

Visible Light Positioning for Robotic Navigation

Tyrel Glass

Supervisors: Fakhrul Alam

Daniel Konings

**School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology**

**A thesis submitted to Auckland University of Technology
in fulfilment of the requirements for the degree of
*Doctor of Philosophy***

August 2026

Dedicated to my loving wife.

Acknowledgments

Many people have helped me along this path. I ask their forgiveness for their omission from this acknowledgment, because in trying to say everything, I would likely succeed in saying nothing. I wish to focus on the outsized impact that my supervisor, Professor Fakhrul Alam, has had not only on my doctoral journey, but on my life as a whole.

I arrived at Massey University in 2015 after a turbulent period, having struggled with drugs and alcohol alongside mental health complications that led to hospitalizations, followed by a 14-month stay in residential rehabilitation shortly before beginning an engineering degree at Massey. Looking back ten years on, I can see how fragile that period was. Several people who shared my recovery journey are no longer here.

Fakhrul has the rare ability to be direct to the point of challenge while also being genuinely caring and compassionate. In my view, these qualities are why he is one of the best educators I have had the privilege to work with.

He can quickly give a full and frank appraisal of deficiencies in my knowledge or work. Rather than leaving me discouraged, he highlights the next steps and gets me moving. He also has a genuine passion for tackling complex topics and deconstructing them to the point where anyone with a genuine desire to learn can understand.

Fakhrul's genuine dedication to teaching led to the formation of a research group made up of some extremely talented students. I had the good fortune to be folded into that mix, working on projects he led. Looking back, the accomplishments of that small group were exceptional.

Throughout my undergraduate journey, I had frequent and meaningful contact with Fakhrul, and he was an exceptional mentor. For anyone who has struggled to find their bearings in life, it is hard to overstate the importance of someone who is always willing to tell the truth. In a role like Fakhrul's, expedience is constantly competing with truth, and choosing truth often requires sacrifice. Although I was not fully conscious of it at the time, I can now see that his influence contributed in no small way to the success of my recovery.

The path from finishing my undergraduate degree to completing this thesis has also been long and winding. In that time, life has included many major events: becoming a sailor, a startup founder, a husband, a homeowner, a father, and an elected councillor. It amazes me

that through all of this, and despite a change of university, Fakhrul has guided this process to completion. He has shown an almost providential sense of timing, easing off when things became too much, then ramping up to capitalize when capacity returned, always expressing optimism that it would be finished while applying a prudent amount of pressure.

The ease with which Fakhrul carries his role has allowed our relationship to grow gradually and naturally into one marked by collegiality, and even friendship. I will always be grateful for the outsized impact he has had on my life.

Abstract

Indoor positioning remains a persistent challenge where Global Positioning System (GPS) signals are unreliable. Visible Light Positioning (VLP) has emerged as a promising solution, leveraging the ubiquity of Light Emitting Diode (LED) lighting and low-cost photodiode receivers to provide centimeter-level accuracy. Despite its potential, two key barriers have limited progress: the absence of large-scale, high-quality datasets for algorithm development and evaluation, and the lack of evidence demonstrating that VLP can support practical robotic navigation.

This thesis addresses both challenges through a set of proposed frameworks supported by experimental investigation. The first study developed an autonomous ground-truthing and dataset collection system that co-locates a photodiode receiver with an HTC Vive tracker on a mobile robotic platform. This system enabled efficient acquisition of 7,344 fingerprint points within a controlled testbed, far exceeding the photodiode-based VLP RSS fingerprint datasets identified in the surveyed literature at the time of publication. Using this dataset, a comprehensive benchmarking of machine learning and channel model approaches for VLP was performed. Results showed that model-based methods retain an advantage when calibration data is scarce, but that machine learning techniques surpass them once trained on sufficiently large datasets. These findings resolved inconsistencies in the literature and highlighted the central role of data scale in VLP performance.

The second study demonstrated single-mode VLP navigation on a differential-drive robot under robust trajectory-level testing, then extended the investigation by combining VLP with odometry through a Long Short-Term Memory (LSTM)-based fusion model. Experimental evaluation showed that the fused approach maintains stable trajectories by learning sequential dependencies in the data. This demonstrated that VLP, when fused with odometry, can provide reliable localization and heading estimation for real robots operating in motion.

Together, these studies advance VLP research from controlled, small-scale laboratory investigations toward practical deployment in robotics. The thesis makes five key contributions: (i) a novel, automated ground-truth and data collection system for VLP, producing the largest experimental RSS fingerprint dataset for photodiode-based VLP identified in the surveyed literature at the time of publication, (ii) robust benchmarking of machine learning

and model-based positioning methods, (iii) a real-world robotic system enabling photodiode-based VLP navigation with full pose estimation under ground truth, (iv) a comparative study of single-mode baselines versus LSTM-based fusion across offline and live navigation experiments, and (v) advancement of VLP from theoretical exploration toward practical deployment in robotic navigation.

While limitations remain, including sensitivity to orientation and tilt, confinement to controlled environments, and reliance on research-grade ground-truth systems, these highlight clear avenues for future work. Potential directions include benchmarking learning-based fusion against model-based filters (e.g., Extended Kalman Filter (EKF) variants), multi-photodiode or hybrid VLP–Inertial Measurement Unit (IMU) receivers, validation in large-scale industrial environments, extension to diverse robotic platforms, and integration with other positioning technologies.

In conclusion, this thesis demonstrates that VLP is a viable localization modality for robotic navigation. By overcoming barriers of dataset scale and by validating fusion-based navigation, the research provides a foundation for deploying VLP in real-world robotic and smart-building applications.

Publications

Glass, T., Alam, F., Legg, M., & Noble, F. (2021). Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning. *Sensors*, 21(9), 3256.
<https://doi.org/10.3390/s21093256>

Glass, T., Konings, D., Legg, M., & Alam, F. Fusion of Visible Light Positioning and Odometry for Robot Navigation. Under review at *IEEE Sensors Journal*.

STUDENT AND SUPERVISOR APPROVALS

By signing you are confirming that the co-author contributions stated in the table(s) below are accurate.

Student Name	Tyrel Glass	Signature	Date	25/02/2026
Supervisor Name	Fakhrul Alam	Signature	Date	25/02/2026

Chapter Number:	3
Manuscript Title:	Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning
Publication Status:	Published
Reference if published:	T. Glass, F. Alam, M. Legg, and F. Noble, "Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning," Sensors, 21(9), 3256, May 2021.
AUTHOR SURNAME: (order as per manuscript)	CONTRIBUTION (May copy from the guidelines above)
Glass	80% (Conception and design of the project and output Acquisition of research data Analysis and interpretation of research data Drafting significant parts of the output Revising output)
Alam	10% (Conception and design of the project and output Interpretation of research data Revising output)
Legg	6 % (Interpretation of research data Revising output)
Noble	4% (Revising output)

Chapter Number:	4
Manuscript Title:	Fusion of Visible Light Positioning and Odometry for Robot Navigation
Publication Status:	Submitted for Publication
Reference if published:	
AUTHOR SURNAME: (order as per manuscript)	CONTRIBUTION (May copy from the guidelines above)
Glass	80% (Conception and design of the project and output Acquisition of research data Analysis and interpretation of research data Drafting significant parts of the output Revising output)
Konings	8% (Acquisition of research data Revising output)
Legg	4 % (Revising output)
Alam	8% (Conception and design of the project and output Revising output)

Table of contents

Acknowledgments	iii
Abstract	v
Publications	vii
Attestation of Authorship	xiii
Statement on Use of Artificial Intelligence Tools	xiv
List of figures	xv
List of tables	xxi
List of Abbreviations	xxiii
1 Introduction	1
1.1 Background	1
1.2 Preview of Literature	1
1.3 Scope and Assumptions	2
1.4 Research Problems	2
1.5 Research Questions	3
1.6 Research Contributions	3
1.7 Research Methodology	4
1.8 Thesis Structure	4
2 Literature Review	7
2.1 Introduction	7
2.2 Foundations and Theoretical Background	8
2.3 Photodiode-Based VLP	9

2.3.1	Photodiode Receivers and Measurement Modalities	9
2.3.2	RSS Extraction, Channel Models, and Fingerprinting	9
2.3.3	Receiver Orientation and Multipath Effects	9
2.3.4	Calibration and Deployment at Scale	10
2.4	Machine Learning and Fingerprinting Approaches	10
2.4.1	Machine Learning Foundations for VLP	11
2.5	Robotic Navigation and Odometry	13
2.6	VLP for Robotic Navigation	13
2.6.1	Camera-Based and Cooperative Navigation	14
2.6.2	Photodiode-Only Navigation	14
2.7	Sensor Fusion Approaches	15
2.7.1	Summary of Fusion Studies Relevant to Chapter 4	16
2.8	Recent Advances in Dataset Collection and Robotic Applications	18
2.9	Summary and Positioning of This Thesis	18
3	Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning	21
3.1	Introduction	22
3.2	System Overview	26
3.2.1	Experimental Setup	26
3.2.2	VLP System	26
3.2.3	HTC Vive Tracking for Accurate Ground Truth	27
3.2.4	Vive Performance Analysis	30
3.2.5	Roborock S50 Robotic Platform	34
3.3	RSS Data	39
3.3.1	Experimental Setup and Data Collection	39
3.3.2	VLP Data Preprocessing	40
3.3.3	Lambertian Channel Model Calibration	43
3.4	Machine Learning Models for Positioning	45
3.5	Channel Model Based Positioning	47
3.5.1	Ranging	49
3.5.2	Positioning	49
3.6	Importance of Data Size	53
3.6.1	Training	53
3.6.2	Testing	53
3.7	Conclusions and Future Work	58

4	Fusion of Visible Light Positioning and Odometry for Robot Navigation	60
4.1	Introduction	61
4.2	Related Work	63
4.2.1	Visible Light Positioning (VLP)	63
4.2.2	Odometry and Dead Reckoning	63
4.2.3	Sensor Fusion for Localization	64
4.2.4	Robustness to Perturbations	64
4.3	Experimental Setup	64
4.3.1	Navigation Environment	64
4.3.2	Mobile Robot Design	65
4.3.3	VLP System Design and Calibration	69
4.3.4	Ground Truth Tracking with HTC Vive	71
4.4	Training dataset construction and Encoder Calibration	72
4.4.1	Overview	72
4.4.2	Encoder Calibration	74
4.4.3	Dataset structure	76
4.5	Single Mode Navigation Models	78
4.5.1	Encoder Odometry	78
4.5.2	VLP	79
4.6	LSTM Fusion Model	79
4.7	Navigation Performance	82
4.7.1	Temporal Error Analysis	83
4.7.2	Quantitative Error Analysis	84
4.7.3	Qualitative Path Analysis	85
4.7.4	Model-Wide Comparison	86
4.8	Captured Robot Experiments	91
4.9	Edge Implementation Considerations	98
4.10	Conclusion and Future Works	98
4.10.1	Summary of Contributions	99
4.10.2	Key Findings	99
4.10.3	Limitations and Future Directions	100
5	Conclusions and Future Works	102
5.1	Introduction	102
5.2	Synthesis of Findings	102
5.2.1	From Chapter 3: Automated Ground Truth and Benchmarking	102
5.2.2	From Chapter 4: VLP–Odometry Fusion for Navigation	103

5.2.3	Cross-Cutting Themes	103
5.2.4	Research Question Answers	107
5.3	Contributions to Knowledge	108
5.4	Limitations	108
5.5	Future Work	110
5.5.1	Technical Extensions	110
5.5.2	Algorithmic Extensions	111
5.5.3	Deployment Pathways	111
5.5.4	Summary	112
5.6	Closing Remarks	113
References		114

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor used artificial intelligence tools or generative artificial intelligence tools (unless it is clearly stated, and referenced, along with the purpose of use), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Tyrel Glass
August 2026

Statement on Use of Artificial Intelligence Tools

This thesis spans approximately six years. The first paper (2021) was completed before generative AI tools were widely available and therefore did not involve their use.

For the second paper and thesis preparation, I used AI tools (primarily ChatGPT) for defined support tasks, including language refinement (grammar, clarity, and structure suggestions), coding assistance (debugging support, code scaffolding, and library syntax lookup), and manuscript production tasks (LaTeX formatting suggestions and workflow support for regenerating figures and tables from updated analysis code).

All substantive research decisions, methods, analyses, interpretations, citations, and conclusions were made by me. AI outputs were treated as draft assistance only and were critically reviewed, edited, and verified before inclusion. No AI tool was used to make independent academic judgments. The intellectual content and final submitted work remain my own.

List of figures

3.1	Office lobby used for the experiment. Two Vive base stations provide accurate positioning for the Vive tracker, which is co-located with the VLP receiver on a mobile robot vacuum.	26
3.2	FDM scheme demonstrated for three luminaires. Each luminaire is driven by a direct-current (DC) signal overlaid with a sinusoid of unique frequency. A photodiode detector receives a signal that is the sum of multiple luminaire outputs scaled by distance. This signal is then converted with a Fast Fourier Transform (FFT), and the component at each luminaire's modulating frequency is used as the RSS.	27
3.3	The IR signals from the Vive base stations caused interference with the VLP receivers as shown in (a) . The Vive signal added substantial noise at frequencies lower than 4 kHz and altered the intensities of the luminaire peaks. To prevent interference, an IR-rejecting photodiode was used and then placed behind IR-rejecting glass (b) . The combination of the two eliminated the interference.	28
3.4	Using Z-X-Z Euler angles to derive an initial alignment between the Vive coordinate reference frame $(\hat{x}_v, \hat{y}_v, \hat{z}_v)$ and the room coordinate frame $(\hat{x}_r, \hat{y}_r, \hat{z}_r)$. A translation aligns the origins (a) ; then, $\hat{z}_v$ is aligned with the $\hat{z}_r, \hat{y}_r$ plane via rotation around $\hat{z}_r$ (b) . This allows a rotation around $\hat{x}_r$ to align $\hat{z}_r$ with $\hat{z}_v$ (c) . Finally, another rotation around $\hat{z}_r$ (d) completes the alignment (e) . . .	31
3.5	CDF for Vive jitter at different locations and orientations. The two runs are very similar indicating that there is no spatial dependence.	33
3.6	Determination of the Vive accuracy. The tracker was moved to 625 locations (a) by the CNC machine. The high accuracy of the CNC allows the Vive errors to be accurately calculated. The error CDF is shown in (b) ; median error is less than 4.5 mm.	33

3.7	Dependence of transform accuracy on the number of calibration points. As more points are used, accuracy improves. The dashed gray line represents the ideal transform calculated using all 625 points. Using randomly chosen calibration points produces inconsistent results.	35
3.8	Selecting calibration points to maximise area covered. If points are chosen by following the pattern in (a), then the transform accuracy converges much faster, as can be seen in (b). With only 3 calibration points, the mean error is below 5 mm (shown in (b)).	35
3.9	The Vive tracker was mounted to the top of the Roborock S5 with care taken to avoid blocking the LiDAR. The VLP tag was mounted in the center of the tracker without blocking the Vive's photodiodes. Both the Vive tracker and the VLP tag were powered using a USB power bank.	36
3.10	Visualization of noise caused by the EMI of the vacuum's motors. The green plot shows a discrete Fourier transform (DFT) of the VLP signal when the motors are off and luminaires are operating. The distinct peaks represent signals detected from the luminaires in proximity. The red plot shows the DFT of the received signal at the same location but with the vacuum motors on. In addition to a large peak at around 12.5 kHz, the background noise level has increased over the entire frequency range.	37
3.11	Setting up the cleaning zone using the Xiaomi Home mobile app. When placed in a new environment, the vacuum will build a map of its environment as shown in (a); the triangle in the room's center is the cardboard defectors placed on the Vive base station stands (Fig. 3.1). Within the app, walls and zones can be placed that will restrict the movement of the robot (b). Once initialized, the robot will then traverse the perimeter (c) before sweeping back and forth to cover the entire area (d).	38
3.12	The area of the lobby was too large to be covered by the Vive 1.0 system, which necessitated splitting the environment into 4 quadrants. Each quadrant required a separate transformation to align the Vive readings with the room's coordinate reference frame. Three quadrants are shown in (a), (b), and (c), respectively.	40
3.13	Calibration points used for deriving the transformations. Twenty-six reference points were carefully mapped out using a laser level and tape measure in such a way that each quadrant would have 10 reference calibration points, sharing four with each neighbour. This reduced the overall number of calibration points required.	41

3.14	RSS plots for luminaires 6, 7, 1, and 4 in (a), (b), (c) and (d) respectively. The signal is largest when the sensor is directly beneath and falls off with distance. No data were collected at the location of the pillar, which can be seen clearly in (d)	42
3.15	Background VLP noise with no active luminaires. One hundred eighty DFTs were computed at 10 s intervals and for each frequency the average was calculated and plotted (green); 1 and 2 standard deviations above the mean level were also calculated to provide an indication of the noise floor.	43
3.16	When both the luminaire and receiver are horizontal, θ_n and φ_n are equal, and d_n then forms the hypotenuse of a right-angle triangle.	44
3.17	RSS vs. distance plots for luminaires 6, 7, 1, and 4 in (a), (b), (c), and (d), respectively. The red lines show the trained Lambertian models. Luminaire 6 in (a) shows clear deviation due to the presence of the pillar; some effects can be seen in luminaires 1 and 7 (in (c) and (b)), although not as severe. Luminaire 7 is located close to the center of the room and therefore does not have as wide a range of distances compared to the others. Clamping of the RSS values can also be clearly seen in all plots, as the distance increases and readings do not follow the Lambertian model (which approaches zero).	46
3.18	The dataset is initially split into 80% training and 20% testing. The ML hyperparameters are then tuned using 5-fold cross-validation of the 80% training data by splitting into 5 folds and iterating 5 times through the folds, each time selecting 4 folds for training and the remaining for validation. The resulting five scores are then averaged to compute the score for a set of hyperparameters. Once tuned, the ML models are retrained using all of the 80% training data and assessed for final performance on the 20% test set.	47
3.19	Mean and CDF for errors of tuned models trained on the entire 80% training set. Errors were evaluated using the 20% test set, and it can be seen, in both (a) and (b) , that ridge regression has the best performance, with RF being the worst.	49
3.20	Actual distances to luminaire vs. distances expected based on the RSS model for luminaires 4 and 6. Both luminaires demonstrate linear behavior up until the photodiodes begin to reach their FOV (gray dashed line). The pillar near luminaire 6 causes a cluster of points that clearly violate the model (a) , where the predicted distance is greater than the actual distance. The predicted distance for each luminaire will be limited due to the clamping of RSS (an example for luminaire 4 in (b)).	50

3.21	CDF and mean of localization errors for channel model-based positioning are benchmarked in (a) and (b), respectively. Spring relaxation was the better performing of the model-based techniques, although its performance was slightly lower than that of RF, the worst-performing ML model. The least-squares method was considerably worse than any of the other methods. The models were trained/calibrated on the entire 80% training set and tested on the 20% test set.	52
3.22	The number of training points used has a large impact on which model has the best performance. Despite being the best performing model when trained on the entire dataset, ridge regression requires a significant amount of training data. In this study, ridge regression first became the best-performing model at around 1580 training points.	54
3.23	For very small training sets, approximately fewer than 33 points in this study, the spring-relaxation model-based approach outperformed the machine-learning methods.	55
3.24	There is a clear difference in the performance of the RF and ridge models, shown by 0 not being contained in the 95% confidence interval (a) and the p -value being clearly below the 0.05 significance level (b).	56
3.25	When comparing SVR and polynomial regression, the performance of the models is more similar. The confidence interval (a) includes 0 a for a large proportion of the data, and the p -value (b) only drops below the 0.05 significance level for a brief period with very few samples.	56
3.26	The t-test between the k-NN and MLP models does indicate a performance difference but only after more than 600 test points have been assessed. This demonstrates the need for a large dataset when making robust comparisons between models. If less than 600 points had been used for testing, then the confidence interval (a) would include 0, the p -value (b) would be larger than 0.05, and the hypothesis that there was no difference in performance would not be rejected.	57
4.1	Experimental setup overview for data collection and navigation experiments; all components are labeled in the figure.	65
4.2	Geometric layout of the workspace, including LED positions, Vive Tracker calibration points, and the CNC/robot operating regions (training, live, and capture areas).	66
4.3	Robot platform with labeled components and a differential-drive layout (two powered wheels with a passive caster).	68

4.4	VLP modulation and RSS extraction: each LED is driven at a distinct square-wave carrier, and the receiver separates these in the FFT to recover per-LED RSS (used for distance inference).	70
4.5	CNC-mounted sensors used to co-locate the VLP receiver and Vive tracker on the same reference point for calibration.	72
4.6	Navigation logic for data collection: the robot executes heading corrections and fixed-length steps toward each target; dashed circles show the distance threshold for declaring a target reached.	75
4.7	Overview of how the models are trained and evaluated, including the offline split and the live-path evaluation protocol.	77
4.8	Temporal error traces over 300 robot steps for LSTM fusion and single-mode baselines. Columns show position (left) and heading (right) errors, and rows correspond to the offline/test split, square, hourglass, and random paths. . .	84
4.9	Square-path trajectories showing drift over time: the top row is early in the run and the bottom row is near the end. The LSTM remains stable, encoder odometry rotates nearly 90° , and VLP is noisy but stable.	86
4.10	Hourglass-path trajectories over time (top row early, bottom row late). The LSTM is the most consistent, encoder odometry degrades further, and VLP is noisy but stable.	87
4.11	CDFs of position (top row) and heading (bottom row) errors for the last 100 steps (late-run performance). Columns correspond to navigation models (LSTM, VLP, Enc. Odom.), and within each panel the curves represent the training, square, hourglass, and random trajectories. The LSTM is best overall but shows its strongest heading performance on the training split; VLP is relatively consistent across paths; and encoder odometry heading errors on the square path fall outside the plotted range.	89
4.12	Translation-only capture error traces. The vertical dashed line marks the perturbation; position error spikes while heading remains low and stable. The fusion model corrects rapidly.	92
4.13	Translation-only capture trajectories. The LSTM navigation diverges from the nominal square, indicating a rapid response to VLP updates. Target points and threshold circles are shown for reference.	93
4.14	Rotation-only capture error traces. Heading errors jump immediately; position errors rise as expected. The fusion model recovers after the perturbation.	94
4.15	Rotation-only capture trajectories. The fusion model responds quickly to VLP updates and realigns with the intended path.	95

-
- 4.16 Combined translation-and-rotation error traces. Heading and position errors jump simultaneously; the fusion model recovers over subsequent steps. . . . 96
- 4.17 Combined translation-and-rotation capture trajectories. The fusion model shows rapid correction to the VLP shift under the compounded perturbation. 97

List of tables

1.1	Problem–Research Question (RQ)–Evidence mapping.	5
2.1	Representative VLP fusion studies relevant to PD-based robotics.	17
2.2	Representative dataset and ground-truth pipelines aligned with the objectives of Chapter 3.	19
3.1	VLP articles listed by size of the experimental dataset. Most articles use a limited number of data points and do not describe how ground truth was recorded.	24
3.2	Variation in Vive measurement in mm for a stationary target. Each run was conducted with a different tracker position and orientation.	32
3.3	Results of the hyperparameter tuning. The listed parameters for each of the models were tuned using two rounds, the first being successive powers of 10, the second being powers of 2. The scoring metric for grid search was RMSE.	48
3.4	Accuracy of the ML in mm after being retrained on the entire 80% training set and tested on the 20% test set.	48
3.5	Accuracy of the model-based methods after being retrained on the entire 80% training set and tested on the 20% test set. The RMSE for both is greater than the 90 th percentile errors, indicating that there are some large errors. All values are in mm.	51
4.1	Summary of input features and ground-truth targets used for supervised navigation model training, including the commanded heading change and step size s_e (robot step length) alongside VLP position measurements. . . .	78
4.2	Expanded input features for the LSTM, including encoder odometry estimates, VLP-derived heading, and heading-change terms from both VLP and encoder odometry; step size is omitted because it is constant.	80

4.3	Median and 90th percentile (P_{90}) heading and position errors for the first (steps 1–100) and last (steps 200–300) segments across all scenarios. The table is intended to highlight typical error and practically relevant upper-tail error as drift develops within each run, rather than to report mean-centered variability measures. Boldface indicates the lowest (best) error value in each scenario.	90
4.4	Performance and memory comparison across development boards.	98
5.1	Quantitative recap of the two studies.	104

List of Abbreviations

AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
AOA	Angle of Arrival
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BLE	Bluetooth Low Energy
CDF	Cumulative Distribution Function
CNC	Computer Numerical Control
CPU	Central Processing Unit
CSV	Comma-Separated Values
DC	Direct Current
DFT	Discrete Fourier Transform
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
EMI	Electromagnetic Interference
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FOV	Field of View
GAT	Graph Attention Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
IR	Infrared
IPS	Indoor Positioning System
k-NN	k-Nearest Neighbors
LED	Light Emitting Diode
LiDAR	Light Detection and Ranging
LOS	Line of Sight
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
ONNX	Open Neural Network Exchange

PCA	Principal Component Analysis
PC	Personal Computer
PD	Photodiode
PDR	Pedestrian Dead Reckoning
RF	Radio Frequency
RMSE	Root Mean Square Error
ROS	Robot Operating System
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
STM32	STMicroelectronics 32-bit Microcontroller Family
SVR	Support Vector Regression
TCN	Temporal Convolutional Network
TOA	Time of Arrival
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
USB	Universal Serial Bus
UWB	Ultra-Wideband
VLC	Visible Light Communication
VLP	Visible Light Positioning
VO	Visual Odometry
VR	Virtual Reality

Chapter 1

Introduction

1.1 Background

Reliable indoor positioning remains an open research challenge. Although the Global Positioning System (GPS) has revolutionized outdoor navigation, it performs poorly indoors due to signal attenuation and multipath effects. This has motivated extensive exploration of alternative Indoor Positioning Systems (IPS) such as Wi-Fi, Bluetooth, Ultra-Wideband (UWB), infrared, ultrasound, and computer vision [1]. However, each of these approaches faces trade-offs in cost, accuracy, scalability, and ease of deployment.

Visible Light Positioning (VLP) has emerged as a promising solution due to the ubiquity of Light Emitting Diode (LED) lighting, low interference, and the potential for centimeter-level accuracy [2, 3]. Photodiode-based Received Signal Strength (RSS) methods remain attractive for low-cost tags and practical deployment [4, 5]. Despite rapid progress, challenges remain around orientation sensitivity, data collection at scale, robustness in real-world environments, and integration with practical robotic navigation. In mobile robotics, localization and navigation depend on multi-sensor fusion to mitigate odometry drift and sensor limitations [6].

1.2 Preview of Literature

A detailed review of related work is presented in Chapter 2, which critically examines prior research in VLP and indoor positioning. Here, we briefly highlight that while there has been a significant body of research, most studies have been limited to small experimental datasets and simplified laboratory environments. Complete pose recovery using single-mode VLP remains sparsely demonstrated, especially for low-cost photodiode-based systems, and robust

closed-loop navigation is rare [7, 8]. Furthermore, work that combines photodiode-based VLP with other sensors, particularly wheel odometry, remains limited, and long-horizon, closed-loop evaluations where localization estimates directly drive motion updates are still scarce [9, 10]. Camera-based VLP has been widely explored [11, 12]. However, image-sensor receivers typically imply higher data throughput and processing overhead than photodiode-based systems, which can be challenging on cost- and power-constrained platforms [13]. This thesis addresses these gaps by developing scalable methods for VLP data collection, benchmarking machine learning and model-based approaches, and demonstrating practical integration of VLP into robot navigation, with a focus on photodiode-based approaches.

1.3 Scope and Assumptions

This thesis focuses on indoor, photodiode-based VLP using RSS measurements from LED lighting infrastructure, with an emphasis on low-cost robot navigation. The scope includes data collection and ground truthing, model-based and learning-based positioning, and fusion with odometry on a differential-drive robot platform. The work does not treat camera-based VLP as a primary receiver, nor does it target very large industrial deployments, multi-floor navigation, or outdoor positioning. Other positioning modalities (e.g., Wi-Fi, UWB) are used only as contextual baselines rather than as systems under study.

The approach assumes that indoor lighting infrastructure is available and that the received light intensity can be reliably measured using a photodiode receiver. It also assumes a largely static environment during data collection and evaluation. Orientation changes are handled through calibration or modeling rather than through explicit 3D pose recovery. Intermittent occlusions or multipath effects are treated as noise to be mitigated through data-driven models or sensor fusion rather than fully eliminated. These assumptions bound the contributions to indoor settings where low-cost sensing and reproducible deployment are the primary constraints.

1.4 Research Problems

The research undertaken in this thesis is motivated by the following problems:

1. Existing VLP studies rely on small datasets and weak ground-truthing, limiting reproducibility, comparability, and robustness of results.
2. RSS-based VLP systems are sensitive to orientation, multipath, and environmental variation, requiring robust data-driven approaches.

3. Robust closed-loop navigation using single-mode VLP, particularly photodiode-based systems, has not been demonstrated, and complete pose recovery (including heading) remains a gap.
4. Robots that rely solely on odometry suffer from accumulated drift, while VLP systems alone are susceptible to noise; the complementary nature of these modalities motivates fusion, yet reliable long-horizon navigation remains unresolved.

1.5 Research Questions

To address the problems identified above, this thesis is guided by the following research questions:

1. How can large-scale, high-accuracy ground truth data be efficiently collected to enable robust evaluation of VLP systems?
2. How do machine learning models compare with classical channel models for VLP when trained on large experimental datasets?
3. How can robust closed-loop navigation be achieved using single-mode VLP, and what is required to recover full pose, including heading, from photodiode-based measurements?
4. How can VLP and odometry be fused to achieve reliable long-horizon navigation, and what are the trade-offs between data-driven fusion approaches and single-mode baselines?

1.6 Research Contributions

The key contributions of this thesis, each delivered through a specific publication chapter, are as follows:

1. **Methodological:** Development of a novel, automated ground-truth and data collection system for VLP, producing the largest experimental RSS fingerprint dataset for photodiode-based VLP identified in the surveyed literature at the time of publication (Chapter 3).
2. **Empirical:** Robust benchmarking of machine learning and model-based positioning methods, demonstrating the conditions under which each approach is most effective (Chapter 3).

3. **Applied:** Design and implementation of a real-world robotic system that enables photodiode-based VLP navigation and supports full pose (position + heading) estimation under ground truth (Chapter 4).
4. **Comparative:** A head-to-head evaluation that benchmarks single-mode VLP and encoder-only baselines against Long Short-Term Memory (LSTM)-based fusion, highlighting accuracy and robustness differences in both offline and live navigation experiments (Chapter 4). Chapter 4 also includes an embedded edge deployment assessment, showing that the stacked inference pipeline is lightweight enough for real-time execution on cost-effective microcontrollers.
5. **Knowledge:** Advancement of VLP research from theoretical exploration toward practical deployment in robotic navigation, bridging a key gap in the literature.

Table 1.1 summarizes how the core research problems map to the research questions, the supporting evidence, and the resulting outputs.

1.7 Research Methodology

This thesis adopts an empirical design–build–evaluate methodology delivered through two linked studies. Chapter 3 develops and validates an automated ground-truth and data-collection framework for photodiode-based VLP, then uses the resulting dataset to benchmark learning-based and channel-model positioning methods. Chapter 4 extends this work to closed-loop robot navigation by evaluating VLP-only, encoder-odometry-only, and LSTM-based fusion under common test conditions. To connect algorithm design with deployment feasibility, Chapter 4 also profiles the complete inference pipeline on representative embedded targets, linking model complexity and latency to real-time navigation requirements.

Validation is conducted in controlled indoor experiments with independent ground truth and consistent position/heading error metrics. The evaluation combines offline testing on held-out data with live trajectory trials, and includes perturbation tests to assess robustness and long-horizon stability.

Each research question is treated as a testable empirical proposition, evaluated through the experimental frameworks in Chapters 3 and 4 and synthesized in Chapter 5.

1.8 Thesis Structure

This thesis is presented in the format of a thesis by publication. Chapter 3 reproduces the published *Sensors* article with minor formatting changes, and Chapter 4 reproduces a paper

Table 1.1 Problem–Research Question (RQ)–Evidence mapping.

Problem	RQ	Evidence / Chapter	Output
P1: Limited large-scale ground truth datasets	RQ1	Chapter 3: Automated ground-truth pipeline and large dataset	7,344-point dataset and validated benchmarks
P2: RSS sensitivity and environmental variation	RQ2	Chapter 3: machine learning (ML) vs. channel-model benchmarking	Comparative performance analysis
P3: Single-mode VLP lacks robust pose recovery in closed-loop navigation	RQ3	Chapter 4: photodiode-based VLP pose recovery in navigation	Evidence of full pose under ground truth
P4: Odometry drift and need for fusion remain unresolved for long-horizon navigation	RQ4	Chapter 4: Single-mode vs. fusion benchmarks	LSTM fusion improves long-horizon accuracy

under review at *IEEE Sensors Journal*. The surrounding chapters provide the narrative linkages: Chapter 2 frames the literature gaps addressed in the papers, while Chapter 5 synthesizes their collective contributions. The research arc therefore begins with a survey of VLP theory and limitations, proceeds to the design of a robotic ground-truthing system and large-scale data collection, uses that dataset to benchmark machine learning and channel model methods, and finally integrates VLP with robot odometry through LSTM-based fusion evaluated against single-mode baselines on real hardware.

Following this introductory chapter and a comprehensive literature review in Chapter 2:

- Chapter 3 presents the first publication: *Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning*, published in *Sensors (Q1)*.
- Chapter 4 presents the second publication: *Fusion of Visible Light Positioning and Odometry for Robot Navigation*, under review at *IEEE Sensors Journal (Q1)*.
- Chapter 5 synthesizes the findings across the publications, discusses implications, and outlines directions for future research.

Chapter 2

Literature Review

2.1 Introduction

This chapter surveys the state of the art in Visible Light Positioning (VLP) for indoor localization with an emphasis on robotics. Seven themes structure the review: (i) theoretical foundations and context, (ii) photodiode (PD)-based VLP methods, (iii) machine learning and fingerprinting approaches, (iv) robotic navigation and odometry, (v) VLP applied to robotic navigation, (vi) sensor fusion strategies, and (vii) recent advances in dataset collection and system deployment. These themes converge on three recurring gaps that motivate the research in Chapters 3 and 4: the prevalence of **low-fidelity or assumed ground truth**, the scarcity of **closed-loop evaluations** in which VLP estimates drive control, and the lack of **low-cost, robot-centered benchmarks**.

A structured search covered Institute of Electrical and Electronics Engineers (IEEE) Xplore, Association for Computing Machinery (ACM) Digital Library, SpringerLink, and Google Scholar, using combinations of keywords such as “visible light positioning,” “robot navigation,” “odometry fusion,” “ground truth,” and “dataset collection.” Forward and backward citation chaining around seminal works ensured coverage of both foundational and emerging strands. The references included in this chapter therefore span analytical channel models, learning-based perception, robotic integration, and fusion with complementary sensors.

Priority was given to work published between 2018 and 2025, while foundational papers earlier than this window were retained where they established core models or evaluation practices. Inclusion focused on studies with clear relevance to robotic navigation, PD-based receivers, and empirical evaluation (simulation or hardware), alongside representative surveys that frame the broader Indoor Positioning Systems (IPS) and VLP landscape.

Across these strands, three shortfalls recur. First, many robotic demonstrations still rely on manual floor grids or assumed trajectories for evaluation, limiting dataset scale and the credibility of centimeter-level claims [14, 15]. Second, autonomous calibration pipelines are only beginning to appear, either via handheld Simultaneous Localization and Mapping (SLAM) rigs [16] or specialized robot platforms [17], leaving a gap for low-cost, repeatable ground-truthing. Third, fusion studies frequently report localization accuracy in isolation without assessing how estimates perform inside closed-loop control [18, 19]. These issues motivate the large-scale, high-fidelity datasets and navigation-focused evaluations developed in Chapters 3 and 4.

Dedicated surveys of ground-truth measurement systems similarly stress that evaluation fidelity hinges on the choice of reference technology, reinforcing the need for independently validated datasets [20].

2.2 Foundations and Theoretical Background

Indoor positioning systems address the limitations of GPS in enclosed environments. Radio Frequency (RF), ultrasonic, infrared, and vision-based localization remain the dominant alternatives, yet each faces trade-offs in accuracy, cost, interference resilience, and infrastructure burden. Comparative surveys chart the broader sensing landscape for Autonomous Mobile Robots (AMRs), reinforcing the need for hybrid solutions that balance accuracy, cost, and deployability [3, 6]. Light Detection and Ranging (LiDAR) odometry surveys emphasize the maturity and expense of state-of-the-art range-based navigation pipelines, highlighting the gap that lightweight optical localization aims to fill [21]. At the application level, large cross-technology benchmarks for AMRs illustrate the accuracy gap between RF-based systems and emerging optical positioning, underscoring the appeal of centimeter-level optical localization [22]. A recent holistic survey by Zhu *et al.* synthesizes VLP fundamentals, algorithm families, system-level design, applications, and open challenges, providing a broad reference point for this review [23]. Against that backdrop, three issues remain consistently reported in the literature: sensitivity of RSS-based methods to orientation changes [24], limitations in ground-truth fidelity in indoor testbeds [20], and real-time processing constraints that shape hardware and algorithm choices [25].

VLP systems are typically categorized by the receiver modality. Camera-based approaches exploit spatial light patterns and embedded identifiers at the cost of high processing requirements, whereas PD-based receivers offer low latency and power consumption but suffer from tilt sensitivity and multipath [2, 3]. PD receivers align with the resource profile of lightweight robots and permit inference at the edge without dedicated Graphics Pro-

cessing Units (GPUs). Recent bio-inspired work illustrates growing interest in low-cost, light-based navigation for resource-constrained collectives, reinforcing the appeal of PD-centric approaches for simple robotic platforms [26]. The remainder of this chapter therefore concentrates on advances that are most pertinent to PD-centric robotics.

2.3 Photodiode-Based VLP

2.3.1 Photodiode Receivers and Measurement Modalities

Photodiode (PD) receivers convert incident light intensity into electrical current, enabling low-power, low-latency sensing compared with image sensors while retaining compatibility with LED-based infrastructure [2, 3, 23]. In PD-based VLP, localization can be derived from Received Signal Strength (RSS), time-based features such as Time Difference of Arrival/Phase Difference of Arrival (TDOA/PDOA), or Angle of Arrival (AOA) cues obtained from apertures or multi-element PD arrays [27, 28]. Hybrid designs combine RSS with angular information to improve observability, but they typically require tighter synchronization, specialized optics, or multi-sensor calibration [29]. These constraints limit current industrial deployment, whereas RSS-based PD approaches remain the most feasible low-cost option for practical robots. Accordingly, this thesis focuses on RSS-centric PD localization.

2.3.2 RSS Extraction, Channel Models, and Fingerprinting

RSS represents the received optical power for each luminaire after demodulation of the modulated lighting signals. In practice, identifiers are encoded into the LED waveform and separated using frequency-domain or correlation-based processing to recover per-LED RSS values [30, 25]. Analytical approaches map RSS to range using Lambertian channel models and then solve for position via least-squares trilateration or related estimators [31, 5]. Fingerprinting instead learns a direct mapping from RSS patterns to position, avoiding explicit channel modeling at the cost of data collection and coverage requirements [3].

