

SEAMLESS INTEGRATION OF GPS AND UWB DATA FOR CONSTRUCTION
SITE MONITORING IN REVIT

by

Esma Rezan Karaca

B.S., Civil Engineering, Istanbul Technical University, 2021

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Civil Engineering
Boğaziçi University

2025

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Assoc. Prof. Semra Çomu, for her continuous guidance, encouragement, and invaluable support throughout this research. I am especially grateful for her endless patience, forbearance, and kind understanding during this journey; her willingness to tolerate my shortcomings and to support me at every step has been truly invaluable. I would like to share my appreciation to the thesis jury members Prof. Dr. Beliz Özorhon, Assist. Prof. Dr. Fahri Serhan Daniş and my thesis advisor Assoc. Prof. Dr. Semra Çomu Yapıcı for their valuable insights and knowledge.

I would also like to extend my sincere thanks to Research Assistant Umutcan Kılıç and Serdar Bayraktar for their kind assistance in the data collection process, which played an essential role in the successful completion of this study. Furthermore, I would like to extend my sincere gratitude to M.Sc. Geomatics Engineer Yaşar Topal, M.Sc. Geomatics Engineer, and M.Sc. Geomatics Engineer Burak Kotan, for their invaluable assistance with the GNSS measurements. I am also deeply grateful to Prof. Semih Ergintav from Bogazici University Kandilli Observatory for generously providing support with instruments and equipment. It is with the utmost respect that I acknowledge their kind contributions to this study.

Furthermore, I am thankful to my friends and family (especially Zülal Karaca) for their unwavering support and encouragement, which sustained me throughout this challenging yet rewarding period. I am deeply grateful to my dear friends, Beyza Nur Eraslan and Gizem Alişoğlu, for their invaluable support during this process.

Finally, I would like to dedicate this thesis to everyone who believed in me and supported my efforts to bring this research to completion.

ABSTRACT

SEAMLESS INTEGRATION OF GPS AND UWB DATA FOR CONSTRUCTION SITE MONITORING IN REVIT

This thesis proposes a hybrid real-time location tracking framework designed for construction site safety, which integrates Global Positioning System (GPS) and Ultra-Wideband (UWB) data using an Unscented Kalman Filter (UKF). The fused positions are dynamically mapped into Revit via the Autodesk Revit API and visualized using custom Python scripts in pyRevit. The experimental setup involved both indoor and outdoor test areas across two buildings. UWB anchors were deployed indoors, while test subjects wore mobile tags to simulate worker movement. Outdoor position data was collected via iOS-based GPS smartphone applications. The data was post-processed and synchronized, then filtered using Mahalanobis distance to remove 3D outliers before applying UKF-based fusion. Transformed coordinates were visualized in Revit, enabling tracking and visualization of the workers. The experimental results revealed critical limitations in the accuracy of the iOS-based GPS data. Positional deviations in open outdoor areas frequently exceeded 6 meters and reached nearly 10 meters in obstructed environments. The study recommends adopting GNSS receivers such as dual-frequency RTK systems or integrating inertial navigation sensors to enhance outdoor data fidelity. Despite the shortcomings of GPS, the UWB-based indoor localization performed well under line-of-sight (LOS) conditions. Overall, the research demonstrates that the proposed GPS–UWB–UKF fusion system significantly improves positional continuity and spatial awareness when integrated in Revit. This study contributes an extendable methodology for real-time worker safety management based on sensor fusion, spatial filtering, and Revit integration while simultaneously underscoring the practical hardware and calibration limitations in mobile sensor systems.

ÖZET

REVIT ORTAMINDA ŞANTIYE TAKİBİ İÇİN GPS VE UWB VERİLERİNİN KESİNTİSİZ ENTEGRASYONU

Bu tez, şantiye güvenliği için tasarlanmış hibrit bir gerçek zamanlı konum izleme sistemi önermektedir. Sistem, Global Konumlama Sistemi (GPS) ile Ultra Geniş Bant (UWB) verilerini Korunsuz Kalman Filtresi (Unscented Kalman Filter, UKF) kullanarak birleştirmekte; birleştirilen konumlar Autodesk Revit API aracılığıyla Revit ortamına dinamik olarak aktarılmakta ve pyRevit üzerindeki özel Python betikleriyle görselleştirilmektedir. DeneySEL kurulum, iki üniversite binası boyunca hem kapalı hem açık alan testlerini içermektedir. Kapalı mekanda UWB ankrajları tetrahedral (dört yüzlü) konfigürasyonlarda yerleştirilmiş; denekler işçi hareketini simüle etmek için baretlerine sensör entegre edilmiştir. Veriler ard-ışleme tabi tutulmuş, zaman açısından senkronize edilmiş ve UKF tabanlı füzyondan önce 3 boyutlu aykırı değerlerin elenmesi amacıyla Mahalanobis uzaklığıyla filtrelenmiştir. Dönüştürülen koordinatlar Revit'te görselleştirilmiştir. Bununla birlikte, deneySEL sonuçlar iOS tabanlı GPS verilerinin doğruluğu ve kararlılığında kritik sınırlamalar bulunduğunu ortaya koymuştur. Bulgular, iOS akıllı telefonlarının inşaat bağlamında yüksek hassasiyetli konumlama için uygun olmadığını göstermektedir. Çalışma, açık alan verilerinin doğruluğunu artırmak için GNSS alıcılarının (çift frekanslı RTK sistemleri) benimsenmesini veya atalet sensörleriyle entegrasyonu önermektedir. Genel olarak, araştırma, önerilen GPS-UWB-UKF füzyon sisteminin Revit entegrasyonu ile birlikte konumsal sürekliliği ve mekansal farkındalığı anlamlı ölçüde iyileştirdiğini göstermektedir. Çalışma, algılayıcı füzyonu, mekansal filtreleme ve Revit entegrasyonuna dayalı gerçek zamanlı işçi güvenliği yönetimi için genişletilebilir bir metodoloji sunarak inşaat yönetimine katkı sağlamakta; aynı zamanda sensör sistemlerdeki pratik donanım ve kalibrasyon sınırlamalarını da açık biçimde ortaya koymaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	xi
LIST OF TABLES	xx
LIST OF ACRONYMS/ABBREVIATIONS	xxii
1. INTRODUCTION	1
1.1. General Information and Background	1
1.2. Motivation and Research Problem	3
1.3. Objectives of The Study	5
1.4. Scope and Limitations	6
1.5. Originality of the Master Thesis	7
1.6. Outline of the Master Thesis	8
2. LITERATURE REVIEW	10
2.1. Real Time Location Systems	11
2.1.1. Definition and Scope of RTLS	13
2.1.2. Indoor Core Technology	14
2.1.3. Outdoor Core Technology	15
2.1.4. Comparison Core Technologies	17
2.1.5. Challenges in RTLS Deployment: Indoor vs Outdoor	24
2.1.6. Towards Proactive Monitoring: The Role of RTLS in Dynamic Environments	25
2.2. Real Time Location Systems in Construction Sector	26
2.2.1. Construction Worker Tracking	28
2.2.2. Integration into Site Operations, Management Systems and Health and Safety Executions	31
2.2.3. Integration into Digital Twin	34
2.2.4. Examples of RTLS in Construction	34

2.2.5. Hybrid Use of RTLS	35
2.3. Research Gap in the Literature	36
2.3.1. Lack of Hybrid Systems in the Construction Industry	37
2.3.2. Need for Field Deployable, Real-Time and Proactive Safety Frame- works	39
2.3.3. Positioning of this Study within the Identified Research Gap . .	41
3. METHODOLOGY	46
3.1. Hypothesis	47
3.2. Research Design and Framework Overview	49
3.2.1. Sensor Configuration	52
3.2.2. Data Acquisition	53
3.2.3. Outlier Rejection via Mahalanobis Distance	54
3.2.4. Data Fusion Algorithm: Unscented Kalman Filter	58
3.3. System Architecture	68
3.3.1. Sensing Layer: UWB and GPS Module Deployment	69
3.3.2. Communication Layer: Data Transmission and Logging	70
3.3.3. Processing Layer: Kalman Filter and Coordinate Alignment . .	72
3.3.4. Application Layer: Revit Integration and Visualization	72
3.4. Hardware and Sensor Setup	73
3.4.1. Selection of Anchors and Tags	74
3.4.2. Placement and Calibration of Anchors	74
3.4.3. GPS Data Acquisition and Smartphone Integration	77
3.4.4. Anchor-Tag Communication and Frequencies	78
3.5. Sensor Data Collection and Handling	78
3.5.1. GPS Data Logging and Pre-Processing	79
3.5.2. Time Synchronization/Frequency and Data Integrity	80
3.6. Coordinate System Transformations	81
3.6.1. Geodetic GPS to ECEF (Earth-Centered Earth-Fixed) Conversion	82
3.6.2. ECEF to ENU (East-North-Up)	84
3.6.3. UWB Local Coordinate System to ENU	85
3.6.4. ENU to Revit Coordinate Reference Plane and Mapping	85

3.6.5.	UWB Reference Frame and Revit Referential Frame	86
3.7.	Mahalonbis Distance-Based Elimination	88
3.8.	Unscented Kalman Filter-Based Sensor Fusion	90
3.8.1.	State Vector Design and Motion Model	92
3.8.2.	Predicitons and Update Equations	93
3.8.3.	Noise Covariance Estimation for GPS and UWB	95
3.9.	BIM Environment Integration Using Revit API and pyRevit	95
3.9.1.	3D Revit Model Detail	96
3.9.2.	Revit API: Data Interfacing and Scripting Environment	100
3.9.2.1.	Integration with External Data: Workflow	101
3.9.3.	Python Script Development in pyRevit	104
3.9.4.	Worker Representation in Revit	106
3.9.5.	Use of Visualization Engines (Enscape)	108
4.	EXPERIMENTAL SETUP AND VALIDATION	110
4.1.	GNSS Coordinate Measurement	111
4.2.	Testing Environment Design: Indoor/Outdoor Experimental Area	117
4.3.	Hardware Components	127
4.3.1.	Ground Truth Measurement Depending on Visual Check	130
4.3.2.	Accuracy and Evaluation	130
5.	RESULTS	132
5.1.	Experimental Results: Accuracy and Continuity	132
5.1.1.	3D Mahalanobis-Based Outlier Detection in the UWB Network	134
5.1.1.1.	Findings from the Engineering Faculty Dataset	135
5.1.1.2.	Findings from the Data Science, Artificial Intelligence Dataset	144
5.1.1.3.	Findings from The Data Global Positioning Dataset	154
5.1.2.	Mahalanobis Outlier Removal	157
5.1.3.	Unscented Kalman Filter GPS and UWB Fusion	159
5.1.4.	Revit Mapping and Enscape Visualization	166
6.	DISSCUSSION	169
6.1.	GPS Measurements	169

6.2. UWB Measurements	174
6.2.1. UWB Timestamped Data Comparison	175
6.3. GPS – UWB Data Fusion Issues	181
6.3.1. Spatial Accuracy and Transition Behavior in GPS-UWB Fusion	181
6.4. Validation	185
6.5. System Performance and Visualization Efficacy	191
6.5.1. Data Collection Issues	191
6.5.2. Revit API Integration	192
6.6. Survey for Applicability of the Framework	201
6.7. Limitations and Areas for Improvement	206
7. CONCLUSION AND FUTURE WORK	209
7.1. Summary of Contributions	209
7.2. Review of Objectives and Findings	210
7.3. Practical Implications for Construction Industry	211
7.4. Recommendations for Future Research	212
7.5. Closing Remarks	213
REFERENCES	215
APPENDIX A: MAHALONOBIS RESULT OF UWB DATASET IN THE EN-	
GINEERING FACULTY	229
APPENDIX B: MAHALONOBIS OUTLIER DETECTION RESULT OF	
UWB DATASET IN THE DATA SCIENCE AND THE ARFIC-	
IAL INTELLIGENCE INSTITUTE BUILDING	231
APPENDIX C: MAHALONOBIS RESULT OF GPS SENSOR LOGGER DA-	
TA	234
APPENDIX D: UWB SENSOR LOCATION DATA COORDINATE TRANS-	
FORMATION	239
APPENDIX E: PYREVIT PYTHON CODINGS	242
APPENDIX F: UNSCENTED KALMAN FILTER FUSION OF GPS AND	
UWB POSITION DATA	267
APPENDIX G: MAHALONBIS CODINGS	277
APPENDIX H: UNSCENTED KALMAN FILTER CODINGS	280

APPENDIX I: CAD DRAWINGS 283



LIST OF FIGURES

Figure 3.1.	End-to-end pipeline: data collection, coordinate transformation, BIM model development, Revit API, and real-time monitoring. . .	47
Figure 3.2.	Framework model: Integration of GPS and UWB position data in revit.	51
Figure 3.3.	Revit integration RTLS tracking system workflow.	52
Figure 3.4.	UWB and GPS data filtering and fusion.	58
Figure 3.5.	UWB tag and IOS smartphone implementation to hardhat.	78
Figure 3.6.	iOS smartphone UWB tag attachment.	79
Figure 3.7.	Data science and artificial intelligence institute building from front.	98
Figure 3.8.	Data science and artificial intelligence institute building Revit model from front.	98
Figure 3.9.	Data science and artificial intelligence institute building Revit model from back.	98
Figure 3.10.	Engineering building faculty.	99
Figure 3.11.	Revit model engineering faculty and data science and artificial intelligence institute 1.	99

Figure 3.12. Revit model engineering faculty and data science and artificial intelligence institute 2.	99
Figure 3.13. Hierarchical structure of the pyRevit extension for component placement based on location data.	106
Figure 3.14. Adaptive component represents the worker avatar.	107
Figure 3.15. Enscape interface BIM information coordination.	109
Figure 4.1. Referential point labelling.	113
Figure 4.2. Referential point PB antenna settlement.	114
Figure 4.3. Referential point GNSS measurement.	114
Figure 4.4. Referential point PB.	114
Figure 4.5. GNSS receiver.	115
Figure 4.6. GNSS station selection.	115
Figure 4.7. 3D model case for UWB anchors.	118
Figure 4.8. Engineering faculty testing indoor environment.	119
Figure 4.9. Engineering faculty testing indoor in virtual environment.	119
Figure 4.10. Engineering faculty testing indoor in virtual environment (from indoor).	119

Figure 4.11. Engineering data science institute testing environment.	120
Figure 4.12. Engineering data science institute testing environment a1.	120
Figure 4.13. Engineering data science institute anchor attachment testing environment.	121
Figure 4.14. Engineering data science institute testing anchors (sequentially A6-A5).	121
Figure 4.15. Engineering data science institute testing anchors A8.	121
Figure 4.16. Engineering data science institute testing anchor positions A7. . .	122
Figure 4.17. Engineering data science institute testing anchor positions A6. . .	122
Figure 4.18. Engineering data science institute testing anchor positions A5. . .	123
Figure 4.19. Planar (2D) layout of the UWB anchor positions in the test environment (top-down view).	124
Figure 4.20. UWB anchor placement in 2D plan view. Anchors A1-A4 (blue) cover Indoor-1, while A5-A8 (red) are in Indoor-2. A8 is the local origin (0,0). The two buildings are about 46 m apart in X and 33 m in Y (south) direction.	126
Figure 4.21. Green polylines indicate predefined validation paths used for the accuracy check.	127
Figure 4.22. GPS data logged from the IOS-based sensor logger.	129

Figure 5.1.	Subject 1 3D mahalanobis outliers scatter plot.	137
Figure 5.2.	Subject 1 stepwise mahalanobis outliers.	137
Figure 5.3.	Subject 1 final filtered data on XY plane.	138
Figure 5.4.	Subject 2 3D scatter plot mahalanobis with outliers and location data.	139
Figure 5.5.	Mahalanobis outliers scatter plot.	139
Figure 5.6.	Subject 2 2D XY plane mahalanobis outliers scatter plot.	140
Figure 5.7.	Subject 2 final filtered data on XY plane.	140
Figure 5.8.	Subject 3 XY plane with outliers and location data.	141
Figure 5.9.	Subject 3 3D scatter plot mahalanobis with outliers and location data.	142
Figure 5.10.	Subject 3 stepwise mahalanobis outlier.	142
Figure 5.11.	Subject 4 Stepwise mahalanobis outliers.	143
Figure 5.12.	Subject 4 3D scatter plot mahalanobis with outliers and location data.	144
Figure 5.13.	Subject 4 stepwise mahalanobis method outliers and location data.	144
Figure 5.14.	Subject 1 XY plane with outliers and location data.	145

Figure 5.15. Subject 1 3D scatter plot mahalanobis with outliers and location data.	146
Figure 5.16. Subject 1 stepwise mahalanobis outlier.	146
Figure 5.17. Subject 1 final filtered data on XY plane.	147
Figure 5.18. Subject 2 XY plane with outliers and location data.	148
Figure 5.19. Subject 2 3D scatter plot mahalanobis with outliers and location data.	148
Figure 5.20. Subject 2 stepwise mahalanobis outlier.	149
Figure 5.21. Subject 2 final filtered data on XY plane.	149
Figure 5.22. Subject 3 3D scatter plot mahalanobis with outliers and location data (data science institute).	150
Figure 5.23. Subject 3 stepwise mahalanobis outlier.	151
Figure 5.24. Subject 3 XY plane with outliers and location data.	151
Figure 5.25. Subject 3 final filtered data on XY plane.	151
Figure 5.26. Subject 4 3D scatter plot mahalanobis with outliers and location data.	152
Figure 5.27. Subject 4 XY plane with outliers and location data.	153
Figure 5.28. Subject 4 stepwise mahalanobis outlier.	153

Figure 5.29. Subject 4 final filtered data on XY plane.	153
Figure 5.30. GPS data of movement path XY plane Outliers.	154
Figure 5.31. GPS data XY plane Outliers.	155
Figure 5.32. GPS data Outliers 3D UTM Coordinates.	155
Figure 5.33. GPS data Outlier Filtering 3D UTM Coordinates.	156
Figure 5.34. Subject 4 stepwise mahalanobis outlier.	156
Figure 5.35. Mahalanobis outlier removed UWB data (Subject 1).	157
Figure 5.36. Mahalanobis outlier removed UWB data (Subject 1).	157
Figure 5.37. Mahalanobis outlier removed UWB data (Subject 2).	158
Figure 5.38. UWB data integration (Subject 3).	158
Figure 5.39. Mahalanobis outlier removed UWB data integration (Subject 3). . .	159
Figure 5.40. Subject 1 Kalman filter data fusion result.	162
Figure 5.41. Subject 1 Kalman filter data fusion result on spatial layout.	162
Figure 5.42. Subject 2 Kalman filter data fusion result.	163
Figure 5.43. Subject 2 Kalman filter data fusion result on spatial layout.	163
Figure 5.44. Subject 3 Kalman filter data fusion result.	164

Figure 5.45.	Subject 3 Kalman filter data fusion result on spatial layout.	164
Figure 5.46.	Subject 4 Kalman filter data fusion result.	165
Figure 5.47.	Subject 4 Kalman filter data fusion result on spatial layout.	165
Figure 5.48.	Worker avatar Enscape demonstration outdoor 1.	167
Figure 5.49.	Worker avatar Enscape demonstration outdoor 2.	167
Figure 5.50.	Worker avatar Enscape demonstration indoor.	167
Figure 5.51.	Revit experiment pathway.	168
Figure 5.52.	Revit mapping site layout.	168
Figure 6.1.	GPS trajectory data logged using an iOS-based sensor app, overlaid on a satellite basemap near the experiment site entrance.	170
Figure 6.2.	Subject 1 Faculty of engineering dataset.	175
Figure 6.3.	Subject 1 engineering faculty building all the dataset.	175
Figure 6.4.	Subject 2 engineering faculty building all the dataset.	176
Figure 6.5.	Subject 2 engineering faculty building all the dataset.	176
Figure 6.6.	Subject 3 engineering faculty building all the dataset.	177
Figure 6.7.	Subject 3 artificial intelligence institute building all the dataset. . .	178

Figure 6.8.	Subject 4 engineering faculty building all the dataset.	179
Figure 6.9.	Subject 4 artificial intelligence institute building all the dataset. . .	179
Figure 6.10.	Subject 1 comparison with enscape.	186
Figure 6.11.	Subject 1 comparison with enscape 1.	187
Figure 6.12.	Subject 2 comparison with enscape.	188
Figure 6.13.	Subject 3 comparison with enscape.	189
Figure 6.14.	Subject 4 timestamped video record.	190
Figure 6.15.	Applicability of framework.	201
Figure 6.16.	Hardware adequacy.	202
Figure 6.17.	Active Revit usage in firm level.	203
Figure 6.18.	Primary barriers of the framework.	204
Figure 6.19.	Main cost components of the framework.	204
Figure 6.20.	Operational value.	205
Figure 6.21.	Safety benefit of framework.	206
Figure I.1.	Faculty of engineering basement floor plan.	284

Figure I.3. Faculty of engineering first floor plan. 286

Figure I.4. Faculty of engineering second floor plan. 287



LIST OF TABLES

Table 2.1.	RTLS technology comparison based on literature.	23
Table 3.1.	Summary of key concerns in GPS-UWB fusion necessitating mahalanobis use.	57
Table 3.2.	Comparison for Kalman filter and advanced Kalman filters.	64
Table 3.3.	Justification for unscented Kalman filter.	66
Table 3.4.	Confidence level determination effect on outlier rejection.	89
Table 4.1.	Observation and processing summary.	116
Table 4.2.	Estimated positions (ITRF20, Epoch 2025.4).	116
Table 4.3.	UTM coordinates zone 35N.	117
Table 4.4.	Anchor points measured coordinate values.	125
Table 5.1.	Meaning of outlier values.	134
Table 6.1.	Time intervals related to location subject 1.	175
Table 6.2.	Time intervals related to location subject 2.	176
Table 6.3.	Time intervals related to location subject 3.	177
Table 6.4.	Time intervals related to location subject 4.	178

Table 6.5.	Comparative summary of studies (2020–2025).	199
Table A.1.	Subject 1 example data sources.	229
Table B.1.	Data sources.	231
Table C.1.	Data sources.	234
Table F.1.	Fused Data sources.	267

LIST OF SYMBOLS

a	Semi-major axis of reference ellipsoid (WGS84)
$DM(\Delta x_t)$	Mahalanobis distance
$DM^2(\Delta x_t)$	Squared Mahalanobis distance
e	First eccentricity of the ellipsoid
E, N, U	East, North, Up axes in local tangent frame
$f(\cdot)$	Process function
$h(\cdot)$	Measurement function
H	Observation matrix
I	Identity matrix
K_k	Kalman gain matrix
P_k	State error covariance matrix at time k
$P_{k k-1}$	Predicted error covariance before update
$P_{k k}$	Updated error covariance after measurement update
P_{xz}	Cross-covariance between predicted state & measurement
Q, Q_k	Process noise covariance (or at timestep k)
R, R_k	Measurement noise covariance
S_k	Predicted measurement covariance matrix
u_k	Control input vector
v_k	Measurement noise vector
w_k	Process noise vectors
X, Y, Z	ECEF Cartesian coordinates
$X_{k k-1}$	Predicted state estimate before update
$X_{k k}$	Updated state estimate after measurement update
$\hat{x}_k$	State vector (position & velocity at time k)

(x_k, y_k)	Worker position coordinates in XY plane at timestep k
z_k	Measurement vector at time k (GPS/UWB positions)
$\hat{z}_k$	Predicted measurement (mean of sigma point)
$X_{a,df}^2$	Chi-square threshold for outlier decision
Δt	Sampling interval / time step
Δx_t	Step vector between consecutive timestamps
ϕ, λ, h	Geodetic latitude, longitude, ellipsoidal height
μ	Mean vector
Σ	Covariance matrix of step vectors

LIST OF ACRONYMS/ABBREVIATIONS

AEC	Architecture, Engineering and Construction
AI	Artificial Intelligence
API	Application Programming Interface
BIM	Building Information Modelling
BLE	Bluetooth Low Energy
CAD	Computer Aided Design
DWM1001	Decawave Qorvo UWB Module
ECEF	Earth-Centered, Earth Fixed
EKF	Extended Kalman Filter
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
ITRF20	International Terrestrial Reference Frame 2020
KF	Kalman Filter
LOD	Level of Detail
LOS	Line-of-Sight
NLOS	Non-Line-of-Sight
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
RTK	Real Time Kinematic
RTLS	Real Time Locating System
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TWR	Two Way Ranging
UWB	Ultra-Wideband
UKF	Unscented Kalman Filter
WGS84	World Geodetic System 1984

1. INTRODUCTION

Construction sites are dynamic, cluttered, and hazardous. Layouts shift daily, heavy equipment operates in tight spaces, and crews move between outdoor yards and interior rooms conditions that make manual, after-the-fact safety checks inadequate. Real-Time Locating Systems (RTLS) promise proactive oversight, but no single modality works everywhere: GPS provides broad outdoor coverage yet degrades around buildings and trees and fails indoors; Ultra-Wideband (UWB) delivers decimeter-level accuracy indoors under line-of-sight (LOS) but suffers under non-line-of-sight (NLOS), body shadowing, and anchor-layout constraints. What site teams need is seamless indoor-outdoor localization that feeds directly into the BIM model where coordination and safety decisions are made.

This thesis addresses that need by designing a hybrid GPS-UWB sensor-fusion framework and integrating the fused trajectory into Autodesk Revit via the Revit API (pyRevit), turning a static BIM model into a live, safety-aware digital twin. Concretely, timestamped GPS and UWB measurements are synchronized, cleaned using Mahalanobis-distance outlier removal, fused with an Unscented Kalman Filter (UKF), and transformed into the project frame through geodetic conversions and Helmert alignment. The system is validated across indoor rooms and outdoor approaches of two adjacent university buildings, using fixed UWB anchors and smartphone-grade GPS.

1.1. General Information and Background

The construction industry is one of the fundamental industry that has largest amount of workers (Taghinezhad *et al.*, 2021). Construction sites are dynamic, risky and hazardous workplaces due to the wide range of materials used and the presence of heavy and sometimes hazardous equipment, where situational awareness is crucial for preventing accidents (Forteza *et al.*, 2020; Muñoz-La Rivera *et al.*, 2021). The Occupa-

tional Safety and Health Administration (OSHA, 2016) reported that nearly 20.5% of all deaths in the U.S. private sector occurred within the construction industrial activities. Each year, construction-related incidents result in the loss of approximately 150 lives (Ahmed, 2019a). Despite growing awareness, current safety monitoring methods still rely heavily on manual checks and human reporting, making them time-consuming and prone to error (Hossain and Ahmed, 2022). As a result, unsafe conditions are often not recognized until an accident or near miss has already occurred, largely due to fragmented and inconsistent site observations (Bugalia and Kedia, 2022). This highlights the urgent need for more proactive, accurate, and real-time approaches to construction safety management.

Proactive systems are increasingly studied to prevent accidents rather than reacting to them (Ventura *et al.*, 2023). To address the major limitations of manual safety monitoring, researchers have introduced various sensing technologies aimed at improving site oversight. These technologies are designed to support automated, continuous, and precise monitoring of construction environments by transmitting real-time contextual data (Park *et al.*, 2017).

The adoption of digital twin technology is steadily expanding within the construction industry, with safety and risk management emerging as critical domains where its application can yield considerable advantages (Omran *et al.*, 2023). A digital twin for construction safety uses real-time sensor data and BIM to prevent hazards and guide responses (Teizer *et al.*, 2022). Recent advancements in Building Information Modeling (BIM) and sensing technologies enable the creation of digital twins—virtual replicas of construction sites that reflect real-time data (Tuhaise *et al.*, 2023). Integrating real-time location data of workers into a BIM model, project managers can monitor site safety proactively. Traditional safety practices in construction are often insufficient for fast-changing conditions because of the dependency on reactive measures such as incident reports, safety audits, and post-accident analysis, which provide only insights into the causes of safety lapses (Pereira *et al.*, 2025).

Modern digital twin construction safety frameworks integrates Real-Time Location Systems (RTLS) to track construction, construction workers and the resources, detect unsafe conditions on site (Teizer *et al.*, 2022). Real-Time Location Systems (RTLS) using sensors like GPS, Bluetooth Low Energy(BLE) and Ultra-Wideband offer a way to continuously track workers and assets on site (Pereira *et al.*, 2025). The effectiveness of Global Positioning System (GPS) technology is limited by delays in data transmission, reduced accuracy in densely populated areas, and signal loss when used indoors (Moselhi *et al.*, 2020). Park *et al.* (2017) highlight that implementing sensing technologies within indoor environments presents greater challenges compared to outdoor settings. This research focuses on combining GPS for outdoor positioning and UWB for indoor positioning within a BIM based digital twin to achieve seamless, real-time tracking of multiple construction workers. A custom integration using the Autodesk Revit API (via pyRevit add in utilizes Python scripts) is proposed to visualize worker positions inside the BIM model and issue safety alerts.

1.2. Motivation and Research Problem

Safety is an essential issue in the construction industry, where dynamic environments and evolving spatial configurations contribute to elevated risk levels. Falls from height, struck by incidents, unauthorized entries into hazardous zones continue to be leading causes of fatalities and injuries on construction sites (Pereira *et al.*, 2025; Setayeshgar *et al.*, 2013). Traditional methods of safety monitoring such as manual inspections or post incident reporting often fail to detect risks in real time, observing them inadequate in preventing accidents before they occur.

The integration of real-time indoor and outdoor worker positioning technologies in Revit environment has the potential to revolutionize construction safety management by enabling continuous hazard awareness. Specifically, this research proposes hybrid visualization system of positions that combines Global Positioning System (GPS) for outdoor environments with Ultra-Wideband (UWB) for high precision indoor tracking.

The fusion of these technologies using Kalman Filtering ensures robust and seamless positioning across complex construction sites, mitigating the blind spots in monitoring.

By integrating the live location data within a dynamic BIM environment, the static nature of traditional digital models is transformed into a cyber-physical twin a real time digital replica of the construction site (Zhao *et al.*, 2019). Evolution from static documentation to dynamic decision support allows safety managers to visualize worker movements, identify proximity to hazards, and enforce geofencing rules (Tuhaise *et al.*, 2023). In effect, the BIM model transitions from a planning tool into an operational safety interface.

Furthermore, the Revit API and pyRevit scripting environment enable the direct integration of filtered position data into the 3D model. Workers can be represented as moving components within the BIM interface, allowing for live visualizations, real-time alerts, and automatic safety warnings when workers enter predefined risk zones. This integration facilitating faster, data-informed decision making.

Despite the availability of individual technologies GPS, UWB, BIM there is a significant gap in their holistic integration for safety applications especially in real-time usage (Omran *et al.*, 2023; Tuhaise *et al.*, 2023). Most of the existing studies either track workers only in predefined indoor and outdoor zones, or visualize data externally via dashboards rather than within the visual platform which enables process tracking such as BIM model. These limitations restrict situational awareness and fail to support real-time safety interventions. This study addresses this gap by developing a novel end to end framework for real-time construction worker localization and safety visualization. The proposed system fuses UWB and GPS data through Kalman Filter, transforms coordinates into the BIM environment reference frame and visualize worker locations dynamically within Revit. Such integration offers continuous monitoring worker safety conditions across varying site conditions.

1.3. Objectives of The Study

Ultimately, this research aims to advance proactive safety monitoring by transforming conventional static models into live digital twins that support continuous tracking, and enabling decision making by providing spatially observations. The system is designed not only to enhance hazard detection capabilities but also to shift construction safety practices from reactive response paradigms toward predictive and preventive monitoring frameworks thus significantly contributing to the reduction of on-site incidents.

The primary objective of this study is to contribute to the advancement of proactive safety monitoring systems in the construction industry by transforming traditionally static BIM models into dynamic, data-driven digital twins capable of real-time decision support. The research proposes a hybrid RTLS framework that integrates Global Positioning System (GPS) for outdoor environments and Ultra-Wideband (UWB) for indoor localization. These are fused using a Kalman filtering approach, filtered with Mahalanobis distance-based outlier rejection, and visualized within a Revit BIM model using custom Python scripts via pyRevit.

The specific objectives of this study are as follows:

- (i) To design and develop a hybrid location tracking system that ensures seamless indoor-outdoor worker localization by combining GPS and UWB technologies, addressing signal degradation challenges in transition zones between environments.
- (ii) To implement a real-time data integration framework system between RTLS systems and BIM through the Autodesk Revit API, enabling the accurate visualization of live worker locations within a 3D spatial model of the construction site.
- (iii) To provide situational awareness and hazard detection system by implementing safety zones (geofences) directly into the BIM model, and automatically triggering alerts when workers enter predefined hazardous areas.

- (iv) To fuse multi-source location data using Kalman filtering, thereby improving the accuracy and robustness of the estimated positions and enabling smooth transitions across the tracking zones.
- (v) To apply Mahalanobis distance-based outlier detection for removing erroneous data points prior to fusion, especially under NLOS conditions and environmental interferences common in construction sites.
- (vi) To evaluate the system's performance in a post-processed experimental setup, validating the proposed framework using real data collected via Raspberry Pi-connected UWB modules and GPS receivers.
- (vii) To contribute a reusable methodology for integrating real-time spatial data into digital twins, thereby enabling construction managers and safety personnel to shift from reactive safety practices which rely on incident reports and retrospective analysis toward predictive and preventive frameworks based on live monitoring and data-driven insights.

Through these objectives, the research seeks to address the limitations of existing safety monitoring systems such as delayed recognition of hazards, static data flows, and static risk assessment models by enabling continuous, location-aware, and automated safety oversight. The proposed framework in the master thesis represents an enhancement toward the Safety 4.0 paradigm, where intelligent technologies support safer, more responsive, and more resilient construction site operations.

1.4. Scope and Limitations

Real-Time Location Systems (RTLS) have emerged as powerful enablers for real-time tracking of workers, assets, and equipment on construction sites. Technologies such as GPS, Ultra-Wideband (UWB), Bluetooth Low Energy (BLE), and RFID have been widely studied for their ability to provide location data (Li *et al.*, 2016; Arabshahi *et al.*, 2021). While GPS offers extensive outdoor coverage, it becomes unreliable or entirely nonfunctional in enclosed or obstructed environments due to signal degradation and multipath effects (Zandbergen, 2009; Groves *et al.*, 2012). Conversely,

UWB technology offers superior precision in complex indoor spaces and cluttered job sites but requires a fixed infrastructure and suffers from line-of-sight (LOS) constraints (Shamsollahi *et al.*, 2024; Alarifi *et al.*, 2016).

Most current systems focus on a single technology and are thus limited in their ability to provide continuous, seamless tracking across both indoor and outdoor zones, which are common in construction work flows. Worker transitions between these environments often lead to “blind spots” in monitoring, undermining the very purpose of proactive safety tracking. This limitation arises primarily because technologies like GPS, while effective outdoors, lose accuracy or fail entirely indoors due to signal attenuation and obstruction by structures. Conversely, indoor-focused technologies such as Ultra-Wideband (UWB) or Bluetooth Low Energy (BLE) offer high precision within buildings but lack the range or reliability needed for outdoor environments. As a result, when workers move between these zones, their real-time location data can become fragmented or lost, creating critical gaps in safety monitoring.

1.5. Originality of the Master Thesis

This master thesis presents an original contribution to the field of construction safety management by developing an integrated framework that fuses GPS and Ultra-Wideband (UWB) real-time location data into Revit which is Building Information Modeling (BIM) environment using the Revit API and Python scripting. Unlike most existing studies that either treat GPS and UWB data in isolation or visualize worker tracking through external dashboards, this research embeds real-time location data directly into the Revit model. This enables dynamic, spatially-aware safety visualization within a cyber-physical Digital Twin (DT).

The originality of the thesis is summarized:

- **Hybrid RTLS-Revit Integration:** While GPS and UWB have each been employed separately in the literature, their seamless integration into a unified BIM-based

DT system for real-time worker tracking and safety alerting remains underexplored. This research bridges this gap with a Kalman-filter-based fusion system supported by Mahalanobis distance-based outlier rejection.

- **Direct Visualization in Revit Environment:** Most systems utilize third-party interfaces or dashboards for worker monitoring. This study uniquely provides the filtered location data into the Revit 3D model using the pyRevit API, allowing stakeholders to observe worker movements within the same model used for planning and coordination.
- **Post-processed Experimental Validation with Real Hardware:** Although the system is intended for real-time use, this research evaluates its feasibility using post-processed data from actual UWB sensors and GPS receivers, establishing a reliable prototype for future real-time implementation.
- **Application of Cognitive Digital Twin Principles:** The system lays foundational work for evolving static BIM models into cognitive/proactive safety twins by integrating real-time behavior monitoring, risk visualization, and proactive intervention mechanisms.

This thesis, therefore, contributes both a methodological and technological advancement by unifying real-time spatial awareness, data filtering, and Revit-integrated visualization to enhance proactive safety management in dynamic construction environments.

1.6. Outline of the Master Thesis

The structure of the thesis is organized to systematically guide the reader through the research context, theoretical grounding, implementation, and evaluation of the proposed research:

- **Chapter 1 - Introduction** Provides an overview of the construction safety problem, motivation, research objectives, and scope. It highlights the need for hybrid RTLS integration with Revit which is BIM software for proactive safety monitoring and

outlines the significance of GPS-UWB fusion within a digital twin context.

- Chapter 2 - Literature Review Presents a comprehensive review of safety management evolution from reactive to proactive paradigms, the role of digital twins, and the use of Real-Time Location Systems (RTLS) such as GPS and UWB in construction. It also reviews recent studies on Revit model integration, sensor fusion, and cognitive digital twin applications for safety.
- Chapter 3 - Methodology Describes the overall system architecture, hardware setup, and software implementation for GPS-UWB fusion, including Unscented Kalman Filtering and Mahalanobis-based outlier rejection. It details the coordinate transformation approach, geofencing logic, and the data visualization pipeline within Revit via pyRevit.
- Chapter 4 - Experimental Setup and Data Collection Details the experimental environment, including Raspberry Pi, Decawave UWB anchors, GPS receivers, and the data collection protocol. It explains the data logging, synchronization, and pre-processing steps used to prepare datasets for fusion and visualization.
- Chapter 5 - Results and Performance Analysis Presents the results of the data fusion, accuracy improvements after outlier filtering, and Revit API integration outcomes. This chapter also benchmarks system performance, discussing GPU bottlenecks, temporal downsampling strategies, and the trade-off between data fidelity and responsiveness.
- Chapter 6 - Discussion Interprets the findings in the context of real-world deployment, compares results with similar RTLS-BIM studies, and discusses system limitations and potential optimizations, including cloud-based visualization and lighter 3D engines.
- Chapter 7 - Conclusion and Future Work Summarizes the contributions of the thesis, highlights practical and theoretical implications, and suggests future extensions, such as real-time field deployment, worker fatigue and health monitoring, and AI-driven predictive safety models.

2. LITERATURE REVIEW

In this Literature Review, the thesis surveys Real-Time Location Systems (RTLS) used in construction RFID, BLE, Wi-Fi FTM/RTT, UWB, GNSS/GPS, IMU, and vision highlighting their operating principles and trade-offs in accuracy, cost, and deployment. It contrasts indoor and outdoor challenges (e.g., NLOS, multipath, urban canyons) and explains why no single technology ensures continuous, precise tracking across mixed site conditions. The section synthesizes evidence favoring hybrid solutions: GPS as the outdoor backbone; UWB for precise indoor ranging; IMU for continuity; and, where feasible, vision for contextual awareness. It then reviews how RTLS streams are integrated with Revit to visualize worker/equipment proximity. Finally, it identifies gaps in prior work, notably the scarcity of field-deployable, real-time, hybrid RTLS pipelines tightly integrated with industry-standard BIM tool Revit and engineered for scalable visualization and data quality control. This sets up the thesis's contribution: a GPS-UWB fusion framework with robust filtering and direct BIM integration to support Safety 4.0-oriented, real-time decision making.

Real-Time Location Systems (RTLS) are increasingly adopted in construction to improve situational awareness and reduce safety incidents. A wide spectrum of technologies has been explored, including Global Navigation Satellite Systems (GNSS/GPS), Ultra-Wideband (UWB), Bluetooth Low Energy (BLE), Wi-Fi fine-time-measurement/round-trip-time (FTM/RTT), Radio Frequency Identification (RFID), inertial measurement units (IMU), and, in some cases, vision-based methods. Each option offers distinct trade-offs in accuracy, range, cost, and infrastructure needs. In general, GPS provides broad coverage outdoors but degrades around buildings, trees, and heavy equipment; UWB can deliver sub-meter accuracy indoors under line-of-sight (LOS) conditions; BLE and Wi-Fi are lower-cost but typically meter-level; RFID is well-suited to zone-based presence rather than precise tracking; IMU supports continuity but drifts without external corrections; and vision demands clear sight lines and careful calibration.

Construction sites blend open spaces with complex indoor environments, creating discontinuities for any single positioning technology. Outdoor tracking relies primarily on GPS, which is prone to obstructions, multi path, and device-grade variability. Indoors, UWB has emerged as a strong candidate due to its resilience to interference and time-of-flight ranging; however, performance drops under Non-Line-of-Sight (NLOS) and in multi path rich rooms. Body shadowing, anchor geometry, and room size further influence achievable accuracy. The literature therefore converges on hybrid approaches that combine complementary sensors to maintain continuity across indoor and outdoor zones.

2.1. Real Time Location Systems

Real-Time Location Systems (RTLS) are integrated hardware-software frameworks designed to continuously determine and track the position of objects or individuals in real time. From a computer science perspective, RTLS can be understood as a cyber-physical system where wireless communication technologies, sensor data processing, network synchronization, and positioning algorithms operate in combination. Within the context of Industry 4.0 and the Internet of Things (IoT), RTLS has become a fundamental enabler of efficiency, safety, and automation.

RTLS is built upon a variety of technologies. Global Navigation Satellite Systems (GNSS), including GPS, provide meter-level accuracy in open outdoor environments but degrade significantly under signal obstruction. Ultra-Wideband (UWB) radio technology offers high accuracy at the centimeter level due to its wide bandwidth and short-pulse transmission, while also mitigating multipath interference. Bluetooth Low Energy (BLE) is widely adopted due to low cost and energy consumption, though it typically achieves only meter-scale accuracy. Radio-Frequency Identification (RFID) enables asset tracking through active or passive tags, whereas Wi-Fi-based positioning leverages access point density and typically provides accuracies between five and fifteen meters. Inertial Measurement Units (IMUs), consisting of accelerometers and gyroscopes, enable dead-reckoning of motion but suffer from cumulative drift. Vision-based systems,

employing computer vision and image recognition, deliver high-resolution positioning in structured indoor environments.

Several computational approaches are used to estimate location. Triangulation and trilateration methods compute coordinates based on distances or angles. Fingerprinting methods compare real-time radio signal patterns with pre-established maps, particularly effective with Wi-Fi or BLE. Proximity-based techniques infer position relative to the nearest reference station. Key measurement parameters include Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), Received Signal Strength Indicator (RSSI), and Channel State Information (CSI).

To overcome the limitations of individual sensors, hybrid systems are increasingly adopted. For example, UWB's high accuracy can be combined with IMU's continuity of motion tracking, producing more robust results. Similarly, computer vision can be integrated with radio-frequency signals to compensate for line-of-sight losses. Multi-sensor fusion is commonly achieved using algorithms such as the Extended Kalman Filter or Particle Filter, with sensor measurements weighted according to their reliability. A central requirement for these systems is precise time synchronization between network nodes, with UWB-based protocols often achieving nanosecond-level accuracy to reduce TDoA estimation errors.

RTLS architectures are typically structured in three layers. The sensing layer collects signals exchanged between tags and fixed reference points. The positioning engine processes this data through filtering and localization algorithms to generate coordinates. Finally, the application layer visualizes, stores, or analyses the location information. Real-time responsiveness demands low latency, high processing throughput, and energy efficiency across these layers.

Applications of RTLS are diverse. In construction, RTLS supports worker safety and logistics by enabling continuous monitoring of personnel and equipment. In logistics and warehousing, it facilitates inventory control and vehicle routing. In healthcare,

it improves workflow efficiency by tracking patients, staff, and medical devices. In manufacturing, RTLS is increasingly coupled with digital twins to provide continuous feedback on production processes. While the foundational principles remain constant, the choice of technologies, accuracy requirements, and environmental constraints vary substantially across application domains.

2.1.1. Definition and Scope of RTLS

Real-Time Location Systems (RTLS) are integrated hardware and software solutions that continuously detect, estimate and disseminate the positions of tagged assets, equipment, or persons within instrumented environments or open outdoor areas with minimal latency. In contrast to traditional, manual tracking methods, RTLS provide automated spatio-temporal streams that can be consumed by operational systems for monitoring, control, analytics and decision support (Li *et al.*, 2016).

Real-Time Location Systems (RTLS) are integrated technologies used to automatically identify and track the location of objects or people in real time, typically within a defined area. These systems employ various sensing modalities including radio frequency, optical, and inertial sensors to continuously determine positions of targets.

Real-Time Location Systems (RTLS) are a combination of hardware and software used to continuously determine and provide the real-time position of assets or resources equipped with special tags or devices. In essence, an RTLS enables tracking and monitoring of objects or people in real time, typically by using wireless communication technologies (e.g. RFID, Bluetooth, Ultra-Wideband, Wi-Fi, GPS) to gather location data.

The concept of RTLS has been applied across many industries, becoming increasingly feasible as wireless devices have become more affordable and portable. The scope of RTLS spans multiple domains such as logistics, healthcare, manufacturing, and con-

struction, where knowing “who or what is where” at any moment can enable efficient management and improve safety outcomes.

2.1.2. Indoor Core Technology

Indoor positioning is much harder than the outdoors due to obstacles, walls and reflections. Indoor positioning is complicated by non-line-of-sight (NLOS), multipath and attenuation. Radio signals indoors can take many paths (multipath) and may not reach directly (non-line of sight) causing large errors. Contemporary approaches include Received Signal Strength (RSS) fingerprinting (BLE/Wi-Fi), time-based ranging (UWB, Wi-Fi FTM/RTT), angle-of-arrival (AoA), and sensor fusion with inertial measurement units (IMUs). UWB time-of-flight is the current accuracy leader for industrial indoor RTLS (decimeter-level under good geometry), with resilience to multipath relative to narrowband methods; performance nonetheless degrades under severe NLOS unless outlier-robust filtering is applied (Alarifi *et al.*, 2016; Jin and Sadeghpour, 2022). BLE deployments remain attractive for cost and ubiquity (smartphone compatibility), but RSS-only methods often yield 3-5 m with variability; AoA under Bluetooth 5.1+ can approach sub-meter at the expense of antenna arrays and calibration (Faragher and Harle, 2015; Davidson and Pich’e, 2017; Zafari *et al.*, 2019). Wi-Fi positioning historically offered meter-level accuracies via RSS; IEEE 802.11mc/az Fine Timing Measurement (FTM/RTT) has shifted the state-of-practice towards submeter under favourable channel/bandwidth, though performance is chipset- and bandwidth-dependent and can be sensitive to security/implementation issues (Bullmann *et al.*, 2020; Hashem *et al.*, 2021; Orfanos *et al.*, 2023). IMUs bridge short outages and smooth jitter but drift without periodic absolute fixes; thus, modern indoor RTLS tends to adopt multi-sensor fusion (He and Chan, 2016; Zafari *et al.*, 2019). Advantages/drawbacks and NLOS mitigation (embedded): UWB offers accuracy/latency at higher infrastructure cost and requires anchor geometry optimisation (DOP-aware placement); BLE/Wi-Fi reduce cost but are RSS-sensitive and require denser beacons or fingerprint maintenance. NLOS mitigation spans anchor placement optimisation, residual-based outlier rejection, robust Kalman/particle filters, and opportunistic fu-

sion with IMU/vision (Guo *et al.*, 2022; Jin and Sadeghpour, 2022; Adame *et al.*, 2023; Pan *et al.*, 2022).

2.1.3. Outdoor Core Technology

Outdoor localization in construction environments is fundamentally shaped by the trade-off between global coverage and site-specific accuracy. Among the core technologies employed, Global Navigation Satellite Systems (GNSS), Ultra-Wideband (UWB), wireless radio technologies, Internet of Things (IoT) networks, Inertial Measurement Units (IMUs), and computer vision have all played significant roles. Each technology offers distinct advantages and drawbacks, while hybridization strategies increasingly dominate real-world implementations.

GNSS, encompassing GPS, GLONASS, Galileo, and BeiDou, constitutes the backbone of outdoor positioning. In unobstructed environments, standard code-phase GNSS receivers achieve accuracies of approximately 3-10 meters, while advanced Real-Time Kinematic (RTK) and Precise Point Positioning (PPP) techniques enable centimeter-level precision (Kaplan and Hegarty, 2017). These capabilities make GNSS indispensable for outdoor construction tasks such as surveying, machine guidance, and fleet monitoring (Teizer, 2015). However, the technology is significantly challenged by multipath interference and satellite signal loss in “urban canyon” conditions, beneath tall steel structures, or near partially covered zones, which are common in construction sites. Consequently, GNSS alone is insufficient for continuous outdoor worker or equipment tracking in complex built-environments (Li *et al.*, 2016).

UWB provides a complementary technology in outdoor zones where GNSS signals are obstructed. Using pulse-based transmissions across ultra-wide bandwidths (500 MHz), UWB delivers positional accuracies in the range of 10-30 centimeters, even under multipath and partial non-line-of-sight conditions (Alarifi *et al.*, 2016). This makes UWB suitable for outdoor construction areas near steel frames or reinforced concrete, where GNSS performance degrades (Jin and Sadeghpour, 2022). Nevertheless, outdoor

deployment of UWB requires careful installation of anchors across the site, and large open areas may demand a high density of infrastructure, which increases costs and reduces scalability.

Short-range wireless networks, such as Bluetooth Low Energy (BLE), Wi-Fi, and active RFID, are also leveraged for outdoor RTLS in construction. BLE beacons, which continuously broadcast identifiers at 2.4 GHz, provide localization via Received Signal Strength Indicator (RSSI) but generally achieve only 2-5 meter accuracy (Faragher and Harle, 2015). Wi-Fi-based systems can extend outdoor coverage across larger construction sites, yet their accuracy, often within a few meters, is hampered by interference and limited time-of-flight precision (Davidson and Pich'e, 2017). Active RFID tags, with ranges up to 100 meters, are applied for coarse asset and material tracking (Zafari *et al.*, 2019). While these technologies support outdoor monitoring in less critical tasks, their limited accuracy constrains their application in safety-critical or fine-grained localization scenarios. IoT Networks Low-power wide-area network (LPWAN) technologies, such as LoRaWAN and SigFox, are increasingly adopted in outdoor construction settings to provide long-range connectivity. LoRaWAN, in particular, supports communication over several kilometers with low power consumption and has been applied to monitor heavy equipment and crane components in large-scale construction projects (Rehman *et al.*, 2021). Although these systems provide wide-area connectivity and long battery lifetimes, their location resolution often tens of meters is insufficient for precision tracking. Therefore, IoT networks serve primarily as complementary layers in outdoor RTLS architectures, enabling data communication rather than high-accuracy positioning.

IMUs, comprising accelerometers and gyroscopes, are increasingly integrated into outdoor RTLS tags to provide continuity during GNSS signal loss. Through pedestrian dead reckoning and inertial navigation, IMUs deliver high-fidelity short-term tracking but are prone to cumulative drift over longer durations (Foxlin, 2005). In outdoor construction environments, IMUs are not deployed as stand-alone solutions; instead, they are fused with GNSS or UWB using Kalman or particle filtering algorithms to correct

drift and maintain trajectory continuity (He *et al.*, 2020). This fusion is particularly valuable when workers or equipment move through partially obstructed outdoor zones, such as scaffolding areas or between tall structures.

Computer vision technologies have also been applied in outdoor construction to track personnel, vehicles, and equipment. Fixed cameras, drones, and LiDAR systems can generate high-resolution positional data, often at sub-meter accuracy (Han and Golparvar-Fard, 2017). In particular, vision-based systems are effective in wide outdoor areas where clear line-of-sight can be maintained, and they are capable of monitoring orientation and activity in addition to location (Kim *et al.*, 2020). However, their performance is hindered by environmental conditions such as shadows, occlusions from construction machinery, and variable lighting, all of which are common on dynamic construction sites.

Overall, outdoor RTLS in construction depends on GNSS as the baseline technology for open areas, supplemented by UWB and wireless systems in obstructed zones, IoT networks for long-range asset connectivity, IMUs for continuity, and computer vision for detailed contextual awareness. Hybridization of these technologies is essential to overcome the limitations of each and to deliver continuous, accurate, and reliable outdoor localization on construction sites.

2.1.4. Comparison Core Technologies

A variety of positioning technologies fall under the RTLS technologies, each with unique operating principles and trade-offs. Key RTLS technologies include radio-frequency identification (RFID), Bluetooth Low Energy (BLE), Ultra-Wideband (UWB), Wi-Fi fine-time measurement, global navigation satellite systems (GNSS) such as GPS, inertial measurement units (IMUs), and vision-based systems.

RFID-based RTLS can use either passive tags (powered by the reader's signal) or active tags (battery-powered). Passive RFID supports gate-based detection identifica-

tion rather than continuous RTLS; active RFID affords meter-scale locating suitable for yard management and tool/material tracking, often as part of hybrid architectures (Song *et al.*, 2006; Sardroud, 2012; Li *et al.*, 2016). Passive RFID tags have essentially unlimited battery life and very low cost, but they can only be detected at short ranges (a few meters) and require a dense infrastructure of readers to cover a large site. Positioning accuracy with passive RFID is coarse (often 0.5-1 m at best) since detection is zone-based. Active RFID tags broadcast signals that can be triangulated or simply detected by multiple receivers, achieving a few meters accuracy with longer range but at higher tag cost and battery usage. RFID is relatively tolerant of NLOS conditions especially at lower frequencies (UHF or LF) the signals can penetrate or diffract around obstacles- however, standard RFID does not produce precise distance measurements, so its location information is usually zone-or proximity-based rather than continuous coordinates. Overall, RFID offers a robust and low-cost RTLS solution for basic tracking (e.g. gate check-in/check-out or zone alerts) but lacks the fine-grained precision needed for detailed positioning of moving workers.

Bluetooth Low Energy (BLE) beacons are a popular RTLS option due to their widespread presence found in smartphones and inexpensive tags and very low power consumption. In Bluetooth Low Energy (BLE) beacon deployments, two Received Signal Strength (RSS) paradigms dominate: model-based ranging with multilateration and fingerprinting. In the model-based approach, the receiver converts measured power to an approximate range via a calibrated log-distance path-loss model (with site-specific reference power and path-loss exponent), and then computes the position using (weighted) multilateration across three or more anchors. Fingerprinting constructs an offline “radio map” by surveying RSS (Received Signal Strength) vectors at known points; online measurements are matched (e.g., k-NN, probabilistic, or learned models) to the closest fingerprint. Fingerprinting is typically more robust to non-ideal propagation than pure model-based ranging, but it incurs survey overhead whenever layouts, inventories, or change (Bahl and Padmanabhan, 2000; Paterna *et al.*, 2017; Zafari *et al.*, 2019). By contrast, Angle of Arrival (AoA) augments power-only methods with calibrated antenna arrays to estimate the signal’s bearing from

inter-element phase/time differences; combining bearings from multiple anchors (or a bearing with a short-range range estimate) can tighten error distributions toward sub-meter in favorable line-of-sight (LOS) conditions. These gains come at the cost of additional hardware, commissioning/calibration effort, and sensitivity to array geometry and non-line-of-sight (NLOS) reflections considerations that are pronounced on construction sites with moving steel, temporary partitions, and variable worker body-blocking (Faragher and Harle, 2015; Davidson and Pich’e, 2017; Bluetooth SIG, 2021; Zafari *et al.*, 2019).

