larger feature, ensuring consistent and reliable spatial extraction across the network.

B Mathematical Formulation of the Central Hub Tracking Logic

To ensure the mobile client receives smooth, highly reliable spatial corrections, the Central Consolidation Hub processes raw infrastructure detections through a custom, 6-state kinematic Kalman Filter. Standard Kalman implementations assume continuous observation; however, the sparse nature of the camera network necessitates specific mathematical modifications to safely handle user transit through unmonitored zones.

1. State and Measurement Models

The state vector x_k at time step k tracks both 3D position and 3D velocity to enable predictive tracking:

$$x_k = \begin{bmatrix} X & Y & Z & V_x & V_y & V_z \end{bmatrix}^T$$

The state transition matrix F projects the current state forward over the time interval Δt using standard Newtonian kinematics:

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Because the YOLO26 edge inference pipeline outputs discrete 3D spatial coordinates without natively calculating velocity, the measurement vector z_k strictly contains positional observations. The measurement matrix H maps the 6-state space to the 3-measurement space:

$$z_k = \begin{bmatrix} X_{obs} \\ Y_{obs} \\ Z_{obs} \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

2. Modification: Blind-Spot Velocity Decay

Due to the sparse deployment of the infrastructure cameras, the system inherently experiences brief visual blind spots between node coverage zones. When the user exits a camera's field of view, the system triggers a blind-predictive state (up to a maximum threshold of 5.0 seconds).

During this blind state, a standard kinematic prediction step would sustain the user's last known velocity indefinitely, resulting in severe positional overshoot if the user physically stopped walking within a blind spot. To mitigate this hazard, a heuristic velocity decay constant ($\gamma = 0.85$) is injected directly into the post-prediction state matrix during every iteration where camera observation is lost:

$$V_{x,k|k-1} = \gamma V_{x,k-1|k-1}$$

$$V_{y,k|k-1} = \gamma V_{y,k-1|k-1}$$

$$V_{z,k|k-1} = \gamma V_{z,k-1|k-1}$$

This architectural modification acts as an artificial spatial drag. It smoothly decelerates the filter's predicted tracking coordinate to a safe halt during extended signal loss, ensuring that the AR trajectory remains physically constrained within the navigable corridor bounds until absolute infrastructure tracking is reacquired.

C Infrastructure Hardware and Extrinsic Calibration Methodology

To achieve accurate infrastructure-assisted localisation, the system requires precise spatial alignment between the physical camera sensors and the digital representation of the Monash Innovation Lab. The deployment utilised OAK-D S2 stereo-vision cameras, selected for their integrated depth perception and wide field of view. Figure 14 provides a visual reference of the physical hardware as deployed within the testing environment.

Figure 14: Physical deployment of the OAK-D S2 stereo-vision camera mounted within the Monash Innovation Lab. This hardware formed the visual foundation of the distributed edge network.

Extrinsic Coordinate Transformation

Before the YOLO26 pedestrian detections could be utilised by the mobile AR client, the local coordinate space of each individual camera (X_c, Y_c, Z_c) had to be mathematically mapped to the global coordinate space of the virtual Unity environment (X_g, Y_g, Z_g). Because the system relies on a static infrastructure, this was achieved through a manual extrinsic calibration process.

First, rather than relying on arbitrary physical floor markings, a structural corner of the Monash Innovation Lab was designated as the primary global origin $(0, 0, 0)$ based on an existing architectural floor map. This physical building corner was then aligned with the $(0, 0, 0)$ coordinate of the 3D virtual model within the Unity engine. For each installed camera, the translational offsets (t_x, t_y, t_z) were manually measured relative to this structural origin. By utilising the building's orthogonal walls as fixed measurement baselines, the physical camera positions were mapped onto the digital floor plan to the highest precision achievable through manual measurement techniques.

Following translation, the rotational offsets (pitch, yaw, and roll) were calculated to determine the camera's exact viewing angle relative to the room's physical X and Z axes. These static translational and rotational values were combined into an extrinsic transformation matrix and hardcoded into the configuration file of each respective Intel NUC edge node.