2.3.3 Receiver Orientation and Multipath Effects

Receiver tilt and multipath reflections remain persistent limitations for PD localization. Mitigation strategies include mechanical leveling or calibration procedures, angular-diversity receivers with multiple PDs to infer incidence geometry, Inertial Measurement Unit (IMU)-aided compensation, and learning-based correction models that map RSS distortions to pose-dependent bias [29, 32, 18, 33]. These approaches improve robustness but introduce

extra hardware or modeling complexity, motivating the thesis focus on calibrated RSS pipelines and low-cost fusion.

2.3.4 Calibration and Deployment at Scale

Deploying VLP beyond laboratory testbeds demands efficient calibration of LED layouts and channel parameters. Amsters *et al.* demonstrate that a mobile robot equipped with camera and inertial sensors can automate beacon identification and location estimation, halving calibration error relative to manual surveys while generating consistent maps across varying lighting conditions [17]. Liang *et al.* extend this notion with LedMapper by formulating LED mapping as a factor-graph Simultaneous Localization and Mapping (SLAM) problem that fuses visual-inertial odometry with sparse priors, yielding workspace-aligned LED maps while quantifying uncertainty and improving repeatability in lab-scale environments [16]. Vieira *et al.* develop a Gazebo-based validation framework that allows algorithm designers to prototype robotic VLP concepts in large virtual spaces before deploying hardware [34]. Despite this progress, comparative evaluations remain fragmented, and few datasets provide millimeter-level ground truth across thousands of robot-mounted samples.

Literature on VLP using PD-based receivers establishes a mature RSS pipeline and emerging non-RSS alternatives, but robustness depends heavily on calibration and orientation handling. In robotics terms, this fragility limits repeatability and makes long-horizon control sensitive to small modeling errors. The requirement is a calibrated, low-cost PD pipeline with reliable ground truth for validation, which motivates the autonomous data collection and calibration strategy developed in Chapter 3.

2.4 Machine Learning and Fingerprinting Approaches

Fingerprinting circumvents the need for explicit channel models by learning a mapping from RSS or temporal signatures to position. Early comparisons revealed that small datasets (< 700 samples) and sparse calibration limit the gains of learning relative to analytical models [3]. Contemporary studies have responded by increasing data density and embracing sequential models that exploit temporal structure. Shu *et al.* report a 69% reduction in positioning error when combining Principal Component Analysis (PCA)-denoised RSS features with Long Short-Term Memory (LSTM) networks, yielding 95% of errors below 5 cm in a 1 m² cell [35]. Wang *et al.* [19] propose a dual-drive fusion architecture that reflects the move toward hybrid learning. Transformer and graph encoders generate confidence scores to adjust filter parameters, cutting average localization error by more than 50% relative to

fixed-parameter baselines. Beyond single-robot settings, Liu *et al.* [36] formulate cooperative multi-robot VLP as a compressed-sensing problem. They use a Generative Adversarial Network (GAN) to improve robustness to sparse observations and highlight the need for large, diverse datasets for statistical generalization. High-capacity networks and hybrid filters extend robustness under noise or sparse anchors, reinforcing the thesis focus on high-fidelity data for machine learning benchmarks [37, 27, 38]. Minimal-infrastructure variants (single-LED or Visual Odometry (VO)-assisted) and broader surveys continue to map the machine learning (ML) design space [39–41].

Synthesis: Learning-based positioning can outperform analytical models when data are abundant, but performance degrades rapidly under sparse or biased datasets. For robotics, this means model selection must be grounded in realistic, high-coverage training data and evaluated over time, not just at isolated points. This requirement directly motivates the large-scale dataset and the long-horizon comparisons in Chapters 3 and 4.

2.4.1 Machine Learning Foundations for VLP

Visible Light Positioning (VLP) methods can be divided broadly into analytic (model-based) and data-driven approaches. In analytic methods, the relationship between Received Signal Strength (RSS) and position is expressed through an explicit physical or geometric model. A common example is to first estimate the distance from the receiver to several luminaires using a calibrated RSS–distance relationship, and then recover position through geometric methods such as trilateration or least-squares lateration [4, 5]. These approaches are attractive because their structure is interpretable, but their performance depends on how well the assumed signal model matches the real environment.

Machine learning approaches instead use collected data to learn the mapping from RSS measurements to position directly. In this setting, the training data consist of many examples of RSS vectors paired with known ground-truth locations. The model is then fitted to these examples so that it can infer position when given new RSS measurements. Different machine learning models use the training data in different ways. A k -nearest neighbors (k -NN) model, for example, stores the training points and estimates position by identifying previous RSS patterns that are most similar to the current observation. In contrast, models such as polynomial regression or a Multilayer Perceptron (MLP) do not retain the full dataset as the predictive mechanism; instead, they learn a set of parameters from the data that define how inputs are transformed into position estimates. During training, these parameters are updated iteratively so that the model better captures the relationship between RSS patterns and spatial location [3, 37].

Some machine learning models further improve performance by combining many simple predictive models into an ensemble. Random forests are a common example: rather than relying on a single highly expressive model, they aggregate the outputs of many decision trees, each of which may only provide a coarse or partially accurate estimate on its own. When combined, however, the ensemble can produce a robust and accurate predictor. This illustrates an important point for VLP fingerprinting: there is no single learning strategy, but rather a range of model families that differ in how they use data, represent patterns, and trade off flexibility against interpretability [42].

Generalization is a central issue for all such models. A model may fit the training data well yet perform poorly at unseen locations if the dataset is too small, unevenly distributed, or unrepresentative of the deployment environment. For this reason, dataset scale, spatial coverage, and validation strategy are not merely implementation details, but fundamental parts of model development [36, 38]. These considerations are directly relevant to Chapter 3, where the effect of data volume on both machine learning and analytic methods is examined systematically.

A further practical issue is that many models contain not only parameters that are learned from data, but also hyperparameters that define the structure or behavior of the model itself. Examples include the number of neighbors in k -NN, the polynomial order in regression, the number and depth of trees in a random forest, or the number of layers and hidden units in an MLP. These values are not learned automatically in the same way as model parameters; instead, they must be selected through a model-selection process, typically using validation data or cross-validation. The choice of hyperparameters can have a substantial effect on performance, meaning that procedures such as hyperparameter tuning are often required before a fair comparison between models can be made [43].

For robotic navigation, the learning problem becomes sequential rather than purely static. Odometry errors accumulate over time, while VLP measurements provide drift-free but potentially noisy absolute updates. The resulting estimation problem is therefore history-dependent: the current best estimate depends not only on the latest measurement, but also on how recent motion and sensing information are interpreted over multiple time steps [44, 19]. This temporal structure motivates recurrent models such as Long Short-Term Memory (LSTM) networks. Unlike static regression models, which operate on the current measurement alone, an LSTM is trained on ordered sequences of inputs so that each estimate can be informed by both the present observation and a window of recent prior measurements. In practice, this means that isolated static samples are not sufficient training data; sequential trajectories of robot motion are required so that temporal dependencies, drift patterns, and measurement corrections can be learned. The amount of prior context provided to the

model, often expressed as the sequence length, is itself a model-design choice that can affect performance. In the context of this thesis, machine learning is therefore relevant not only for static RSS fingerprinting, but also for learning how temporal measurement patterns can stabilize navigation when VLP is fused with odometry [35].

2.5 Robotic Navigation and Odometry

Reliable robotic navigation typically begins with wheel encoder odometry, whose limitations are well documented. Chong and Kleeman’s seminal analysis quantified systematic and random odometry errors, providing calibration techniques such as UMBmark and closed-form covariance propagation that remain standard practice [44]. Industrial surveys continue to advocate multi-sensor fusion to counteract drift, but many proposed solutions rely on LiDAR, visual odometry, or Ultra-Wideband (UWB) sensors that are cost-prohibitive for minimalist platforms [6]. Consequently, low-cost robots continue to rely on encoders despite cumulative drift, making them prime candidates for augmentation with lightweight references such as VLP. This context sets the stage for evaluating how VLP can complement odometry both analytically and through data-driven fusion, as pursued in Chapter 4.

Synthesis: Odometry is ubiquitous but drifts, and low-cost robots lack the sensor budget to rely on heavy SLAM stacks. The robotics requirement is therefore a lightweight absolute reference that can arrest drift without expensive hardware. This motivates the photodiode-based VLP plus encoder fusion architecture evaluated in Chapter 4.

2.6 VLP for Robotic Navigation

PD-only navigation prototypes remain comparatively sparse and seldom evaluate closed-loop performance with rigorous ground truth [45, 7]. Most published navigation results are camera-based or cooperative and rely on internal maps rather than independent ground truth [10, 46, 15]. Camera-centric TurtleBot systems report centimeter-level accuracy but depend on manual grids or assumed trajectories, limiting reproducibility and comparability [14]. This leaves a clear gap for PD-centric robot navigation that is evaluated with independent ground truth, and motivates the focus of Chapter 4 on rigorous, closed-loop assessment of PD-based VLP and fusion on simple robots. Evaluation norms increasingly emphasize pointwise accuracy and trajectory-level metrics such as path deviation, drift over time, and closed-loop task success under independent ground truth [14, 10, 8, 47]. The following subsections therefore begin with camera-based and cooperative navigation, then contrast the

limited PD-only literature to motivate the low-cost, externally validated fusion experiments in Chapter 4.

Synthesis: Existing navigation demonstrations show feasibility but often lack independent ground truth or long-horizon evaluation. In robotics terms, this leaves open whether localization accuracy translates into reliable closed-loop behavior. The requirement is trajectory-level evaluation with external ground truth, which is addressed in Chapter 4 via repeatable paths and controlled perturbations.

2.6.1 Camera-Based and Cooperative Navigation

Camera receivers broaden the field of view and encode LED identifiers but at higher computational cost. Guan *et al.* fused rolling-shutter VLP, LiDAR-SLAM, and wheel odometry within the Robot Operating System (ROS), demonstrating autonomous navigation with static accuracy around 2.5 cm and 42 ms latency [10]. Earlier ROS-based deployments already achieved centimeter-scale localization using smart LED fixtures, though they relied on manual grids for validation and lacked closed-loop control [14, 48]. Later demonstrations hybridized VLP with LiDAR to improve robustness while maintaining ROS integration [49]. Cooperative schemes extend these ideas to multi-robot teams: Xu *et al.* combined LED anchors with inter-robot communication, achieving 6.2 cm error for a micro-Unmanned Aerial Vehicle (micro-UAV) swarm using a Computer Numerical Control (CNC) ground-truthing platform [47], while Yan *et al.* demonstrated cooperative camera-VLP fusion for paired TurtleBot3 platforms [15]. Recent work has also examined single-luminaire configurations: Zhu *et al.* combine visual odometry with one luminaire and one camera, resolve pose ambiguities without additional sensors, and report sub-10 cm accuracy in prototype experiments [40]. This camera-based approach provides a useful contrast to the photodiode-plus-encoder focus adopted in this thesis.

2.6.2 Photodiode-Only Navigation

The earliest demonstrations of VLP-controlled robots showcased feasibility but lacked rigorous evaluation. Sharifi *et al.* identified and modulated LEDs using a photodiode receiver on a differential-drive robot, yet they reported no quantitative accuracy and relied on oscilloscope traces to verify demodulation [45]. Hsueh *et al.* integrated VLP with lamp-to-robot communication for waypoint following, again judging success qualitatively [7]. Chew *et al.* advanced the state of the art by executing parameterized paths on a CNC platform using PD-only localization [8]. The median path errors of 27 mm underline the potential of PD receivers for closed-loop control when actuation is precise. Konings *et al.* complemented

these efforts with a hybrid visible light and Zigbee system that achieved sub-6 cm median error [9]. However, their study remained limited to room-level passive localization without closed-loop control. These studies demonstrate feasibility while highlighting the need for quantitative ground truth and dynamic evaluation.

Projection-based systems take a different path by encoding localization and control commands into high-speed projected imagery. Hiraki *et al.* [50] implemented pixel-level visible light communication with a digital light projector. It enabled simultaneous localization and routing of multiple robots in time-varying velocity fields, demonstrating the breadth of light-based navigation.

Synthesis: Projection systems highlight alternative control channels but rely on bespoke infrastructure and do not generalize to typical LED deployments. For practical indoor robotics, the requirement is to work with standard lighting and compact receivers. This reinforces the PD-centric scope adopted in the thesis.

2.7 Sensor Fusion Approaches

Fusion remains the most active frontier for VLP-enabled robotics. Hua *et al.* pioneered PD-camera fusion via a filter-based estimator with fixed-lag smoothing, achieving 8.7 cm/12.2 cm accuracy in 2D/3D respectively on a mobile robot while handling interference [51]. Wang *et al.*'s tightly coupled data-model architecture adapts filter parameters on the fly using learned confidence metrics, improving resilience to occlusion and tilt [19]. Recent transformer- and graph-based variants clarify how these learned components are being used. *VLP-BERT* applies Bidirectional Encoder Representations from Transformers (BERT)-style contextual encoding to denoise visible-light signals before tightly coupled Visible Light Positioning (VLP) and Pedestrian Dead Reckoning (PDR) fusion, targeting blockage and poor signal quality [52]. By contrast, *DeepVLP* organizes Light Emitting Diode (LED) measurements as a graph so that inter-LED correlations and attention weights can suppress interference before Long Short-Term Memory (LSTM)-based trajectory refinement [53]. Together with Wang *et al.*'s transformer/Graph Attention Network (GAT)-enhanced confidence estimation, these studies show a broader shift from plain temporal recurrence toward architectures that explicitly model signal context, anchor relationships, and observation reliability, albeit with higher model complexity and limited closed-loop robot validation. Sun *et al.* employ factor-graph optimization to fuse PD and inertial cues, explicitly modeling inclination and blockage [18]. Within the SLAM community, Huang *et al.* augment RTAB-Map with VLP constraints to bound LiDAR and visual odometry drift, albeit with a heavy sensor payload moved manually through the environment [46]. The diversity of modalities and

estimators underscores the need for direct comparative evaluations on resource-constrained robots. This gap is addressed in Chapter 4 by evaluating LSTM-based fusion against single-mode baselines under identical trajectories and ground truth. Wearable and event-based platforms further expand the fusion landscape. Li *et al.* combined photodiode VLP with pedestrian dead reckoning using a nonlinear filter, achieving sub-meter accuracy in handheld and foot-mounted trials that could extend to Automated Guided Vehicles (AGVs) [54]. Wang *et al.* recently merged photodiodes with event cameras, using particle filtering to sustain single-LED tracking under aggressive motion or occlusion [38]. Such results complement the low-cost, encoder-centric fusion investigated in Chapter 4.

For the low-cost encoder+VLP setting studied in this thesis, however, a direct model-based filter baseline is not uniquely defined. The VLP subsystem does not provide an independent heading measurement; instead, heading must be inferred from successive position estimates, which couples heading and position noise and makes filter performance sensitive to pseudo-measurement construction, dependence assumptions, and covariance design. Fixed-parameter filters offer interpretable update structure under explicit process and measurement models, while adaptive variants improve robustness by learning confidence or tuning terms within that same filtering framework. Chapter 4 therefore prioritizes an LSTM-based recurrent model as a first fusion study, because the collected dataset is explicitly sequential and allows nonlinear temporal relationships to be learned directly from experimental trajectories rather than prescribed through hand-specified update equations. A head-to-head comparison against carefully formulated EKF / UKF variants remains an important open comparative question rather than a trivial baseline omission.

Synthesis: Fusion research demonstrates that combining complementary sensors improves robustness, but many systems rely on camera or IMU-heavy payloads. The robotics requirement for this thesis is a fusion method that is feasible on low-cost platforms and comparable across models. Chapter 4 answers this by benchmarking LSTM-based fusion against single-mode baselines using identical inputs and trajectories.

2.7.1 Summary of Fusion Studies Relevant to Chapter 4

Table 2.1 highlights that state-of-the-art fusion strategies often trade interpretability for adaptivity, or vice versa, and seldom evaluate long-term drift on lightweight robots. Chapter 4 contributes by comparing data-driven fusion against single-mode baselines under identical trajectories and high-fidelity ground truth.

Table 2.1 Representative VLP fusion studies relevant to PD-based robotics.

Study	Fusion modality	Evaluation setup	Key findings and limitations
Hua <i>et al.</i> [51]	PD + smartphone camera; filter-based fusion with fixed-lag smoothing	Mobile robot; 5 m × 5 m indoor testbed	Achieves 8.7 cm (2D) / 12.2 cm (3D) accuracy under interference; limited to semi-real-time processing with camera Field of View (FOV) constraints.
Sun <i>et al.</i> [18]	PD + IMU; tightly coupled graph optimization	Device-level platform; tilt and blockage scenarios	Handles inclination and blockage with decimeter-level Six Degrees of Freedom (6DoF) accuracy; not evaluated in closed-loop robot control.
Wang <i>et al.</i> [19]	PD + IMU; transformer/Graph Attention Network (GAT)-enhanced filter-based fusion	Device-level experiments with occlusion	Reduces localization error by >50% via learned confidence; computational complexity and robot integration untested.
Guan <i>et al.</i> [10]	Camera VLP + LiDAR-SLAM + wheel odometry	TurtleBot3; ROS navigation stack	Demonstrates 2.5 cm static accuracy and real-time path execution; high-cost sensor suite and lack of external ground truth.
Huang <i>et al.</i> [46]	VLP constraints within LiDAR/visual SLAM	AGV platform moved manually; Jetson-class compute	Integrates VLP constraints to curb SLAM drift; evaluation is quasi-static and sensor payload is heavy for low-cost robots.

2.8 Recent Advances in Dataset Collection and Robotic Applications

High-quality datasets and repeatable evaluation frameworks are essential for comparing algorithms. The calibration and fusion comparisons above reveal that available datasets either lack ground-truth fidelity, rely on costly sensors, or omit closed-loop control. Cooperative localization studies for micro-UAVs and service robots continue to expand the application space, yet most depend on carefully scripted trajectories and either motion-capture ground truth or planar CNC stages [47, 36]. Ground-truth measurement surveys underline the trade-offs between optical motion capture systems and time-of-flight/ultrasonic platforms, balancing accuracy, coverage, and deployment cost [20]. Recent robotic demonstrations focus on operational concerns such as Field-Programmable Gate Array (FPGA)-based demodulation [25] and PD-only navigation on CNC platforms [8]. However centimeter-level accuracy claims remain difficult to compare across studies because common metrics and datasets are scarce. Chapter 3 responds by combining an automated ground-truthing rig (HTC Vive tracker) with a large-scale fingerprinting campaign. Chapter 4 then examines how sensor fusion translates into trajectory fidelity for a low-cost differential-drive robot.

Table 2.2 contextualizes the automated ground-truth pipeline developed in Chapter 3. Prior work either relies on specialized hardware (robot-mounted camera rigs), sacrifices fidelity through simulation, or excludes optical modalities altogether. The work presented in the chapter therefore fills a gap by combining low-cost robotics with millimeter-accurate tracking to generate the largest experimental RSS fingerprint dataset for photodiode-based VLP identified in the surveyed literature at the time of publication.

Synthesis: The literature lacks shared benchmarks that tie localization accuracy to long-term navigation behavior. For robotics, that gap makes it difficult to judge whether reported centimeter errors are meaningful in practice. The requirement is a dataset and evaluation protocol that connect pointwise accuracy to trajectory outcomes, which is delivered in Chapters 3 and 4.

2.9 Summary and Positioning of This Thesis

The literature highlights rapid progress in PD-based modeling, machine learning, and fusion, yet three shortcomings persist: (i) centimeter-level errors are frequently reported without independent ground truth; (ii) few studies close the loop between localization and control on low-cost robots; and (iii) dataset scale and public availability remain limited. The contributions that follow address these gaps by (a) developing an automated robotic data

Table 2.2 Representative dataset and ground-truth pipelines aligned with the objectives of Chapter 3.

Study	Platform and scale	Ground truth / calibration	Key contribution and limitations
Amsters <i>et al.</i> [17]	Differential-drive robot; office-scale LED network	Robot-mounted camera and inertial sensing for beacon identification	Automated calibration halves positional error versus manual surveys; requires bespoke robot and significant visual infrastructure.
Liang <i>et al.</i> [16]	Handheld dual-camera rig; lab and office deployments	Visual-inertial SLAM with factor-graph optimization	Produces workspace-aligned LED maps efficiently; relies on handheld mapping and priors such as ceiling height.
Vieira <i>et al.</i> [34]	Simulated Autonomous Mobile Robot (AMR) in Gazebo; 20 m $\times$ 20 m virtual arena	Gazebo truth model	Modular validation environment accelerates algorithm testing; simulation lacks real-world multipath and sensor noise.
Windolf <i>et al.</i> [55]	Vicon motion capture; lab-scale volume	Optical motion capture (marker-based)	High-precision ground truth; high cost and limited coverage constrain deployment.
Lazik <i>et al.</i> [56]	Indoor testbed; anchor-based ultrasound system	Ultrasound time-of-flight ranging	Low-cost ranging infrastructure; requires synchronization and dense anchor deployment.
Khoshrangbaf <i>et al.</i> [22]	Kobuki AMR; 431 m ² warehouse	Anchor geometry; RF positioning baselines	Provides cross-technology baseline (Wi-Fi/Bluetooth Low Energy/Ultra-Wideband (BLE/UWB)) for AMRs; optical modalities not included, so centimeter-level optical claims remain unbenchmarked.

collection pipeline with millimeter-accurate ground truth, (b) benchmarking analytical and machine learning approaches on the resulting large-scale dataset, and (c) evaluating data-driven fusion against single-mode baselines for real-time navigation on a resource-constrained platform. These steps progress VLP from controlled experiments toward practical deployment in robotics, directly answering the research questions posed in Chapter 1.

Chapter 3

Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning

Preface

This chapter is based on the publication:

Glass, T.; Alam, F.; Legg, M.; Noble, F. Autonomous Fingerprinting and Large Experimental Data Set for Visible Light Positioning. *Sensors* **2021**, *21*, 3256. <https://doi.org/10.3390/s21093256>

The candidate was the primary author of this work. **Author contribution:** The candidate led the system design, data collection, analysis, and manuscript preparation. Co-authors provided supervision, feedback, and editorial input.

The published article is open access under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, which permits reuse with attribution (<https://creativecommons.org/licenses/by/4.0/>).

The published version of the paper has been reformatted to match the style of this thesis. Minor edits have been made for consistency in notation, referencing, and formatting. The scientific content remains unchanged.

Linking Overview

Chapter 2 identified three persistent gaps in the VLP literature: limited ground-truth fidelity, small experimental datasets, and a lack of evaluations that translate into reproducible benchmarks. This chapter addresses the first two of those gaps by establishing a scalable data-collection pipeline that couples a low-cost mobile robot with a high-precision tracking system. The central contribution is methodological: a practical way to gather thousands of photodiode Received Signal Strength (RSS) measurements with millimeter-level ground truth across a realistic indoor space. This foundation enables robust comparisons across positioning methods and supports the analyses presented here.

The paper also establishes the empirical baseline for the remainder of the thesis. It quantifies how dataset size changes the performance of channel-model and fingerprinting approaches, and it documents the calibration steps required to make RSS-based ranging viable in a real environment. The dataset and benchmarking results therefore provide both a practical resource and a conceptual reference point for later chapters.

These findings should, however, be interpreted within the scope of the test environment. The chapter establishes a strong benchmark for RSS-based VLP in a real indoor office setting, including practical features such as occlusion, room geometry, and ambient environmental variation, but it does not by itself establish performance across the wider range of buildings, lighting layouts, and deployment conditions that may be encountered in practice.

Finally, this chapter provides the static positioning evidence needed before moving to navigation. The robot-assisted collection system and the verified ground-truth pipeline ensure that later navigation experiments can be assessed against a credible baseline. Chapter 4 then builds on this work by shifting from static accuracy to dynamic performance, evaluating single-mode VLP navigation and its fusion with odometry under real-time, closed-loop motion.

3.1 Introduction

Localization and navigation systems have been widely used since the development of Global Positioning System (GPS) [57]. However, GPS struggles to operate reliably indoors due to signal degradation and multipath propagation [1]. A large number of potential applications such as smart guidance in large facilities, e.g., hospitals and shopping malls, ambient assisted living in smart homes, and asset tracking have led researchers to develop Indoor Positioning System (IPS) using a wide variety of techniques including Infrared (IR) [58], Wi-Fi [59],

Bluetooth [60], Zigbee [61], ultra-wideband (UWB) [62], acoustics [63], and computer vision systems [64].

The rapid adoption of the Light Emitting Diode (LED) for illumination and the advancement of Visible Light Communication (VLC) [65] technology have made Visible Light Positioning (VLP) [66] a promising option for IPS. VLP falls into two broad categories: camera-based [67] and photodiode-based [5]. The lower price and power costs of photodiodes make them preferable for certain applications such as asset tracking. While several signal characteristics like Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA), etc. have been reported in the literature, Received Signal Strength (RSS) is most widely utilized for photodiode-based VLP [2]. RSS-based positioning can be achieved with cost-effective hardware without requiring any synchronized infrastructure.

RSS-based VLP can be broadly categorized into model-based [4] or fingerprinting approaches [68]. Model-based approaches develop calibrated signal attenuation–distance relationships that are used to estimate the range of the target from multiple luminaires whose locations are known a priori. Once three or more ranges are available, the target can be localized. Fingerprint techniques, on the other hand, capture a labeled offline dataset of locations and corresponding RSS of multiple luminaires and use this to build a machine learning algorithm capable of predicting online locations. For both approaches, there is a need for a large number of experimental measurements of RSS and the positions at which they were taken with high confidence in accuracy. This data is used to calibrate the model, train the algorithms (e.g., regressors) as well as to evaluate the accuracy of the localization system.

Collecting accurate ground truth data is a major challenge for VLP and indoor positioning research. Current techniques are labor-intensive and restrict the amount of data that can be collected. Generally, data collection is accomplished by placing a receiver on a series of pre-marked reference locations and manually recording the position and the corresponding RSS. This method requires a substantial time investment for both measuring the reference locations as well as the numerous repositionings of the receiver during data collection. The painstaking nature of the method can also introduce error as user fatigue develops over long sessions of data collection. This has resulted in VLP research being conducted with limited-size datasets. Table 3.1 presents the data collection sizes for recent VLP experiments. Most of the reported testbeds are much smaller than a typical real-world environment, which makes it difficult to assess how the findings would translate into operational performance. Furthermore, the largest experimental RSS fingerprint dataset for photodiode-based VLP identified in the surveyed literature at the time of publication has only 693 recorded locations [69], which is an order of magnitude lower than the 7344 recorded locations over 43.5 m²

presented in this paper. Furthermore, most of the works do not discuss the ground truth recording methodology when, in most cases, reporting mean localization errors of 50 mm or lower. This also casts doubt on the accuracy of such reported measurements.

Table 3.1 VLP articles listed by size of the experimental dataset. Most articles use a limited number of data points and do not describe how ground truth was recorded.

Research	Number of Data Points	Year	Testbed Size Mean		Method	Ground Truth
			(mxm)	Accuracy (mm)		
Alam et al. [69]	693	2018	3.3×2.1	27	WKNN	Laser rangefinder
Guo et al. [70]	225	2019	0.7×0.7	50	KNN, RF, ELM	Not stated
Vongkulbhisal et al. [71]	160	2012	1.8×1.2	148	KNN	Not stated
Zhang et al. [72]	100	2019	1.8×1.8	34	NN	Not stated
Zhang et al. [73]	100	2019	1.8×1.8	18.8	NN, MMBP	Not stated
Chuang et al. [74]	81	2019	0.5×0.5	40	Regression	Not stated
Wu et al. [75]	81	2020	0.5×0.5	30	Regression	Not stated

The lack of large datasets has led to researchers collecting small datasets and then using interpolation [75] and regeneration [69] to produce larger fingerprints. However, due to non-ideal signal characteristics such as multipath propagation, such methods are less accurate. When training machine learning (ML) models, regularization has been used to accommodate small training datasets [72, 73]. However, due to the lack of sufficient testing points, it is difficult to assess the generality of the models to data not encountered, and the small volume of data used dramatically increases the likelihood of overfit. Using small datasets reduces the accuracy scores for both fingerprinting and model-based methods as there are not enough testing points to provide a robust sample. Furthermore, the difficulty in collecting data severely limits the effectiveness of fingerprinting systems in changing environments as a new dataset would be required after any part of the environment is altered.

A system that can automatically capture the ground truth position of a VLP receiver (often termed a “tag”) accurately would resolve the issues present with current data collection methods. High accuracy commercial systems, such as motion capture and SLAM, exist; however, they are prohibitively expensive, and the sensors can be bulky. Recently, lower-cost systems using ultrasonic localization have been used [76]. However, such systems have a mean accuracy of ~20 mm [77], which is several times worse than the Vive. This paper outlines the use of the HTC Vive tracking platform [78], specifically the lighthouse base stations and tracker, for ground truth recording. The Vive system is an order of magnitude more accurate than VLP systems, allowing it to reliably evaluate the accuracy of the VLP system under investigation. The Vive system is used along with a Xiaomi Roborock S5 robotic vacuum [79] that automates the repositioning of the target around the test area, making

the entire data collection system free from user involvement once the setup is complete. The paper offers the following contributions:

1. It presents a new technique for automating the collection of large VLP datasets with high positioning accuracy and wide area coverage. This allows high accuracy calibration and validation of VLP systems. This is the first time such a system has been reported in the literature.
2. The implementational details of the ground truth recording system are provided. This will enable other researchers to develop their own systems to collect VLP data over an arbitrarily large area. We also discuss key issues like mitigating the Vive's IR interference and the robotic platform's Electromagnetic Interference (EMI) on the VLP system. Experimental study demonstrates the consistency and the accuracy of the Vive's positioning performance.
3. A large dataset containing visible light RSS from 11 luminaires at 7344 test locations over a 43.5 m² area is constructed and made available online for other researchers to utilize in their own research. To the best of the authors' knowledge, this is the largest experimental RSS fingerprint dataset reported for photodiode-based VLP in the surveyed literature at the time of publication, and it is an order of magnitude larger than previously reported datasets.
4. The dataset is used to provide example evaluations of common machine-learning-based positioning. We demonstrate that a large dataset is necessary not only for training models, but also for robust performance comparisons. For large datasets, ridge regression is found to outperform other machine-learning techniques in terms of localization accuracy. The Lambertian RSS-distance channel model is calibrated for ranging that enables localization using linear-least-squares- and iterative-spring-relaxation-based lateration. Such model-based positioning is shown to be well suited for a small dataset, with the ML-based positioning being superior when a large dataset is available.

The rest of the paper is organized as follows. Section 2 describes the components used for data collection: the VLP system, the Vive tracking including an analysis of ground truth performance and coordinate transformation, and use of the robotic vacuum platform. Section 3 describes how the experiments were conducted and compares the data collected with theoretical models. Section 4 details the performance of some common ML-based localization algorithms trained on the collected data, and Section 5 does the same with two

channel-model-based methods. Section 6 highlights the need for large datasets for both training and testing. Section 7 concludes the paper with suggestions for future work.

3.2 System Overview

3.2.1 Experimental Setup

The test environment consisted of a $6.5 \times 7 \times 2.4$ m office lobby that was cleared of furniture (see Fig. 3.1). There are 11 bespoke luminaires spaced as evenly as possible within the constraints imposed by the existing lighting and air conditioning installations. A pillar is present in the room that obstructs the light from several luminaires in specific locations.

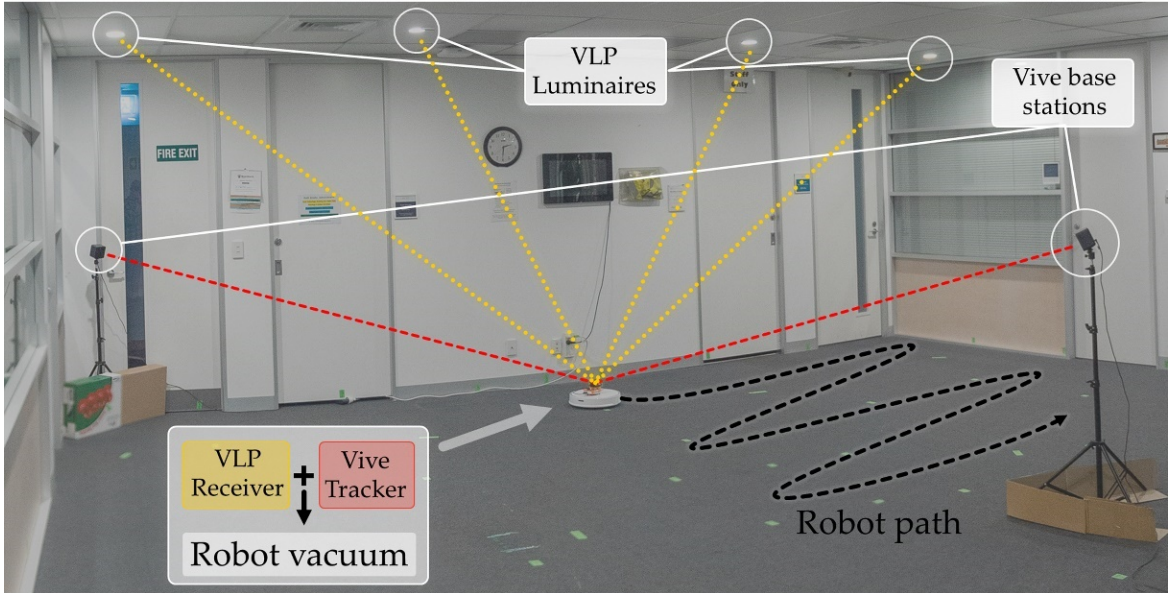


Fig. 3.1 Office lobby used for the experiment. Two Vive base stations provide accurate positioning for the Vive tracker, which is co-located with the VLP receiver on a mobile robot vacuum.

3.2.2 VLP System

The details of the VLP system employed in this work can be found in reference [5]. The 11 luminaires used in the experimental setup were distinguished from each other using frequency division multiplexing (FDM; see Fig. 3.2), with custom driver boards inserting an unmodulated sine wave between 2 to 22 kHz. The receiver tag comprises a photodiode feeding into a trans-impedance amplifier with filtering to remove ambient light fluctuation

and 100 Hz power line flicker. It was found that the IR signals emitted from the Vive lighthouses interfered with the VLP system (Fig. 3.3a). To overcome this, an IR-rejecting photodiode [80] was used and placed behind a section of IR blocking glass (Fig. 3.3b).

The received signal at the output of the amplifier is sampled at a rate of 50 kHz by an STM32 microcontroller [81], and the data are sent to an ESP8266 Wi-Fi chip [82] that hosts a Hypertext Transfer Protocol (HTTP) server. A computer connected to the local network can obtain a new reading using a simple HTTP GET request. Demultiplexing and RSS extraction are performed using a Fast Fourier Transform (FFT). The power of the resulting spectra at each of the 11 multiplexing frequencies is used as a measure of the RSS for the corresponding luminaire.

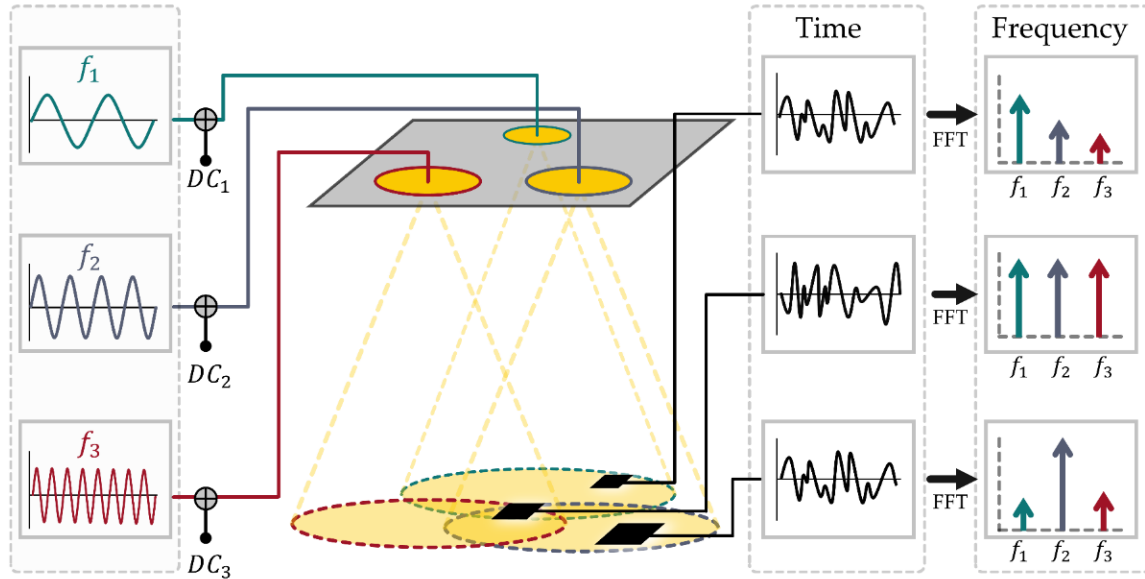


Fig. 3.2 FDM scheme demonstrated for three luminaires. Each luminaire is driven by a direct-current (DC) signal overlaid with a sinusoid of unique frequency. A photodiode detector receives a signal that is the sum of multiple luminaire outputs scaled by distance. This signal is then converted with a Fast Fourier Transform (FFT), and the component at each luminaire’s modulating frequency is used as the RSS.