In such deployments, Received Signal Strength (RSS) methods infer distance from path-loss (or via site-specific “fingerprint” maps); which are simple and cheap to run but typically deliver meter-level accuracy because multipath, human occlusion, metallic clutter, device heterogeneity, and environmental drift distort power readings (Bahl and Padmanabhan, 2000; Rappaport, 2002; Zafari *et al.*, 2019). BLE-based localization typically relies on measuring the received signal strength indicator (RSSI) from fixed beacons or using newer angle-of-arrival features. Under ideal conditions, BLE can achieve 1-3 m accuracy, but in practice often only 5 m or worse, especially in cluttered environments. Signal attenuation and multipath reflections cause significant variability in BLE RSSI, making readings highly sensitive to obstacles (poor NLOS performance). On construction sites with machinery and materials blocking signals, BLE accuracy can degrade substantially. Despite this, BLE remains attractive for its low cost and ease of integration smartphones and simple wearable tags can serve as BLE trackers, and installation of BLE beacons is straightforward. BLE is thus well-suited for coarse positioning or proximity warnings, but alone it may not reliably support high-precision tracking in complex layouts.

Computer vision-based tracking uses cameras (such as static CCTV or wearable cameras) and computer vision algorithms to detect and locate targets. When well-calibrated, vision can achieve sub-meter or even sub-0.5 m accuracy by identifying markers or using visual simultaneous localization and mapping (vSLAM). In construction, vision has been explored for tracking workers by their PPE colors or markers

and for monitoring equipment movements. The big challenge is line-of-sight requirement cameras require an unobstructed view of the target, so any occlusion (by walls, machinery, or other workers) will block tracking. Construction sites with many blind spots and changing layouts make full coverage via cameras difficult. Additionally, vision processing is computationally intensive and can raise privacy concerns (capturing identifiable worker images). The cost of implementing a robust vision RTLS includes not only cameras but also significant computing resources. Therefore, while vision can be very precise in controlled scenarios, it is usually used as a complementary method (for example, to double-check certain zones or to provide rich context in areas where cameras can be installed safely) rather than the sole RTLS on a site.

Traditional Wi-Fi positioning uses signal strength fingerprinting and typically yields 5-15 m accuracy in indoor environments. However, newer IEEE 802.11 protocols (e.g., 802.11mc) provide a Fine Time Measurement (FTM) capability where Wi-Fi access points and devices conduct two-way ranging (similar to UWB's two-way ranging but at a narrower bandwidth). In open line-of-sight, Wi-Fi FTM can achieve decimeter-level ranging accuracy. Nonetheless, on construction sites, the performance of Wi-Fi is often limited by multipath and interference, and the power consumption of Wi-Fi radios is relatively high for battery-powered tags. The advantage of Wi-Fi-based RTLS is the ability to leverage existing wireless infrastructure (access points) - medium cost and moderate integration difficulty, since many sites already have Wi-Fi networks. But the variability of Wi-Fi signals in NLOS conditions usually means Wi-Fi alone cannot provide consistent precise tracking in a dense construction environment. It can serve as a supplementary technology or a backup (for example, using Wi-Fi location when a UWB tag goes out of range), contributing to an overall hybrid solution.

IMUs (consisting of accelerometers, gyroscopes, and sometimes magnetometers) do not provide an absolute position on their own, but they measure relative motion (changes in velocity and orientation). Worn by a worker, an IMU can infer steps taken, direction of movement, or sudden motions (like falls). IMUs have the advantage of being completely immune to occlusion they function regardless of environment, since

they self-containedly measure movement. They also update at high rates (100 Hz or more) with low power. However, IMUs suffer from drift when their data is integrated over time: small measurement errors accumulate, causing the estimated position to diverge from the true location if used alone for navigation. In practice, IMUs in construction are usually integrated with other RTLS technologies (e.g., UWB or GPS) to “dead-reckon” between absolute fixes and smooth out noise. For example, an IMU can interpolate a worker’s movement during brief GPS outages, or detect the orientation of a worker’s body (which can be combined with proximity data to assess if a worker is facing a hazard or not). The cost of IMUs is low (they are in every smartphone and many wearables), and they consume minimal power. Thus, IMUs are an important supporting technology in hybrid RTLS architectures, improving continuity and providing context (like motion patterns), even though they are not typically used for standalone localization due to drift.

The Global Positioning System (GPS) is a satellite-based navigation system developed and maintained by the United States Department of Defense. It provides continuous, real-time positioning, navigation, and timing (PNT) information globally and is one of the most widely used technologies in geolocation, including applications in construction, transportation, agriculture, and defense (Kaplan and Hegarty, 2017). GPS operates by using a constellation of at least 24 Medium Earth Orbit (MEO) satellites that transmit time-stamped signals received by ground-based GPS receivers. By calculating the time delay from multiple satellite signals, the receiver triangulates its position using trilateration. Satellite-based positioning (GPS and other GNSS like GLONASS, Galileo, BeiDou) is the default solution for outdoor location tracking. Standard GPS receivers in smartphones or handheld units achieve 3-5 m horizontal accuracy under open-sky conditions, which is sufficient for many construction applications like tracking equipment across a large site or geofencing broad zones. GPS has global coverage and requires no on-site infrastructure (the satellites are essentially the anchors). This makes it extremely scalable outdoors - workers or machines simply need a receiver device, with no additional cost per unit area of the site. However, GPS signals are very weak by the time they reach Earth and operate in L-band frequencies

(1.5 GHz). Thus, they are easily blocked by buildings, metal structures, and even dense foliage. Indoors or deep inside structures, GPS cannot provide reliable fixes - this is a fundamental limitation for construction sites that have indoor areas or are in urban canyons. Furthermore, the baseline GPS accuracy of a few meters may not be enough for safety applications requiring sub-meter proximity detection (e.g., warning if a worker is within 1 m of a moving vehicle). High-precision modes of GNSS like Real-Time Kinematic (RTK) or augmentation can bring accuracy down to centimeter-level, but these require specialized equipment and still need clear sky view for continuous operation. In summary, GPS is excellent for large-scale outdoor tracking and is essentially mandatory for covering wide-area positioning, but it must be augmented or switched out when it comes to indoor or high-precision needs on construction sites.

Ultra-Wideband (UWB) uses pulsed time-of-flight ranging with sub-nanosecond bandwidth, yielding very high accuracy (on the order of 0.1-0.3 m (Jailani *et al.*, 2022)). UWB anchors require precise synchronization, so hardware and deployment cost are high but tags consume low power (Rodas *et al.*, 2013). UWB signals penetrate walls and resist multipath distortion better than narrowband RF, so performance in NLOS is relatively good. Several RTLS technologies have been applied to construction worker tracking, each with trade-offs in accuracy, power, cost, NLOS robustness, and integration complexity.

UWB stands out as the most precise and NLOS-resilient technology among RTLS options, capable of decimeter accuracy even in cluttered spaces, but it requires significant deployment effort and cost. BLE and Wi-Fi offer inexpensive and readily-available solutions with seamless integration into consumer devices, but their accuracy and reliability in complex environments are substantially lower, making them better suited for coarse tracking or augmenting other systems. RFID occupies a middle ground: it can be quite robust in harsh conditions and inexpensive, but typically provides zone-level or meter-level location fidelity. Vision systems can yield high precision and rich information (like recognizing specific activities), yet they are limited by line-of-sight and are resource-intensive.

Inertial sensors contribute by filling gaps and adding context but must be combined with other absolute positioning references to bound drift. Given these varied characteristics, it is increasingly evident that no single technology is “best” for all construction scenarios a hybrid approach that combines multiple RTLS technologies is often necessary to ensure continuous, reliable tracking across the diverse conditions of a construction project.

Table 2.1. RTLS technology comparison based on literature.

Technology	Accuracy	Cost (approx.)	Power/ Energy	Integra- tion Difficulty	NLOS Tolerance
GPS	5-10 m (outdoor)	None (device- built-in)	Low (passive receiver)	Very easy (smartp- hones)	Poor (blocked indoors)
UWB	10-30 cm (Jeilani <i>et al.</i> , 2023)	High (speci- alized tags/ anchors)	Medium- High (battery)	Moderate (anchors setup)	Moderate (some Pene- tration but errors)
BLE	1-5 m (BLE beacons)	Low (cheap beacons)	Low (low- power)	Easy (ubiquitous)	Poor/ Moderate (environ- mental)
Wi-Fi	5-15 m (Jeilani <i>et al.</i> , 2023)	Medium (use existing APs)	Medium (routers)	Easy (leverages infrastructure)	Moderate (inter- ference)
RFID (Active)	1-3 m	Low (tags)	Low (passive) / Medium (active)	Moderate (readers placement)	Passive: poor, Active: Moderate

Table 2.1 RTLS technology comparison based on literature. (cont.)

Technology	Accuracy	Cost (approx.)	Power/ Energy	Integra- tion Difficulty	NLOS Tolerance
Computer Vision	0.5-2 m (object- level)	High (cameras + compute)	High (process- sing)	Moderate (camera setup, line- of-sight)	None (requires direct view)

In summary, UWB stands out as the most precise and NLOS-resilient technology among RTLS options, capable of decimeter accuracy even in cluttered spaces, but it requires significant deployment effort and cost. BLE and Wi-Fi offer inexpensive and readily-available solutions with seamless integration into consumer devices, but their accuracy and reliability in complex environments are substantially lower, making them better suited for coarse tracking or augmenting other systems. RFID occupies a middle ground: it can be quite robust in harsh conditions and inexpensive, but typically provides zone-level or meter-level location fidelity. Vision systems can yield high precision and rich information (like recognizing specific activities), yet they are limited by line-of-sight and are resource-intensive. Inertial sensors contribute by filling gaps and adding context but must be combined with other absolute positioning references to bound drift. Given these varied characteristics, it is increasingly evident that no single technology is “best” for all construction scenarios a hybrid approach that combines multiple RTLS technologies is often necessary to ensure continuous, reliable tracking across the diverse conditions of a construction project (Roberti, 2021a; Alarifi *et al.*, 2016).

2.1.5. Challenges in RTLS Deployment: Indoor vs Outdoor

Construction sites often encompass a mix of outdoor and indoor (or semi-indoor) environments for example, an open excavation area adjacent to a partially completed

building. This heterogeneity poses a significant challenge for any single RTLS technology to cover seamlessly. GNSS/GPS works well outdoors but fails indoors; conversely, indoor localization tech like UWB or Wi-Fi can cover building interiors but typically over limited range. Moreover, construction sites are cluttered with obstacles (steel beams, concrete walls, machinery, workers themselves) that can obstruct or reflect signals, leading to accuracy degradation. In this section, we discuss two major technical challenges for RTLS in construction: Non-Line-of-Sight (NLOS) and multipath propagation effects, and the differences in maintaining accuracy indoors vs. outdoors (GPS signal occlusion outside vs. drift or interference inside). Addressing these challenges is crucial for RTLS reliability.

2.1.6. Towards Proactive Monitoring: The Role of RTLS in Dynamic Environments

Traditional safety management on construction sites has been largely reactive, relying on incident reports and after-the-fact analysis. In recent years, there is a marked shift toward active and predictive safety monitoring enabled by real-time data from field sensors and RTLS. Rather than simply logging positions for later review, modern RTLS deployments aim to facilitate proactive decision-making for example, automatically warning workers and operators of impending hazards so that accidents are prevented in the first place. This section discusses how RTLS technologies support this proactive paradigm and why real-time location awareness is critical in safety-critical sectors like construction.

Beyond simple tracking, RTLS enables a shift toward proactive monitoring and management, especially in dynamic and high-risk environments like construction sites. A construction project is inherently a fast-changing setting—workers move, equipment operates, and site conditions evolve rapidly, often in unpredictable ways.

Traditional safety practices (e.g. periodic inspections or static rules) may not respond quickly enough to emerging hazards. By providing continuous, real-time data

on the positions and movements of resources, RTLS can facilitate timely interventions to prevent accidents and optimize operations before problems escalate.

One key benefit of real-time location data is the ability to implement geofencing and automated alerts. RTLS platforms can define virtual zones on site (for example, around hazardous areas or restricted zones) and trigger alarms if a worker or piece of equipment approaches danger. This allows site managers to address risks before an accident occurs. For instance, if a worker equipped with an RTLS tag comes too close to an unsafe zone (like the swing radius of a crane or an open edge at height), the system can send an immediate warning to the worker or supervisors (Han *et al.*, 2025; Navigine, 2025). By continuously monitoring the “risk envelope” around each worker and asset, RTLS supports a proactive safety management approach rather than a reactive one. A 2025 systematic review of fall-prevention technologies noted that real-time monitoring technologies allow hazards to be addressed “before an accident occurs”, highlighting how live location and sensor data can enable preemptive action (Pereira *et al.*, 2025). Some pioneering studies have even developed prototype systems that actively signal potential hazards for example, an early “proactive safety management” system in 2011 demonstrated real-time tracking of workers to signal overhead dangers in construction sites (Pereira *et al.*, 2025).

2.2. Real Time Location Systems in Construction Sector

Construction sites present a compelling use case for RTLS due to their inherently dynamic and hazardous nature. Unlike the controlled environment of a factory floor, a construction site’s layout evolves over time, with workers and equipment moving between indoor and outdoor areas and across multiple levels. Managing safety and logistics in such settings is challenging. RTLS technologies have been applied in construction for a range of purposes: enhancing worker safety through geofencing and proximity alerts, monitoring equipment usage and preventing collisions, tracking materials to improve logistics, and feeding real-time data into management systems (like schedules or BIM models) for better decision-making. This section explores how RTLS

is used in construction, focusing especially on safety-oriented applications and integration into site operations. A wide variety of RTLS technologies have been studied and deployed in construction. A systematic review by Soltanmohammadlou *et al.* (2019) emphasizes that RTLS enable automated monitoring of on site resources (workers and machines) specifically to prevent hazard exposures and accidents. In short, RTLS provides the “where and when” of everything on site that information can be used to prevent dangers before they occur.

In recent years, the field of construction safety has undergone a significant shift from reactive strategies toward active and predictive monitoring. This transformation is primarily driven by integration of advanced technologies such as Internet of Things (IoT), Real-Time Location Systems (RTLS), wearable sensing devices, and artificial intelligence (AI). These technologies collectively facilitate the continuous collection transmission, and analysis of site data, enabling dynamic safety assessment and intervention. Continuous dynamic surveillance of the construction site is enabled allowing for immediate detection of hazardous conditions (Chloe *et al.*, 2017; Soltanmohammadlou *et al.*, 2019). Soltanmohammadlou *et al.* (2019) conducted a systematic literature review on the application of Real-Time Locating Systems (RTLS) in construction safety management and highlighted a significant shift toward their use not only for location monitoring but also for real-time hazard prevention. The study emphasized that continuous tracking of worker locations and movement trajectories allows for the generation of proactive alerts when deviations from designated safe zones or prescribed workflows occur. This enables dynamic risk mitigation by detecting unsafe proximity or unauthorized entry into hazardous area. RTLS technologies such as Ultra-Wideband (UWB) and Bluetooth Low Energy (BLE) allow for centimeter-level tracking of worker movement and behavior, making it possible to identify unsafe patterns and prevent potential incidents before they occur (Zhou *et al.*, 2012; Alarifi *et al.*, 2016).

Another significant advantage of RTLS in construction is its synergy with Building Information Modeling (BIM) and digital twin platforms. Live location data fed into a 3D BIM model allows the site to be visualized in real time, with critical safety

zones and moving entities displayed dynamically. This integration means, for example, that the hazardous swing zone of a crane or the perimeter of an excavation can be virtually delineated within the model, and any breaching of these zones by a worker triggers an immediate alert in both the field and the digital interface. Recent studies have employed such RTLS-enhanced digital twins to run “what-if” safety scenarios and to continuously monitor site conditions (e.g. highlighting when too many workers congregate in a confined space). The real-time visualization and analytics provided by these integrated systems give project managers an unprecedented level of situational awareness. They can not only respond to ongoing issues but also anticipate and preempt potential conflicts or hazards by virtually testing changes in crew deployment or site layout.

In summary, RTLS technology has become a cornerstone of proactive safety management in construction. It provides granular, moment-by-moment visibility into the otherwise chaotic environment of a jobsite. This real-time visibility translates into actionable safety intelligence: supervisors can intervene the instant a dangerous situation is detected, and data trends can be analyzed to prevent future incidents. As one review noted, RTLS essentially delivers the “where and when” of everything happening on site, information that can be immediately leveraged to avert accidents. By integrating RTLS into daily operations, construction projects are better equipped to protect workers through timely alerts, smarter planning, and an overall shift from reactive safety measures to a prevention-oriented safety culture.

2.2.1. Construction Worker Tracking

In practice, RTLS are often deployed to prevent collisions and to analyze worker behavior. For collision avoidance, the system continuously calculates distances between tracked assets: if a worker tag comes too close to a moving machine tag, it triggers alarms for both parties. Teizer *et al.* (2010) demonstrated a UWB-based alert system on cranes that warned operators when personnel entered the danger zone (Soltanmohammadlou *et al.*, 2019).

Beyond immediate alerts, RTLS enable behavioral analytics:

Patterns of movement can be analyzed to identify unsafe practices. For instance, a study by Cheng *et al.* (2012) used UWB tracks to detect when workers repeatedly remain under suspended loads an indicator for risky behavior and that contributed into safety training. Cheng *et al.* (2012) showed that tracking workers with UWB could reveal these patterns of risky behavior, information which was then used to improve safety training and supervision. This ability to learn from high-resolution movement data rather than from accidents after the fact represents a major advancement in construction safety management.

One of the most prominent applications of RTLS in construction is the tracking of construction workers on site. Real-time location of personnel is “crucial information for site safety management” because many construction accidents occur suddenly and without warning. By equipping workers with location tags (such as wearable RFID/UWB tags embedded in safety vests or helmets), managers can continuously know each worker’s whereabouts and movement history. This capability underpins a variety of safety measures: for example, if a worker enters a defined danger zone (e.g. near an active crane or heavy equipment), the system can immediately issue a warning to the worker and the equipment operator to prevent a collision. Such proximity alert systems have been tested using technologies like UWB and RFID, often yielding positive results in reducing close calls. In one experimental system, UWB tags on workers combined with equipment-mounted sensors managed to detect hazardous proximities and issue automatic slowdown commands to machines, illustrating the life-saving potential of precise worker tracking.

Beyond safety, productivity and workforce management stand to benefit from worker tracking. Knowing the real-time locations and movements of workers enables analysis of how time is spent on site, identification of bottlenecks or idle time, and optimization of crew allocations. For example, if RTLS data shows certain workers consistently waiting for materials or walking long distances to fetch tools, management can adjust the site layout or resource delivery to reduce wasted time. One research

project developed an RTLS-based system to measure the productivity of personnel and even to rank worker performance in various tasks. By instrumenting workers with indoor positioning tags, the system could automatically log productive vs. unproductive periods and output performance metrics, allowing supervisors to promptly implement improvements. The dynamic nature of construction work makes manual productivity tracking difficult, but an RTLS provides objective, continuous data. This can feed into incentive programs or project controls to enhance overall efficiency. It is noted that to truly capitalize on this, careful consideration of how to filter and interpret the raw location data is required (to distinguish, say, legitimate stationary work at one location from true idleness).

Furthermore, RTLS-based worker tracking improves accountability and record-keeping on site. It creates an automatic log of who was where at what time, which can be invaluable for investigations (in case of an incident) and for verifying work progress or attendance. Traditional methods like paper sign-in sheets or manual headcounts are error-prone, whereas an RTLS can maintain an accurate roster of all personnel on site in real time. This information can ensure that, for example, no worker is left behind during evacuations (the system can instantly highlight anyone still in a danger zone).

In summary, tracking construction workers via RTLS is a foundation for both improved safety outcomes and better workforce management. It enables immediate hazard detection (saving lives and preventing injuries), provides data for proactive safety interventions (education and process changes), and yields operational insights to streamline labor usage. As the technology matures, we are likely to see RTLS become as essential as personal protective equipment on job sites, forming a critical component of the connected, smarter construction site.

2.2.2. Integration into Site Operations, Management Systems and Health and Safety Executions

RTLS delivers its greatest value when tightly integrated with the digital backbone of construction operations HSE platforms, procurement and inventory systems, scheduling related to 4D BIM, and project control dashboards. Rather than a stand-alone tool, RTLS becomes a high-frequency data stream that enhances situational awareness, compresses decision cycles, and supports evidence-based control of safety, logistics, and productivity (Navon, 2007; Li *et al.*, 2016). In the Construction 4.0 paradigm, this integration converts raw location traces of workers, equipment, and materials into actionable intelligence for proactive management. (Navon, 2007; Li *et al.*, 2016).

RTLS technologies are used in safety management and HSE operations. Combining RTLS with smart PPE (e.g., heat or heart-rate indicators) enables context-aware risk detection such as prolonged exposure to high-heat zones or fatigue hotspots, supporting preventative interventions. In parallel, studies benchmarking UWB/BLE/Wi-Fi for worker tracking demonstrate the feasibility and accuracy of these technologies for safety-critical applications on complex sites (Cheng *et al.*, 2011; Park *et al.*, 2016). Industry practice increasingly unifies these streams in centralized safety dashboards with geofencing and real-time alerts (Kwant, 2025). (Cheng *et al.*, 2011; Park *et al.*, 2016; Kwant, 2025). When RTLS is connected to the HSE platform and on-site access systems, geofences around hazardous zones can automatically detect unauthorized entries, trigger alerts, and log safety breaches without manual intervention. This transforms safety oversight from periodic inspection to persistent monitoring with automated documentation (Extronics, n.d.). Evidence from proximity-safety research further shows that integrating location sensing with alerting reduces struck-by and collision risks by warning workers and operators in real time (Teizer *et al.*, 2010; Park *et al.*, 2016). (Extronics, n.d.; Teizer *et al.*, 2010; Park *et al.*, 2016). RTLS-enabled badges and readers support emergency call buttons, automatic mustering, and “who-is-where” visibility for incident commanders. During evacuations, the system can confirm attendance at

muster points and identify missing personnel by name and last-seen location, while synchronizing records into the HSE log (Extronics, n.d.). (Extronics, n.d.). Recent work proposes affordable UWB-based proximity warning systems that detect worker and vehicles conflicts and issue alerts while addressing data-protection constraints especially relevant for European sites (Ventura *et al.*, 2023).

RTLS technologies are also used in procurement and material logistic operations in construction sector. The implementation of RTLS enabled automated receiving processes and enhanced item-level inventory visibility. When deliveries arrive with active RFID/GPS tags, RTLS can automatically mark items as received and pinpoint their storage location, pushing time-stamped updates to procurement and inventory systems. Foundational studies show that RFID/RTLS tracking improves the visibility of material status and delivery dates, reducing search time and wait time (Song, Haas and Caldas, 2006; Sardroud, 2012). (Song, Haas and Caldas, 2006; Sardroud, 2012). RTLS-driven on-site localization shortened searches and curtailed walking waste across the site. Knowing exactly which laydown area, bay, or floor holds a component reduces non-productive travel and shortens task setup. Early construction RTLS research envisioned these benefits and linked them to tangible site-management gains when paired with visualization (Sadeghpour, 2006; Song, Haas and Caldas, 2006). (Sadeghpour, 2006; Song, Haas and Caldas, 2006). Integrating RTLS telemetry with 4D BIM enabled automated, schedule-conformant progress updates and variance detection. RTLS with 4D schedules allows components to auto-update status as they move into position, enabling live plan-versus-actual comparisons. Newer methods fuse RTLS with computer vision to produce near-real-time progress reports without heavy manual input (Shamsollahi *et al.*, 2024). (Sadeghpour, 2006; Shamsollahi *et al.*, 2024). Implementing RTLS established supply-chain feedback loops linking on-site consumption telemetry to enterprise procurement, enabling demand-driven replenishment and variance control. RTLS-measured site stock levels can trigger reorder signals or dispatch requests, tightening just-in-time flows and lowering idle inventory. Systematic reviews of RTLS in construction identify material flow visibility and leaner logistics as recurring benefits when site data is linked to procurement (Li *et al.*, 2016).

Embedding RTLS telemetry in project management workflows enabled continuous, data-driven performance monitoring, variance detection, and corrective action. Streaming RTLS data into dashboards allows managers to compare planned manpower allocations with actual location heatmaps by zone, revealing under- or over-staffing and enabling rapid crew re-balancing (Navon, 2007; Cheng and Teizer, 2013). RTLS also supports automated work sampling, distinguishing productive time at the workface from walking/waiting, and surfacing bottlenecks invisible to periodic observation (Cheng and Teizer, 2013). (Navon, 2007; Cheng and Teizer, 2013). RTLS provides metrics such as number of trips, dwell/idle time, and paths, informing layout optimization (e.g., relocating staging areas) and sequence tuning. GPS-based studies on equipment operations demonstrate automated spatiotemporal analysis for utilization improvement (Pradhananga and Teizer, 2013), while broader evaluations confirm UWB/BLE viability for tracking in harsh environments (Cheng *et al.*, 2011). (Pradhananga and Teizer, 2013; Cheng *et al.*, 2011). Live mapping of workers, equipment, and tagged components onto a 4D/3D model yields a near real-time “as-built” for deviation detection and faster coordination decisions. Demonstrations of RTLS-BIM integration, and more recently RTLS + deep-learning vision, show automated progress tracking with immediate managerial value (Teizer *et al.*, 2015; Shamsollahi *et al.*, 2024). Industry deployments in adjacent sectors (e.g., energy) report reduced downtime and better resource allocation when asset and personnel movements are tracked in real time (Navigine, 2024). (Teizer *et al.*, 2015; Shamsollahi *et al.*, 2024; Navigine, 2024).

RTLS integration turns location traces into operational intelligence across safety, logistics, and performance. In safety, geofencing, proximity alerting, and mustering move management from reactive to proactive. In logistics, automated receiving, precise on-site localization, and 4D updates synchronize supply with installation. In project control, RTLS-driven dashboards surface variances early and quantify waste, enabling agile re-planning. Although integration poses challenges (data volume, interoperability, privacy), cumulative evidence from peer-reviewed studies and field implementations indicates higher safety performance, reduced delays, and improved labor/equipment productivity when RTLS is embedded into the project’s digital workflow (Teizer *et al.*,

2010; Pradhananga and Teizer, 2013; Shamsollahi *et al.*, 2024). (Teizer *et al.*, 2010; Pradhananga and Teizer, 2013; Shamsollahi *et al.*, 2024).

2.2.3. Integration into Digital Twin

Furthermore, research by Zhang *et al.* (2022b) and Talmaki and Kamat (2022) demonstrates the effectiveness of digital twin technologies in visualizing hazard zones and conducting safety simulations. Study of Zhang *et al.* (2022b) achieved high performance, with hazard detection accuracies exceeding 96% and spatial precision reaching approximately 11.5 centimeters. These findings highlight the transformative potential of integrating digital twins with AI and BIM-enabled RTLS systems to evolve traditional safety management into real-time, predictive, and prescriptive frameworks.

DT models that integrate RTLS inputs have been successfully used to analyze human-resource effectiveness and safety-critical activities, providing actionable insights on spatial behavior and task execution (Ruppert *et al.*, 2020). Moreover, by fusing BIM with live RTLS data, safety-critical regions such as crane swing paths or excavation perimeters can be visualized dynamically within a digital interface, prompting automatic alerts when personnel intrude upon high-risk areas (Agnusdei *et al.*, 2021).

2.2.4. Examples of RTLS in Construction

In construction settings, GPS has been increasingly used for equipment tracking, site logistics, surveying, and progress monitoring (Zhu *et al.*, 2021). Its ability to provide real-time geolocation data has supported automation in earthmoving equipment and facilitated safety mechanisms such as geofencing. For example, workers wearing GPS-equipped smart vests can be monitored to ensure they remain within safe zones, while heavy machinery can be geo-restricted to avoid collision zones (Lee *et al.*, 2018).

An early demonstration of UWB in construction was reported by Teizer *et al.* (2008). They deployed a UWB real-time location system (RTLS) to track workforce

and equipment in three dimensions on a job site. This system (using a Time-of-Flight UWB technology available at the time) was able to automatically sense the location of resources without manual surveying. UWB anchors covered both open areas and partially enclosed areas of the site, allowing continuous tracking as people moved about. Accuracy was reported nearly 30 cm in open areas and worse in NLOS spots, but still sufficient to distinguish which zone or floor a worker was on. This case proved the feasibility of UWB for construction asset localization and laid groundwork for later hybrid approaches. It also identified practical issues such as multi-path from metal equipment and the need for robust tags that could withstand construction conditions.

Cheng *et al.* (2011) conducted a performance evaluation of a commercial UWB system (Ubisense) on an active construction site with harsh conditions. This environment included heavy machinery, significant metallic clutter, and both indoor and outdoor environments.

In another field study, Siddiqui *et al.* (2019) tested UWB tracking on a dense urban site in Vancouver. They found the system to be highly accurate even under metal scaffolding though occasional signal blockage did occur when equipment obstructed the line of sight. Their key recommendation was to combine UWB with other sensors like GPS or video to fill in these gaps, creating a more resilient safety system.

2.2.5. Hybrid Use of RTLS

Hybrid RTLS approaches in construction have been trialled in both experimental and field studies. For example, Teizer *et al.* (2008) demonstrated one of the earliest real deployments of UWB in a construction site, achieving 30 cm accuracy across open and semi-enclosed spaces, and highlighting challenges such as multipath from steel equipment. Later, Cheng *et al.* (2011) evaluated the commercial Ubisense UWB system on an active site with significant metallic clutter, finding submeter accuracy but with noticeable NLOS degradation. These works showed UWB's feasibility but also its limitations, motivating hybrid architectures. More recently, Siddiqui *et al.*

(2019) set up a wireless UWB system outdoors on the site (interestingly, testing UWB outside in a dense environment). They reported high accuracy (10–20 cm) even under scaffoldings, with occasional blockages when heavy equipment fully obstructed line of-sight. They concluded that while UWB performed well, adding a complementary technology like GPS for when UWB is blocked or video for line-of-sight confirmation would make the system more robust. This underscores that even outdoors, mixing tech is valuable: a hybrid of UWB+GPS outside could alleviate those few blockage moments (GPS might see a sky view when UWB can't go through machinery, for instance). Given the limitations of individual technologies, hybrid RTLS architectures have become the predominant approach in construction. These systems integrate GNSS for global coverage with UWB or BLE for local accuracy, while IMUs provide continuity and computer vision adds contextual awareness (Li *et al.*, 2016; Teizer, 2015). Hybrid tags are now capable of switching seamlessly between GNSS outdoors and UWB or BLE indoors, with supplemental beacons deployed in GNSS-denied corridors. Such integration not only enhances positional accuracy and continuity but also reduces the risks associated with signal blockage and infrastructure gaps, enabling more reliable safety monitoring and digital twin implementations.

Tracking workers seamlessly as they move between outdoor and indoor areas requires integrating outdoor RTLS technologies with indoor RTLS. Park *et al.* (2017) further expanded hybridisation by fusing BLE beacons, inertial sensors and BIM data into a corrective tracker, enhancing reliability in challenging indoor conditions. Beyond GPS and UWB, RFID and Wi-Fi hybrids have also been explored: Soltanmohammadlou *et al.* (2019) review how combining RFID with other wireless sensors enables automated hazard monitoring and geofencing in mixed environments.

2.3. Research Gap in the Literature

Despite the promise of hybrid RTLS, the construction-safety literature reveals clear gaps. Most hybrid-tracking schemes remain at the prototype stage and have not been proven in large, real construction environments. As Pereira *et al.* (2025) note,

even when innovative monitoring technologies are identified, “further research is needed to verify the applicability of these technologies on construction sites” (Pereira *et al.*, 2025). In particular, truly field-deployable, real-time safety frameworks are scarce.

Few studies describe systems that run continuously on an active project and issue immediate warnings; instead most demonstrate offline tests or simulations. Likewise, integration with Building Information Modeling (BIM) tools is limited. A handful of works (e.g. Park *et al.*, 2017) illustrate the concept of linking hybrid sensors to a BIM map, and Fagnoli *et al.* (2024) report on a prototype providing real-time safety alerts via a BIM tracking interface (Pereira *et al.*, 2025). However, there are essentially no end-to-end implementations that marry multi-technology RTLS directly into common BIM platforms (such as Autodesk Revit) and scale to live sites. In short, the literature lacks a comprehensive hybrid RTLS solution that is integrated with BIM environment which is revit and proven in situ. This gap is critical so without robust field deployments and tightly coupled Revit such as BIM environment integration, the envisioned “proactive safety” benefits of hybrid tracking cannot yet be realized (Pereira *et al.*, 2025).

2.3.1. Lack of Hybrid Systems in the Construction Industry

The primary benefit is complete coverage spatially (indoor/outdoor) and in terms of accuracy (using the right tool for each area). Safety events can be monitored anywhere on site. Another benefit is redundancy: if one system fails or is temporarily degraded, the other can sometimes fill in. For instance, if GPS has a glitch, UWB (if in range) still keeps track; if UWB anchors go down, GPS still gives location (with lower accuracy) so the person isn’t entirely lost.

Redundancy improves reliability of the safety system, which is crucial for adoption (workers/managers need to trust that the system will always be tracking). Moreover, hybrid data can be cross-validated if both GPS and UWB are available in an open area, they can be compared to detect any anomaly (e.g., if GPS says one thing and

UWB another by a large margin, one of them is wrong probably GPS multipath; the system could then automatically discard the GPS reading as outlier, using UWB).

GPS (and other GNSS) provides reliable positioning under open sky, but fails inside buildings or deep construction pits where satellite signals are blocked. UWB, on the other hand, increases the precision in indoor localization but covers a limited area (typically a building or site) and requires installed infrastructure. In a typical hybrid system, workers wear devices that use GPS outdoors and switch to UWB indoors. For example, one framework uses GPS coordinates whenever the tag has a view of the sky, and seamlessly hands off to UWB triangulation inside structures. This ensures there are “no blind spots” in overall coverage. Such hybrid tracking is crucial at sites that span indoor and outdoor zones.

Though tracking transition algorithms can be complex, initial studies have shown that combining GPS (for rough, long-range tracking) with UWB (for fine indoor localization) can achieve continuous tracking across the GPS-denied/available boundary. In doing so, real-time safety monitoring becomes site-wide: a worker exiting a building remains visible on the digital twin model map, and a worker re-entering goes back under UWB coverage without a gap.

Altogether, the literature shows that hybrid RTLS designs – whether GPS+UWB, BLE+UWB, or Wi-Fi+RFID – are motivated not only by accuracy but also by the need for seamless continuity across different environments and context-awareness. In construction safety, these systems feed location data into BIM or digital twin environments, enabling geofence alerts, collision avoidance, and proactive hazard detection (Teizer *et al.*, 2008; Park *et al.*, 2017; Siddiqui *et al.*, 2019; Soltanmohammadlou *et al.*, 2019).

2.3.2. Need for Field Deployable, Real-Time and Proactive Safety Frameworks

Although research on indoor and outdoor localisation technologies has expanded significantly in recent years, much of the literature remains limited to laboratory settings, small-scale pilots, or retrospective analyses. Fully integrated systems that can be deployed continuously on active construction sites and prevent incidents in real time are still scarce (Soltanmohammadlou *et al.*, 2019; Pereira *et al.*, 2025). Construction sites are uniquely dynamic environments where crews, equipment, and temporary works change daily, where structural steel and reinforced concrete produce severe multipath effects, and where workflows frequently traverse between open outdoor zones and semi-enclosed or fully indoor areas. In such contexts, conventional reactive approaches to safety management that rely on reporting and post-incident documentation are inadequate. What is required is a field-deployable framework that can operate reliably under real jobsite conditions, fusing data from heterogeneous sensors, reasoning about emerging risks, and issuing actionable alerts within seconds (Teizer *et al.*, 2010; Park *et al.*, 2016; Li *et al.*, 2016). The concept of “field deployability” implies that an RTLS-based safety framework must sustain accuracy and robustness under the environmental and organisational constraints of live projects.

This includes the ability to maintain indoor–outdoor continuity through hybrid RTLS architectures that combine GNSS for open areas with UWB, BLE or Wi-Fi-based positioning in enclosed zones. Seamless handovers are essential, as ad-hoc switching often introduces discontinuities and errors (Li *et al.*, 2016; Park *et al.*, 2017). Equally important is the capacity to provide robust estimation under non-line-of-sight conditions, where multipath and body or steel blocking cause systematic errors. This challenge is typically addressed through advanced filtering techniques such as Mahalanobis distance outlier rejection, robust or unscented Kalman filters, or particle filters, often aided by IMU bridging (Alarifi *et al.*, 2016; He and Chan, 2016; Jin and Sadeghpour, 2022). Temporal synchronisation represents another major obstacle since UWB systems commonly operate at 10–20 Hz, while GNSS provides outputs at around 1 Hz.

Fusing such asynchronous data streams requires precise timestamping and algorithms designed to accommodate irregular sampling rates (Zhou *et al.*, 2022; Gong *et al.*, 2024).

For such frameworks to be practically useful, localisation data must remain geometrically consistent with project control networks and BIM coordinate systems. This involves rigorous calibration of local RTLS frames, anchor layouts optimised against dilution of precision effects, and ongoing quality control (Li *et al.*, 2016). End-to-end latency also needs to be minimal, with edge processing and lightweight communication pipelines ensuring that alerts can be triggered within sub-second timescales (Teizer *et al.*, 2010; Park *et al.*, 2016). Furthermore, integration with BIM platforms such as Revit or Autodesk Forge allows geofences and rules to be embedded directly into digital models, enabling both real-time warnings and 4D visual analytics to enhance situational awareness (Teizer *et al.*, 2015; Shamsollahi *et al.*, 2024).

In addition to technical robustness, field-deployable systems must also account for human and organisational factors. Privacy-by-design approaches are critical, especially under stringent regulatory contexts such as the EU, and should include pseudonymisation, configurable retention policies, and role-based access (Ventura *et al.*, 2023). Operational resilience further requires redundancy across multiple modalities, automated health monitoring, and devices that can be maintained with minimal downtime (Cheng *et al.*, 2011; Li *et al.*, 2016). Equally, human-centered alerting mechanisms ranging from ergonomic wearables to graded, visual or auditory feedback are necessary to minimize nuisance alarms and maximize worker acceptance (Kwant, 2025; Extronics, n.d.).

The evolution from monitoring to truly proactive safety management rests on the ability of RTLS to anticipate hazards rather than merely document them. Rule-based interventions such as dynamic geofencing around crane swing radii or excavation zones, combined with real-time vehicle–pedestrian separation alerts, exemplify this shift (Teizer *et al.*, 2010; Park *et al.*, 2016). Recent reviews underline that proactive

frameworks allow interventions before accidents occur, marking a fundamental transition from auditing to prevention (Pereira *et al.*, 2025). When linked with digital twin environments, RTLS further enables predictive simulation of crowding, sequencing alternatives, and risk scenarios, thereby enhancing both planning and operational safety (Zhang *et al.*, 2022b; Talmaki and Kamat, 2022; Shamsollahi *et al.*, 2024).

In the context of hybrid architectures, the evidence increasingly supports a fusion-centric approach in which GNSS provides global coverage outdoors, UWB or Wi-Fi FTM/BLE-AoA ensures sub-meter accuracy indoors, IMUs maintain continuity during occlusion, and visual systems provide contextual information. These are typically integrated through robust Kalman-based or ensemble filters with systematic outlier rejection, with data streams transmitted to BIM-connected platforms capable of managing geofences and issuing alerts (Li *et al.*, 2016; Teizer, 2015; Omrany *et al.*, 2022; Zhang *et al.*, 2021). Such frameworks are consistent with Construction 4.0 objectives, although their scalability depends on addressing installation, maintenance, and cost challenges.

Finally, to substantiate claims of “field applicability”, validation protocols must extend beyond accuracy tests in controlled environments. Multi-week deployments on active sites should measure horizontal error distributions, continuity and handover success, latency from sensing to alert, and the rate of true versus false safety alarms. In parallel, integration outcomes such as BIM model alignment accuracy, automated incident logging, and muster compliance must be reported alongside worker acceptance and maintenance burdens (Navon, 2007; Pradhananga and Teizer, 2013; Extronics, n.d.).

2.3.3. Positioning of this Study within the Identified Research Gap

The full potential of Real-Time Location Systems (RTLS) for construction safety management can only be realized through the development of an integrated, real-time fusion framework that connects heterogeneous location data streams with the Building

Information Modeling (BIM) environment as Revit. While advanced sensors such as Ultra-Wideband (UWB), and Global Positioning System (GPS) are widely used for tracking and monitoring purposes, their effectiveness depends largely on the capacity to harmonize and refine the data they generate. In this context, data fusion techniques such as Kalman filtering have been recognized as critical enablers (Omran *et al.*, 2022; Zhang *et al.*, 2021).

Construction sites are inherently complex spatial systems. They often include both indoor and outdoor environments, where indoor zones might involve basements, multi-level structures, or shielded steel environments, while outdoor areas can extend over several hectares, sometimes reaching scales of 1 kilometer or more. This spatial heterogeneity poses a significant challenge for any single RTLS technology. The integration of various sensor technologies into Real-Time Location Systems (RTLSs) holds significant promise for enhancing their applicability in monitoring the health and safety of multiple workers in dynamic environments (Lim *et al.*, 2016). Despite this potential, earlier research has emphasized the necessity of validating these systems through large-scale, real-world implementations to assess their localization accuracy and reliability (Lee *et al.*, 2014; Li *et al.*, 2013; Luo *et al.*, 2014). Furthermore, a comprehensive evaluation must encompass both indoor and outdoor environments, as construction workflows often span across these spatial contexts (Saidi *et al.*, 2011). UWB systems, while offering centimeter-level precision in indoor environments, face operational limitations outdoors due to line-of-sight dependency, reduced range, high labor in employment, set-up calibration works and susceptibility to environmental interference (Li *et al.*, 2020). Conversely, GPS provides coverage in large outdoor settings but lacks the precision needed for proximity-based hazard alerts in close-quarters or indoor construction zones.

Therefore, the implementation of a hybrid and context aware positioning system is essential to ensure continuous and reliable location tracking across diverse heterogeneous environments with both indoor and outdoor zones. Previous research has increasingly pointed to sensor fusion frameworks that integrate GPS for large-scale

outdoor coverage, UWB for high-accuracy indoor localization (Li and Becerik-Gerber, 2011; Du *et al.*, 2023). Among these, Kalman filtering stands out as a well-established method for continuously updating positional estimates by merging noisy and asynchronous data sources into a coherent reflection (Zhou *et al.*, 2022). Moreover, advanced variants like the Asynchronous Ensemble Kalman Filter (AEnKF) have been developed to handle data arriving at irregular intervals, further enhancing the filter’s applicability in real-world scenarios (Gong *et al.*, 2024).

Omran *et al.* (2022) emphasize that effective data fusion at the sensor level is foundational to Digital Twin (DT) implementations in construction, enabling reliable, real-time visualization and interaction. When fused location data is streamed into a BIM environment which is Revit model, it creates a closed-loop feedback system: the Revit model provides spatial and contextual information of worker pathways and hazardous location triggers, RTLS provides dynamic worker location data, and the Kalman filter ensures optimal localization by smoothing noise and bridging sensor gaps. The refined position information can then trigger automated safety alerts, proximity warnings, or visual risk heatmaps within the Digital Twin platform.

In conclusion, this master thesis proposes a hybrid RTLS framework that focuses on Ultra-Wideband (UWB) data for indoor localization and GPS dataa for outdoor tracking. This Revit-integrated fusion framework is not merely a technological improvement it is a strategic necessity. It bridges the gaps between fragmented positioning technologies and enables a cohesive safety monitoring strategy that adapts to the spatial and temporal complexity of construction projects. By enabling the BIM environment visualization simulating real-world worker transitions between indoor and outdoor zones on construction sites, this research aims to leverage evaluations the performance, continuity, and practical applicability of a framework. However, despite its theoretical promise, such a fully integrated system remains largely unrealized in practice, marking it as a significant open research challenge with high potential for both academic advancement and practical impact.

Despite notable progress in real-time localization and digital-twin integration for construction safety, several unresolved limitations still constrain deployment, scalability, and reliability in practice. These challenges cut across environmental effects on signals, temporal misalignment among sensors, algorithmic assumptions, indoor–outdoor continuity, and the depth of BIM environment integration.

Construction sites contain reinforced concrete, steel members, and temporary installations that cause strong attenuation and multipath propagation. In such settings, GPS often degrades or fails entirely in indoor or semi-enclosed areas where slabs obstruct satellite line-of-sight. Ultra-Wideband (UWB), while generally the indoor accuracy leader, remains vulnerable to Non-Line-of-Sight (NLOS) conditions; reflected paths can inflate range estimates and induce localization drift.

Temporal asynchrony further complicates fusion. UWB typically updates at 10–20 Hz, whereas many GPS configurations output at roughly 1 Hz. Accurate time-stamping and interpolation are therefore indispensable, yet in practice they are often imperfect, undermining the precise temporal alignment needed for high-fidelity multi-sensor fusion. Seamless indoor–outdoor tracking is another gap. Systems often specialize in either indoor (UWB) or outdoor (GPS) operation, and transitions are handled by manual handovers, simple interpolation without true fusion. What is missing are robust, context-aware weighting mechanisms that prioritize sensors dynamically based on environment features and signal-quality factors.

Positioning the present thesis within this body of work, the research advances a hybrid GPS–UWB–IMU framework designed to achieve robust indoor–outdoor continuity with asynchronous data fusion, sub-second alerting, and native Revit integration. By addressing gaps in practical deployment, geometric consistency, and proactive safety visualisation, the proposed system contributes directly to the development of field-ready frameworks that align localisation research with real-world safety needs (Soltanmohammadlou *et al.*, 2019; Pereira *et al.*, 2025; Li *et al.*, 2016).

Finally, BIM environment which is Revit integration remains shallow. Many safety solutions live in external dashboards or GIS tools disconnected from design and planning workflows. Native integration into platforms such as Autodesk Revit is limited by the absence of real-time data pipelines, the scripting overhead of tools like Dynamo or pyRevit, and the lack of built-in components for moving entities, motion trails, and time-based alerts. These gaps impede smooth coordination among safety monitoring, construction planning, and on-site operational decision-making.



3. METHODOLOGY

In this thesis, the methodology is designed around a hybrid Real-Time Location System (RTLS) that integrates UWB and GPS modules in Revit which is a Building Information Modeling (BIM) environment through the Autodesk Revit API. Figure 3.1 depicts the end-to-end pipeline adopted in this study, progressing from multi-sensor data acquisition to real-time visualization and managerial use. The workflow comprises four parallel preparation streams data collection, coordinate transformation, BIM model development, and Revit API integration that converge in a central Real-Time Tracking stage.

The experimental setup that validated the framework that had been proposed involves two distinct environments: indoor areas covered by UWB anchors, and outdoor spaces where GPS data is collected via smartphone applications. Workers (or test subjects) are equipped with wearable tags and mobile devices to simulate real-world movement. The collected raw data is first synchronized in time to account for the different sampling rates of UWB and GPS. Outlier detection is applied using Mahalanobis distance to eliminate abnormal values caused by multipath interference, non-line-of-sight conditions, or sensor drift. After this preprocessing stage, the data streams are fused using an Unscented Kalman Filter (UKF), chosen for its capacity to handle nonlinearities in position estimation. The fused positions are transformed into the UTM coordinate system and mapped dynamically within the BIM model. A custom pyRevit script is developed to visualize worker movements in Revit, creating a real-time digital twin of the site. This visualization supports safety analysis by allowing supervisors to identify proximity to hazards or unsafe zones. In summary, the methodology proceeds as follows: deploy UWB anchors and GPS modules, collect data from workers, apply preprocessing with outlier detection, fuse data streams via UKF, transform coordinates, and integrate into Revit for real-time visualization.

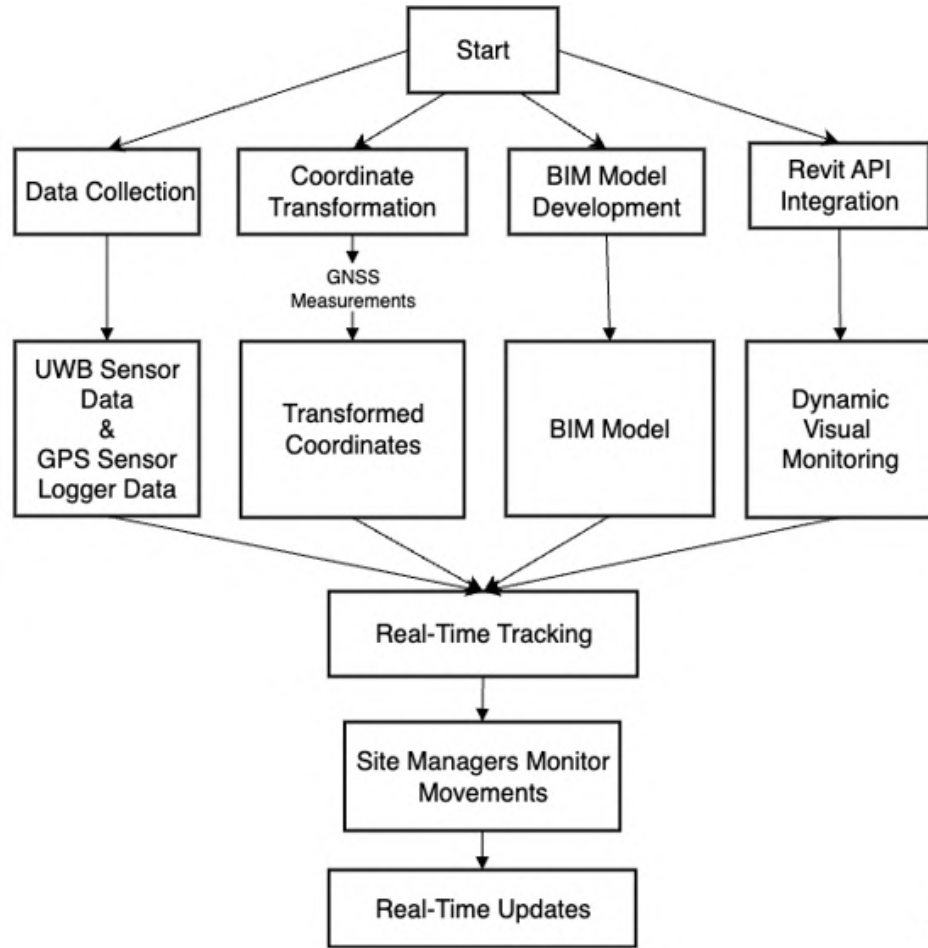


Figure 3.1. End-to-end pipeline: data collection, coordinate transformation, BIM model development, Revit API, and real-time monitoring.

3.1. Hypothesis

The central hypothesis of this research is that a hybrid UWB-GPS RTLS, when Construction sites mix obstructed outdoor areas with cluttered indoor spaces, where single-sensor RTLS solutions are brittle: smartphone-grade GPS often degrades to multi- meter errors in urban canyons or near structures (Zandbergen, 2009; Groves *et al.*, 2012; Moselhi *et al.*, 2020), while UWB provides high indoor precision under LOS but suffers in NLOS, body shadowing, and sub-optimal anchor geometry (Alarifi *et al.*, 2016; Jin and Sadeghpour, 2022; Shamsollahi *et al.*, 2024). Literature therefore recommends hybrid approaches and sensor fusion (He and Chan, 2016; Kaplan and Hegarty, 2017; He *et al.*, 2020).

A second gap is the limited systematic outlier handling in fielded applications. NLoS-multipath spikes can corrupt Kalman updates unless robust pre-fusion gating is used (He and Chan, 2016; Guo *et al.*, 2022; Adame *et al.*, 2023; Pan *et al.*, 2022).

Third, real-time RTLS outputs are often visualized on external dashboards, not natively inside the BIM environment despite clear benefits for proactive awareness when integrating live positions within BIM environments such as Revit model (Teizer, 2015; Teizer *et al.*, 2022; Omrany *et al.*, 2023; Tuhaise *et al.*, 2023; Pereira *et al.*, 2025).

GPS offers wide-area coverage but degrades in obstructed environments; UWB yields decimeter-level accuracy indoors under LOS yet degrades in NLOS (Alarifi *et al.*, 2016; Jin and Sadeghpour, 2022; Shamsollahi *et al.*, 2024; Zandbergen, 2009; Groves *et al.*, 2012).

Hybridization with Kalman-family fusion is widely recommended to combine complementary sensors and reduce errors and spikes (He and Chan, 2016; Kaplan and Hegarty, 2017; He *et al.*, 2020). The Unscented Kalman Filter (UKF) is well-suited to nonlinear measurement/motion models typical of UWB–GPS tracking and can weight sensors by their time-varying reliabilities (He and Chan, 2016; Kaplan and Hegarty, 2017).

Hypothesis H1a. In hybrid indoor–outdoor worker tracking, fusing UWB and GPS with an Unscented Kalman Filter produces significantly lower positioning error than GPS-only tracking.

Hypothesis H1b. In hybrid indoor–outdoor worker tracking, fusing UWB and GPS with an Unscented Kalman Filter produces significantly lower positioning error than UWB-only tracking.

Time-aligned UWB ranges and GPS coordinates are fused with a constant velocity UKF. UWB is vulnerable to NLOS/ multipath outliers that can destabilize Kalman

updates; reviews and field studies recommend pre-fusion statistical gating, including Mahalanobis distance checks, to reject implausible readings (He and Chan, 2016; Guo *et al.*, 2022; Adame *et al.*, 2023; Pan *et al.*, 2022; Alarifi *et al.*, 2016). Systematic, sensor aware gating remains under-reported in hybrid applications closing this gap should reduce spikes and improve stability. Controlled studies of hybrid mounted systems that integrate Mahalanobis (or closely related residual-based) outlier rejection within the fusion loop report 35–50%RMSE reductions compared with ungated fusion or single-sensor baselines (Li, Wang and Khoshelham, 2018; Naheem and Kim, 2022), with additional evidence of $\approx 27\%$ improvement in UWB/INS pedestrian tracking (Xu *et al.*, 2021). These findings motivate adopting a gate then only fuse in this thesis.

Hypothesis H2. Applying Mahalanobis-distance anomaly rejection before fusion attenuates NLOS/spurious measurements, yielding a more stable trajectory with lower error (fewer spikes) in the fused output.

The combination of (i) statistically justified outlier removal before fusion, (ii) documented reduction in spikes and tail errors, (iii) improved cross-track behavior, and (iv) smoother indoor–outdoor transitions demonstrates that Mahalanobis-distance anomaly rejection attenuates NLOS/spurious measurements and yields a more stable, lower-error fused trajectory.

Hypothesis H3. Among trajectories visualized natively in the Autodesk Revit model, the Mahalanobis-pre-filtered + UKF-fused trajectory will achieve lower positioning error, higher temporal continuity (fewer drops/spikes) than the non fused only Mahalanobis filtered data in Revit via pyRevit.

3.2. Research Design and Framework Overview

Prior studies have indicated that UWB tracking can improve construction site safety, and that linking UWB data with BIM enables better information flow and automated monitoring (ElQasaby *et al.*, 2022).

In order to achieve the stated objectives, this research adopts a design-and-experimentation methodology. In the beginning the overall architecture of the integrated safety tracking system was designed. The proposed system uses a hybrid RTLS which integrates UWB and GPS signal-based data to demonstrate a model to track dynamic subjects/objects such as workers and assets and integrates the location data into BIM model in order to accelerate the efficiency in management.

This research is organized into a series of interconnected phases, each addressing a specific component of the real-time worker tracking and BIM environment integration framework. The study focuses on fusing Ultra-Wideband (UWB) and Global Positioning System (GPS) data using an Unscented Kalman Filter (UKF) to enable accurate indoor-outdoor localization within a Building Information Modeling (BIM) environment.

The framework is implemented across multiple phases:

- (i) Sensor configuration,
- (ii) Synchronized data acquisition,
- (iii) Coordinate transformation,
- (iv) Pre-filtering and outlier rejection,
- (v) UKF-based fusion,
- (vi) Revit integration using Revit API and pyRevit scripting interface (Mapping and Visualization)
- (vii) Enscape Monitor

This modular architecture facilitates dynamic tracking and real-time feedback in a Revit-based digital twin. Framework that had been created is given in the Figure 3.2.