During live execution, when the OAK-D camera extracts a 3D bounding box centroid in its own local camera space, the Intel NUC immediately multiplies this local coordinate by the extrinsic transformation matrix. This ensures that every coordinate published to the MQTT broker by the edge nodes is already mapped to the absolute global coordinate system, allowing the Unity client to seamlessly inject the data without requiring complex local coordinate math on the mobile device.

D Visual Documentation of Critical Spatial Failures

As defined in Section 4.5.2, a critical spatial failure occurs when uncorrected Visual-Inertial Odometry (VIO) drift causes the virtual navigational path to diverge into physical architectural boundaries. To ensure maximum methodological rigour, a zero-tolerance boundary heuristic was applied during the post hoc video coding of all experimental runs.

Figure 15 provides first-person visual examples of direct routing failures captured during the unassisted Baseline trials. In these instances, accumulated translational drift caused the virtual navigational line to route the user directly into impassable physical structures.

Furthermore, Figure 16 illustrates the strict application of the zero-tolerance boundary heuristic. Because an AR navigation system must guarantee fundamental spatial safety, a path is only considered viable if it maintains complete clearance from permanent architecture. Any degree of architectural clipping is logged as an objective environmental collision.

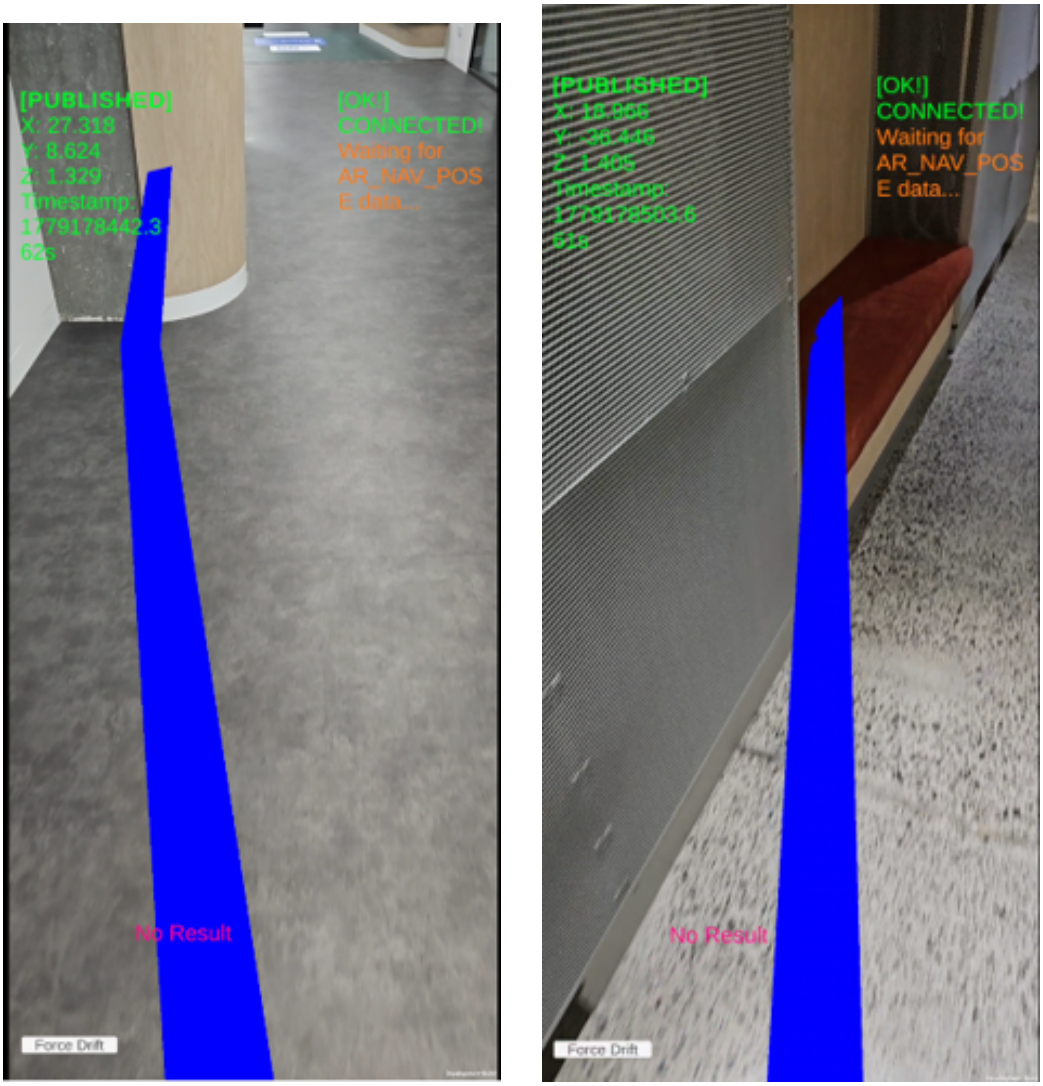


Figure 15: Demonstration of direct critical spatial failures. In both instances, the AR navigational path routes directly into and through the structural walls, directing the user into impassable boundaries.