3.2.3 HTC Vive Tracking for Accurate Ground Truth

Accurate calibration and benchmarking of VLP localization methods require precise knowledge of the locations where VLP data measurements are made. To achieve this, the HTC Vive tracking system [78] was used. This system enables high-precision positioning for Virtual Reality (VR). Its low-cost and accurate tracking performance have led to interest in its use

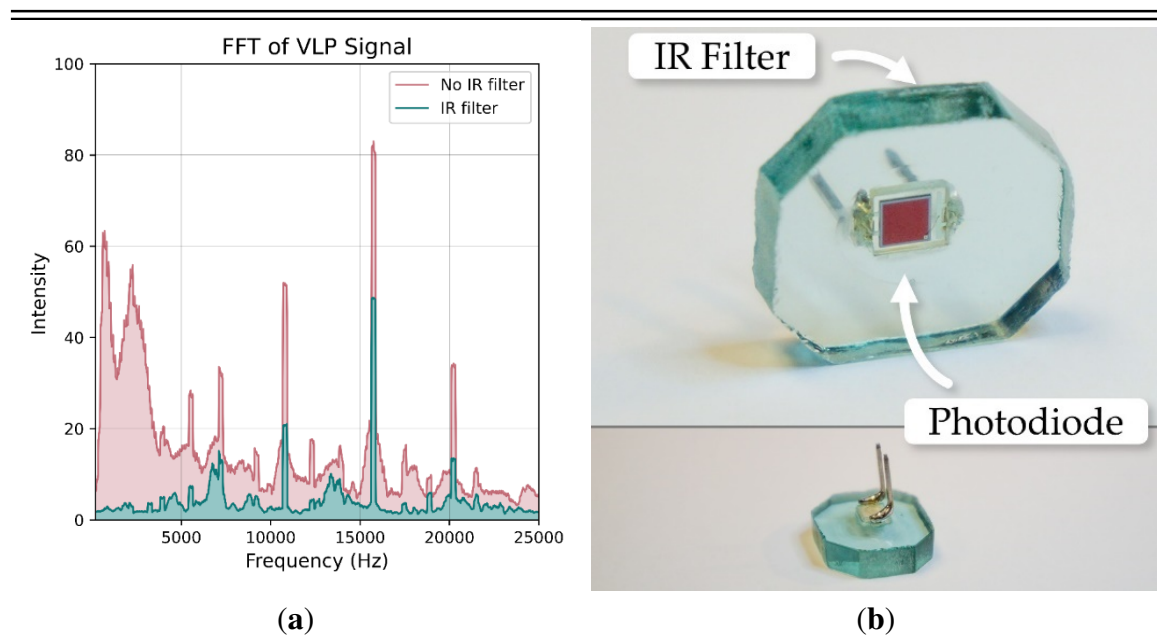


Fig. 3.3 The IR signals from the Vive base stations caused interference with the VLP receivers as shown in (a). The Vive signal added substantial noise at frequencies lower than 4 kHz and altered the intensities of the luminaire peaks. To prevent interference, an IR-rejecting photodiode was used and then placed behind IR-rejecting glass (b). The combination of the two eliminated the interference.

for various research applications [83–87]. The basic tracking system consists of two or more fixed “Lighthouse” base stations and a mobile Tracker. The system has been reported to provide tracking accuracy of about 5.8 mm [87] and therefore seems suitable for determining the ground truth position during experimental VLP data collection. Co-locating the Vive tracker with a VLP receiver eliminates the need for painstaking manual measurements.

The tracker provides 6 DoF (degrees of freedom) localization $(\hat{X}, \hat{Y}, \hat{Z}, \gamma, \beta, \alpha)$ and uses Bluetooth to communicate to a wireless dongle connected to a host PC running SteamVR [88]. The pyopenvr [89] library provides Python bindings for the OpenVR virtual reality SDK [90] that accesses SteamVR, allowing for scripting control of the tracking system. The Triad OpenVR wrapper [91] simplifies the interface, providing a simple Python function to request the tracker’s current position.

The tracker’s location is provided with respect to a local coordinate system centered at the first base station with axes aligned along its orientation, which is tilted during setup (Fig. 3.4a). This necessitates a transformation to map between the reference coordinate frames. Deriving this transform involves placing the tracker at known locations within the test environment and building a set of points in each reference frame with the same location. The mapping from the Vive to the room coordinates can then be derived and applied to any subsequently recorded points.

In theory, only three non-collinear reference points are required to define the transform. However, any errors in these points would be propagated to the final transform, and the use of multiple points reduces the likelihood of error. To avoid an over-defined solution when using multiple points, the Iterative Closest Point (ICP) algorithm [92], a point set registration technique for determining the spatial transformation that aligns two sets of points, is used to determine the final transform. The algorithm first assigns a correspondence from the target set to the point set using a nearest-neighbor approach. A transformation is then derived using a translation followed by rotation to attempt to align the matched points. The process is repeated with both the transformation and correspondence, improving at each iteration until the point-to-point distance is minimized, or a maximum number of iterations is reached.

To ensure convergence of the ICP, a rough initial alignment is required so that the point-to-point correspondence settles on the correct mapping. This alignment is derived using Euler’s rotation theorem [92] applied to points that roughly align with the three coordinate unit vectors. First, the origins of the two reference frames are aligned by applying a translation (see (3.1) and Fig. 3.4a):

$$T = \begin{bmatrix} -local\ origin's\ (x) \\ -local\ origin's\ (y) \\ -local\ origin's\ (z) \end{bmatrix} \quad (3.1)$$

Next are three rotations that follow the form of a $\hat{Z}$ - $\hat{Y}$ - $\hat{Z}$ fixed angle set [93].

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$R_x(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (3.3)$$

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

where γ , β , and α are the rotation angles around $\hat{Z}$, $\hat{Y}$, and $\hat{Z}$ axes, respectively. γ , β , and α approximations are derived by using dot and cross products between the points roughly aligning with the unit vectors in each coordinate system. All the transformation matrices then can be used to perform transformation of a $[3 \times N]$ matrix of array of N measurement point coordinates X_v from the Vive's coordinate system into the room's coordinate system using

$$X_{room} = (R_z \cdot R_x \cdot R_z) X_v + T \quad (3.5)$$

Fig. 3.4 shows the step-by-step transformation. After the rough alignment between calibration points, the Open3D library [94] is used to implement the ICP algorithm deriving the transformation matrix from the Vive coordinate frame to the room. This transformation matrix needs to be derived each time a new experimental setup is used. In the next sub-section, we outline an analysis of the number of calibration points required to accurately determine the transformation.

3.2.4 Vive Performance Analysis

Two sets of experiments were done to verify the accuracy of the Vive positioning system. Firstly, the system was tested for 'jitter', the amount of variation the system reported for a stationary object. Secondly, the tracking accuracy was investigated using a Computer Numerical Control (CNC) platform with a resolution of 0.025 mm.

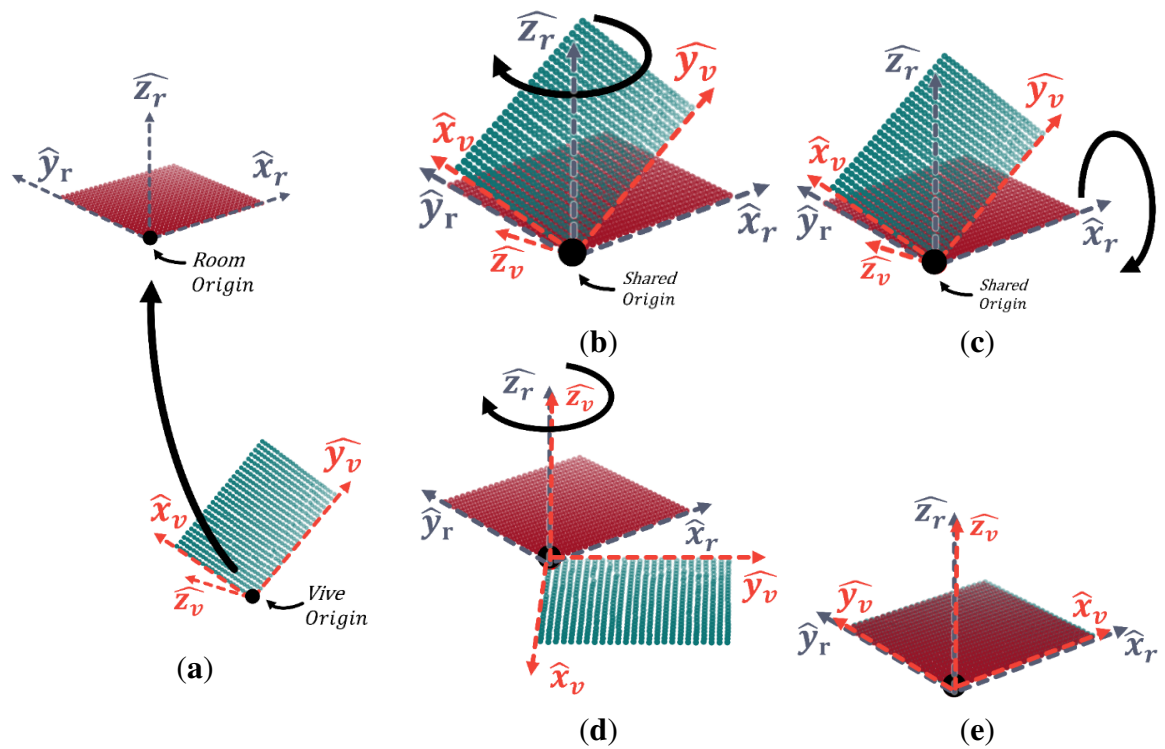


Fig. 3.4 Using Z-X-Z Euler angles to derive an initial alignment between the Vive coordinate reference frame ($\hat{x}_v, \hat{y}_v, \hat{z}_v$) and the room coordinate frame ($\hat{x}_r, \hat{y}_r, \hat{z}_r$). A translation aligns the origins (a); then, $\hat{z}_v$ is aligned with the $\hat{z}_r, \hat{y}_r$ plane via rotation around $\hat{z}_r$ (b). This allows a rotation around $\hat{x}_r$ to align $\hat{z}_r$ with $\hat{z}_v$ (c). Finally, another rotation around $\hat{z}_r$ (d) completes the alignment (e).

Consistency of Vive's Position Measurement

To investigate the system's consistency, the two Vive base stations were placed 3 m apart, and the tracker was kept immobile on a stationary surface in the view of both stations. Two data collection runs were repeated: A and B. For both runs, 100 measurements were taken at intervals of 1.5 s. Runs A and B were conducted with the tracker at different locations and orientations to determine if these factors affected measurement consistency.

Table 3.2 shows metrics for the difference in the Euclidean distance between the estimated position and mean for the 100 readings. The median, 90th percentile, and max variations were below 0.15, 0.35, and 0.65 mm, respectively, for both runs. These results indicate the positional variation is small enough to avoid needing multiple ground truth readings (with subsequent averaging) at any test location. These results are in agreement with previously reported work [87].

It can also be seen in Fig. 3.5 that the Cumulative Distribution Function (CDF) of the error distance for all one hundred points for both runs are very similar despite having different orientations and being placed in different locations, suggesting that the system has a low systematic error.

Table 3.2 Variation in Vive measurement in mm for a stationary target. Each run was conducted with a different tracker position and orientation.

Run	Mean	Median	90th Percentile	Max
A	0.138	0.115	0.249	0.460
B	0.137	0.109	0.263	0.642

Accuracy of the Vive System

To investigate the system's accuracy, the performance of the VR system was tested using a custom-made 2D CNC machine with an accuracy of 0.025 mm. The CNC machine's tool position can be set by providing the desired x, y coordinates. Mounting the Vive tracker to the tool allows its true position to be accurately determined, which can then be compared with the Vive positional outputs. A total of 625 tracker position measurements were taken at 50 mm intervals within a 1.2×1.2 m space shown in Fig. 3.6a, with the orientation of the tracker kept constant. Rough alignment followed by ICP was applied to the Vive's output to align the two coordinate reference frames, with the ICP algorithm using all data points to derive the transformation. The error CDF for the CNC points is shown in Fig. 3.6b.

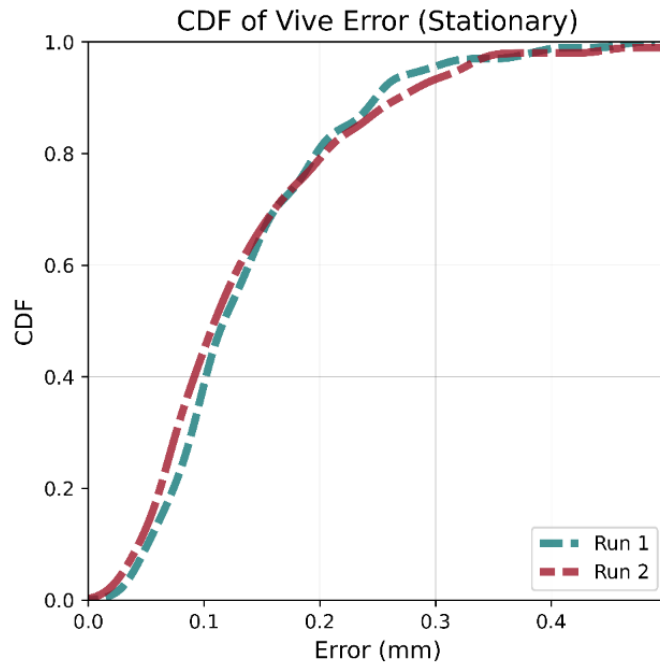


Fig. 3.5 CDF for Vive jitter at different locations and orientations. The two runs are very similar indicating that there is no spatial dependence.

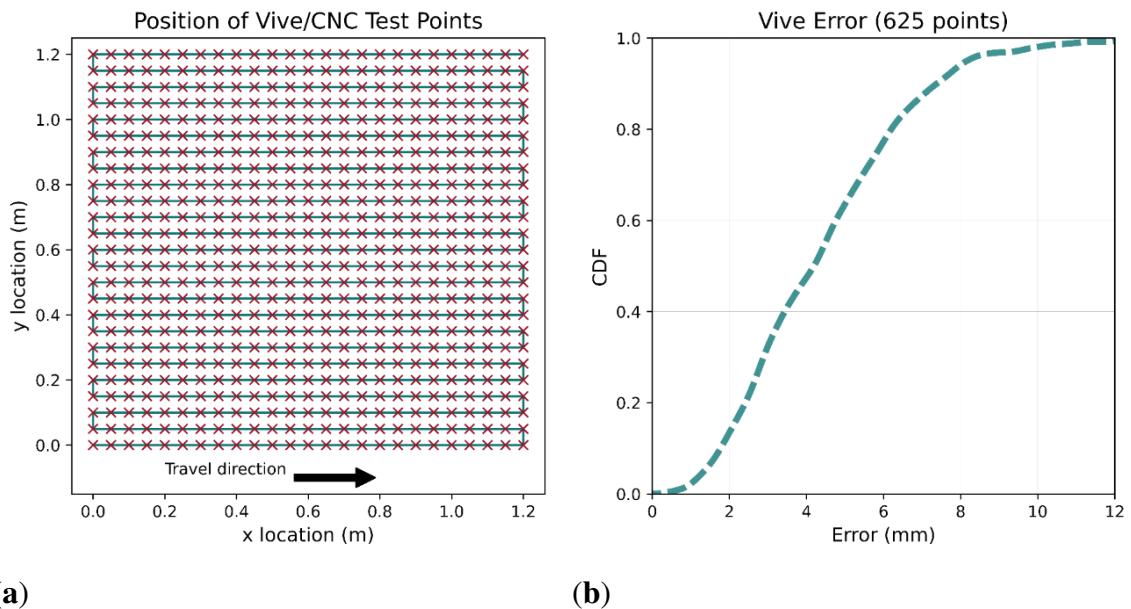


Fig. 3.6 Determination of the Vive accuracy. The tracker was moved to 625 locations (a) by the CNC machine. The high accuracy of the CNC allows the Vive errors to be accurately calculated. The error CDF is shown in (b); median error is less than 4.5 mm.

Number of Calibration Points Needed

The results presented in the previous sub-section demonstrate that the ICP algorithm can map the Vive's position coordinates to the room's reference frame with high accuracy. However, the ICP used all 625 points when finding the best transformation. Due to the impracticality of utilizing such a large number of calibration points, an investigation into the accuracy of the transformation using a limited number of points was undertaken, first using points at random, then using carefully chosen locations.

To evaluate the accuracy of ICP when using p number of points, a transformation matrix is computed using the p points and then applied to all 625 points. The accuracy is then computed as the mean distance from each transformed Vive point to the ground truth position provided by the CNC. Starting with $p = 3$, (the minimum number of points required to derive the transformation) and increasing the number of points provides a way to determine the number required.

Fig. 3.7 shows the mean accuracy error vs. the number of calibration points when the points are chosen at random. Three different runs are plotted, each one using a different random shuffling of the 625 points. As p is increased, the accuracy of the transformation improves; however, the random selection of points produces inconsistent results. Since this work required 2D ground truth, the Z-axis was nullified when comparing the transformations, future work involving 3D positioning will need to re-validate the optimum number of reference points needed.

If, rather than selecting the calibration points randomly, each new point is chosen to be as far as possible from any existing calibration point, fewer points are needed. An example of such point selection is shown in Fig. 3.8a. If the initial point is at the bottom right (point 1), the next point selected will be point 2 (top right) as this is the furthest away. Subsequent points are then chosen to be as far away from any other calibration point as possible. When using this method, the transformation error converges close to the lowest overall error with fewer calibration points as demonstrated in Fig. 3.8b. It can be seen that five calibration points result in an error that remains below 5 mm.

3.2.5 Roborock S50 Robotic Platform

To eliminate the need for manual repositioning of the target node, the Roborock S5 robotic vacuum platform was used. Both the Vive Tracker and VLP receiver were mounted to the top of the vacuum (see Fig. 3.9). The Roborock S5 has a suite of sensors including ultrasonic ranging, collision bump sensors, IR cliff sensors, and, most importantly, a Light Detection and Ranging (LiDAR) sensor that allows the robot to implement Simultaneous Localization

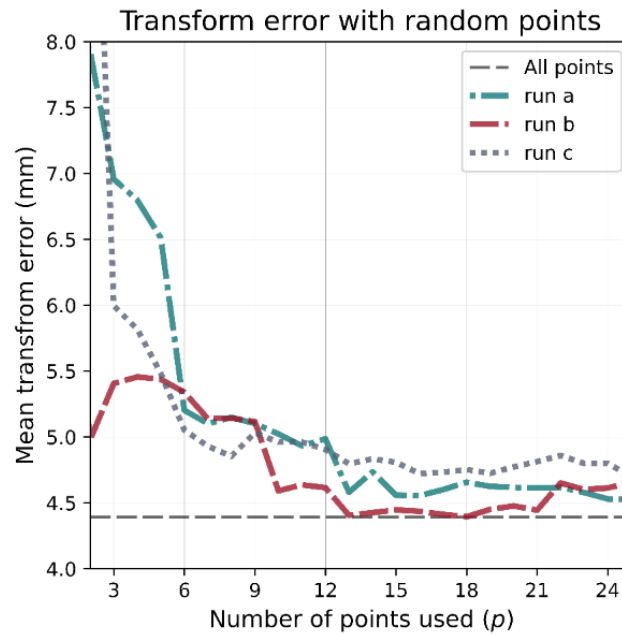


Fig. 3.7 Dependence of transform accuracy on the number of calibration points. As more points are used, accuracy improves. The dashed gray line represents the ideal transform calculated using all 625 points. Using randomly chosen calibration points produces inconsistent results.

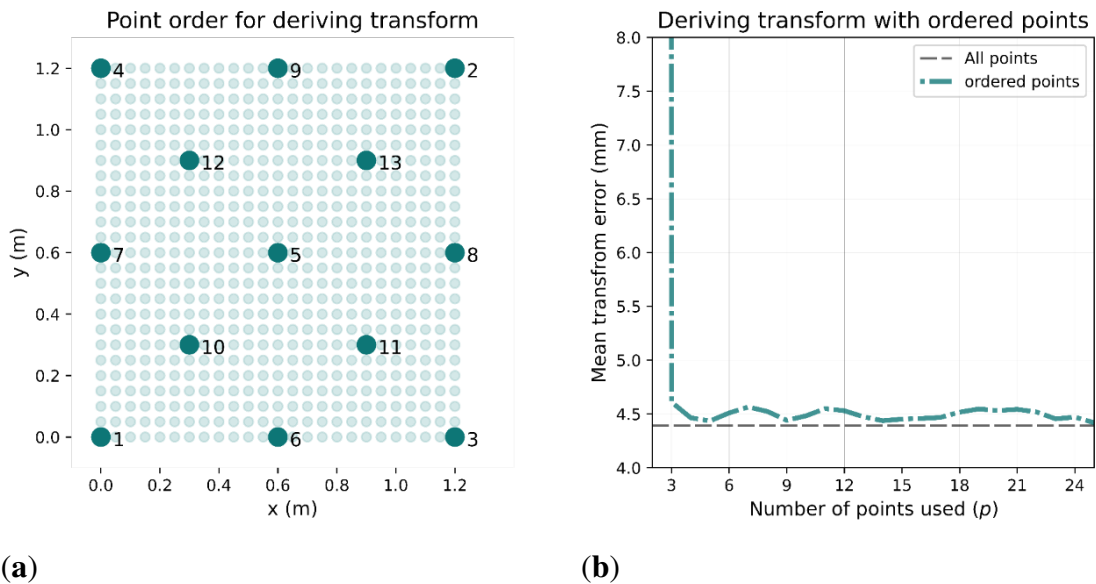


Fig. 3.8 Selecting calibration points to maximise area covered. If points are chosen by following the pattern in (a), then the transform accuracy converges much faster, as can be seen in (b). With only 3 calibration points, the mean error is below 5 mm (shown in (b)).

and Mapping (SLAM) to build detailed maps of the environment for the purpose of increasing cleaning efficiency. The robot has two major processors [95, 96]: an Allwinner R-16 with four ARM A7 CPU cores and an STM32 microcontroller. The device runs Ubuntu with Player 3.10-svn, which provides the mapping functionality.

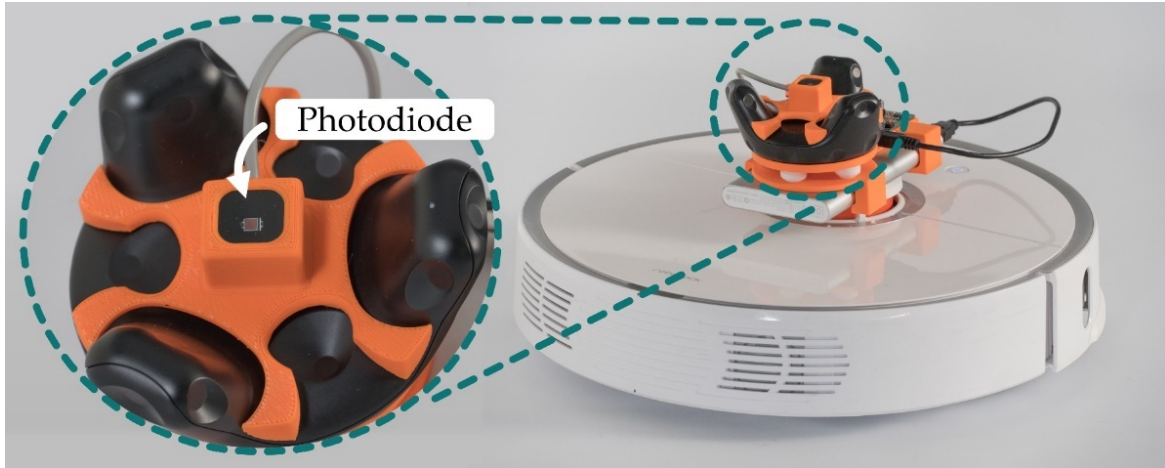


Fig. 3.9 The Vive tracker was mounted to the top of the Roborock S5 with care taken to avoid blocking the LiDAR. The VLP tag was mounted in the center of the tracker without blocking the Vive’s photodiodes. Both the Vive tracker and the VLP tag were powered using a USB power bank.

The vacuuming and brush electric motors could not be removed without the operating system disabling the vacuum entirely. Therefore, to reduce power consumption, the vacuum was modified by removing the motor driving gears. Upon restart, the robot has an initialization routine of around 5 s, during which time the motors wind up. This places a limit on the collection frequency of the system. If stationary readings were not required, then the robot would not need to be stopped. A small delay is also needed between start and stop Application Programming Interface (API) commands of around 0.5 s to prevent the system from temporary locking messages. The robot vacuum’s motors can introduce noise into the VLP system as shown in Fig. 3.10. If readings were to be taken while the robot is still moving, some form of electromagnetic shielding would be required. However, during this experiment, the robot was fully stopped before taking each reading, meaning this was not required.

The open-source home automation software platform Home Assistant [97] was used to control the area where VLP measurements were made. It has integrated control for the Roborock S5 vacuum and provides a Python package exposing the robot’s API calls. Before using the API package, a token for the device must be extracted to authenticate commands [98]. Once set up, the API provides a convenient way to issue numerous commands, which

include starting/stopping of automated cleaning, forward/backward/rotation commands, and moving to specific coordinates (in the robot's map reference frame). The Python API calls can also be used in conjunction with the Xiaomi Home mobile app [99], which allows for easy setup configuration. After placing the robot in a new location and starting a cleaning routine, the app builds a map of the environment and displays the robot position within it.

As shown in Fig. 3.11, after placing “virtual walls” (in red), a “cleaning zone” can be defined (light color), and the robot will be restricted in its movement. Once the “cleaning routine” begins, the robot travels around the perimeter before sweeping back and forth over the entire area. The API commands can pause the operation to allow for fingerprinting measurements to be captured and then resume “cleaning” without disrupting progress.

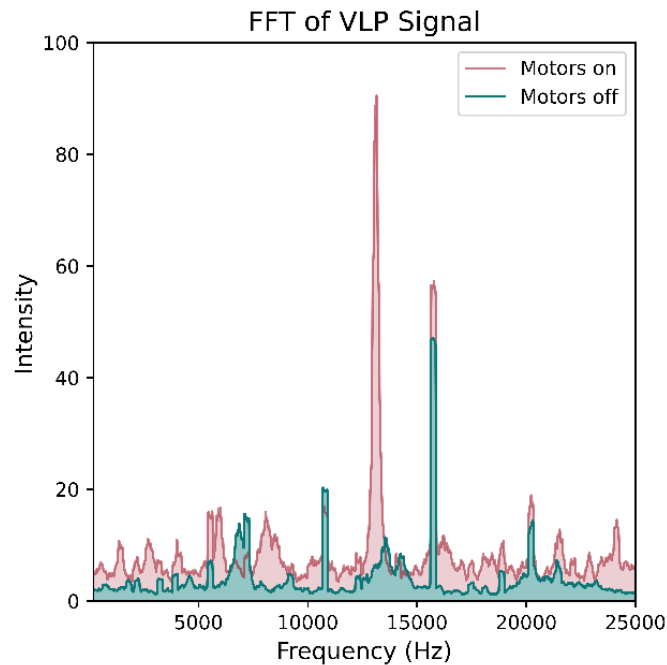


Fig. 3.10 Visualization of noise caused by the EMI of the vacuum's motors. The green plot shows a discrete Fourier transform (DFT) of the VLP signal when the motors are off and luminaires are operating. The distinct peaks represent signals detected from the luminaires in proximity. The red plot shows the DFT of the received signal at the same location but with the vacuum motors on. In addition to a large peak at around 12.5 kHz, the background noise level has increased over the entire frequency range.

The utility of the map stems from the ability to define “no-go” regions within the app that limit the operating range of vacuum. This allows control of where the robot will focus its data collection when performing fingerprinting and provides easy segmentation of an environment. The robot can also avoid colliding with walls, pillars, and other objects within

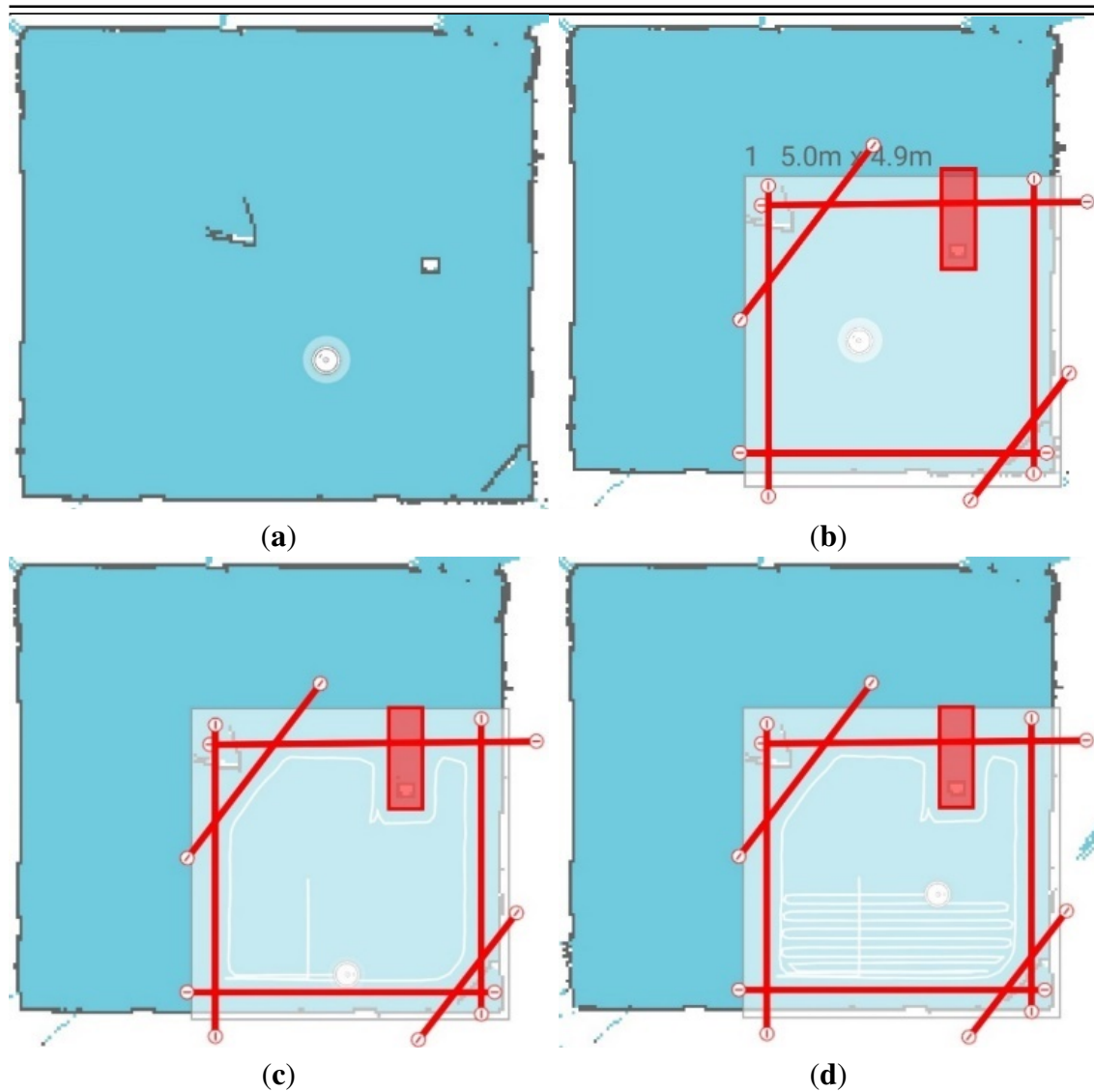


Fig. 3.11 Setting up the cleaning zone using the Xiaomi Home mobile app. When placed in a new environment, the vacuum will build a map of its environment as shown in (a); the triangle in the room's center is the cardboard deflectors placed on the Vive base station stands (Fig. 3.1). Within the app, walls and zones can be placed that will restrict the movement of the robot (b). Once initialized, the robot will then traverse the perimeter (c) before sweeping back and forth to cover the entire area (d).

the environment which reduces the possibility of damaging attached sensors and avoids wasted time as the robot would otherwise use the bump sensor and continually ‘bounce’ off walls.

3.3 RSS Data

3.3.1 Experimental Setup and Data Collection

The HTC Vive 1.0 version that was used in this work was unable to track the robot over the entire $6.5 \text{ m} \times 7 \text{ m}$ floor area of the lobby. The newer v2.0 version of these base stations [100] allows for tracking with 4 stations covering $10 \text{ m} \times 10 \text{ m}$, with the possibility of this increasing further in the future [101], which would cover the entire lobby area and provide more line-of-sight links in environments with obstacles. To collect ground truth with the v1.0 lighthouses, the room was split into four quadrants, taking measurements at each quarter (Fig. 3.12 shows the area of three quadrants) and then merging the dataset. This process could be repeated indefinitely, allowing for Vive ground truth tracking of an arbitrarily large area. Reference locations for calibration were placed on the lobby floor via manual measurements with the aid of a laser level line. The reference points were selected to provide common points between quadrants to reduce the required number of points (see Fig. 3.13). The investigation into the number of required reference points shown in Fig. 3.8b demonstrated that at least 5 were needed. However, 10 points were used for each quadrant to safeguard against erroneous readings. Before taking VLP measurements in each quadrant, the Vive lighthouses were set up and the tracker was positioned on each of the 10 reference positions, and measurements were taken. These recorded points were then used to define the translation from the Vive coordinate frame to the room frame using the process outlined in Section 2.3. The Roborock S5 was configured using the mobile app to sweep the quadrant while avoiding walls and the Vive base stations. A Python script running on a desktop connected to the local network was used for data collection and robot control. The script would call an API to pause the vacuum, make an HTTP request to the VLP receiver for a new reading, and then call the Vive API to get the ground truth position. The data were timestamped and written to a comma-separated values (CSV) file. The cleaning routine was then resumed with another API call. While moving, the change in position of the robot was measured by continuous polling of the Vive tracker, and once it moved a certain threshold distance (180 mm), the robot would pause to take another reading. This method was found to perform much better than a simple time delay due to the robot taking more time at edges, where it must turn around, than in the middle of a long straight run.

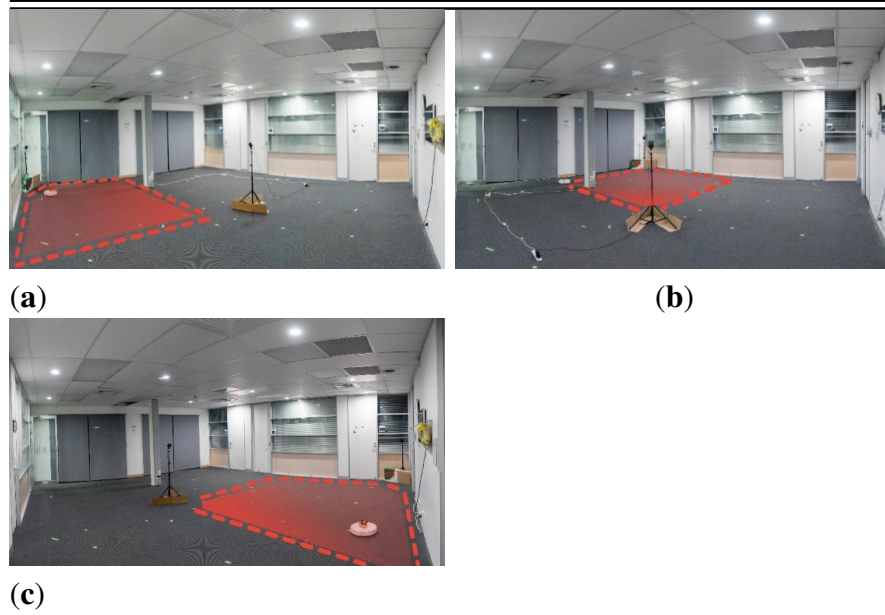


Fig. 3.12 The area of the lobby was too large to be covered by the Vive 1.0 system, which necessitated splitting the environment into 4 quadrants. Each quadrant required a separate transformation to align the Vive readings with the room's coordinate reference frame. Three quadrants are shown in (a), (b), and (c), respectively.

3.3.2 VLP Data Preprocessing

Data collected from the four quadrants were merged into one set, and the VLP readings were processed to extract the RSS values from each of the 11 luminaires. The experiment collected VLP data and Vive ground truth at 7477 different locations. Outliers in the dataset were identified using the z component of the Vive tracker position. Values that deviated from the tracker's height (which remained constant) indicated that the tracker had made erroneous position estimates. This could have been caused by line-of-sight (LOS) disruptions, network traffic congestion, or the IMU dead reckoning used by the Vive malfunctioning. Positions where the z estimate differed from the overall mean by more than 3 standard deviations were discarded, which equated to 133 points (or 1.8%). Therefore, the final dataset contains RSS values at 7344 measurement locations.

The collected dataset can be visualized by plotting the RSS for a particular luminaire at each fingerprint location. Fig. 3.14 shows plots for a selection of luminaires. As expected, the highest RSS values were beneath a luminaire and falling off with distance.

The photodiode used in the receiver has a half angle of 60 degrees, which limits the maximum range at which it can reliably detect signals from the luminaires. Furthermore, beyond a certain distance, the VLP signal falls below the level of the noise floor. To obtain an

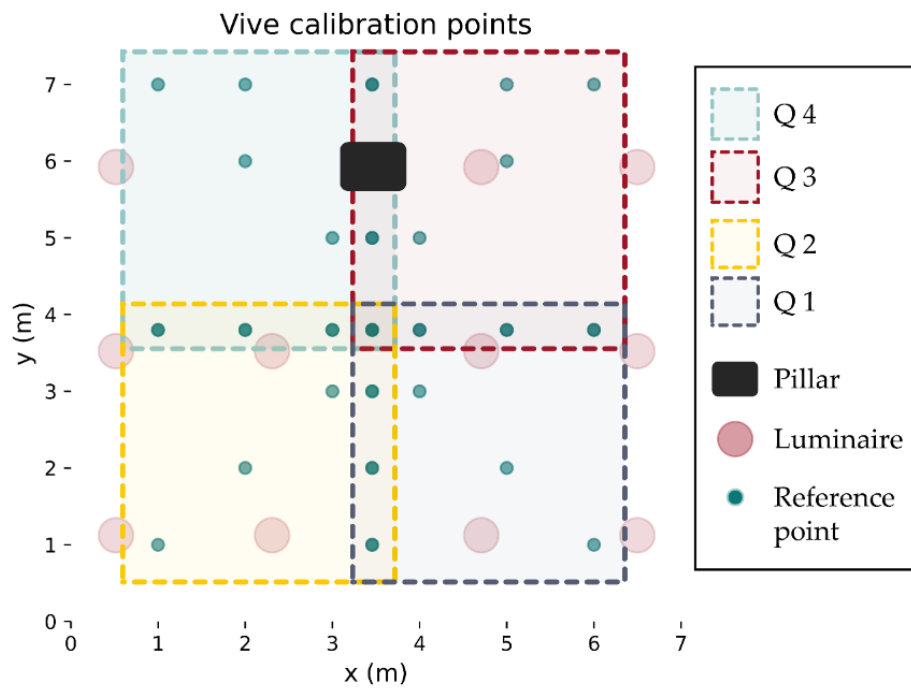


Fig. 3.13 Calibration points used for deriving the transformations. Twenty-six reference points were carefully mapped out using a laser level and tape measure in such a way that each quadrant would have 10 reference calibration points, sharing four with each neighbour. This reduced the overall number of calibration points required.

approximate measure of the noise floor, the receiver was placed at a fixed position in the room and the VLP luminaires were powered for illumination without injecting the modulation signal. One hundred eighty readings were taken at intervals of 10 s. Fig. 3.15 shows the average noise level over all frequencies of interest as well as the 1 and 2 standard deviations above. From this figure, it can be seen that the noise is generally below “15” and used as a threshold to clamp the RSS readings to 15 (if they are below the threshold).