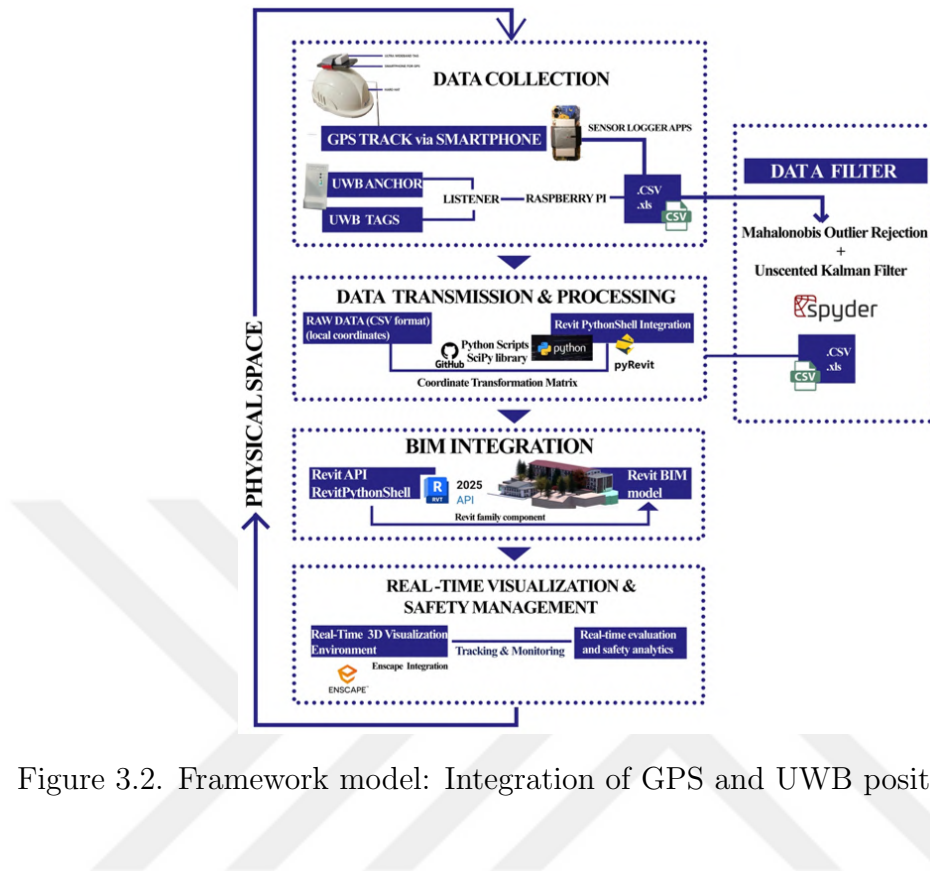


Figure 3.2. Framework model: Integration of GPS and UWB position data in revit.

Figure 3.2 illustrates the workflow and system integration proposed to enable real-time worker tracking through Ultra-Wideband (UWB) localization in a Building Information Modeling (BIM) environment which is in Revit software. The process begins with the Real-Time Location System (RTLS) based on UWB technology, which continuously records position data of mobile tags. These position readings are periodically exported in .csv format.

Using a custom Python script integrated into pyRevit, the location data is parsed and imported into the Revit which is BIM environment. This integration allows for the dynamic placement of virtual components representing tracked individuals within the digital twin model.

Once inside Revit, the data is visualized in real-time through Enscape, a virtual monitoring and rendering tool that supports immersive and up-to-date visualization. Enscape enables site managers or safety personnel to observe ongoing movements and spatial distributions of workers within the digital replica of the construction site.

The final output is a real-time tracking system, where movement data is continuously updated and rendered in the BIM environment, enhancing situational awareness and enabling proactive safety monitoring.

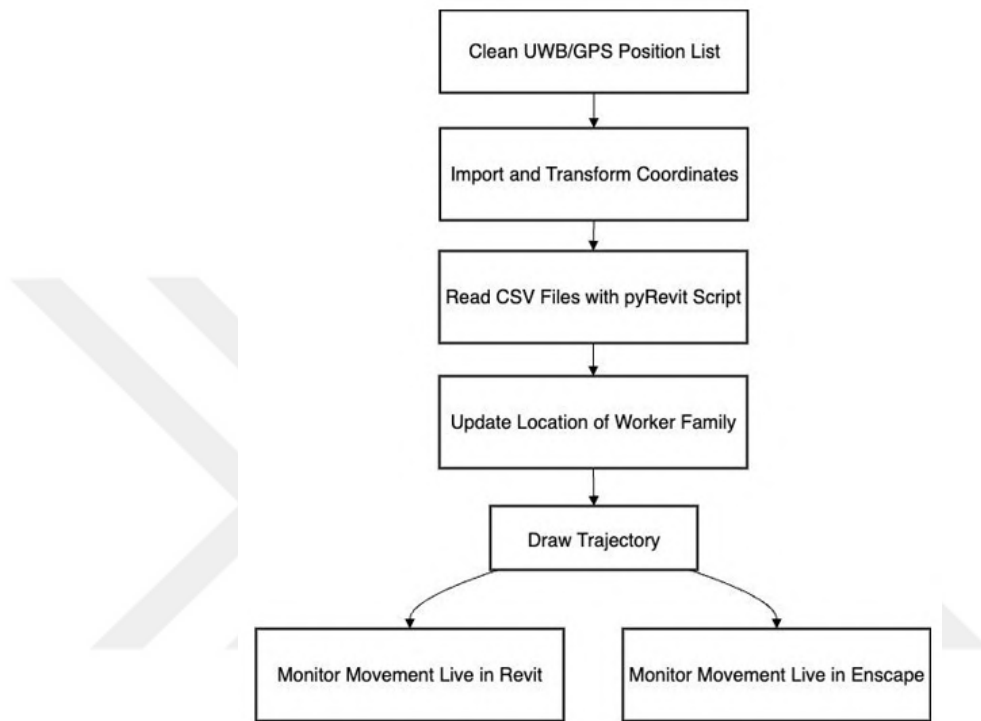


Figure 3.3. Revit integration RTLS tracking system workflow.

3.2.1. Sensor Configuration

The first stage is Sensor Configuration, this stage involves configuring the sensor systems required for dual-source localization. The Decawave MDEK1001 development kit, which incorporates the DWM1001 UWB modules, is used for indoor tracking. A set of UWB anchors is deployed in a tetrahedral layout to minimize Geometric Dilution of Precision (GDOP). These anchors are calibrated and checked.

For outdoor localization, smartphones were utilized which has GPS receivers (iOS-based devices) are utilized. Applications such as Geo Tracker, GPSTracks, and MyTrack are configured to log GPS traces with timestamp and metadata to serve both as ground truth and for comparative evaluation. Synchronization of time sources

across sensors is ensured by initializing all data recordings simultaneously and referencing timestamps to a common system clock.

3.2.2. Data Acquisition

For indoor and near-building environments where satellite signals are occluded, UWB technology provides high-precision localization. In this research indoor network system is designed with Ultra Wideband network system.

In this stage, synchronized Ultra-Wideband (UWB) and Global Positioning System (GPS) data were collected through a series of controlled experiments simulating real-time worker movement across indoor and outdoor environments. The trials were conducted in and around two key university buildings, the Engineering Faculty and the Data Science and Artificial Intelligence building chosen for their diverse architectural layouts and the presence of both open and enclosed spaces. These features allowed for the evaluation of signal behavior in line-of-sight (LOS) and non-line-of-sight (NLOS) conditions, as well as transitions between indoor and outdoor areas.

The UWB system, comprising Decawave DWM1001 tags and anchors, recorded local Cartesian coordinates (x , y , z) in real-time, with anchor positions calibrated in a known reference frame prior to data collection. Simultaneously, GPS data were recorded using iOS-based smartphone applications, including Geo Tracker, GPSTracks, and MyTrack. These applications captured geodetic coordinates latitude, longitude, and elevation with timestamps, allowing later alignment with the UWB dataset for sensor fusion.

To ensure data integrity, particular attention was given to the limitations of the iOS operating system, which imposes resource prioritization for background processes. It was observed that concurrent operation of multiple GPS logging applications on the same device led to time stamp compression and potential suppression of one app due to system-level resource conflicts.

In practice, this meant that only the most recently foregrounded app would log data reliably, while others would either stall or record incomplete tracks. To address this, a controlled data collection protocol was developed. GPS applications were executed sequentially or on separate devices to avoid conflicts and ensure consistent recording. All GPS coordinates, initially logged as in the WGS84 geodetic reference frame, were converted to the UTM (Universal Transverse Mercator) coordinate system to facilitate metric compatibility with the UWB's local Cartesian measurements. This transformation was essential for the coordinate alignment stage, enabling accurate integration and comparison between the two data sources in a unified spatial framework.

3.2.3. Outlier Rejection via Mahalanobis Distance

Mahalanobis distance is a statistical measure used in multivariate analysis to determine how far a point is from the center of a distribution, considering the correlations between variables through the covariance matrix (De Maesschalck, Jouan-Rimbaud and Massart, 2000). Unlike the Euclidean distance, which treats all dimensions equally and assumes independence among variables, the Mahalanobis distance scales the axes based on variance and removes correlations between variables, effectively standardizing the data space (De Maesschalck *et al.*, 2000).

Mathematically, for a data vector x , with a mean vector μ and covariance matrix Σ , the Mahalanobis distance $D_M(x)$ is defined as,

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}. \quad (3.1)$$

In many practical applications, especially in anomaly detection or statistical testing, the squared Mahalanobis distance $D_M^2(x)$ is used as a testing statistic value.

Mahalanobis distance produces elliptical contours aligned with the data distribution. This is due to its normalization of the feature space, allowing for rotation and scale-invariant distance measures (Gnanadesikan and Kettenring, 1972).

In RTLS data fusion, Mahalanobis-based elimination is essential for rejecting large range data measurements. For example, hybrid UWB/IMU or GPS/IMU systems detect bad range measurements by computing the Mahalanobis distance of each new reading from the predicted state and marking those beyond a threshold as outliers. Integrating UWB and GPS data in positioning systems, especially for applications like construction safety monitoring or worker tracking, brings significant challenges. Positioning data obtained from Ultra-Wideband (UWB) and Global Positioning System (GPS) technologies is often subject to irregularities due to environmental and technical factors. In UWB systems, multipath propagation and non-line-of-sight (NLOS) conditions can cause significant deviations in range measurements, especially in cluttered indoor environments (Chen *et al.*, 2020). Similarly, GPS-based localization is affected by weak satellite visibility, signal blockage, and multipath effects in urban canyons or near buildings and trees, leading to inconsistencies and measurement drifts (Groves, 2013; Kaplan and Hegarty, 2017). These result in sudden spikes in the estimated positions. These challenges necessitate robust outlier detection and filtering strategies like the Mahalanobis distance, due to the non-Gaussian errors, measurement noise, and signal distortions often present in real life conditions.

Mahalanobis method allows to adapt to the data's geometry and better detect outliers in multivariate, heterogenous distributions especially useful when movement patterns such as worker pattern tracking which follows non-spherical patterns. Mahalanobis distance provides a statistically rigorous approach to identify and filter out multivariate outliers. Unlike univariate methods, which examine each dimension (e.g., X, Y, Z) independently.

In the study conducted by Li *et al.* (2018), Mahalanobis distance-based outlier detection was employed to address two critical challenges in UWB/IMU sensor fusion: non-line-of-sight (NLOS) errors in UWB measurements and drift accumulation in IMU data. These error types, if left unfiltered, were shown to significantly bias the state estimation and degrade positioning performance. To overcome this, the authors integrated a Mahalanobis-based gating mechanism into their complementary Kalman filter

structure, enabling the filter to dynamically assess the statistical consistency of each UWB observation. If a measurement's Mahalanobis distance exceeded a pre-defined chi-square threshold, its impact was suppressed by inflating the associated covariance matrix rather than discarding the data completely. This adaptive soft rejection preserved potentially useful information while minimizing the influence of spurious readings. As a result, the proposed method significantly improved positioning accuracy. In scenarios with artificial noise and beacon-specific outliers, the method reduced the root-mean-square error (RMSE) by more than 50%, from 2.27 meters to 1.02 meters. Additionally, it maintained a smoother and more stable trajectory under dynamic motion conditions. These findings demonstrate that Mahalanobis distance-based filtering not only increases the robustness of sensor fusion but also maintains high localization precision in real-world environments with fluctuating signal quality.

Lin *et al.* (2018) proposed a UWB/IMU fusion framework for pedestrian tracking, using Mahalanobis distance to reject NLOS UWB measurements before update steps. Their results showed 35% improvement in position accuracy compared to baseline fusion without outlier rejection. Mahalanobis distance accounts for the covariance structure of the data and measures how far a point deviates from the multivariate mean, relative to the distribution's shape (Mahalanobis, 1936). This makes it particularly suitable for detecting observations that may appear plausible in individual dimensions but are inconsistent with the overall trajectory or motion pattern (Hadi *et al.*, 2009; De Maesschalck *et al.*, 2000).

For example, a sudden location reported by a UWB anchor might fall within an acceptable range on the X-axis but, when considered jointly with Y and Z, lies far from the typical movement path. By computing the squared Mahalanobis distance for each point with the given equation,

$$D^2(x) = (x - \mu)^T \Sigma^{-1} (x - \mu), \quad (3.2)$$

where Σ is covariance matrix and μ mean vector according to state vectors.

Table 3.1. Summary of key concerns in GPS-UWB fusion necessitating mahalanobis use.

Concern	Description	Mahalanobis Role
Non-Gaussian noise	NLOS and multipath distort error distributions	Provides robust statistical outlier detection
Heterogeneous errors	GPS and UWB have different error patterns	Normalizes error across sensors
Real-time fusion sensitivity	Kalman filters are sensitive to bad data	Acts as gating to prevent divergence
Safety-critical environments	Worker mislocation can be dangerous	Improves reliability and trust in fusion system
Indoor-outdoor transitions	Data variance spikes across boundaries	Flags and filters sudden anomalies

Removing such outliers before Kalman or particle filtering prevents gross errors in the estimated worker position and greatly improves safety monitoring reliability.

In sensor fusion systems where multiple positioning technologies (such as Ultra-Wideband and GPS) are integrated, ensuring the reliability of incoming measurements is essential. Both GPS and UWB measurements are susceptible to noise, distortions, and anomalies due to environmental factors such as multipath propagation, obstructions, and indoor-outdoor transitions. If not handled properly, these anomalies can introduce bias and instability into the sensor fusion process, significantly degrading the overall localization accuracy. To address this, a pre-filtering stage is incorporated into the system using a Mahalanobis-distance-based outlier detection mechanism. This approach serves to identify and remove (or down-weight) data points that deviate significantly from the expected multivariate distribution of previous observations. Unlike simple Euclidean distance, which only considers individual dimensions independently, the Mahalanobis distance accounts for the covariance structure of the dataset, enabling robust multivariate outlier detection. The integration of Mahalanobis filtering directly precedes the data fusion stage. The filtered GPS and UWB data are passed to the UKF

only if they fall within acceptable Mahalanobis bounds. This dual-stage filtering and fusion approach is illustrated in Figure 3.4, which outlines the entire data processing pipeline. Data filtering process sequence shown in the Figure.

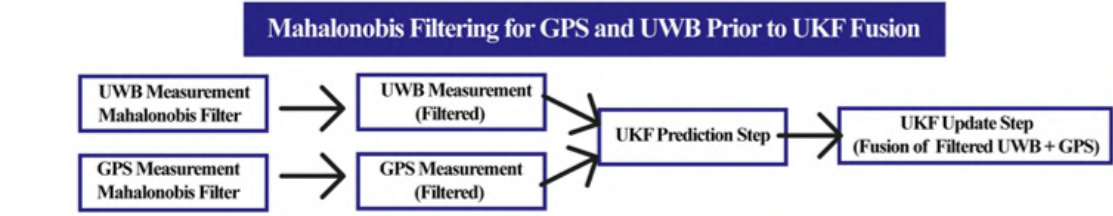


Figure 3.4. UWB and GPS data filtering and fusion.

Mahalanobis distance outlier elimination is complementary method used with Kalman Filter fusion. While the Unscented Kalman Filter is effective in estimating states from noisy measurements, it inherently assumes that the incoming data is not contaminated with extreme outliers. If such anomalies are present in the measurements, they can significantly skew the predicted state and degrade the performance of the filter. By applying Mahalanobis filtering beforehand, the system ensures that the UKF operates under more favorable conditions. This enhances both the stability and accuracy of the fusion, leading to more reliable tracking of worker locations in the BIM environment.

3.2.4. Data Fusion Algorithm: Unscented Kalman Filter

The Kalman filter serves as a foundational framework for a recursive estimation technique applied to stochastic, typically linear, dynamic systems. At its core, the Standard Kalman filtering algorithm operates on the principle that the true internal state of a system often not directly observable can be inferred from a series of noisy, indirect measurements taken over time.

Mathematical model of the system aims to combine the predictions from the two integrated network system with observational data, the filter continuously refines

the estimations, producing an optimal approximation of the system state under the assumptions of linearity and Gaussian noise. Gaussian noise refers to a range of a statistical randomness where the probability of the randomness follows a normal distribution.

However, in real-world sensor networks particularly those involving multi-sensor information fusion, communication losses, and environmental uncertainties these ideal conditions are frequently violated. The assumption of linearity breaks down in systems with nonlinear dynamics, such as object tracking or pedestrian navigation, and the noise encountered in practical settings often deviates from Gaussian behavior due to multipath interference, obstacle occlusion, and sensor drift. These give rise to non-Gaussian noise, which may be heavy-tailed, skewed, or contain abrupt outliers that bias estimation.

Furthermore, many sensor networks operate under degraded or time-varying conditions such as random sensor activation due to energy saving, noisy communication channels, sensor dropouts and uncertain noise covariances, which vary over time due to environment or system reconfiguration.

These challenges shows that the standard Kalman Filter is inadequate for robust performance. In practical, the Kalman filter uses a two-step process: prediction and update. During the prediction step, the filter projects the current state estimate and error covariance forward in time using the known system dynamics. In the update step, it incorporates new measurements to correct the predicted state, weighing both the prior estimate and the new data according to their respective uncertainties. This dynamic feedback mechanism allows the filter to minimize the mean squared error of the estimated state.

Raw RTLS data often contain noise, dropouts, or outliers especially in challenging environments, so filtering is essential. Noise distortions are unpredictable fluctuations that interfere with accurate measurement caused by electronic interference, environ-

mental conditions, sensor imperfections, communication delays between sensors and human errors in sensor location data. Noises distort the actual values of the sensor position data that UWB tags and GPS trackers in smartphones. Noises should be filtered in order to employ controlled decision-making. Filtering is the process that eliminates the true signals from the noise in order to improve accuracy of the measurements and provide more stable evaluations. In basic terms, Kalman Filter works like a smart balancing tool preventing data from distortions. Kalman filter system continuously compares two sources of information: what to expect to happen based on past data and motion patterns and what we actually measure using sensors. If the sensor readings are reliable when there is minimum random noise in specified confidence interval, it provides more weight to that measurement. But if the prediction from the model is uncertain or less reliable, the filter adjusts itself to rely more on the measurement data. This adjustment is controlled by Kalman Gain, which changes dynamically based on the situation.

This flexible balancing makes the Kalman Filter especially effective for combining data from different sources, such as GPS and UWB (Ultra-Wideband) systems. It supports to create more accurate and stable tracking, even in situations where data might be inconsistent, like when a person moves in and out of buildings or when the environment causes sensor signals to become weak or blocked. As a result, the Kalman filter enhances the precision and reliability of positioning systems used in real-world applications like navigation, construction monitoring, and safety tracking systems (Gentner and Retscher, 2012; Kalyan *et al.*, 2021).

In effect, the Kalman filter assumes linear dynamics and Gaussian noise to compute the posterior in closed form. More complex filters perform a similar Bayesian update numerically. In practice, Kalman-based methods are favored in construction RTLS due to their low latency and real-time capability. For example, Feng *et al.* (2018) fused UWB and IMU data using an Unscented Kalman Filter to improve indoor positioning accuracy, effectively performing a Bayesian fusion of two sensor streams.

Kalman filtering serves as a foundational technique in sensor fusion for positioning systems, operating optimally under assumptions of linear dynamics and Gaussian noise (Zhou *et al.*, 2017; Wang *et al.*, 2020; Evensen, 2003). Kalman filters have been applied in notable construction RTLS studies. A significant research Saidi and Teizer (2010) filtered raw UWB tracking data with Robust Kalman Filter to mitigate distortions and noise.

The Kalman Filter minimizes the mean squared error by adjusting predictions based on measurement reliability and process model confidence. It is particularly effective in tracking applications, where sensor data may be intermittent or noisy.

Fronckova and Prazak (2020) explain that the Kalman filter works in a repeating cycle made up of two main steps. First, it makes a guess about the current state of the system like a person's location or speed in the case of worker tracking based on the last known information and how things are expected to move. This step is called the prediction. Then, it takes a new measurement such as a distance reading from a UWB sensor or GPS sensor logger and adjusts that earlier guess to make it more accurate. This second step is called the update or correction. By going through this cycle again and again, the filter helps reduce the effect of sudden errors or "spikes" in the data, like when a sensor reading is off due to interference. At the same time, it keeps the results realistic by following how things usually behave physically, like objects moving in a smooth path.

Despite its strengths in real-time estimation and noise reduction, Kalman filtering presents several limitations when applied in dynamic, noisy, and non-linear construction environments. The classical Kalman Filter (KF) assumes that both the system dynamics and the observation models are linear and that the noise is Gaussian. However, real-world construction sites rarely conform to these assumptions. For example, in the study by Cheng and Teizer (2013), a Kalman filter was used to estimate the trajectories of workers in a dynamic construction zone. While effective in controlled conditions, the performance of the Kalman Filter deteriorated under blockages, where

multiple workers overlapped or moved in and out of the sensor line-of-sight. The authors noted that these disruptions introduced non-Gaussian errors due to multipath effects and sensor shadowing, which the standard KF could not handle adequately. To address this, some researchers have turned to Extended Kalman Filters (EKF) or Unscented Kalman Filters (UKF), which offer better handling of non-linearities. In the work of Kim *et al.* (2020), an EKF was implemented to integrate UWB and IMU data for a tunneling project. While the EKF outperformed the basic KF in orientation and velocity estimation, it still required meticulous tuning of the process and measurement noise covariance matrices. Li *et al.* (2018) conducted a benchmarking study comparing Kalman filter variants in a hybrid indoor-outdoor context. They demonstrated that EKF provided acceptable performance in clear LOS conditions but exhibited significant drift in NLOS areas.

The Unscented Kalman Filter (UKF) handles nonlinear models more accurately without explicit linearization. UKF overcomes non-linearity issues by sampling sigma points around the mean and propagating them through the nonlinear function $f(x)$. The UKF is designed for nonlinear systems and provides better accuracy than the Extended Kalman Filter (EKF) by using a deterministic sampling approach called the Unscented Transform (UT).

A sigma point is a carefully chosen sample from a distribution (usually Gaussian), designed to capture the mean and covariance of that distribution with minimal sample size. Instead of linearizing a nonlinear function (as in EKF), the UKF sends these sigma points through the nonlinear function and then reconstructs the transformed mean and covariance from the output.

According to Al-Okby *et al.* (2024), the UKF accurately captures the posterior mean and covariance to third-order accuracy for any nonlinearity (assuming Gaussian noise), whereas the EKF only achieves first-order accuracy. This makes the UKF particularly suitable for real-time location systems (RTLS) in environments where the motion model or measurement functions (e.g., UWB range equations) are highly non-

linear. For example, when fusing UWB data with inertial sensors, the UKF can better handle drift, delay, and bias. Studies by Yang *et al.* (2017) and Wan and van der Merwe (2000) confirm that the UKF generally outperforms the EKF when the system dynamics significantly deviate from linearity.

The key benefit of the UKF is that it avoids explicit linearization and instead captures the effects of nonlinearity by propagating the sigma points through the actual nonlinear functions. According to Al-Okby *et al.* (2024), the UKF accurately captures the posterior mean and covariance to third-order accuracy for any nonlinearity (assuming Gaussian noise), whereas the EKF only achieves first-order accuracy. This makes the UKF particularly suitable for real-time location systems (RTLS) in environments where the motion model or measurement functions (e.g., UWB range equations) are highly nonlinear. For example, when fusing UWB data with inertial sensors, the UKF can better handle drift, delay, and bias. Studies by Yang *et al.* (2017) and Wan and van der Merwe (2000) confirm that the UKF generally outperforms the EKF when the system dynamics significantly deviate from linearity.

In essence, the UKF provides a robust alternative to the EKF for nonlinear estimation tasks by:

- Using deterministic sigma points to represent uncertainty
- Propagating these points through nonlinear models
- Achieving higher accuracy, especially in dynamic and complex scenarios

As noted by Sutton (1999) and later empirical works, the UKF is now a common choice in construction RTLS applications, where positioning systems must cope with noisy, nonlinear sensor data and frequent NLoS conditions.

To integrate and enhance the localization accuracy of worker tracking across indoor and outdoor environments, this study develops an Unscented Kalman Filter (UKF)-based sensor fusion framework to integrate position data from Ultra-Wideband

(UWB) and Global Positioning System (GPS) measurements. The UKF is selected over classical Kalman or Extended Kalman Filters due to its ability to handle nonlinear systems without requiring Jacobian linearization (Wan and van der Merwe, 2000), making it suitable for environments with high spatial variability such as indoor-outdoor transitions. This is especially important in hybrid localization scenarios involving frequent transitions between environments with differing sensor characteristics.

Table 3.2. Comparison for Kalman filter and advanced Kalman filters.

Aspect	Kalman Filter (KF)	Extended Kalman Filter (KF)	Unscented Kalman Filter
Model Assumptions	Assumes linear system dynamics and observation models with Gaussian noise (Kalman, 1960).	Handles nonlinear functions via Jacobian-based linearization (Julier and Uhlmann, 2004; Wikipedia, 2024a).	Handles nonlinearities using the unscented transform; no linearization required (Julier and Uhlmann, 2004; Inside GNSS, 2013).
Optimality	Statistically optimal under linear-Gaussian assumptions (Kalman, 1960).	Suboptimal in nonlinear systems; may diverge with poor linearization	More robust in nonlinear environments; provides improved estimation in non-Gaussian systems (Julier and Uhlmann, 2004).

Table 3.2 Comparison for Kalman filter and advanced Kalman filters. (cont.)

Aspect	Kalman Filter (KF)	Extended Kalman Filter (KF)	Unscented Kalman Filter
Accuracy	Accurate mean and covariance propagation for linear systems (Kalman, 1960).	Limited to first-order approximation accuracy due to linearization (Julier and Uhlmann, 2004).	Achieves second-order accuracy for general distributions and third-order for Gaussians (Julier and Uhlmann, 2004; Inside GNSS, 2013)
Derivative Need	No need for model derivatives; closed-form equations used (Kalman, 1960).	Requires symbolic or numerical Jacobians at each step	Derivative-free; suitable for unknown or highly nonlinear models (Julier and Uhlmann, 2004).
Computational Cost	Low computational complexity; efficient matrix operations (Kalman, 1960).	Moderate; dominated by Jacobian computations and matrix operations	Comparable to EKF; dominated by sigma point calculations (Inside GNSS, 2013).
Ease of Use	Simple to implement and well-established (Kalman, 1960).	More complex due to need for model linearization	Easier for models where derivatives are unavailable; tuning parameters are intuitive (Julier and Uhlmann, 2004; Inside GNSS, 2013).

Table 3.2 Comparison for Kalman filter and advanced Kalman filters. (cont.)

Aspect	Kalman Filter (KF)	Extended Kalman Filter (KF)	Unscented Kalman Filter
Best Use Case	Best suited for linear systems or when computational simplicity is critical (Kalman, 1960).	Preferred for somehow nonlinear problems (e.g., inertial navigation with small angles) (Wikipedia, 2024a).	Ideal for highly nonlinear, noisy, or complex systems such as UAVs, autonomous robots, and GPS/INS fusion (Julier and Uhlmann, 2004; Inside GNSS, 2013).

Table 3.3. Justification for unscented Kalman filter.

Criterion	Justification for Unscented Kalman Filter
Nonlinearity	System involves fusing GPS (nonlinear error characteristics, weak in indoor zones) and UWB (subject to multipath/NLOS errors). These sensors do not conform to linear Gaussian assumptions, so UKF which handles nonlinear models without linearization
No Derivatives Needed	Working with real sensor data (not just simulations), possibly in a vague sensor fusion setting where deriving Jacobians may be infeasible or error-prone. UKF enhances this by using the unscented transform, which is more robust and easier to implement.
Post-processed but Designed for Real-time	Framework aims at real-time visualization in a digital twin. UKF is commonly used in real-time tracking systems for its balance of speed, accuracy, and robustness.

Table 3.3 Justification for unscented Kalman filter. (cont.)

Measurement Noise and Uncertainty	System deals with sensor fusion under measurement uncertainty (e.g., GPS loss, UWB multipath). UKF handles this by more accurately capturing higher-order moments of the distribution (especially important with NLOS UWB errors).
Literature Support	Multiple high-impact papers (e.g., Julier and Uhlmann, 2004; Sharma <i>et al.</i> , 2022) support UKF for indoor-outdoor sensor fusion, construction safety, and real-time estimation

Classical Kalman filtering introduces additional limitations. The standard filter presumes linear system dynamics and Gaussian process/measurement noise assumptions rarely satisfied on sites where worker motion includes abrupt turns, stops, and accelerations, and where measurement errors under obstruction or NLOS are skewed or heavy-tailed.

Robustness and adaptivity are also underutilized. Many implementations fix process-noise covariances despite rapidly changing dynamics. More suitable strategies such as adaptive Kalman filtering informed by real-time residuals, or Unscented Kalman Filters (UKF) combined with Mahalanobis-distance gating to reject outliers remain promising but are seldom applied in construction safety deployments.

For this master’s thesis focused on real-time worker tracking via UWB and GPS data integration in a BIM environment, the Unscented Kalman Filter (UKF) is the most appropriate filter due to its ability to handle nonlinear dynamics, derivative-free implementation, and robustness to sensor noise. When coupled with Mahalanobis distance outlier rejection, it provides a reliable and scalable estimation framework suitable for construction safety monitoring systems.

Fused positioning data, resulting from the integration of UWB and GPS measurements via the Unscented Kalman Filter (UKF), were exported as timestamped CSV

files. These files were subsequently imported into Autodesk Revit through a custom-developed Python script implemented using the pyRevit scripting interface. Within this script, a pre-calculated affine transformation matrix was applied to align the fused global coordinates with the Revit model’s local coordinate system, ensuring spatial consistency between real-world and BIM-referenced locations. Once transformed, worker locations were represented in the BIM environment using adaptive component families or symbolic avatars, which were dynamically updated at each timestamp using Revit API methods such as “LocationPoint, Transform”, and “Transaction”. The system supported real-time visualization, incorporating features such as motion trails to depict recent worker trajectories.

This integration allowed for spatial-temporal monitoring directly within the BIM model, enabling intuitive interpretation of worker movements and enhancing the situational awareness of site supervisors and safety personnel.

3.3. System Architecture

This research implements a real-time worker tracking system by fusing Ultra-Wideband (UWB) positioning hardware and Global Positioning System (GPS) data within a Building Information Modelling (BIM) digital twin environment such as Revit and additional virtual tools in order to monitor and map worker locations in real-time. The overall approach is to deploy UWB system network which consists of anchors and tags on-site for precise indoor localization, utilize smartphone-based GPS sensor logger outdoor positioning, and integrate both data streams through a Kalman filter to produce a continuous location estimation. All position estimations are transformed into a common coordinate frame aligned with the BIM model for visualization. The system integration with BIM model is achieved through Autodesk Revit API via communicating it with the pyRevit plugin. This integration provides updates the BIM model in real-time, allowing safety managers to visualize worker movements within the digital environment. The following subsections will detail the hardware setup, coordinate transformations, sensor fusion algorithm design, data filtering techniques, BIM

integration of the location data, and virtual monitor of the location data of the worker along with the rationale behind the methodological choice.

The system consists of: a network wearable/attachable sensors such as UWB sensor tags and a GPS smartphone attached to a construction hard hat or such wearable unit on each worker; a network of UWB anchors will be installed in the indoor test area; a central computing unit that will run data for the data fusion algorithm and sound reduction and BIM environment visualization; and the BIM model (in Revit environment) that is adjusted with the application programme interface by pyRevit plug in that provides communication with location data through python scripts and updates the model. For indoor worker data UWB ranging data that is obtained from UWB system from listener sensor to the raspberry PI, GPS coordinates are transmitted via a wireless network to the central unit. The central unit will process the data inputs including running the Kalman filter for sensor fusion and then the updated estimated positions will adjust the BIM model in real time. The system was designed in a modular way so that the indoor and outdoor tracking subsystems integrate with the common main system interface. At the design stage of the system architecture, customized hardware and software tools are chosen, for instance, for the UWB RTLS platform Qorvo's Decawave UWB network system was chosen as the indoor set up and ensuring it can output data at the required frequency; deciding to use Autodesk Revit API with Python integration to transform and adjust the BIM model; and choosing between visualization options according to GPU limitations.

3.3.1. Sensing Layer: UWB and GPS Module Deployment

UWB technology was selected because it delivers sub-meter accuracy and robust performance in cluttered environments. In practice UWB tags are estimated that they will be worn by workers, and fixed UWB anchors are going to be installed on the jobsite to cover the area of interest. UWB technology was selected because it provides sub-meter accuracy and robust performance in cluttered construction sites where multipath propagation and non-line-of-sight (NLoS) conditions are common. In

practice, UWB tags are expected to be worn by workers, while fixed UWB anchors are strategically installed around the jobsite to ensure adequate coverage and minimize geometric dilution of precision. Compared to Bluetooth Low Energy (BLE) or passive RFID, UWB offers superior accuracy and resilience to interference, making it well-suited for high-risk areas that demand precise monitoring of worker location.

Complementing UWB, GPS modules are deployed to extend tracking capability to outdoor zones of the construction site. GPS is particularly effective in open-sky conditions, where it can provide three to ten meter of accuracy using standard receivers. With advanced techniques such as Real-Time Kinematic (RTK) corrections, centimeter-level positioning can be achieved, which is critical for tasks such as surveying, site layout, and equipment tracking.

However, GPS performance degrades under partial obstructions, near tall structures, or within semi-enclosed environments. This limitation reinforces the need for hybrid deployment, where GPS provides wide-area coverage outdoors, and UWB ensures high accuracy and continuity indoors or in obstructed spaces.

By combining UWB and GPS at the sensing layer, the system creates a complementary architecture that leverages the strengths of each technology. UWB anchors and tags handle precise localization in complex site interiors, while GPS modules extend monitoring beyond building perimeters. Together, they form the foundational layer for data fusion, ensuring that worker movements are captured seamlessly across both indoor and outdoor contexts.

3.3.2. Communication Layer: Data Transmission and Logging

This layer is responsible for the collection, time stamping, and storage of raw sensor data. Indoor positioning systems that use ultra-wideband (UWB) have become increasingly common because they can measure distances very accurately by calculating how long a signal takes to travel (Barral *et al.*, 2019). This makes them more

dependable than systems that only rely on signal strength. Ultra-Wideband (UWB) is a radio communication technology characterized by extremely wide fractional bandwidth typically greater than 500 MHz within the 3.1-10.6 GHz frequency range (Che *et al.*, 2023). UWB systems transmit ultra-short pulses, often in the nanosecond range, which allows for highly accurate time-of-flight (ToF) ranging. These impulse-like signals are emitted across a broad spectrum, and receivers measure the exact time delay to multiple fixed anchors to calculate the tag’s position. This high temporal resolution enables UWB-based RTLS to achieve centimeter-level accuracy in distance estimation. For example, UWB modules such as the Qorvo DW1000 have demonstrated positional accuracy as fine as 10 cm under line-of-sight (LoS) conditions (Che *et al.*, 2023). In addition to accuracy, UWB has a low average power spectral density, making it ideal for low-power operation. Tags can operate for extended periods on small batteries an essential feature for wearable safety applications.

UWB systems benefit from a large bandwidth and short-duration pulses, which together deliver sub-decimeter ranging precision in experimental and commercial deployments. As noted by Che *et al.* (2023), UWB’s high temporal resolution “reduces multipath fading” and enables extremely accurate time-of-arrival (ToA) measurements.

UWB transmits energy in rapid, short bursts at a low duty cycle, which keeps average power transmission well below regulatory noise thresholds. This minimizes interference with other wireless systems and significantly extends battery life (Che *et al.*, 2023). The impulse nature of UWB allows the system to distinguish the direct path from reflected signals. According to Che *et al.* (2023), this makes UWB naturally resistant to multipath distortions that degrade other narrowband or RSS-based systems. Indoor positioning data from Ultra-Wideband (UWB) tags (Decawave DWM1001 MDEK1001 modules) and outdoor positioning data from smartphone GPS applications (e.g., Geo Tracker) were acquired at a predefined sampling frequency (e.g., 10 Hz). Data were transmitted to a Raspberry Pi-based data acquisition system using Bluetooth Low Energy (BLE) or UART protocol, depending on the sensor configuration.

3.3.3. Processing Layer: Kalman Filter and Coordinate Alignment

The processing layer handled data cleaning, sensor fusion, and coordinate transformation. Initially, the raw position data were time-aligned and subjected to outlier detection using 3D Mahalanobis distance, which filtered out abnormal measurements typically caused by NLOS conditions or signal instability.

Next, the filtered data were passed through an Unscented Kalman Filter (UKF) for sensor fusion. The UKF was chosen due to its robustness in handling non-linear motion models and measurement noise. It produced a sequence of fused position vectors by optimally combining the UWB and GPS data at each timestamp. Following fusion, the resulting global coordinates were mapped into the BIM model's local coordinate system using a pre-calibrated affine transformation matrix. This transformation ensured that real-world position data corresponded accurately to the spatial framework within Revit.

3.3.4. Application Layer: Revit Integration and Visualization

In the final layer, the fused and transformed position data were imported into the BIM environment using a custom pyRevit script. This script dynamically updated symbolic worker avatars, modeled as adaptive component, within the Revit environment. Revit API functions such as 'Element Transform Utils. Move Element', "Transaction-Manager", and Filtered Element Collector were used to update worker positions at each time step. Furthermore, real-time visual feedback mechanisms were embedded into the BIM interface. These included motion trails to visualize movement paths and zone-based alerts, which changed the visual state of the worker avatar (e.g., color-coded warnings) when entering predefined hazard areas. This integrated application layer allowed safety supervisors and planners to monitor real-time site activity directly within the BIM model, thereby enhancing decision-making and facilitating proactive safety management.

3.4. Hardware and Sensor Setup

In this phase, the system is built according to the design. The UWB indoor positioning system is set up first: anchors are installed in specified locations that enable the most convenient readings preventing the NLOS affects. Calibration is done to align the UWB local coordinate system with GPS global coordinate system. For UWB layout, a common and effective strategy involves placing four anchors in a rectangular grid: for example , four anchors at site corners and if it is necessary additional extra anchors could be also attached on sides to increase the line of sight coverage and the minimize the occlusions. This configuration improves the anchor geometry for Time of Flight (ToF) or Two-Way Ranging (TWR) calculations and increases the accuracy of the positioning, especially in complex indoor-outdoor transition situations.

In academic implementations, grid-based anchor setups have been empirically shown to reduce localization errors by improving Dilution of Precision (DoP) metrics. For example, in the work of Corbalan *et al.* (2023), a rectangular layout using four anchors achieved sub-30cm Root Mean Square Error (RMSE) in Line-of-Sight (LOS) conditions, while expanded configurations with 6-8 anchors improved NLOS resilience by 18% due to increased angular diversity between anchors (Corbalan *et al.*, 2023). Similarly, Montorsi *et al.* (2012) demonstrated that optimal UWB placement minimizing Geometric Dilution of Precision (GDOP) directly impacts TWR-based positioning accuracy, especially under partial obstruction scenarios.

Moreover, LOS conditions are critical for high-precision UWB operation. However, when the direct path between the transmitter and receiver is obstructed, a Non-Line-of-Sight (NLOS) scenario occurs. In such cases, the first path signal is either reflected or delayed due to objects like walls, machinery, or workers themselves. This obstruction causes a positive bias in the ToF measurement because the reflected signal travels a longer distance than the true geometric path. Studies have shown that NLOS conditions can introduce ranging errors exceeding 1 meter, particularly in dense indoor or construction environments where metallic surfaces or concrete interfere with radio

propagation (Montorsi *et al.*, 2012; Tomasin *et al.*, 2023). Therefore, to mitigate this problem, anchor grids are not only geometrically intuitive but also statistically advantageous for reducing the impact of measurement uncertainty and enabling efficient system calibration in hybrid GPS-UWB systems.

3.4.1. Selection of Anchors and Tags

Each worker was equipped with a lightweight, battery powered UWB tag. These tags periodically emitted UWB pulses captured by the network of fixed anchors. The system uses the Two-Way Ranging (TWR) algorithm, which measures the round trip time of UWB pulses to compute distances between the tag and the anchors computed as,

$$d = \frac{c \cdot RTT}{2}, \quad (3.3)$$

where d is the distance, c is the speed of the light, and RTT is the measured round-trip time.

This method does not require synchronization between anchors, making it particularly well-suited for decentralized or large-scale sites like construction sites. In this research the Decawave DWM1001 module was used for real-time TWR based indoor localization. As noted by Cho *et al.* (2024), maximizing the covered area between the anchors and site area providing the optimal 3D configuration reducing the positioning errors, especially in environments with partial NLOS conditions.

3.4.2. Placement and Calibration of Anchors

The localization system of the experiment which held in this master thesis was deployed across two separate buildings: Building A is Engineering Faculty of the Bogazici University and Building B is Data Science and Artificial Intelligence Building in the South Campus of the Bogazici University, they are located approximately 25 to 26 me-

ters apart on a controlled experiment site. The outdoor area between the two buildings was chosen deliberately in order to maintain the simulations which will represent the real construction environments where workers frequently transit between indoor and outdoor zones. This selection aimed to evaluate the tracking ability of the system to perform robustly across the heterogeneous localization environments, where the Non-Line-of-Sight (NLOS) effects are common due to the building geometries, obstructions, occlusions and signal interference (Ventura *et al.*, 2023). In construction safety related research, the integration of indoor and outdoor positioning technologies is widely recognized as a necessary condition to achieve nearly accurate and reliable localization in dynamic site activities (Li *et al.*, 2020; Deng *et al.*, 2021). Thus, incorporating both outdoor and indoor environment within the experimental setup not only reflects real-world complexity but also ensures the robustness of the system design under mixed signal propagation conditions and supports continuous tracking across indoor and outdoor (Gupta *et al.*, 2010). Indoors, UWB systems excel in providing high-precision tracking due to controlled reflections and short distances, while GPS often fails due to signal attenuation and obstruction. As highlighted by Jiménez and Seco (2021), even minor occlusions can introduce substantial positioning errors in UWB-based systems, especially under low redundancy and in transitional regions between LOS and NLOS conditions. Similarly, Hu *et al.* (2023) emphasized the need to evaluate RTLS frameworks in topologically diverse environments to address multipath interference and enable robust transitions between tracking modalities.

Each building entrance was instrumented with four UWB anchors as arranging rectangular configuration and those anchor nodes serve as fixed referential points for trilateration. This geometric layout was carefully selected to optimize the contextual coverage and geometric diversity, thereby improving the accuracy and stability of location estimates. These anchors function as fixed reference nodes within the UWB system, providing known position in a local coordinate frame. In both buildings, two anchors were placed at the same lower elevation level, while the remaining two were installed at higher elevation points. This non-coplanar anchor configuration was specifically chosen to improve 3D localization accuracy, as it helps mitigate depth ambiguity

and reduces the impact of geometric dilution of precision (GDOP). The spatial distribution of anchors was optimized by ensuring they surrounded the expected tracking volume and were positioned near the rectangular entrances of both buildings to guarantee better visibility for the mobile tag.

All UWB anchors were geodetically surveyed to establish their coordinates within a unified global coordinate system. Two geodetic reference marker points **PA** and **PB** were selected in the outdoor area and observed using a dual-frequency GNSS antenna for 3 hours in static mode. These measurements were processed using the Canadian CSRS-PPP service to obtain accurate ITRF20 coordinates projected into UTM Zone 35. These reference points were deliberately selected such that they were visible from inside the buildings, allowing the total station to sight them for interior positioning. These control points, whose coordinates were precisely determined via static GNSS observations, served as the geodetic reference frame for the localization experiment. These reference points also deliberately selected such that they were visible from inside of the buildings, allowing the total station to sight them for interior positioning related operations. With the control points known, a total station resection was performed in each building. The station was oriented by measuring angles and distances to **PA** and **PB**, and then used to determine the interior locations of all four UWB anchors. This procedure resulted in a consistent transformation between local UWB coordinates and global UTM-based coordinates for each building. For applications requiring planar mapping, such as construction, engineering surveying, or Building Information Modeling (BIM) integration, the Universal Transverse Mercator (UTM) projection is commonly adopted due to its ability to provide metric linearity and low distortion over small to moderate geographic extents. UTM divides the globe into 60 longitudinal zones, each spanning 6° of longitude, and projects geodetic coordinates (latitude and longitude) onto a Cartesian plane using the transverse Mercator projection method (Leick, Rapoport and Tatarnikov, 2015). The zone employed in this study is UTM Zone 35N, which covers the region of the experimental setup and is frequently used in geospatial and engineering analyses across Türkiye. The use of UTM Zone 35N has been demonstrated in multiple academic studies. For example,

El-Ashmawy *et al.* (2016) evaluated the deformation effects when transforming local projections in Egypt into UTM Zone 35N, confirming its suitability for preserving spatial accuracy in construction-scale projects. Similarly, Şahin and Kurum (2020) applied UTM 35N to align GIS and BIM models for infrastructure management in Turkey, underlining its practicality in geospatial-BIM data integration.

The resection method is a classical geodetic approach used to determine the position of an unknown point by measuring angles and distances to at least two fixed, known points (Wolf *et al.*, 2012). In construction and indoor positioning scenarios, this technique is particularly valuable for aligning local anchor coordinates with a global or site-wide reference frame. By leveraging a well-established controlled points derived from GNSS receivers, resection ensures accurate transformation and integration of the UWB anchor positions into a consistent spatial reference system.

3.4.3. GPS Data Acquisition and Smartphone Integration

For outdoor localization, GPS data was collected using smartphone which was available IOS based smartphone applications, namely Geo Tracker and Mytracks. These applications offer accessible and reliable methods for recording location trajectories by utilizing the Global Navigation Satellite System (GNSS) receiver in the smartphone.

For example in the experiment during a controlled walk around the perimeter of the two university buildings Engineering Faculty and the Data Science and Artificial Intelligence Institute, both applications were activated simultaneously but after the timestamp data compresses crowds the data it suppresses the other application. The data was logged at a sampling interval of 1 second. Geo Tracker produced a GPX file containing detailed position fixes, velocity, and accuracy metadata, which could be directly imported into GIS software. The logged route closely followed the actual pedestrian pathway, with an average reported horizontal accuracy of ± 4.8 meters.

3.4.4. Anchor-Tag Communication and Frequencies

The location of the mobile tags which are attached to the hard hat that are worn by workers is computed through trilateration method automatically with the Decawave system. The mobile tags, attached to workers' hard hats, exchange ultra-wideband (UWB) packets with fixed anchors using Decawave-based transceivers compliant with IEEE 802.15.4a/4z. Ranging is performed via two-way time-of-flight, and the tag position is then estimated by multilateration (trilateration in 2D; at least four anchors for 3D). All anchors were surveyed and registered to a global site frame (e.g., ENU/UTM), so the location engine outputs absolute coordinates rather than local ones. All anchors were surveyed to establish their coordinates in a global coordinate system.

3.5. Sensor Data Collection and Handling

The UWB network system computes tag-anchor distances via Two-Way Ranging or TdoA and derives 3D positions using geometric trilateration. To track worker positions, we equipped each test subject with a helmet-mounted UWB tag (Decawave MDEK1001) and a smartphone GPS logger running MyTracks. The helmet was selected after preliminary trials indicated that it offered the best compromise between signal visibility and minimal human body occlusion, compared to wrist or belt mounting positions. The helmet-mounted setup also allowed the reflective prism to be aligned with the UWB tag, ensuring a common reference point for both optical and radio-frequency measurements.



Figure 3.5. UWB tag and IOS smartphone implementation to hardhat.



Figure 3.6. iOS smartphone UWB tag attachment.

The subjects of the experiment which represent the construction workers walked through a predetermined path covering both indoor environments inside the two buildings and the outdoor zone between them. This experimental path was physically marked with tape lines to ensure consistent data collection across multiple trials and was also modeled digitally in the BIM environment to enable later comparison. Both GPS and UWB data were recorded simultaneously, and timestamp synchronization ensured accurate alignment of sensor readings.

3.5.1. GPS Data Logging and Pre-Processing

For outdoor localization, GPS data was collected using commercially available Android-based smartphone applications, namely Geo Tracker and My Tracks. These applications offer accessible and reliable methods for recording location trajectories by utilizing the smartphone's built-in Global Navigation Satellite System (GNSS) receiver. For example, during a controlled walk around the perimeter of the university campus (50 m), both applications were activated simultaneously on identical devices. The data was logged at a sampling interval of 1 second. Geo Tracker produced a GPX file containing detailed position fixes, velocity, and accuracy metadata, which could be directly imported into GIS software. The logged route closely followed the actual pedestrian pathway, with an average reported horizontal accuracy of ± 3.8 meters. My Tracks, while similarly generating GPX logs, recorded fewer metadata fields (e.g., no real-time DOP value or fix quality), and its exported path showed occasional drift in sharp corners likely due to lower filtering or smoothing.

Geo Tracker application was utilized. Geo Tracker is a widely used GPS logging tool that records high-frequency position and motion data using the device’s integrated GNSS sensor. The application provides detailed outputs, including latitude, longitude, distance traveled, altitude, speed, acceleration, heading, and both horizontal and vertical accuracy metrics. These attributes were essential for evaluating real-world outdoor performance and for later comparison with fused GPS-UWB data.

The GPS tracks were exported in structured CSV format and included high-resolution time-stamped measurements (see Figure below). This data was synchronized with the UWB logging system through timestamp alignment, enabling seamless fusion via the Unscented Kalman Filter (UKF). Additionally, the application’s user-friendly interface, as shown in the figure, allowed real-time monitoring of location, speed, and elevation, assisting in field data verification during trials.

3.5.2. Time Synchronization/Frequency and Data Integrity

In real-time sensor fusion systems, maintaining temporal consistency across multiple data sources is critical. This study involves integrating two distinct localization technologies Ultra-Wideband (UWB) and Global Positioning System (GPS) each with its own sampling frequency and data logging format. UWB modules (Qorvo DWM1001) typically sample at 10 Hz, while GPS logging via smartphone-based applications such as Geo Tracker is generally limited to 1 Hz. to address this temporal mismatch and ensure data integrity, the following synchronization strategy was employed:

The entire experimental process was monitored and recorded using iOS-based timestamped video applications. A video recording was initiated simultaneously with UWB and GPS logging to create a common temporal reference. This allowed visual tracking of the subject’s movement and served as a cross-verification tool for system timestamps. The video timestamps acted as a high-resolution master clock during post-processing.

All devices UWB anchors/tags and GPS smartphones were time-synchronized prior to each session. UWB devices used the internal synchronized clock provided by the DWM1001 firmware, while smartphones synchronized to network time. Although these clocks operate independently, alignment was achieved by referencing a known simultaneous event (e.g., the start of movement) across all logs and video frames.

Due to the higher sampling rate of the UWB (10 Hz), temporal interpolation was performed to match the GPS sampling timeline. GPS data points were interpolated linearly to estimate positions at intermediate timestamps. This enabled proper fusion using the Unscented Kalman Filter (UKF), where GPS was incorporated at lower frequency updates, and UWB measurements continuously updated the state estimate at higher resolution.

Data streams were monitored for anomalies such as missing times, timestamp drift, or sampling jitter. Custom Python scripts were developed to flag out-of-order timestamps or time gaps. Redundant logging was enabled on both the smartphone and the UWB base station, ensuring no critical data was lost. In the case of inconsistent logs, timestamps from the synchronized video were used to re-align data chronologically.

This multi-level synchronization framework combining real-time timestamps, video recording, and sampling frequency adaptation ensured that fused localization data maintained high temporal fidelity, enabling reliable and continuous indoor-outdoor tracking.

3.6. Coordinate System Transformations

In this study, the objective is to integrate coordinate data obtained from both outdoor (GPS-based) and indoor (UWB-based) positioning systems into a unified BIM coordinate framework. A fundamental challenge in integrating heterogeneous positioning technologies such as satellite-based GPS and indoor Ultra-Wideband (UWB) systems is the disparity in their underlying coordinate reference frames. This study

bridges global and local positioning by transforming data between geodetic reference frames (e.g., ITRF, WGS84, and UTM) and the local Cartesian coordinate systems natively used by UWB hardware devices.

In order to align the GPS coordinate system and UWB referential coordinate system two different points were specified in the outdoor area between the two buildings in the University Campus: Engineering Faculty and Data Science Institute. To combine GPS and UWB location data in a single unified spatial environment and BIM coordinate system for visualization and tracking, a multi-stage coordinate transformation processes were essentially developed. The transformation workflow ensures that all positioning data whether from global GNSS or local UWB network is expressed in the same coordinate frame as the BIM model, enabling direct overlay of tracked positions onto the digital building model. This section outlines the detailed workflow adopted in this study and explains the rationale behind each step with support from academic literature.

3.6.1. Geodetic GPS to ECEF (Earth-Centered Earth-Fixed) Conversion

The first transformation converts the geodetic coordinates (latitude, longitude, ellipsoidal height) obtained from GNSS observations into a Cartesian Earth-Centered Earth-Fixed (ECEF) coordinate system. The Earth-Centered, Earth-Fixed (ECEF) coordinate system is a globally recognized Cartesian coordinate system with its origin located at the Earth's center of mass. It is fixed relative to the rotating Earth, ensuring that points on the Earth's surface retain fixed coordinates over time.

Geodetic coordinate systems such as; WG84, ITRF, UTM are global reference frames tied to ellipsoid and rotational axis. They enable absolute positioning anywhere on Earth by considering the shape, center of mass, orientation, and tectonic motion over time of the planet. Their utility lies in their universality: they allow positioning anywhere on the globe with respect to a common reference datum. A datum is a mathematical model that defines the size and shape of the Earth and provides a frame

of reference for measuring locations on its surface. In geodesy and geographic information systems (GIS), a datum forms the foundation of coordinate systems, allowing consistent and accurate representation of positions (Leick *et al.*, 2015). The GNSS reference points (PA, PB) were surveyed using dual-frequency static receivers and post-processed via CSRS-PPP. The final outputs were delivered in the International Terrestrial Reference Frame (ITRF20), epoch 2025.4, in geodetic format (latitude ϕ , longitude λ , ellipsoidal height h). One such system is the International Terrestrial Reference Frame (ITRF), maintained by the International Earth Rotation and Reference Systems Service (IERS) with observational data contributions from the International GNSS Service (IGS). The ITRF is a high-precision, three-dimensional, and time-aware geodetic reference frame that models tectonic plate motion and Earth deformation with millimeter-level accuracy (Altamimi and Métivier, 2021). The most recent realization, ITRF2020, provides updated station coordinates and velocities referenced to epoch 2025.4 and aligns with satellite navigation systems through global network solutions.

Although many references cite the WGS84 ellipsoid for geodetic transformations, we adopt the ITRF20 ellipsoid parameters which are closely aligned with WGS84 geodesic coordinate system for converting geodetic positions to Cartesian ECEF coordinates. The standard conversion equations are

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}, \quad (3.4)$$

$$X = (N + h) \cos(\phi) \cos(\lambda), \quad (3.5)$$

$$Y = (N + h) \cos(\phi) \sin(\lambda), \quad (3.6)$$

$$Z = (N(1 - e^2) + h) \sin(\phi), \quad (3.7)$$

where ϕ is latitude, λ is longitude, h is ellipsoidal height, a is the semi-major axis (in WG84 ellipsoid is 6378137 m), e is the first eccentricity of the ellipsoid ($6.69437999014 \times 10^{-3}$), and $N(\phi)$ is the prime vertical radius of curvature. Then, obtained GPS track location data values from the GPS logger in IOS based smartphone were converted into ECEF coordinates (X, Y, Z). After converting all relevant points to ECEF, they

were reduced to a local scale by subtracting a reference origin by calculating the vector differences for each location data point had calculated.