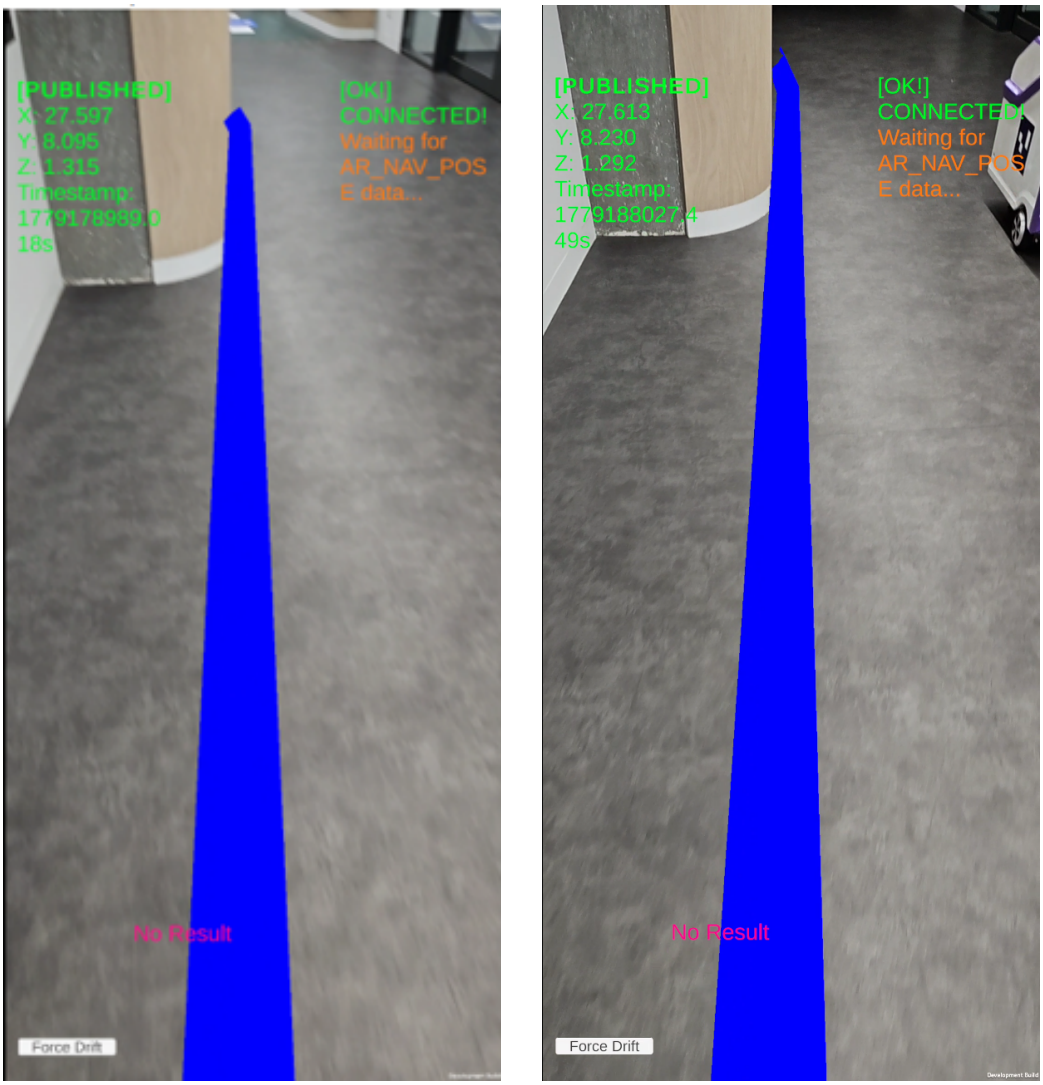


Figure 16: Visualisation of the zero-tolerance failure threshold. (Left) The navigational line grazes the physical wall. Despite being a minor intersection, this is logged as a critical spatial failure. (Right) A valid navigational path that passes extremely close to the structural pillar but maintains complete visual clearance, resulting in a successful trajectory segment.

E Outlier Analysis and Data Pre-processing Impact

As detailed in Section 4.5.5, an Interquartile Range (IQR) filter ($1.5 \times IQR$) was applied to the raw Final Drift Error (FDE) data prior to the primary statistical analysis. This data pre-processing step was strictly implemented to isolate the core algorithmic tracking