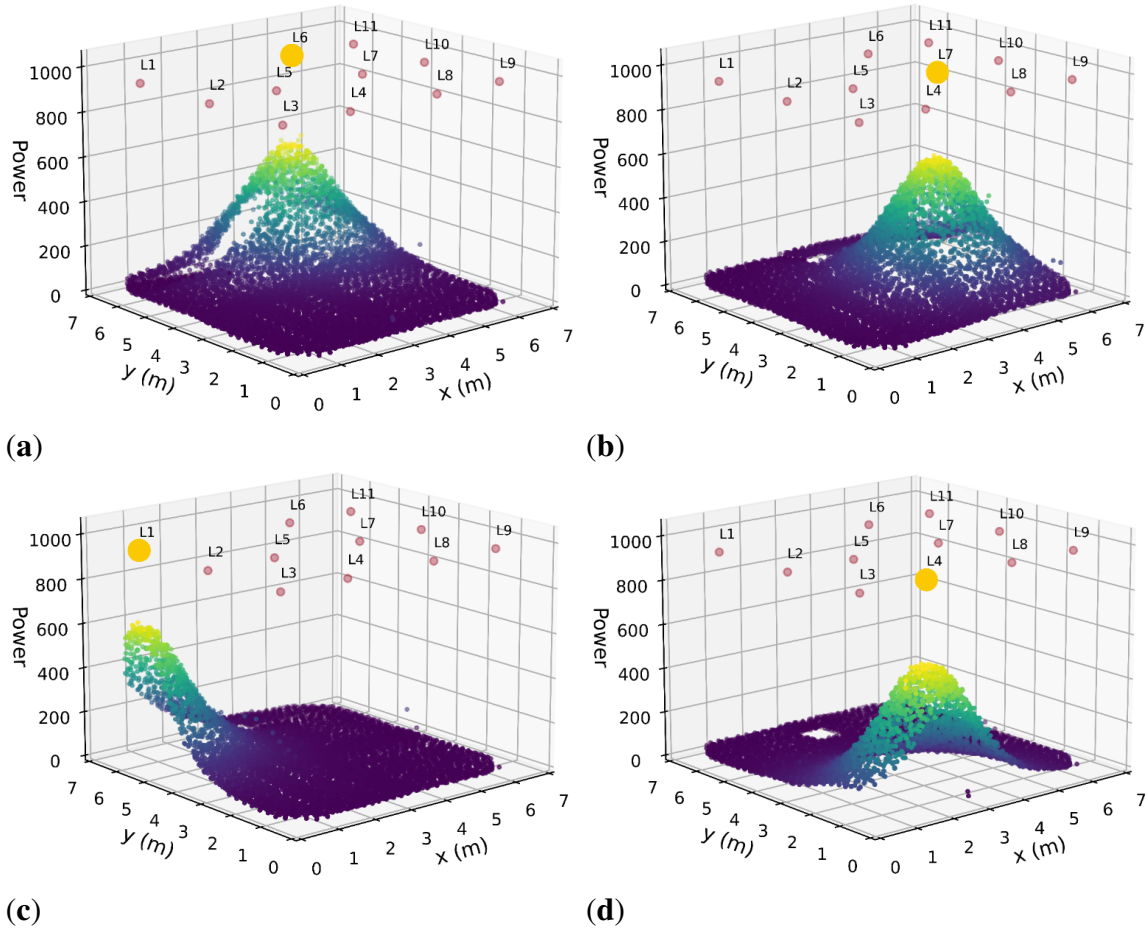


Fig. 3.14 RSS plots for luminaires 6, 7, 1, and 4 in (a), (b), (c) and (d) respectively. The signal is largest when the sensor is directly beneath and falls off with distance. No data were collected at the location of the pillar, which can be seen clearly in (d).

To the best of the authors’ knowledge, this dataset is the largest reported experimental RSS fingerprint data collection for photodiode-based VLP in the surveyed literature at the time of publication, comprising 7344 labeled locations over 43.5 m². The final labeled dataset is publicly available online [102]. To support reproducibility and reuse, the released repository includes the full labeled RSS fingerprint dataset, luminaire coordinates, and an example Jupyter notebook demonstrating how to load the data, visualize the measurements,

and train and assess a sample machine-learning regressor. This makes the dataset directly usable for independent benchmarking and for comparing alternative VLP models on a common experimental basis. However, the repository is best understood as a reusable benchmark resource rather than a complete reproduction package for every aspect of the original experimental workflow.

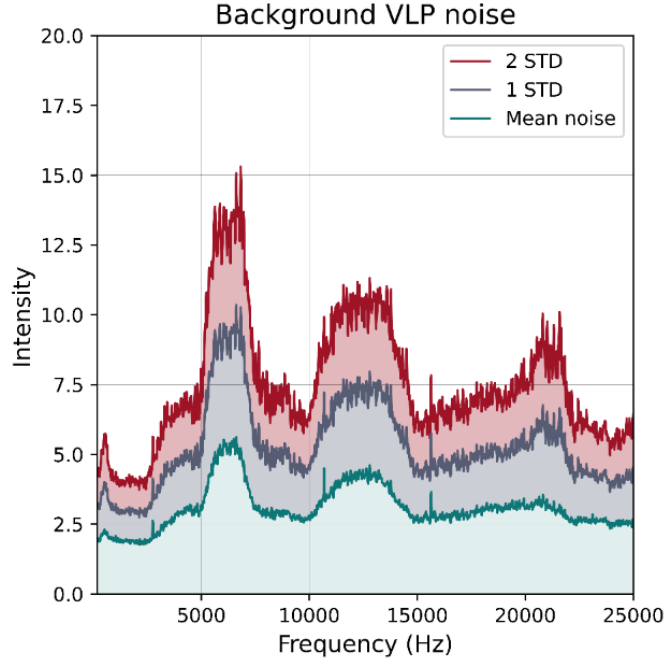


Fig. 3.15 Background VLP noise with no active luminaires. One hundred eighty DFTs were computed at 10 s intervals and for each frequency the average was calculated and plotted (green); 1 and 2 standard deviations above the mean level were also calculated to provide an indication of the noise floor.

3.3.3 Lambertian Channel Model Calibration

The Lambertian model that relates RSS to the distance of the target from the luminaire (and several other parameters) can be expressed as [103]:

$$P_{r_n} = \frac{P_{t_n}}{d_n^2} \left(\frac{m_n + 1}{2\pi} \right) \cos^{m_n}(\varphi_n) A \cos(\theta_n) \quad (3.6)$$

Here, P_{r_n} is the received power at the target location at a distance d_n away from luminaire n , P_{t_n} is the transmitted power of that luminaire, m_n is the Lambertian order, φ_n and θ_n are the irradiance and incidence angles, respectively, and A is the photodiode area. Please see Fig. 3.16 for an illustration of some of the channel parameters.

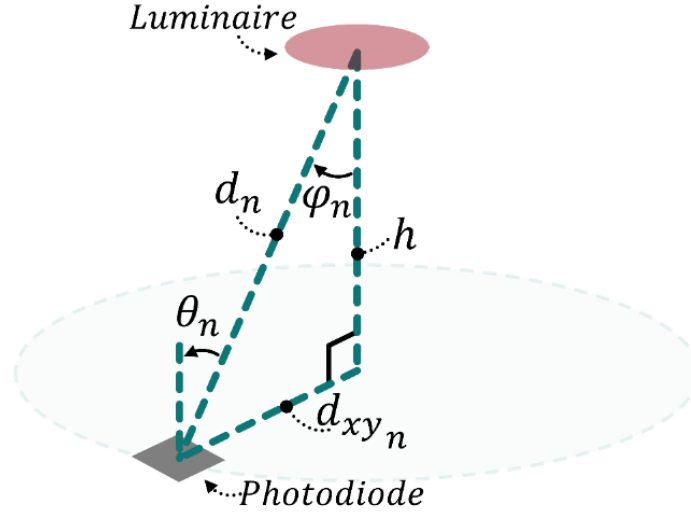


Fig. 3.16 When both the luminaire and receiver are horizontal, θ_n and φ_n are equal, and d_n then forms the hypotenuse of a right-angle triangle.

P_{r_n} is extracted following the process described in Section 3.2. Since the luminaires and the photodiode are parallel for the experimental setup, (3.6) simplifies to

$$P_{r_n} = P_{r_n,0} \left(\frac{d_{n,0}}{d_n} \right)^{m_n+3} \quad (3.7)$$

Here, $P_{r_n,0}$ is the RSS at a chosen reference location, which is $d_{n,0}$ distance away from the n th luminaire. Model calibration is performed by obtaining the values of the three parameters: m_n , $P_{r_n,0}$, and $d_{n,0}$.

$P_{r_n,0}$ and $d_{n,0}$ are derived by averaging the RSS and the distance of three closest points to the n th luminaire. Once $P_{r_n,0}$ and $d_{n,0}$ have been determined for each luminaire, (3.7) can be rearranged to determine [69]

$$m_n = \frac{\log \left(\frac{P_{r_n}}{P_{r_n,0}} \right)}{\log \left(\frac{d_{n,0}}{d_n} \right)} - 3 \quad (3.8)$$

m_n can then be estimated at a set number of data points and then averaged.

Fig. 3.17 shows the distance vs. RSS scatterplot for a selection of luminaires with the calibrated model overlaid in red. Of note is Luminaire 6, which displays clear deviation due to its close proximity to a pillar.

3.4 Machine Learning Models for Positioning

To demonstrate the utility of a dense fingerprint, several positioning algorithms were implemented on the dataset and the effect of varying volumes of training data investigated. Positioning was framed as a multi-target supervised regression problem, $\hat{\mathbf{y}} = f(\mathbf{X})$, where both the x, y coordinates of the receiver $\hat{\mathbf{y}} = (\hat{x}, \hat{y})$ were predicted using the RSS values from 11 luminaires at each position, i.e., $\mathbf{X} = (RSS_1, RSS_2, \dots, RSS_{11})$. The machine-learning models implemented were polynomial regression, polynomial ridge regression [104], Support Vector Regression (SVR) [105], random forest (RF) [42], weighted k-nearest neighbors (k-NN) [106], and multilayer perceptron (MLP) [107]. These models were selected as they are some of the most common machine-learning algorithms used for regression [43]. Algorithms were implemented using scikit-learn version 0.24.1 [108]. SVR does not support multitarget output, so two regressors were trained, one for each of the x and y components of the coordinates.

Each ML model has various hyperparameters that need to be specified, and these can have a large impact on performance. To make fair comparisons between models, the hyperparameters were tuned using fingerprint data. Table 3.3 lists the hyperparameters requiring tuning for each model. The dataset was partitioned using an 80–20 train–test split. The training data was used for hyperparameter tuning, as well as for retraining the tuned models. The tuned models were then validated against the 20% test set to make the final performance assessment. The overall process for training and testing is illustrated in Fig. 3.18.

The method of tuning comprised two successive rounds of grid-search, where the hyperparameter values were swept through a set range. For the first sweep, progressive powers of 10 were used ($\dots 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2 \dots$) as a range for each parameter and every combination of values was tested. Testing of a set of hyperparameters was undertaken using 5-fold cross-validation on 80% of the training data. The training set was split into five distinct subsets with the model being trained and evaluated five times, selecting a different fold for evaluation every time and training on the remaining four. The 5 evaluation scores were then averaged to provide an overall evaluation. Mean Square Error (MSE) was the performance metric used for selecting the best parameters. If the best performing parameters were at the upper or lower range of values, the range was extended to ensure the optimum values were contained within the search range. This process was then repeated with powers of 2

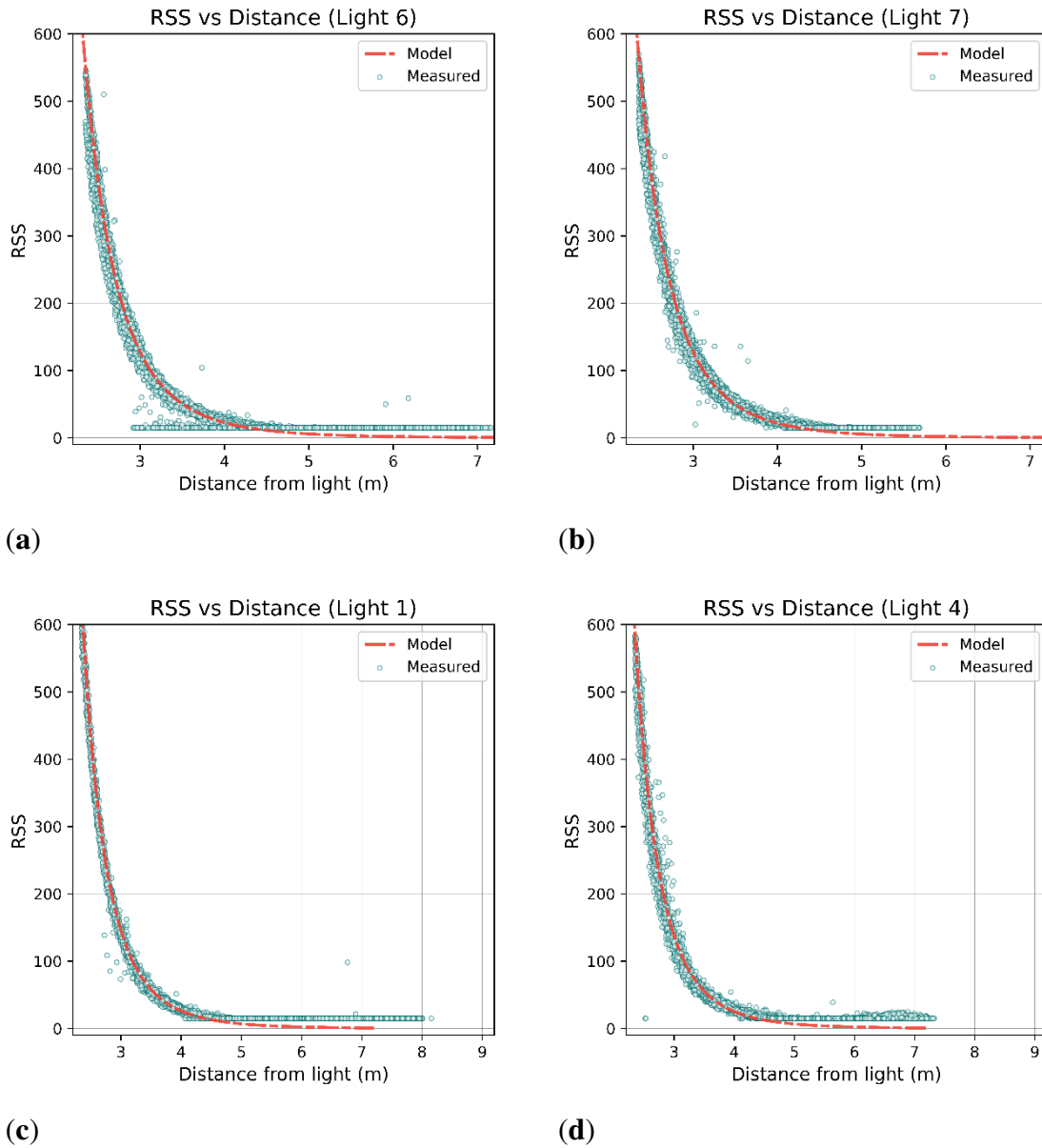


Fig. 3.17 RSS vs. distance plots for luminaires 6, 7, 1, and 4 in (a), (b), (c), and (d), respectively. The red lines show the trained Lambertian models. Luminaire 6 in (a) shows clear deviation due to the presence of the pillar; some effects can be seen in luminaires 1 and 7 (in (c) and (b)), although not as severe. Luminaire 7 is located close to the center of the room and therefore does not have as wide a range of distances compared to the others. Clamping of the RSS values can also be clearly seen in all plots, as the distance increases and readings do not follow the Lambertian model (which approaches zero).

$(2^{-1}, 2^0, 2^1, 2^2)$, with the best performing set providing the final hyperparameters for each model. The choice of cross-fold validation rather than a train/validation split was to avoid hyperparameter overtraining by constantly evaluating on a smaller subset of the data. The tuned parameters and the cross-validation score for each model are presented in Table 3.3.

After the optimum hyperparameters were determined, each model was then re-trained on the entire 80% training set (instead of cross-fold validation) and tested against the 20% test set. The localization accuracies of the various models on the 20% test set are listed in Table 3.4. Using all metrics (mean, median, RMSE), ridge regression was the best performing model with a mean of 84.4 mm. Fig. 3.19 shows the mean error as well as the CDF for each of the models.

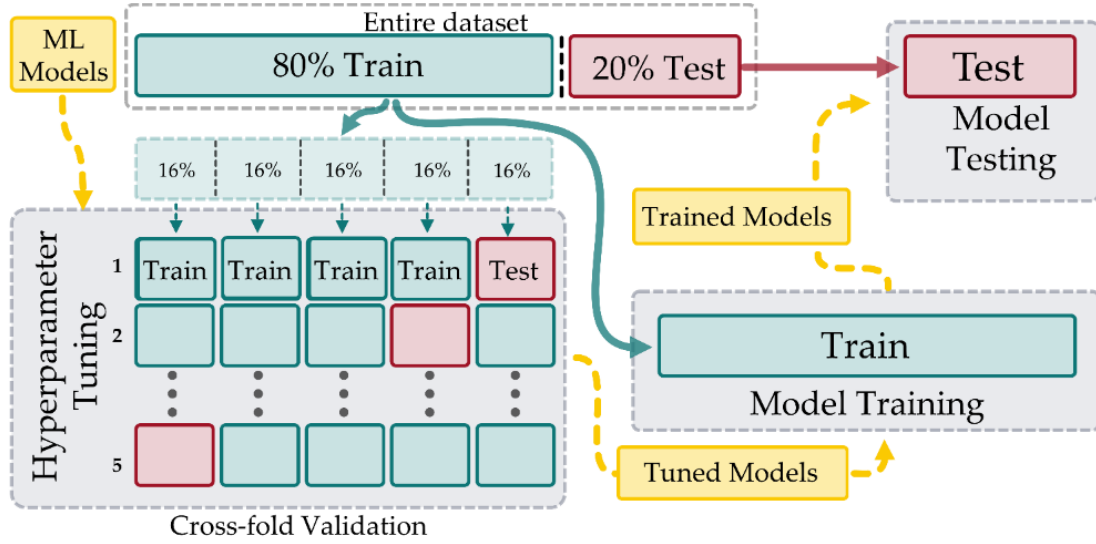


Fig. 3.18 The dataset is initially split into 80% training and 20% testing. The ML hyperparameters are then tuned using 5-fold cross-validation of the 80% training data by splitting into 5 folds and iterating 5 times through the folds, each time selecting 4 folds for training and the remaining for validation. The resulting five scores are then averaged to compute the score for a set of hyperparameters. Once tuned, the ML models are retrained using all of the 80% training data and assessed for final performance on the 20% test set.

3.5 Channel Model Based Positioning

Unlike the ML-based localization presented in the previous section, these methods rely on modeling the physical characteristics of the systems. The two methods presented both involve ranging, estimating distance from a luminaire, followed by computing a position from multiple distance estimates to fixed points using lateration.

Table 3.3 Results of the hyperparameter tuning. The listed parameters for each of the models were tuned using two rounds, the first being successive powers of 10, the second being powers of 2. The scoring metric for grid search was RMSE.

Model	Parameters	Cross Validation Score RMSE (mm)
Polynomial regression	Degree: 4	109
Ridge regression	Degree: 5; Alpha: 0.05	97.4
SVR	Kernel: poly; Epsilon: 0.05; Degree: 3; Coef0: 1; C: 10	130
RF	Estimators: 200; Max depth: 10; Min samples split: 2; Min samples leaf: 1; Max leaf nodes: 200	120
k-NN	k-nearest neighbors: 5; Weights: Distance; Distance Metric: Manhattan	114
MLP	Layers: 5; Nodes/layer: 100; Alpha (L_2 penalty): 10^{-5}	109

Table 3.4 Accuracy of the ML in mm after being retrained on the entire 80% training set and tested on the 20% test set.

Model	Median	Mean	RMSE	90th Percentile
Polynomial regression	76.3	114	395	172
Ridge regression	68.9	84.4	133	144
SVR	82.0	97.6	143	165
RF	109	125	166	220
k-NN	94.1	109	151	190
MLP	99.9	114	156	197

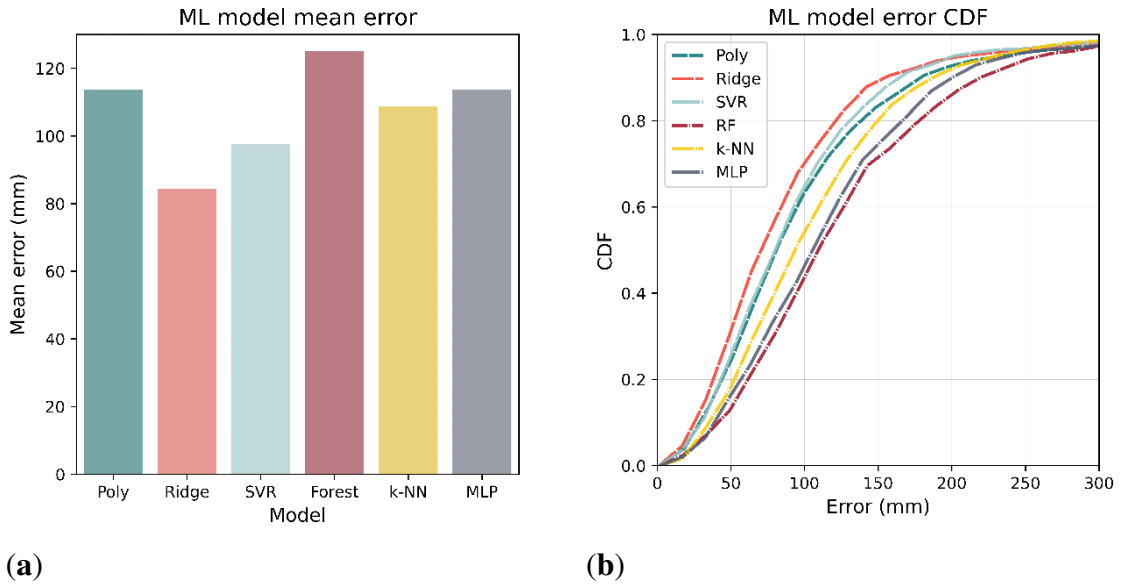


Fig. 3.19 Mean and CDF for errors of tuned models trained on the entire 80% training set. Errors were evaluated using the 20% test set, and it can be seen, in both (a) and (b), that ridge regression has the best performance, with RF being the worst.

3.5.1 Ranging

The RSS-distance model calibrated in Section 3.3 can be used to estimate the distance of the target to a luminaire from an RSS reading:

$$d_n = d_{n,0} \left(\frac{P_{r_n,0}}{P_{r_n}} \right)^{\frac{1}{m_n+3}} \quad (3.9)$$

The estimated ranges can then be compared against the ground truth recorded distances (see Fig. 3.20). As the distance from a luminaire increases, the accuracy of the estimate tends to degrade as indicated by increasing spread.

3.5.2 Positioning

Following ranging, two different algorithms were implemented for position estimation: iteration using linear least squares [109] and spring relaxation [5].

Linear Least Squares

The least-squares method involves determining the intersection of several circles centered at the x-y location of each luminaire (which was precisely measured) with radii given by the

estimated radial distance determined by ranging. A receiver at a distance d_{XY_n} (calculated by $\sqrt{d_n^2 - h^2}$ using Fig. 3.16, where h is the ceiling height) from luminaire n at location x_i, y_i will lie on a circle with the equation given by

$$d_{XY_n}^2 = (x - x_i)^2 + (y - y_i)^2 \quad (3.10)$$

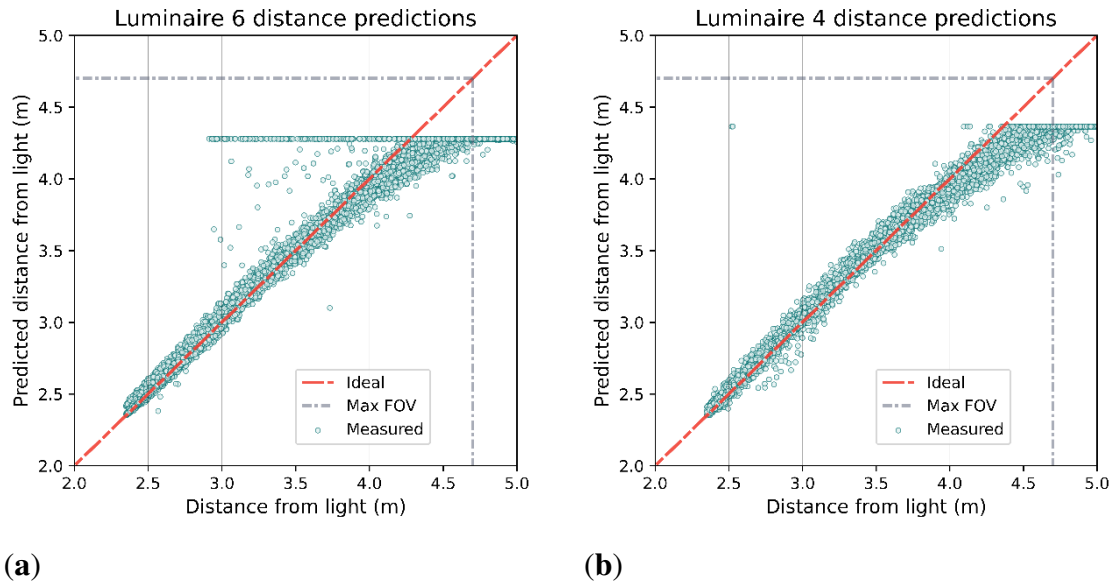


Fig. 3.20 Actual distances to luminaire vs. distances expected based on the RSS model for luminaires 4 and 6. Both luminaires demonstrate linear behavior up until the photodiodes begin to reach their FOV (gray dashed line). The pillar near luminaire 6 causes a cluster of points that clearly violate the model (a), where the predicted distance is greater than the actual distance. The predicted distance for each luminaire will be limited due to the clamping of RSS (an example for luminaire 4 in (b)).

For a localization system with N luminaires, there will exist N such equations. Expansion, subtraction, and rearrangement of these equations leads to the following [109]:

$$\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_N - x_1 & y_N - y_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 + d_{XY_1}^2 - d_{XY_2}^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 + d_{XY_1}^2 - d_{XY_3}^2 \\ \vdots \\ x_N^2 - x_1^2 + y_N^2 - y_1^2 + d_{XY_1}^2 - d_{XY_N}^2 \end{bmatrix} \quad (3.11)$$

This is of the form $\mathbf{Ax} = \mathbf{b}$, where $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ are the luminaire locations and $d_{XY_1}, d_{XY_2}, \dots, d_{XY_N}$ are the estimated distances based on ranging. Given

inherent signal and system noise, the circles will not overlap perfectly, and a least-squares approximation can be used:

$$\bar{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b} \quad (3.12)$$

Spring Relaxation

Spring relaxation utilizes the same ranging process based on the Lambertian model in (3.6) but estimates the final position via an iterative process that simulates hypothetical springs connecting the estimated position with each N luminaire [5]. The relaxed length of each spring is the estimated ranging distance given by (3.9). At each possible x, y location, a force can be calculated based on the extension/compression of each spring. Beginning with an initial location estimate given as the mean position of the N luminaires with highest RSS readings, the algorithm iteratively refines the position by moving a small step in the net force generated by the hypothetical springs. After each new position the forces are then recalculated, and the next step direction is determined. The algorithm is run until it converges on a solution, determined by the next step being below a set threshold.

To determine N , the optimum number of luminaires to use for lateration or the spring relaxation method, 5-fold cross validation was again implemented using the 80% training set. For both methods, it was found that $N = 4$ was the best number of luminaires. The channel models were then calibrated using the entire training set and validated against the test set. Fig. 3.21 gives the mean errors as well as CDF for both methods along with Ridge regression, the best-performing ML positioning technique, for comparison. It can be observed that among the two lateration techniques, the spring relaxation-based method is more accurate (also seen in Table 3.5). However, both are outperformed by the Ridge regression-based positioning.

Table 3.5 Accuracy of the model-based methods after being retrained on the entire 80% training set and tested on the 20% test set. The RMSE for both is greater than the 90th percentile errors, indicating that there are some large errors. All values are in mm.

Model	Median	Mean	RMSE	90th Percentile
Spring relaxation	106	150	292	258
Least squares	130	248	450	590

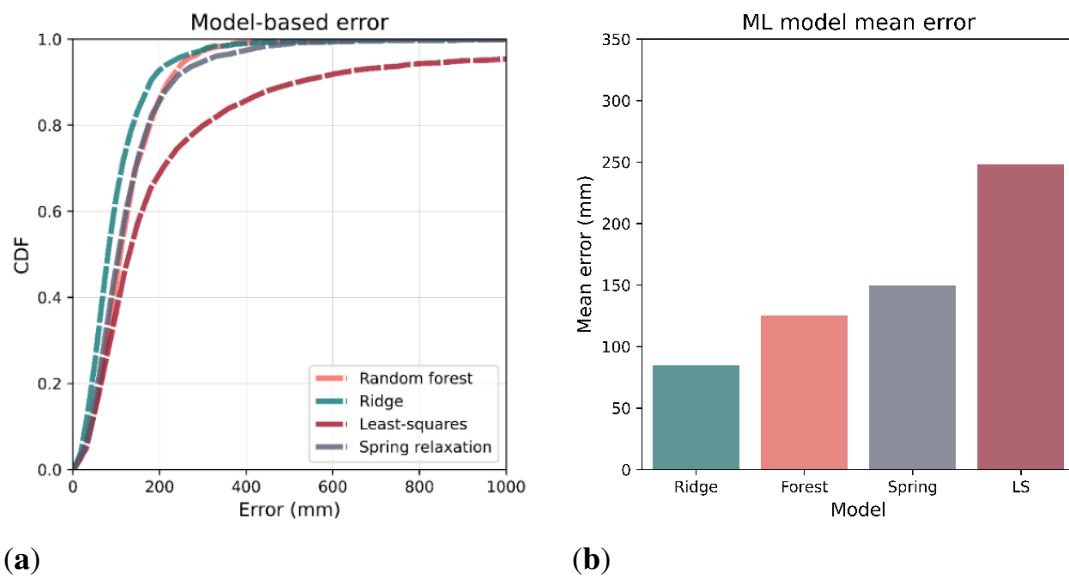


Fig. 3.21 CDF and mean of localization errors for channel model-based positioning are benchmarked in (a) and (b), respectively. Spring relaxation was the better performing of the model-based techniques, although its performance was slightly lower than that of RF, the worst-performing ML model. The least-squares method was considerably worse than any of the other methods. The models were trained/calibrated on the entire 80% training set and tested on the 20% test set.

3.6 Importance of Data Size

3.6.1 Training

To investigate each ML- and channel-based model's dependence on the size of training data, the models were repeatedly trained or calibrated on progressively larger fractions of the training data after they had been randomly shuffled. Each model was trained on 20 training samples and its performance evaluated on the test set; this was then repeated for 40, 60, etc. The plots for the number of training points vs. mean test accuracy are shown in Fig. 3.22. If the training data are restricted to just 100 samples, SVR outperforms all other methods. In fact, for training points 45–1560, SVR in general has the lowest mean error. As more training data are used, ridge regression becomes the best performing model; however, in this study it first became the best-performing model at around 1580 training points.

Fig. 3.23 shows the same results from the same analysis but using a step size of 2 with a maximum 100 training points. With 31 training points, the mean accuracy of spring relaxation is 258 mm. At the same number of training points, the mean accuracy of ridge regression is 1562 mm, almost an order of magnitude greater.

The crossover points observed in Figs. 3.22 and 3.23 should be interpreted as approximate empirical transition regions within this study, rather than as definitive thresholds. Confidence intervals were not calculated for the training-size curves, so any exact claim about the location of a crossover point should ideally be supported by additional statistical analysis. This is consistent with the approach used in Section 3.6.2, where formal statistical testing was applied when comparing model performance directly. The purpose of the present analysis is therefore not to make a generalized claim about which model is universally best, but to show that comparative accuracy depends strongly on training data volume. Within the present dataset, model-based methods were more competitive at very low training volumes, whereas machine-learning methods became more competitive as training volume increased, which aligns with intuition.

3.6.2 Testing

Having a large dataset allows for robust comparisons between models as the confidence in each model's accuracy can be ascertained using hypothesis testing. As an example, we can compare the performance between ridge regression and random forest using a paired t-test on the errors for each model at test positions. The null hypothesis, assuming no true mean difference between the performance between the models, would need to be rejected before we can confidently conclude a model's superiority. Fig. 3.24a shows the 95% confidence interval

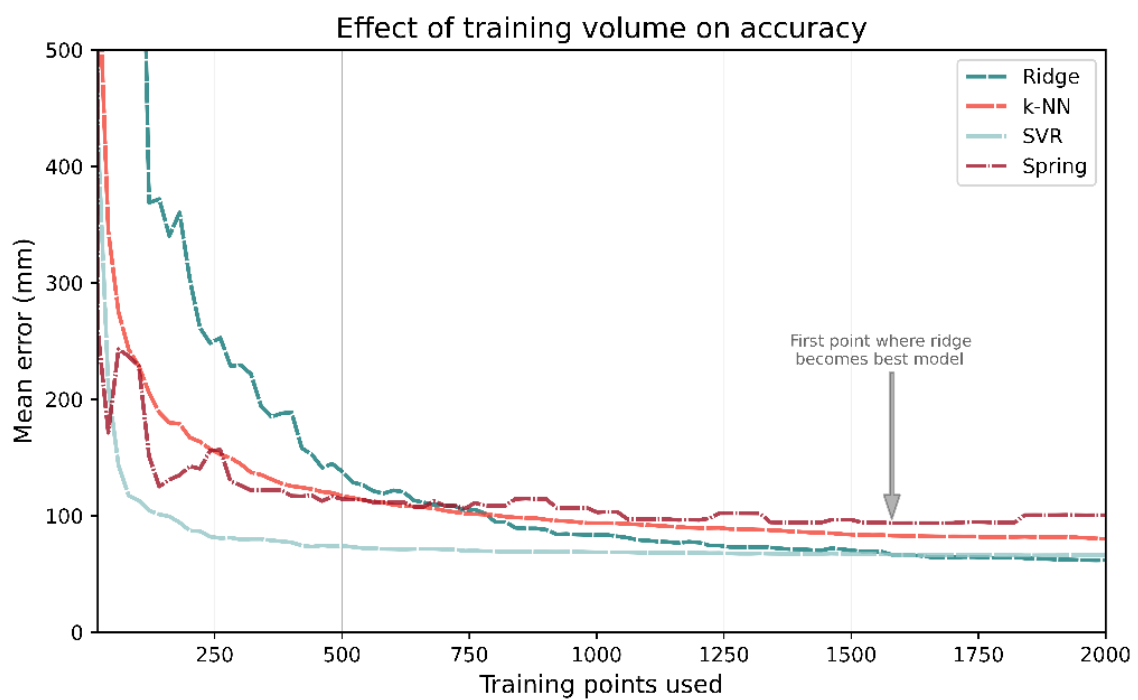


Fig. 3.22 The number of training points used has a large impact on which model has the best performance. Despite being the best performing model when trained on the entire dataset, ridge regression requires a significant amount of training data. In this study, ridge regression first became the best-performing model at around 1580 training points.

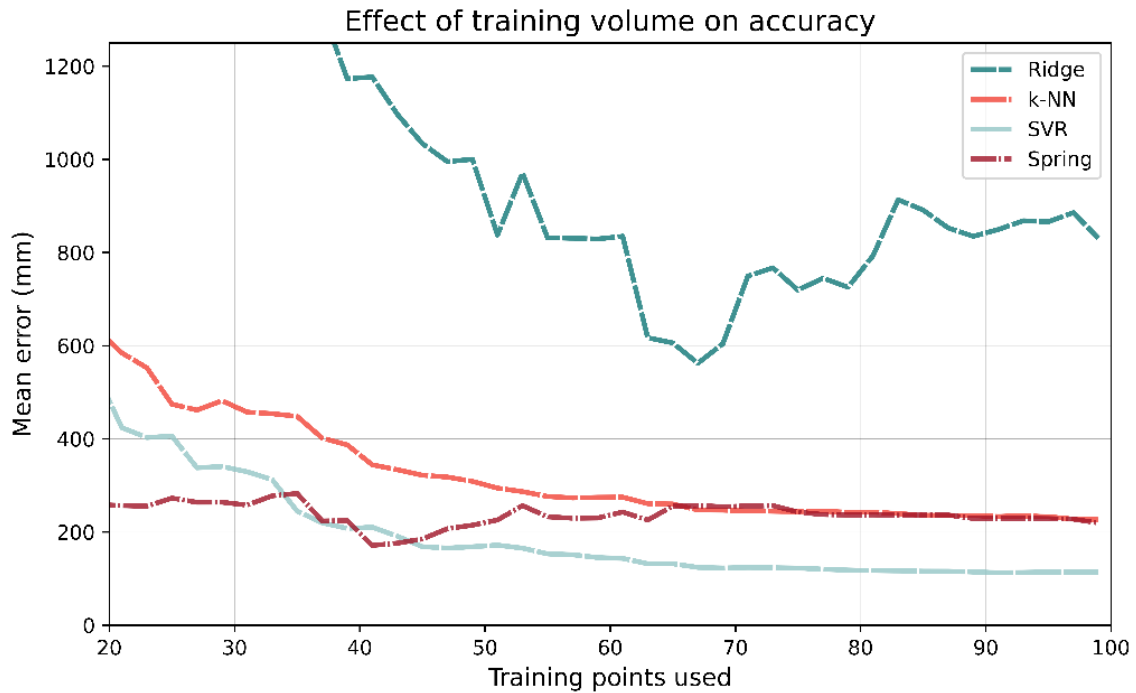


Fig. 3.23 For very small training sets, approximately fewer than 33 points in this study, the spring-relaxation model-based approach outperformed the machine-learning methods.

for the difference in the positioning error and Fig. 3.24b displays the p -value for the t-test. Since the two models were the best and worst performing, the results are as expected. Note that 0 is not included within the 95% confidence interval and the p -value is well below the 0.05 threshold. This rejects the hypothesis that there is no difference in model performance.

If we instead compared models whose performances are more similar such as SVR and polynomial regression, with SVR having better performance on the entire test set (Table 3.4), the results are not as conclusive. The confidence interval for the performance interval (Fig. 3.25a) contains 0 most of the time. The t-test p -value only dips below the 0.05 significance value for a short interval and then increases again. This casts doubt on the superiority of SVR over polynomial regression.

The comparison of k-NN and MLP shows that they exhibit different behavior. k-NN is the better performing model (as shown in Table 3.4); however, a confidence interval for the mean difference in model performance (Fig. 3.26a) includes 0 until around 600 test points are assessed. This results in the t-test failing to reject the hypothesis that there is no difference in model performance until 600 test points. This demonstrates that a large dataset is necessary not only for training models but also for robust performance comparisons.