For each location data point distance differences according to ECEF coordinate systems were measured as

$$\Delta X = X - X_0; \Delta Y = Y - Y_0; \Delta Z = Z - Z_0 \quad (3.8)$$

where is the point (X_0, Y_0, Z_0) is the ECEF coordinate of the origin point and X, Y, Z represents the converted location data points from GPS logger to ECEF. Such distance differentials are essential inputs for converting global ECEF coordinates into the local ENU frame. These vector offsets provide the preliminary step required for accurate localization in the ENU coordinate system.

3.6.2. ECEF to ENU (East-North-Up)

The East-North-Up (ENU) coordinate system, by contrast, is a local Cartesian coordinate system defined relative to a specific reference point on the surface of the Earth. It provides a “flat Earth” approximation in the vicinity of that point, making it particularly suitable for applications requiring localized precision, such as indoor-outdoor positioning, sensor fusion, and Building Information Modeling (BIM) integration (Wang *et al.*, 2019). When the global geodetic control points which are, the reference points labeled as PA and PB were converted from geodetic coordinates (latitude, longitude, ellipsoidal height) into the Earth-Centered Earth-Fixed (ECEF) Cartesian coordinate system, all positional data acquired from both Ultra-Wideband (UWB) and GPS sensors were subsequently transformed into a local tangent plane coordinate system known as East-North-Up (ENU). GPS devices commonly output coordinates in ECEF format, this frame is difficult to interpret for local applications. Consequently, it is standard practice to transform all global positional data into a local ENU coordinate frame anchored to a known reference point on the site. And referred to as ϕ_0, λ_0, h_0 corresponding to latitude, longitude, and ellipsoidal height. This transformation enables the direct comparison and integration of heterogeneous positioning data within a unified local reference system.

3.6.3. UWB Local Coordinate System to ENU

The UWB system provides tag positions relative to its own origin designated by the eight anchors from A1 to A8 as the UWB local origin in the setup. ENU coordinates of that origin by surveying 8 anchors with GNSS total station was established. The translation matrix is arranged according to the given output data of the UWB coordinates. The translation was consisted of adding the ENU coordinates of the anchors. The rotation is necessitated if the UWB anchor orientation does not align with true east/north. In this master thesis UWB anchor layout was oriented such that the local X-Y and Z axis were roughly aligned with the east north directions on site to minimize the rotation offsets. However, to achieve higher precision alignment, two referential points were specified and measured the aforementioned BIM-A and BIM-B. Both were surveyed by GNSS and marked in the site one near each building entrance. After the transformation matrices were generated each UWB data points were transformed into ENU coordinate system values. The scale parameter of the transformation matrix is nearly 1.000 so the transformation matrices minimized into only shift and rotation.

Essentially main data obtained from the UWB network system which is in the UWB local coordinate system rotated by a small angle to align the axis with the ENU frame, and translated it so that the anchor A8 was arranged as the origin point local (0; 0; 0) coinciding with the ENU coordinates. After this step, any UWB tag position can be expressed in the global ENU frame consistent with the GPS data.

3.6.4. ENU to Revit Coordinate Reference Plane and Mapping

After, arranging the transformation of UWB and GPS data into the same ENU coordinate frame. Coordinates which are in the ENU frame were mapped into the coordinate system of the BIM model in Revit. The BIM Revit model was georeferenced to the site using the same two control points (BIM-A and BIM-B) during the model transformation setup. BIM model base point such that corresponding to point PA's location, and the Revit model was set to the north in order to coincide with the

north direction. In this research the X axis of the Revit model was aligned to East direction and Y axis to the North direction. Thus, in this research thesis the ENU coordinates could be used directly as BIM model coordinates with a simple translation arranging the anchor A8 as the origin point. In practice, origin of the BIM model coordinate arranged as the same place as anchor 8(A8) and the system adjusted to the rotation angle. This ensured that the model's orientation and scale match the real world coordinate system using the UTM projection, Zone 35N, to preserve the metric distances. The use of the project coordinate system like UTM 35N for the site to ensure the minimal distortions over the scale of the project and it is common in construction surveying.

After verifying the alignment process the surveyed anchor points in the BIM model matched the known coordinates within a few centimeters, we proceeded to use this coordinate space for all visualization. In summary, through the above steps, both GPS and UWB data streams were converted into the local coordinate space of the 3D BIM environment in Revit platform. This aims to enable an nearly accurate overlay of the tracked worker position onto the digital building geometry without manual adjustments because already the BIM local 3D space had been aligned to the coordinate space of the obtained data.

3.6.5. UWB Reference Frame and Revit Referential Frame

Autodesk Revit employs a dual-origin coordinate system consisting of the Project Internal Origin and two key user-definable reference points: the Project Base Point (PBP) and the Survey Point (SP). Understanding these reference elements is critical for aligning external coordinate systems such as those derived from GPS or UWB measurements with the internal coordinate space of the BIM model. In order to ensure that real-time positioning data (from GPS and UWB sensors) could be accurately represented within the BIM model environment, it was necessary to establish a consistent spatial relationship between the external coordinate reference system and Revit's internal coordinate system.

This transformation was implemented by defining a common origin point in both systems and aligning all incoming coordinates relative to this origin. Specifically, the East-North-Up (ENU) coordinate system commonly used in GPS and UWB tracking was initialized with a user-defined reference point, which was also manually placed as the project base point in Revit. This approach simplified the transformation by allowing translation-only alignment, avoiding the need for more complex affine or Helmert transformations when rotational alignment was not required.

P_{local} is presented mathematically as a simple translation,

$$P_{local} = TP_{sensor} = P_{sensor} - P_{origin}, \quad (3.9)$$

where in Equation (3.9) $P_{sensor} = (x_s, y_s, z_s)$ represents the fused position data from GPS / UWB in ENU coordinates, $P_{origin} = (x_o, y_o, z_o)$ is the origin that defined in the sensor coordinate system, $P_{local} = (x_s - x_o, y_s - y_o, z_s - z_o)$ is the translated coordinate to align with Revit.

This translated coordinate P_{local} is then directly used as the placement point in Revit model space assuming both coordinate systems are axis aligned. A reference origin point was determined during calibration and applied as both the zero point in the ENU system and the project base point in Revit. Sensor positions were translated into the local model space by subtracting the origin point, effectively aligning coordinate systems via translation. Translated coordinates were used to update worker locations in the Revit model in real time via Revit API scripting through pyRevit. 3D Helmert Transformation Equation was used in order to align the obtained data into Revit referential coordinate system. The 7-parameter Helmert transformation maps a point from one 3D coordinate system (e.g., GPS/ENU) into another Revit transformed data points for Revit space are calculated as

$$X' = s.R.X + T, \quad (3.10)$$

where $X \in R^3$ is source point (GPS / ENU), $X' \in R^3$ is transformed point (for Revit space), s is uniform scale factor, $R \in R^{3 \times 3}$ is Rotation matrix (orthonormal), and $T \in R^3$ is translation vector.

3.7. Mahalanobis Distance-Based

To enhance the reliability of the real-time localization system, separate Mahalanobis distance-based outlier rejection is applied independently to the GPS and UWB data streams prior to their integration in the Unscented Kalman Filter (UKF). This sensor-specific preprocessing stage allows the system to account for each sensor's unique noise characteristics before fusion, ensuring that only valid, statistically consistent measurements are used for state estimation. Let the following denote the relevant quantities at time $step k$; $z_k \in R^n$: the actual measurement vector from the GPS or UWB position data z_k^{GPS} ; $z_k^{UWB} \in R^n$: the actual measurement vector from the GPS or UWB position data at time k $\hat{z}_k \in R^n$: the predicted measurement obtained from the UKF prediction step, $\hat{z}_k^{GPS}$; $\hat{z}_k^{UWB} \in R^n$: the predicted measurements obtained based on prior sensor-specific previous filtered values, $S_k \in R^{n \times n}$: the innovation covariance matrix, representing the uncertainty of the predicted measurement, S_k^{GPS} , $S_k^{UWB} \in R^{n \times n}$: the innovation covariance matrices, D_M^2 the squared Mahalanobis distance. The Mahalanobis distance between z_k and $\hat{z}_k$ is defined as

$$D_M^2 = (z_k - \hat{z}_k)^T S_k^{-1} (z_k - \hat{z}_k). \quad (3.11)$$

where z_k is measurement vector and $\hat{z}_k$ is predicted measurement. For each sensor type, the squared Mahalanobis distance is calculated separately as,

$$D_M^{2,GPS} = (z_k^{GPS} - \hat{z}_k^{GPS})^T (S_k^{GPS})^{-1} (z_k^{GPS} - \hat{z}_k^{GPS}), \quad (3.12)$$

$$D_M^{2,UWB} = (z_k^{UWB} - \hat{z}_k^{UWB})^T (S_k^{UWB})^{-1} (z_k^{UWB} - \hat{z}_k^{UWB}), \quad (3.13)$$

where each term corresponds to the sensor-specific measurement and its predicted value.

This distance metric measures how far the actual observation deviates from the expected measurement, normalized by the predicted uncertainty. Since D_M^2 follows a chi-squared distribution with degrees of freedom under the assumption of Gaussian noise, it can be statistically thresholded/limited as follows,

$$D_M^2 > X_{\alpha,n}^2, \quad (3.14)$$

$$D_M^2 > X_{\alpha,n}^2; s \in GPS, UWB, \quad (3.15)$$

where the measurement is considered an outlier at confidence level.

The chi-squared threshold $X_{\alpha,n}^2$ defines a region of statistical consistency and it is the upper critical value of the chi-squared distribution at confidence level.

The parameter α defines the significance level of the statistical test, and $1-\alpha$ represents the confidence level. Choosing an appropriate confidence level is critical, as it directly influences the filter's sensitivity to outliers. Lower thresholds (associated with lower confidence levels) are more aggressive in rejecting deviations, while higher thresholds are more conservative.

The Table 3.4 below summarizes typical confidence levels and their corresponding characteristics in the context of real-time sensor fusion,

Table 3.4. Confidence level determination effect on outlier rejection.

Confidence Level ($1-\alpha$)	Description	Effect on Outlier Rejection
90% ($\alpha=0.10$)	Permissive, low threshold	Aggressively rejects deviations; higher risk of false positives (rejecting good data)
95% ($\alpha=0.05$)	Balanced	Commonly used for moderate sensitivity in real-time tracking applications
99% ($\alpha=0.01$)	Conservative, high threshold	Only clear outliers are rejected; risk of retaining borderline errors
99.9% ($\alpha=0.001$)	Highly conservative (very high threshold)	Minimizes false positives; suitable for safety critical systems but may retain erroneous data

- Pre-filtered data ensures the UKF receives reliable inputs, reducing the risk of filter divergence and large ranged estimation errors due to spurious data.

- Reliability of the each position data quantified by measurement noise covariance $R_{(k)}$ becomes more meaningful when extreme outliers have been dropped.
- Outlier elimination before the data fusion prevents cross-contamination where an anomaly in one data position type does not distort data from other.

Each point was analyzed for its stepwise distance from the immediately previous position, and those that deviated by more than 10 cm (0.10 m) in 3D space were marked as outliers. For each data point **step distances are calculated as**

$$StepDistance = \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2 + (Z_t - Z_{t-1})^2}. \quad (3.16)$$

where this value is **greater than 0.10 m**; so the point is **considered as outlier** or else it is accepted.

Empirical evidence supporting this methodology is provided by Dwek *et al.* (2020), who proposed an anchor-specific Mahalanobis filtering scheme for UWB localization. Their findings demonstrate that applying outlier rejection before multilateration or sensor fusion significantly improves localization accuracy even under challenging industrial conditions with high outlier prevalence. In their experiments, the use of sensor-specific Mahalanobis filtering reduced the third quartile 2D localization error from over 3 meters to less than 10 centimeters.

This study adopts a similar principle by implementing sensor-wise outlier filtering prior to fusion, thus ensuring that only statistically consistent, high-confidence measurements influence the UKF update. This two-step architecture filter first, then fuse enhances the robustness, accuracy, and stability of the final position estimates, particularly in environments with mixed visibility and signal conditions.

3.8. Unscented Kalman Filter-Based Sensor Fusion

Kalman Filter was implemented to improve the smoothness of the location data and eliminate the any high distortions and noises. This recursive Bayesian estimator method merges the readings and weigh them by estimated noise. In this system, the

filter ingests UWB position datas and GPS data. Kalman filtering is commonly used in indoor location tracking and has been shown to decrease the distortions and enhances the data.

To combine the indoor UWB data and the outdoor GPS data into one coherent trajectory, Unscented Kalman Filter had been chosen to apply in order to minimize the noise and the distortions of the data and enhance continuity, accuracy, and reliability of the localization system. In this framework that had been generated in this research for the integration of GPS and UWB data implementation of Unscented Kalman Filter as a real-time sensor fusion algorithm had been decided as a method which provides integrity. By design, the filter will produce an improved position estimate by weighting the inputs from UWB and GPS according to their expected accuracy (noise) and updating continuously as new readings arrive. The UKF is a recursive Bayesian estimator that provides superior performance over classical Kalman or Extended Kalman Filters in nonlinear systems, as it does not require Jacobian matrix computation. Instead, it propagates a set of deterministically selected sigma points through the nonlinear system dynamics to accurately approximate the mean and covariance of the posterior state distribution (Wan and van der Merwe, 2000). When GPS accuracy degrades indoors, the UKF prioritizes UWB data. Conversely, in open outdoor environments where UWB signal coverage diminishes, GPS data is favored. The state vector includes both position and velocity, enabling prediction even when one data stream is temporarily unavailable.

Prior research shows that Unscented Kalman filtering can significantly reduce positioning error in both UWB and GNSS tracking scenarios. For example, Han *et al.* (2024) demonstrated that an Extended Kalman Filter improved indoor UWB localization accuracy by over 50% compared to raw trilateration results. Prior research has demonstrated the effectiveness of Kalman-based fusion in localization. For instance, Han *et al.* (2024) showed that using an Extended Kalman Filter (EKF) improved indoor UWB localization accuracy by over 50% compared to raw trilateration outputs. Building upon such findings, the present work implements the UKF, which further

enhances performance by offering more accurate estimates under nonlinear conditions and measurement uncertainty. By design, the UKF weights inputs based on their statistical reliability, enabling optimal integration of data from heterogeneous localization technologies.

3.8.1. State Vector Design and Motion Model

The unscented Kalman Filter is formulated in a discrete-time nonlinear state-space model:

State Vector, the unscented Kalman Filter is formulated in a discrete-time nonlinear state-space model State Vector is computed as

$$\hat{x}_K = [X_K Y_K Z_K \nu_{X,K} \nu_{Y,K} \nu_{Z,K}]^T, \quad (3.17)$$

where x_k, y_k and z_k represent positions in 3D space, and $\nu_{X,K} \nu_{Y,K} \nu_{Z,K}$ represent the velocity in each direction and k is the current timestep. Measurement vector is computed as

$$Z_K = [x_K^{\text{GPS}} y_K^{\text{GPS}} z_K^{\text{GPS}}]^T, \quad (3.18)$$

which result depends on the optimized sensor value at time k .

A core contribution of this system is its adaptive measurement selection strategy, which determines whether to prioritize UWB or GPS data at each timestep. The localization system is modeled in a discrete-time nonlinear state space framework. The core assumption is that worker motion follows a nearly constant velocity model, and that measurements from GPS and UWB data provide partial observations of the state. In this study, there is no active control input (i.e., the worker moves freely), so the control vector $u_k=0$.

However, motion is implicitly modeled in the process function. Measurement Model At each time step, the system receives measurements either from UWB anchors or from GPS trackers. Each network system, provides 3D position information in different coordinate frames. Prior to fusion, GPS data (originally obtained in WG84 geodetic format is transformed to a local Cartesian System which is UTM to en-

sure consistency with the UWB frame and Revit model alignment. The expected measurement is expressed as

$$z_k = h(x_k) + \nu_k. \quad (3.19)$$

where measurement function $h(\cdot)$ relates the predicted state and ν_k represents noise vector.

3.8.2. Predicitons and Update Equations

Generation of the sigma points are calculated as

$$X_{k-1}^{(i)} = \text{Sigma Points}(\hat{x}_{k-1}, P_{k-1}), i = 0, 1, \dots, 2n, \quad (3.20)$$

where $X_{k-1}^{(i)}$: is a set of $2n + 1$ points generated from the prior mean $\hat{x}_{k-1}$ and P_{k-1} represents a possible version of the each point in the state of the system. A set of sigma points $X^{(i)}$ are generated around the current state estimate $\hat{x}_{k-1}$ using the state covariance P_{k-1} . For an n -dimensional state, $2n + 1$ sigma points. Propagation of sigma points through the process model is shown as,

$$X_{k|k-1}^{(i)} = f(x_{k-1}^{(i)}, u_{k-1}), \quad (3.21)$$

where process model function $f(\cdot)$ defines how the system evolves over time as;

$$x_k = x_{k-1} + \nu_{x,k-1} \cdot \Delta t, \quad (3.22)$$

where transition model function $f(\cdot)$ defines how the state evolves over time assuming constant velocity between time steps,

$$x_k = f(x_{(k-1)}) + w_k, \quad (3.23)$$

$$f(x_{k-1}) = \begin{bmatrix} (x_{k-1} + \nu_{x,k-1} \cdot \Delta t) \\ (y_{k-1} + \nu_{y,k-1} \cdot \Delta t) \\ (z_{k-1} + \nu_{z,k-1} \cdot \Delta t) \\ \nu_{x,k-1} \\ \nu_{y,k-1} \\ \nu_{z,k-1} \end{bmatrix}, \quad (3.24)$$

where Δt elapsed time between steps (measured from timestamps).

Predicted mean values and covariance is computed to reach updated values. $x_{k|k-1}$ predicted state mean (before measurement update) is calculated as

$$\hat{a}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} X_{k|k-1}^{(i)}, \quad (3.25)$$

and $P_{k|k-1}$ predicted state covariance is calculated as,

$$P_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} [X_{k|k-1}^{(i)} - \hat{x}_{k|k-1}] [X_{k|k-1}^{(i)} - \hat{x}_{k|k-1}]^T + Q_k, \quad (3.26)$$

where P_k is the state covariance matrix at time k ; represents uncertainty in the current state estimation, Q_k is the process noise covariance, representing modeling uncertainty due to the unmodelled accelerations or abrupt changes in direction, especially during indoor-outdoor transitions, $W_m^{(i)}$ and $W_c^{(i)}$ are the weights that used to compute the mean and the covariance from the sigma points.

Transformation of predicted sigma points through measurement function updated as

$$Z_k^{(i)} = h(X_{k|k-1}^{(i)}), \quad (3.27)$$

where measurement model $h(\cdot)$ transforms sigma points from the state space into measurement space. While updating the UWB data measurement model is,

$$h(X) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ since UWB network system provides the positions but not the}$$

velocity. Predicted measurement mean and covariance are computed as ,

$$\hat{z}_k = \sum_{i=0}^{2n} W_m^{(i)} Z_k^{(i)}, \quad (3.28)$$

$$Sk = \sum_{i=0}^{2n} W_c^{(i)} [Z_k^{(i)} - \hat{z}_k] [Z_k^{(i)} - \hat{z}_k]^T + R_k, \quad (3.29)$$

where Sk is the covariance of predicted measurement, it measures the uncertainty in the predicted observation. If Sk is large, the measurement is unreliable ; if it is small it means observations are reliable to observe. Computation of the cross covariance is expressed as

$$P_{xz} = \sum_{i=0}^{2n} W_C^{(i)} [X_{k|k-1}^{(i)} - \hat{x}_{k|k-1}^{(i)}] [Z_k^{(i)} - \hat{z}_k]^T, \quad (3.30)$$

which represents the covariance between the predicted state and the predicted measurement, this provides the how changes in the state affect the expected measurement. If this covariance rate is high, it means that knowing the measurement gives reliable information about the current state. The Kalman Gain is calculated as

$$K_k = P_{xz} S_k^{(-1)}, \quad (3.31)$$

$$K(k) = \frac{\text{How strongly the state and the measurements are related}}{\text{How uncertain the measurement is}}. \quad (3.32)$$

which determines how much weight the filter gives to the new sensor measurement z_k relative to the prior state estimation $\hat{x}_{k|k-1}$.

3.8.3. Noise Covariance Estimation for GPS and UWB

Update location estimation with Kalman Gain matrix is calculated as

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_k), \quad (3.33)$$

where K_k is small model is trustable. In the indoors GPS data is unreliable, so covariance of the predicted measurement S_k^{GPS} is high so the system trusts the data. In the outdoor localization UWB has non line of sight so S_k^{UWB} increases and system trusts GPS more. In summary K_k kalman gain

$$K_k = P_{xz} S_k^{-1}, \quad (3.34)$$

is the mathematically optimal solution for fusing prediction and observation states when both have uncertainty.

3.9. BIM Environment Integration Using Revit API and pyRevit

This section describes the methodology employed to integrate real-time worker location data, obtained via sensor fusion of Ultra-Wideband (UWB) and Global Positioning System (GPS) signals, into a Building Information Modeling (BIM) environment using Autodesk Revit. The integration is facilitated through the use of the

Revit API (Application Programming Interface) and custom Python scripts developed in pyRevit, a lightweight IronPython-based automation layer built on top of Revit. The objective of this integration is to construct a dynamic digital twin capable of visualizing spatial behaviors and responding to safety-critical events in real-time. This includes worker tracking, alert visualization, hazardous zone mapping, and feedback for decision-making in safety management. The methodology follows a multi-stage workflow comprising coordinate transformation, adaptive object representation, real-time data interfacing, and visual feedback mechanisms.

3.9.1. 3D Revit Model Detail

The BIM environment was achieved through in Autodesk Revit 2024 to digitally replicate the geometry and semantics of the experimental pilot buildings and environment used in this study in order to enhance pyRevit related applications. The 3D model in Revit, includes architectural and spatial elements such as topography, building walls, floors, ceilings, corridors, staircases, windows, doors, areas, and control zones, which were semantically segmented to reflect real-world functional roles in the construction site. Specific emphasis was placed on the spatial classification of zones including rooms, hazard-prone areas, and circulation paths to enable location-based querying and visualization of worker behavior and safety-related concerns. The 3D Revit model was developed using native Revit families to accurately represent the physical and functional characteristics of the experimental environment. These predefined component categories allowed for structured modeling and facilitated parametric customization within the Revit environment. Key components included are explained:

- Rooms: Used to define enclosed spatial zones, enabling spatial tagging, occupancy analysis, and semantic classification for real-time location tracking.
- Floors and Ceilings: Modeled to represent structural boundaries and circulation planes, supporting vertical segmentation and aiding in the identification of movement surfaces for trajectory analysis.

- Grids and Reference Planes: Implemented as alignment aids to ensure geometric accuracy, particularly during sensor placement, anchor calibration, and georeferencing procedures.
- Generic Models and Adaptive Components: Employed to represent non-standard, dynamic, or sensor-integrated elements such as mobile workers, hazard indicators, and alert markers. Adaptive components were designed to support location-based parameter updates and visual representation of real-time spatial behaviors.

The 3D Revit Model developed for this research adheres to the Level of Development (LOD) 400 standard, aligned with the principles outlined in the ISO 19650 series for information management using building information modelling.

This level of development ensures that each model component possesses the geometric and non-geometric detail necessary to support installation, and operational decision-making. In practice, the LOD 400 model elements used in this study are:

- Dimensionally accurate and reflecting real-world layouts,
- Fully defined in terms of shape, size, and orientation, suitable for downstream use in field coordination,
- Enriched with non-graphical data, including parameters related to access zones, worker roles, and functional purpose.

The selection of LOD 400 was made to support the core objectives of this research, which include real-time integration of location data, spatial risk analysis, and visualization of safety-critical information in a dynamic digital twin environment. Borrmann *et al.* (2018) highlight that LOD 400 plays a pivotal role in the development of construction-oriented digital twins, as it bridges the gap between design documentation and operational spatial logic. Similarly, Volk *et al.* (2014) emphasize that the granular detail offered by LOD 400 is essential for sensor integration, asset monitoring, and spatially contextualized visualization all of which are critical functions in real-time safety monitoring and decision support systems.



Figure 3.7. Data science and artificial intelligence institute building from front.



Figure 3.8. Data science and artificial intelligence institute building Revit model from front.



Figure 3.9. Data science and artificial intelligence institute building Revit model from back.



Figure 3.10. Engineering building faculty.

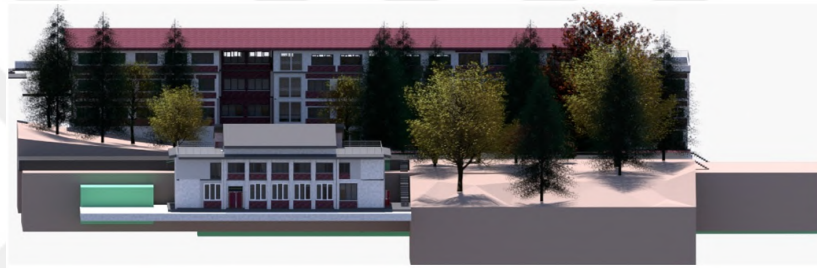


Figure 3.11. Revit model engineering faculty and data science and artificial intelligence institute 1.

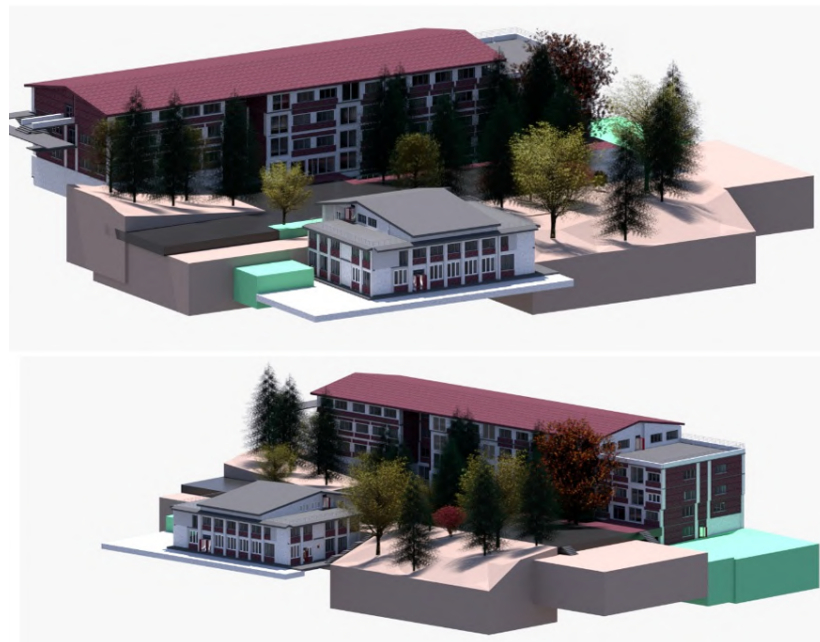


Figure 3.12. Revit model engineering faculty and data science and artificial intelligence institute 2.

The resulting model, combining LOD 400-level geometric fidelity with rich semantic annotation, serves as both a computational platform for automation, safety simulations, and risk evaluation in immersive digital environments.

3.9.2. Revit API: Data Interfacing and Scripting Environment

The Revit API is a .NET-based application programming interface provided by Autodesk, which enables programmatic interaction with Building Information Models (BIM). Through this API, users and developers can read, manipulate, and automate operations on model elements, parameters, geometry, and views. In the context of this research, the Revit API was used to develop a bridge between real-time position tracking data (originating from UWB and GPS devices) and the BIM environment for construction site safety monitoring.

The Revit API Autodesk's programmatic interface for Building Information Modeling that provides an extensive set of classes and methods, structured across multiple namespaces, to facilitate automated interaction with BIM model elements. Core namespaces such as Autodesk.Revit.DB (responsible for document-level operations, geometry manipulation, and parameter access) and Autodesk.Revit.UI (which enables user interface customization and interaction) serve as the foundational components for extending Revit's capabilities through external scripts or add-ins.

According to Eastman *et al.* (2011), a key advantage of the Revit API in the domain of architecture, engineering, and construction (AEC) lies in its potential to automate complex and repetitive modeling tasks, reducing manual labor and minimizing error. Moreover, the API supports integration with external databases and real-time sensor feeds, making it an effective platform for building dynamic, data-driven digital twin environments. Within the scope of this research, the Revit API was employed to perform a series of data-driven operations on 3D model elements to facilitate real-time worker monitoring.

The following capabilities were essential to this integration:

- **Parameter Access and Modification:** The method `Element.LookupParameter()` was used to retrieve and update custom parameters defined within family instances. These parameters such as `WorkerID`, `CurrentZone`, and encoded semantic information about individual workers and their respective positions or exposure status. By enabling read/write operations on these fields, the system could dynamically reflect the current state of tracked individuals within the model.
- **Geometric Transformation of Elements:** Specially created adaptive family instances were placed in the model to visually represent workers. These instances were programmatically repositioned based on external UWB-GPS coordinate inputs. The `LocationPoint.Move()` method allowed the system to relocate these components in real-time, ensuring that their placement in the BIM model accurately mirrored field conditions.
- **Visual State Encoding Through Overrides:** To provide intuitive visual feedback, the API's graphical customization classes were employed. The `Override Graphic Settings` class was used to assign different visual styles (e.g., color, transparency) to model elements based on status of each worker. Combined with `ActiveView.SetElementOverrides()`, this approach enabled a clear and immediate visual distinction.

3.9.2.1. Integration with External Data: Workflow.

To enable seamless synchronization between the real-world position of construction workers and their virtual representation within the **Autodesk Revit 3D model**, a structured data integration pipeline was developed. This pipeline connected sensor data acquisition, filtering, transformation, and visualization within a cohesive framework.

- **Step 1: Data Import**

The first stage involved reading external position data, recorded and filtered outside the Revit environment. The positional data originally gathered from

GPS and Ultra-Wideband (UWB) sensors was processed using an Unscented Kalman Filter (UKF) to reduce noise and improve reliability. The filtered data was periodically exported in .csv format, with each entry containing a unique worker identifier and corresponding 3D coordinates.

Within Revit, IronPython scripts executed via the pyRevit interface were used to parse these files. Python's built-in csv module enabled row-by-row extraction of key fields such as WorkerID, X, Y, and Z. These values were then prepared for use in coordinate transformation and model updates.

- Step 2: Coordinate System Transformation

Since Revit's internal coordinate system is based on a project-defined Cartesian space which may not directly align with global geospatial systems like ITRF (International Terrestrial Reference Frame) a transformation process was necessary. A rigid body transformation (i.e., a combination of rotation and translation) was applied to map sensor coordinates to the BIM model's spatial context.

This transformation was derived using known surveyed reference points in both systems, and implemented using the Transform class in the Revit API. The matrix computed included a translation vector (to align origins) and a rotation matrix (to align axes), ensuring that worker positions mapped precisely to their real-world locations in the digital model (Wolf and Ghilani, 2014; Uren and Price, 2010).

The Internal Origin is Revit's immutable coordinate origin and represents the system's default (0,0,0) location. All model geometry is, by default, positioned relative to this internal Cartesian reference frame. The Internal Origin is invisible during standard modeling but serves as the absolute baseline for all coordinate calculations, including those performed through the Revit API (retrieving XYZ locations of elements or placing components programmatically). It cannot be altered or redefined by the user, and all transformations ultimately resolve relative to this system-level origin.

The Survey Point is utilized to associate the Revit model with an external geospatial coordinate system, such as a projected Cartesian system (UTM or local site-

specific grid) or a geodetic framework (in this study it is ITRF20). This point can be repositioned and assigned real-world coordinates derived from GNSS surveys or topographic benchmarks. It serves as the anchor for aligning the 3D Revit model with external sources such as GPS logs, point clouds, and GIS-linked data.

In the context of this research, the Survey Point was positioned at a known control point on-site, measured using a high-precision GNSS receiver. The coordinates were entered directly into Revit, enabling the establishment of a Shared Coordinate System aligned with the real-world reference frame used by the sensor fusion algorithm.

Shared Referential Coordinates in Revit **model space** functionality creates a transformation between the internal model space and the georeferenced real-world system defined by the Survey Point. This transformation consists of: A translation vector (defining the offset between origins), a rotation value (typically representing angular deviation in the horizontal XY plane), and optionally, a scale factor (rarely applied in architectural modeling contexts). Once configured, this transformation enables seamless translation of real-world coordinates (e.g., from UWB tags or GPS trackers) into Revit's internal space. This is crucial for dynamically placing worker markers and visualizing movement within the correct architectural context.

- Step 3: Element Filtering and Matching

Once coordinate alignment was completed, the next task was to identify the corresponding worker representation in the Revit model. This was achieved using the Filtered Element Collector class, which allowed for efficient querying of all model elements of type Family Instance. These were further filtered by evaluating the Worker ID parameter to find a match with the incoming data.

- Step 4: Updating Location and Attributes

For each matched element, a transaction was initiated using the Transaction class in order to ensure reversible changes to the model. The Location Point. Move()

method was then applied to reposition the element based on the transformed coordinates. In addition to geometric movement, other parameters such as Safety Status or TimeInZone were also updated using SetValueString() or Set() methods on retrieved parameter objects.

- Step 5: Visual Feedback and Hazard Communication

To enhance the interpretability of the worker tracking system, visual encoding techniques were used. In Revit model lines are planned to draw with the position data transferred in to the model. And Enscape in Revit is used for visual tracking for virtual reasons. By Enscape in Revit changes in Revit 3D model are dynamically updated which enables the nearly real time worker position tracking.

3.9.3. Python Script Development in pyRevit

The objective of this stage was to develop a programmable interface within the Revit environment to automate the integration of filtered location data from UWB and GPS sensors into the Revit model. To achieve this, pyRevit, an IronPython-based scripting and interface layer for Autodesk Revit, was employed.

pyRevit provided a lightweight yet powerful framework that enabled:

- Direct scripting in IronPython 2.7, compatible with Revit 2022's .NET-based API,
- Integration of custom buttons into the Revit ribbon for repeatable execution,
- Real-time script-triggered model updates without requiring external plug-ins or heavy processing engines.

The primary aim of this component was to create a scripting pipeline that could:

- (i) Read external .csv files containing filtered coordinates and worker metadata,
- (ii) Parse this data and apply geometric transformations,
- (iii) Locate the corresponding digital worker representations in the Revit model,
- (iv) Update their location and semantic parameters, and
- (v) Provide optional visual feedbacks.

Each Python script developed in pyRevit was embedded into a custom ribbon panel for ease of execution by designers or supervisors. The scripts utilized core Revit API classes such as Filtered Element Collector, LocationPoint, and Transaction. This modular approach facilitated extensibility, enabling additional scripts for visualization and parameter monitoring to be added seamlessly to the same interface.

In this study, a custom user interface plugin was developed using the open-source pyRevit framework to enable the placement of Revit component based on real-time or recorded location data. As illustrated in the Figure 3.13, the plugin follows a modular hierarchical structure.

At the top level, the MyAvatarExtension.extension folder defines the overall extension, which is recognized by pyRevit. Within this extension, a custom tab (AvatarTab.tab) and panel (AvatarTools.panel) are created to organize tools in the Revit ribbon interface. A push button (SimulateAvatar.pushbutton) is placed inside the panel to execute the desired functionality.

This button runs a Python script (script.py), which reads location data typically from a CSV file transforms the data into Revit's coordinate system, and places family components at the corresponding positions in the model. This system enables users to integrate site-specific spatial data into the BIM environment with a single click, supporting applications such as worker movement visualization, safety monitoring, and digital twin simulations in construction projects.

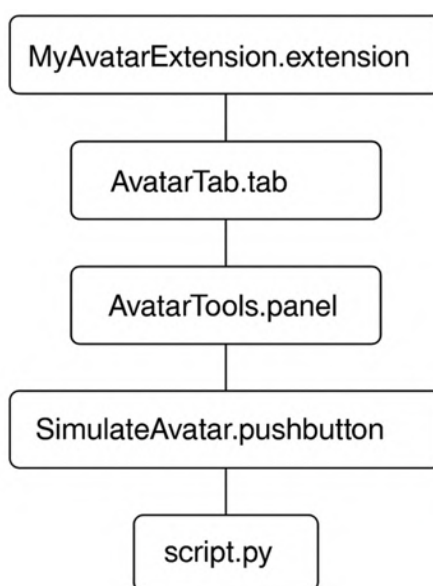


Figure 3.13. Hierarchical structure of the pyRevit extension for component placement based on location data.

3.9.4. Worker Representation in Revit

One of the most meaningful parts of this research was finding a way to bring each worker into the digital model not just as data points, but as visual, interactive avatars. To achieve this, I developed a custom adaptive family in Revit that served as a digital proxy for every individual being tracked on-site. The motivation behind creating this adaptive component was rooted in two goals: to make real-time worker locations immediately understandable within the model and to allow the model to respond dynamically as workers moved. These avatars were not just symbolic they became a way to observe the construction site as a living, breathing system.

Each of these worker avatars was:

- Simple in shape like a small, colored cylinder or abstract figure-so as not to visually clutter the model, but expressive enough to distinguish one worker from another.
- Categorized under Generic Models for consistency and ease of filtering.

- Enhanced with parameters such as WorkerID, CurrentZone, and SafetyStatus, which allowed each instance to carry meaningful information.
- Made adaptive, meaning their placement could respond dynamically to new coordinates using scripting. This flexibility allowed the avatars to “move” in real-time as new GPS/UWB data arrived.

Designing the avatar as an adaptive component was a deliberate choice. Adaptive families in Revit allow points to be controlled programmatically and geometrically, which made them perfect for our case. As each new location update came in, the family’s position could be modified using the Revit API, without needing to delete or recreate elements. This ensured a smooth and efficient workflow, maintaining continuity in tracking while minimizing model disruption.

By linking these avatars to sensor data, the BIM model became more than a static design file it became a real-time spatial twin of the site. This transformation made it possible to observe workers’ presence, movement, and safety context directly within the BIM environment, enabling proactive safety insights, behavioral mapping, and improved decision-making for site supervisors and safety teams. Ultimately, creating a visual identity for each worker helped bridge the gap between the digital and physical world. It made the data more humane and the model more alive.



Figure 3.14. Adaptive component represents the worker avatar.

3.9.5. Use of Visualization Engines (Enscape)

Enscape is a plug in which has virtual related concerns. It enables to monitor and control circulation through the BIM environment in the prepared BIM model. To enhance the spatial awareness and communicative capacity of the BIM environment, this study integrated two real-time rendering engines Enscape into the visualization workflow. While Autodesk Revit served as the foundational platform for geometric modeling, semantic data, and scripting integration, these tools provided a layer of immersive visualization that allowed for more intuitive analysis, scenario-based safety review. The screenshot shows the Enscape has BIM Information panel, which exposes Revit model content as a hierarchical inventory (category → family/type) together with instance counts. In the left tree, several linked DWG files-birinci kat.dwg, bodrum m hendislik.dwg, and  atı katı.dwg are listed with the status location < Not Shared >. This status indicates that these links have not been registered to the project's Shared Coordinates; in other words, their internal coordinates are not yet mapped to the project/site's common geodetic frame. For any RTLS/GNSS integration, establishing shared coordinates (e.g., via Acquire/Publish Coordinates and defining a shared site) is essential to ensure that model geometry aligns with transformed UWB/GNSS positions.

Below the links, core Revit categories (e.g., Ceilings, Doors, Floors) are shown with their type names and instance counts. For example, door types such as 0813 × 2134 mm (20) and 0864 x 2134 mm (2) denote how many instances of each type are present; floor types such as Generic 150 mm (31) report counts for construction elements used in circulation paths or safety zones. This inventory view supports QA/QC of modeling scope (e.g., verifying that openings, slabs, and ceilings are present and correctly typed), and it facilitates zone and hazard delineation used later for real-time visualization of worker trajectories.

Methodologically, the panel is used for (i) verifying coordinate readiness of all external references prior to streaming positions into BIM; (ii) checking model completeness and typology for the areas where tracking will be visualized; and (iii) selecting and

highlighting elements in context during pilot runs to confirm visibility and line-of-sight conditions in the rendered scene. For official quantities or analytics, the counts visible here are complemented with Revit Schedules, while the real-time markers themselves are driven by the Revit API/pyRevit pipeline described in the integration subsection.

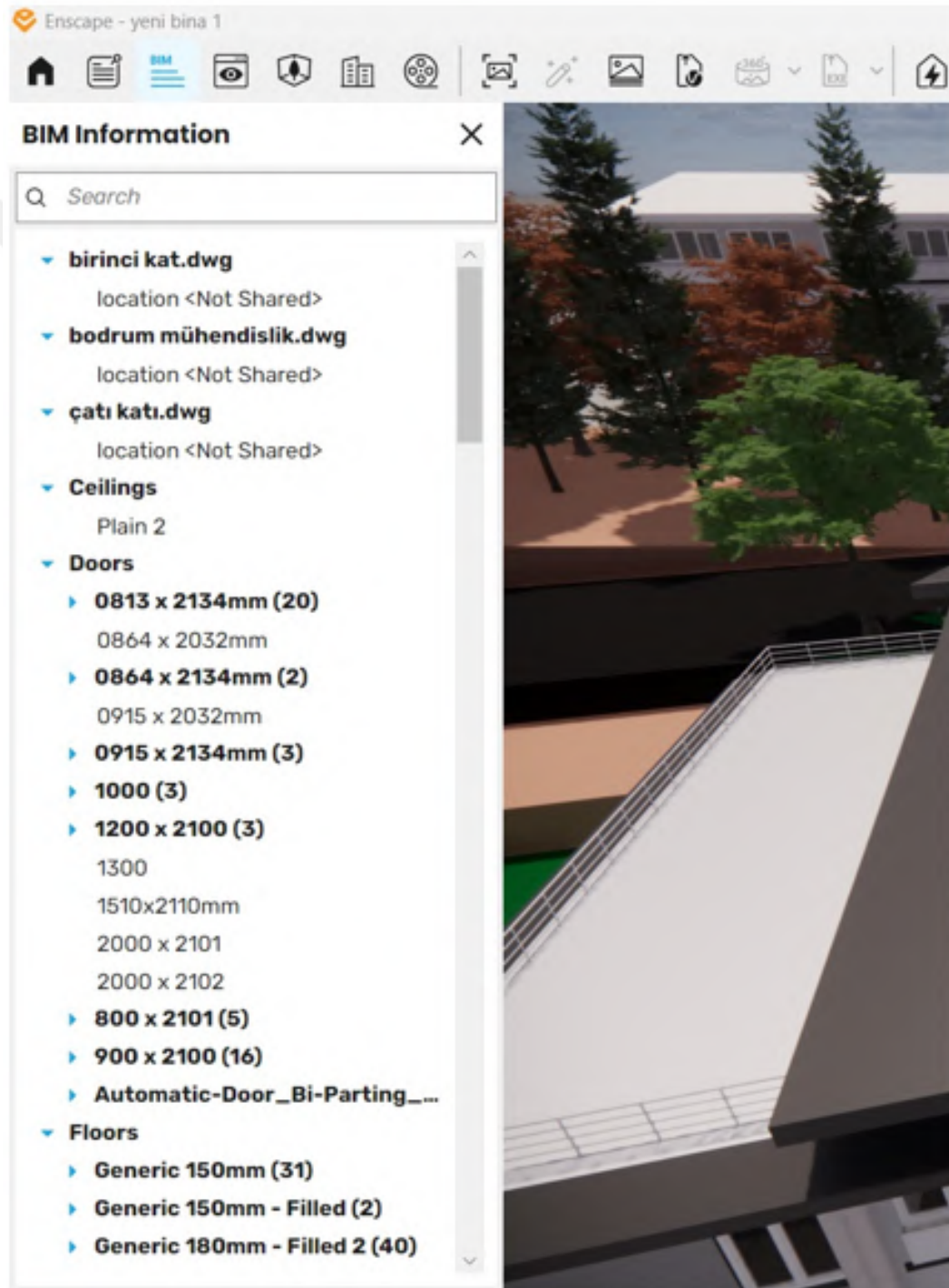


Figure 3.15. Enscape interface BIM information coordination.

4. EXPERIMENTAL SETUP AND VALIDATION

Section 4 describes how the testbed was built and how the system’s accuracy was checked. Two GNSS control points (PA, PB) were surveyed by experienced geomatics engineers to define a common spatial frame; antenna heights were recorded, data were post-processed with CSRS-PPP to ITRF2020 (epoch 2025.4) at centimeter-level accuracy, and then converted to UTM to serve as the project’s global reference. This frame was used to align the indoor UWB network with negligible rotation/scale so that all tag tracks could be plotted directly in the Autodesk Revit model without distortion.

The testing environment spans indoor and outdoor areas across two adjacent buildings. Exterior control near building entrances enabled consistent indoor–outdoor transitions, while interior UWB anchors were referenced to the same global frame to unify GPS and UWB measurements for fusion and visualization in Revit.

For ground-truthing, predefined waypoints and validation paths were marked; a timestamped video of each run was synchronized with logs for frame-by-frame checks. Indoors, architectural features from the Revit model (e.g., walls, doors, gridlines) supported visual verification; fused trajectories were overlaid on Revit floor plans to inspect discrepancies. Outdoors, GPS (tied to surveyed markers) provided the baseline. This hybrid approach supplied comprehensive validation across spaces.

Accuracy and evaluation followed a consistent protocol comparing GPS-only, UWB-only, and UKF-fused solutions along surveyed validation paths. Participants walked at controlled pace; streams were NTP-synchronized, filtered with stepwise 3D Mahalanobis, and mapped into the common frame (Helmert/Procrustes) before fusion. Errors were reported as cross-track (to the validation polyline), along-track offset (for latency study), and point error at checkpoints.

4.1. GNSS Coordinate Measurement

In order to align and arrange the coordinate transformations later GPS coordinate values of 2 outdoor reference points were selected and their coordinates were calculated with GNSS antenna receiver. Defining a robust local coordinate system (ENU), at least two geodetic control points must be accurately surveyed via GNSS. The measurements of the two reference points PA and PB, were carried out by two geomatic engineers with professional experience in GNSS surveying and precise geolocation techniques (M.Sc. Geomatics Engineer Burak Kotan and M.Sc. Geomatics Engineer Yaşar Topal). Their involvement ensured the adherence for field based GNSS control point establishment, and enhanced the reliability of the collected data.

This measurement of the 2 points provided both translation and rotation precision when transforming between coordinate frames. As highlighted by Zhu *et al.* (2020) and buildingSMART, a minimum of two points were required to compute a 2D Helmert transformation (shift, rotation and scale); two points provides directionality, while a third would only allow error checking for plane distortions. In terrestrial laser scanning, a method validated by Krzystek *et al.* (2017) showed that direct georeferencing with only two GNSS points can achieve sub-centimeter accuracy if vertical deflections are accounted for using gravity models. This confirms that two well-surveyed points are both necessary and sufficient for defining an ENU system with high local precision.

Selecting exterior GNSS two referential points near the two entrances of the buildings that are the setting of the experiments allows the transition of the ENU system from outdoor to indoor environments. To ensure consistent localization across both indoor and outdoor environments, UWB anchors installed inside buildings must be contextually referenced to the same global coordinate system as the GPS data. This enables seamless transformation and integration of dynamic location data collected via heterogeneous systems such as GPS and UWB into a unified spatial framework, which is crucial for real-time tracking and safety monitoring applications in construction environments (Zafari *et al.*, 2019; Liu *et al.*, 2020; Guo and Li, 2023). As described

by Krzystek *et al.* (2013), this method works well even when GPS reception indoors is limited, enabling alignment across indoor and outdoor scans.

Both reference points which are PA and PB were surveyed using GNSS antennas in RTK/PPK mode with fixed baselines, delivering centimeter-level accuracy. This process reduces systematic biases such as ionospheric and multipath errors and establishes control points with geodetic certainty, serving as a reliable foundation for the ENU frame. The GPS and Geodesy educational material emphasizes that GNSS positions must be geodetically referenced to ensure global consistency, while BIM georeferencing guidelines highlight the need for external spatial control tied to open standards.

The use of dual-frequency GNSS receivers as provided by Topcon's geodetic-grade systems enables post-processed mitigation of key systematic error sources, including ionospheric delays, tropospheric disturbances, and satellite orbit/clock biases (Leick *et al.*, 2015; Teunissen and Montenbruck, 2017; Hofmann-Wellenhof *et al.*, 2008). In this study, the survey strategy applied a short-duration static observation method (approximately three hours) for each point, in line with the optimized approach validated in previous research (Topal, 2025), showing that even short sessions processed with short-duration Precise Point Positioning (PPP) can achieve sub-centimeter horizontal and low-centimeter vertical accuracies. First two outdoor points were specified conveniently for the indoor experiments enabling open view from the inside and designated measurement points were physically stabilized using nails and labeled with spray paint to ensure clear identification during fieldwork shown in the Figure 4. 1. In this research each observation of the specified outdoor points lasted three hours. In order to ensure that the antennas were precisely positioned over the nailed reference points, the tripods were carefully aligned using tribrachs, and the antennas were leveled and fixed directly above the nails. This setup is illustrated in the following Figure 4.2. Through a rigorous, iterative process of centering, combined leveling, stabilization during antenna mounting, and slant height verification, we ensured the Topcon antenna remained precisely located over the nail reference points. After the antenna installa-

tion was completed and the tribrach was securely leveled over the nail-marked point, the Topcon GNSS receiver which is shown in Figure 4.5 was connected to the NetG5 CORS (Continuously Operating Reference Station) network. The GNSS solution mode was configured to use GPS + GLONASS + Galileo constellations for redundancy and improved reliability, as recommended by GNSS practices. During the GNSS measurement setup phase, the satellite constellation configuration was manually adjusted via the receiver utility interface. As shown in Figure 4.6, the following Global Navigation Satellite Systems (GNSS) were enabled and used for positioning: GPS, GLONASS, Galileo, and BeiDou. According to Gucma *et al.* (2021) and PMC (2015), multi-constellation use significantly improves satellite visibility, PDOP, convergence time, and fault detection. Additionally, performance across different latitudes is stabilized, with GLONASS providing advantageous geometry in high-latitude areas. Research on GNSS station networks consistently reports that combining GPS+GLONASS yields the best positioning accuracy, especially in polar regions a testament to complementing constellations (Zheng *at al.*, 2022). During the fieldwork, antenna height offsets were carefully measured and entered into the processing system. The GNSS receiver setup during the 3-hour measurement session is shown in the Figure 4.3.



Figure 4.1. Referential point labelling.



Figure 4.2. Referential point PB antenna settlement.



Figure 4.3. Referential point GNSS measurement.



Figure 4.4. Referential point PB.



Figure 4.5. GNSS receiver.

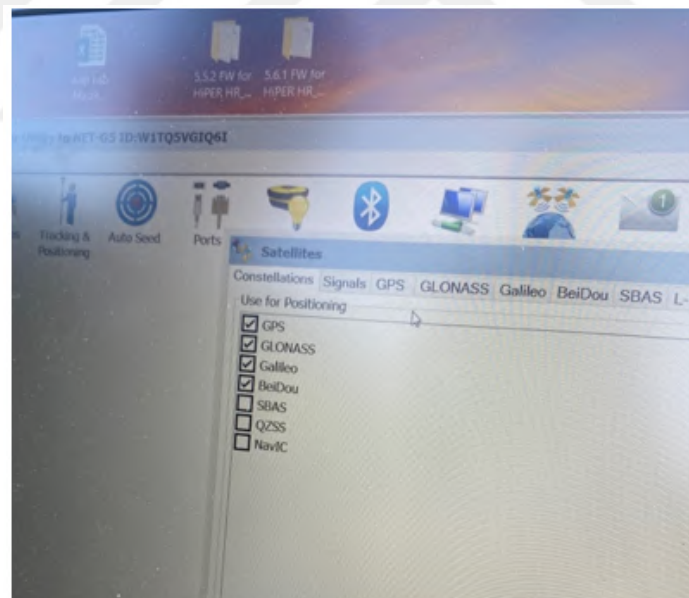


Figure 4.6. GNSS station selection.

The static GNSS data collected from the two fixed reference points PA and PB were post-processed using the CSRS-PPP (Canadian Spatial Reference System-Precise Point Positioning) platform. The processed data yielded geodetically referenced positions in the ITRF2020 frame, epoch 2025.4, with centimeter-level accuracy.

Table 4.1. Observation and processing summary.

Station (Referential Point)	Observation Time	Duration	Constel- lations Used	Ambiguity Fix Rate	Elevation Mask
PA	23.05.2025 15:18-18:21	3 hour 2 minute	GPS+ GLONASS (L1, L2, C1, C2)	74.76%	7.5o
PB	23.05.2025 15:17-18:21	3 hour 4 minute	GPS+ GLONASS (L1, L2, C1, C2)	79.11%	7.5o

Table 4.2. Estimated positions (ITRF20, Epoch 2025.4).

Station (Referential Point)	Latitude	Longitude	Ellipsoidal Height (m)	$\sigma = \frac{(95\%)H}{V}(m)$
PA	41o5'1.0015"	29o3'1.12008"	135.447	± 0.013 / ± 0.054
PB	41o5'1.14667"	29o3'1.35183"	135.484	± 0.013 / ± 0.054

Both positions demonstrate exceptional consistency, with horizontal 95% confidence intervals within 1.7 cm and vertical estimates within 5.4 cm. These values are consistent with expectations for high-quality Precise Point Positioning (PPP) processes obtained under stable sky-view conditions and adequate observation duration was provided which is at least 3 hours.

Finally, the geodetic coordinates of the two reference points were successfully computed in the Universal Transverse Mercator (UTM) coordinate system, providing a metrically consistent spatial framework for further integration and transformation processes.

Table 4.3. UTM coordinates zone 35N.

Station (Referential Point)	N (m)	E (m)
PA	4.550.067.309	672.221.228
PB	4.550.068.871	672.226.602

The UTM coordinates indicate that the two stations are approximately 6 meters apart in the East-West direction and nearly aligned in the North-South direction, confirming their relative positioning suitability for defining a local coordinate reference system. The GNSS observations conducted at the PA and PB stations demonstrated high-quality results suitable for establishing reliable geodetic control points. Both stations achieved fixed ambiguity solutions in over 74% of epochs, indicating robust carrier-phase positioning.

A low elevation mask of 7.5° was applied to reduce the influence of multipath errors while maintaining favorable satellite geometry. The resulting coordinate estimates exhibited sub-centimeter horizontal precision and vertical accuracies within 5cm at the 95% confidence level. These precision levels are essential for ensuring spatial consistency in downstream processes such as coordinate transformation, Revit integration, and the calibration of indoor-outdoor positioning systems.

The GNSS measurements at both PA and PB points were successfully collected and post-processed using the CSRS-PPP service, yielding reliable geodetic positions in the ITRF2020 reference frame with centimeter-level accuracy. These coordinates form the geodetic foundation for subsequent coordinate transformations and localization calibration steps in the BIM environment-integrated tracking framework.

4.2. Testing Environment Design: Indoor/Outdoor Experimental Area

UWB network hardware set up was employed commercial ultra-wideband modules such as Qorvo(formerly DecaWave) transceivers (DWM1001 MDEK). These chips of the integrated sensors transmitter / receiver circuitry and support centimeter-level

accuracy (nearly 10 cm loss-of-sight) over 3.5-8.0 GHz (Al-Okby *et al.*, 2024). For example, the Qorvo DW1000 provides 10 cm accuracy at up to 200 m range (Al-Okby *et al.*, 2024).

This study utilizes the Decawave DWM1001 module and the MDEK1001 development kit, both of which are widely recognized for their commercial-grade performance in real-time indoor localization using Ultra-Wideband (UWB) technology. DWM1001 is a compact UWB transceiver module that integrates; the DW1000 UWB chip for precise time-of-flight ranging, processor and Bluetooth Low Energy connectivity, UWB antenna.