performance from extreme, random experimental anomalies (e.g., physical user tripping or momentary hardware stalling).

To ensure complete methodological transparency, Figure 17 visualises the raw, unfiltered distributions of the FDE datasets across all three structural routes. The boxplots clearly illustrate the presence of extreme upper-bound outliers within the unassisted Baseline runs, particularly during the extended Path 3 trajectory, where catastrophic collisions occurred.

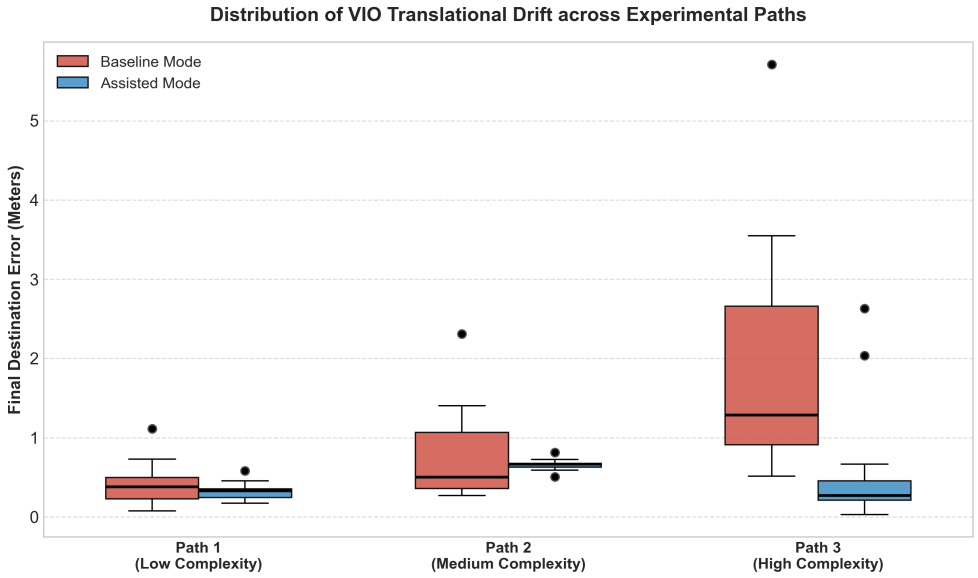


Figure 17: Boxplot distribution of the raw Final Drift Error (FDE) datasets prior to IQR filtering. Black markers indicate extreme anomalies representing critical tracking failures or experimental interruptions.