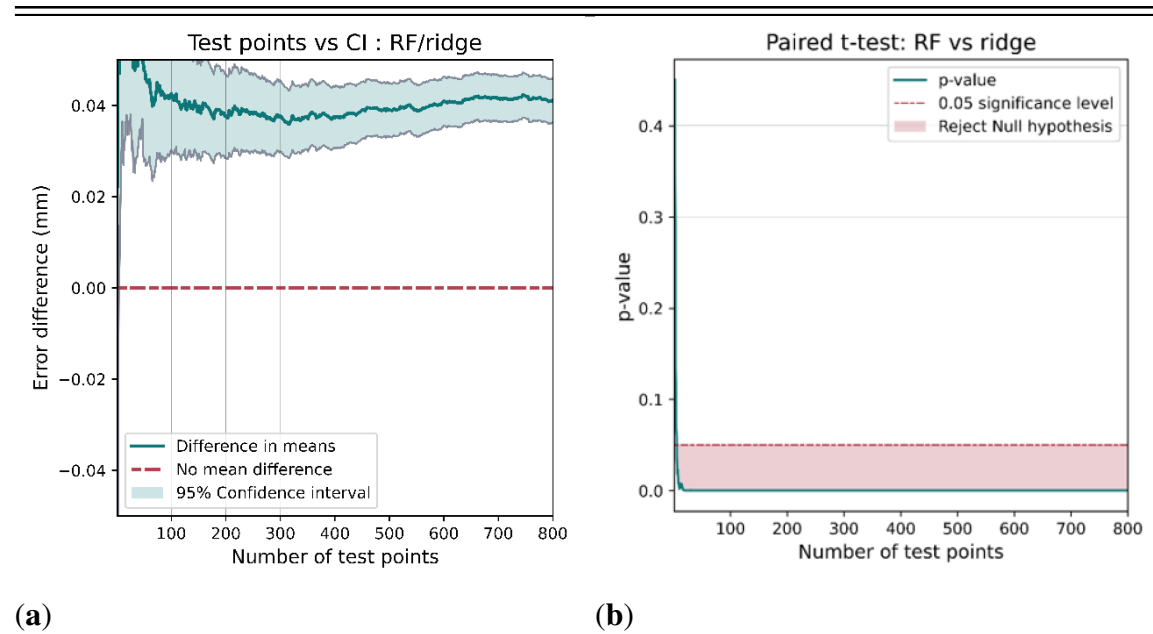


Fig. 3.24 There is a clear difference in the performance of the RF and ridge models, shown by 0 not being contained in the 95% confidence interval (a) and the p -value being clearly below the 0.05 significance level (b).

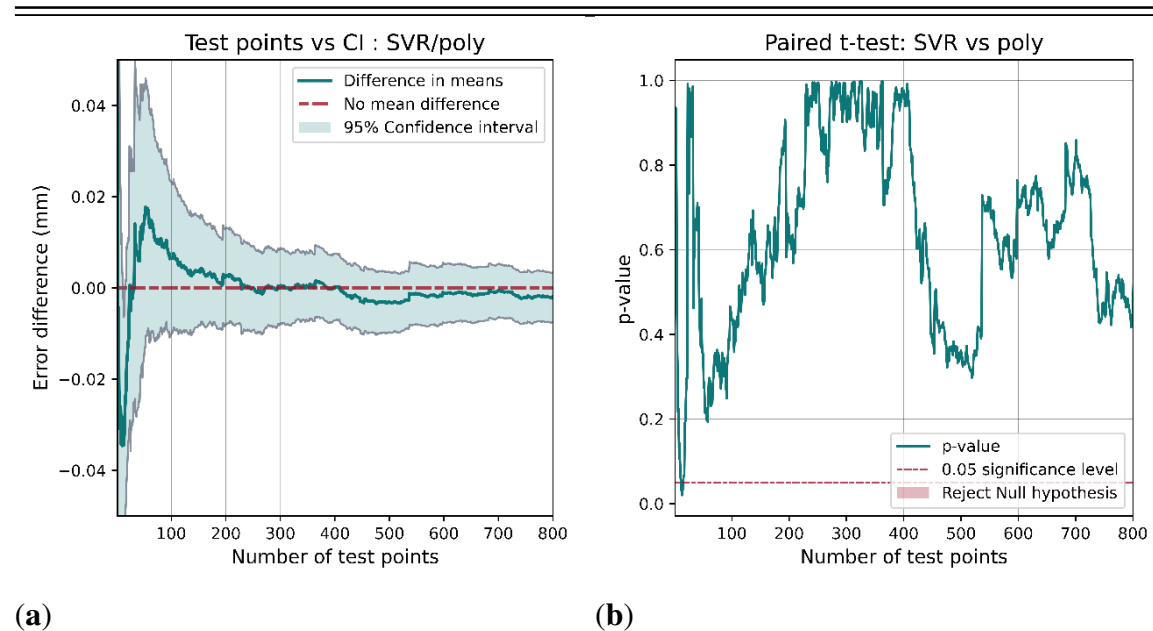


Fig. 3.25 When comparing SVR and polynomial regression, the performance of the models is more similar. The confidence interval (a) includes 0 a for a large proportion of the data, and the p -value (b) only drops below the 0.05 significance level for a brief period with very few samples.

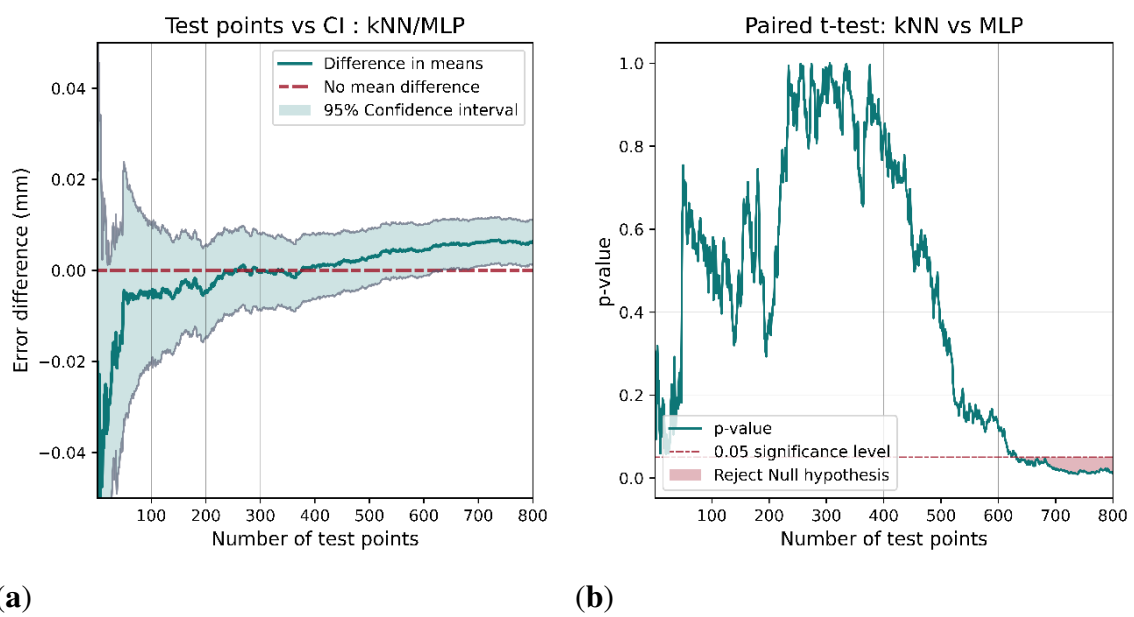


Fig. 3.26 The t-test between the k-NN and MLP models does indicate a performance difference but only after more than 600 test points have been assessed. This demonstrates the need for a large dataset when making robust comparisons between models. If less than 600 points had been used for testing, then the confidence interval (a) would include 0, the p -value (b) would be larger than 0.05, and the hypothesis that there was no difference in performance would not be rejected.

3.7 Conclusions and Future Work

An autonomous data collection method for collecting RSS data for a VLP system was developed. This is the first such system reported in the literature to allow the automated collection of a large VLP dataset over a wide area. The experimentally collected RSS dataset is, to the best of the authors' knowledge, the largest reported experimental RSS fingerprint dataset for photodiode-based VLP in the surveyed literature at the time of publication, comprising 7344 locations over 43.5 m². It has been made available online for other researchers. Examples of performance analyses of different VLP techniques using this dataset were presented. Memoryless positioning techniques using both ML- and model-based ranging and lateration were investigated. Ridge regression was found to be the most accurate method for localization, achieving mean and 90th percentile errors of 84.4 and 144 mm. However, if only very limited training data are available, with the crossover occurring at approximately 33 points in this study, model-based positioning using spring relaxation performs most accurately with a mean of 258 mm.

It should be noted that RSS was used as the feature for the ML algorithms. Feature generation, which derives new variables from the raw RSS values, such as powers and interactions [110] or RSS ratios [111], can improve localization accuracy. It is also possible to train multiple models on subsets of the luminaires, then aggregate the predictions [112] or frame the positioning as a classification problem [113]. These investigations provide scope for future research. Furthermore, decoupling the x, y prediction into two single-target methods results in independent predictions and the relationships between the variables are lost and cannot be exploited. Analysis of this problem is beyond the scope of this paper but may be worthwhile as only limited cases of decoupled models have been investigated [113, 114]. A separate, yet similar issue is the method of loss function used to train multi-target regressors. ML positioning models are often trained using one-dimensional loss functions. However, since positioning deals with two-dimensional data, the model should be trained to minimize the Euclidean error of the positioning system. Future research can investigate the utilization of a 2D custom loss function for ML training. The analysis presented in this paper was restricted to data collected at a constant height and with obstacles removed from the test area. Future work could investigate how 3D positioning or changing environments affect the findings presented here.

Author Contributions: Conceptualization, T.G. and F.A.; methodology, T.G. and F.A.; software, T.G.; formal analysis, T.G.; investigation, T.G.; writing: original draft preparation, T.G.; writing: review and editing, F.A., M.L. and F.N.; supervision, F.A., M.L. and F.N.; funding acquisition, F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The VLP data collected is available for download at <https://github.com/tyrel-glass/Public-VLP-Dataset>. For data citation, please cite this article.

Conflicts of Interest: The authors declare no conflict of interest.

Chapter 4

Fusion of Visible Light Positioning and Odometry for Robot Navigation

Preface

This chapter is based on the manuscript:

Glass, T., Konings, D., Legg, M., & Alam, F. *Fusion of Visible Light Positioning and Odometry for Robot Navigation*. Under review at *IEEE Sensors Journal*.

The candidate was the primary author of this work. **Author contribution:** I designed the study, performed the data collection and analysis, and drafted the manuscript. Co-authors provided supervision, guidance, and editorial input.

The manuscript has been formatted to match the style of this thesis. Minor edits have been made for consistency in notation, referencing, and formatting. The scientific content remains unchanged.

Linking Overview

Chapter 3 established a reliable, large-scale dataset and showed how Received Signal Strength (RSS)-based positioning methods behave under controlled conditions. The next step is to test whether those insights carry into navigation, where odometry drift, actuator imperfections, and temporal dependencies are unavoidable. This chapter therefore shifts the focus from static positioning to dynamic, closed-loop robot control, directly addressing the thesis research questions on single-mode VLP navigation and multimodal fusion.

The literature reviewed in Chapter 2 shows that most navigation results use camera-based or cooperative systems, with comparatively few studies demonstrating photodiode-only navigation in real-time. In response, this work adopts a low-cost differential-drive platform with a photodiode receiver and wheel encoders, aligning with the thesis emphasis on resource-constrained robots. The experiments generate a trajectory-level dataset that complements the static fingerprints in Chapter 3 and provides a ground-truth basis for evaluating long-horizon behavior.

The scope of the evidence presented in this chapter should, however, be read carefully. The experiments were conducted in a small, controlled indoor workspace using planar motion, fixed receiver height, and a rigidly mounted photodiode receiver on a simple differential-drive robot. Accordingly, the results demonstrate feasibility and comparative performance under these conditions, rather than establishing robustness to significant tilt, cluttered industrial environments, or broader cross-platform deployment.

Within this setting, the chapter evaluates single-mode baselines, VLP-only and encoder odometry, alongside a Long Short-Term Memory (LSTM)-based fusion model that integrates absolute RSS measurements with motion continuity over time. The analysis focuses on drift, recovery from perturbations, and stability across long trajectories, including capture events. The aim is to establish whether learned temporal fusion improves over the single-mode references on a low-cost robot, rather than to exhaustively compare all model-based filter variants. In this measurement setting, heading is inferred from successive VLP positions rather than observed directly, making a fair EKF / UKF benchmark itself a non-trivial formulation problem that remains open for future work.

The results complete the thesis narrative by showing how VLP can move from offline estimation to robust navigation, building directly on the measurement and benchmarking foundation established in the preceding chapter. More specifically, they show that photodiode-based VLP and VLP-odometry fusion can support near-level, planar indoor navigation in the tested environment, while leaving broader generalization questions open.

4.1 Introduction

Robots capable of indoor navigation hold immense utility across various sectors, including manufacturing, logistics, healthcare, and warehouse management [6]. The ability to navigate autonomously or semi-autonomously enables robots to perform tasks efficiently and effectively, saving time and labor costs.

Odometry continues to be widely used due to its simplicity, low cost, and proprioceptive nature, which allows it to function without reliance on external infrastructure [115]. It

estimates linear and angular displacement from wheel rotations, enabling basic navigation on a broad range of mobile platforms.

Visible light positioning (VLP) offers a complementary exteroceptive solution. It leverages existing lighting infrastructure to provide centimeter-level positioning accuracy at low cost [3, 23, 66]. Its inherent compatibility with smart building ecosystems makes it attractive for deployment in environments such as warehouses, hospitals, and manufacturing facilities.

Odometry and VLP differ fundamentally in their temporal characteristics. Odometry is a memory-based approach that ensures temporal smoothness and continuity in position estimates. However, it is inherently susceptible to cumulative error arising from encoder imperfections, variable surface friction, payload variations, battery voltage fluctuations, temperature changes, and mechanical wear [44]. These uncertainties accumulate over time, resulting in drift and degraded navigation accuracy. In contrast, the base form of VLP is memoryless, determining position directly from instantaneous signal measurements without reliance on prior states. Consequently, it is immune to cumulative drift but can exhibit noise and discontinuities in dynamic environments [3]. The complementary nature of these two modalities makes their fusion conceptually intuitive.

This paper mitigates the inherent limitations of each single-mode method by fusing VLP and odometry, resulting in a more accurate and robust indoor navigation solution. A Long Short-Term Memory (LSTM) network is used for multi-modal fusion, learning nonlinear error dynamics from sequential measurements.

To the best of the authors' knowledge, this work is among the first to demonstrate that a photodiode-based VLP system alone can estimate both the planar position and heading of a mobile robot for real-time navigation. While prior VLP studies have primarily focused on position estimation or relied on additional sensing modalities to recover orientation [7, 10, 18], the present study shows that a purely PD-based VLP architecture can support full 2D pose estimation suitable for closed-loop motion control. Accordingly, this paper makes the following contributions:

1. The first photodiode-based VLP system shown to estimate full 2D pose (planar position and heading) for real-time robot navigation without auxiliary orientation sensors;
2. A unified sensor-fusion framework integrating encoder odometry with VLP measurements, with a data-driven LSTM fusion strategy developed and evaluated for robust navigation;
3. A comprehensive experimental evaluation that benchmarks single-mode (VLP-only and encoder-odometry-only) and LSTM-based multimodal fusion methods with respect to position accuracy, heading accuracy.

4. An experimental evaluation of the robustness of the proposed LSTM-based fusion method under disturbances using captured robot trajectories.
5. An assessment of the feasibility of deploying the proposed multimodal fusion algorithm on resource-constrained edge devices.

4.2 Related Work

4.2.1 Visible Light Positioning (VLP)

Visible light positioning exploits existing LED lighting using either photodiode receivers or image sensors. In photodiode-based systems, Received Signal Strength (RSS) is the primary feature because it enables low-cost hardware without requiring infrastructure synchronization. Non-RSS methods such as angle-of-arrival (AOA), time-of-arrival (TOA), and time-difference-of-arrival (TDOA) have also been explored in VLP [35, 28, 116, 27]. However, these methods typically require tighter synchronization, multi-element receivers, or complex calibration. This reduces their practicality for low-cost, real-world robot deployments. Most RSS-based PD methods rely on model-based trilateration [4] or fingerprinting [71]. These approaches offer low cost and low latency but are sensitive to receiver tilt and multipath effects [5]. Camera-based rolling-shutter systems can decode LED identifiers and achieve centimeter-level accuracy in controlled environments, albeit at significantly higher computational cost [3, 13]. Deployment-scale calibration and mapping pipelines have been explored using robots or handheld rigs to estimate LED layouts [17, 16]. For robotics, camera-based ROS integrations demonstrate accurate localization and path execution, whereas PD-only navigation has mostly been shown on idealized platforms or in qualitative demos [10, 8, 7, 117]. Evidence remains limited that low-cost PD-only systems can recover full 2D pose and support closed-loop navigation with independent ground truth.

4.2.2 Odometry and Dead Reckoning

Wheel-encoder odometry remains the default localization method for many mobile robots because it is simple and inexpensive, but it accumulates systematic and stochastic errors that grow with path length and turning. Classical calibration and error modeling (e.g., UMBmark) show path-dependent drift that cannot be eliminated without external references [44]. This motivates fusing odometry with absolute measurements such as VLP, especially for low-cost robots that cannot carry LiDAR or vision stacks [6].

4.2.3 Sensor Fusion for Localization

Fusion approaches combine short-term odometry with absolute measurements to reduce drift. In the VLP domain, tightly coupled VLP/IMU systems and data-model hybrids adapt filter confidence under tilt and blockage, improving robustness over standalone positioning [18, 19]. Robotic systems have fused VLP with LiDAR-SLAM and wheel odometry in ROS to enable navigation, and hybrid RF+VLP schemes provide coarse-to-fine localization [118, 9]. Direct evaluations of learned sequence models for encoder+VLP fusion on simple robots remain scarce. These multi-sensor pipelines increase system complexity and calibration burden, leaving PD-only navigation and fusion comparatively underexplored.

4.2.4 Robustness to Perturbations

Robustness has typically been evaluated under measurement degradations such as LED occlusions, receiver tilt, or multipath, with filtering or graph-based estimators mitigating these effects [18, 19]. Some robotic VLP systems report dynamic tracking, but few quantify recovery when the robot state is externally perturbed during closed-loop navigation or provide independent ground truth in such tests [10, 8]. This leaves a gap in understanding how VLP-odometry fusion behaves under physical disturbances that break motion model assumptions.

4.3 Experimental Setup

4.3.1 Navigation Environment

The experimental setup (Figure 4.1) consisted of a custom-designed robot constructed using off-the-shelf components. It operated on a common floor panel base, which provided a uniform and reproducible surface. A bespoke two-axis CNC platform was mounted onto this base and aligned using two fixed mounting points. The CNC was used to collect the VLP training dataset and to calibrate the HTC Vive system which served as the ground-truth reference.

The VLP system comprises four transmitters constructed from consumer-grade LED luminaires. The luminaires and two Vive *base stations* were mounted on an aluminum frame, which was suspended above the floor using four tripods.

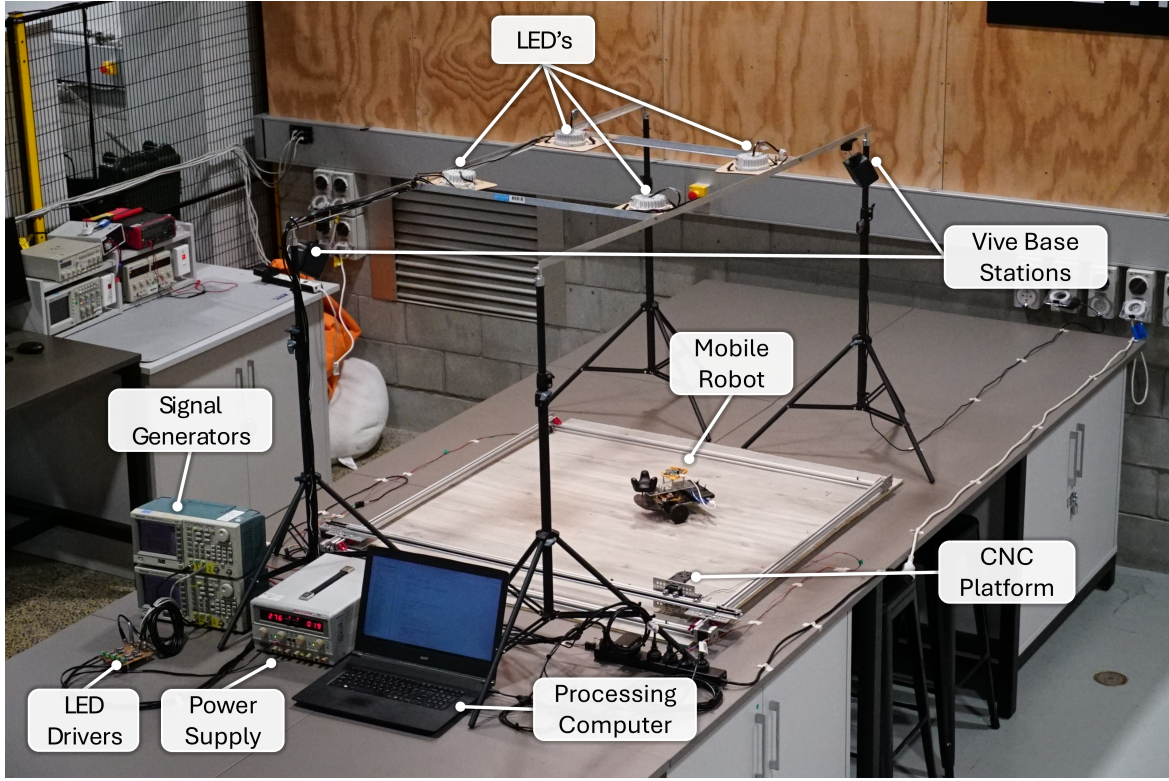


Fig. 4.1 Experimental setup overview for data collection and navigation experiments; all components are labeled in the figure.

The luminaires were positioned at the corners of a rectangular configuration:

$$\begin{aligned} &(0.145 \text{ m}, 0.145 \text{ m}), \quad (0.145 \text{ m}, 0.925 \text{ m}), \\ &(0.905 \text{ m}, 0.925 \text{ m}), \quad (0.905 \text{ m}, 0.145 \text{ m}), \end{aligned}$$

with all luminaires mounted at a uniform height of 1.040 m above the floor surface. Figure 4.2 shows the layout of the experimental setup.

A 2-axis belt-driven CNC platform was custom built, spanning 1.0 m × 0.9 m and actuated by NEMA 17 stepper motors. An OpenBuilds Black-Box Motion Control System (MCS) running Grbl firmware was controlled from a Python script via UART. Once calibration was complete, the CNC actuator was moved out of the way, and the robot operated freely within the CNC frame.

4.3.2 Mobile Robot Design

A simple differential wheel robot was custom-built for testing the proposed navigation strategies. It can be seen in Figure 4.3 with both the onboard Vive tracker and the VLP

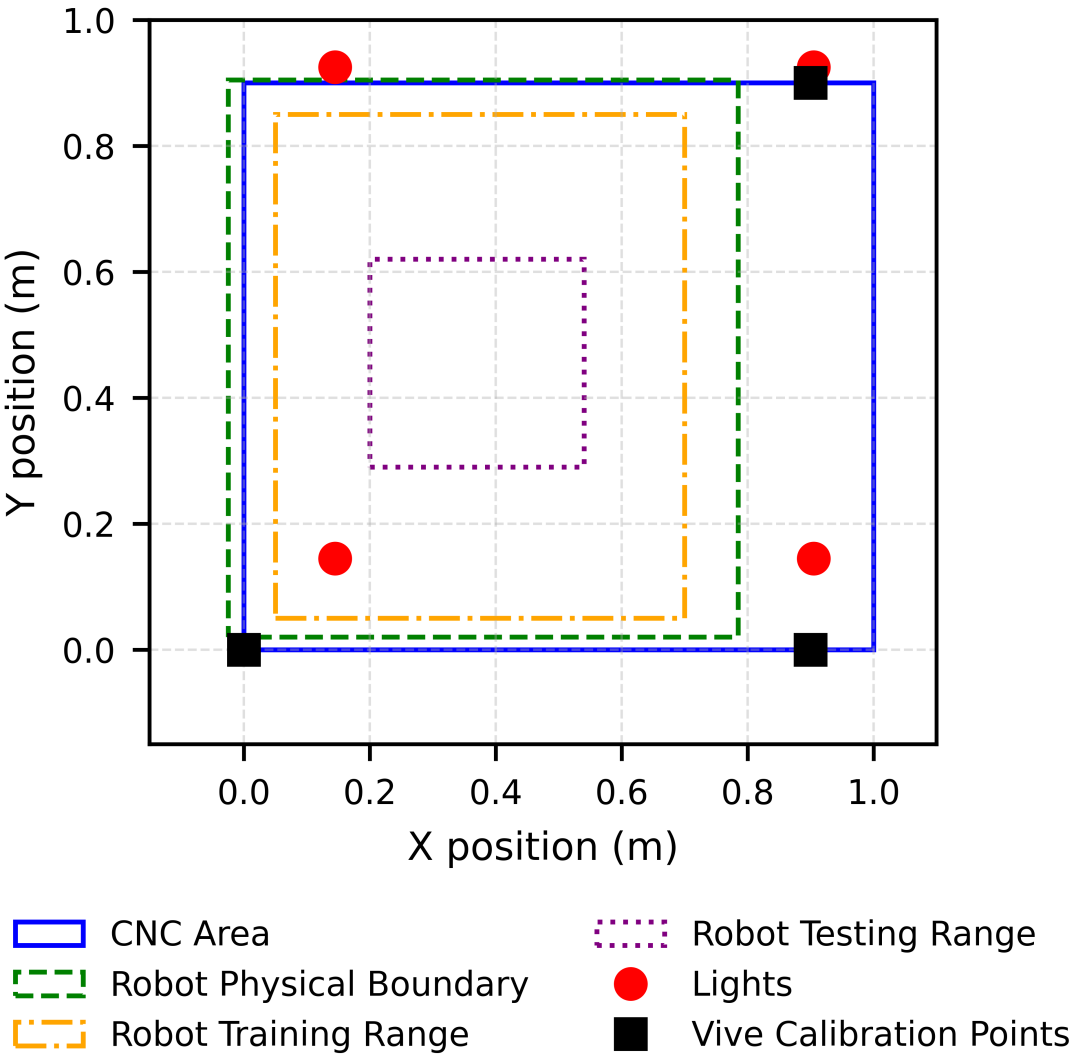


Fig. 4.2 Geometric layout of the workspace, including LED positions, Vive Tracker calibration points, and the CNC/robot operating regions (training, live, and capture areas).

receiver board. The robot uses two metal DC gear motors with integrated quadrature encoders, driven by a dual motor driver. An Arduino Nano V3.0 handles onboard control, while an ESP8266 module provides wireless TCP communication with a remote computer. Each motor incorporates a 34:1 metal gearbox for reliable locomotion. Further details and component list can be found online at [119]. The robot measured 0.21 m in length and 0.175 m in width. The maximum distance from any corner of the robot to the midpoint between the two wheels was 0.14 m. Since the robot rotated about this midpoint, a minimum clearance of 0.14 m was required to rotate in place without collision.

All navigation experiments were conducted on a flat, rigid floor-panel base, with the VLP receiver mounted rigidly at the robot center. No direct pitch or roll sensing was logged during live trials, and no active tilt compensation was applied. The experiments therefore assumed near-level operation, with any incidental receiver tilt expected to remain small for the tested planar setup. Even so, unmeasured tilt would alter the photodiode incidence angles relative to the luminaires, perturb the RSS values, and therefore affect both the VLP position estimates and the heading estimate derived from successive VLP positions. The present results should therefore be interpreted as applying to near-level planar operation rather than to scenarios with substantial receiver attitude variation.

Onboard Control and Odometry

The robot operates using two primary movement types:

- **Forward motion:** Both wheels turn in the same (forward) direction for a set number of encoder counts, with the controller matching wheel velocities to maintain a straight trajectory.
- **In-place rotation:** Wheels turn in opposite directions for a set number of encoder counts, with the controller matching wheel velocities to change heading without translation.

Navigation is performed by sequentially reaching predefined target points $P_1, P_2, \dots, P_n$. Given physical constraints, a threshold distance d_t defines successful target acquisition.

The robot's state vector is given by:

$$\mathbf{x} = [x, y, \theta]^T, \quad (4.1)$$

where (x, y) represents the robot's position, θ is its orientation and T is the transpose operation.

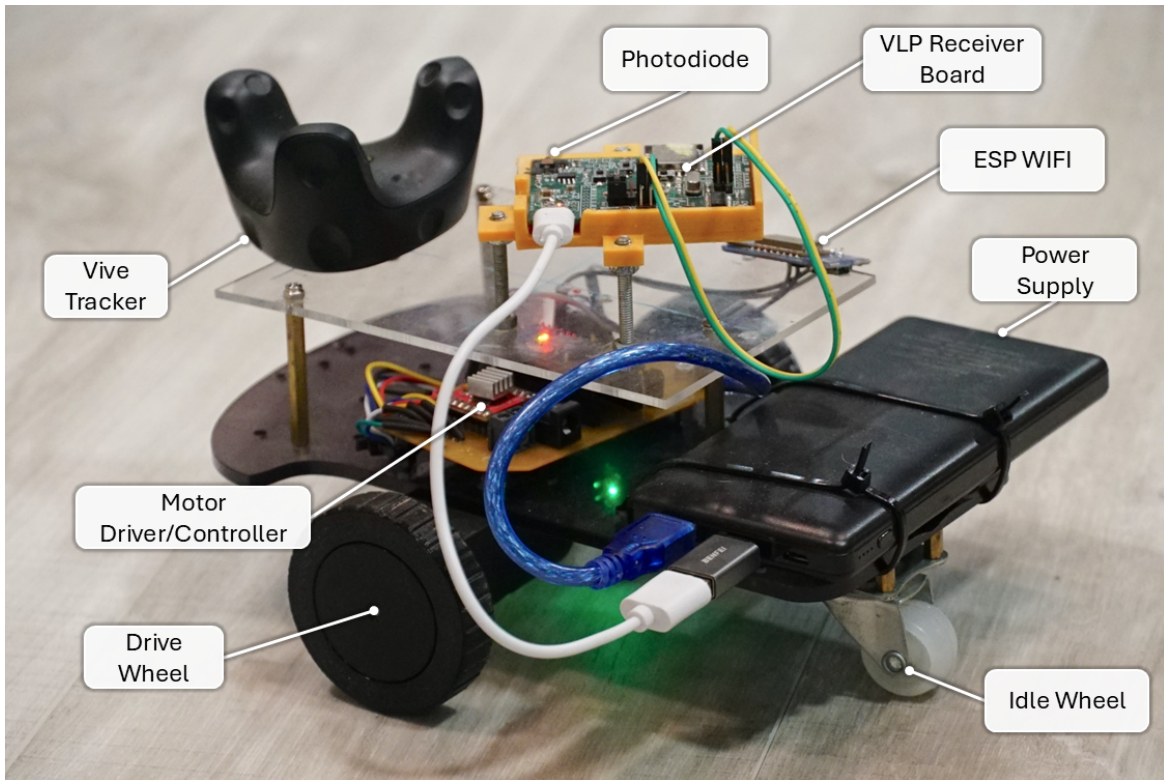


Fig. 4.3 Robot platform with labeled components and a differential-drive layout (two powered wheels with a passive caster).

The microcontroller implements a control system that uses encoder-based feedback and proportional speed adjustments. Real-time wheel rotation is measured by the quadrature encoders driving interrupts and a PWM-based motor control dynamically adjusts speed to maintain balance. The controller also minimizes the difference in total rotation between the wheels. At a given point of time, if one wheel has traveled a larger rotational angle than the other despite currently having equal speed, the power will be reduced. Each wheel also stops independently once the total target encoder count has been reached.

A preliminary calibration was performed to map encoder steps to physical movement magnitudes. A series of manually issued commands directed the robot to move or rotate by known encoder values. For each command, the actual displacement or rotation was measured using the Vive ground truth system. The details of the ground truth system are provided in Section 4.3.4. Based on this procedure, the preliminary calibration yielded an estimate of 5.28 encoder steps per degree of rotation and 4.45×10^3 encoder steps per meter of translation. These estimates were used to collect the training data and were later refined through a more rigorous calibration procedure (see Section 4.4.2).

4.3.3 VLP System Design and Calibration

The VLP system employs a fingerprint-based approach that uses RSS as the feature vector for positioning (see Figure 4.4).

The transmitters are built from consumer-grade 10 W LED luminaires (model REX100DL WHWWDIM) [4]. Each luminaire is driven by a square-wave carrier at a unique frequency—15 kHz, 17 kHz, 20 kHz, and 23 kHz—generated via a function generator and delivered through a modulator circuit powered by a 30 V supply. The receiver performs spectral decomposition to identify the contributing sources based on their characteristic frequency components (see Figure 4.4).

The receiver tag consists of a photodiode, a transimpedance amplifier, and an active high-pass filter to remove both 60 Hz power-line flicker and the DC component of ambient light. A final inverting amplifier scales the signal for analog-to-digital conversion (ADC), which is performed at 50 kHz by a microcontroller. More details of the receiver tag can be found in [5]. Each VLP measurement comprises 512 samples, corresponding to a 10.4 ms acquisition window.

Samples are transferred via UART to an ESP8266 module, which transmits the data over WiFi using TCP to a Python script on a host computer. On the host, signals are centered, normalized, Hanning-windowed, and transformed via FFT. LED amplitudes are extracted by locating the peak magnitude within 8 FFT bins of each carrier frequency. Please refer to Alam et al. [69] for details on recovering square-wave-modulated signals with a photodiode-based receiver and using them as RSS feature vectors.

VLP Fingerprint Dataset

Two datasets were acquired using the CNC platform by mounting the VLP receiver board on the actuator (see Figure 4.5).

Training Set:

- 1517 spatially regular points arranged in a 0.025 m grid across the full area.
- At each point, 10 measurements were collected and the resulting RSS readings were averaged.

Validation Set:

- 528 locations derived from a 0.040 m base grid, each displaced by a small random offset in both x and y directions. This perturbation was used to discourage overfitting to a regular grid.
- Edge locations were clamped to ensure the entire set remained within platform bounds.

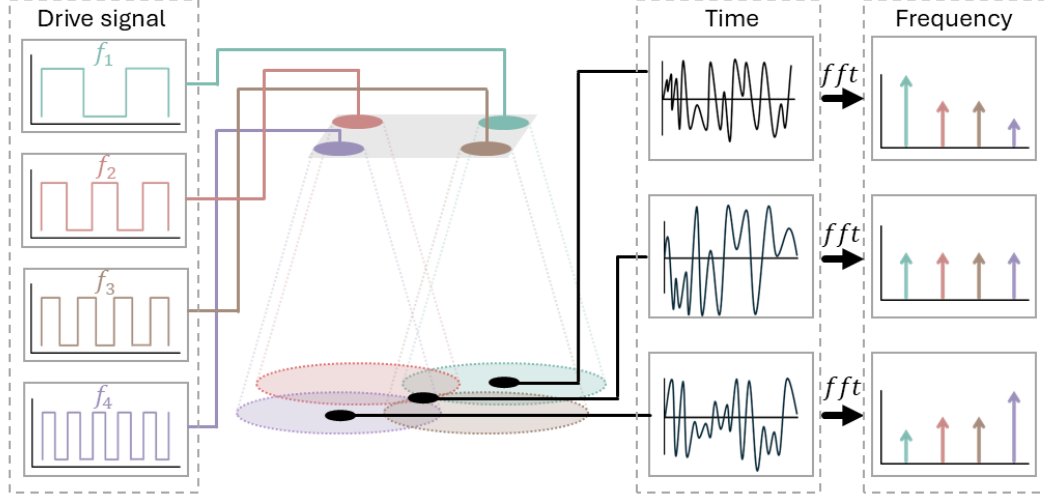


Fig. 4.4 VLP modulation and RSS extraction: each LED is driven at a distinct square-wave carrier, and the receiver separates these in the FFT to recover per-LED RSS (used for distance inference).

VLP Model Training

The model is a feedforward neural network that maps 4-element RSS vectors to 2D Cartesian positions (x, y) . The model consists of two 32-unit ReLU layers with L2 regularization, batch normalization, and 0.1 dropout, followed by a linear 2D output layer. Training used the Adam optimizer with an exponentially decaying learning rate $\eta_{\text{epoch}} = 0.01 \cdot 0.995^{\text{epoch}}$.

Training continued for up to 10,000 epochs with a batch size of 32 and early stopping (patience = 30) to halt training when no further improvement on the validation set was observed.

Validation Performance

The final VLP model was evaluated on the validation dataset. The median error was 0.0188 m, with an RMSE of 0.0237 m. The 90th-percentile (P_{90}) error remained under 0.035 m, and the maximum observed error was 0.0816 m. The error statistics indicate that the achieved accuracy is consistent with expectations for photodiode-based RSS fingerprinting in indoor

environments. It should be noted that the neural network architecture was chosen as a compromise between localization accuracy and computational cost. It delivers performance in line with existing literature while remaining lightweight enough for embedded implementation [35, 120]. Model-based methods (e.g., least squares trilateration [109] or spring relaxation [5]) can be considered in future work for runtime or memory efficiency.

These validation results should also be interpreted within the bounds of the testbed. The model was trained and validated in a single controlled environment with fixed luminaire geometry, constant receiver height, and assumed near-level operation. Performance in more cluttered spaces, under stronger reflections, or with greater receiver tilt remains to be established.

4.3.4 Ground Truth Tracking with HTC Vive

The HTC Vive tracking system was employed to accurately determine the robot's position and heading. Its effectiveness in research settings has been well established [121, 122]. The system consists of two fixed "Lighthouse" base stations and a mobile Tracker that connects via Bluetooth to a wireless dongle attached to a host PC.

The Tracker reports its position in a local coordinate system centered on the first base station, with axes aligned to that station's orientation. Consequently, a transformation is required to express the Tracker's measurements in the room's coordinate frame. To derive this transformation, the Tracker is placed at several known positions within the test environment (see Figure 4.2), and corresponding points are recorded in both coordinate frames. The resulting mapping from the Vive system to room coordinates is then applied to all subsequent positional data. A detailed description of this mapping procedure is provided in our previous work [121]. The CNC machine was used to accurately position the Vive Tracker (see Figure 4.5 with the tracker placed on the actuator) when capturing the calibration points.