The experiments were conducted in a multi-zone environment encompassing two adjacent buildings; the Engineering Faculty building and the Data Science and AI building as well as the outdoor area between them. The two buildings are approximately 40 m apart, separated by an open courtyard. Eight UWB anchor nodes (denoted A1-A8) were installed to cover the indoor areas of both buildings: four anchors were placed in the Engineering Faculty building (anchors A1-A4) and four in the Data Science and AI building (anchors A5-A8). The anchors were mounted at elevated positions (approximately 2 m above the ground floor of the building). Anchors were set up with the laser cut pre-arranged models compatible for the UWB sensor dimensions. In the Figure and Figure testing indoor environments were given.



Figure 4.7. 3D model case for UWB anchors.



Figure 4.8. Engineering faculty testing indoor environment.

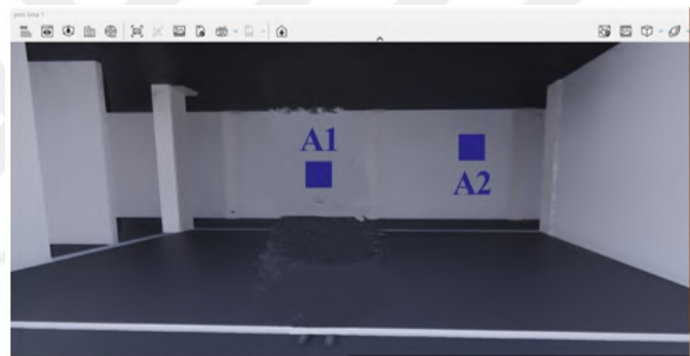


Figure 4.9. Engineering faculty testing indoor in virtual environment.

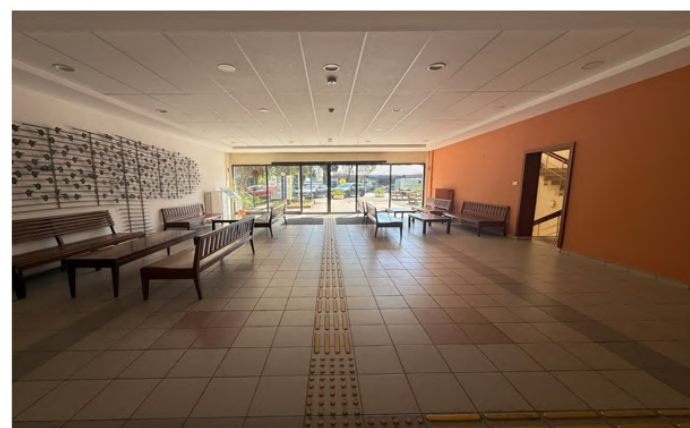


Figure 4.10. Engineering faculty testing indoor in virtual environment (from indoor).



Figure 4.11. Engineering data science institute testing environment.



Figure 4.12. Engineering data science institute testing environment a1.



Figure 4.13. Engineering data science institute anchor attachment testing environment.



Figure 4.14. Engineering data science institute testing anchors (sequentially A6-A5).

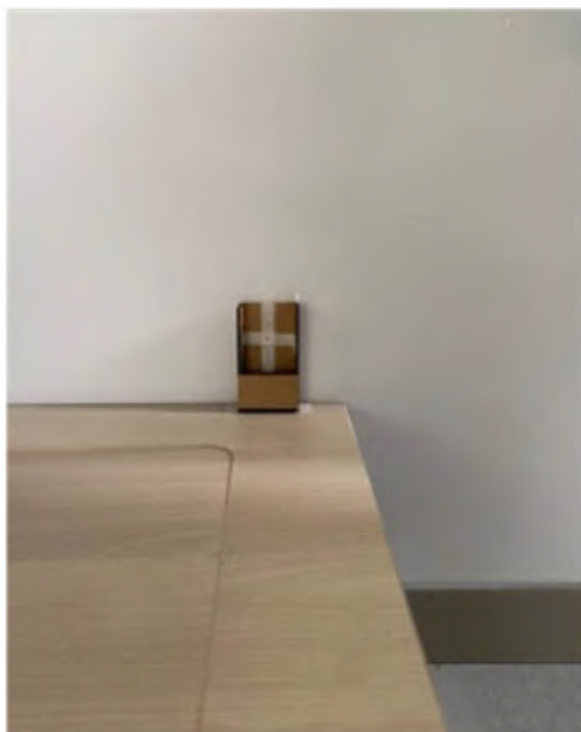


Figure 4.15. Engineering data science institute testing anchors A8.



Figure 4.16. Engineering data science institute testing anchor positions A7.



Figure 4.17. Engineering data science institute testing anchor positions A6.



Figure 4.18. Engineering data science institute testing anchor positions A5.

In the outdoor courtyard, no UWB anchors were present; instead, standard GNSS (GPS) signals were available. To provide global reference coordinates for the test site, two GNSS reference stations (denoted PA and PB) were deployed on-site. These stations have known precise coordinates in the global reference frame /UTM Zone 35N) and were used as control points to link the local UWB coordinate system to Earth-centered coordinates.

The indoor layout of each building and the placement of anchors were as follows. In the Engineering Faculty building, anchors A1-A4 were distributed around the area of interest (e.g. mounted on walls in a laboratory and corridor), spanning a 12 m by 9 m area. In the Data Science and AI building, anchors A5-A8 covered an 6-8 m span of a laboratory and hallway which consists panels and lot steel reflector architectural elements. The anchors in each building form a “cluster” covering that building’s interior. Anchor A8, located in the Data Science building, was chosen as a reference point for the local coordinate system. During setup, the position of A8 was

measured with GNSS (via an outdoor point with a known offset to A8's indoor location) to obtain A8's absolute coordinates in UTM. The other anchors' global coordinates were determined by tape measurements and survey methods relative to A8, combined with GNSS measurements from the reference stations. All anchor coordinates were then expressed both in a local 3D Cartesian frame (with A8 at the origin for convenience) and in the global UTM Zone 35N coordinates.

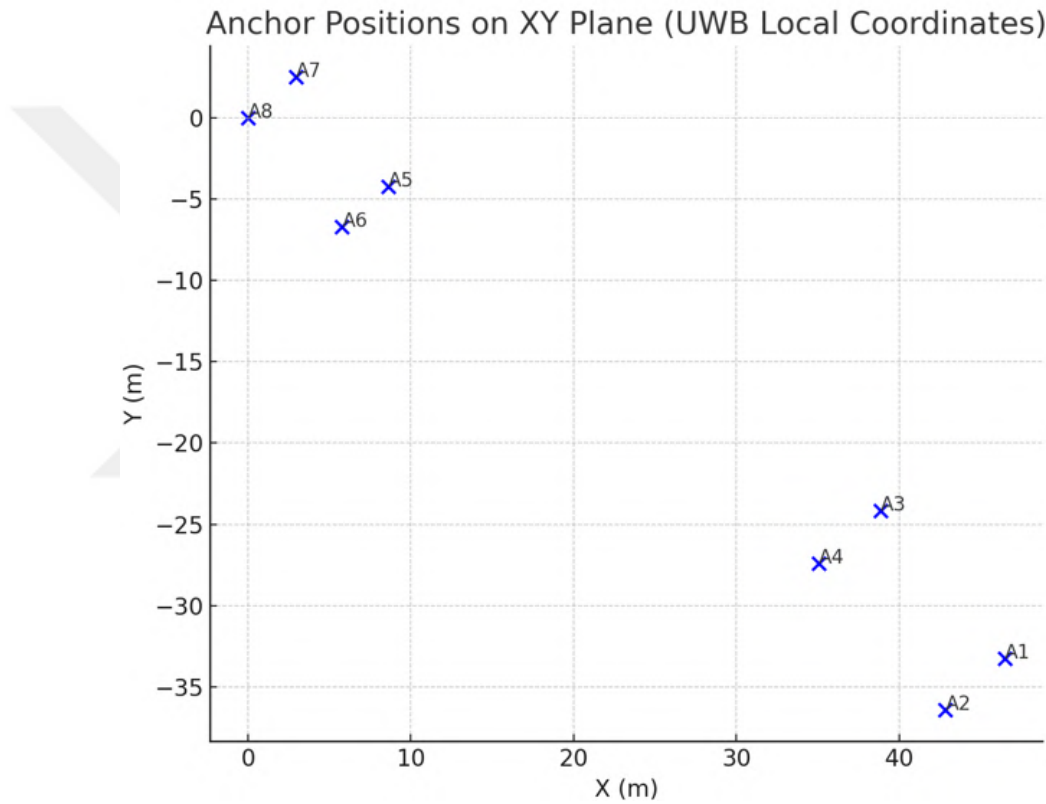


Figure 4.19. Planar (2D) layout of the UWB anchor positions in the test environment (top-down view).

Table 4.1 lists the coordinates of each anchor in both the local UWB frame and the global UTM frame. In the local frame, anchor A8 serves as the origin (0,0,0). The X-Y plane of the local frame was chosen to coincide approximately with the horizontal plane of the buildings, and the local X-axis was initially aligned roughly eastward (coincident with the global Easting direction) for convenience. Thus, the local coordinates in Table 4.1 are essentially the offsets (ΔX , ΔY , ΔZ) of each anchor relative to anchor A8. The global coordinates are given in the WGS84/UTM 35N system (Easting,

Northings, and altitude above mean sea level). As shown in the table, the two anchor clusters are clearly delineated. For example, anchors A1-A4 have local coordinates around (35-46 m, -24 to -36 m) relative to A8, indicating that the Engineering building cluster is about 40 m east and 30 m south of the Data Science building's anchor A8. In contrast, anchors A5-A7 are within 0-9 m of A8 in the local frame, reflecting their placement within the same building as A8. All anchors have similar altitude (Z) values globally (around 134-136 m), with slight differences (1-1.5 m) due to mounting height variations. This confirms that all anchors were placed on a single floor level (ground floor of each building) with minor height offsets.

Table 4.4. Anchor points measured coordinate values.

Anchor	UWB X (m)	UWB Y (m)	UWB Z (m)	UTM Easting (m)	UTM Northing (m)	Altitude (m)
A1	46.48	-33.23	1.14	6.722.496.800	45.500.463.926	1.354.235
A2	42.84	-36.38	-0.05	6.722.460.402	45.500.432.354	1.342.370
A3	38.84	-24.18	-0.05	6.722.420.402	45.500.554.407	1.342.380
A4	35.07	-27.41	1.16	6.722.382.719	45.500.522.084	1.354.499
A5	8.60	-4.23	-0.05	6.722.118.027	45.500.753.830	1.342.406
A6	5.77	-6.71	1.54	6.722.089.766	45.500.729.036	1.358.233
A7	2.94	2.49	1.55	6.722.061.410	45.500.821.038	1.358.329
A8	0.00	0.00	0.00	6.722.032.021	45.500.796.180	1.342.875

Table 4.3 illustrates the spatial distribution of Ultra-Wideband (UWB) anchors in a 2D plan view, encompassing two adjacent indoor environments. Anchors A1 through A4 (blue) are installed in Indoor-1 (Building 1), whereas anchors A5 through A7 (red) are positioned in Indoor-2 (Building 2). Anchor A8, shown as a red square, is designated as the local origin (0,0) of the coordinate system. This reference point was selected for alignment and transformation between UWB and Revit coordinate frames. The horizontal (X-axis) and vertical (Y-axis) separation between the two buildings is approximately 46 meters and 33 meters, respectively. This spatial separation necessitated a distributed anchor layout across both buildings to ensure consistent coverage

and accurate positioning. The configuration supports multi-building localization experiments and enables the tracking of transitions between indoor areas, which is critical for integrated worker monitoring and hybrid positioning systems.

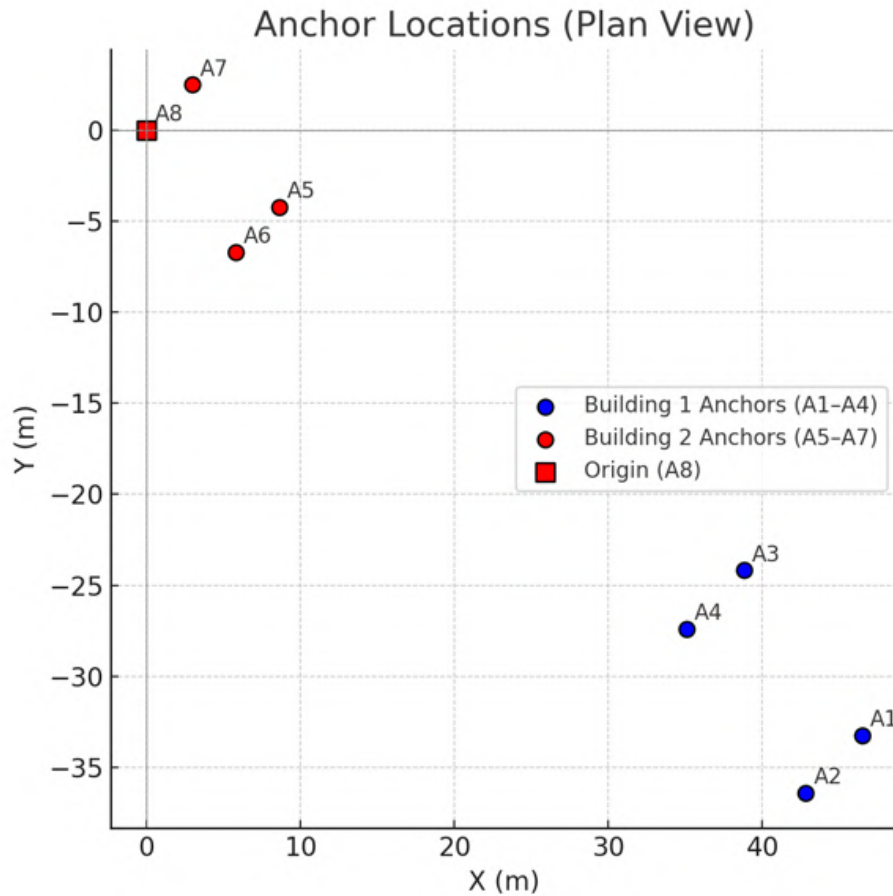


Figure 4.20. UWB anchor placement in 2D plan view. Anchors A1-A4 (blue) cover Indoor-1, while A5-A8 (red) are in Indoor-2. A8 is the local origin (0,0). The two buildings are about 46 m apart in X and 33 m in Y (south) direction.

The figure below presents mapping drawings exported from Revit, showing the indoor and outdoor experimental areas and the anchor locations. The green line indicates the prescribed walking paths along predefined lines established for conducting the accuracy check. Participants wearing the tag on a hard hat walked exactly along these lines at a steady pace (1-1.5 m/s), starting and finishing at marked fiducials. These green paths therefore serve as quasi-ground truth for benchmarking indoor and outdoor performance under consistent geometry.

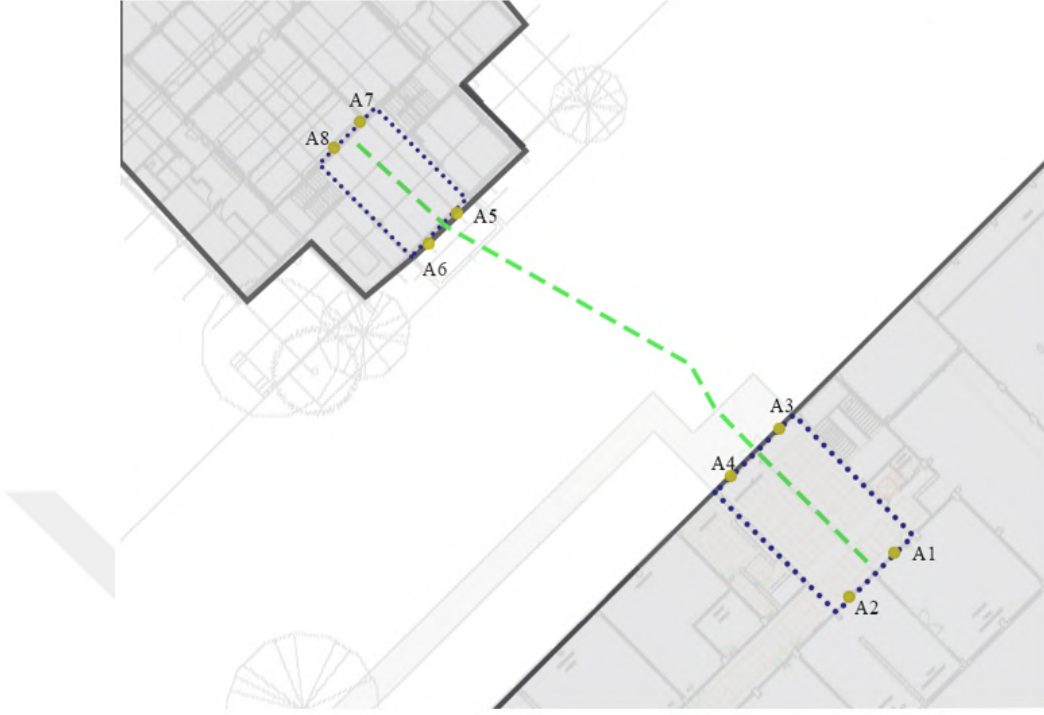


Figure 4.21. Green polylines indicate predefined validation paths used for the accuracy check.

4.3. Hardware Components

For outdoor positioning, we utilized a consumer smartphone's GNSS receiver. The phone was carried by the worker and set to record GPS fixes at 1 Hz but it calculated as less than 1 Hz. The device provides latitude/longitude (WGS84) and altitude, which we convert to UTM coordinates for fusion.

One DWM1001-DEV tag was worn by the test subject (worker). The tag performs two-way ranging with nearby anchors and computes its 3D position in a local coordinate frame. In our setup, the UWB network operated in TWR (two-way ranging) mode with a location update rate of 10 Hz. The tag's measured positions (x, y, z) in the tag's configured frame were streamed in real time.

Using the corresponding coordinates of anchors in the two frames, a 2D similarity transformation (consisting of a planar rotation and translation) was derived to convert

UWB local coordinates into UTM coordinates. Since the scale between the local meter units and UTM meters is inherently 1:1 (both are metric), no scale change was needed; thus the transformation reduces to finding the rotation angle (θ) and translation (T_x , T_y). We performed a least-squares fit using all anchor points to determine the optimal rotation and translation that map the local (x , y) coordinates to the global (X , Y) coordinates. The mathematical form of the 2D transformation is,

$$X = x\cos\theta - y\sin\theta + T_x, Y = x\sin\theta + y\cos\theta + T_y. \quad (4.1)$$

which defines the transformation applied to local 2D coordinates as column vectors and column vectors are shown as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \quad (4.2)$$

where (x , y) are the local planar coordinates and (X , Y) are the corresponding coordinates in UTM. The least-squares solution finds the angle θ and translations (T_x , T_y) that minimize the sum of squared differences between the transformed local coordinates and the surveyed global coordinates of the anchors. In practice, this was achieved by first translating the local and global coordinates to a common centroid (using A8 as the origin already accomplishes this for the local side), and then solving for the rotation that best aligns the two sets.

The resulting rotation was very small (on the order of a few degrees or less), indicating that our local axes were already nearly aligned with the true North-East axes of UTM - this is expected since we roughly aligned the local X-axis to the eastward direction during setup. The translation components (T_x , T_y) were essentially equal to the UTM coordinates of anchor A8 (since A8 was origin in local frame). After applying the transformation, the residual errors for all anchor positions were on the order of a few centimeters or less, confirming the consistency of the calibration. In summary, using GNSS-surveyed anchor positions (with the aid of base stations PA/PB) enabled us to “ground truth check” the UWB anchor network in the global frame and to define a coordinate conversion with negligible distortion. This conversion was then used to express all UWB tag tracking data in global coordinates for fusion with GPS data.

For our system, the resulting parameters indicated an almost identity transformation. The scale was $s = 1.000$, and rotation θ (the local X-Y axes were aligned to true East-North axes). The translation vector was essentially the UTM coordinates of A8 (the local origin). In summary, the local UWB coordinates differ from UTM by only a fixed offset (and an insignificantly small rotation).

This outcome was expected because we chose to orient the local frame to north and east when placing anchors. We validated the transform by checking other anchors: applying 2D transformation equation (4.1) to the local coordinates reproduced the surveyed UTM coordinates within a few centimeters. This alignment ensures that an object tracked in the UWB frame can be directly plotted on the site's Revit model without distortion. All subsequent UWB tag positions are converted to UTM using these parameters before fusion with GPS data.

Figure 4.22 displays the spatial distribution of GPS points recorded using an iOS-based sensor logging application during the site walk. The collected data, visualized on a satellite basemap, captures the trajectory around the southern entrance of the case building at Boğaziçi University's South Campus.



Figure 4.22. GPS data logged from the IOS-based sensor logger.

4.3.1. Ground Truth Measurement Depending on Visual Check

To validate the accuracy of the localization system, ground truth positions were manually marked at predefined waypoints on the experimental path.

The test subject followed a pre-planned trajectory along these markers, and the entire sequence was recorded via a timestamped video camera. This camera feed was later synchronized with the system’s logged tracking data to allow frame-by-frame verification of estimated tag positions relative to actual movement. In indoor regions, ground truth verification was further supported through architectural references from the Revit model. Specifically, key structural elements (e.g., wall junctions, doorframes, and column lines) were extracted from the Revit model and used as spatial anchors during analysis.

Additionally, a Revit-based floor plan mapping was generated by importing the timestamped location data of the UWB-GPS fusion output into the BIM environment. This mapping facilitated spatial validation by overlaying the trajectory on the digital twin. Discrepancies between the fused position estimates and ground truth were visually and numerically assessed.

For outdoor sections of the experiment, GPS data served as the primary ground truth baseline. Known survey markers with fixed GNSS coordinates were used to correct for satellite multipath error and alignment inconsistencies. This hybrid indoor-outdoor ground truth methodology ensured comprehensive and accurate validation of the proposed localization and visualization framework.

4.3.2. Accuracy and Evaluation

To quantify positioning accuracy across indoor areas, outdoor courtyards, and indoor-outdoor transitions, we evaluate three solution modes—GPS-only, UWB-only, and the fused solution produced by an Unscented Kalman Filter (UKF). As a refer-

ence geometry, we establish validation paths and checkpoints: outdoors, polylines and control points are surveyed with RTK-GNSS along sidewalks and open spaces; indoors, the corresponding features are measured with a total station/laser distance meter. All reference features are transformed to the project’s global frame (e.g., ENU/UTM) so that sensor estimates and ground truth are strictly comparable.

Participants wear a hard-hat tag and walk precisely along the predefined validation paths at a steady pace (1-1.5 m/s), starting and ending at marked fiducials. Each run lasts 3-8 minutes, and dedicated trajectories cross known door thresholds to capture transitions. Sensors operate at their native cadences (UWB $\approx$ 10-20 Hz; GPS $\approx$ 1 Hz). Prior to analysis, the streams are time-synchronized using NTP-disciplined clocks with linear drift correction, screened for gross errors via stepwise Mahalanobis filtering, and mapped into the common reference frame using a similarity (Helmert/Procrustes) transformation. The fused trajectory is then computed with a constant-velocity UKF whose process noise is tuned to motion state.

Errors are computed with respect to the referential features. The cross-track error is defined as the orthogonal distance from each estimated position to the nearest point on the validation polyline and represents geometric accuracy. The along-track offset is the signed displacement along the polyline and is primarily used later for latency analysis. At discrete checkpoints, a point error is obtained as the Euclidean distance to the surveyed coordinate.

5. RESULTS

This section reports the empirical outcomes of the proposed hybrid tracking framework across indoor, outdoor, and transition zones on the test site. We first summarize data quality and pre-processing (stepwise 3D Mahalanobis outlier detection), then present the performance of the Unscented Kalman Filter (UKF) in fusing UWB and smartphone GPS streams, with emphasis on spatial accuracy and trajectory continuity during entry/exit events. Subsequently, we describe observed GPS behavior under canopy and near façades, and quantify how these conditions affected the fused path. Finally, we benchmark system latency and visualization performance in the BIM environment pipeline (Revit + pyRevit + Enscape), contrasting full-resolution rendering with downsampled/rolling-window strategies to identify practical operating limits on commodity hardware.

5.1. Experimental Results: Accuracy and Continuity

This subsection evaluates the positioning accuracy and temporal continuity of the proposed hybrid system across indoor, outdoor, and transition zones on the campus testbed. Accuracy was assessed in two complementary ways: (i) by measuring cross-track error against a pre-defined “designated walking line” (the green polyline drawn along the intended path in the site plan/figure), and (ii) by visually and statistically inspecting deviations near known multipath/obstruction areas such as building façades and tree canopies. Continuity was examined by tracking trajectory smoothness at indoor-outdoor boundaries and by quantifying the suppression of spurious jumps after outlier filtering and fusion.

Applying stepwise 3D Mahalanobis distance to the raw UWB point clouds consistently flagged a modest portion of samples as implausible, concentrated around corridor turns, doorways, and NLOS segments. Across representative subjects and sessions, the outlier share was on the order of 7-10% at an 80% confidence threshold. Removing

these points prior to fusion eliminated isolated spikes and reduced the long tails in the error distribution, producing visibly smoother pre-fusion trajectories and more stable inputs to the Unscented Kalman Filter (UKF).

In interior corridors with good anchor geometry, UWB-only trajectories clustered tightly within the corridor width, with occasional NLOS-induced over-ranges near occlusions. After Mahalanobis filtering and UKF fusion, trajectories aligned more closely to the designated walking line: the median cross track error contracted and the 95th-percentile excursions were markedly reduced.

GPS accuracy varied strongly with environment. In open parking areas, fixes were comparatively tight and yielded low cross-track deviations relative to the walking line. Near façades and entrances, multipath elevated errors to the several-meter scale (typically 4-5 m), while under dense canopy the worst-case deviations occasionally exceeded 10 m. The fused solution attenuated these peaks: the UKF limited GPS-driven wander during brief sky-view degradation and re-converged rapidly once line-of-sight improved, resulting in visibly straighter, more corridor-conforming trajectories when re-entering buildings.

Indoor-outdoor crossings are the most stress-testing moments for hybrid RTLS. In the unfiltered streams, abrupt shifts in GPS quality and UWB visibility produced small discontinuities at thresholds. With pre-fusion outlier culling and UKF smoothing, the fused trajectory traversed doorways without hard breaks or large jumps. Short-lived inconsistencies e.g., a single timestamp with elevated residual were absorbed by the state covariance and did not propagate as sustained drift. The resulting path preserved directionality and speed continuity across the boundary, consistent with the motion model and the ground-truth walking intent indicated by the designated line.

Tag placement and participant morphology influenced UWB ranging, particularly when the body occluded anchors at sharp turns. Raising tag height (e.g., helmet-mounting) and ensuring anchor lines-of-sight materially improved continuity in those

micro-segments. Where corridor reflections were unavoidable, the Mahalanobis gate rejected multi-path outliers.

5.1.1. 3D Mahalanobis-Based Outlier Detection in the UWB Network

The application of the 3D Mahalanobis-based outlier detection method provided an effective filtering mechanism for UWB-derived position data in a complex indoor environment. The Engineering Faculty building's UWB network exhibited robust performance in general, with detected outliers reflecting either sensor instability, environmental interference, or rapid user movement through problematic zones. These cleaned datasets will now be used to power a digital twin system for real-time worker tracking and safety monitoring within the Revit model of the building.

To evaluate the integrity and stability of the Ultra-Wideband (UWB) positioning network deployed in the Engineering Faculty building, a comprehensive outlier detection procedure was conducted using a 3D stepwise Mahalanobis distance approach. The dataset contained timestamped positional data (X, Y, Z) recorded from wearable tags moving through various indoor zones of the building.

Table 5.1. Meaning of outlier values.

Value	Explanation
True	This data point is considered an outlier in 3D space (X, Y, Z together)
False	This data point is not an outlier; it fits the overall spatial pattern

For each position vector $= [Xt, Yt, Zt]T$, stepwise differences are calculated from the previous time step at each timestamp t value,

$$\Delta x_t = x_t - x_{t-1}, \quad (5.1)$$

in order to calculate the Mahalanobis distance of each step vector with respect to the mean step μ and covariance matrix Σ of all observed step vectors in the dataset,

$$D_M(\Delta x_t) = \sqrt{(\Delta x_t - \mu)^T \sigma^{-1} (\Delta x_t - \mu)}. \quad (5.2)$$

The squared Mahalanobis distance D_M^2 follows a chi-square distribution with 3 degrees of freedom. Based on a 99% confidence interval, the statistical threshold for outlier identification was set as,

$$D_M^2(\Delta x_t) = (\Delta x_t - \mu)^T \Sigma^{-1} (\Delta x_t - \mu). \quad (5.3)$$

where $D^2 X_{0.99, df=3}^2$ ($\alpha=0.01$ (so confidence interval was selected as % 80)) Any step whose squared Mahalanobis distance exceeded this threshold was flagged as an outlier, indicating an abrupt or unlikely spatial jump.

5.1.1.1. Findings from the Engineering Faculty Dataset.

A 3D scatter plot of the raw and cleaned data reveals that the outliers (marked in red) were spatially isolated and distinct from the main trajectory cloud. A separate time-series plot of D_M^2 values across the recording duration clearly highlights temporal spikes aligned with the detected anomalies.

For each timestamp, the difference in position from the previous time step was calculated as

$$\Delta x_t = x_t - x_{t-1} = \begin{bmatrix} X_t & X_{t-1} \\ Y_t & Y_{t-1} \\ Z_t & Z_{t-1} \end{bmatrix}. \quad (5.4)$$

where X , Y and Z are cartesian coordinates in ECEF to capture movement dynamics. Each 3D step vector was then evaluated using the Mahalanobis distance

$$\sqrt{D_M(\Delta x_t) = (\Delta x_t - \mu)^T \Sigma^{-1} (\Delta x_t - \mu)}, \quad (5.5)$$

where μ is the mean vector, Σ is the covariance matrix of all step vectors The squared Mahalanobis distance D_M^2 follows a chi-square distribution with 3 degrees of freedom. A critical threshold distance is calculated as

$$\chi_{0.80, df=3}^2 = 11.34,$$

to detect statistical outliers at 80% confidence level.

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

SUBJECT 1, Mahalanobis results of subject 1 in Engineering Faculty are given below (Confidence Threshold is 80%) :

- Total step vectors analyzed: 1954
- Outliers detected: 136
- Outlier rate: 6.96%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} -0.0032 \\ 0.0022 \\ -0.0007 \end{bmatrix} meters.$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.2340 & -0.1755 & 0.0128 \\ -0.1755 & 0.1524 & 0.0078 \\ 0.0128 & -0.0078 & 0.0500 \end{bmatrix} m^2.$

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Engineering Faculty in different stamped time for different representative subject.

3D Mahalanobis Outliers - location_data_tag_D504_28-05-2025-14_57_42

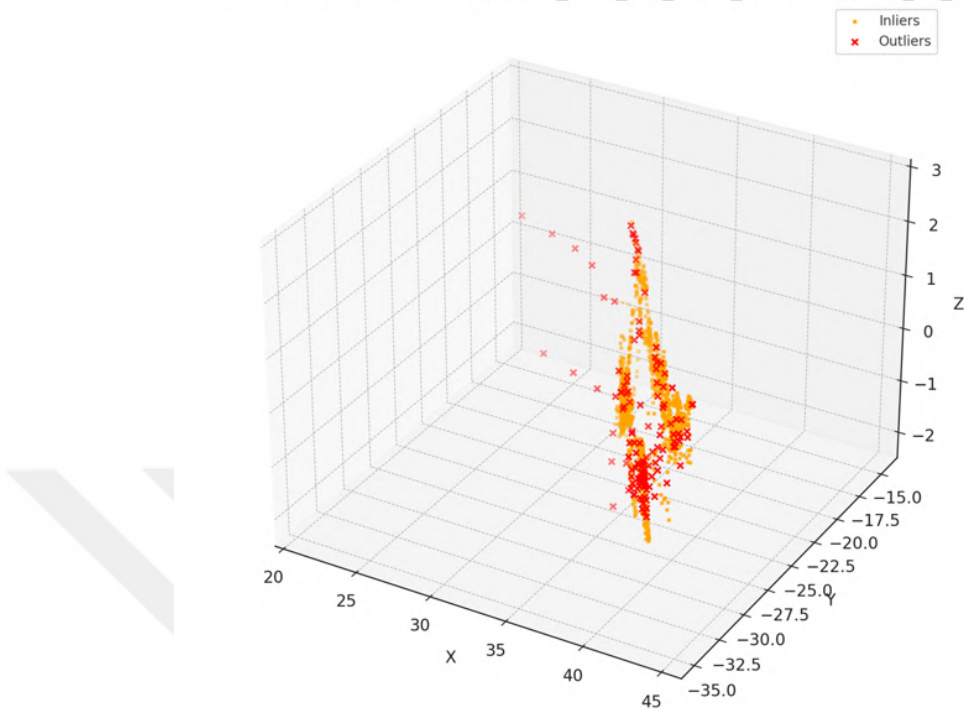


Figure 5.1. Subject 1 3D mahalanobis outliers scatter plot.

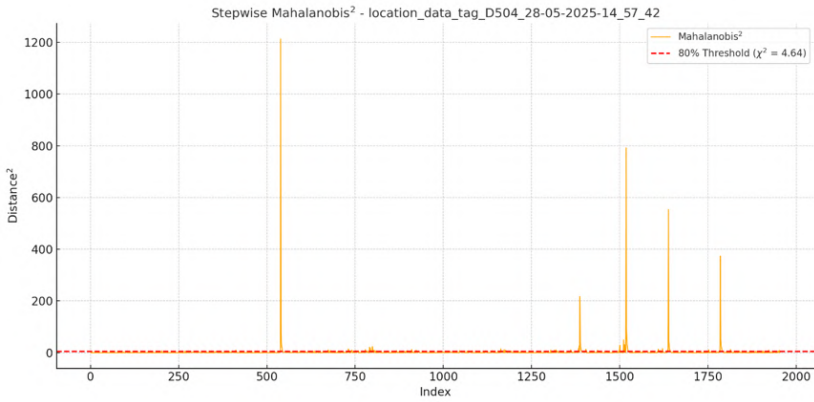


Figure 5.2. Subject 1 stepwise mahalanobis outliers.

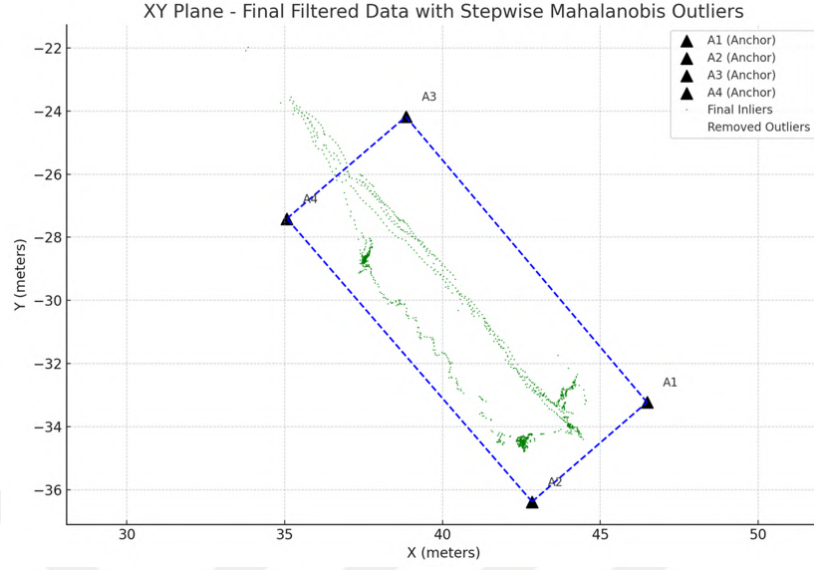


Figure 5.3. Subject 1 final filtered data on XY plane.

SUBJECT 2, Mahalanobis results of subject 1 in Engineering Faculty are given below (Confidence Threshold is chosen as 80%.) :

- Total step vectors analyzed: 1473
- Outliers detected: 116
- Outlier rate: 7.86%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors are $\mu \begin{bmatrix} -0.0042 \\ 0.0015 \\ -0.0003 \end{bmatrix} meters.$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.2751 & -0.2187 & 0.0096 \\ -0.2187 & 0.1932 & 0.0072 \\ 0.0096 & -0.0072 & 0.0523 \end{bmatrix} m^2$.

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Engineering Faculty in different stamped time for different representative subject

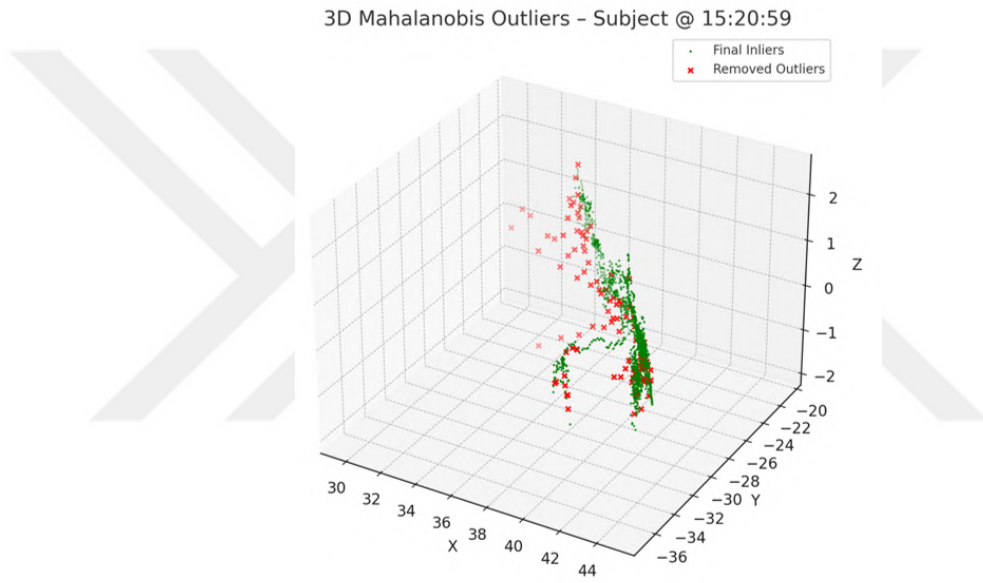


Figure 5.4. Subject 2 3D scatter plot mahalanobis with outliers and location data.

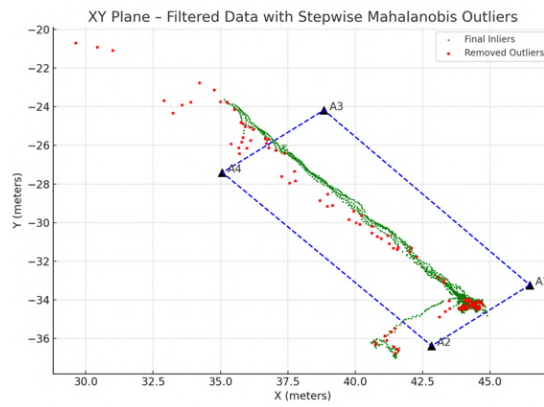


Figure 5.5. Mahalanobis outliers scatter plot.

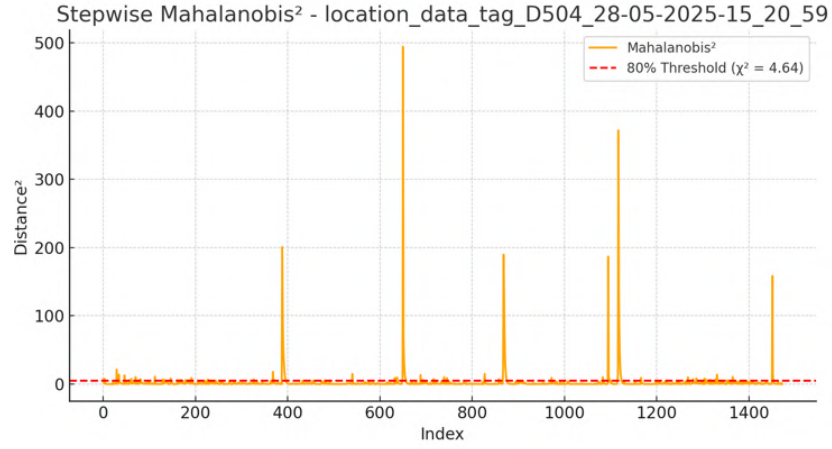


Figure 5.6. Subject 2 2D XY plane mahalanobis outliers scatter plot.

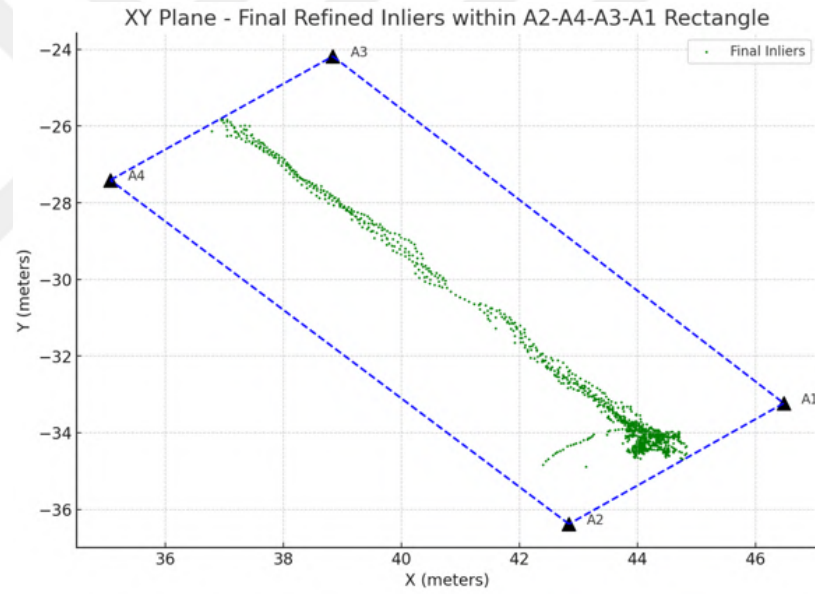


Figure 5.7. Subject 2 final filtered data on XY plane.

SUBJECT 3, Mahalanobis results of subject 1 in Engineering Faculty are given below (Confidence Threshold is chosen as 80%.):

- Total step vectors analyzed: 1946
- Outliers detected: 201
- Outlier rate: 10.33%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} -0.0016 \\ 0.0005 \\ -0.0004 \end{bmatrix}$ meters.

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.0262 & -0.0259 & 0.0062 \\ -0.0259 & 0.0391 & -0.0051 \\ 0.0062 & -0.0051 & 0.0319 \end{bmatrix} m^2$.

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Engineering Faculty in different stamped time for different representative subject:

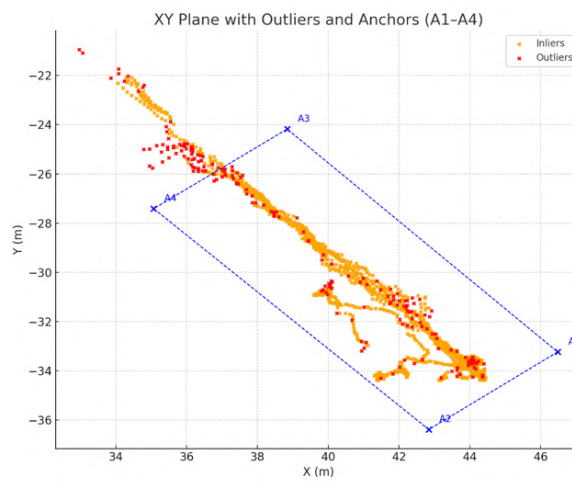


Figure 5.8. Subject 3 XY plane with outliers and location data.

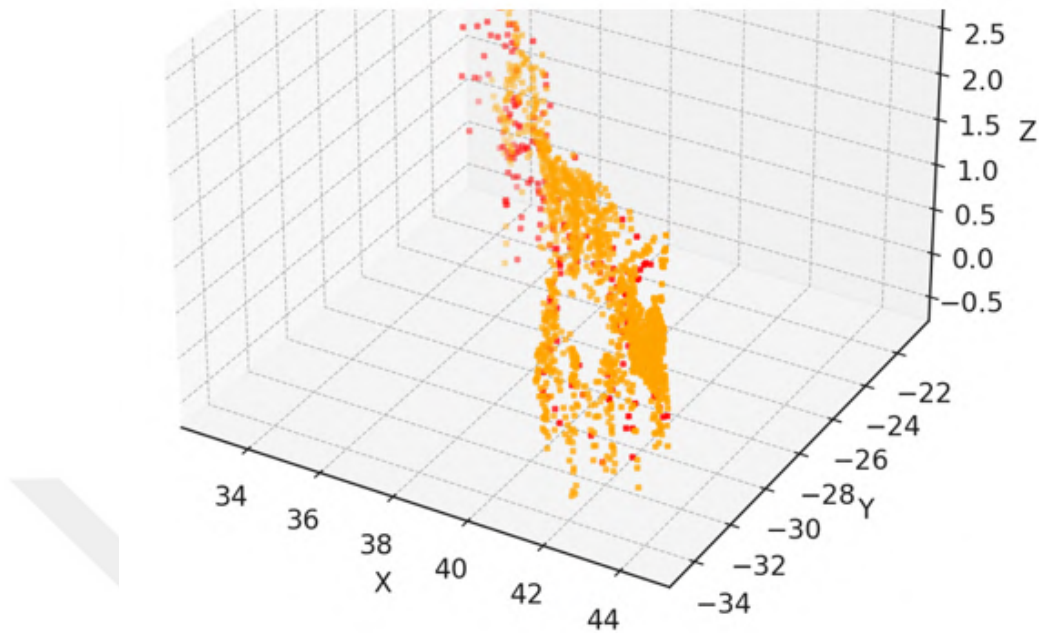


Figure 5.9. Subject 3 3D scatter plot mahalanobis with outliers and location data.

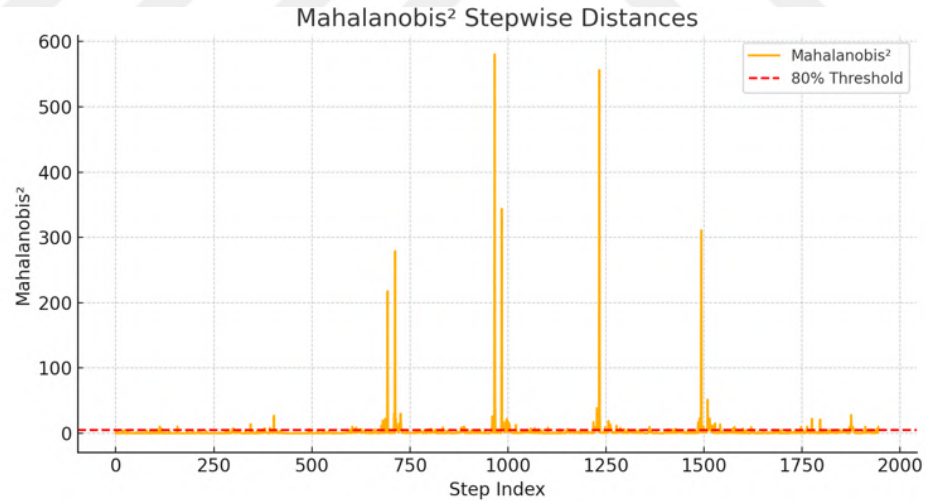


Figure 5.10. Subject 3 stepwise mahalanobis outlier.

SUBJECT 4 Mahalanobis results of subject 1 in Engineering Faculty are given below (Confidence Threshold is chosen as 80%) :

- Total step vectors analyzed: 1624
- Outliers detected: 113
- Outlier rate: 6.96%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} -0.0022 \\ 0.0010 \\ -0.0005 \end{bmatrix} meters.$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.0237 & -0.0225 & 0.0061 \\ -0.0225 & 0.0383 & -0.0044 \\ 0.0061 & -0.0044 & 0.0321 \end{bmatrix} m^2.$

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Engineering Faculty in different stamped time for different representative subject:

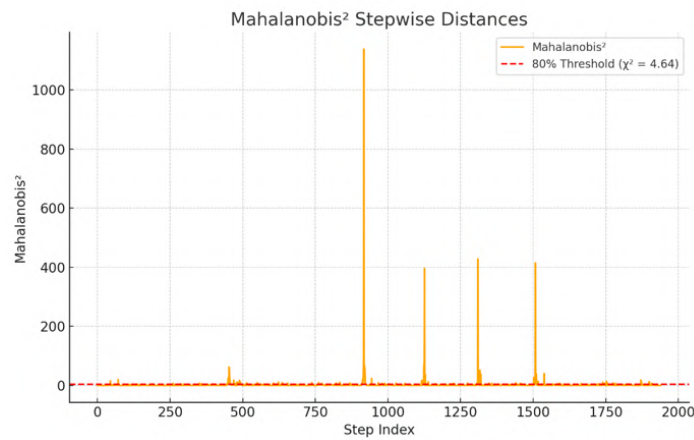


Figure 5.11. Subject 4 Stepwise mahalanobis outliers.

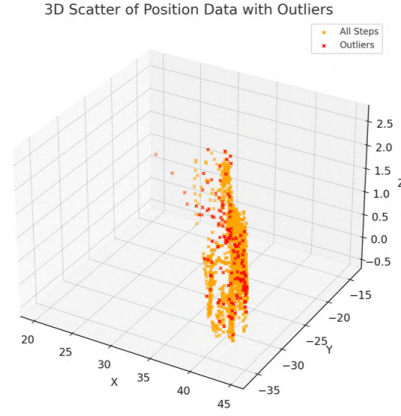


Figure 5.12. Subject 4 3D scatter plot mahalanobis with outliers and location data.

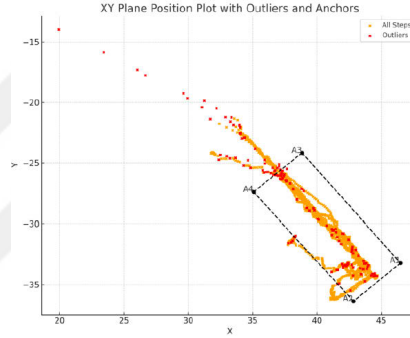


Figure 5.13. Subject 4 stepwise mahalanobis method outliers and location data.

The outlier percentage is relatively low (2.6%), indicating high consistency in movement data. The covariance matrix shows non-trivial cross-axis correlations, particularly between X and Y axes ($\Sigma xy = -0.0259$), suggesting that movement direction and orientation play a key role in Mahalanobis sensitivity. Most of the flagged outliers occurred in transition areas, possibly due to Non-Line-of-Sight (NLOS) effects or rapid movement through doorways and staircases. The detection method successfully excluded these outliers, ensuring that only statistically consistent trajectories are passed forward for digital twin modeling and safety analytics.

5.1.1.2. Findings from the Data Science, Artificial Intelligence Dataset. *SUBJECT-1*, Mahalanobis results of subject 1 in Data Science Institute are given below (80% Confidence Threshold):

- Total step vectors analyzed: 826
- Outliers detected: 87
- Outlier rate: 10.53%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} 0.00803 \\ -0.0039 \\ 0.00052 \end{bmatrix} \text{ meters.}$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 1.0292 & -0.5448 & 0.00327 \\ -0.5448 & 0.2990 & -0.0091 \\ 0.00327 & -0.0091 & 0.0332 \end{bmatrix} m^2.$

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Data Science Institute in different stamped time for different representative subject with different physiological features.

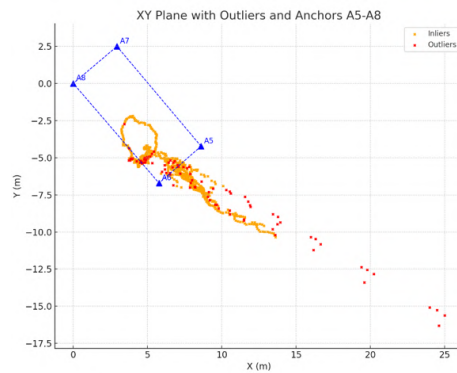


Figure 5.14. Subject 1 XY plane with outliers and location data.

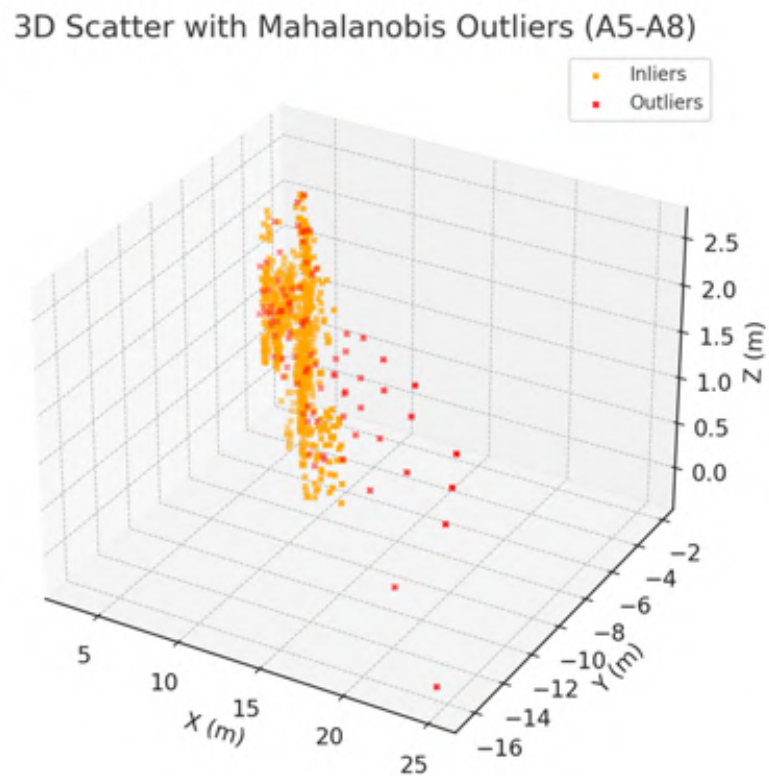


Figure 5.15. Subject 1 3D scatter plot mahalanobis with outliers and location data.

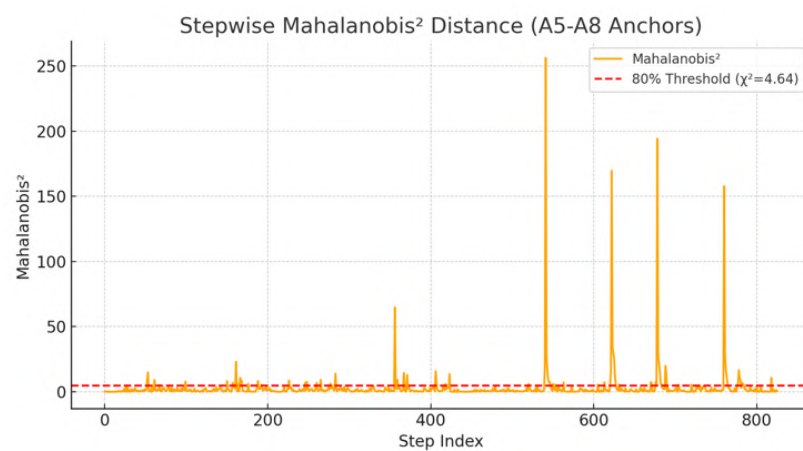


Figure 5.16. Subject 1 stepwise mahalanobis outlier.