Furthermore, Table 5 provides a quantitative breakdown of the calculated mean trajectory improvement before and after the application of the IQR filter. The data demonstrates that while the Baseline variance was significantly

tightened by removing the extreme anomalies (narrowing the Confidence Intervals), the core statistical narrative remains consistent regardless of the outliers' inclusion.

Table 5: Statistical Impact of Outlier Exclusion on Mean Trajectory Improvement

Navigation Route	**Raw Data (With Outliers)**		**Filtered Data (Outliers Removed)**	
	Mean Improvement	Bootstrap 95% CI	Mean Improvement	Bootstrap 95% CI
Path 1 (Short)	0.075 m	[-0.056, 0.226]	0.043 m	[-0.059, 0.147]
Path 2 (Loop)	0.117 m	[-0.141, 0.415]	0.008 m	[-0.183, 0.215]
Path 3 (Full)	1.303 m	[0.561, 2.123]	1.301 m	[0.825, 1.842]

F Expanded Continuous Trajectory Deviation Analysis

While Section 6.2 presents the aggregated mean spatial deviation across the Path 3 trajectory lifecycle, understanding the temporal behaviour of individual experimental runs provides deeper insight into the system's correction mechanics.

Figure 18 visualises the raw Dynamic Time Warping (DTW) spatial deviation for all individual Path 3 trials, plotted directly against the tracking frame index. This time-series representation explicitly highlights the aggressive, near-linear accumulation of translational drift experienced by the unassisted Baseline VIO pipeline (warm colours).

Conversely, the Assisted architecture runs (cool colours) demonstrate the periodic correction pattern characteristic of the infrastructure checkpoints. As the user walks, minor VIO drift naturally accumulates; however, upon entering a camera's field of view, the error is repeatedly suppressed back to a near-zero baseline via the MQTT coordinate payload injection, effectively preventing catastrophic navigational failure.

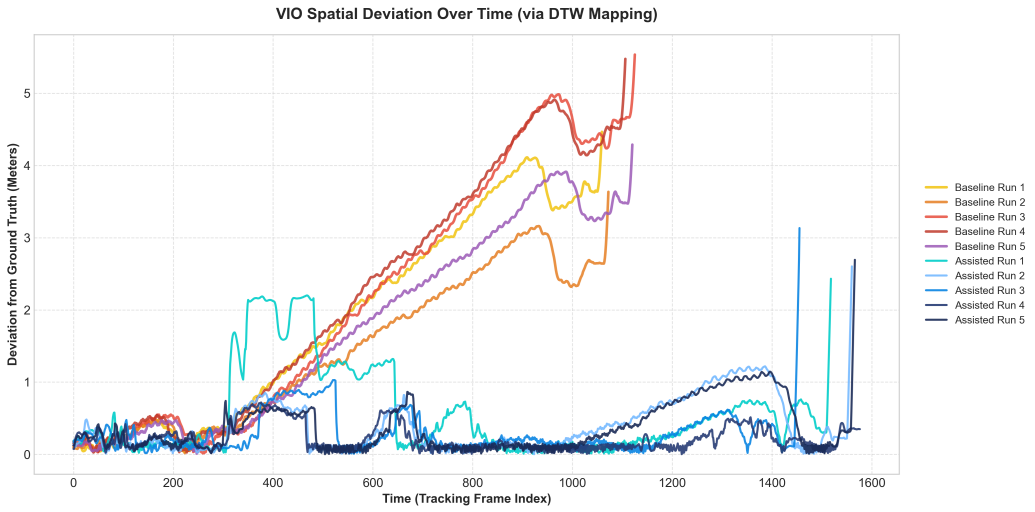


Figure 18: Time-series plot of VIO spatial deviation (measured via DTW) for individual experimental runs on Path 3. The sharp vertical drops in the Assisted runs (cool colours) represent the exact moments of spatial correction, successfully eliminating the compounding drift seen in the Baseline runs (warm colours).