After calibration, the Vive Tracker was transferred to a platform on the mobile robot at the same height (see Figure 4.3), with its orientation fixed by two mounting pins. The Tracker's orientation center is located at the middle of the device, where a mounting pin hole exists. In contrast, the center of the mobile robot is defined as the midpoint between its two drive wheels along the axle. Since the VLP photodiode receiver is positioned at the robot's center to support VLP-based estimation (see Figure 4.3), the Vive Tracker was offset by placing it 0.070 m in front of the robot's center. Consequently, the Vive-reported coordinates (x_{vive} and y_{vive}) must be translated to reflect the robot center's location. Using the Vive's yaw measurement—which matches the robot's yaw due to the Tracker's fixed orientation—the mapping is performed as follows:

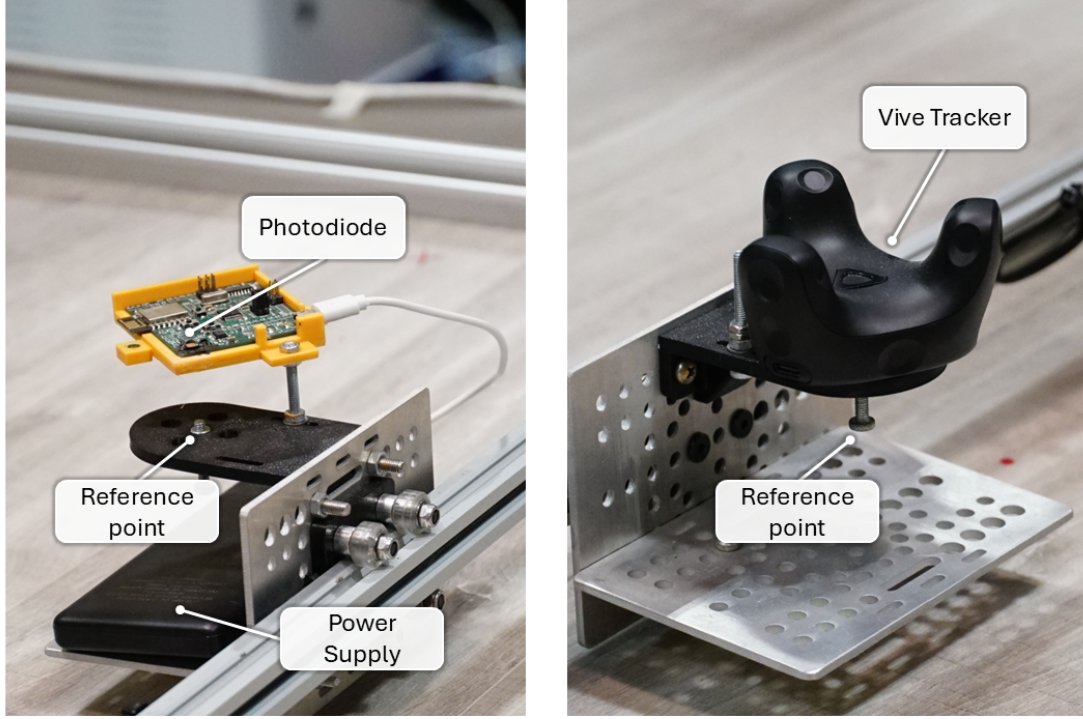


Fig. 4.5 CNC-mounted sensors used to co-locate the VLP receiver and Vive tracker on the same reference point for calibration.

$$x_{\text{robot}} = x_{\text{vive}} + 0.07 \cdot \sin(\text{vive}_{\text{yaw}}) \quad (4.2)$$

$$y_{\text{robot}} = y_{\text{vive}} + 0.07 \cdot \cos(\text{vive}_{\text{yaw}}) \quad (4.3)$$

4.4 Training dataset construction and Encoder Calibration

4.4.1 Overview

The goal of the data collection process is to construct a dataset comprising encoder-driven robot movements (turns and steps), corresponding VLP-based location estimates, and accurate ground-truth measurements of position and heading.

This dataset enables calibration of encoder odometry by deriving accurate mappings between encoder steps and actual robot motion. It also supports the development of a navigation model that fuses encoder odometry with VLP measurements.

For the training data, 10 separate runs were completed, with the number of movements (rotation followed by step) ranging from 354 to 1727 movements. In total, 13,186 turn and step movements were captured.

The dataset was generated by using real-time HTC Vive tracking to determine the motion commands sent to the robot, which were then executed via encoder odometry using the initial—albeit imperfect—calibration between encoder counts and actual displacement. This approach produces trajectories that capture the robot’s natural motion errors under encoder-based control while keeping it within the physical constraints of the environment.

Generation of target points

Training target points were sampled uniformly within the workspace; bounds were

$$x \in [0.05, 0.70], \quad y \in [0.05, 0.85].$$

To avoid clustering, each point was restricted to lie at least 0.10 m from its predecessor.

Navigation to points for training data

To navigate to a target point, the robot compares its current heading—provided by the Vive tracking system—to the target bearing, which is computed from the robot’s current position (also recorded by the Vive) and the coordinates of the target point. The required heading correction is then calculated. If this correction exceeds 20° , a command is issued to rotate the robot by the necessary amount. The rationale behind the 20° threshold is explained later in Section 4.4.2.

The robot executes this rotation using an encoder odometry control scheme, relying on the initial calibration of 5.28 encoder steps per degree. This process is subject to errors arising from encoder imprecision and an initial mapping offset (which is later corrected after dataset collection).

Following the heading adjustment, the robot executes a forward motion targeting a distance of 0.10 m using encoder odometry control calibrated at 4.45×10^3 encoder steps per meter.

When the robot reaches within 0.07 m of the target point, as determined by the Vive system, the target is considered reached, and the navigation sequence advances to the next point.

The goal at this stage is not to precisely follow a predefined trajectory, but rather to generate a dataset of encoder-driven movements spanning a wide range of headings and

spatial positions, while accurately tracking ground truth position using the Vive system. Figure 4.6 illustrates the navigation method for a subset of a navigated path.

At each step, a VLP reading is also collected, consisting of three consecutive measurements. These are then fed into the VLP positioning model to derive a VLP position estimate.

4.4.2 Encoder Calibration

The collected robot movements allow precise calibration of the encoder odometry model. This was achieved by comparing the number of encoder steps issued—based on the initial rough calibration—with the actual displacements recorded by the HTC Vive tracker. Specifically, the issued encoder commands were derived either from heading adjustments computed via Vive measurements or from the 0.10 m target step size.

Step Calibration

Each forward step was intended to cover a target distance of 0.10 m. Based on the initial calibration of 4.45×10^3 encoder counts per meter, this translated to:

$$\text{Issued counts} = 0.10 \text{ m} \times 4.45 \times 10^3 \frac{\text{counts}}{\text{m}} = 445 \text{ counts.} \quad (4.4)$$

However, analysis of the ground truth data showed that the actual mean distance travelled per step was 0.1118 m. This indicates that the initial mapping slightly overestimated the step size.

With a mean of 0.1118 m and a standard deviation of only 0.00015 m, the encoder odometry control was found to be highly precise, and the observed discrepancy in distance is attributed primarily to the initial calibration error rather than execution noise.

A corrected calibration factor was then computed as:

$$\text{Updated calibration} = \frac{445 \text{ counts}}{0.1118 \text{ m}} \approx 3.98 \times 10^3 \frac{\text{counts}}{\text{m}}. \quad (4.5)$$

Turn calibration

Initially, heading changes were assumed to follow a linear relationship with encoder counts, centered at zero. Analysis of the collected data instead showed a directional asymmetry between left and right turns, likely due to drivetrain friction. To address this, separate linear models were fit for each turning direction. With a minimum turning threshold of 20° , the

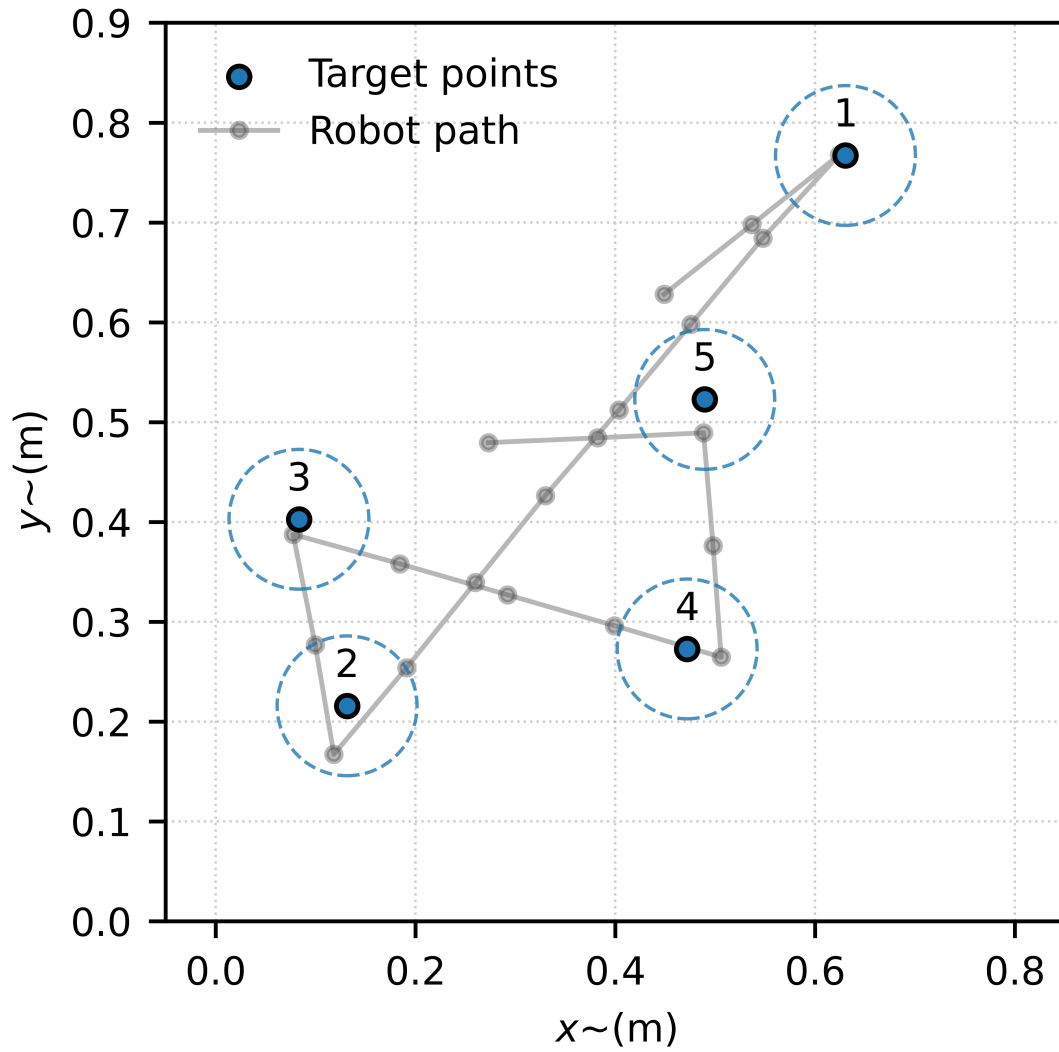


Fig. 4.6 Navigation logic for data collection: the robot executes heading corrections and fixed-length steps toward each target; dashed circles show the distance threshold for declaring a target reached.

fitted lines no longer needed to pass through the origin, allowing an offset term and different slopes for left- and right-hand turns.

As noted earlier, each movement step involves either a turn exceeding 20° or motion along the current heading. Without such a threshold, the controller would issue frequent small corrections dominated by heading noise, producing many short corrective turns with limited navigational benefit. Empirically, the relative error (turn error divided by commanded angle) increased sharply below 20° , likely due to drivetrain slack, static friction, and motor spin-up variability, even though absolute errors were smaller. This behavior motivated the minimum turn threshold of 20° . Future work could quantify how varying this threshold affects navigation accuracy and stability.

4.4.3 Dataset structure

The paths recorded during experimental robot runs were used to construct a dataset suitable for training navigation models (Table 4.1). Each entry in the dataset represents a complete movement cycle, consisting of a heading correction followed by a forward step, and concluded with both VLP and HTC Vive measurements.

The **target values** (ground truth) are the robot's final position and heading—specifically, the x and y coordinates and the heading angle α —as measured by the HTC Vive tracker at the end of each movement.

The **input features** comprise the raw control commands and onboard sensor estimates available to the robot during execution:

- **Encoder input:** the commanded heading change $\Delta\theta_e$ and fixed step size s_e , derived using the corrected mapping from encoder counts to physical displacement.
- **VLP input:** the VLP-estimated position (x_{vlp}, y_{vlp}) recorded at the end of the movement.

This structure captures both the control and sensor data associated with each movement, paired with the corresponding ground-truth position and heading, enabling supervised learning of motion prediction models. Figure 4.7 summarizes the data splits and model usage.

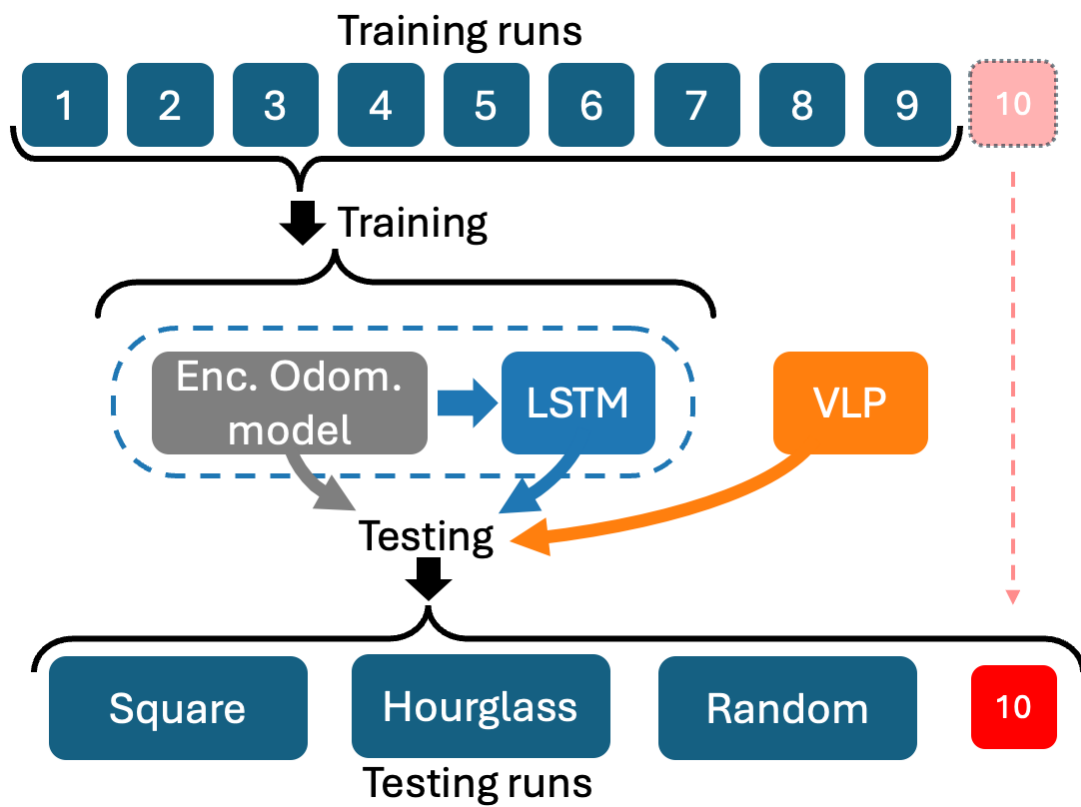


Fig. 4.7 Overview of how the models are trained and evaluated, including the offline split and the live-path evaluation protocol.

Table 4.1 Summary of input features and ground-truth targets used for supervised navigation model training, including the commanded heading change and step size s_e (robot step length) alongside VLP position measurements.

Pt #	x_{gt}	y_{gt}	α_{gt}	$\Delta\theta_e$	s_e	x_{vlp}	y_{vlp}
1	x_1	y_1	α_1	$\Delta\theta_{e,1}$	$s_{e,1}$	$x_{vlp,1}$	$y_{vlp,1}$
2	x_2	y_2	α_2	$\Delta\theta_{e,2}$	$s_{e,2}$	$x_{vlp,2}$	$y_{vlp,2}$
$\vdots$		$\vdots$			$\vdots$		$\vdots$
n	x_n	y_n	α_n	$\Delta\theta_{e,n}$	$s_{e,n}$	$x_{vlp,n}$	$y_{vlp,n}$

4.5 Single Mode Navigation Models

These models estimate the robot's position and heading using only one sensing modality. Two methods are considered: encoder odometry (Enc. Odom.) and Visible Light Positioning (VLP).

4.5.1 Encoder Odometry

Encoder odometry integrates incremental motion updates from wheel encoders, starting from a known initial pose.

The robot state vector at time step n is defined as:

$$\mathbf{x}_{enc,n} = \begin{bmatrix} x_{enc,n} & y_{enc,n} & \theta_{enc,n} \end{bmatrix}^\top, \quad (4.6)$$

where $x_{enc,n}$ and $y_{enc,n}$ represent the estimated position in the plane, and $\theta_{enc,n}$ is the estimated heading angle.

The initial state is set from a known starting pose:

$$\mathbf{x}_{enc,0} = \begin{bmatrix} x_0 & y_0 & \theta_0 \end{bmatrix}^\top.$$

At each time step $k = 1, 2, \dots, n$, incremental motion updates are computed from the wheel encoder measurements. The coordinate frame is defined with x increasing left-to-right, y increasing forward, and $\theta = 0$ aligned with the positive y -axis:

$$\Delta x_k = d_k \sin \theta_{enc,k}, \quad (4.7)$$

$$\Delta y_k = d_k \cos \theta_{enc,k}, \quad (4.8)$$

$$\theta_{enc,k} = \theta_{enc,k-1} + \Delta\theta_k, \quad (4.9)$$

where d_k is the distance traveled and $\Delta\theta_k$ is the heading change during the step.

The position at time step n is obtained by integrating these displacements:

$$x_{\text{enc},n} = x_0 + \sum_{k=1}^n d_k \sin \theta_{\text{enc},k}, \quad (4.10)$$

$$y_{\text{enc},n} = y_0 + \sum_{k=1}^n d_k \cos \theta_{\text{enc},k}. \quad (4.11)$$

$$\theta_{\text{enc},n} = \theta_0 + \sum_{k=1}^n \Delta \theta_k, \quad (4.12)$$

The initial pose (x_0, y_0, θ_0) is set at the start of each run using the Vive measurement; in a deployed system, it could be re-established by docking or aligning to a known reference station.

4.5.2 VLP

Visible Light Positioning (VLP) provides absolute position estimates, with heading inferred from the displacement between consecutive measurements.

The robot state vector at time step n is defined as:

$$\mathbf{x}_{\text{vlp},n} = \begin{bmatrix} x_{\text{vlp},n} & y_{\text{vlp},n} & \theta_{\text{vlp},n} \end{bmatrix}^T, \quad (4.13)$$

where $x_{\text{vlp},n}$ and $y_{\text{vlp},n}$ are the VLP location estimates, and $\theta_{\text{vlp},n}$ is the heading angle, computed from the displacement between consecutive estimates:

$$\theta_{\text{vlp},n} = \text{atan2}(x_{\text{vlp},n} - x_{\text{vlp},n-1}, y_{\text{vlp},n} - y_{\text{vlp},n-1}). \quad (4.14)$$

4.6 LSTM Fusion Model

The Long Short-Term Memory (LSTM) network fuses encoder odometry and VLP measurements by learning the relationship between motion and observation sequences directly from data.

Feature Engineering

The raw dataset (Table 4.1) was expanded to generate input features for the LSTM. Since movement was restricted to a constant step size, step length was omitted. Additional features from encoder odometry included the estimated x , y , and heading (Section 4.4.2). From the VLP system, the x and y positions, the VLP-derived heading, and the change in heading

Table 4.2 Expanded input features for the LSTM, including encoder odometry estimates, VLP-derived heading, and heading-change terms from both VLP and encoder odometry; step size is omitted because it is constant.

Point #	Targets			Encoder Input				VLP Input			
	$\mathbf{x}$	$\mathbf{y}$	$\boldsymbol{\theta}$	$\mathbf{x}_e$	$\mathbf{y}_e$	$\boldsymbol{\theta}_e$	$\Delta\boldsymbol{\theta}_e$	$\mathbf{x}_{vlp}$	$\mathbf{y}_{vlp}$	$\boldsymbol{\theta}_{vlp}$	$\Delta\boldsymbol{\theta}_{vlp}$
1	x_n	y_n	θ_n	$x_{e,n}$	$y_{e,n}$	$\theta_{e,n}$	$\Delta\theta_{e,n}$	$x_{vlp,n}$	$y_{vlp,n}$	$\theta_{vlp,n}$	$\Delta\theta_{vlp,n}$
2	x_{n-1}	y_{n-1}	θ_{n-1}	$x_{e,n-1}$	$y_{e,n-1}$	$\theta_{e,n-1}$	$\Delta\theta_{e,n-1}$	$x_{vlp,n-1}$	$y_{vlp,n-1}$	$\theta_{vlp,n-1}$	$\Delta\theta_{vlp,n-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	x_1	y_1	θ_1	$x_{e,1}$	$y_{e,1}$	$\theta_{e,1}$	$\Delta\theta_{e,1}$	$x_{vlp,1}$	$y_{vlp,1}$	$\theta_{vlp,1}$	$\Delta\theta_{vlp,1}$

between consecutive VLP estimates were incorporated. Table 4.2 summarizes the expanded feature set.

Data Preparation and Sequential Windowing

Each LSTM input sample is:

$$\mathbf{X} \in \mathbb{R}^{T \times d}, \quad (4.15)$$

where T is the sequence length and $d = 8$ is the number of synchronized features. The target is the state vector:

$$\mathbf{y} = [x \ y \ \theta]^\top. \quad (4.16)$$

with heading encoded as:

$$[\sin(\theta) \ \cos(\theta)]^\top. \quad (4.17)$$

to ensure continuity across the wraparound at $\pm\pi$.

Sequences are generated using a sliding-window approach: from a time series $\{\mathbf{x}_t\}_{t=1}^N$, overlapping windows of $T = 25$ samples are extracted with stride 1:

$$\mathbf{X}^{(i)} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+24}]. \quad (4.18)$$

During hyperparameter tuning, only the most recent n steps of each window were retained, with n ranging from 5 to 25; the final model used $n = 20$.

The dataset was split into training (Datasets 1–8), validation (Dataset 9), and test (Dataset 10), resulting in:

$$\begin{aligned} X_{\text{train}} &\in \mathbb{R}^{(11758, 25, 8)}, \\ X_{\text{valid}} &\in \mathbb{R}^{(633, 25, 8)}, \\ X_{\text{test}} &\in \mathbb{R}^{(305, 25, 8)}. \end{aligned}$$

Missing values are zero-filled, and all inputs are normalized via a trainable Normalization layer.

Output and loss function

The LSTM is a sequence-to-vector model with two output heads: position and heading. The position head predicts (x, y) with a linear activation, while the heading is encoded as a 2D unit vector $[\sin(\theta), \cos(\theta)]$ to avoid discontinuities at $\pm\pi$ and is produced by a tanh activation. Accordingly,

$$\mathbf{y}_{\text{pred}} = \begin{bmatrix} x_{\text{pred}} & y_{\text{pred}} & \sin(\theta_{\text{pred}}) & \cos(\theta_{\text{pred}}) \end{bmatrix}^{\top},$$

which is compared to the ground truth target

$$\mathbf{y}_{\text{true}} = \begin{bmatrix} x_{\text{true}} & y_{\text{true}} & \sin(\theta_{\text{true}}) & \cos(\theta_{\text{true}}) \end{bmatrix}^{\top}.$$

The total training loss is the sum of mean-squared errors from the two heads (equal weighting):

$$\begin{aligned} \mathcal{L} = & \text{MSE}([x, y]_{\text{pred}}, [x, y]_{\text{true}}) \\ & + \text{MSE}([\sin \theta, \cos \theta]_{\text{pred}}, [\sin \theta, \cos \theta]_{\text{true}}). \end{aligned}$$

During evaluation, the heading is recovered as

$$\hat{\theta} = \arctan 2(\sin \theta_{\text{pred}}, \cos \theta_{\text{pred}}) \quad (4.19)$$

. The average positional error is computed as:

$$\epsilon_{\text{pos}}^{\text{LSTM}} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_{\text{pred}}^{(i)} - x_{\text{true}}^{(i)})^2 + (y_{\text{pred}}^{(i)} - y_{\text{true}}^{(i)})^2}, \quad (4.20)$$

Hyperparameter Tuning

Model selection was performed in four stages using Keras Tuner's RandomSearch to minimize validation loss:

Neurons and Layers: Search over $N \in [1, 4]$ LSTM layers and $U_i \in \{8, 16, 32, 64, 128\}$ units per layer. Best: $N = 2$, $U_0 = 32$, $U_1 = 8$.

Regularization: Dropout and recurrent dropout $\in \{0.0, 0.01, 0.025, 0.05\}$; optional batch and layer normalization. Best:

Layer 1: dropout = 0.0, recurrent = 0.025,
layer norm = True,
Layer 2: dropout = 0.01, recurrent = 0.05,
no normalization.

Optimization: Learning rate $\eta \in [10^{-4}, 10^{-2}]$, schedulers (none, cosine, exponential), optimizers (Adam, RMSprop, SGD, Nadam). Best: $\eta \approx 3.37 \times 10^{-3}$, no scheduler, Nadam optimizer.

Sequence Length: Effective sequence length $L \in \{5, 10, 15, 20, 25\}$. Best: $L = 20$.

Final Model Summary

The final model is a two-layer LSTM with:

- Input: (20, 8) — last 20 steps of each 25-sample window,
- LSTM 1: 32 units, recurrent dropout 0.025, layer normalization,
- LSTM 2: 8 units, dropout 0.01, recurrent dropout 0.05, no normalization.

Outputs:

- 2D position — dense, linear activation,
- Heading ($\sin \theta, \cos \theta$) — dense, tanh activation.

Trained with Nadam optimizer ($\eta \approx 3.37 \times 10^{-3}$), using equal-weight mean squared error loss for both outputs.

4.7 Navigation Performance

To evaluate the navigation models, we employed a two-stage framework comprising an offline evaluation on held-out data and a live evaluation in which models controlled the robot in real time. LSTM hyperparameters were selected using runs 1–8 for training and run 9 for validation; the final model was retrained on runs 1–9 and tested on run 10. VLP required no

tuning, and encoder odometry (Enc. Odom.) used datasets 1–9 for calibration. For dataset 10, 300 samples were evaluated beginning at sample 20 to satisfy the LSTM’s input window.

Live experiments used 300 movement steps per trial. Three trajectories were tested: a square path, an hourglass path, and a random path. The square path consisted of four repeated targets forming a loop. The hourglass path reordered these same targets to generate alternating clockwise and counterclockwise turns. The random-path targets were generated using the same sampling procedure as the training data, with a minimum spacing of 0.112 m.

4.7.1 Temporal Error Analysis

Figure 4.8 presents heading and position error over the full 300-step runs for all scenarios.

Offline Evaluation

VLP-only estimates show short-term noise but maintain long-term consistency. Encoder odometry performs well initially but drifts rapidly, with heading error exceeding 15° and position error exceeding 0.15 m. LSTM fusion reduces drift and maintains the tightest error bounds.

Square Path

The square-path results highlight rapid drift in encoder odometry. Heading error exceeds 80° and position error reaches 0.3 m. VLP shows irregular fluctuations but no cumulative drift. The LSTM remains stable across the run, with a slight increase in error near the end.

Hourglass Path

Encoder drift grows until approximately 200 steps and then stabilizes. Drift remains lower than in the square path but is still substantial. LSTM errors remain consistently below the single-mode baselines.

Random Path

The random path produces moderate and variable errors. Encoder heading error remains low for the first 100 steps and then rises. Position drift remains limited due to shorter step lengths and frequent direction changes. The LSTM generally outperforms the single-mode baselines.

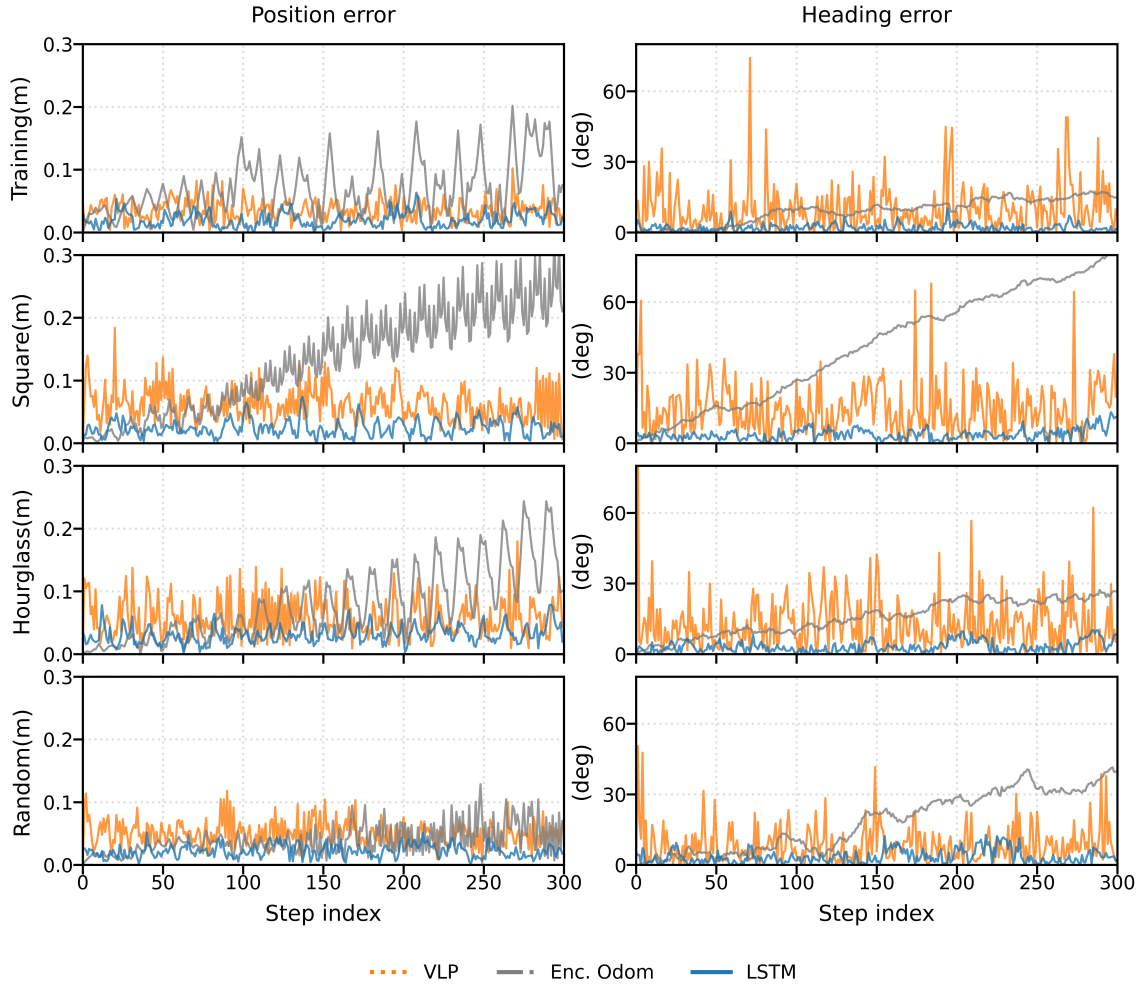


Fig. 4.8 Temporal error traces over 300 robot steps for LSTM fusion and single-mode baselines. Columns show position (left) and heading (right) errors, and rows correspond to the offline/test split, square, hourglass, and random paths.

4.7.2 Quantitative Error Analysis

Table 4.3 reports the median and P_{90} errors for the first and last 100 steps, capturing time-varying behavior within each 300-step evaluation. These results are consistent with the trends observed in the temporal analysis presented in Section 4.7.1. In the following discussion, median errors are used for clarity; however, the P_{90} errors exhibit similar behavior. This presentation was chosen deliberately: the table is intended primarily to quantify how typical error and practically relevant upper-tail error evolve between the early and late portions of each run, rather than to summarize variability about the mean. For navigation, upper-tail errors are often more operationally significant than standard-deviation-type spread, because they more directly affect target acquisition reliability and obstacle-clearance margin.

Figure 4.8 complements the table by showing the full temporal evolution and spread of the errors across each run.

Offline

Encoder odometry accuracy degrades sharply over time. VLP remains steady. LSTM fusion maintains low heading error across both segments and the most stable position accuracy.

Square Path

LSTM performance is stable, with heading error increasing only from 2.68° to 4.36° and position error increasing slightly. Encoder drift is severe, with heading error rising from 14.4° to 68.9° and position error increasing from 0.037 m to 0.212 m. VLP remains drift-free but noisier than LSTM.

Hourglass Path

The LSTM achieves the best heading accuracy, increasing only from 2.3° to 3.7° . Encoder drift is initially moderate but reaches 0.13 m in position. VLP position accuracy remains near 0.05 m, while its heading errors are larger than LSTM.

Random Path

The LSTM remains the best overall performer. VLP remains stable. Encoder drift follows expected patterns, with heading error rising from 4.62° to 32.03° .

4.7.3 Qualitative Path Analysis

Qualitative path plots were produced only for the square and hourglass trajectories, as offline and random paths lack geometric structure (Figures 4.9 and 4.10).

Square Path

Encoder odometry exhibits large cumulative rotation by the end of the run. VLP shows oscillatory corrections but limited long-term deviation. The LSTM follows the intended path closely and maintains consistent corner alignment, aided by the 20° turn threshold.

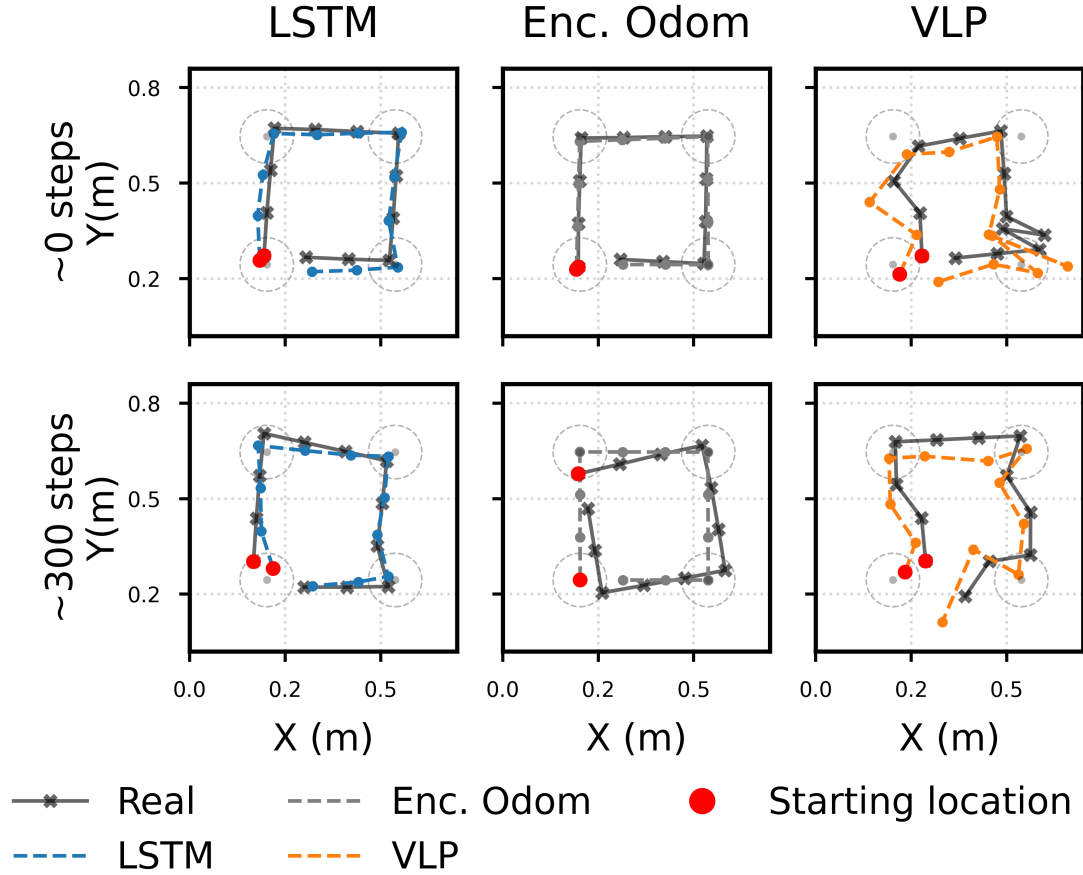


Fig. 4.9 Square-path trajectories showing drift over time: the top row is early in the run and the bottom row is near the end. The LSTM remains stable, encoder odometry rotates nearly 90° , and VLP is noisy but stable.

Hourglass Path

Encoder odometry begins accurately but accumulates drift as the run progresses. The center of the trajectory shifts due to non-integer step alignment and the 0.07 m target-acquisition radius. VLP displays over- and under-correction behavior similar to the square path. The LSTM tracks both the intended and actual robot trajectories closely and resolves small deviations over time.

4.7.4 Model-Wide Comparison

Figure 4.11 compares all models across all scenarios using the final 100 steps. Across all evaluations, the LSTM demonstrates the strongest generalization and robustness, with its

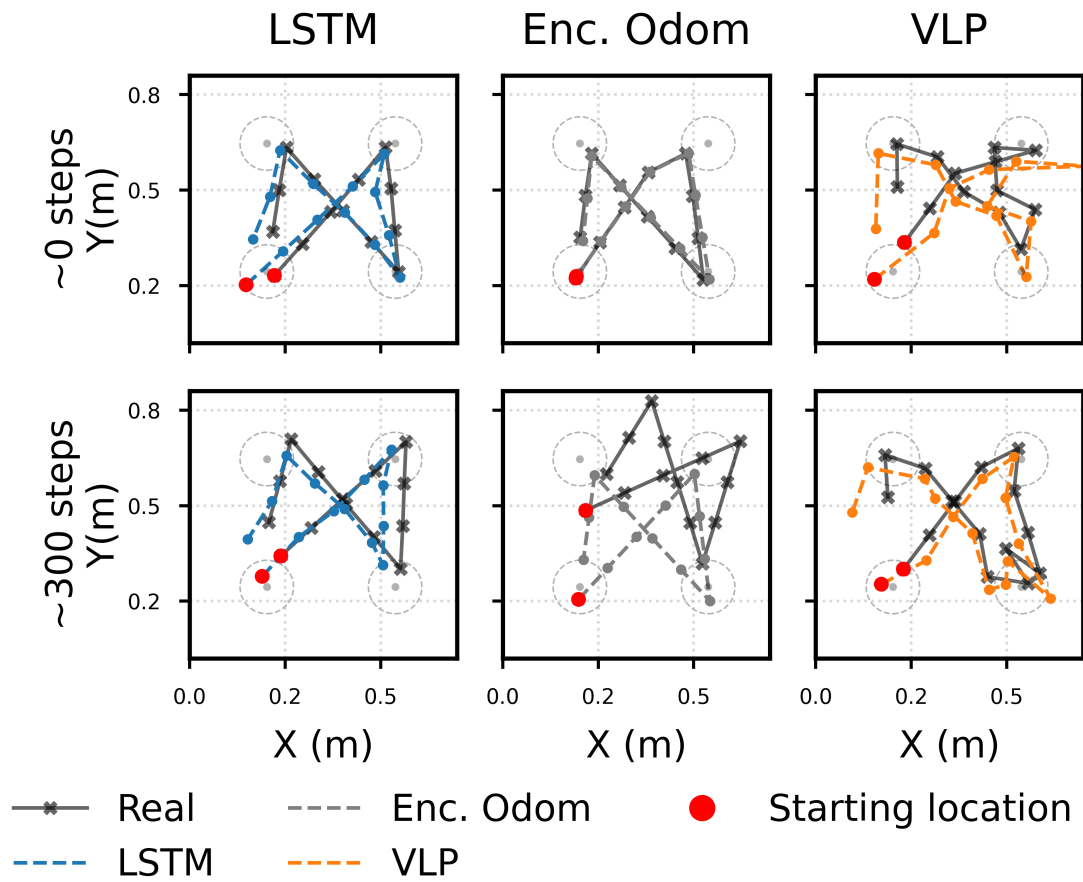


Fig. 4.10 Hourglass-path trajectories over time (top row early, bottom row late). The LSTM is the most consistent, encoder odometry degrades further, and VLP is noisy but stable.

lowest heading and position errors on the training split and stable performance across test trajectories.