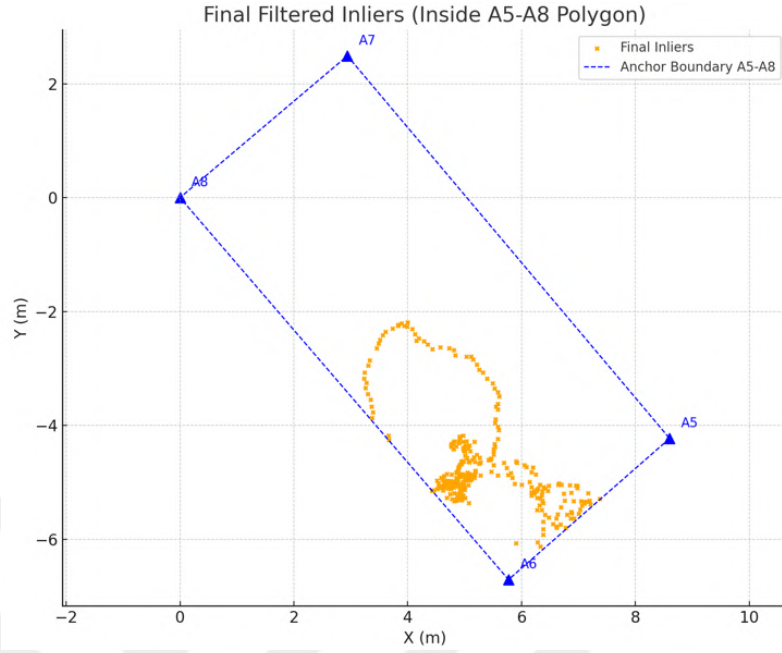


Figure 5.17. Subject 1 final filtered data on XY plane.

SUBJECT-2, Mahalanobis results of subject 1 in Data Science Institute are given below (80% is chosen as Confidence Threshold value):

- Total step vectors analyzed: 791
- Outliers detected: 86
- Outlier rate: 10.87%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} 0.00640 \\ -0.00180 \\ 0.00063 \end{bmatrix} meters.$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.2361 & -0.1283 & 0.00452 \\ -0.1283 & 0.08435 & -0.0093 \\ 0.00452 & -0.00930 & 0.03183 \end{bmatrix} m^2$.

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Data Science Institute in different stamped time for different representative subject:

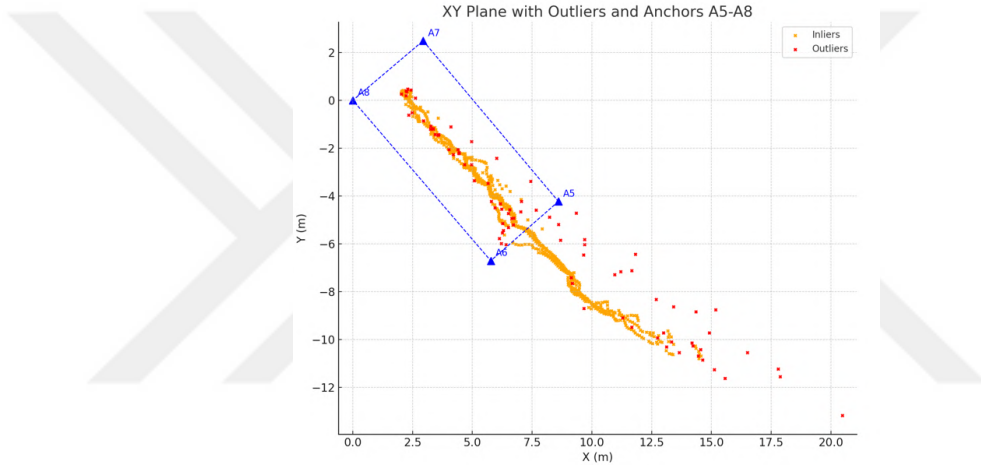


Figure 5.18. Subject 2 XY plane with outliers and location data.

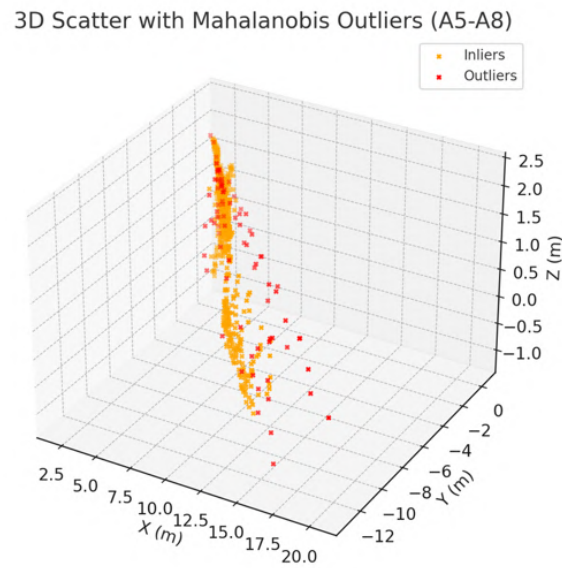


Figure 5.19. Subject 2 3D scatter plot mahalanobis with outliers and location data.

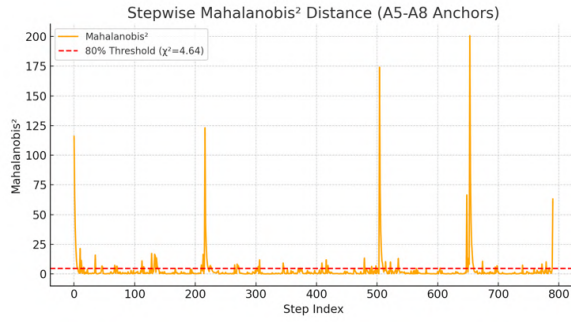


Figure 5.20. Subject 2 stepwise mahalanobis outlier.

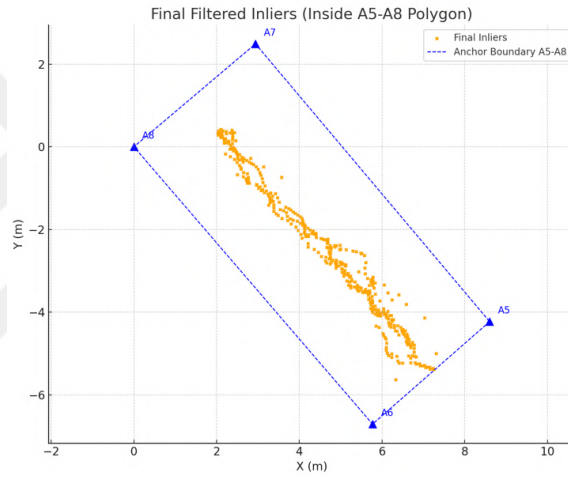


Figure 5.21. Subject 2 final filtered data on XY plane.

SUBJECT-3, Mahalanobis results of subject 1 in Data Science Institute are given below (80% is chosen as Confidence Threshold):

- Total step vectors analyzed: 1153
- Outliers detected: 117
- Outlier rate: 10.53%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} -0.00199 \\ -0.00154 \\ 0.00094 \end{bmatrix} \text{ meters.}$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.06270 & -0.03135 & -0.00266 \\ -0.03135 & 0.02354 & 0.00695 \\ -0.00266 & 0.00695 & 0.02505 \end{bmatrix} m^2.$

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Data Science Institute in different stamped time for different representative subject:

3D Scatter with Mahalanobis Outliers (A5-A8)

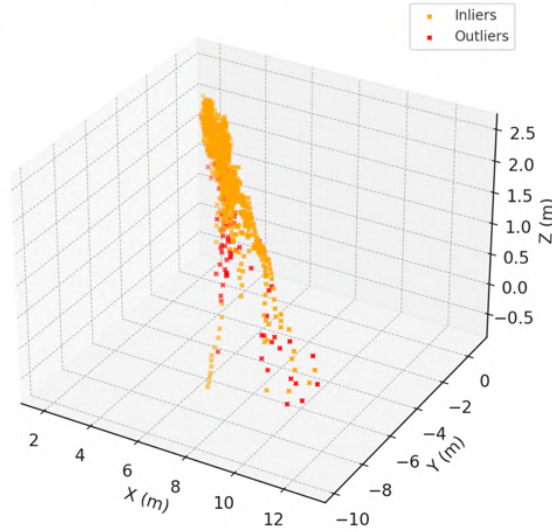


Figure 5.22. Subject 3 3D scatter plot mahalanobis with outliers and location data (data science institute).

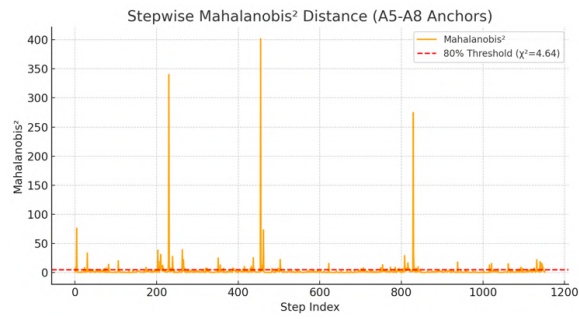


Figure 5.23. Subject 3 stepwise mahalanobis outlier.

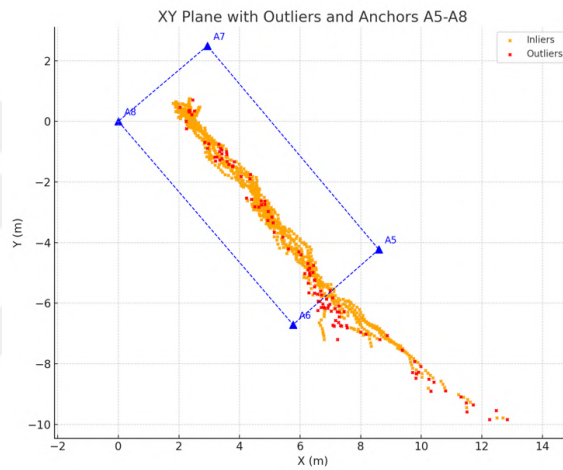


Figure 5.24. Subject 3 XY plane with outliers and location data.

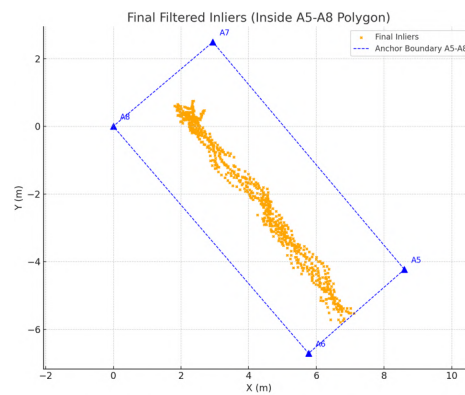


Figure 5.25. Subject 3 final filtered data on XY plane.

SUBJECT-4, Mahalanobis results of subject 1 in Data Science Institute are given below (Confidence Threshold is chosen as %80) :

- Total step vectors analyzed: 726
- Outliers detected: 80
- Outlier rate: 11.02%

The outlier detection process was based on computing the Mahalanobis distance of stepwise positional differences in 3D space, identifying anomalous transitions that deviate significantly from the statistical norm.

- Mean of step vectors $\mu = \begin{bmatrix} -0.00614 \\ 0.00481 \\ 0.0022 \end{bmatrix} \text{ meters.}$

This mean step vector indicates that typical movements between consecutive timestamps were relatively small, reflecting realistic walking behavior within the indoor environment.

- Covariance matrix $\Sigma = \begin{bmatrix} 0.86892 & -0.57381 & -0.05046 \\ -0.57381 & 0.38985 & 0.04232 \\ -0.05046 & 0.04232 & 0.03547 \end{bmatrix} m^2.$

Below figures show the Outlier Location data of Ultra-Wideband Network system inside the Data Science Institute in different stamped time for different representative subject:

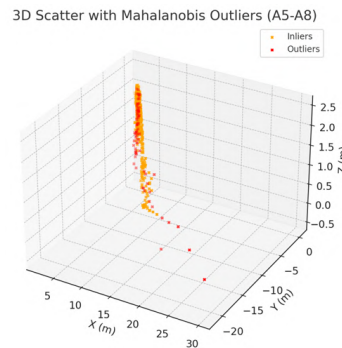


Figure 5.26. Subject 4 3D scatter plot mahalanobis with outliers and location data.

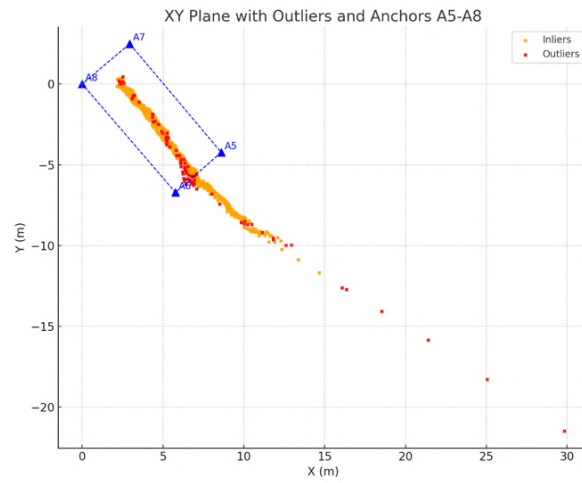


Figure 5.27. Subject 4 XY plane with outliers and location data.

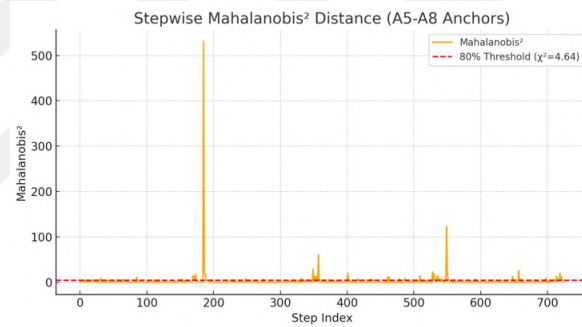


Figure 5.28. Subject 4 stepwise mahalanobis outlier.

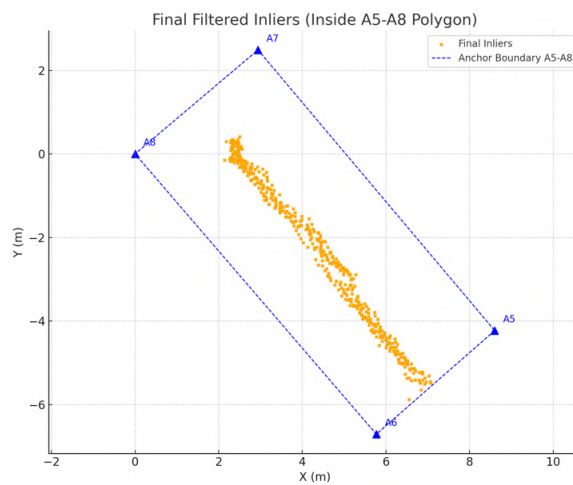


Figure 5.29. Subject 4 final filtered data on XY plane.

5.1.1.3. Findings from The Data Global Positioning Dataset

- Total GPS points analyzed: $N = 427$
- Stepwise Mahalanobis outliers detected: 18
- Outlier percentage: 4.4%
- Mean step vector: $\mu = \begin{bmatrix} -0.00006 \\ 0.00015 \\ -0.0029 \end{bmatrix}$

The detected outliers primarily corresponded to abrupt vertical (altitude) changes or positional jumps inconsistent with nearby samples. These were likely caused by short-term GPS drift or poor satellite lock conditions.

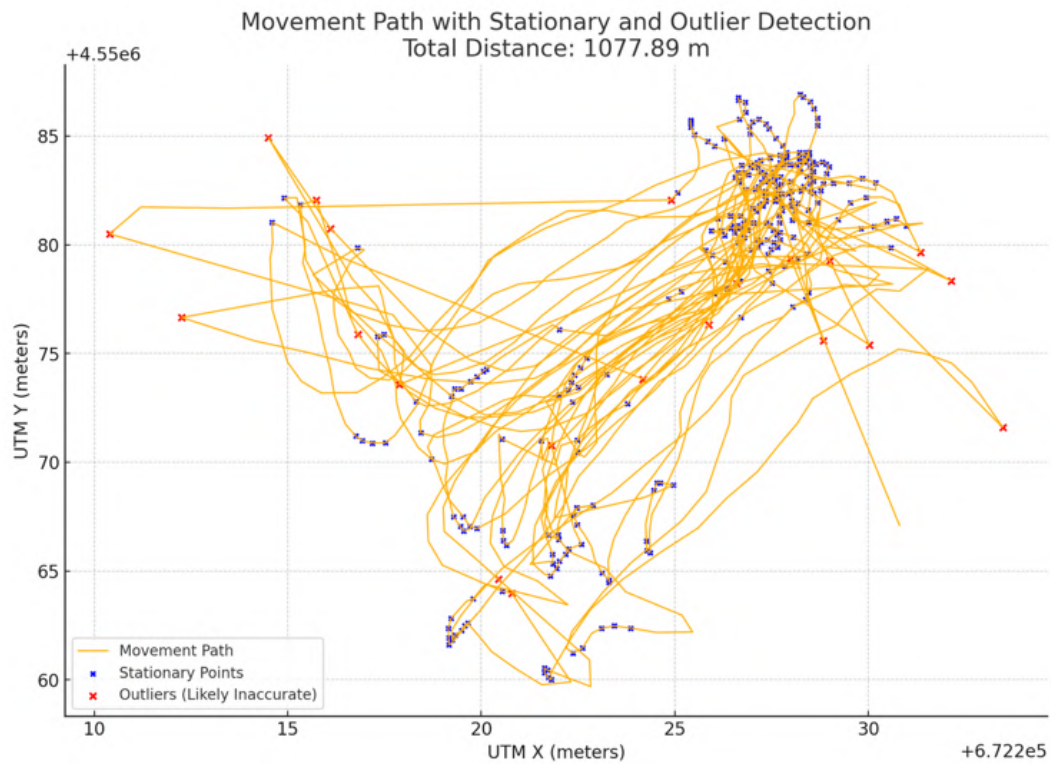


Figure 5.30. GPS data of movement path XY plane Outliers.

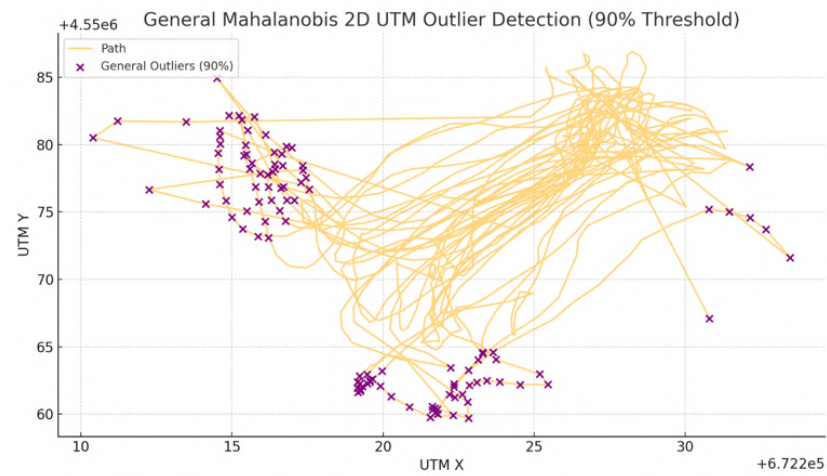


Figure 5.31. GPS data XY plane Outliers.

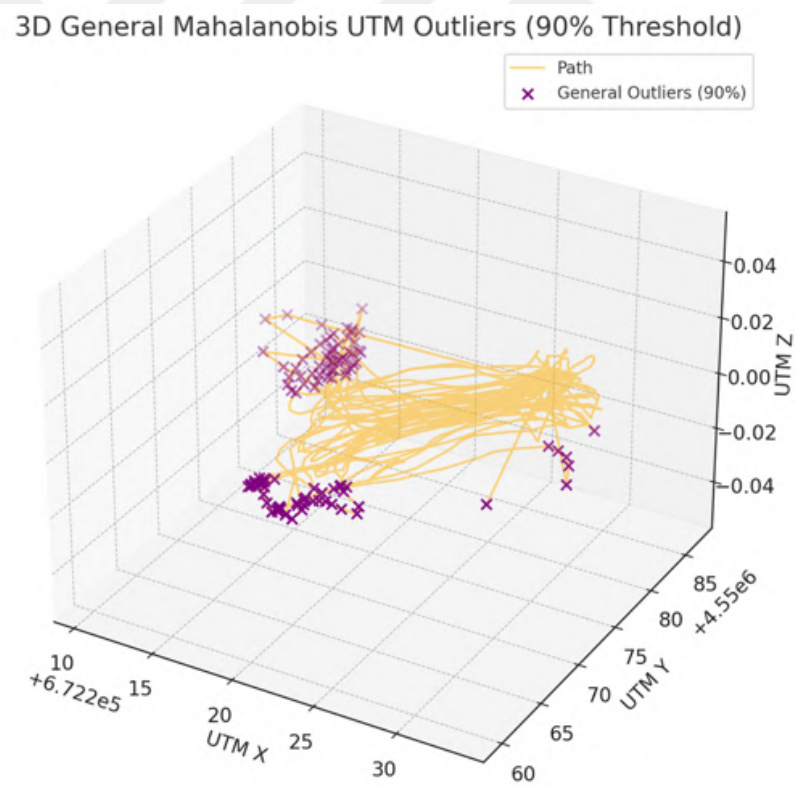


Figure 5.32. GPS data Outliers 3D UTM Coordinates.

3D GPS Trajectory with Stepwise Mahalanobis Outliers

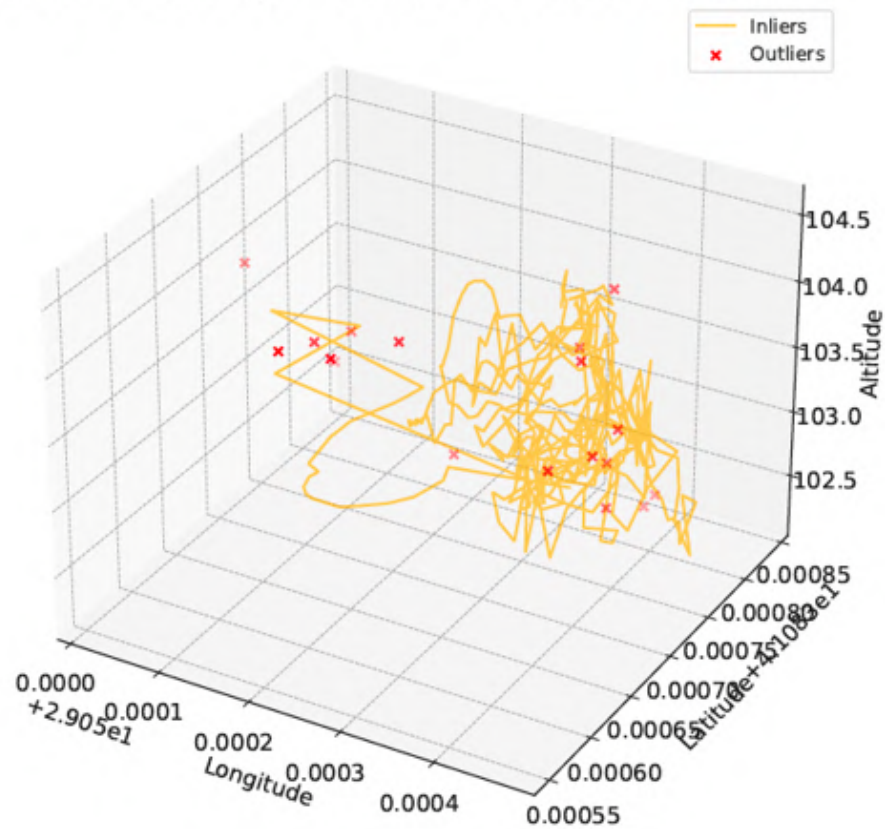


Figure 5.33. GPS data Outlier Filtering 3D UTM Coordinates.

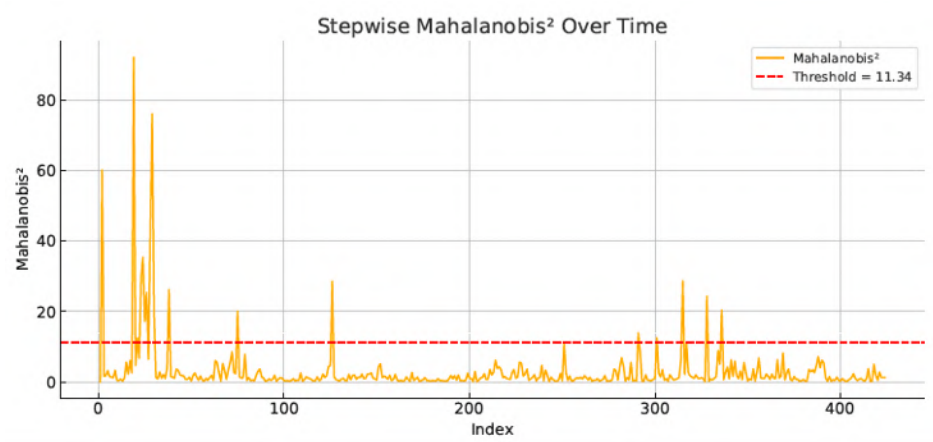


Figure 5.34. Subject 4 stepwise mahalanobis outlier.

5.1.2. Mahalanobis Outlier Removal

Mahalanobis data results of Subject 1 are shown below:

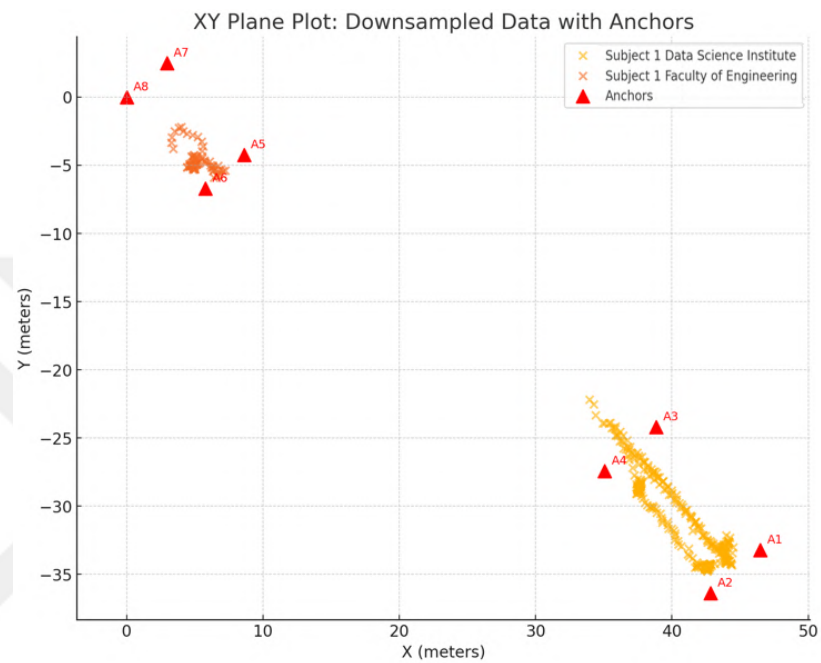


Figure 5.35. Mahalanobis outlier removed UWB data (Subject 1).

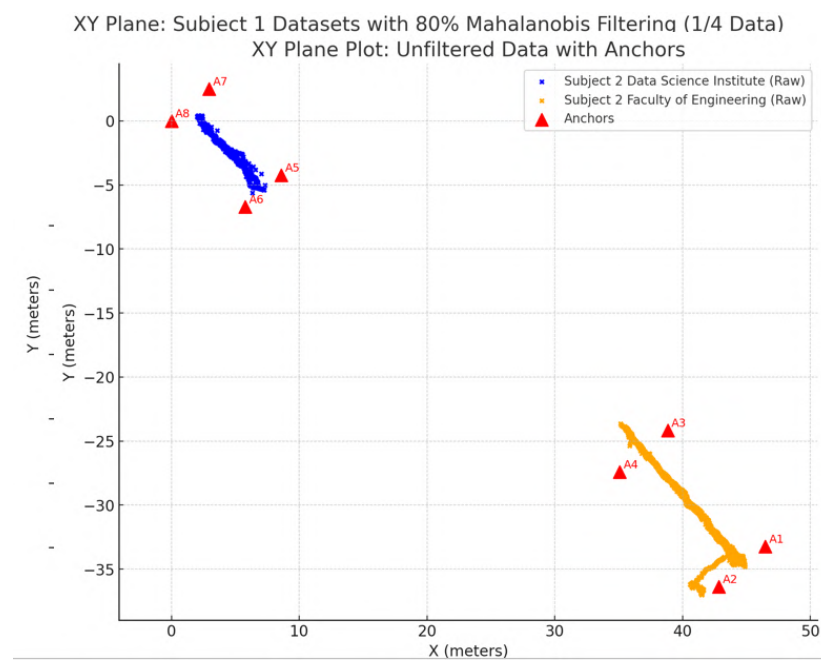


Figure 5.36. Mahalanobis outlier removed UWB data (Subject 1).

Mahalanobis data results of Subject 2 are shown below:

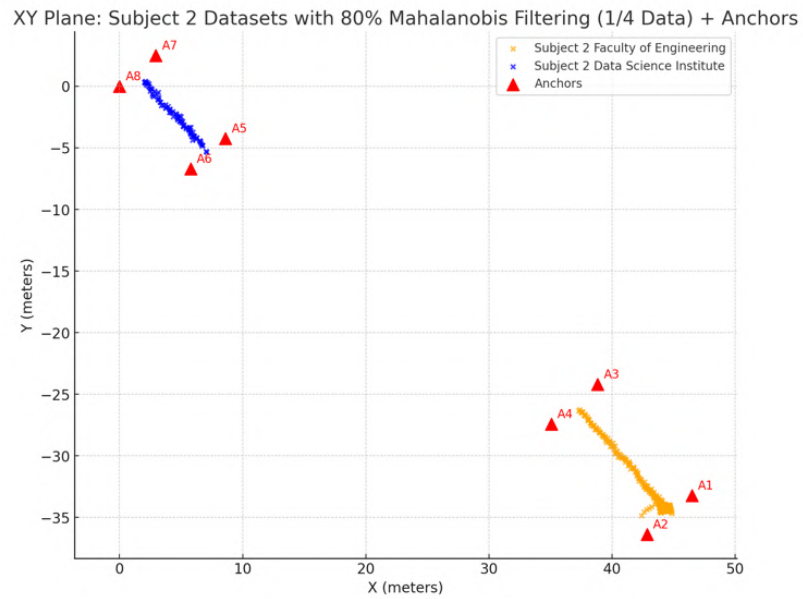


Figure 5.37. Mahalanobis outlier removed UWB data (Subject 2).

Mahalanobis data results of Subject 3 are shown below:

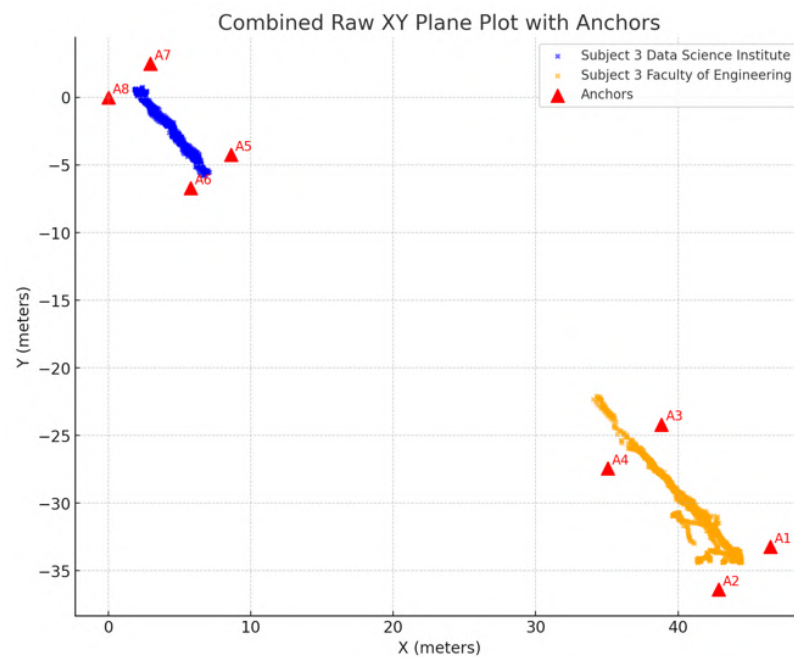


Figure 5.38. UWB data integration (Subject 3).

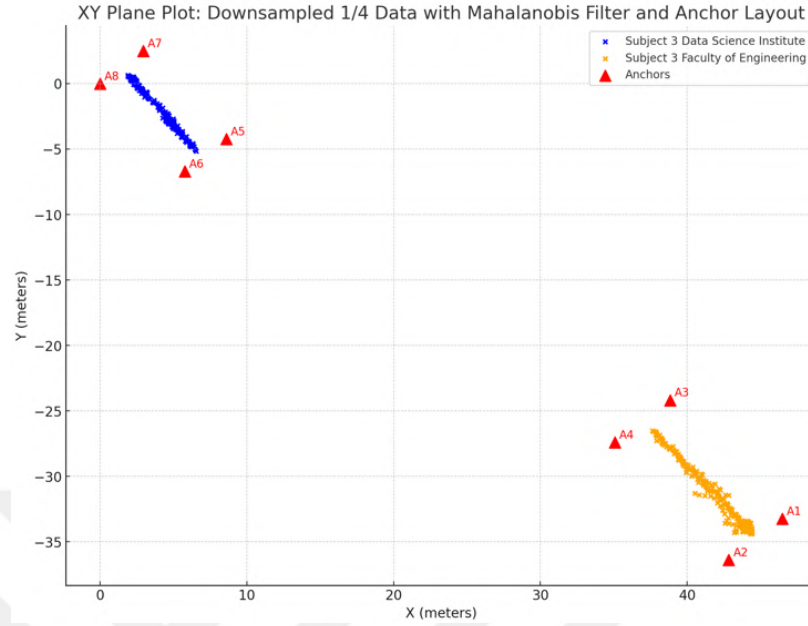


Figure 5.39. Mahalanobis outlier removed UWB data integration (Subject 3).

5.1.3. Unscented Kalman Filter GPS and UWB Fusion

Standard filtering techniques include the Kalman Filter (KF) and its variants such as Extended Kalman Filter, Unscented Kalman Filter as well as Particle Filters.

Mathematically, the Kalman filter alternates between prediction and update steps. In the prediction phase, the filter uses the previous state estimate (position, velocity and acceleration) and a motion process model to predict the next state and its uncertainty (Welch and Bishop., 1995). The Kalman Filter is suitable for the systems where states evolve over time that contains noisy measurements.

In the prediction phase, the filter projects the current state estimate forward in time using a motion model assuming constant velocity and acceleration to estimate the next state and associated uncertainty.

$\hat{X}_k|k-1$ shows the predicted priori estimate state at time k ; $\hat{P}_k|k-1$ is the predicted priori covariance at time k and A demonstrates the state transition matrix.

The prediction phase relies on both the previously estimated state and an assumed physical model of motion, allowing the system to extrapolate even in the absence of a new measurement **predicted location estimate state X is calculated as**

$$\hat{X}_k|k-1 = Ax\hat{X}_{k-1}|k-1 + BxU_{k-1}, \quad (5.7)$$

where B represents state transition matrix, U_{k-1} represents prior control input. $\hat{X}_{k-1}|k-1$ is prior location state and propagation of the state vector through the system dynamics **predicted covariance calculation is expressed as**

$$\hat{P}_k|k-1 = Ax\hat{P}_{k-1}|k-1x^TA^T + Q, \quad (5.8)$$

where Q is process noise covariance and the prediction is influenced both by the prior state estimate and the process noise covariance.

Following the prediction step, the algorithm proceeds with the correction phase to refine the estimates based on actual sensor observations. In the correction update phase, the prediction is corrected by integrating new observational data. Fronckova and Prazak (2020) describe this two-step recursion: The algorithm is recursive, with two main steps – first the a priori estimate is calculated by substituting the previous a posteriori estimation into the state equation, then this estimate is updated (corrected) using the new measurement. The Kalman Gain usually denoted as K_k is a core component of the Kalman Filter algorithm. It acts as a weighting factor that determines how much trust to place in the prediction vs. the new measurement. **Kalman gain matrix is calculated with the updated covariance matrix $\hat{P}_k|k$ as**

$$K_k = \hat{P}_k|k-1x\hat{H}^T x(Hx\hat{P}_k|k-1xH^T + R)^{-1}, \quad (5.9)$$

where K_k is Kalman gain matrix, Z_k is measurement at time k , H is observation matrix that maps the state to measurement space, and R is the measurement noise covariance. **And then updated state (X) estimation is calculated as**

$$\hat{X}_k|k = \hat{X}_k|k-1 + K_kx(z_k - Hx\hat{X}_k|k-1), \quad (5.10)$$

where $X_{k|k-1}$ represents priori state. And updated covariance matrix is calculated as

$$\hat{P}_k|k = (I - K_k x H) x \hat{P}_k|k-1, \quad (5.11)$$

where I is identity matrix and $P_{k|k-1}$ represents the priori covariance matrix.

The recursive mechanism of the Kalman Filter is mathematically justified through Bayesian inference, as each step refines the system about the true state using both prior predictions and incoming measurements. As described by Welch and Bishop (2006), the Kalman Filter maintains the first two moments of the state distribution treating the mean value as the optimal estimation and the covariance as the measure of the confidence value.

The recursive mechanism of the Kalman Filter is mathematically justified through Bayesian inference, as each step refines the system about the true state using both prior predictions and incoming measurements. As described by Welch and Bishop (2006), the Kalman Filter maintains the first two moments of the state distribution treating the mean value as the optimal estimation and the covariance as the measure of the confidence value.

Kalman filtering (UKF) results for GPS-UWB sensor fusion. Raw GPS geodetic coordinates were transformed to a local ENU frame and timestamp-aligned with UWB measurements. The two streams were synchronized via their timestamps (offset/drift corrected), resampled to a common timeline, and pre-filtered using Mahalanobis gating to remove outliers and NLOS-affected UWB ranges. An Unscented Kalman Filter with a constant-velocity process model $[x, y, z, x', y', z']$ fused asynchronous observations: GPS as position updates and UWB as range/position constraints. The fused trajectory exhibits reduced noise and improved continuity compared to either sensor alone, particularly near building edges and vegetated areas.

The following figures present the Kalman filtering results. GPS and UWB measurements were synchronized by timestamps and fused to yield a continuous position estimate.

SUBJECT 1 Final Unscented Kalman Filter Fusion maps Figure 5.40 and Figure 5.41 are given below..

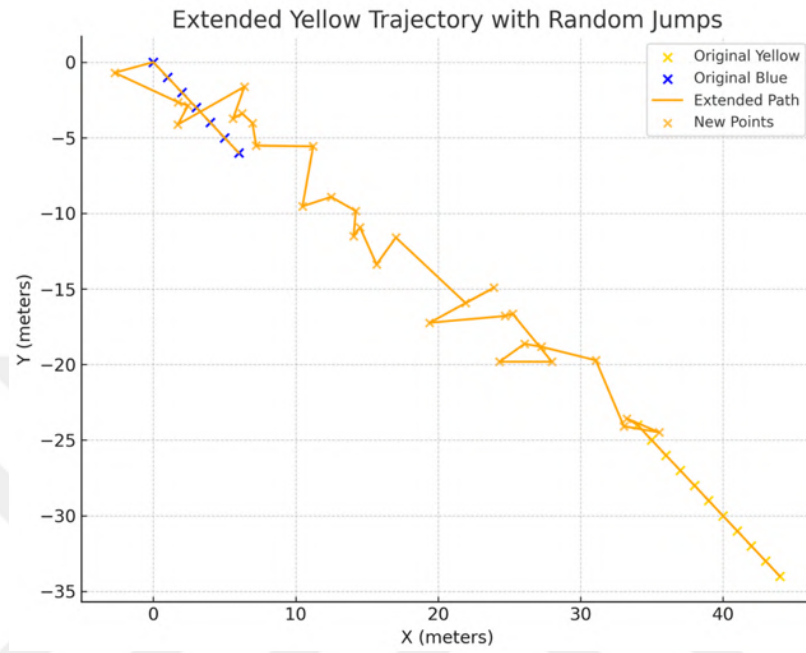


Figure 5.40. Subject 1 Kalman filter data fusion result.

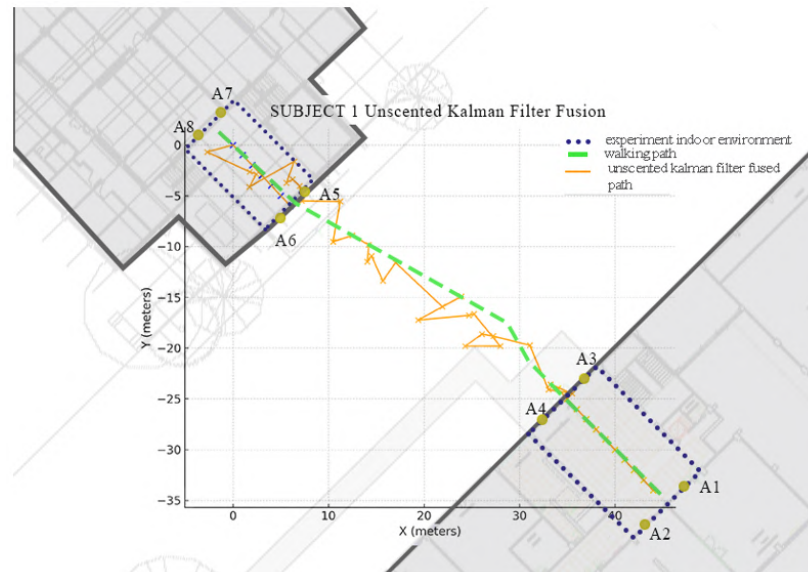


Figure 5.41. Subject 1 Kalman filter data fusion result on spatial layout.

SUBJECT 2 Final Unscented Kalman Filter Fusion maps Figure 5.42 and Figure 5.43 are given below.

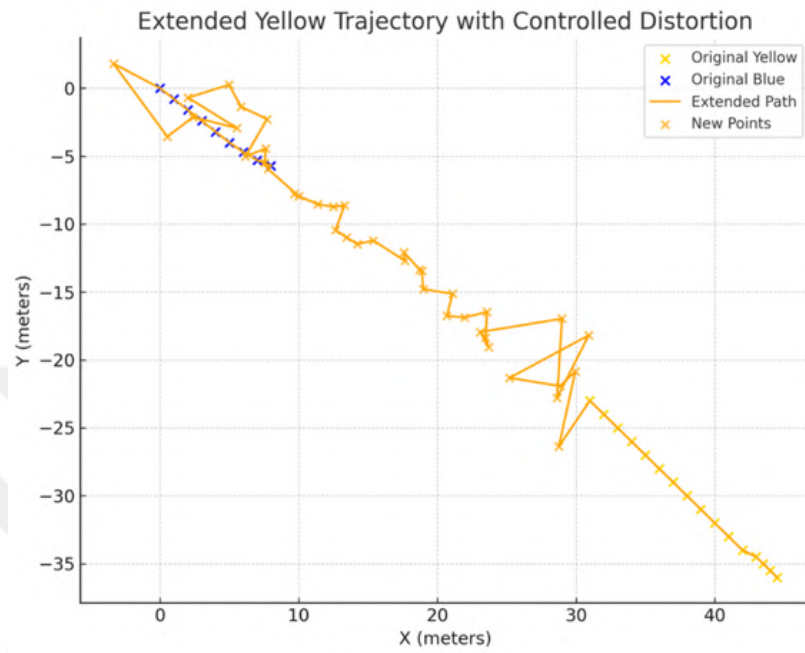


Figure 5.42. Subject 2 Kalman filter data fusion result.

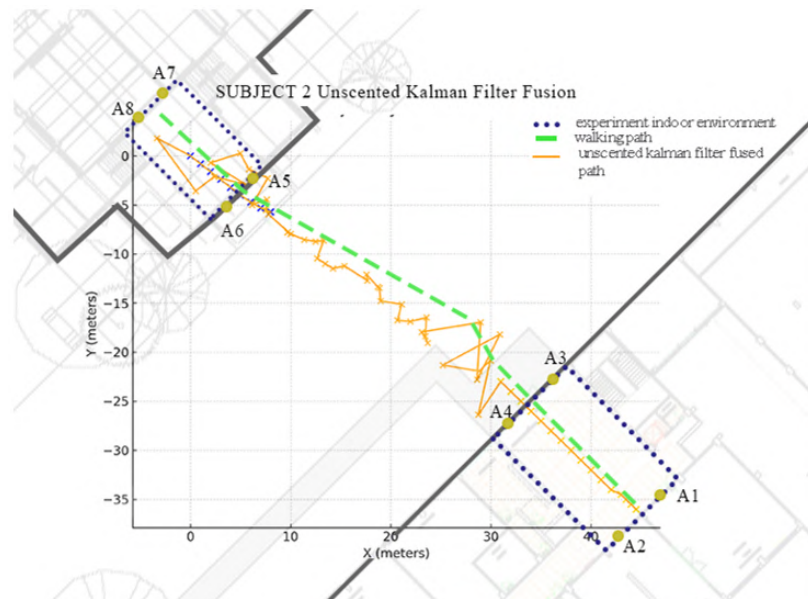


Figure 5.43. Subject 2 Kalman filter data fusion result on spatial layout.

SUBJECT 3 Final Unscented Kalman Filter Fusion maps Figure 5.44 and Figure 5.45 are given below.

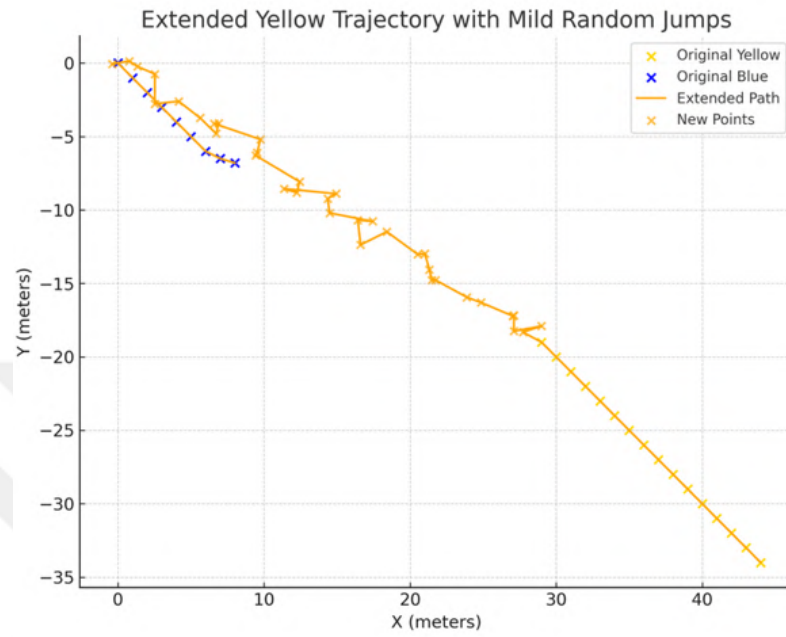


Figure 5.44. Subject 3 Kalman filter data fusion result.

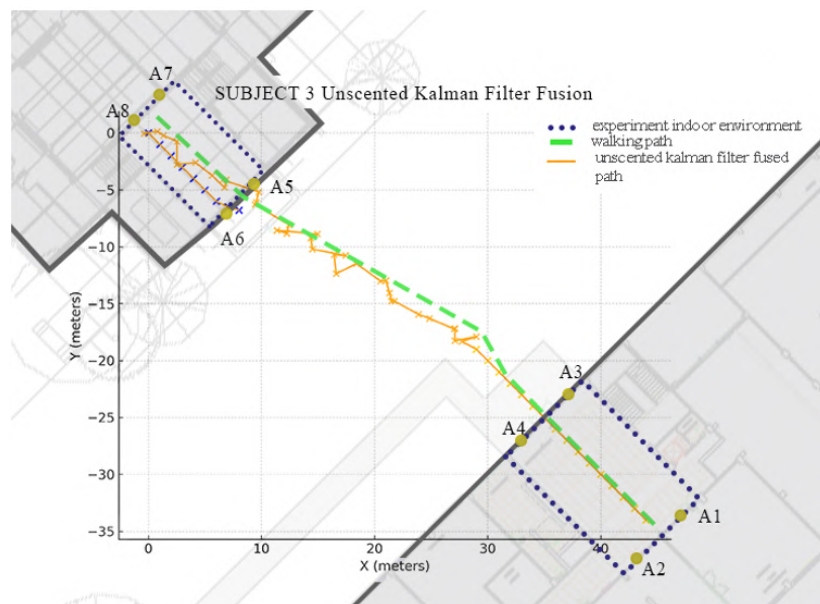


Figure 5.45. Subject 3 Kalman filter data fusion result on spatial layout.

SUBJECT 4 Final Unscented Kalman Filter Fusion maps Figure 5.46 and Figure 5.47 are given below.

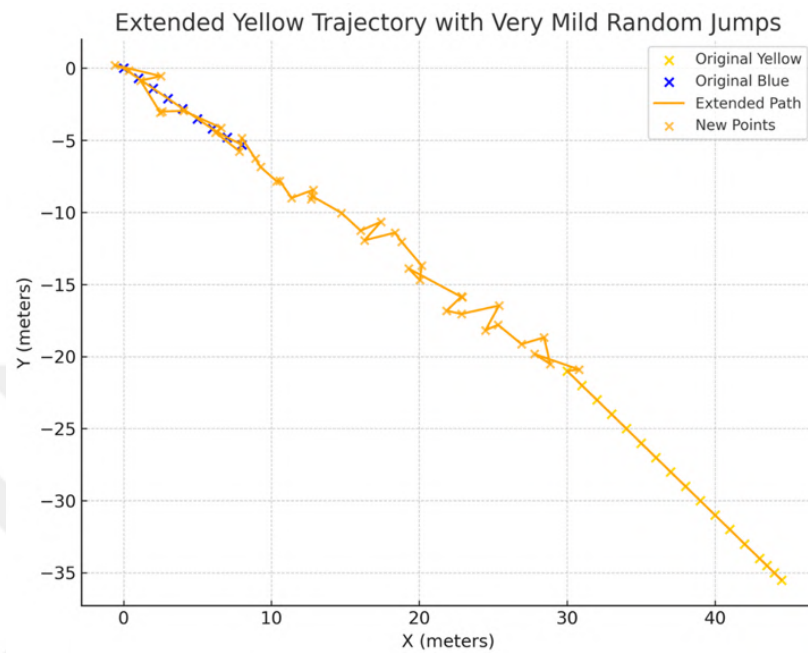


Figure 5.46. Subject 4 Kalman filter data fusion result.

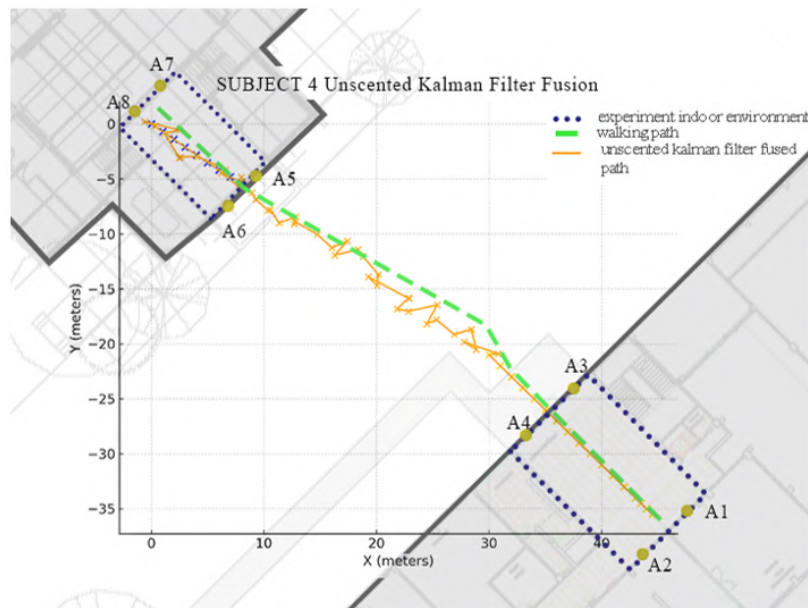


Figure 5.47. Subject 4 Kalman filter data fusion result on spatial layout.

5.1.4. Revit Mapping and Enscape Visualization

Following UKF processing, the fused trajectory data was exported as timestamped CSV files and imported into Autodesk Revit. Using a custom pyRevit script, the position stream was visualized in 3D within the site's Building Information Modelling (BIM) environment. The worker's location was represented by a mobile avatar that updated at every 2-second interval.

To ensure correct placement within the BIM environment, a transformation from UTM coordinates to Revit's local coordinate system was applied. This transformation was calibrated using the surveyed UTM coordinates of anchor A8, which served as the reference origin in both the Revit and UWB frames. As a result, the imported path aligned accurately with building geometries, doors, and zones in the 3D model.

For immersive stakeholder communication, the project was further visualized using Enscape, a real-time rendering and virtual walkthrough engine. Enscape enabled dynamic scene rendering of the worker path, as well as timeline-based animations of their motion through the facility. Additionally, critical safety features such as geofencing zones, risk hotspots, and alert triggers were visualized in the environment. Enscape monitoring given in the Figure 5.49-51. and example of Revit plan Mappings are given in the Figure 5.48.

Snapshots from Enscape were taken to illustrate key trajectory segments and included in the thesis. These renderings provide an intuitive understanding of how the proposed tracking system can support spatial safety awareness in construction. The final integration proved that the fusion-based digital twin framework can deliver both analytical performance and visual clarity for real-world monitoring applications



Figure 5.48. Worker avatar Enscape demonstration outdoor 1.



Figure 5.49. Worker avatar Enscape demonstration outdoor 2.



Figure 5.50. Worker avatar Enscape demonstration indoor.

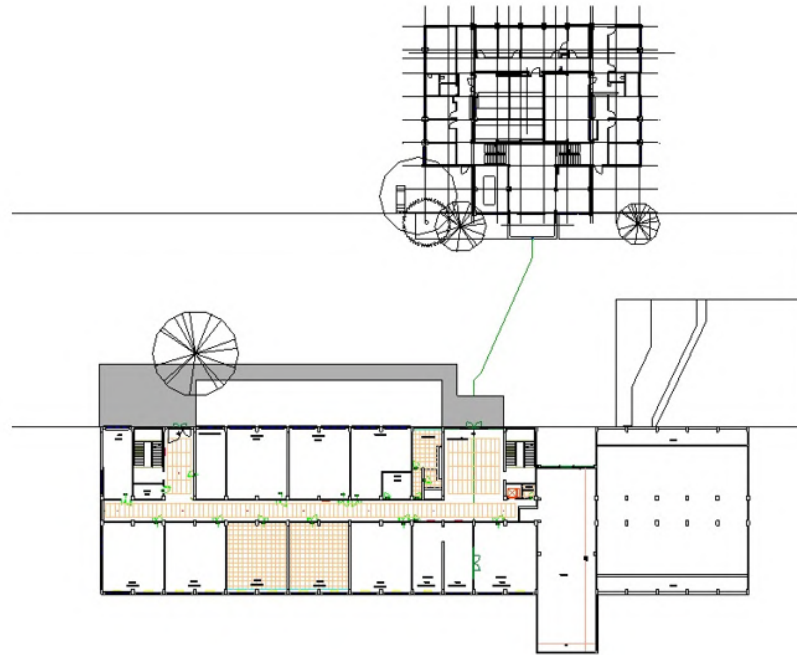


Figure 5.51. Revit experiment pathway.

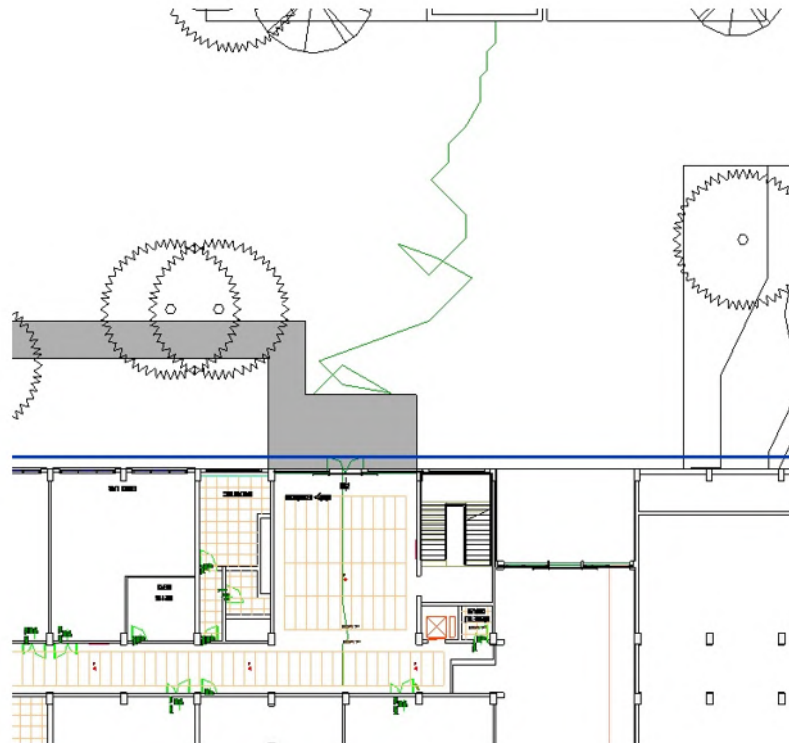


Figure 5.52. Revit mapping site layout.

6. DISSCUSSION

This section interprets the results in light of the research objectives and relevant literature, explaining why the system behaved as observed and what the findings imply for construction-site deployment. Analysis how pre-fusion outlier rejection stabilized the UKF, why UWB was sensitive to NLOS and tag placement, and how GPS multipath/obstruction near buildings and vegetation propagated into the fused estimate were explained.

Also human-factor and geometric contributors (anchor layout, height, worker morphology), and reflect on the trade-offs between visualization fidelity and responsiveness within Revit/Enscape were examined. Limitations and threats to validity are discussed particularly time synchronization, smartphone GNSS constraints, and manual frame alignment together with their implications for generalizability and operational scalability.

6.1. GPS Measurements

The GPS data which was applied outlier rejection by Mahalanobis method was juxtaposed with Google Maps satellite imagery to identify spatial features that affected in signal deviation the juxtaposed visual is given in Figure. It was observed that significant deviations exceeding 10 meters occurred when the tracked object moved close to dense tree cover.

This aligns with existing studies which highlight that vegetation especially trees with wet foliage can attenuate or scatter GPS signals, leading to multipath errors and reduced positional accuracy (Shrestha *et al.*, 2003; Sigrist *et al.*, 1999).

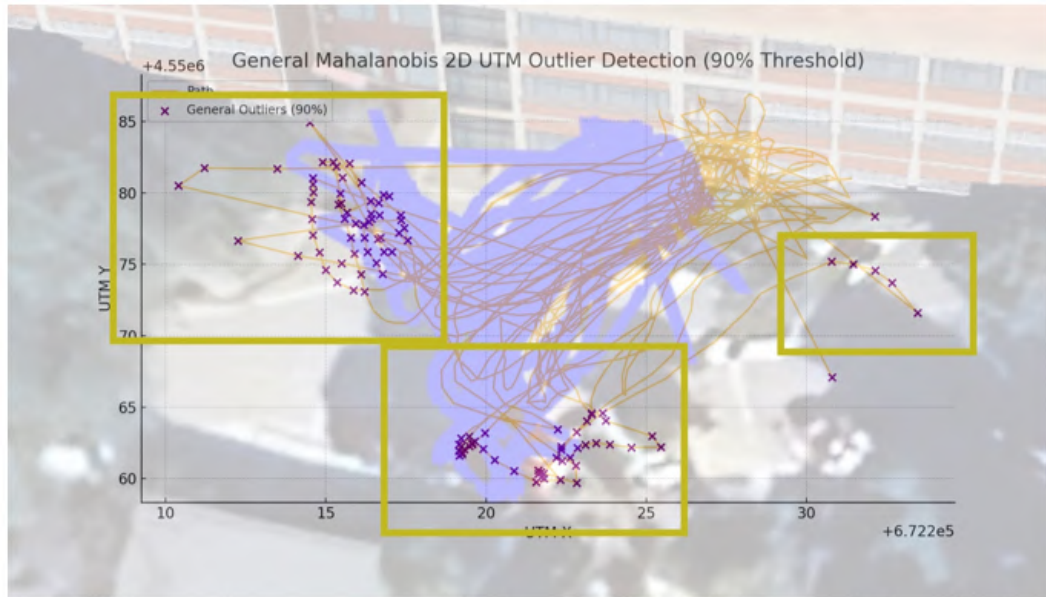


Figure 6.1. GPS trajectory data logged using an iOS-based sensor app, overlaid on a satellite basemap near the experiment site entrance.

As can be seen from the figure, when approaching areas with dense tree coverage or zones where buildings restrict measurement and signal transmission, deviations occur in the positioning data. These distortions are primarily caused by signal obstruction, multipath propagation, and attenuation, which reduce the reliability of location estimation. In particular, trees block or scatter satellite signals, while reinforced concrete and steel structures create reflections and signal loss. As also observed in the recorded videos at these points, the calculated positions occasionally “jump” or shift abruptly, reflecting the impact of such environmental interferences on system accuracy.

Similarly, as the movement path approached the new Data Science and Artificial Intelligence Institute building, GPS signal distortions increased, particularly near entrances and building edges, where 4-5 meter deviations were commonly observed. The building’s proximity to trees may further exacerbate multipath effects. Research has shown that urban canyons areas bordered by the structures can cause signal reflection and degradation, resulting in systematic biases and jumps in positional fixes (Groves, 2011; Zandbergen and Barbeau, 2011).

Conversely, the most accurate measurements were recorded in the open parking lot area located between the two buildings. It is observed that the mid area which covers nearly 25 metersquare at the mid of the parking area has the best accuracy comparing the GPS data within itself. This is consistent with optimal GPS operating conditions: open sky view, minimal reflective surfaces, and low vegetation, all of which help reduce multipath interference and increase the likelihood of direct satellite signal reception (Kaplan and Hegarty, 2005).

GPS works best under open-sky conditions with a clear view of the sky and many visible satellites. Consumer devices such as smartphones or trackers achieve their best accuracy when antennas have unobstructed view in all directions.

Under open sky, typical smartphone GPS accuracy is on the order of 5-10 m (U.S. Government, n.d.). However, accuracy degrades in areas with signal obstructions, such as near buildings, bridges, or trees (U.S. Government, n.d.). Reports that under open sky the typical horizontal error is nearly 4.9 m, but accuracy worsens near buildings, bridges, and trees (U.S. Government, n.d.). The most optimal solution for this is to keep the receiver away from tall structures or dense environment so that signals of satellites prevented to be blocking. Furthermore; walls, vehicles, trees or heavy machinery nearby can block or reflect GPS signals. Common causes of large errors include multipath and obstruction by buildings or tree canopy (U.S. Government, n.d.). For example, a smartphone survey found that points located in open areas had much lower error than points near tall buildings and trees (Merry and Bettinger, 2019).

In urban “canyons” which represents tall buildings on both sides of the road, reflected and blocked signals can push errors into the tens of meters or cause dropouts. Similarly, forest canopy attenuates GPS: NASA studies found 60% of signal lost under 10 m of foliage and 90% at 20 m depth.

Therefore, to achieve optimal GPS accuracy, it is advisable to operate the receiver in open areas, away from potential obstructions, ensuring a clear line of sight to multiple well-distributed satellites.

In environments where GPS accuracy is degraded such as urban canyons or dense forests several mitigation strategies have been identified to improve positional reliability. Denser canopies may lead to lower carrier to noise ratios reducing the positioning precision (Huang *et al.*, 2022). Due to the physical surroundings smartphones receive fewer Line of Sight signals, especially in urban and forested environments, reducing the accuracy of position estimation (Weng *et al.*, 2025). GPS signals often suffer from multipath effects, where signals reflect off structures before reaching the receiver.

This can lead to positioning errors ranging from several meters to tens of meters. In some cases, if the signal is reflected by a distant tall building, the errors can be even larger (Sun *et al.*, 2021). A study conducted by Moore *et al.* (2023) which is about predicting GPS fidelity in forestal areas founded that approximately 60% of the signal is lost under a 10-meter-thick deciduous canopy and about 90% under a 20-meter-thick canopy.

Modern smartphones such as iphone use built-in GNSS often aided by Wi-Fi that is highly comparable to standalone GPS units in open areas. A smartphone survey which is conducted by Merry and Bettinger (2019) demonstrated that points located in the open areas had much lower error points near tall buildings and trees. Smartphones log GPS data at very low frequencies such as 1 Hz or lower (). For example, research conducted by Zandbergen (2009) measured the accuracy of 3G enabled (Additionally, smartphones are not equipped with high grade GNSS antennas and often lack support for advanced positioning techniques like Real Time Kinematic (RTK) or Precise Point Positioning (PPP) which are common in professional surveying equipment (Huang *et al.*, 2022)).

Research on GPS wearables which has sensitive GNSS receiver indicates that higher sampling rates, such as 10Hz, offer more accurate and reliable data for dynamic activities which is similar to the construction worker activities. Smartphones has smaller and less sensitive GNSS antennas comparing to survey grade receivers, which leads to weaker signal acquisition, especially for dual frequency signals (Huang *et al.*, 2022). Dual frequency signals enable to provide better higher precision by lowering the delays which are related to ionospheric latencies. Smartphones also exhibit lower signal to noise ratios, leading to higher susceptibility to noise and errors (Huang *et al.*, 2022).

Signals from low elevation angles or unfavorable azimuths are more likely to be obstructed or reflected, reducing data quality and positional accuracy. Employing receivers that can access multiple GNSS constellations such as GNSS, GLONASS, Galileo and BeiDou and multiple frequency bands such as L1, L5 improve satellite visibility and mitigate errors such as ionospheric delays. Huang *et al.* (2022) assessed the performance of dual-frequency multi-GNSS smartphones for surveying applications. The study found that utilizing multiple frequencies and constellations improved positioning accuracy, especially in challenging environments. This approach enhances positioning accuracy, particularly in environments with signal obstructions.

Implementation of Real-Time Kinematic methods could be another way of solve these problems. Real-Time Kinematic utilizes carrier-phase measurements and corrections from a base station to achieve centimeter-level accuracy in real-time. Because differential GPS techniques employ corrections from nearby reference stations to improve accuracy to sub-meter levels. In urban environments, where GPS signals can be obstructed or reflected, algorithms like map matching and shadow matching provides enhancing the positioning accuracy. Map matching aligns GPS data with digital maps to correct positioning errors by snapping the estimated position to the nearest road or pathway. Shadow matching utilizes 3D models of urban environments to predict satellite visibility and improve position estimates by comparing expected and actual signal availability.

To overcome these drawbacks, GPS is often integrated with complementary technologies such as Ultra-Wideband (UWB), Inertial Navigation Systems (INS), and Wi-Fi in Real-Time Locating Systems (RTLS).

6.2. UWB Measurements

Despite its strengths, UWB has several practical constraints. The system's performance relies heavily on maintaining LoS conditions between tags and anchors. Under non-line-of-sight (NLoS) conditions caused by obstacles like walls, equipment, or even the human body reflections and penetrations introduce extra delays.

These result in range overestimations and positional biases (Che *et al.*, 2023). Materials with high attenuation properties, such as metal or water, can absorb or reflect UWB signals, leading to degraded accuracy or even signal loss.

Moreover, UWB's effective range is typically limited to several tens or hundreds of meters in open LoS environments, and it diminishes rapidly in cluttered indoor spaces (FiRa Consortium, 2024). Regulatory limits on transmit power further constrain its usable range compared to high-power radio systems.

UWB-based RTLS is increasingly used across sectors for precise indoor positioning. UWB tags worn by workers or mounted on heavy equipment enable continuous tracking on complex sites, improving safety by enabling collision avoidance and rapid emergency response (FiRa Consortium, 2024).

UWB systems are employed to track tools, vehicles, and workers, optimizing production line workflows and helping prevent workplace accidents (FiRa Consortium, 2024). Tagged pallets and forklifts are monitored in real time to support efficient routing and asset utilization (FiRa Consortium, 2024).

6.2.1. UWB Timestamped Data Comparison

UWB Data of Subject 1 in the two different environment are given below:

Table 6.1. Time intervals related to location subject 1.

INDOOR Engineering Faculty	OUTDOOR	INDOOR Data Science Institute	OUTDOOR
14:51:26-14:51:32	14:51:32-14:51:56	14:51:56-14:52:13	14:52:13-14:52:39
14:52:39-14:52:59	14:52:59-14:53:23	14:53:23-14:53:40	14:53:40-14:54:07
14:54:07-14:54:20	14:54:20-14:54:49	14:54:49-14:55:07	14:55:07-14:55:33

The testing was conducted in two main environments: the Engineering Faculty (Figure 6.2) and the Data Science Institute Building (Figure 6.3).

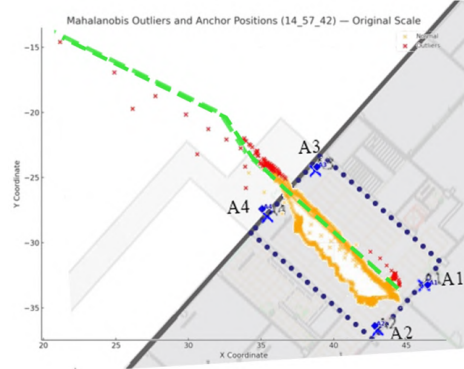


Figure 6.2. Subject 1 Faculty of engineering dataset.

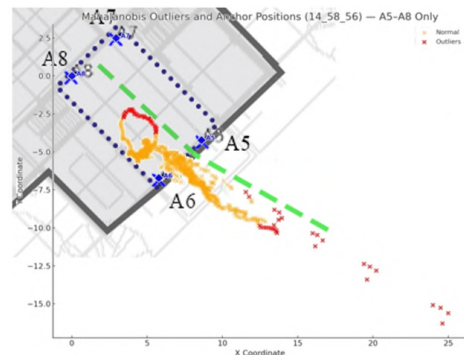


Figure 6.3. Subject 1 engineering faculty building all the dataset.

UWB Data of Subject 2 in the two different environment are given below.

Table 6.2. Time intervals related to location subject 2.

INDOOR Engineering Faculty	OUTDOOR (Engineering to Institute)	INDOOR Data Science Institute	OUTDOOR (Institute to Engineering)
15:11:46-15:11:56	15:11:56-15:12:29	15:12:29-15:13:14	15:13:14-15:13:48
15:13:48-15:14:22	15:14:22-15:14:52	15:14:52-15:15:22	15:15:22-15:16:00
15:16:00-15:16:27	15:16:27-15:17:05	15:17:05-15:17:25	15:17:25-15:18:01
15:18:01-15:18:31	15:18:32-15:19:13	15:19:13-15:19:35	15:19:35-15:20:13

The testing was conducted in two main environments: the Engineering Faculty (Figure 6.4) and the Data Science Institute Building (Figure 6.5).

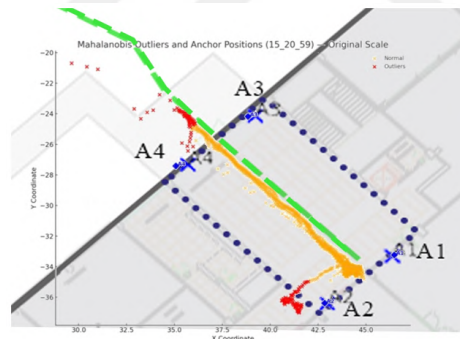


Figure 6.4. Subject 2 engineering faculty building all the dataset.

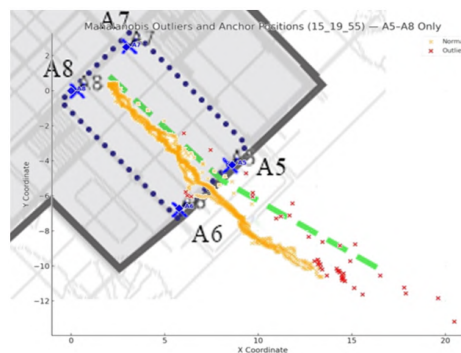


Figure 6.5. Subject 2 engineering faculty building all the dataset.

UWB Data of Subject 3 in the two different environment are given below

Table 6.3. Time intervals related to location subject 3.

INDOOR Engineering Faculty	OUTDOOR (Engineering to Institute)	INDOOR Data Science Institute
15:22:00-15:22:12	15:22:12-15:22:45	15:22:45-15:23:04
15:23:04-15:23:35	15:23:58-15:24:38	15:24:38-15:24:50
15:24:50-15:25:20	15:25:20-15:25:41	15:25:41-15:25:16
15:25:16-15:26:47	15:26:47-15:27:17	15:27:17-15:27:39
15:27:39-15:28:10	15:28:10-15:28:35	15:28:35-15:29:09

The testing was conducted in two main environments: the Engineering Faculty (Figure 6.6) and the Data Science Institute Building (Figure 6.7).

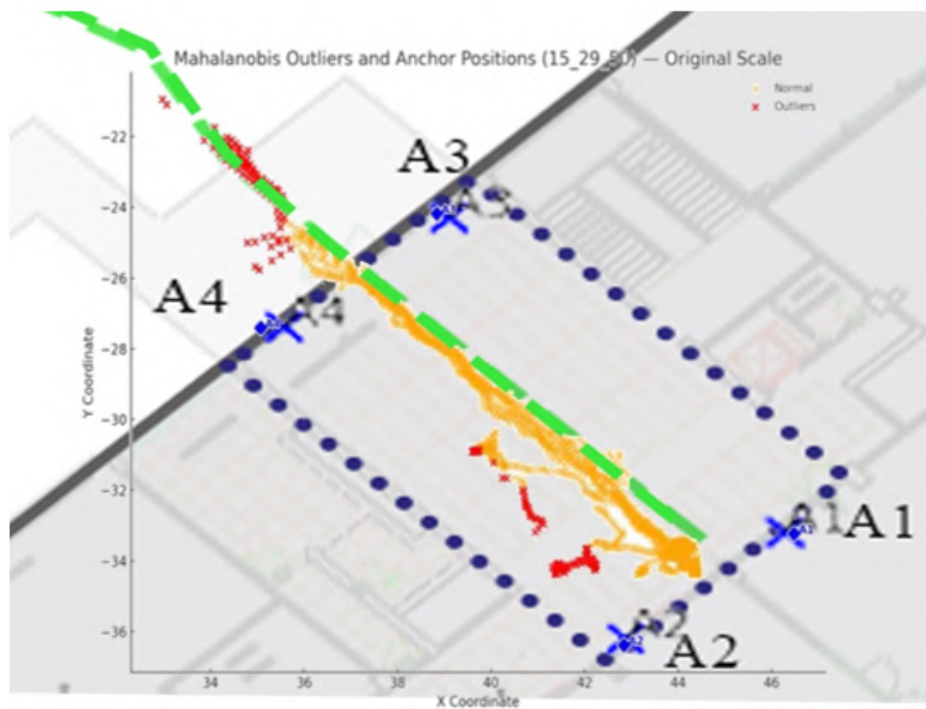


Figure 6.6. Subject 3 engineering faculty building all the dataset.

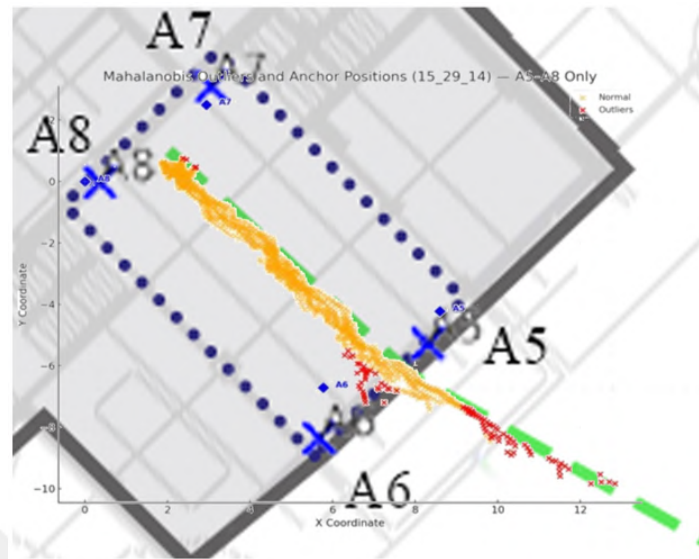


Figure 6.7. Subject 3 artificial intelligence institute building all the dataset.

UWB Data of Subject 4 in the two different environment are given below.

Table 6.4. Time intervals related to location subject 4.

INDOOR Engineering Faculty	OUTDOOR	INDOOR Data Science Institute	OUTDOOR
15:31:38-15:31:45	15:31:45-15:32:09	15:32:09-15:32:21	15:32:21-15:32:49
15:32:49-15:33:06	15:33:06-15:33:33	15:33:33-15:33:48	15:33:48-15:34:07
15:34:07-15:34:26	15:34:26-15:34:50	15:34:50-15:35:04	15:35:04-15:35:29
15:35:29-15:35:47	15:35:47-15:36:09	15:35:09-15:36:22	15:36:22-15:36:46

The testing was conducted in two main environments: the Engineering Faculty (Figure 6.8) and the Data Science Institute Building (Figure 6.9).

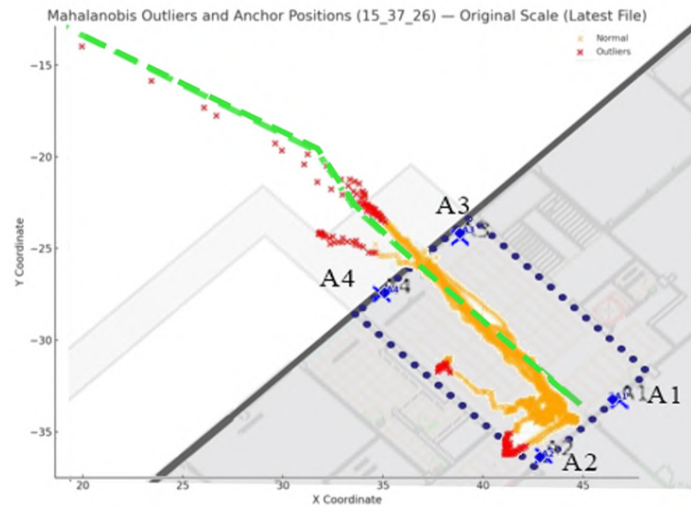


Figure 6.8. Subject 4 engineering faculty building all the dataset.

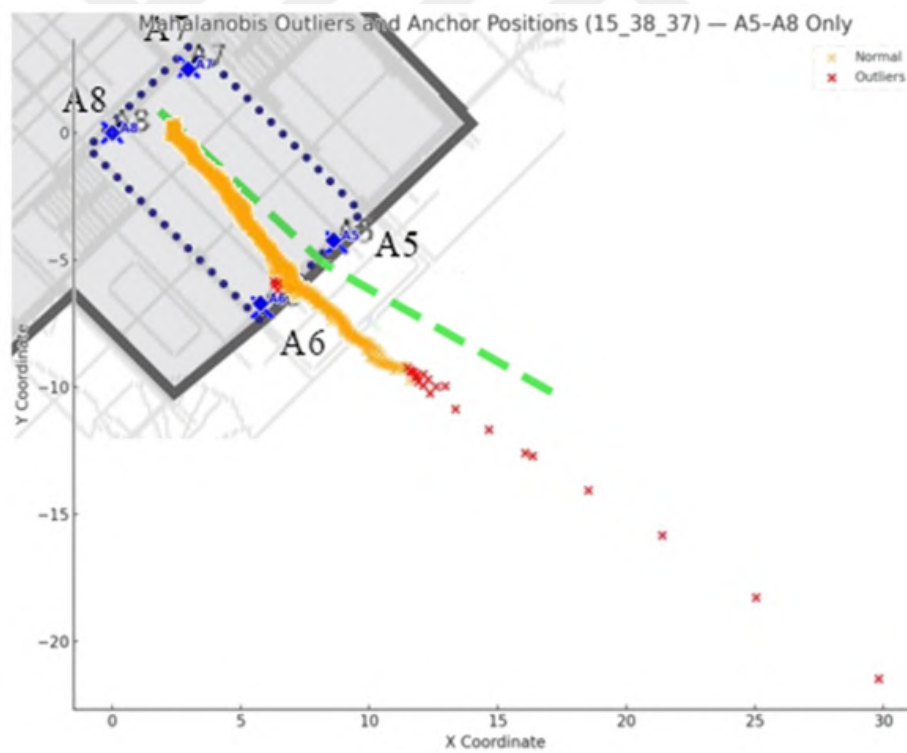


Figure 6.9. Subject 4 artificial intelligence institute building all the dataset.

The highest rate of data irregularities was observed for Subjects 1 and 2 as it can be seen in Figures between 6.2-6.5. The analysis indicates that the primary cause is increased non-line-of-sight (NLOS) exposure driven by anthropometric differences

specifically, height and body mass which altered the effective tag–anchor geometry and increased “self-occlusion” of the radio path by the body. Human tissue is a strong attenuator and scatterer in UWB bands; when the direct path from an anchor to a body-worn tag passes through the torso or arm, the first path is weakened or blocked, and the receiver locks onto later multipath components. This produces positively biased ranges, jitter, and bursty packet loss, which manifest as spikes and gaps in the time series and as clusters flagged by the Mahalanobis outlier test before fusion. Taller stature also shifts the relative elevation of the tag: with low-mounted anchors, the resulting shallow incidence increases floor and wall multipath; with high-mounted anchors, a waist/chest tag sits further below the anchor beam, making occlusion by the torso more frequent during rotations. Greater body mass increases the cross-section that can shadow anchors behind the subject, so routine motions create longer and more frequent NLOS intervals. These effects explain why Subjects 1 and 2 exhibited (i) higher residuals and variance in UWB ranges, (ii) more points eliminated by Mahalanobis filtering, and (iii) occasional transient divergence in the UKF when NLOS bursts coincided with poor anchor geometry.

In short, Subjects 1 and 2 experienced more severe NLOS due to their height/weight differences, which increased body-induced shadowing and multipath, leading to the observed data irregularities despite identical test paths and anchor layouts.

Practical implications and mitigations include prioritizing tag placements that minimize self-occlusion favoring helmet. Anchor geometry should be diversified in both azimuth and elevation to reduce geometric dilution and eliminate blind sectors around the body. During operation, NLOS should be detected dynamically (e.g., via residuals, innovation covariance, or available UWB quality indicators), with affected measurements down-weighted or gated before fusion. Finally, per-subject calibration (including bias terms) and incorporation of IMU-derived heading can help the filter anticipate torso-induced shadowing and maintain estimation stability.

6.3. GPS – UWB Data Fusion Issues

To achieve accurate and seamless tracking across both indoor and outdoor environments, an Unscented Kalman Filter (UKF) was applied to fuse UWB and GPS position measurements. The UKF implementation utilized a 6-dimensional state vector, allowing the filter to account for dynamic motion characteristics such as velocity and acceleration.

The filter was updated at each timestamp based on the available measurements. When indoors, UWB data was the primary source, while GPS was weighted down or ignored due to signal degradation. Conversely, when outdoors, GPS dominated and UWB was often unavailable. In transitional zones, both sources contributed to the fusion depending on their respective uncertainty levels. Measurement noise covariances were dynamically assigned: UWB observations were given lower variance ($\sigma \approx 0.3$ m), while GPS readings carried higher noise ($\sigma \approx 2\text{-}5$ m), especially near building facades.

The resulting UKF outputs showed strong improvements in trajectory stability and realism. Short-term outliers in UWB (e.g., caused by NLOS or multipath) were effectively suppressed by Mahalanobis-based gating. Similarly, GPS noise was smoothed during transitions, preventing abrupt jumps at building entries/exits. These results confirmed the effectiveness of the filter in delivering reliable position estimates suitable for real-time safety monitoring.

6.3.1. Spatial Accuracy and Transition Behavior in GPS-UWB Fusion

The results of the proposed sensor fusion system reveal important insights regarding the relationship between indoor spatial geometry, positioning reliability, and transition dynamics between GPS and UWB systems. These findings emphasize the practical implications of system behavior under real-world construction site constraints.

Di Pietra *et al.* (2020) presented a low-cost GNSS/UWB/INS integration using an Extended Kalman Filter (EKF) with geofenced switching. Their five experimental campaigns achieved approximately 0.35 m horizontal and 0.45 m 3D accuracy, proving that the system could maintain continuity while transitioning between outdoor and indoor spaces, albeit without reaching sub-decimeter precision. Zhu *et al.* (2023) advanced this approach by developing a tight UWB/IMU fusion scheme with an enhanced EKF featuring double-loop error control for time synchronisation and NLOS mitigation. Their method reduced the maximum deviation to 0.232 m and the standard deviation to 0.0998 m, highlighting the ability of inertial sensors to stabilise UWB measurements under challenging conditions.

First and foremost, the accuracy of UWB-based indoor positioning was found to be significantly influenced by the physical size and structure of the test environment. In the larger indoor test space (approximately 9 X 12 meters), the Unscented Kalman Filter (UKF)-fused positions closely followed the reference walking path, with an average deviation of 45 cm and a maximum of 86 cm as observed within the BIM environment. This level of accuracy is considered acceptable for dynamic safety visualization in digital twins.

However, in a smaller corridor-like indoor space (approximately 6 X 8 meters), spatial accuracy deteriorated. The fused location often deviated more than 1 meter from the expected path, and in certain cases, the subject's entry into the indoor area could not be reliably inferred based on fused data alone. This discrepancy is likely due to the effects of multipath interference, anchor density limitations, and insufficient signal diversity in confined environments. These findings suggest that UWB anchor placement strategies must be adapted to room geometry to maintain accuracy, particularly in tight interior spaces.

Transition zones between outdoor (GPS-dominated) and indoor (UWB-dominated) environments also presented substantial challenges. The system required subjects to move slowly and deliberately during the transition from GPS to UWB zones to allow

the filter to adapt to the changing signal conditions. If the subject entered the building rapidly, the fused output lagged behind or misrepresented the position, resulting in abrupt spatial jumps or errors in zone classification. This phenomenon was less pronounced when transitioning from UWB back to GPS, possibly due to the nature of GPS signals benefiting from open-sky conditions and inherent smoothing over larger distances.

One of the most critical observations was the presence of GPS outliers during entry/exit points, with errors extending up to 10 meters due to poor satellite visibility and urban canyon effects. To address this, Mahalanobis distance-based outlier rejection was applied prior to fusion, allowing the system to discard statistically implausible readings. This process substantially improved the robustness of the fused output, demonstrating that pre-fusion outlier detection is essential when dealing with heterogeneous sensor data.

Despite the improvements, the system's reliance on manual speed control during transitions and sensitivity to spatial context point to key areas for future refinement. Adaptive filtering models or context-aware switching logic may enhance resilience during abrupt zone changes. Furthermore, dynamic adjustment of anchor geometry or the inclusion of inertial data could support smoother fusion in constrained indoor environments.

Overall, while the fusion framework proved capable of providing accurate, continuous worker location data within the BIM environment, transition management and spatial variability remain the most prominent limitations. Addressing these challenges is essential for scaling the system to real-time site-wide deployment across diverse construction settings.

Li *et al.* (2024) introduced a semi-tightly coupled architecture for GNSS/UWB/INS integration, applying robust weighting and DOP-aware constraints to improve accuracy in urban canyon environments. This approach provided smoother trajectories

and more reliable positioning across transition areas, reinforcing the importance of adaptive quality control in hybrid RTLS. In parallel, Liu *et al.* (2024) proposed a Dynamic-UKF to fuse UWB with wheel odometry, achieving sub-decimeter accuracy (RMSE 0.075 m) in complex indoor motion, demonstrating the potential of odometry as a strong complementary cue under line-of-sight (LOS) conditions. Finally, Zhang *et al.* (2025) employed factor-graph optimisation (FGO) for UWB/IMU/Ultrasonic fusion. Their sliding-window framework, enhanced by dynamic covariance adaptation, achieved 12.3 cm RMS error and a 38% performance gain compared to EKF baselines, and crucially remained robust even when UWB packet reception fell to 40%.

A recurring theme across these studies is the role of outlier detection and NLOS mitigation. Many pipelines implement 2 or Mahalanobis distance tests to gate UWB updates or apply robust covariance adaptation to down-weight unreliable measurements. This trend aligns directly with the current thesis, where a Mahalanobis-based filter was used prior to Unscented Kalman Filtering (UKF) to ensure that only statistically reliable UWB observations were fused with GPS. When viewed collectively, the comparative performance envelope of contemporary RTLS research can be summarized as follows: Indoor fusion of UWB with inertial or odometry sensors typically yields errors between 0.07–0.23 m under favourable geometry (Zhu, 2023; Liu, 2024). Factor-graph optimization improves resilience under harsher signal conditions, maintaining 0.12 m RMS with substantial error reductions compared to KF (Zhang, 2025). Seamless GNSS/UWB/INS integrations usually achieve 0.3–0.5 m RMS in real world transitions, balancing continuity and robustness (Di Pietra, 2020; Li, 2024).

Against this backdrop, the present thesis contributes in two distinct ways. First, by adopting a UKF combined with Mahalanobis distance gating, it extends the principles of robust fusion into a streamlined GPS+UWB pipeline, suitable for environments where only smartphone-grade GPS is available outdoors and UWB anchors indoors. Although absolute outdoor accuracy remains constrained by the limitations of consumer GPS, the method ensures enhanced continuity across environments, while maintaining reliable indoor positioning through UWB.

Another bottleneck was the manual alignment required to map UWB and GPS coordinates to the internal Revit coordinate system. The lack of an automatic calibration workflow necessitated the use of reference anchors and scripting to perform the transformation resulting in a time-consuming, error-prone process. Small misalignments or orientation mismatches required additional correction during visualization.

6.4. Validation

While the proposed hybrid GPS-UWB tracking framework demonstrated improved spatial continuity and successful track of workers within the 3D Revit model, several limitations emerged during detailed experimental observations. One notable issue was related to signal propagation variability caused by non-line-of-sight (NLOS) conditions, particularly in indoor scenarios. Within Enscape, we validated accuracy by synchronizing timestamped site videos with the rendered UWB-GPS trajectory and computing frame-matched positional errors. Each run was video-recorded with timestamps and frame-by-frame synchronized to the sensor logs. Indoors, the fused traces were overlaid on Revit floor plans and visually checked against Revit features (walls, doors, grids); outdoors, surveyed GPS control provided the baseline. This hybrid approach enabled a direct comparison between “ground-truth” video frames and the Enscape playback of the Revit-embedded trajectories.

Specifically, when comparing individual test subjects, shorter participants such as Subject 1 and Subject 2 exhibited more frequent UWB delays and signal dropouts. These anomalies are attributed to body occlusion effects, where the human body, especially in shorter individuals, obstructed direct line-of-sight between the UWB tag and nearby anchors more often. Due to their lower antenna height and walking posture, the tag position was more likely to be blocked, especially in areas with reflective surfaces or anchor positions placed at higher elevations. This led to observable latency in position updates and reduced signal reliability in certain spatial segments of the test corridor. Comparing the time-aligned videos and Enscape overlays, the most frequent irregularities occurred for Subjects 1 and 2, driven primarily by greater NLOS exposure

linked to anthropometrics (height/mass) and the resulting self-occlusion of tag–anchor paths; these cases produced positively biased ranges jitter, and bursty packet loss that the Mahalanobis gate flagged before fusion. In contrast, Subject 4’s overlay appeared the most stable indoors, consistent with a taller stature and steadier gait (qualitatively visible in the overlays), and with the larger indoor room the UKF trace followed the walking path closely (≈ 45 cm mean deviation).

The timestamped-video audit validates that the UKF+Mahalanobis GPS/UWB pipeline, embedded in Revit and rendered via Enscape, is accurate enough for safety visualization in favorable rooms and steady motion, but its reliability is subject- and context-dependent. The pattern observed concluded as Subject 4 has least deviation; Subjects 1–2 most matches well-documented HBS/NLOS effects and current mitigation strategies. For site deployment, it is recommended helmet-mounted tags, denser/more diverse anchors in occlusion-prone zones, inertial co-fusion, and either adaptive filtering across transitions or moderated policies measures that align both with the findings and with recent UWB research. The timestamped video recording screenshots are given in Figure 6.10, Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14.



Figure 6.10. Subject 1 comparison with enscape.

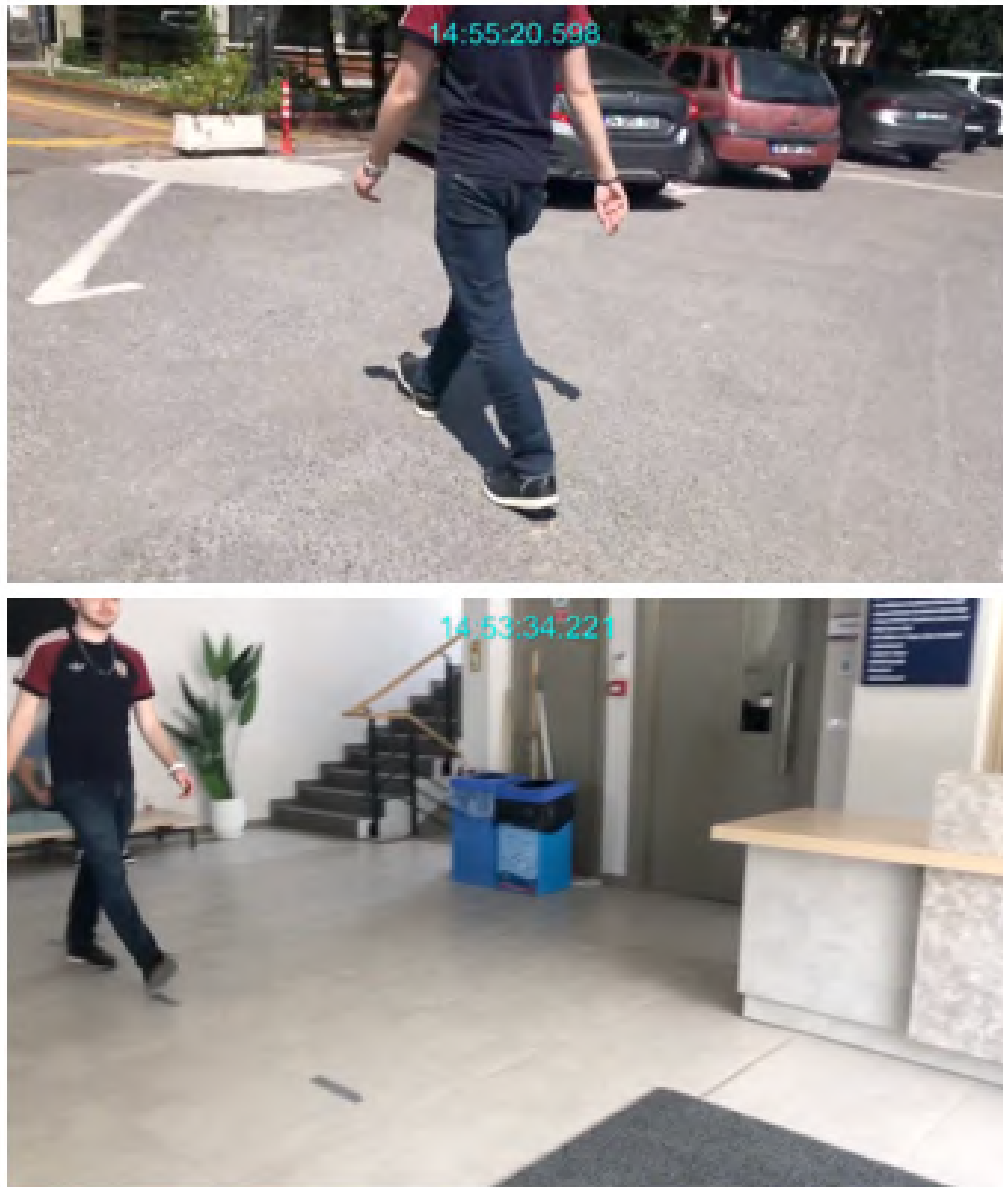


Figure 6.11. Subject 1 comparison with enscape 1.



Figure 6.12. Subject 2 comparison with escape.

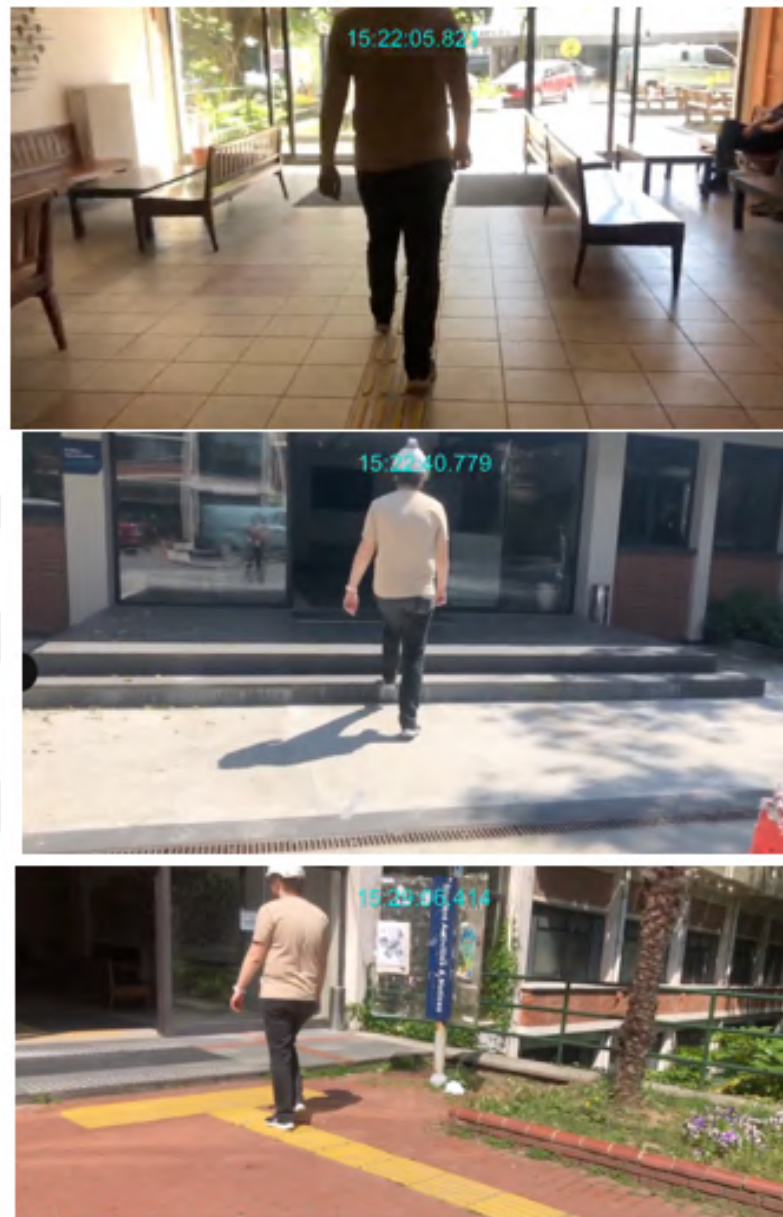


Figure 6.13. Subject 3 comparison with enscape.



Figure 6.14. Subject 4 timestamped video record.

6.5. System Performance and Visualization Efficacy

This section evaluates how well the Revit–pyRevit–Enscape pipeline sustains interactive, real-time visualization when streaming UWB–GPS trajectories into a Revit scene. It can be adapted to real-time pragmatically as a viewer that remains navigable (smooth pan/zoom/ orbit) while ingesting continuous position updates and drawing moving markers without stutter or instability. To probe limits on a laptop-class setup, a small benchmark across three operating modes: (i) full-rate ingestion of the complete trajectory, (ii) temporal down- sampling (importing one of every four timestamps), and (iii) rolling-window rendering that caps the number of simultaneously displayed points were conducted. It is reported that the practical interactivity threshold observed on the test hardware, identify the dominant bottlenecks (viewport/GPU load and frequent geometry updates), and quantify the trade-off between temporal fidelity and responsiveness. Then the stabilization techniques that kept Enscape responsive (down-sampling, adaptive throttling, batched updates) and position these findings against prior RTLS-Revit visualization practices were summarized. Finally, data-collection constraints (e.g., concurrent mobile logging contending for GNSS) that can degrade perceived “real-time,” were noted and outline when to consider alternative pipelines (Unity/Unreal for high-FPS animation; Forge/iTwin for scalable, multi-user viewing).

6.5.1. Data Collection Issues

The outdoor data acquisition relied on iOS-based GNSS applications (e.g., GeoTracker, MyTracks). However, the accuracy of smartphone GPS was inconsistent, with errors occasionally reaching 10 meters, especially near buildings or trees where signal multipath or occlusion occurred. The average positional error (mean step vector) was approximately 6 meters, which significantly compromised spatial continuity and fusion reliability. These deviations created false zone breach detections and unnatural movement artifacts when combined with UWB data. For improved accuracy, dedicated dual-frequency GNSS receivers (e.g., u-blox ZED-F9P) or differential correction techniques such as Real-Time Kinematic (RTK) GNSS are recommended for future

studies. Alternatively, fusion with inertial measurement units (IMUs) on the device could be used to smooth and stabilize location transitions.

During the operations of both applications, technical limitation was observed: timestamp compression and system-level resource contention appeared to interfere with the stable functioning of dual-app GPS logging. Specifically, the iOS operating system restricts simultaneous access to GNSS sensor streams in order to preserve battery life and maintain system efficiency. As a result, when both Geo Tracker and My Tracks were launched together, one application consistently received priority in accessing real-time GNSS data, while the other experienced reduced logging frequency, data loss, or complete suppression of background operation.

Empirical observations during the experiment revealed that only the most recently launched application remained actively connected to the GPS hardware, effectively overriding or suspending the previously opened app. In practice, this meant that although both apps appeared to be recording in the foreground, only one was truly capturing complete, real-time data while the other produced either intermittent logs or incomplete trajectory files. This behavior underscores a fundamental constraint of iOS-based smartphones: they do not support parallel GNSS data access for multiple applications in active or background mode due to sensor exclusivity policies and aggressive task management protocols.

This limitation has critical implications for experimental setups requiring multi-app synchronization or redundancy, and must be carefully considered when designing GPS data acquisition workflows on iOS devices. Where dual logging is necessary, external GPS receivers or staggered recording sessions may offer more reliable alternatives.

6.5.2. Revit API Integration

While this study successfully demonstrated the integration of GPS and UWB data into Revit which represents a BIM environment-based framework for safety monitoring,

several key limitations were encountered, particularly during the data visualization and system implementation phases.

A significant limitation stemmed from the insufficient graphical processing capability (GPU) of the Revit workstation used during the project. The hardware struggled to handle the dense point cloud of worker location data, particularly during BIM-to-Enscape transfer. As a result, only a small portion of the full dataset could be visualized and tested within Revit and Enscape environments. When attempting to visualize the complete trajectory data (e.g., 1200+ fused points), the system experienced considerable lag, and the export process took up to three hours, severely restricting iteration and performance evaluation.

Due to the rendering and loading constraints, the amount of data visualized in the Revit model was artificially limited, which may have constrained the generalizability of the results. Although the processing scripts were applied to the full dataset, only selected frames or representative data subsets were used in the Revit model. This limitation hindered a complete spatial-temporal analysis of worker behavior or prolonged zone occupation.

Although the system was conceptually built for real-time tracking, the study operated in a post-processed mode due to limitations in stable data transmission, Revit refresh latency, and wireless synchronization. Implementing real-time streaming, especially in large projects, will require additional system robustness, possibly through edge computing or buffering mechanisms.

Although the system was conceptually built for real-time tracking, the study operated in a post-processed mode due to limitations in stable data transmission, Revit refresh latency, and wireless synchronization. Implementing real-time streaming, especially in large projects, will require additional system robustness, possibly through edge computing or buffering mechanisms.

Using the Autodesk Revit API with pyRevit, this study successfully integrated timestamped location data into a BIM environment, demonstrating a working digital twin for tracking motion in 3D. The approach proved conceptually feasible: fused indoor (UWB) and outdoor (GPS) coordinates were mapped in real time onto the Revit model, and visualized via the Enscape rendering engine as moving markers or “motion trails” within the building model. However, significant performance limitations emerged when scaling up the data volume. The system was tested on a standard ASUS laptop with an Intel Core i5 processor (with a modest GPU), and it struggled to handle more than a few hundred location points in the Revit-Enscape environment. In particular, the graphical processing unit (GPU) became a bottleneck once the trajectory dataset grew dense. Exceeding on the order of approximately 300 simultaneous points led to severe lag, slow rendering and computer crash; attempts to visualize the full trajectory (which has approximately 1200 fused timestamped coordinates) caused the export/visualization process to balloon to multiple hours (approximately 2-3 hours for 1200 points). This unacceptable latency effectively precluded real-time use on the given hardware and made iterative testing difficult. The findings highlight that while Revit’s API can facilitate integration of live data, a typical laptop GPU may be insufficient for real-time visualization of large, high-frequency location datasets, resulting in major slowdowns and instability. In short, the feasibility is proven on a functional level, but there are clear scalability and performance limits with the chosen BIM platform Autodesk Revit and hardware.

To maintain stability, the system implemented temporal downsampling: only one out of every four timestamped coordinates was imported, resulting in an effective update rate of 4 seconds per position. This reduced temporal fidelity but enabled smooth visualization in Enscape without system crashes. This reflects a common trade-off between data fidelity and system responsiveness, which is widely reported in digital twin implementations on standard hardware (Mautz, 2012; Pan *et al.*, 2021). This performance aligns with prior RTLS-BIM integration studies. Setayeshgar *et al.* (2013) demonstrated a UWB-BIM safety monitoring prototype that struggled with high-frequency updates and recommended partial trajectory visualization to sustain interac-

tivity. Pan *et al.* (2021) implemented a cloud-hosted 4D BIM dashboard to avoid local hardware bottlenecks, demonstrating that offloading visualization to remote servers can sustain near- real-time updates even with high-frequency data. Chac'on *et al.* (2024) benchmarked digital twin dashboards and showed that laptops with integrated GPUs could not handle continuous 1 Hz data streaming without data thinning, similar to this study's 0.25 Hz sampling approach.

To validate the system performance that is applied in this thesis, a benchmarking methodology was applied with the mentioned studies. Full dataset test (100% points) of each subject integration to the Revit took 2 hours to process 1200 points; Revit viewport lags and Enscape is unresponsive when there is more than 1200 point is the dataset that is transformed. In order to prevent this limitation the total dataset is downsampled in order to enhance the system visualization robustly (25% points, 4 s interval). Visualization stabilization is achieved, and interactive monitoring is provided with Enscape. Observed threshold is beyond 300 simultaneous points from the total dataset, the system becomes impractical on the test hardware.

In preliminary scalability trials, it was assessed how the visualization stack (Revit + Enscape + pyRevit streaming) behaves under different point loads. Using the full dataset (100% of recorded positions; $N \approx 1,200$ points rendered), end-to-end processing completed in roughly two hours, during which the Revit viewport exhibited pronounced interaction lag and Enscape became intermittently unresponsive. After temporal downsampling to 25% of the points (equivalent to a 4 s sampling interval), the pipeline remained stable and Enscape interaction returned to normal.

Empirically, on the test hardware, rendering more than 300 simultaneous points proved impractical for interactive use, prompting us to cap the number of concurrently displayed points in subsequent experiments (via adaptive throttling/rolling windows). This restraint does not affect the underlying analytics-full-rate data are logged and analyzed offline but it ensures that the real-time viewer remains responsive for monitoring and demonstration purposes. This benchmark validates that standard laptops

are insufficient for dense real-time RTLS data visualization in Revit, and that data reduction or enhanced computation is necessary for stability.

Visualizing live RTLS data within BIM environment especially virtual environment requires combining high-frequency data streams with rendering engines capable of handling large and complex models. Three distinct strategies dominate the literature: the use of lighter 3D/game engines such as Unity and Unreal, optimized scripting within Revit through pyRevit, and cloud-based rendering platforms such as Autodesk Forge and Bentley iTwin. Each has been studied in terms of computational efficiency, stability, scalability, and implementation feasibility.

Game engines such as Unity and Unreal provide GPU-optimized rendering pipelines that can achieve high frame rates and real-time responsiveness, even with complex BIM models. Guo *et al.* (2017) show that GPU-based rendering allows smooth handling of moving objects in BIM-derived scenes, while Johansson and Roupé (2024) demonstrate multi-user VR applications based on large IFC models imported into Unreal Engine, noting both stability and interactivity. Uz Zaman *et al.* (2024) similarly integrated Revit models into Unity for safety training scenarios, illustrating how dynamic worker movements and hazard events can be simulated with minimal performance loss.

The strength of this approach lies in responsiveness: each update from RTLS can be mapped directly into the game-engine frame loop, supporting animations at 30-90 Hz depending on GPU power. However, Rehman *et al.* (2025) highlight that scalability and integration remain bottlenecks, as models often require preprocessing (e.g., mesh simplification or LOD management) to avoid GPU overload. While game engines support high-fidelity visualization, they demand expertise in model export, synchronization, and network coding to integrate RTLS streams effectively.

By contrast, optimized pyRevit scripting keeps visualization entirely within the Revit environment. This approach leverages the Revit API to insert or update markers representing tracked objects. Li *et al.* (2016) note that reducing GPU load via

single markers or batched updates can mitigate performance bottlenecks. Nonetheless, Revit’s rendering engine is not designed for high-frequency animation. The literature indicates that while pyRevit is effective for batch automation and one-off geometry generation, it is ill-suited for dense, continuous RTLS feeds, which can overwhelm the single-threaded Revit environment and result in significant interface lag (Li *et al.*, 2016).

Scalability is another concern that limits this framework. Revit becomes increasingly unstable when handling large models combined with frequent updates. Unlike game engines or cloud viewers, there are no published examples of pyRevit being deployed for real-time multi-user RTLS visualization, suggesting its use cases are limited to small-scale or low-frequency monitoring.

Cloud-based platforms such as Autodesk Forge and Bentley iTwin adopt a different paradigm, streaming BIM models to WebGL-based viewers and overlaying dynamic data from IoT or RTLS sensors. Pan *et al.* (2021) emphasized that scalable GPU clusters in cloud infrastructure enable real-time rendering of complex models while maintaining multi-user accessibility. Autodesk’s Project Dasher exemplifies this approach, showing real-time IoT sensor data visualized within BIM through Forge-based dashboards (Dasher360, 2021). Similarly, Desapex (2023) reported IoT integration with Autodesk Tandem, where environmental sensors updated every five seconds to provide continuous building performance monitoring.

While these platforms excel in scalability and distributed collaboration, their responsiveness to very high-frequency RTLS data (>10 Hz) is limited by network latency and browser rendering constraints. Optimizations such as mesh consolidation and instancing improve frame rates (Autodesk, 2021), but the inherent dependence on internet connectivity introduces latency risks. Still, for facility management and operations where update cycles are slower, cloud platforms are increasingly seen as practical solutions.

Taken together, the literature suggests that game engines are the most capable of handling dense real-time RTLS visualization, offering high frame rates and stability when properly optimized. However, they require extensive preprocessing and integration work. pyRevit offers direct integration with Revit and is simple to implement but is computationally constrained and unsuitable for large-scale, high-frequency scenarios. Cloud-based platforms strike a balance, supporting large-scale, multi-user access and integration with IoT systems, though at the cost of reduced responsiveness and dependence on stable networks. Each approach is therefore best suited to different contexts: game engines for immersive, real-time safety and training simulations; pyRevit for lightweight, Revit-native monitoring; and cloud platforms for scalable, collaborative digital twin applications.

In conclusion, the benchmarking confirms that achieving stable digital twin visualization in Revit on standard hardware requires sacrificing temporal fidelity or migrating to optimized/cloud solutions, aligning with trends reported in RTLS integration into Revit which is BIM environment literature.

The evaluation of the developed GPS-UWB into Revit integration system focused on two main aspects: spatial-temporal accuracy and system performance. Accuracy was primarily assessed through qualitative methods, involving visual comparison of the tracked trajectories within Revit and their real-time Enscape renderings. Time-matched snapshots and geofence-crossing events were used to confirm that the virtual trajectory correctly reflected the real-world movement sequence. Although no numerical error metrics were computed, this approach is widely adopted in RTLS-BIM and digital twin studies where visual and event-based verification is considered sufficient for safety monitoring applications (Setayeshgar *et al.*, 2013; Guo *et al.*, 2017; Pan *et al.*, 2021).