VLP provides steady but spatially dependent accuracy, and the results confirm that PD-only VLP can recover full 2D pose under ground-truth evaluation. The spatial dependence is most likely driven by local workspace biases: structured trajectories revisit the same regions repeatedly, amplifying location-specific errors relative to the more diffuse random path.

Encoder odometry is least reliable due to cumulative drift, with the largest errors on the square path and the smallest on the random path; heading and position errors do not correlate, highlighting nonlinear drift effects. These findings demonstrate both the viability of PD-only VLP and the clear advantage of data-driven fusion for real-time indoor navigation.

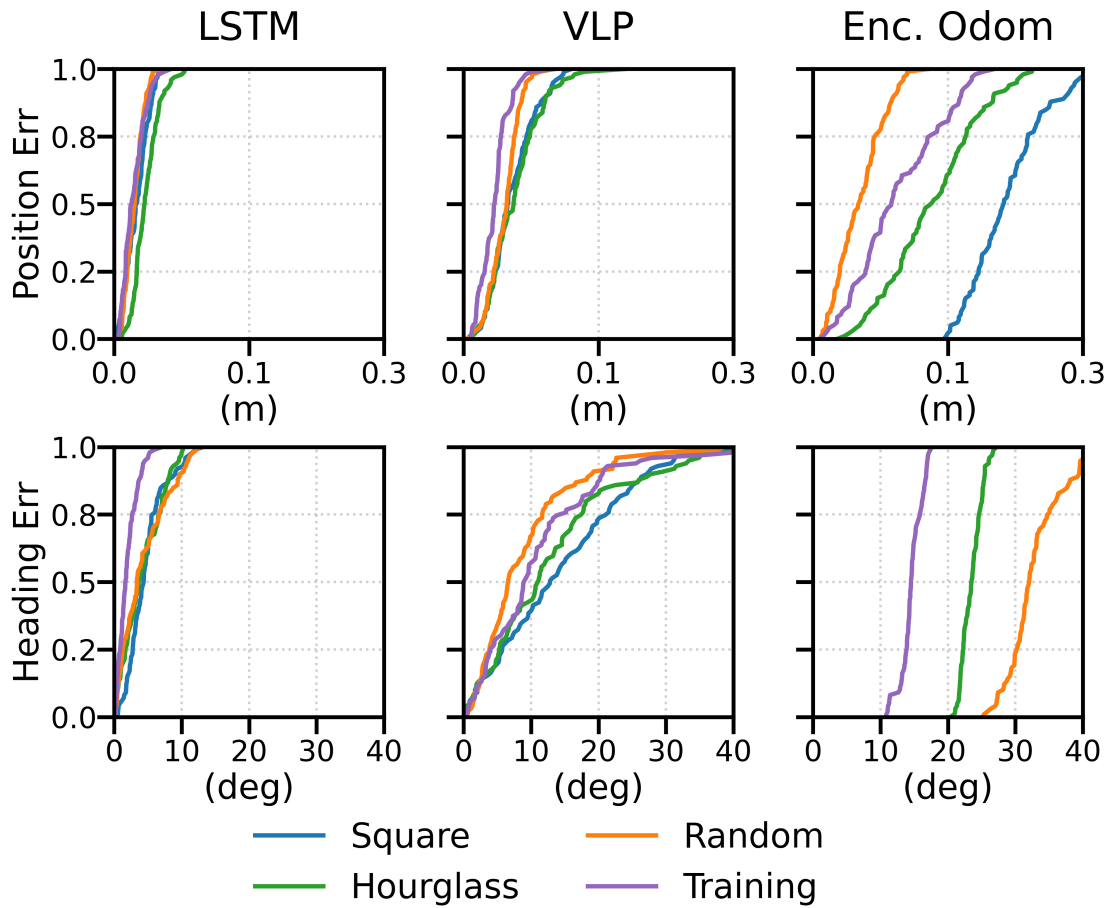


Fig. 4.11 CDFs of position (top row) and heading (bottom row) errors for the last 100 steps (late-run performance). Columns correspond to navigation models (LSTM, VLP, Enc. Odom.), and within each panel the curves represent the training, square, hourglass, and random trajectories. The LSTM is best overall but shows its strongest heading performance on the training split; VLP is relatively consistent across paths; and encoder odometry heading errors on the square path fall outside the plotted range.

Table 4.3 Median and 90th percentile (P_{90}) heading and position errors for the first (steps 1–100) and last (steps 200–300) segments across all scenarios. The table is intended to highlight typical error and practically relevant upper-tail error as drift develops within each run, rather than to report mean-centered variability measures. Boldface indicates the lowest (best) error value in each scenario.

Training (testing split)					Square path				
Model	First 100		Last 100		Model	First 100		Last 100	
	Median	P_{90}	Median	P_{90}		Median	P_{90}	Median	P_{90}
<i>Heading error (degrees)</i>					<i>Heading error (degrees)</i>				
LSTM	1.4	3.3	1.7	4.1	LSTM	2.7	4.8	4.4	9.0
VLP	5.8	20.6	8.9	20.6	VLP	13.0	26.5	12.8	27.2
Enc. Odom.	1.6	9.6	14.5	16.8	Enc. Odom.	14.4	21.6	68.9	77.6
<i>Position error (meters)</i>					<i>Position error (meters)</i>				
LSTM	0.018	0.034	0.019	0.040	LSTM	0.021	0.036	0.025	0.043
VLP	0.035	0.058	0.034	0.055	VLP	0.063	0.101	0.049	0.092
Enc. Odom.	0.042	0.076	0.087	0.165	Enc. Odom.	0.037	0.068	0.212	0.281
Hourglass path					Random path				
Model	First 100		Last 100		Model	First 100		Last 100	
	Median	P_{90}	Median	P_{90}		Median	P_{90}	Median	P_{90}
<i>Heading error (degrees)</i>					<i>Heading error (degrees)</i>				
LSTM	2.3	4.9	3.7	8.3	LSTM	2.2	5.8	3.4	9.7
VLP	9.9	24.0	11.0	27.9	VLP	6.4	18.3	6.6	19.0
Enc. Odom.	7.9	10.3	23.5	25.5	Enc. Odom.	4.6	10.3	32.0	39.4
<i>Position error (meters)</i>					<i>Position error (meters)</i>				
LSTM	0.024	0.047	0.034	0.054	LSTM	0.019	0.029	0.023	0.036
VLP	0.049	0.111	0.056	0.092	VLP	0.051	0.080	0.048	0.066
Enc. Odom.	0.021	0.053	0.131	0.201	Enc. Odom.	0.029	0.041	0.051	0.090

4.8 Captured Robot Experiments

While nominal path-following performance captures the stability of autonomous motion, real-world operation can involve unforeseen disturbances such as physical displacement or tampering. To assess robustness under such conditions, a series of controlled capture experiments were conducted in which the robot's pose was deliberately altered during live navigation without updating its internal state. These interventions simulated tracking loss or external interference. Three types of interventions were applied:

1. **Translation-only:** a lateral displacement of approximately 0.15 m.
2. **Rotation-only:** an in-place rotation of about 90° without translation.
3. **Combined:** both 0.15 m translation and 90° rotation.

For each perturbation, one experimental run was completed using the LSTM-based framework, resulting in three trials in total. Prior to intervention, the robot autonomously followed a square trajectory to ensure steady-state operation, and the LSTM accumulated at least 20 navigation steps of history. A smaller navigation square than in standard testing (see Figure 4.2) was adopted to provide sufficient recovery space, and the corner targets were adjusted to match the expected post-perturbation headings. These experiments are therefore best interpreted as illustrative stress tests rather than statistically replicated robustness benchmarks. Repeated trials would be expected to show some variation in peak error, recovery time, and residual offset due to the exact perturbation magnitude, timing, and local VLP accuracy at the intervention point.

Translation-Only Perturbation

At the top-left corner of the trajectory, the robot was displaced diagonally by ≈ 0.15 m while maintaining its heading. The fusion model exhibited an immediate increase in position error, as can be seen in Figure 4.12. The fusion model produced a rapid correction followed by a minor rebound before stabilizing at a residual error of ≈ 0.1 m. Heading errors were largely unaffected. The corresponding trajectory plot, shown in Figure 4.13, reveals a visible deviation from the nominal square as the fusion model adapts to the new VLP observations. Notably, the fusion model decoupled heading and position estimates, with the inferred heading often differing from the geometric vector between consecutive positions.

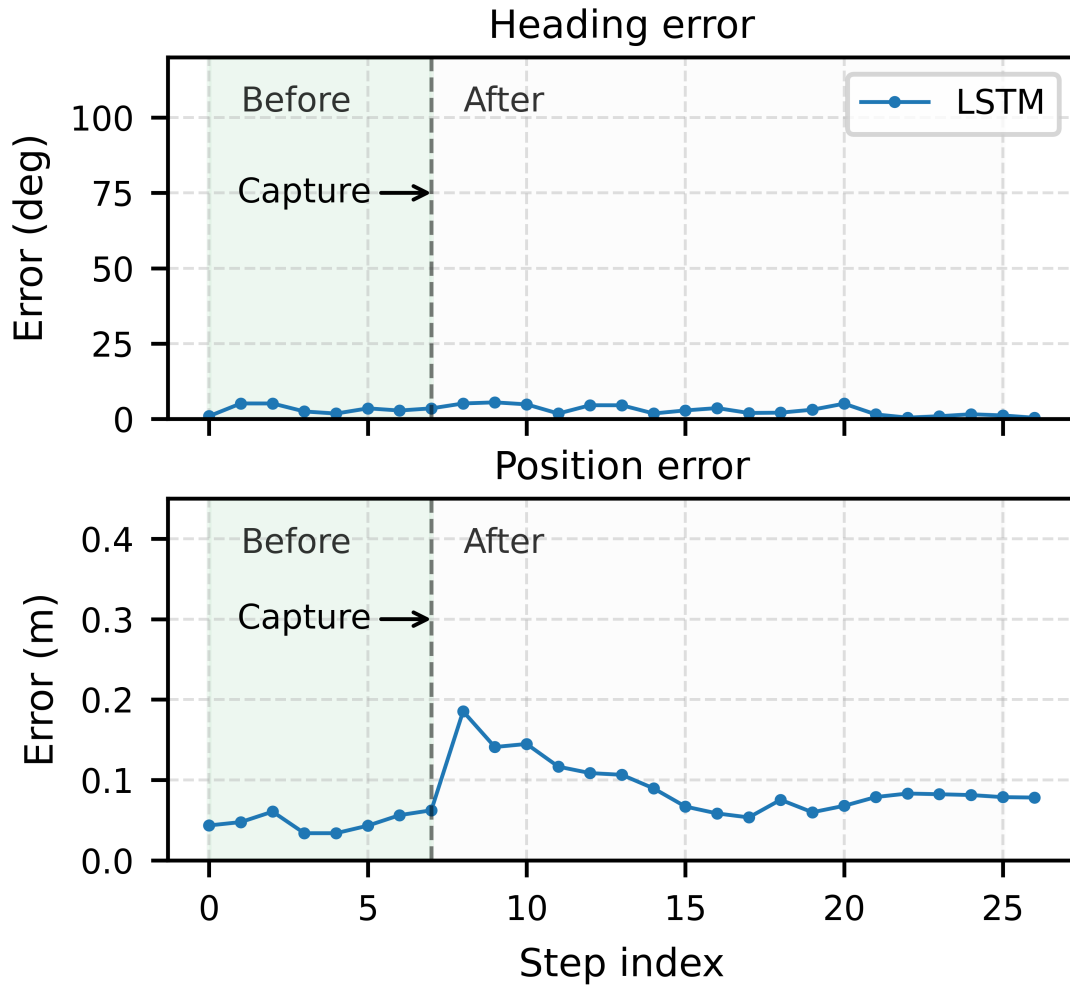


Fig. 4.12 Translation-only capture error traces. The vertical dashed line marks the perturbation; position error spikes while heading remains low and stable. The fusion model corrects rapidly.

Rotation-Only Perturbation

When manually rotated by approximately 90° , the fusion model exhibited a large heading deviation ($\approx 80^\circ$), as illustrated in Figure 4.14. The fusion model began correcting immediately. Position errors rose sharply after rotation and then reduced to ≈ 0.08 m, subsequently falling below 0.05 m after recovery. The estimated trajectories and heading evolution are shown in Figure 4.15, where the fusion model's path converges toward the true trajectory.

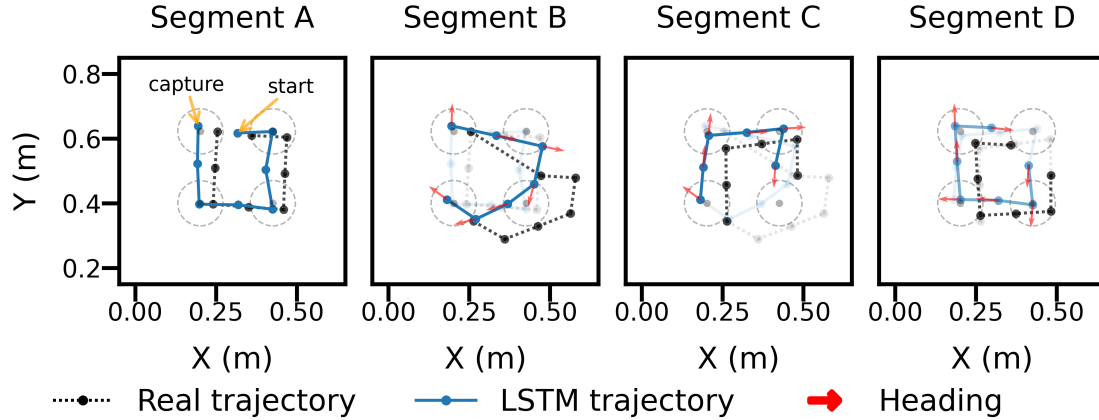


Fig. 4.13 Translation-only capture trajectories. The LSTM navigation diverges from the nominal square, indicating a rapid response to VLP updates. Target points and threshold circles are shown for reference.

Combined Translation and Rotation Perturbation

Under simultaneous 0.15 m translation and 90° rotation, the fusion model experienced significant initial errors, as shown in Figure 4.16. Immediately after the perturbation, the heading error exceeded $\approx 60^\circ$. The fusion model gradually recovered via VLP corrections. Initial position errors reached ≈ 0.25 m, followed by monotonic improvement. The reconstructed trajectories shown in Figure 4.17 further illustrate that the fusion model immediately recognized the displacement and adjusted its internal step-size assumptions dynamically, though it under-compensated for the first post-perturbation step.

Across all capture experiments, the fusion model re-aligned to the true trajectory after each perturbation. These findings indicate that the recurrent fusion approach can adapt to nonlinear disturbances by integrating temporal context and nonlinear measurement dependencies, but the evidence remains limited to single representative trials in the present workspace.

Future work could incorporate synthetic capture events during LSTM training (e.g., by skipping or perturbing trajectory segments) to further enhance robustness. However, such augmentation may trade off nominal-condition accuracy, therefore needing a careful balance between robustness and baseline performance.

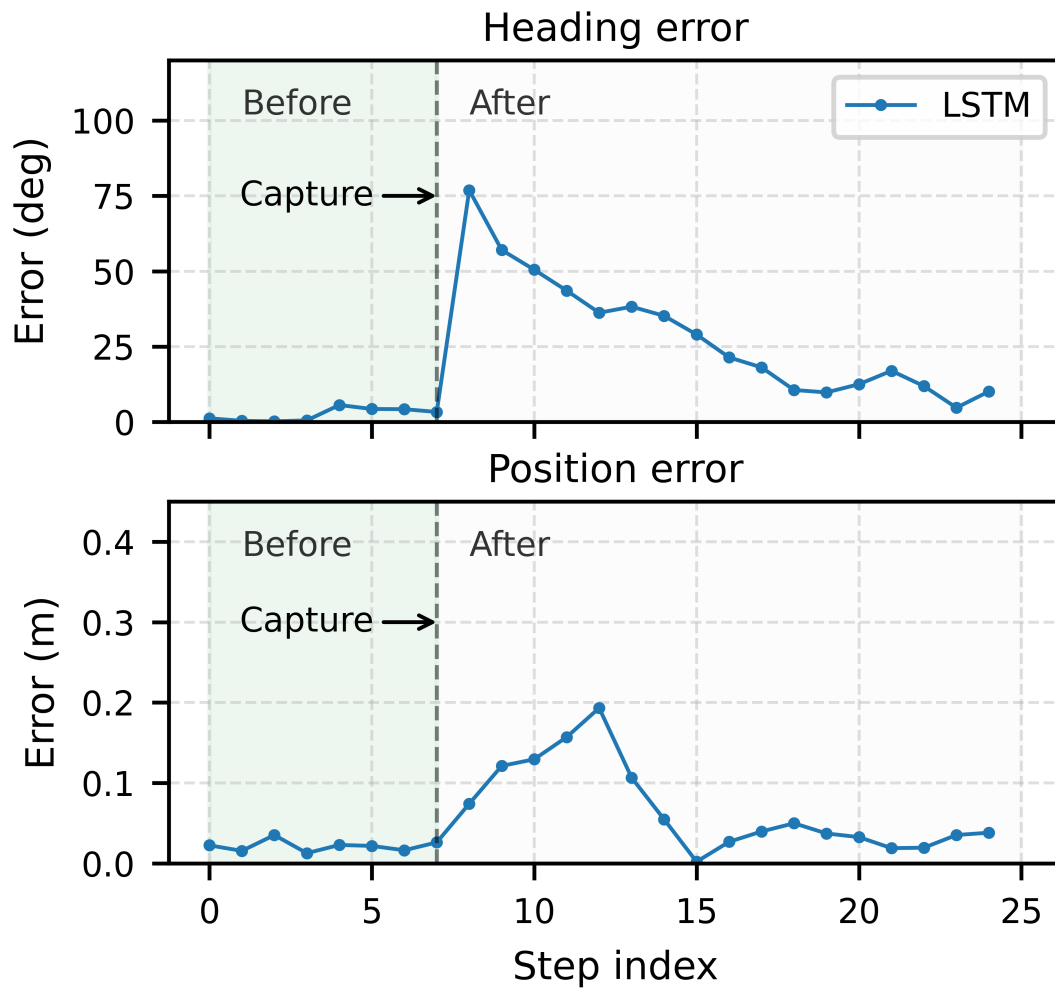


Fig. 4.14 Rotation-only capture error traces. Heading errors jump immediately; position errors rise as expected. The fusion model recovers after the perturbation.

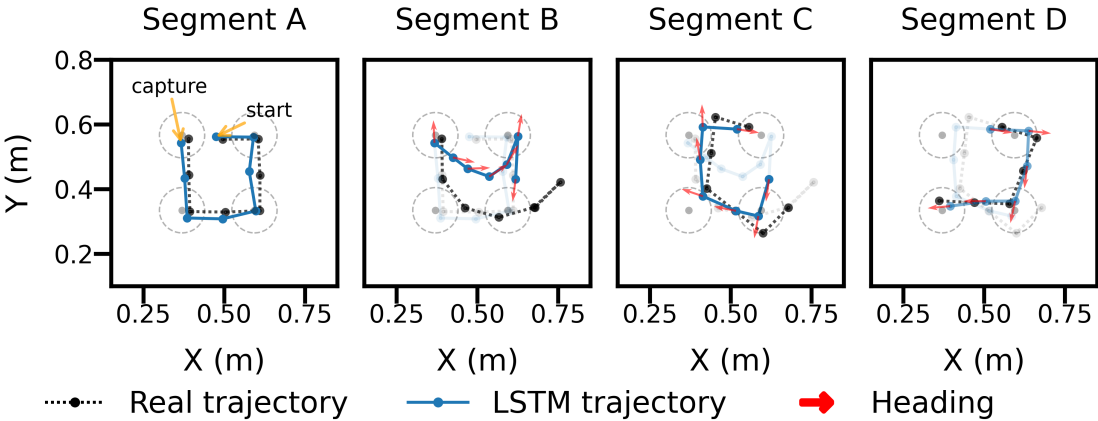


Fig. 4.15 Rotation-only capture trajectories. The fusion model responds quickly to VLP updates and realigns with the intended path.

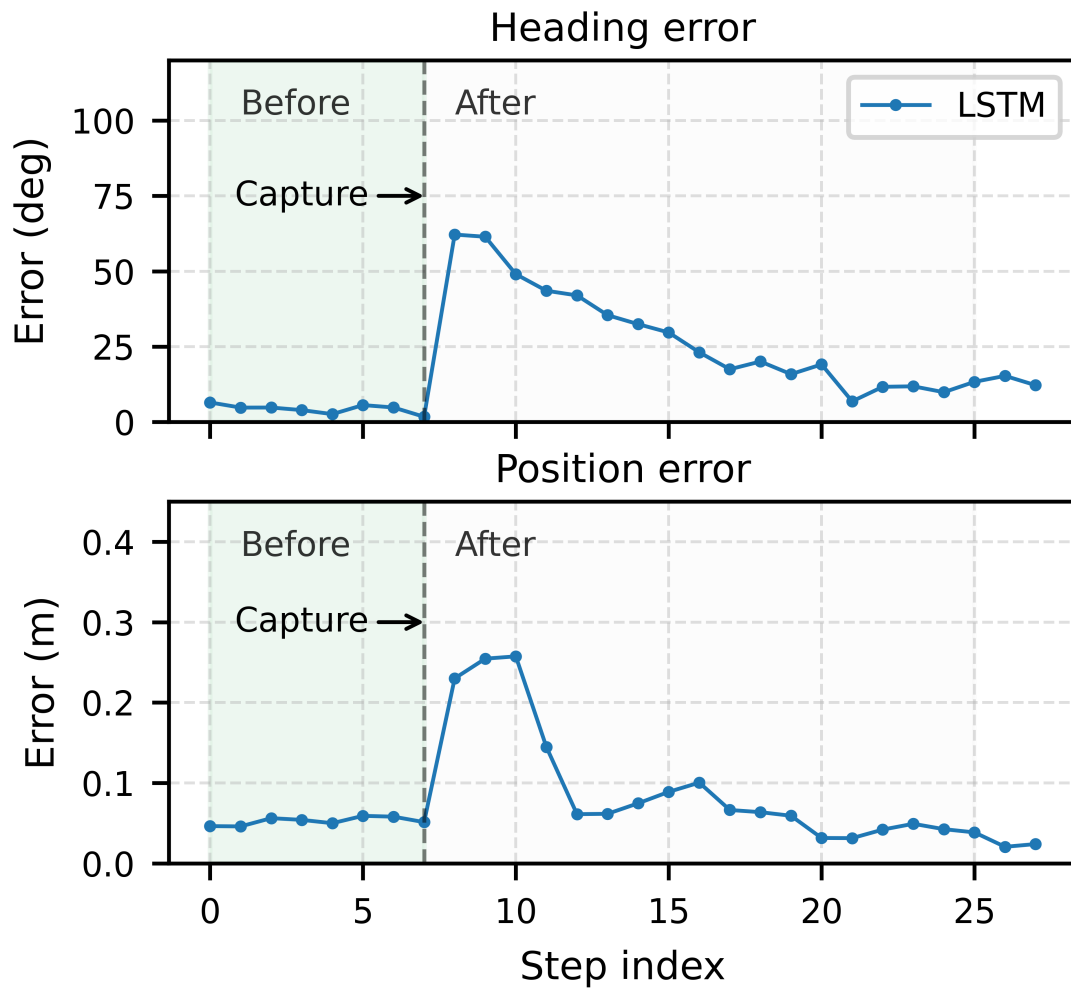


Fig. 4.16 Combined translation-and-rotation error traces. Heading and position errors jump simultaneously; the fusion model recovers over subsequent steps.

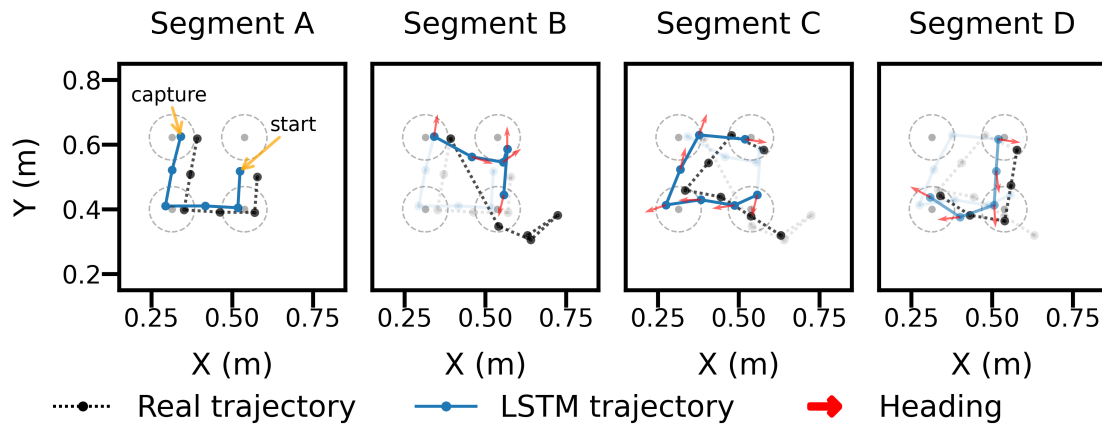


Fig. 4.17 Combined translation-and-rotation capture trajectories. The fusion model shows rapid correction to the VLP shift under the compounded perturbation.

4.9 Edge Implementation Considerations

For this work, the mobile robot streamed its data to a PC that ran the localization algorithms and provided navigation instructions over wireless links. In a real-world deployment, this would all run onboard the mobile robot. Therefore, the performance of the machine learning approach on resource-constrained embedded hardware is an important consideration. The real-world deployment feasibility of the proposed fusion method was analyzed using ST Edge AI Developer Cloud to quantify hardware utilization and inference performance across multiple STM32 embedded platforms. The complete inference pipeline, comprising the feedforward RSS-based VLP network, odometry model, and final LSTM fusion network, was implemented in MATLAB for validation. The stacked network was then exported to ONNX format and analyzed using ST Edge AI Developer Cloud. Table 4.4 summarizes the model inference time and deployment platform characteristics. Optimizing for inference time with float32 precision, the network requires 8,282 multiply-accumulate (MACC). These modest resource requirements enable deployment on a wide range of cost-effective microcontrollers.

Table 4.4 Performance and memory comparison across development boards.

Board	Time (ms)	Freq. (MHz)	Opt.	Flash (B)	RAM (B)	Act. (B)	Weights (B)
NUCLEO-F401RE	1.427	84	time	47502	3008	1056	33236
NUCLEO-G474RE	0.962	170	time	47502	3008	1056	33236
NUCLEO-H743ZI2	0.203	480	time	47422	3104	1152	33236

The inference latency of 1.427 ms on the NUCLEO-F401RE enables update rates exceeding 700 Hz, which significantly exceeds the requirements for typical indoor mobile robot navigation [123]. Standard robot localization frameworks such as ROS robot_localization operate at 30 Hz by default [124], while mobile robot navigation stack nodes such as move_base have a default controller_frequency of 20 Hz [125]. Even for smaller, more dynamic platforms, update rates of 50 Hz are considered sufficient [126]. These results demonstrate the viability of the proposed approach for deployment on resource-constrained embedded hardware for real-time robot navigation without requiring specialized accelerators.

4.10 Conclusion and Future Works

This work delivers an end-to-end demonstration of photodiode-based VLP for closed-loop robot navigation, combining a purpose-built dataset, LSTM fusion, and live capture-recovery tests under ground-truth validation.

4.10.1 Summary of Contributions

The main contributions of this work are as follows:

- **System development:** A low-cost differential-drive robot and a photodiode-based VLP system integrated with HTC Vive ground truth for repeatable navigation experiments.
- **Dataset construction:** A multi-run dataset coupling encoder commands, VLP measurements, and ground-truth pose for calibration and fusion model training.
- **Model implementation:** Encoder odometry, VLP-only pose recovery, and an LSTM fusion model with tailored feature engineering.
- **Evaluation framework:** Offline and live navigation benchmarks across structured and random paths with consistent error metrics.
- **Robustness experiments:** Controlled capture tests that quantify recovery from translation and rotation perturbations.
- **Embedded Model Validation:** The developed stacked neural network structure was benchmarked on embedded hardware to justify inference latency sufficient for real-world embedded deployment.

4.10.2 Key Findings

The experiments show that a PD-only VLP system can recover full 2D pose for real-time navigation and serves as a viable absolute reference in indoor settings. Fusing VLP with encoder odometry improves long-horizon accuracy relative to either modality alone, and the fusion model consistently performs best across paths and perturbations.

Across three live paths (last 100 steps), LSTM fusion reduced median errors to 0.023–0.034 m (position) and 3.4–4.4° (heading), versus VLP-only 0.048–0.056 m / 6.6–12.8°; encoder odometry drift reached 0.21 m / 69° on the square path. Capture tests show rapid recovery after translation and rotation events, indicating that recurrent fusion can reconcile inconsistent measurements under abrupt state changes.

At the same time, these results should be bounded to the conditions under which they were obtained: a single controlled indoor workspace, planar motion on a flat surface, fixed receiver height, and a rigidly mounted photodiode receiver without explicit tilt compensation. The findings therefore support feasibility and comparative benefit in near-level indoor operation, rather than broad claims of robustness across cluttered, multi-site, or industrial deployments.

4.10.3 Limitations and Future Directions

Several limitations bound the scope of these findings: experiments were conducted in a single environment with fixed LED geometry and constant receiver height; motion was planar and photodiode tilt was not modeled or instrumented directly; step size was fixed; robustness tests were limited to a small number of trials, including single representative perturbation trials; and the HTC Vive ground truth is not deployable in real-world settings (though alternative lab-grade systems such as optical motion capture could further improve reference accuracy). As a result, performance may degrade in larger or more cluttered spaces, under stronger reflections or occlusions, or on robots that experience greater pitch/roll variation.

These limitations should be addressed in future work by evaluating in larger and more diverse environments, incorporating orientation sensing, testing variable step sizes and higher-speed motion, and expanding capture-event trials across hardware. In addition, alternative estimators and sequence models (e.g., EKF/UKF, factor-graph methods, TCNs, Transformers, GRUs) and embedded quantization effects should be studied to better characterize accuracy–compute trade-offs.

Funding

This work received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability

The data and code supporting this study are available at <https://github.com/tyrel-glass/VLP-Odometry-Fusion.git>.

Author Contributions

Tyrel Glass: Conceptualization; methodology; data collection; software; validation; formal analysis; visualization; writing — original draft; writing — review and editing.

Fakhrul Alam: Primary supervision; methodology guidance; writing — review and editing.

Daniel Konings: Secondary supervision; guidance on research direction; embedded feasibility; writing — review and editing.

Mathew Legg: Secondary supervision; guidance on research direction; writing — review and editing.

Chapter 5

Conclusions and Future Works

5.1 Introduction

This chapter integrates the findings from the publications presented in Chapters 3 and 4. It provides an overall synthesis of the research, discusses contributions to knowledge and practice, highlights limitations, and outlines directions for future work. The chapter is organized as follows: Section 5.2 synthesizes the findings of the two studies; Section 5.3 revisits the main contributions to knowledge; Section 5.4 identifies key limitations; and Section 5.5 proposes future work.

5.2 Synthesis of Findings

5.2.1 From Chapter 3: Automated Ground Truth and Benchmarking

Chapter 3 established the empirical foundation for the thesis by introducing an autonomous ground-truthing pipeline that couples a low-cost mobile robot with the HTC Vive tracking system. This enabled the collection of a large, high-fidelity Received Signal Strength (RSS) fingerprint dataset (7,344 points across a realistic indoor workspace), addressing the reproducibility and scale limitations identified in the literature. The chapter also documented the practical calibration steps required to make RSS-based VLP viable in a real environment, including interference mitigation and coordinate-frame alignment.

Using the resulting dataset, Chapter 3 benchmarked representative machine learning regressors against channel-model-based positioning. The results show that data volume is a decisive factor: model-based methods remained competitive when training data was scarce, while learning-based approaches improved as the dataset scaled. In this sense, the chapter

provides both a reusable dataset and a clear empirical baseline for understanding how model complexity and training data interact in photodiode-based VLP.

5.2.2 From Chapter 4: VLP–Odometry Fusion for Navigation

Chapter 4 translated the static positioning insights into dynamic navigation by building and evaluating a low-cost differential-drive robot equipped with photodiode VLP and wheel encoders. The study produced a trajectory-level dataset that combines encoder odometry, VLP-derived estimates, and external ground truth, enabling a direct assessment of closed-loop performance rather than isolated localization accuracy. This shift from static benchmarks to navigation is central to the thesis contribution, because it tests whether VLP can actually stabilize robot motion over extended runs.

The chapter also delivered a controlled comparison between single-mode baselines (encoder odometry and VLP-only) and multi-modal fusion using a Long Short-Term Memory (LSTM)-based recurrent model. Across the evaluated trajectories, the LSTM maintained higher long-horizon stability, while the single-mode baselines degraded as odometry drift accumulated or VLP noise persisted. These results demonstrate the practical value of VLP–odometry fusion and provide the empirical basis for the cross-chapter trade-offs discussed later in this synthesis.

Chapter 4 also extends the contribution beyond accuracy by evaluating edge deployment feasibility of the full inference stack on STM32-class platforms. The reported inference latencies indicate that the proposed fusion pipeline can meet typical robot localization and control update rates on resource-constrained hardware, supporting the practical deployability of the approach.

Table 5.1 provides a compact quantitative recap of the two studies without restating the full analyses.

5.2.3 Cross-Cutting Themes

While Chapters 3 and 4 addressed distinct problems, several themes emerge when the findings are considered together.

Data scale and evaluation fidelity. Both studies show that conclusions about VLP performance depend on the scale and fidelity of evaluation. Chapter 3 establishes a large, millimeter-accurate dataset that enables reliable benchmarking and exposes where small-sample results can be misleading. Chapter 4 extends this to trajectory-level evaluation, where long-horizon, closed-loop runs reveal behaviors that are not captured by pointwise accuracy

Table 5.1 Quantitative recap of the two studies.

Result	Headline metric	Implication / trade-off
Dataset scale (Ch. 3)	7,344 RSS points over 43.5 m ² from 11 luminaires	Scale enables learning-based gains; small datasets remain a constraint in practice.
Best positioning accuracy (Ch. 3)	Ridge regression mean error 84.4 mm (80/20 split); small-data case: spring relaxation 258 mm with 31 points (ridge 1562 mm)	Learning wins with scale, model-based wins when calibration data are scarce.
Navigation fusion (Ch. 4)	Late-run medians (steps 200–300) across scenarios: LSTM 0.019–0.034 m / 1.7–4.4°; VLP 0.034–0.056 m / 6.6–12.8°; Enc. Odom. 0.051–0.212 m / 14.5–68.9°	Multi-modal fusion is most stable over long horizons; single-mode baselines drift or remain noisy as errors accumulate.

alone. Together, the papers show that robust claims require both extensive data and evaluation protocols that reflect how systems are used in practice.

Single-mode limits versus multi-modal gains. Across both papers, single-mode localization remains constrained by its measurement biases. In Chapter 3, RSS-only positioning accuracy is bounded by environmental variability and calibration quality. Chapter 4 shows that encoder odometry alone accumulates drift, while VLP-only navigation remains noisy and spatially dependent. Multi-modal fusion mitigates these limitations by combining complementary signals, yielding more stable long-horizon trajectories than either single-mode baseline. This reinforces the thesis focus on evaluating fusion against strong single-mode references rather than in isolation.

Static positioning versus dynamic navigation. The two chapters also form a natural progression. Chapter 3 focused on *static* positioning benchmarks, providing foundational evidence that large-scale datasets and machine learning could yield centimeter-level accuracy. Chapter 4 advanced this to *dynamic* conditions, showing that fusion with odometry allowed a robot to maintain accurate trajectories over extended movements. Importantly, Chapter 4 implemented closed-loop control in which pose estimates directly determined the next motion commands. This addresses a key gap in the literature where localization accuracy is often evaluated without closing the loop. This progression demonstrates that the challenges of reliable static localization and dynamic navigation are not independent; the tools and

insights from dataset scale and modeling approaches carry forward into real-world robotic deployment.

Interpreting CNC and free-robot benchmarks. Comparisons between Computer Numerical Control (CNC)-style VLP benchmarks and free mobile-robot experiments should also be interpreted carefully. CNC-based data collection often uses regular geometric sampling patterns, which can unintentionally favor fingerprint-based machine-learning models if part of the spatial structure is learned implicitly rather than solely the RSS-to-position relationship. This concern motivated the use of randomly offset validation points in Section 4.3.3 rather than relying only on a perfectly regular grid. CNC platforms can also provide much tighter positional reference than free-robot experiments, whereas mobile-robot benchmarking is typically limited by millimeter-scale external tracking uncertainty and by motion-induced variation during operation. Finally, CNC benchmarks usually hold receiver pose fixed, whereas a mobile robot changes heading and may experience small tilt variations. For photodiode-based VLP, these differences matter because practical RSS measurements can depend on receiver pose through tilt, mounting asymmetry, packaging effects, or other non-ideal angular-response behavior. Static CNC benchmarks and free-robot navigation experiments therefore do not test identical localization conditions, and stronger results on one should not be interpreted as directly transferable to the other.

Unresolved challenges. Finally, both studies highlight persistent challenges that cut across VLP research. Orientation and tilt were recognized as important but unresolved sources of error in Chapter 3, and Chapter 4 did not directly address them in robotic navigation. Similarly, both studies were conducted in controlled indoor environments. Validation in larger and more complex industrial spaces represents a natural progression of this work. Another remaining research direction is the systematic comparison of model-based and learning-based fusion approaches under identical experimental conditions. In the Chapter 4 setting, this is not a single obvious benchmark: the VLP subsystem does not provide an independent heading observation, so a filter-based design must choose how to construct pseudo-heading updates, how to treat their dependence on position estimates, and how to tune the associated noise structure. These choices could materially affect the behavior of EKF, UKF, or related estimators, making rigorous comparison a substantial follow-on study rather than a trivial add-on. These themes set the stage for the limitations and future work outlined later in this chapter.

Taken together, the two studies address a central gap in the literature: the combination of large-scale, high-fidelity datasets with rigorous navigation evaluation. Chapter 3 delivers

the scale and ground truth needed to benchmark photodiode VLP reliably, while Chapter 4 demonstrates that integrating VLP with odometry can stabilize real robot trajectories over extended runs. This positions VLP as a viable component of low-cost indoor navigation systems rather than a purely laboratory technique. It also reinforces the value of Light Emitting Diode (LED) lighting as dual-purpose illumination and localization infrastructure, and the role of reproducible datasets in benchmarking low-cost systems.

A practical contribution of this thesis is the release of data and code artifacts that improve reproducibility beyond what is typical in experimental VLP studies. Chapter 3 is supported by a public dataset repository containing the full RSS-position dataset, luminaire metadata, and example analysis code, while Chapter 4 includes a public repository for the navigation and fusion study. These materials enable independent benchmarking, re-analysis, and extension of the reported methods, even though long-term usability would be strengthened by clear versioning, documentation, and archival hosting in addition to a live GitHub repository.

General principles. Beyond the specific results, the thesis supports three broader principles that inform system design and evaluation practice.

1. **Data scale influences method selection.** As training data grows, learning-based models become increasingly advantageous, whereas channel models remain more dependable when calibration data are scarce. Therefore system designers should select machine learning (ML) or model-based positioning based on expected data availability, and evaluations should report learning curves or multiple data regimes rather than single-point accuracy.
2. **Long-horizon navigation reveals failure modes hidden in static tests.** Drift, bias accumulation, and heading instability only become apparent when trajectories are executed over hundreds of steps. Consequently, navigation systems should be benchmarked with time-segmented metrics (early/late) and structured paths that expose systematic errors, not just static localization snapshots.
3. **Fusion is effective only when modality roles are explicit.** VLP provides absolute position anchoring, while odometry provides short-term motion continuity; neither is sufficient alone over long horizons. As a result, fusion designs should preserve these complementary roles and evaluations should include single-mode baselines plus perturbation tests to verify recovery behavior.

5.2.4 Research Question Answers

RQ1: How can large-scale, high-accuracy ground truth data be efficiently collected to enable robust evaluation of VLP systems?

- An automated collection pipeline couples a low-cost robot with HTC Vive tracking to capture dense fingerprints without manual repositioning.
- The approach yields a large, high-accuracy dataset (7,344 RSS points over 43.5 m²) and documents calibration/coordinate alignment steps needed for repeatable ground truth.

RQ2: How do machine learning models compare with classical channel models for VLP when trained on large experimental datasets?

- On the full dataset, ridge regression achieves the lowest mean error (84.4 mm), showing that learning-based methods benefit from scale.
- In data-scarce regimes, channel-model approaches (e.g., spring relaxation) remain competitive and can outperform ML, indicating a clear data-availability trade-off.

RQ3: How can robust closed-loop navigation be achieved using single-mode VLP, and what is required to recover full pose, including heading, from photodiode-based measurements?

- Single-mode photodiode-based VLP can support closed-loop navigation when evaluated with independent ground truth and when heading is recovered from sequential position estimates.
- Full pose recovery in this thesis is enabled by pairing photodiode-based VLP position updates with heading inferred from consecutive VLP positions and initial pose alignment at the start of each run.

RQ4: How can VLP and odometry be fused to achieve reliable long-horizon navigation, and what are the trade-offs between data-driven fusion approaches and single-mode baselines?

- LSTM-based fusion delivers stable long-horizon trajectories by combining absolute VLP anchoring with encoder motion continuity.
- Single-mode baselines are informative but limited: encoder odometry drifts over time, while VLP-only navigation remains spatially dependent and noisier across trajectories.

5.3 Contributions to Knowledge

This thesis makes five contributions to knowledge, each aligned with the research questions in Chapter 1 and evidenced across Chapters 3 and 4.

1. **Methodological:** Development of a novel, automated ground-truth and data collection system for VLP, producing the largest experimental RSS fingerprint dataset for photodiode-based VLP identified in the surveyed literature at the time of publication (7344 points over 43.5 m²; Chapter 3/*Sensors* publication).
2. **Empirical:** Robust benchmarking of machine learning and model-based positioning methods, demonstrating the conditions under which each approach is most effective (Chapter 3).
3. **Applied:** Design and implementation of a real-world robotic system that enables photodiode-based VLP navigation and supports full pose (position + heading) estimation under ground truth (Chapter 4 paper under review at *IEEE Sensors Journal*).
4. **Comparative:** A comparative evaluation that benchmarks single-mode VLP and encoder-only baselines against LSTM-based fusion, highlighting accuracy and robustness differences in both offline and live navigation experiments (Chapter 4).
5. **Knowledge:** Advancement of VLP research from theoretical exploration toward practical deployment in robotic navigation, bridging a key gap in the literature.

5.4 Limitations

Although the research presented in this thesis demonstrates the potential of Visible Light Positioning for robotic navigation, the findings should be read within clearly bounded experimental conditions. The thesis establishes feasibility and comparative performance for photodiode-based VLP and VLP–odometry fusion in controlled indoor settings, but it does not yet establish robustness across the full range of operating conditions likely to be encountered in practical deployment.

The first limitation concerns orientation and tilt. Photodiode-based VLP systems are inherently sensitive to the receiver’s angle of incidence. Chapter 3 identified this as a persistent issue in the literature, and the Chapter 4 experiments did not estimate or compensate for tilt in real-time. Instead, the navigation experiments relied on planar motion, a rigidly mounted receiver, and the assumption that incidental pitch and roll remained small. This

means the reported accuracy is strongest for near-level operation and may be optimistic for scenarios involving uneven surfaces, chassis oscillation, or deliberate changes in receiver attitude. A stronger follow-on study would instrument receiver attitude directly and evaluate performance across controlled tilt ranges. This is particularly important because yaw changes arise naturally during navigation, whereas unmeasured pitch and roll more directly alter the photodiode incidence geometry and can therefore bias the RSS-to-position relationship even when the robot remains in the same spatial region.

A second limitation relates to environment and scale. Both studies were conducted in controlled indoor spaces with fixed lighting geometry, limited clutter, and relatively stable optical conditions. These choices were appropriate for isolating algorithm behavior and for producing reproducible benchmarks, but they also constrain generalization. The thesis does not yet show how the proposed methods behave in larger, more cluttered, or mixed-use environments where occlusion, heterogeneous reflections, lighting variation, or partial recalibration could materially affect performance. As a result, deployment claims should be interpreted as laboratory-scale evidence of potential rather than proof of industrial robustness.

The third limitation concerns platform dependence. The Chapter 4 robot was intentionally simple, low-cost, and differential-drive, which aligned with the thesis objective of studying resource-constrained platforms. However, the observed navigation behavior and fusion benefits are still specific to this robot architecture, its controller, and its motion profile. The thesis does not yet establish that the same gains will transfer directly to robots with different drive kinematics, higher speeds, stronger vibration, or more complex autonomy stacks.

Finally, the ground-truth system employed for evaluation provided millimeter-level reference accuracy, which was valuable for rigorous laboratory benchmarking, but it is not itself a deployable solution. This creates a gap between research verification and operational validation. The thesis therefore offers strong evidence for controlled experimental evaluation, while leaving open the question of how best to validate comparable performance in larger real-world environments using deployable reference systems.

Taken together, these limitations do not weaken the core contributions of the thesis; rather, they clarify the boundaries of what has been demonstrated. Specifically, the thesis shows that photodiode-based VLP and VLP–odometry fusion can support accurate near-level indoor navigation in the tested controlled settings. It does not yet establish robustness to significant tilt, cross-site transfer, cluttered industrial deployments, or broad robot-platform generalization. These boundaries form a natural bridge to the future-work directions discussed next.

5.5 Future Work

The limitations identified in this thesis also provide opportunities for further investigation. Extending the research in these directions will both strengthen the practical utility of VLP for robotic navigation and broaden its theoretical contributions to the field of indoor positioning.

5.5.1 Technical Extensions

One clear direction is addressing the persistent challenge of orientation and tilt. Future implementations should be developed along three complementary paths. The first is mechanical stabilization, such as more rigid sensor mounting, lower-vibration chassis design, passive leveling, or isolated sensor mounts that reduce pitch and roll at the source. The second is explicit sensing and compensation, where an IMU or comparable attitude sensor is used to estimate receiver tilt and correct, down-weight, or reject VLP observations that are inconsistent with the expected photodiode response. The third is data-driven robustness, where training and validation data are deliberately collected over a range of controlled receiver attitudes so that fingerprinting or fusion models learn to remain stable under moderate tilt variation. Exploring these strategies experimentally, rather than only in simulation, would help establish their viability for deployment in mobile robots. Extending such approaches to reliable 3D positioning with photodiode receivers remains an open direction.

Another technical extension lies in the environments and datasets used for evaluation. While the experiments in this thesis were conducted in controlled indoor laboratories, future studies should examine performance in larger and more complex spaces such as warehouses, hospitals, or retail environments. These contexts introduce multipath reflections, occlusions, and heterogeneous lighting that can stress both calibration and localization. Expanding the Chapter 3 dataset across sites and lighting configurations, and re-validating ground-truth alignment, would strengthen generalization claims and enable cross-site benchmarking.

At industrial scale, the challenge is not only whether accuracy degrades, but also whether the full workflow remains practical. Larger deployments are likely to increase infrastructure-planning burden, calibration and re-calibration effort, and sensitivity to layout changes, occlusion, and partial lighting failure. This suggests that scalable VLP systems may need zone-based or hierarchical localization workflows, where coarse location is established first and higher-accuracy tracking is then maintained within locally calibrated regions.

Finally, testing with more sophisticated robotic platforms will help determine the generality of the results. Extending sensor fusion methods to legged robots, autonomous vehicles, or drones would provide evidence of scalability across robotic architectures and reveal new challenges related to dynamics, computation, and energy efficiency.

5.5.2 Algorithmic Extensions

There needs to be a direct benchmark of model-based fusion (e.g. EKF / UKF variants) against LSTM-based fusion using identical data, trajectories, and evaluation protocols. This would establish when analytic structure is sufficient and when learned temporal models provide measurable advantages. In this thesis, the challenge is that heading is not measured directly by the VLP subsystem, but inferred from successive positions, so a filter-based baseline must make explicit choices about pseudo-measurement construction, dependence handling, and covariance tuning. A fair model-based benchmark therefore requires careful formulation rather than simply inserting a generic EKF. Beyond this, hybrid methods that combine model-based updates with learned corrections remain promising. Emerging architectures such as temporal convolutional networks or transformers may capture longer-range dependencies in VLP and odometry data, while the perturbation experiments in Chapter 4 motivate training regimes that inject synthetic capture events and evaluate recovery time and post-capture error alongside compute cost.

A further priority is end-to-end onboard validation on the target robot. While Chapter 4 provides initial embedded profiling that indicates feasible latency and memory use on low-power platforms, full in situ validation remains needed. The key next step is therefore confirming closed-loop timing, throughput, and power behavior under sustained operation, rather than relying only on tool-based deployment estimates.

5.5.3 Deployment Pathways

Beyond technical and algorithmic advances, future work should consider the pathways to deployment at scale. Integration of VLP into commercial LED lighting infrastructure would allow positioning to leverage existing installations, but will require standardization of modulation schemes, interference management, and interoperability with lighting controls. Pilot deployments in logistics, healthcare, or manufacturing environments could demonstrate both feasibility and return on investment. Such deployments should also be expected to expose performance limitations not fully visible in the present thesis, including sensitivity to clutter, changing lighting geometry, greater receiver tilt, and reduced calibration stability over time.

VLP should also be considered as part of a heterogeneous localization ecosystem rather than as a universally preferred standalone solution. Relative to mainstream indoor robot-localization technologies such as Wi-Fi, Bluetooth, and UWB, photodiode-based VLP is attractive because it can provide high positioning accuracy while using a simple, low-cost receiver. Its main trade-offs are greater dependence on a stable optical environment and a

more substantial infrastructure requirement, since VLP luminaires typically provide more limited coverage than RF anchors. However, this burden can be partly offset by the dual use of LED lighting for both illumination and positioning.

These trade-offs suggest a practical role for VLP within hybrid stacks. Combining it with technologies such as IMU, Wi-Fi, Bluetooth, UWB, or vision could provide redundancy and robustness while allowing each modality to play to its strengths. An IMU can support short-term motion continuity and provide attitude estimates that help interpret or reject distorted VLP observations. UWB can provide coarse global coverage, initialization, and continuity through regions where lighting geometry is sparse or temporarily blocked. VLP can then contribute high local accuracy in well-instrumented spaces. The Chapter 4 results already support the value of this complementarity, because long-horizon behavior improved when sensing modes with different strengths were combined rather than used in isolation. In that sense, the contribution of VLP is not that it universally replaces RF-based localization, but that it offers a credible low-cost, high-accuracy component within broader indoor localization systems.

Finally, the publicly released fingerprinting dataset from Chapter 3 provides a starting point for community benchmarks. Encouraging independent re-use, for example through shared leaderboards or comparative studies, would help standardize evaluation practice and accelerate progress on reproducible VLP research.

5.5.4 Summary

In summary, future research should: (i) tackle unresolved orientation and tilt issues through novel receiver designs or hybrid sensing, and extend PD positioning to reliable 3D localization; (ii) expand datasets and validate performance in diverse, complex environments; (iii) extend evaluation to a wider range of robotic platforms; (iv) prioritise benchmarking of model-based fusion (e.g. EKF variants) against LSTM fusion and explore advanced and hybrid algorithms, including real-time and embedded implementations; and (v) pursue deployment pathways that integrate VLP with both lighting systems and other indoor positioning technologies while cultivating shared datasets and benchmarks.

Pursuing these directions will build on the contributions of this thesis and further advance VLP from a promising research domain to a mature and deployable solution for robotic navigation.

5.6 Closing Remarks

In closing, this thesis connects large-scale, ground-truthed VLP benchmarking with trajectory-level navigation evaluation, demonstrating that PD-based VLP can underpin reliable motion when fused with odometry. The work therefore positions VLP as a practical component of low-cost indoor navigation systems and provides a foundation for future deployment studies.

References

- [1] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.
- [2] T.-H. Do and M. Yoo, “An in-depth survey of visible light communication based positioning systems,” *Sensors*, vol. 16, no. 5, p. 678, 2016.
- [3] M. Afzalan and F. Jazizadeh, “Indoor positioning based on visible light communication: A performance-based survey of real-world prototypes,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–36, 2019.
- [4] F. Alam, B. Parr, and S. Mander, “Visible light positioning based on calibrated propagation model,” *IEEE Sensors Letters*, vol. 3, no. 2, pp. 1–4, 2018.
- [5] F. Alam, N. Faulkner, M. Legg, and S. Demidenko, “Indoor visible light positioning using spring-relaxation technique in real-world setting,” *IEEE Access*, vol. 7, pp. 91 347–91 359, 2019.
- [6] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, “A review of sensing technologies for indoor autonomous mobile robots,” *Sensors*, vol. 24, no. 4, p. 1222, Jan. 2024.
- [7] Y.-H. Hsueh, P.-C. Su, Y.-F. Lin, H.-H. Huang, and T.-C. Liu, “Robot indoor navigation using visible light communication,” in *2018 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)*, Oct. 2018, pp. 1–6.
- [8] M.-T. Chew, F. Alam, F. K. Noble, M. Legg, and G. S. Gupta, “Visible light positioning-based robot localization and navigation,” *Electronics*, vol. 13, no. 2, p. 368, Jan. 2024.
- [9] D. Konings, B. Parr, C. Waddell, F. Alam, K. M. Arif, and E. M. Lai, “HVLP: Hybrid visible light positioning of a mobile robot,” in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2017, pp. 1–6.
- [10] W. Guan, L. Huang, S. Wen, Z. Yan, W. Liang, C. Yang, and Z. Liu, “Robot localization and navigation using visible light positioning and SLAM fusion,” *Journal of Lightwave Technology*, vol. 39, no. 22, pp. 7040–7051, Nov. 2021.
- [11] R. Amsters, D. Holm, J. Joly, E. Demeester, N. Stevens, and P. Slaets, “Visible Light Positioning Using Bayesian Filters,” *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5925–5936, Nov. 2020.

- [12] K. Bera, R. Parthiban, and N. Karmakar, "A truly 3d visible light positioning system using low-resolution high-speed camera, LiDAR, and IMU sensors," *IEEE Access*, vol. 11, pp. 98 578–98 585, 2023.
- [13] G. Chen, W. Chen, Q. Yang, Z. Xu, L. Yang, J. Conradt, and A. Knoll, "A novel visible light positioning system with event-based neuromorphic vision sensor," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 211–10 219, Sep. 2020.
- [14] X. Li, Z. Yan, L. Huang, S. Chen, and M. Liu, "High-accuracy and real-time indoor positioning system based on visible light communication and mobile robot," *International Journal of Optics*, vol. 2020, no. 1, p. 3124970, 2020.
- [15] Z. Yan, W. Guan, S. Wen, L. Huang, and H. Song, "Multirobot cooperative localization based on visible light positioning and odometer," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–8, 2021.
- [16] Q. Liang, Y. Sun, C. Liu, M. Liu, and L. Wang, "LedMapper: Toward efficient and accurate LED mapping for visible light positioning at scale," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.
- [17] R. Amsters, E. Demeester, N. Stevens, and P. Slaets, "Calibration of visible light positioning systems with a mobile robot," *Sensors*, vol. 21, no. 7, p. 2394, Jan. 2021.
- [18] X. Sun, Y. Zhuang, X. Yang, J. Huai, T. Huang, and D. Feng, "Tightly coupled VLP/INS integrated navigation by inclination estimation and blockage handling," *Satellite Navigation*, vol. 6, no. 1, p. 7, Mar. 2025.
- [19] X. Wang, Y. Zhuang, Y. Huang, X. Cao, T. Yu, J. Zhou, Z. Zheng, and N. El-Sheimy, "A robust data-model dual-drive fusion for IMU and visible light integrated localization in complex changing environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 4, pp. 8922–8938, Aug. 2025.
- [20] T. Yoshida, K. Kaji, S. Ogiso, R. Ichikari, H. Uchiyama, T. Kurata, and N. Kawaguchi, "A survey of ground truth measurement systems for indoor positioning," *Journal of Information Processing*, vol. 31, pp. 15–20, 2023.
- [21] D. Lee, M. Jung, W. Yang, and A. Kim, "LiDAR odometry survey: Recent advancements and remaining challenges," *Intelligent Service Robotics*, vol. 17, no. 2, pp. 95–118, Mar. 2024.
- [22] M. Khoshrangbaf, V. K. Akram, M. Challenger, and O. Dagdeviren, "An experimental evaluation of indoor localization in autonomous mobile robots," *Sensors*, vol. 25, no. 7, p. 2209, Jan. 2025.
- [23] Z. Zhu, Y. Yang, M. Chen, C. Guo, J. Cheng, and S. Cui, "A survey on indoor visible light positioning systems: Fundamentals, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 3, pp. 1656–1686, Jun. 2025.
- [24] N. Stevens and H. Steendam, "Influence of transmitter and receiver orientation on the channel gain for RSS ranging-based VLP," in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*. IEEE, 2018, pp. 1–5.

- [25] J. K. Park and J. T. Kim, "Interleaved binary demodulation technique for low-cost and fast-response visible light positioning using palindromic sequences," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 653–665, Jan. 2024.
- [26] Q. Zhao, T. Yang, G. Tang, Y. Yang, F. Dong, Z. Xi, Y. Zou, M. Xu, S. Li, C. Wang, and G. Xie, "Bio-inspired swarm of underwater robots: A review," *Bioinspiration & Biomimetics*, vol. 20, no. 4, p. 041002, Jun. 2025.
- [27] X. Cao, Y. Zhuang, G. Chen, X. Wang, X. Yang, and B. Zhou, "A visible light positioning system based on a particle filter and deep learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 3, pp. 2735–2748, Jun. 2024.
- [28] Y. S. Eroglu, I. Guvenc, N. Pala, and M. Yuksel, "AOA-based localization and tracking in multi-element VLC systems," in *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*. IEEE, 2015, pp. 1–5.
- [29] S.-H. Yang, H.-S. Kim, Y.-H. Son, and S.-K. Han, "Three-dimensional visible light indoor localization using AOA and RSS with multiple optical receivers," *Journal of Lightwave Technology*, vol. 32, no. 14, pp. 2480–2485, 2014.
- [30] P. Saengudomlert, P. Ubolkosold, and K. L. Sterckx, "Development of a position and orientation estimation system using extended Kalman filtering for indoor visible light positioning," *AEU - International Journal of Electronics and Communications*, vol. 192, p. 155684, Mar. 2025.
- [31] E. Aparicio-Esteve, Á. Hernández, J. Ureña, and J. M. Villadangos, "Visible light positioning system based on a quadrant photodiode and encoding techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5589–5603, Aug. 2020.
- [32] T. Yuan, Y. Xu, Y. Wang, P. Han, and J. Chen, "A tilt receiver correction method for visible light positioning using machine learning method," *IEEE Photonics Journal*, vol. 10, no. 6, pp. 1–12, 2018.
- [33] H. Yang, S. Zhang, A. Alphones, C. Chen, K.-Y. Lam, Z. Xiong, L. Xiao, and Y. Zhang, "An advanced integrated visible light communication and localization system," *IEEE Transactions on Communications*, vol. 71, no. 12, pp. 7149–7162, Dec. 2023.
- [34] M. Vieira, R. Costa, A. Pereira, and P. Fonseca, "A validation framework for visible light positioning in mobile robotics," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, May 2016, pp. 47–52.
- [35] Y.-H. Shu, Y.-H. Chang, Y.-Z. Lin, and C.-W. Chow, "Real-time indoor visible light positioning (VLP) using long short-term memory neural network (LSTM-NN) with principal component analysis (PCA)," *Sensors*, vol. 24, no. 16, p. 5424, Jan. 2024.
- [36] S. Liu, X. Wang, J. Song, and Z. Han, "Cooperative robotics visible light positioning: An intelligent compressed sensing and GAN-enabled framework," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 3, pp. 407–418, Apr. 2024.

- [37] D.-C. Lin, C.-W. Chow, C.-W. Peng, T.-Y. Hung, Y.-H. Chang, S.-H. Song, Y.-S. Lin, Y. Liu, and K.-H. Lin, "Positioning unit-cell model duplication with residual concatenation neural network (RCNN) and transfer learning for visible light positioning (VLP)," *Journal of Lightwave Technology*, vol. 39, no. 20, pp. 6366–6372, Oct. 2021.
- [38] Z. Wang, W. Wang, X. Liu, L. Yu, C. Chen, and X. Zhang, "Fusion-driven visible light positioning using photodiode and event camera," *Optics Express*, vol. 33, no. 18, pp. 38 449–38 465, Sep. 2025.
- [39] W. Yang, L. Qin, X. Hu, and D. Zhao, "Indoor visible-light 3D positioning system based on GRU neural network," *Photonics*, vol. 10, no. 6, p. 633, 2023. [Online]. Available: <https://www.mdpi.com/2304-6732/10/6/633>
- [40] Z. Zhu, Y. Yang, M. Chen, C. Guo, J. Hao, and S. Cui, "Visible light positioning with visual odometry: A single-luminaire-based positioning algorithm," *IEEE Transactions on Communications*, vol. 72, no. 8, pp. 4978–4991, Aug. 2024.
- [41] M. Irshad, W. Liu, L. Wang, and M. U. R. Khalil, "Cogent machine learning algorithm for indoor and underwater localization using visible light spectrum," *Wireless Personal Communications*, vol. 116, no. 2, pp. 993–1008, Jan. 2021.
- [42] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [43] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.
- [44] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, Apr. 1997, pp. 2783–2788 vol.4.
- [45] H. Sharifi, A. Kumar, F. Alam, and K. M. Arif, "Indoor localization of mobile robot with visible light communication," in *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2016, pp. 1–6.
- [46] P.-H. Hunag, S.-C. Liu, and C.-H. Yang, "Improved RTAB-Map algorithm based on visible light positioning," *IEEE Access*, vol. 13, pp. 136 854–136 863, 2025.
- [47] S. Xu, L. Zhang, X. Wang, J. Chen, F. Wei, and Y. Wu, "Indoor cooperative localization for a swarm of micro UAVs based on visible light communication," *IEEE Systems Journal*, vol. 17, no. 4, pp. 6504–6515, Dec. 2023.
- [48] W. Guan, S. Chen, S. Wen, Z. Tan, H. Song, and W. Hou, "High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning," *IEEE Photonics Journal*, vol. 12, no. 2, pp. 1–15, 2020. [Online]. Available: <https://doi.org/10.1109/JPHOT.2020.2981485>
- [49] W. Liang, S. Wen, L. Huang, and W. Guan, "High-accuracy robot indoor navigation using visible light positioning and LiDAR fusion," in *2022 Conference on Lasers and Electro-Optics (CLEO)*, May 2022, pp. 1–2.

- [50] T. Hiraki, S. Fukushima, and T. Naemura, "Projection-based localization and navigation method for multiple mobile robots with pixel-level visible light communication," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, Dec. 2016, pp. 862–868.
- [51] L. Hua, Y. Zhuang, Y. Li, Q. Wang, B. Zhou, L. Qi, J. Yang, Y. Cao, and H. Haas, "FusionVLP: The fusion of photodiode and camera for visible light positioning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 796–11 811, Nov. 2021.
- [52] X. Wang, X. Cao, T. Yu, Z. Zheng, Z. Zhang, and Y. Yu, "VLP-BERT: BERT-enhanced IMU and visible light tightly coupled integration positioning system," *IEEE Internet of Things Journal*, vol. 12, no. 17, pp. 36 697–36 712, Sep. 2025.
- [53] X. Yang, Y. Zhuang, M. Shi, Q. Meng, J. Xiong, and Y. Cao, "DeepVLP: A graph-neural-network-based denoising and signal optimization framework for visible light positioning," *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5106–5123, Jun. 2025.
- [54] Z. Li, W. Yang, L. Xiao, X. Xiong, Z. Wang, and X. Zou, "Integrated wearable indoor positioning system based on visible light positioning and inertial navigation using Unscented Kalman Filter," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, Oct. 2019, pp. 1–6.
- [55] M. Windolf, N. Götzén, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the Vicon-460 system," *Journal of Biomechanics*, vol. 41, no. 12, pp. 2776–2780, 2008.
- [56] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, "ALPS: A Bluetooth and ultrasound platform for mapping and localization," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 73–84.
- [57] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Artech House, 2002.
- [58] E. Aitenbichler and M. Muhlhauser, "An IR local positioning system for smart items and devices," in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings*. IEEE, 2003, pp. 334–339.
- [59] C. Yang and H.-R. Shao, "WiFi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.
- [60] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "RSSI-based Bluetooth low energy indoor positioning," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014, pp. 526–533.
- [61] W.-H. Kuo, Y.-S. Chen, G.-T. Jen, and T.-W. Lu, "An intelligent positioning approach: RSSI-based indoor and outdoor localization scheme in Zigbee networks," in *2010 International Conference on Machine Learning and Cybernetics*, vol. 6. IEEE, 2010, pp. 2754–2759.

- [62] Y. Zhou, C. L. Law, Y. L. Guan, and F. Chin, "Indoor elliptical localization based on asynchronous UWB range measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 248–257, 2010.
- [63] C. Sertatı, M. A. Altinkaya, and K. Raoof, "A novel acoustic indoor localization system employing CDMA," *Digital Signal Processing*, vol. 22, no. 3, pp. 506–517, 2012.
- [64] J. Z. Liang, N. Corso, E. Turner, and A. Zakhor, "Image based localization in indoor environments," in *2013 Fourth International Conference on Computing for Geospatial Research and Application*. IEEE, 2013, pp. 70–75.
- [65] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2047–2077, 2015.
- [66] M. Maheepala, A. Z. Kouzani, and M. A. Joordens, "Light-based indoor positioning systems: A review," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 3971–3995, Apr. 2020.
- [67] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014, pp. 447–458.
- [68] Y. Chen, W. Guan, J. Li, and H. Song, "Indoor real-time 3D visible light positioning system using fingerprinting and extreme learning machine," *IEEE Access*, vol. 8, pp. 13 875–13 886, 2019.
- [69] F. Alam, M. T. Chew, T. Wenge, and G. S. Gupta, "An accurate visible light positioning system using regenerated fingerprint database based on calibrated propagation model," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2714–2723, 2018.
- [70] X. Guo, F. Hu, N. R. Elikplim, and L. Li, "Indoor localization using visible light via two-layer fusion network," *IEEE Access*, vol. 7, pp. 16 421–16 430, 2019.
- [71] J. Vongkulbhisal, B. Chantaramolee, Y. Zhao, and W. S. Mohammed, "A fingerprinting-based indoor localization system using intensity modulation of light emitting diodes," *Microwave and Optical Technology Letters*, vol. 54, no. 5, pp. 1218–1227, 2012.
- [72] H. Zhang, J. Cui, L. Feng, A. Yang, H. Lv, B. Lin, and H. Huang, "High-precision indoor visible light positioning using deep neural network based on the Bayesian regularization with sparse training point," *IEEE Photonics Journal*, vol. 11, no. 3, pp. 1–10, Jun. 2019.
- [73] H. Zhang, J. Cui, L. Feng, A. Yang, H. Lv, B. Lin, and H. Huang, "High-precision indoor visible light positioning using modified momentum back propagation neural network with sparse training point," *Sensors*, vol. 19, no. 10, p. 2324, 2019.
- [74] Y.-C. Chuang, Z.-Q. Li, C.-W. Hsu, Y. Liu, and C.-W. Chow, "Visible light communication and positioning using positioning cells and machine learning algorithms," *Optics Express*, vol. 27, no. 11, pp. 16 377–16 383, 2019.

- [75] Y. Wu, K. Hsu, Y. Liu, C. Hong, C. Chow, C. Yeh, X. Liao, K. Lin, and Y. Chen, "Using linear interpolation to reduce the training samples for regression based visible light positioning system," *IEEE Photonics Journal*, vol. 12, no. 2, pp. 1–5, Apr. 2020.
- [76] R. Amsters, E. Demeester, N. Stevens, and P. Slaets, "In-depth analysis of unmodulated visible light positioning using the iterated extended Kalman filter," *Sensors*, vol. 19, no. 23, p. 5198, 2019.
- [77] R. Amsters, E. Demeester, N. Stevens, Q. Lauwers, and P. Slaets, "Evaluation of Low-Cost/High-Accuracy Indoor Positioning Systems," in *Proceedings of the 2019 International Conference on Advances in Sensors, Actuators, Metering and Sensing (ALLSENSORS)*, Athens, Greece, 2019, pp. 24–28.
- [78] HTC Corporation, "Vive tracker guidelines," Developer documentation, 2018, accessed July 27, 2026. [Online]. Available: <https://developer.vive.com/resources/hardware-guides/vive-tracker-developer-guidelines/>
- [79] Roborock, "Roborock s5 robot vacuum and mop," Product page, 2021, accessed August 12, 2026. [Online]. Available: <https://us.roborock.com/pages/roborock-s5>
- [80] OSRAM Opto Semiconductors, "SFH 2440 silicon PIN photodiode datasheet," Manufacturer datasheet, 2021, accessed August 12, 2026. [Online]. Available: https://dammedia.osram.info/media/resource/hires/osram-dam-5467146/SFH%202440_EN.pdf
- [81] STMicroelectronics, "STM32F103CB mainstream performance line, ARM cortex-M3 microcontroller," Product page, 2021, accessed August 12, 2026. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103cb.html>
- [82] Espressif Systems, "ESP8266 wi-Fi microcontroller," Product page, 2021, accessed August 12, 2026. [Online]. Available: <https://www.espressif.com/en/products/socs/esp8266>
- [83] P. Caserman, A. Garcia-Agundez, R. Konrad, S. Göbel, and R. Steinmetz, "Real-time body tracking in virtual reality using a Vive tracker," *Virtual Reality*, vol. 23, no. 2, pp. 155–168, 2019.
- [84] A. Peer, P. Ullrich, and K. Ponto, "Vive tracking alignment and correction made easy," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 653–654.
- [85] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.
- [86] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, "HTC Vive: Analysis and accuracy improvement," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2610–2615.
- [87] A. Borrego, J. Latorre, M. Alcaniz, and R. Llorens, "Comparison of Oculus Rift and HTC Vive: Feasibility for virtual reality-based exploration, navigation, exergaming, and rehabilitation," *Games for Health Journal*, vol. 7, no. 3, pp. 151–156, 2018.

- [88] Valve Corporation, “SteamVR,” Software platform page, 2021, accessed August 12, 2026. [Online]. Available: <https://store.steampowered.com/steamvr>
- [89] C. Bruns, “pyopenvr documentation,” GitHub repository documentation, 2020, accessed July 27, 2026. [Online]. Available: <https://github.com/cmbruns/pyopenvr>
- [90] Valve Corporation, “OpenVR software development kit,” GitHub repository and API documentation, 2020, accessed July 27, 2026. [Online]. Available: <https://github.com/ValveSoftware/openvr>
- [91] Triad Semiconductor, “Triad OpenVR python wrapper,” GitHub repository, 2020, accessed July 27, 2026. [Online]. Available: https://github.com/TriadSemi/triad_openvr
- [92] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, pp. 15–16, 2006.
- [93] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.
- [94] Open3D Contributors, “Open3D: A modern library for 3D data processing,” Project website, 2021, accessed August 12, 2026. [Online]. Available: <http://www.open3d.org/>
- [95] F. Giese, “Dustcloud documentation,” GitHub repository documentation, 2018, accessed July 27, 2026. [Online]. Available: <https://github.com/dgiese/dustcloud-documentation/tree/master/roborock.vacuum.s5>
- [96] D. Giese, “Security Analysis of the Xiaomi IoT Ecosystem,” Master’s thesis, Darmstadt University of Technology, Darmstadt, Germany, 2019.
- [97] Home Assistant Contributors, “Home assistant,” GitHub repository, 2021, accessed August 12, 2026. [Online]. Available: <https://github.com/home-assistant>
- [98] Home Assistant Contributors, “Xiaomi miio integration,” Integration documentation, 2021, accessed August 12, 2026. [Online]. Available: https://www.home-assistant.io/integrations/xiaomi_miio/
- [99] Xiaomi Inc., “Xiaomi home,” Mobile application listing, 2021, accessed August 12, 2026. [Online]. Available: https://play.google.com/store/apps/details?id=com.xiaomi.smarthome&hl=en_NZ&gl=US
- [100] HTC Corporation, “Vive base station 2.0,” Product page, 2021, accessed August 12, 2026. [Online]. Available: <https://www.vive.com/eu/accessory/base-station2/>
- [101] B. Lang, “HTC experiments with 16 base stations for SteamVR tracking 2.0, unlocking multi-room VR,” Road to VR article, 2017, accessed August 12, 2026. [Online]. Available: <https://www.roadtovr.com/htc-experiments-with-16-base-stations-steamvr-tracking-2-0-multi-room/>
- [102] T. Glass, “Public VLP dataset,” GitHub repository, 2021, accessed August 12, 2026. [Online]. Available: <https://github.com/tyrel-glass/Public-VLP-Dataset>

- [103] W. Gu, M. Aminikashani, P. Deng, and M. Kavehrad, "Impact of multipath reflections on the performance of indoor visible light positioning systems," *Journal of Lightwave Technology*, vol. 34, no. 10, pp. 2578–2587, 2016.
- [104] R. M. Rifkin and R. A. Lippert, "Notes on regularized least squares," MIT Computer Science and Artificial Intelligence Laboratory, Technical Report MIT-CSAIL-TR-2007-025, May 2007. [Online]. Available: <https://dspace.mit.edu/handle/1721.1/37318>
- [105] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [106] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [107] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.
- [108] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: <https://jmlr.org/papers/v12/pedregosa11a.html>
- [109] R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*. John Wiley & Sons, 2011, vol. 27.
- [110] J. He, C. Hsu, Q. Zhou, M. Tang, S. Fu, D. Liu, L. Deng, and G. Chang, "Demonstration of high precision 3D indoor positioning system based on two-layer ANN machine learning technique," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, Mar. 2019, pp. 1–3.
- [111] S. Zhang, P. Du, C. Chen, and W. Zhong, "3D indoor visible light positioning system using RSS ratio with neural network," in *2018 23rd Opto-Electronics and Communications Conference (OECC)*, Jul. 2018, pp. 1–2.
- [112] S. Ni, F. Wang, S. Han, J. Pang, and S. You, "Two methods of accurate indoor positioning based on visible light communication system and artificial neural network," in *2019 18th International Conference on Optical Communications and Networks (ICOON)*. IEEE, 2019, pp. 1–3.
- [113] I. Alonso-González, D. Sánchez-Rodríguez, C. Ley-Bosch, and M. A. Quintana-Suárez, "Discrete indoor three-dimensional localization system based on neural networks using visible light communication," *Sensors (Basel, Switzerland)*, vol. 18, no. 4, p. 1040, Mar. 2018.
- [114] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

- [115] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. MIT Press, 2011.
- [116] C.-Y. Hong, Y.-C. Wu, Y. Liu, C.-W. Chow, C. H. Yeh, K.-L. Hsu, D.-C. Lin, X.-L. Liao, K.-H. Lin, and Y.-Y. Chen, “Angle-of-arrival (AOA) visible light positioning (VLP) system using solar cells with third-order regression and ridge regression algorithms,” *IEEE Photonics Journal*, vol. 12, no. 3, pp. 1–5, Jun. 2020.
- [117] D. Konings, N. Faulkner, F. Alam, E. M.-K. Lai, and S. Demidenko, “FieldLight: Device-free indoor human localization using passive visible light positioning and artificial potential fields,” *IEEE Sensors Journal*, vol. 20, no. 2, pp. 1054–1066, 2019.
- [118] W. Guan, L. Huang, B. Hussain, and C. P. Yue, “Robust robotic localization using visible light positioning and inertial fusion,” *IEEE Sensors Journal*, vol. 22, no. 6, pp. 4882–4892, Mar. 2022.
- [119] T. Glass, “VLP-Odometry-Fusion,” GitHub repository, 2025, accessed July 27, 2026. [Online]. Available: <https://github.com/tyrel-glass/VLP-Odometry-Fusion>
- [120] R. Zhu, M. Van den Abeele, J. Beysens, J. Yang, and Q. Wang, “Centimeter-level indoor visible light positioning,” *IEEE Communications Magazine*, vol. 62, no. 3, pp. 48–53, Mar. 2024.
- [121] T. Glass, F. Alam, M. Legg, and F. K. Noble, “Autonomous fingerprinting and large experimental data set for visible light positioning,” *Sensors*, vol. 21, no. 9, p. 3256, May 2021.
- [122] N. Faulkner, B. Parr, F. Alam, M. Legg, and S. N. Demidenko, “CapLoc: Capacitive sensing floor for device-free localization and fall detection,” *IEEE Access*, vol. 8, pp. 187 353–187 364, Oct. 2020.
- [123] Z. Huang, G. Ye, P. Yang, and W. Yu, “Application of multi-sensor fusion localization algorithm based on recurrent neural networks,” *Scientific Reports*, vol. 15, no. 1, p. 8195, 2025.
- [124] T. Moore, S. Macenski, and contributors, “robot_localization: Nonlinear State Estimation Nodes for ROS,” ROS package, Robot Operating System, Charles River Analytics, Inc., 2026, GitHub repository and package documentation; version 3.10.1; accessed July 27, 2026. [Online]. Available: https://github.com/cra-ros-pkg/robot_localization
- [125] E. Marder-Eppstein and R. N. contributors, “ROS Navigation Stack (move_base, noetic-devel branch),” GitHub repository source code, Open Source Robotics Foundation, 2026, archived repository; accessed July 27, 2026. [Online]. Available: https://github.com/ros-planning/navigation/blob/noetic-devel/move_base/src/move_base.cpp
- [126] ArduPilot Development Team, “ArduPilot Rover Scheduler Loop Rate (SCHED_LOOP_RATE),” Project documentation, ArduPilot Project, 2026, accessed July 27, 2026. [Online]. Available: <https://ardupilot.org/rover/docs/parameters.html>