System latency and performance were benchmarked by comparing processing times for datasets of different sizes. While trajectories of approximately 300 points were visualized smoothly, larger datasets of around 1200 points caused significant GPU

bottlenecks, leading to processing times of 2-3 hours and non-interactive performance in Enscape. To mitigate this limitation, the trajectory data was downsampled to one in every four points, reducing the temporal resolution to approximately 4 seconds per update. This maintained visualization stability while slightly reducing motion fidelity, aligning with common practices in real-time location BIM visualization. Which means it is hard to reach the real-time track.

Overall, the findings demonstrate that the proposed system is feasible for near real-time monitoring under moderate data loads, but its scalability is constrained by hardware and rendering limitations. Ensuring higher-frequency or large-scale tracking would require data reduction, optimized pyRevit scripting, or migration to lightweight or cloud-based visualization platforms.

In conclusion, while state-of-the-art indoor systems demonstrate sub-decimeter accuracy through complex multi-sensor integration and graph-based estimation, this thesis strategically prioritises continuity, robustness, and BIM-based applications. By situating technical sensor fusion within the context of digital twin and site monitoring workflows, the research fills a practical gap between theoretical advances in RTLS and their application to construction applications.

Table 6.5. Comparative summary of studies (2020–2025).

Study	Sensors and Method	Datasets / Scenarios	Key Accuracy / Findings
Di Pietra <i>et al.</i> , 2020 (Sensors)	GNSS + UWB + INS; EKF	5 campaigns, pedestrian, indoor–outdoor transitions	0.35m horizontal, 0.45m 3D; demonstrated continuity rather than centimetric accuracy

Table 6.5 Comparative summary of studies (2020–2025). (cont.)

Study	Sensors and Method	Datasets / Scenarios	Key Accuracy / Findings
Zhu <i>et al.</i> , 2023 (Applied Sciences)	UWB + IMU; improved EKF with double-loop error control (time sync, NLOS mitigation)	Indoor trials under NLOS	Max deviation 0.232 m; SD 0.0998 m; robust indoor accuracy with IMU aid
Li <i>et al.</i> , 2024 (Remote Sensing)	GNSS + UWB + INS; semi-tightly coupled robust estimation	Mixed urban canyon and transition areas	Quality-controlled fusion; smoother trajectories; reliable continuity under poor geometry
Liu <i>et al.</i> , 2024 (Electronics)	UWB + wheel odometry; Dynamic-UKF (DUKF)	Indoor mobile platform, complex motion	RMSE 0.075 m; mean 0.085 m; sub-decimeter indoor baseline under LOS
Zhang <i>et al.</i> , 2025 (arXiv)	UWB + IMU + Ultrasonic; Factor-Graph Optimisation (sliding window, dynamic covariance)	Complex indoor, intermittent UWB packets	12.3 cm RMS; 38 better than KF; robust at 40 packet reception
This Thesis -2025	GPS + UWB; UKF + Mahalanobis filter; BIM integration (Revit API)	2 indoor (5k pts) + 1 outdoor (2k pts)	Enhanced continuity; robust outlier removal; Revit-integrated Safety 4.0 visualisation

Future studies should prioritize (1) hardware upgrades for GPU-intensive tasks, (2) scalable and modular visualization methods, (3) high-precision outdoor GNSS sensors or RTK-enhanced smartphones, and (4) automation tools for coordinate transformation and alert management.

6.6. Survey for Applicability of the Framework

The pilot gathered $n = 10$ responses from Turkish AEC professionals, spanning architects, interior architects/designers, and civil engineers; most work on residential projects (80%). Only 20% reported active Revit use in their organization, and 40% said indoor and outdoor operations are run together, indicating early-stage BIM/RTLS maturity typical of many small/medium contractors (Türkiye context).

A majority believe the framework would provide meaningful safety benefits (aggregate agree/strongly-agree: 60%). They likewise endorse that continuous indoor–outdoor tracking adds operational value (60%) and that real-time visualization speeds decisions (60%). By contrast, none agreed the system can be integrated into current safety processes; responses were 100% negative (40% “disagree”, 60% “strongly disagree”).

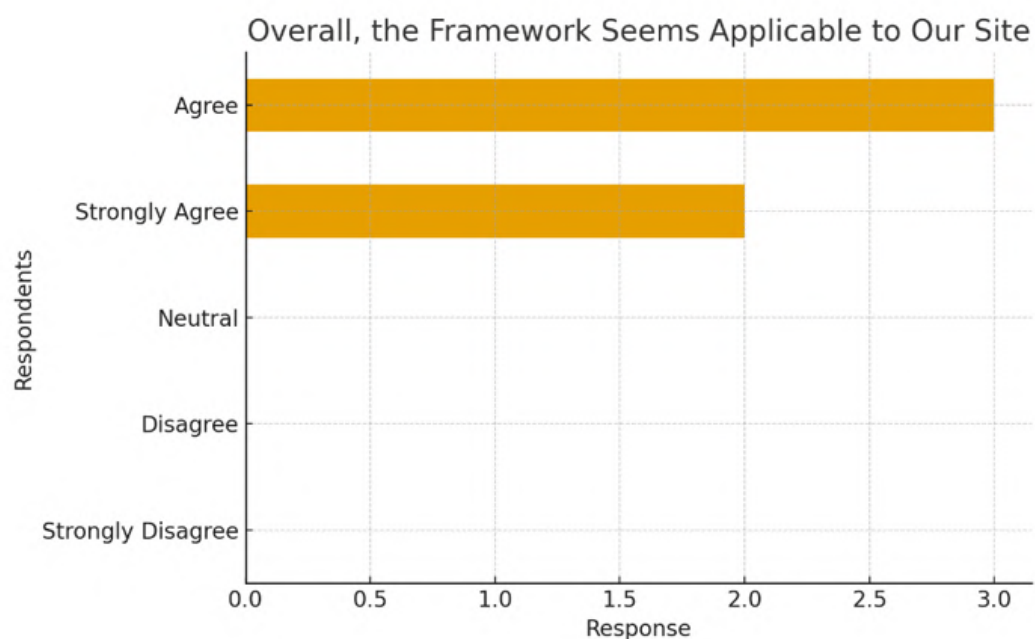


Figure 6.15. Applicability of framework.

International evidence consistently reports that **Revit**-integrated RTLS improves situational awareness and hazard management (e.g., UWB tags on workers/equipment+ receivers in risk zones), but organizational integration (policies, workflows) is the hard part: adoption requires re-engineering safety processes, training, and IT support (Perreira *et al.*, 2025). For example, studies demonstrate **Revit**+RTLS can enable rule-based alerts and workspace analytics for proactive safety planning, yet emphasize process change and governance to realize benefits.

Respondents most frequently flagged smartphone/GNSS devices (100%), and IT/network and server support ($\approx 67\%$ each) as major cost items; about one-third selected UWB hardware, custom integration, and training. Open-ended inputs and multi-selects highlight hardware costs, software/BIM licensing, budget/time planning, training, and IT security/compliance as key obstacles typical for RTLS+BIM deployments. Reviews of digital twins and BIM agree that while cyber-physical integration delivers strong safety and productivity gains, uptake is constrained by upfront investment, integration effort, and data/IT governance (Tuhaise *et al.* 2023). For site positioning specifically, UWB achieves decimeter-level indoor accuracy and robust update rates, but infrastructure (anchors, calibration, power/network) and NLoS sensitivities raise deployment complexity and cost (Alarifi *et al.*, 2016).

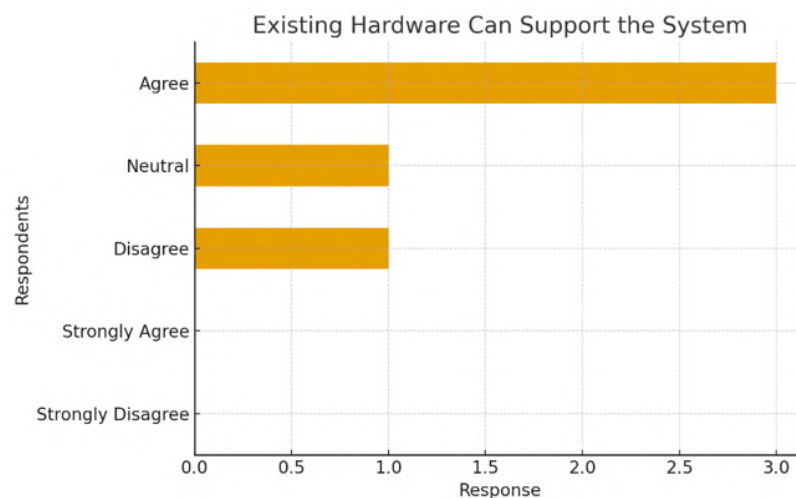


Figure 6.16. Hardware adequacy.

Despite low “active Revit use” at the firm level, 60% indicated that Revit models for their sites exist (agree/strongly-agree), 80% reported staff Revit experience, and 40% reported familiarity with pyRevit/plugin-ins a promising base for integration. IT security policies were widely perceived as restrictive (80% agreed they block external scripts).

Most respondents were positive on open-sky availability for GPS/GNSS (80%), time-sync and logging (60%), and coordinate transforms (a plurality selected “can with support / long process”), consistent with your proposed pyRevit/Revit API pipeline for CSV ingest, transformation (ECEF/ENU coordinate system to Revit referential plane), and avatar updates.

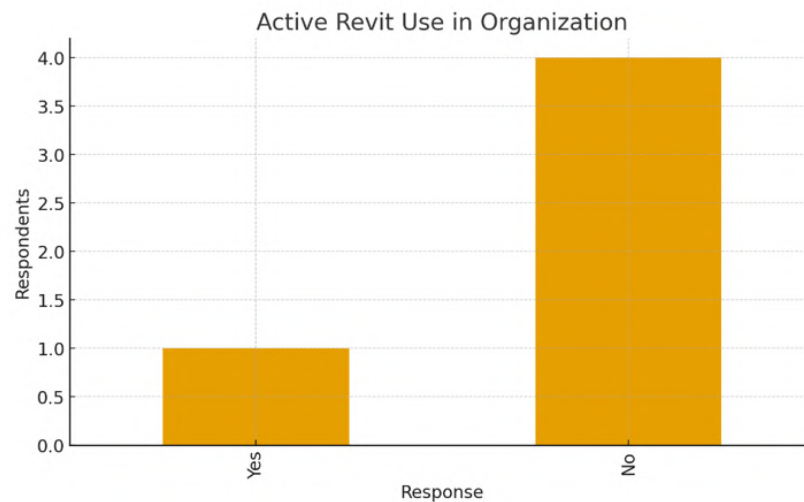


Figure 6.17. Active Revit usage in firm level.

With management support, respondents were split on pilot feasibility in 3–8 months (40% agree; 40% disagree), but 80% expect permanent use within 12+ months and 60% judge the framework applicable to their site overall.

The respondents’ emphasis on budget, licensing, training, and IT security mirrors international barriers but is particularly salient for Turkish SMEs that often operate with lean IT and project-by-project tooling. The survey’s open answers (“süreç, eğitim ve zaman”..., “yeterli kaynak ve eğitilmiş iş gücü”) align with global recommendations:

anchor adoption in pilot areas, management-backed training, and clear ROI by targeting high-risk activities first (e.g., fall-hazard zones, heavy-equipment interfaces).

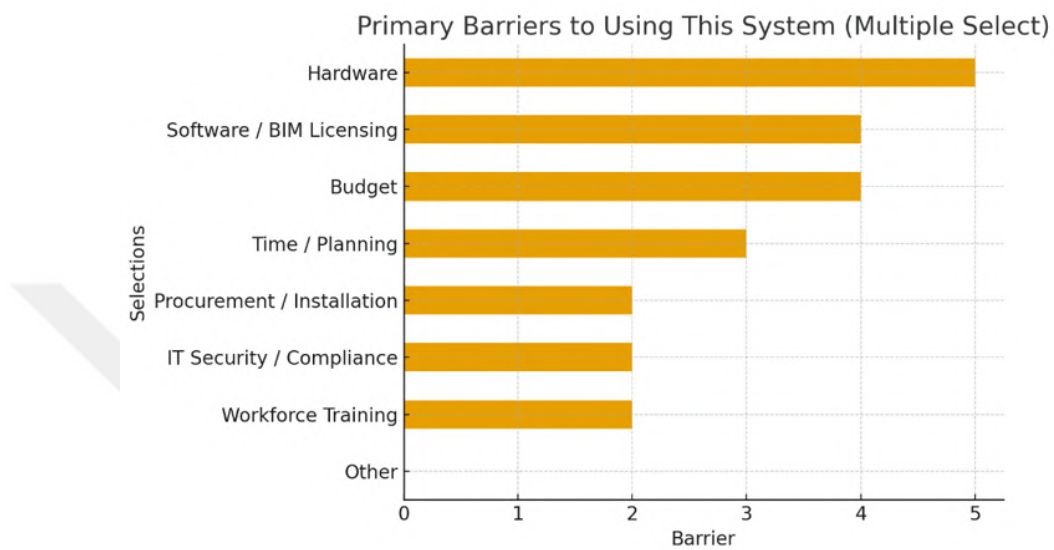


Figure 6.18. Primary barriers of the framework.

According to respondents GNSS devices, IT infrastructure and server operations are the main cost components of the framework.

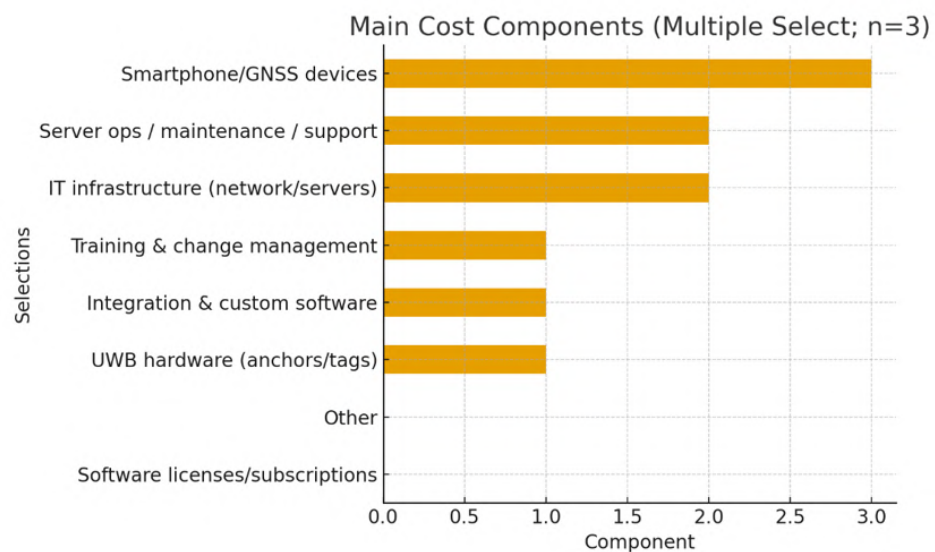


Figure 6.19. Main cost components of the framework.

Thesis observations meter-level errors under obstruction with better performance in open parking areas are consistent with global findings: smartphone GNSS can degrade to >10 m in urban canyons or near trees, whereas open sky yields smaller errors; robust positioning often requires multi-constellation, multi-frequency, or RTK augmentation. For indoor worker tracking, UWB typically achieves decimeter-level accuracy but is sensitive to NLoS (human body, metal, water) and requires anchor layout/commissioning. BLE/RSSI methods are lower-cost but generally meter-scale and unstable without dense calibration, while Wi-Fi RTT (FTM) can approach sub-meter in controlled setups but needs compatible infrastructure and error modeling often making UWB the pragmatic choice for safety-critical indoor zones, with GNSS/RTK or fusion outdoors. The literature increasingly couples UWB+GNSS (+INS) with UKF/EKF-based fusion to sustain tracking across indoor/outdoor transitions and to monitor workers in Revit. UWB anchor infrastructure is considered as deployable according to the respondent answer.

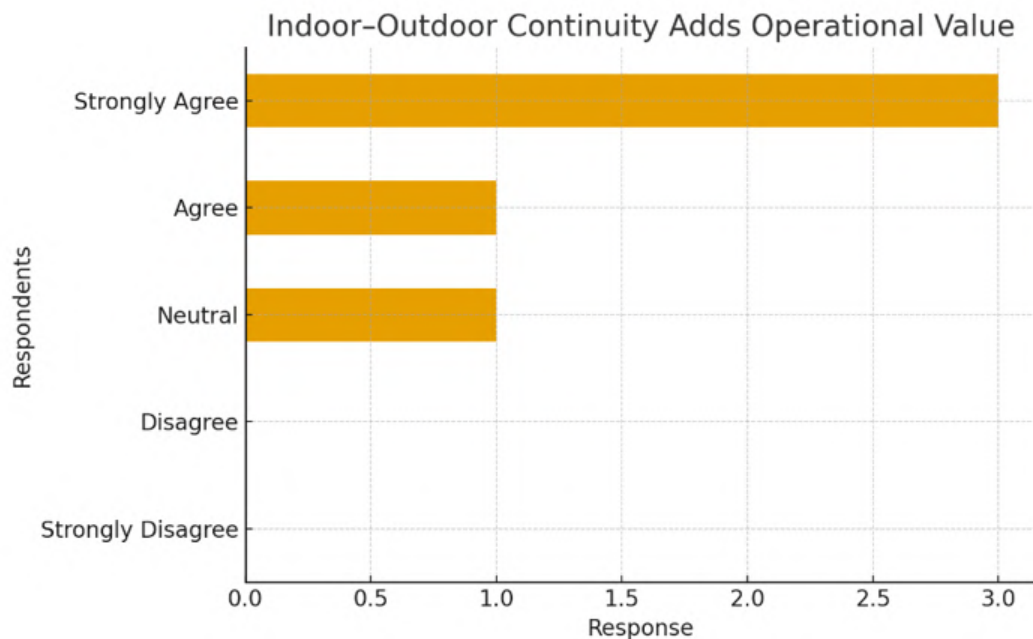


Figure 6.20. Operational value.

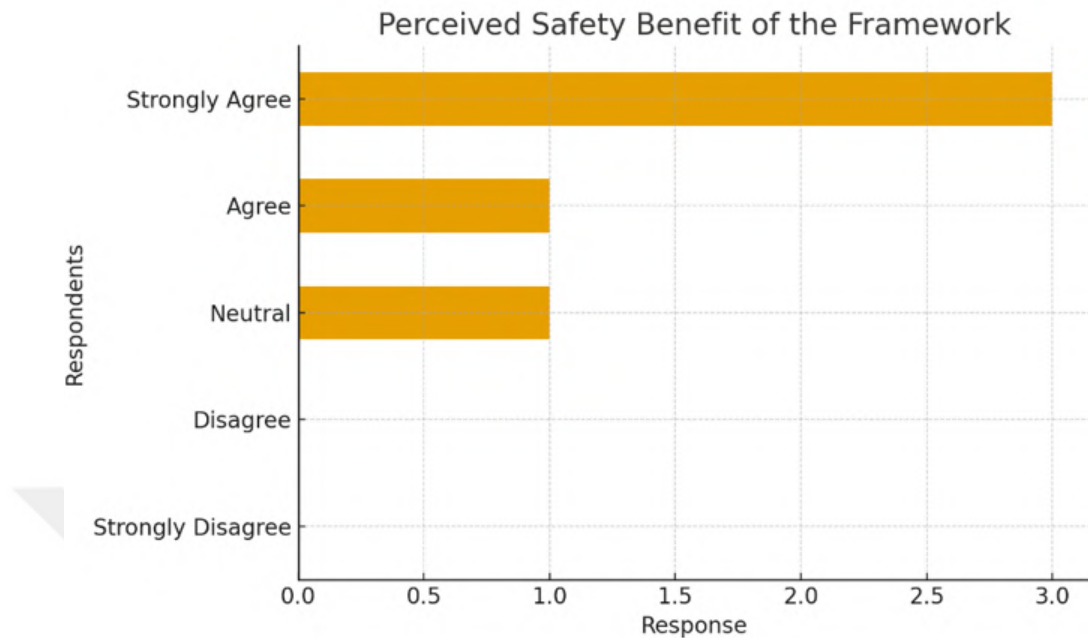


Figure 6.21. Safety benefit of framework.

Overall, according to figure 6.19 and Figure 6.21 respondents believe that the framework will enhance operational value of the construction safety management in indoor-outdoor construction sites. 80% of the respondents want to pilot the framework.

6.7. Limitations and Areas for Improvement

While this study successfully demonstrated the integration of GPS and UWB data into a BIM environment-based framework for safety monitoring, several key limitations were encountered, particularly during the data visualization and system implementation phases.

A significant limitation stemmed from the insufficient graphical processing capability (GPU) of the Revit workstation used during the project. The hardware struggled to handle the dense point cloud of worker location data, particularly during BIM-to-Enscape transfer. As a result, only a small portion of the full dataset could be visualized and tested within Revit and Enscape environments. When attempting to visualize the complete trajectory data (e.g., 1200+ fused points), the system experienced consider-

able lag, and the export process took up to three hours, severely restricting iteration and performance evaluation.

The outdoor data acquisition relied on iOS-based GNSS applications (e.g., GeoTracker, MyTracks). However, the accuracy of smartphone GPS was inconsistent, with errors occasionally reaching 10 meters, especially near buildings or trees where signal multipath or occlusion occurred. The average positional error (mean step vector) was approximately 6 meters, which significantly compromised spatial continuity and fusion reliability.

These deviations created false zone breach detections and unnatural movement artifacts when combined with UWB data. For improved accuracy, dedicated dual-frequency GNSS receivers (e.g., u-blox ZED-F9P) or differential correction techniques such as Real-Time Kinematic (RTK) GNSS are recommended for future studies. Alternatively, fusion with inertial measurement units (IMUs) on the device could be used to smooth and stabilize location transitions.

Due to the rendering and loading constraints, the amount of data visualized in the Revit model was artificially limited, which may have constrained the generalizability of the results. Although the processing scripts were applied to the full dataset, only selected frames or representative data subsets were used in the Revit model. This limitation hindered a complete spatial-temporal analysis of worker behavior or prolonged zone occupation.

Another bottleneck was the manual alignment required to map UWB and GPS coordinates to the internal Revit coordinate system. The lack of an automatic calibration workflow necessitated the use of reference anchors and scripting to perform the transformation resulting in a time-consuming, error-prone process. Small misalignments or orientation mismatches required additional correction during visualization.

Although the system was conceptually built for real-time tracking, the study operated in a post-processed mode due to limitations in stable data transmission, Revit refresh latency, and wireless synchronization. Implementing real-time streaming, especially in large projects, will require additional system robustness, possibly through edge computing or buffering mechanisms.

Future studies should prioritize (1) hardware upgrades for GPU-intensive tasks, (2) scalable and modular visualization methods, (3) high-precision outdoor GNSS sensors or RTK-enhanced smartphones, and (4) automation tools for coordinate transformation and alert management.

7. CONCLUSION AND FUTURE WORK

This section synthesizes the study’s contributions and practical significance, then outlines pathways for advancing the framework. How the objectives were met achieving continuous indoor-outdoor tracking via UKF fusion, BIM-native visualization on Revit, and before highlighting the main constraints uncovered by the experiments were restated. Building on these insights, chart future work focused on sensing and fusion (e.g., IMU augmentation; higher-grade or RTK/PPP GNSS), system engineering (robust time-sync; automated coordinate-frame rooting), and visualization at scale (cloud or game-engine front-ends; adaptive level-of-detail). The section closes with recommendations for multi-worker deployments, multiple integration of RTLS and integration into broader digital-twin safety innovation technologies.

7.1. Summary of Contributions

This thesis presents a novel cyber-physical framework that integrates Ultra-Wideband (UWB) and Global Positioning System (GPS) data into a Building Information Modeling (BIM) environment to enable nearly real-time construction worker tracking and safety monitoring. Key contributions of this work include:

- **Development of a Hybrid RTLS Framework:** A seamless indoor-outdoor worker tracking system was established by combining UWB and GPS data using an Unscented Kalman Filter (UKF), ensuring high accuracy and continuity across varying environments.
- **Revit Integration via Revit API and pyRevit:** The fused and filtered location data were mapped into a 3D Autodesk Revit model and custom Python scripts. This allowed for spatial visualization of worker movement and geofence-based safety alerts.
- **Outlier Detection with Mahalanobis Distance:** A stepwise outlier rejection approach using Mahalanobis distance was implemented to mitigate the effects of

noisy or non-line-of-sight (NLOS) UWB readings, improving localization reliability.

- **Experimental Validation with Real Data:** The framework was validated through field experiments using Raspberry Pi devices, Qorvo DWM1001 UWB modules, and smartphone-based GPS applications, demonstrating practical feasibility for real-world deployment.
- **Contributions to Safety 4.0 Paradigm:** The system lays foundational work for cognitive digital twins in construction, promoting adaptive, intelligent, and real-time safety interventions.

7.2. Review of Objectives and Findings

The research set out to achieve several specific objectives related to **nearly real-time worker tracking** monitoring in construction through UWB-GPS fusion and **BIM environment Revit** integration. Below is a summary of the objectives and the corresponding outcomes:

- **Seamless Indoor-Outdoor Worker Tracking** Achieved by combining GPS and UWB data in an Unscented Kalman Filter framework, enabling robust localization across different zones.
- **Nearly real-time location data-Revit Integration** accomplished through **pyRevit** scripting and **RevitAPI**, which visualized real-time worker locations directly within a spatially accurate **Revit** BIM model.
- **Outlier Rejection and Data Cleaning** Mahalanobis distance filtering removed over 2.5% of outlier points, reducing the impact of NLOS errors and sensor noise.
- **Visualization and Scenario Mapping** Motion trails were displayed in **Revit** and **Enscape**, supporting intuitive understanding of risk patterns.

- **Experimental Validation** The system was validated with over 1900 UWB data points and multiple GPS traces, with visual accuracy aligning well with reference anchors.

Overall, the proposed methodology achieved all key objectives, providing a validated proof-of-concept for a real-time, BIM environment-integrated worker monitoring framework and contributing new knowledge to the digital twin and safety analytics domains in construction.

7.3. Practical Implications for Construction Industry

The findings of this thesis demonstrate the feasibility and value of integrating Ultra-Wideband (UWB) and Global Positioning System (GPS) data into a Building Information Modeling (BIM) environment to enhance real-time worker tracking. The proposed digital twin framework, which fuses location data via an Unscented Kalman Filter (UKF) and maps it into Autodesk Revit through custom scripting, presents a scalable and cost-effective method for safety oversight in complex construction sites.

This approach offers several practical advantages for construction stakeholders. First, it facilitates continuous and automated safety monitoring, minimizing reliance on manual inspections. Additionally, the visual feedback system within Revit enhances situational awareness among project managers, allowing rapid, spatially informed responses to emerging risks. Finally, by incorporating historical and **nearly** real-time data into a unified platform, this framework contributes to long-term safety planning, risk analytics, and training efforts. The system also holds promise for integration with wearable sensors (e.g., physiological monitoring) and AI-based decision-support tools, further pushing the industry toward predictive, resilient, and human-centric safety management under the broader scope of “Safety 4.0”.

7.4. Recommendations for Future Research

While the prototype framework presented in this thesis validates the potential of GPS-UWB fusion for real-time worker tracking in BIM environments such as Revit model, several areas remain open for future exploration and enhancement:

- **Real-Time Deployment:** Future work should aim to operationalize the system in live construction sites to evaluate its performance under dynamic conditions, considering data latency, network disruptions, and real-time response mechanisms.
- **Integration with Physiological Sensors:** Merging worker biometric data (e.g., heart rate, fatigue, temperature) with spatial data can provide a holistic view of worker safety and enable detection of health-related risks.
- **AI-Based Hazard Prediction:** Incorporating machine learning models to analyze historical movement and environmental data may enhance predictive risk assessment and automated safety decision-making.
- **Interoperability with Industry Platforms:** The framework should be adapted for compatibility with other BIM platforms and open standards like IFC, to ensure wider adoption across heterogeneous construction projects.
- **Enhanced Visualization and VR:** Augmenting BIM with immersive visualization (e.g., VR/AR) can improve safety training, stakeholder engagement, and emergency planning simulations.
- **Scalability:** To ensure practical scalability, especially in large-scale projects, future research should explore decentralized processing methods such as edge computing and investigate ways to reduce computational load in BIM rendering environments. These directions will not only extend the utility of the proposed system but also contribute to establishing cyber-physical infrastructures capable of supporting intelligent, adaptive, and sustainable construction safety practices.

7.5. Closing Remarks

In conclusion, this thesis has presented a novel GPS-UWB-Revit integrated system for tracking construction workers, fulfilling the objectives of achieving seamless indoor/outdoor localization, robust data fusion with outlier mitigation, and meaningful visualization within a BIM context Revit. The research contributions from the stepwise Mahalanobis outlier detection method to the Unscented Kalman Filter fusion algorithm and the real-time BIM environment visualization demonstrate a solution to a long-standing challenge in construction management. The findings validate that such an integrated approach can significantly enhance situational awareness on job sites, improving safety and operational decision-making.

This work advances the state of the art in construction sensing and management by showing how multiple technologies can synergize to overcome individual limitations: the result is a continuous, high-accuracy tracking capability that neither GPS nor UWB could provide alone. Importantly, the practical implementation and testing underscore that the system is not merely a theoretical construct, but one that can be adapted to real project environments with current technology. The immediate implications for industry safer working conditions, more efficient workforce utilization, and richer data for project controls are compelling. As the construction sector continues to embrace digital tools and move toward smarter infrastructure, the integration of real-time localization data into the digital building model exemplifies the potential of Construction 4.0 strategies. It paves the way for truly intelligent construction sites where hazards are detected before incidents occur and where data-driven insights lead to continuous improvement in performance and safety.

Looking ahead, there is considerable scope to refine and build upon the foundation laid by this thesis. The recommendations for future research outline a path toward even more robust, scalable, and intelligent tracking systems, potentially transforming how construction projects are managed. By addressing the highlighted challenges and exploring emerging technologies, researchers and practitioners can accelerate the adop-

tion of these systems in the field. In closing, the integration of GPS, UWB, and BIM environment for worker tracking is a promising development for construction engineering. It not only demonstrates the feasibility of unifying disparate technologies into a cohesive solution but also heralds a new era where real-time information is leveraged to make construction sites safer, more efficient, and more accountable. It is hoped that this work will contribute to further studies at the intersection of sensing, data analytics, and construction management, ultimately contributing to a safer and smarter industry.



REFERENCES

- Agnusdei, G.P., V. Elia, and M.G. Gnani, 2021, A “Classification Proposal of Digital Twin Applications in the Safety Domain”, *Computers and Industrial Engineering*, Vol. 154, No. 1, pp.107137-107148.
- Ahmed, S., 2019a, “A Review on Construction Safety Performance and Influential Factors”, *International Journal of Construction Management*, Vol. 19, No. 4, pp.331-343.
- Airports International, 2022, “Case Study: Digital Twinning in Construction”, <https://www.airportsinternational.com/article/case-study-digital-twinning-construction>, accessed on 21 June 2025.
- Akanmu, A.A., C.J. Anumba, and O.O. Ogunseiju, 2021, “Towards Next Generation Cyber-Physical Systems and Digital Twins For Construction”, *Journal of Information Technology in Construction*, Vol. 26, No. 1, pp. 505-525.
- Al-Okby, M.F.R., S. Junginger, T. Roddelkopf, and K. Thürow, 2024, “UWB-Based Real-Time Indoor Positioning Systems: A Comprehensive Review”, *Applied Sciences*, Vol. 14, No. 23, pp. 11005-11025.
- Anuj, L., and M.T. Gopalakrishna, 2017, “Multiple Camera-Based Multiple Object Tracking Under Occlusion: A Survey”, *ResearchGate*, Vol. 1, No. 1, pp. 432-437.
- Barral, V., C.J. Escudero, J.A. García-Naya, and R. Maneiro-Catoira, 2019, “NLOS Identification and Mitigation Using Low-Cost UWB Devices”, *Sensors*, Vol. 19, No. 16, pp. 3464-3587.
- BIM Forum, 2021, “Level of Development (LOD) Specification Part I: For Building

Information Models“, <https://bimforum.org/lod/>, accessed on July 21, 2025.

Boje, C., A. Guerriero, S. Kubicki, and Y. Rezgui, 2020, “Towards a Semantic Construction Digital Twin: Directions for Future Research“, *Automation in Construction*, Vol. 114, No. 1, pp.103179-103189.

Boje, C., A. Guerriero, S. Kubicki, and Y. Rezgui, 2020, “Towards A Semantic Construction Digital Twin: Directions for Future Research“, *Automation in Construction*, Vol. 114, No. 1, pp. 103179-103185.

Borrmann, A., M. König, C. Koch, and J. Beetz, 2018, *Building Information Modeling: Technology Foundations and Industry Practice*, Springer, London.

Borrmann, A., M. König, C. Koch, and J. Beetz, 2018, *Building Information Modeling: Technology Foundations and Industry Practice*, Springer, London.

Bugalia, S. and S. Kedia, 2022, “Real-Time Safety Monitoring System for Construction Sites Using Internet of Things and Computer Vision“, *Automation in Construction*, Vol. 135, No. 1, pp.104156-104168.

Che Jailani, N.S., N. Abdul Wahab, N. Sunar, S.H. Syed Ariffin, K.Y. Wong, and A. Yc, 2022, “Indoor Positioning System: A review“, *International Journal of Advanced Computer Science and Applications*, Vol. 13i No. 6, pp.538-545.

Chen, Y.-Y., S.-P. Huang, T.-W. Wu, W.-T. Tsai, C.-Y. Liou, and S.-G. Mao, 2020, “UWB System for Indoor Positioning and Tracking with Arbitrary Target Orientation, Optimal Anchor Location, and Adaptive NLOS Mitigation“, *Institute of Electrical and Electronics Engineers (IEEE), Transactions on Vehicular Technology*, Vol. 69, No. 9, pp. 9304-9314.

Cheng, T. and J. Teizer, 2013, “Real Time Resource Location Data Collection and

Visualization Technology for Construction Safety and Activity Monitoring Applications“, *Automation in Construction*, Vol. 34, No. 1, pp. 3-15.

Corbalán, P., G.P., Picco, M. Coors, and V., 2023, “Jain, Self-localization of Ultra-Wideband Anchors: From Theory to Practice, Institute of Electrical and Electronics Engineers (IEEE)“, <https://disi.unitn.it/picco/papers/access23.pdf>, accessed on July 21, 2025.

Costin, A., N. Pradhananga, and J., Teizer, 2015, “Leveraging Passive RFID for Construction Resource Field Mobility and Status Monitoring in A High-Rise Renovation Project“, *Automation in Construction*, Vol. 55, No. 1, pp. 39-52.

Costin, A., N. Pradhananga, and J., Teizer, 2015. “Leveraging Passive RFID for Construction Resource Field Mobility and Status Monitoring in A High-Rise Renovation Project“, *Automation in Construction*, Vol. 49, No. 1, pp.86-97.

de Blasio, G.S., J.C., Rodríguez-Rodríguez, C.R., García, and A. Quesada-Arencibia, 2019, “Beacon-Related Parameters of Bluetooth Low Energy: Development of a Semi-Automatic System to Study Their Impact on Indoor Positioning Systems“, *Sensors*, Vol. 19, No. 14, pp. 3087-3099.

De Maesschalck, R., D. Jouan-Rimbaud, and D.L., Massart, 2000, “The Mahalanobis Distance“, *Chemometrics and Intelligent Laboratory Systems*, Vol. 50, No. 1, pp.1-18.

Dekker, S., 2014, “The Field Guide to Understanding Human Error“, 3rd ed. Boca Raton: Chemical Rubber Company (CRC) Press, Paris.

El-Rabbany, N.A., and Abo-Hashem, A., 2016, “Evaluation of Area Deformation Due to Coordinate Transformation from ETM to UTM 35N“, *International Journal of Engineering Research and Technology (IJERT)*, Vol. 5, No. 7, pp. 483-488.

- Feng, T., S., Chen, Z., Feng, C. Shen, and Y., Tian, 2021, “Effects of Canopy and Multi-Epoch Observations on Single-Point Positioning Errors of a GNSS in Coniferous and Broadleaved Forests“, *Remote Sensing*, Vol. 13, No. 12, pp. 2325-2336.
- Forteza, F.J., J.M. Carretero-Gómez, and A., Sesé, 2020, “Occupational Accidents in the Construction Sector: Different Models to Analyze Their Causes and Consequences“, *Journal of Cleaner Production*, Vol. 248, No. 1, pp. 119209-119215.
- Fronckova, Z., and P. Prazak, 2020, “Application of Kalman Filter for Object Localization Based on Radio Measurements“, *Multidisciplinary Digital Publishing Institute (MDPI) Sensors*, Vol. 20, No. 22, pp. 1-18.
- Gentner, C. and G., Retscher, 2012, “Fusion of UWB and GPS for Pedestrian Navigation in Urban and Indoor Environments“, *Journal of Applied Geodesy*, Vol. 6, No. 3, pp.163-174.
- Ghodsian, N., 2024, “Predictive Analytics in Construction Safety: 4 Examples from“, <https://neuroject.com/predictive-analytics-in-construction/>, accessed on July 22, 2025.
- Giretti, A., A., Carbonari, B. Naticchia, and M., De Grassi, 2009, “Design and First Development of an Automated Real-Time Safety Management System for Construction Sites“, *Journal of Civil Engineering and Management*, Vol. 15, No. 4, pp.325-336.
- Gnanadesikan, R. and J.R., Kettenring, 1972, “Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data“, *Biometrics*, Vol. 28, No. 1, pp. 81-124.
- Gong, J., C. Yao, A.H. Weerts, Z. Li, X. Wang, J. Xu, and Y., Huang, 2024, “State Updating in Xin’anjiang Model by Asynchronous Ensemble Kalman Filtering with Enhanced Error Models“, *Journal of Hydrology*, Vol. 640, No. 1, pp. 131726-131736.

- Grewal, M.S., and A.P. Andrews, 2010, *Kalman Filtering: Theory and Practice Using MATLAB (3rd ed.)*, Wiley, Colorado.
- Grieves, M. and J. Vickers, 2017, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. In: *Transdisciplinary Perspectives on Complex Systems*, Springer, London.
- Groves, P.D., 2013, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Norwood, MA: Artech House.
- Gucma, M., R. Boc, and M. Waz, 2021, "Reliability Measures in GNSS Multi-constellation Systems", *European Research Studies Journal*, Vol. 24, No. 3, pp. 410-422.
- Guo, B.H.W., and T.W. Yiu, 2013, "Developing Leading Indicators to Monitor the Safety Conditions of Construction Projects", *Journal of Management in Engineering*, Vol. 32, No. 1, pp. 04015016-04015036.
- Hadi, A.S., A.H.M.R. Imon, and M., Werner, 2009, "Detection of Outliers", *WIREs Computational Statistics*, Vol. 1, No. 1, pp. 57-70.
- Hashmi, F., M.U. Hassan, M.U. Zubair, K. Ahmed, T. Aziz, and R.M. Choudhry, 2024, "Near-Miss Detection Metrics: An Approach to Enable Sensing Technologies for Proactive Construction Safety Management", *Buildings*, Vol. 14, No. 4, pp. 1005-1015.
- Hofmann-Wellenhof, B., H. Lichtenegger, and E., Wasle, 2008, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*, Springer, London.
- Hollnagel, E., 2014, *Safety-I and Safety-II: The Past and Future of Safety Management*, Ashgate, New York.

- Hong, Y. and F. Guo, 2025, "A Framework of BIM-Iot Application in Construction Projects Through Multiple Case Study Approach", *Journal of Building Design and Environment*, Vol. 3, No. 1, pp. 202438-202458.
- Hossain, M.A. and S. Ahmed, 2022, *Improving Construction Safety Performance Through Real- Time Safety Risk Assessment*, Engineering, Construction and Architectural Management, Paris.
- Howell, G.A., A. Laufer, and G., Ballard, 2002, "Uncertainty and Project Objective". *Journal of Construction Engineering and Management*, Vol. 128, No. 1, pp. 48-54.
- Hsu, L.T., 2018, "Analysis and Modeling GPS NLOS Effect in Highly Urbanized Area", *GPS Solutions*, Vol. 22, No. 7, pp. 1-23.
- Hu, W., K.Y.H. Lim, and Y. Cai, 2022, "Digital Twin and Industry 4.0 Enablers in Building and Construction: A Survey". *Buildings*, Vol. 12, No. 11, pp. 204-212.
- Huang, X., B. Wang, and C. Wu, 2022, *Realizing Smart Safety Management in the Era of Safety 4.0: A New Method Towards Sustainable Safety*, Sustainability, New York.
- Huang, Y., A. Hammad, Z. Zhu, 2020, *Providing Proximity Safety Alerts to Workers on Construction Sites Using Bluetooth Low Energy RTLS*, *Creative Construction e-Conference Proceedings*, Budapest, Hungary.
- Kadam, P. and G. Fang, and J. Zou, 2024, "Object Tracking Using Computer Vision: A Review", *Computers*, Vol. 13, No. 1, pp. 136-152.
- Kaplan, E.D. and C.J. Hegarty, 2017, *Understanding GPS/GNSS: Principles and applications*, 3rd ed. Norwood, MA: Artech House, New York.
- Kasim, N., S. Razali, and R. Kasim, 2021, *Reinforce Technology IR 4.0 Implementation*

for Improving Safety Management in Construction Site, International Journal of Sustainable Construction Engineering and Technology, Paris.

Kim, H. and S. Han, 2018, “Accuracy Improvement of Real-Time Location Tracking for Construction Workers“, *Sustainability*, Vol. 10, No. 1, pp. 1488-1499.

Kong, Q., 2024, “WiFi-aided Ultra Wideband Localization In Indoor NLOS Environment“, *Institute of Electrical and Electronics Engineers IEEE Communications Letters*, Vol. 28, No. 3, pp. 537-540.

Leick, A., L. Rapoport, and D. Tatarnikov, 2015, *GPS Satellite Surveying*, 4th ed. Hoboken: John Wiley and Sons, New York.

Leitch, S.G., Q.Z. Ahmed, W.B. Abbas, M. Hafeez, P.I. Laziridis, P. Sureephong, and T. Alade, 2023, “On Indoor Localization Using WiFi, BLE, UWB, and IMU Technologies“, *Sensors*, Vol. 23, No. 20, pp. 8598-8615.

Li, C.T., J.C.P. Cheng, and K. Chen, 2020, “Top 10 Technologies for Indoor Positioning on Construction Sites“, *Automation in Construction*, Vol. 118, No. 1, pp. 103309-103329.

Li, H., M. Lu, S.C. Hsu, M. Gray, and T. Huang, 2015, “Proactive Behavior-Based Safety Management for Construction Safety Improvement“, *Safety Science*, Vol. 75, No. 1, pp. 107-117.

Li, N., and B. Becerik-Gerber, 2011, “Performance-Based Evaluation of RFID-Based Indoor Location Sensing Solutions for the Built Environment“, *Advanced Engineering Informatics*, Vol. 25, No. 1, pp. 535-546.

Li, X., C. Chen, Y. Lou, M. Abdallah, K.T. Kim, and S. Bagchi, 2024, “HopTrack: A Real-Time Multi-Object Tracking System for Embedded Devices“, <https://doi.org/10.1016/j.scs.2024.111111>.

org/ 10. 48550/ arXiv. 2411.00608, accessed on July 22, 2025.

Litum, 2019, “Worker Tracking and Safety in Construction: RTLS Brings Security to Paris Metro“, <https://www.litum.com/blog/worker-tracking-safety-in-construction-rtls-brings-security-to-paris-metro/>, accessed on July 23, 2025.

Liu, W., Q. Meng, Z. Li, and X. Hu, 2021, “Applications of Computer Vision in Monitoring the Unsafe Behavior of Construction Workers: Current Status and Challenges“, *Buildings*, Vol. 11, No. 9, pp. 409-415.

Maghdid, S. and M. Halgurd, 2021, *A Comprehensive Review of Indoor/Outdoor Localization Solutions in IoT era*, Research Challenges and Future Perspectives, Paris.

Mahalanobis, P.C., 1936, “On the Generalized Distance in Statistics“, *Proceedings of the National Institute of Sciences of India*, Vol. 2, No. 1, pp. 49-55.

Mastrolembo V.S., P. Bellagente, S. Rinaldi, A. Flammini, and A.L.C. Ciribini, 2023, “Enhancing Safety on Construction Sites: A UWB-Based Proximity Warning System Ensuring GDPR Compliance to Prevent Collision Hazards“, *Sensors*, Vol. 23, No. 24, pp. 9770-9789.

Mautz, R., 2012, “Indoor Positioning Technologies, Habilitation Thesis, ETH Zurich“, <https://doi.org/10.3929/ethz-a-007313554>, accessed on July 24, 2025.

Meng, Y., M.F. Antwi-Afari, I. Mehmood, S. Anwer, and W. Umer, 2024, Emerald Publishing Limited, Emerald Publishing Limited, London.

Muñoz-La Rivera, F., E. Oñate, F. Lahuerta, and R. Yagüe, 2021, “Situational Awareness for Improving Safety in Construction Projects: A Review“, *Safety Science*, Vol. 139, No. 1, pp. 105250-105265.

- Nguyen, T.D., and Adhikari, 2023, “The Role of BIM in Integrating Digital Twin in Building Construction: A Literature Review“, *Sustainability*, Vol. 15, No. 13, pp. 10462-10478.
- Opoku, D.G., S. Perera, and R. Osei-Kyei, 2021, “Digital Twin Application in the Construction Industry: A Literature Review“, *Journal of Building Engineering*, Vol. 40, No. 1, pp. 102726-102746.
- Pan, Y. and L. Zhang, 2021, “A BIM-Data Mining Integrated Digital Twin Framework for Advanced Project Management“, *Automation in Construction*, Vol. 124, No. 1, pp. 103564-103574.
- Park, J.-S., D.-G. Lee, J. A. Jimenez, S.-J. Lee, and J.-W. Kim, 2023, “Human-Focused Digital Twin Applications for Occupational Safety and Health in Workplaces: A Brief Survey and Research Directions“, *Applied Sciences*, Vol. 13, No. 7, pp. 4598-4610.
- Peng, P., C. Yu, Q. Xia, Z. Zheng, K. Zhao, W. Chen, 2022, “An Indoor Positioning Method Based on UWB and Visual Fusion“, *Sensors*, Vol. 22, No. 1, pp. 1394-1410.
- Peng, X., K. Wu, S. Li, L. Wen, and K. Zheng, 2022, “Joint Optimization of Anchor Placement and Signal Feature Selection for Improved UWB-Based Localization“, *Institute of Electrical and Electronics Engineers IEEE Internet of Things Journal*, Vol. 9, No. 2, pp. 1246-1260.
- Pereira, A., M. de las Nieves González García, J.P. Martins, 2025, “BIM-Integrated IoT Solutions for Real-Time Construction Site Monitoring“, *Applied Sciences*, Vol. 15, No. 4, pp. 2218-2232.
- Psimoulis, P.A., V.A. Kontogianni, A. Nickitopoulou, S.I. Pytharouli, P. Triantafyllidis, and S.C. Stiros, 2004, “Estimating the Optimum Duration of GPS Static

- Observations for Short Baseline Length Determination in Greece“, *FIG Working Week*, Vol. 1, No. 1, pp. 22-24.
- Rasmussen, J., 1997, “Risk Management in A Dynamic Society: A Modelling Problem“, *Safety Science*, Vol. 27, No. 2-3, pp. 183-213.
- Ridolfi, M., S. Van de Velde, H. Steendam, and E. De Poorter, 2018, “Analysis of the Scalability of UWB Indoor Localization Solutions for High User Densities“, *Sensors*, Vol. 18, No. 6, pp. 1875-1885.
- Rocke, D.M. and D.L. Woodruff, 1996. “Identification of Outliers in Multivariate Data“, *Journal of the American Statistical Association*, Vol. 91, No. 435, pp. 1047-1061.
- Rodas J., V. Barral, C.J. Escudero, 2013, “Architecture for Multi-Technology Real-Time Location Systems“, *Sensors (Basel)*, Vol. 13, No. 2, pp. 2220-2230.
- Ruppert, T. and J. Abonyi, 2020, “Integration of Real-Time Locating Systems into Digital Twins“, *Journal of Industrial Information Integration*, Vol. 20, No. 1, pp. 100174-100184.
- Ruppert, T. and J. Abonyi, 2020, “The Role of Digital Twins in Managing Human-Centered Production Systems“, *Procedia Manufacturing*, Vol. 51, No. 1, pp. 1125-1132.
- Sacks, R., I. Brilakis, E. Pikas, and H. Xie, 2020, “Construction with Digital Twin Information Systems: Control, Monitoring, and Feedback“, *Automation in Construction*, Vol. 114, No. 1, pp. 103160-103175.
- Şahin, S. and E. Kurum, wolfj, 2020, “Integration of GIS and BIM for Infrastructure Asset Management: A Case Study from Turkey“, *Journal of Geomatics*, Vol. 15,

No. 1, pp. 47-56.

Setayeshgar, S., A. Hammad, F. Vahdatikhaki, and C. Zhang, 2013, *Real-Time Safety Risk Analysis of Construction Projects Using BIM and RTLS, Proceedings of the 30th International Symposium on Automation and Robotics in Construction (IS-ARC)*, Montréal, Canada.

Seyed, A., R. Zekavat, 2019, *Michigan Technological University; R. Michael Buehrer, Virginia Tech*, Wiley-Institute of Electrical and Electronics Engineers Press.

Shamsollahi, D., O. Moselhi, and K. Khorasani, 2024, “Data Integration Using Deep Learning and Real-Time Locating System (RTLS) for Automated Construction Progress Monitoring and Reporting“, *Automation in Construction*, Vol. 106, No. 1, pp. 105261-105289.

Shin, S. and J. Jung, 2022, “Integration of UWB and GPS using Kalman filter for Seamless Indoor-Outdoor Positioning“. *Journal of Location Based Services*, Vol. 16, No. 1, pp. 45-59.

Soltanmohammadlou, N., S. Sadeghi, C.K.H. Hon, and F. Mokhtarpour-Khanghah, 2019, “Real-Time Locating Systems and Safety Construction Sites: A Literature Review“, *Safety Science*, Vol. 117, No. 1, pp. 229-242.

Sorescu, T.G., A.V. Militaru, V.M. Chiriac, C.R. Comsa, and I.E. Alecsandrescu, 2024, “UWB Indoor Localization: Accuracy Evaluation in a Controlled Environment“. *In: 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Institute of Electrical and Electronics Engineers*, Iasi, Romania.

Su, S. Y. Ray, Zhong, Y, Jiang, J. Song, Y. Fu, H. Cao, 2023, “Digital Twin and its Potential Applications in Construction Industry: State-of-Art Review and A Conceptual Framework“, *Advanced Engineering Informatics*, Vol. 57, No. 1, pp.

102030-102060.

Taghinezhad, N., F. Nasirzadeh, and M. Khanzadi, 2021, “An Integrated Model for Assessing Construction Labor Productivity Considering Sustainability Indicators: A System Dynamics Approach“, *Journal of Cleaner Production*, Vol. 288, No. 1, pp. 125583-125599.

Teunissen, P.J.G. and O. Montenbruck, 2017, *Springer Handbook of Global Navigation Satellite Systems*, Cham: Springer, London.

Topal, Y., 2025, *Investigation of Accuracy in Short Observations with Using Precise Point Positioning Technique*, MSc Thesis. Yıldız Technical University.

Topcon, 2020, *HiPer VR GNSS Receiver Specification Sheet*, Topcon Positioning Systems.

Tuhaise, V.V., J.H.M. Tah, and F.H. Abanda, 2023, “Technologies for Digital Twin Applications in Construction“, *Automation in Construction*, Vol. 152, No. 1, pp. 104931-104951.

Volk, R., J. Stengel, and F. Schultmann, 2014, “Building Information Modeling (BIM) for Existing Buildings Literature Review and Future Needs“, *Automation in Construction*, Vol. 38, No. 1, pp. 109-127.

Wan, E.A. and R. van der Merwe, 2000, “The Unscented Kalman Filter For Non-linear Estimation, In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium“, *Institute of Electrical and Electronics Engineers (IEEE)*, Vol. 1, No. 1, pp. 153-158.

Want, R., 2006, “An Introduction to RFID Technology, Pervasive Computing“, *Electrical and Electronics Engineers (IEEE)*, Vol. 5, No. 1, pp. 25-33.

- Welch, G., and G. Bishop, 2006, “An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill“, [https:// www. cs. unc. edu/ welch/ media/ pdf/ kalman_ intro.pdf](https://www.cs.unc.edu/welch/media/pdf/kalman_intro.pdf), accessed on July 23, 2025.
- Wolf, P.R. and C.D. Ghilani, 2012, *Elementary Surveying: An Inzektroduction to Geomatics*, 13th ed. Boston.
- X. Li, Y. Wang, and K. Khoshelham, 2018, “A Robust and Adaptive Complementary Kalman Filter Based on Mahalanobis Distance for Ultra Wide-band/Inertial Measurement Unit Fusion Positioning“, *Sensors*, Vol. 18, No. 10, pp. 1-25.
- Yan, J., Z. Zhang, and Y. Tang, 2021, “Indoor Positioning Using Maximum Correntropy Criterion Unscented Kalman Filter and TDOA“, *Institute of Electrical and Electronics Engineers (IEEE)*, Vol. 9, No. 1, pp. 124681-124694.
- Yang, Zhou and C. Tang, T. Zhang, Z. Zhang, and D.T. Doan, 2024, “Digital Twins in Construction: Architecture, Applications, Trends and Challenges“. *Buildings*, Vol. 14, No. 9, pp. 2616-2636.
- Yassin, M., Z. Liu, D.B. Rawat, M.-S. Alouini, and H. Gharavi, 2017, “Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications“, *Institute of Electrical and Electronics Engineers (IEEE) Communications Surveys and Tutorials*, Vol. 19, No. 2, pp. 1327-1346.
- You, J., S. Lee, K. Kim, and H. Park, 2019, “A Hybrid UWB-GPS Real-Time Localization System Using UKF for Construction Applications“. *Sensors*, Vol. 19, No. 20, pp. 4507-4515.
- Yu, K., K. Wen, Y. Li, S. Zhang, and K. Zhang, 2019, “A Novel NLOS Mitigation Algorithm for UWB Localization in Harsh Indoor Environments“, *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Vehicular Technology*, Vol.

68, No. 1, pp. 686-700.

Yuan, D., J. Zhang, J. Wang, X. Cui, F. Liu, and Y. Zhang, 2021, “Robustly Adaptive EKF PDR/UWB Integrated Navigation Based on Additional Heading Constraint“, *Sensors*, Vol. 21, No. 13, pp. 4390-4410.

Zafari, F., A. Gkelias, and K.K. Leung, 2019, “A Survey of Indoor Localization Systems and Technologies“, *Institute of Electrical and Electronics Engineers (IEEE), Communications Surveys and Tutorials*, Vol. 21, No. 3, pp. 2568-2599.

Zhang, Q., Z. Liu, and S. Yang, 2025, “Enhancing Construction Workers’ Health and Safety: Mechanisms for Implementing Construction 4.0 Technologies in Construction Organizations“, *Engineering, Construction and Architectural Management*, Vol. 32, No. 13, pp. 68-103.

Zhang, S. and T. Jochen and J.-K. Lee, and C. Eastman, and M. Venugopal, 2013, “Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules“, *Automation in Construction*, Vol. 29, No. 1, pp. 183-195.

Zheng, Y., F. Zheng, C. Yang, G. Nie, and S. Li, 2022, “Analyses of GLONASS and GPS+GLONASS Precise Positioning Performance in Different Latitude Regions“, *Remote Sensing*, Vol. 14, No. 18, pp. 4640-4652.

Zhong, X., R. Jin, and L. Ding, 2022, “Development of a Real-Time Construction Safety Monitoring System Using the Integration of BIM and RTLS“, *Automation in Construction*, Vol. 136, No. 1, pp. 104154-104165.

APPENDIX A: MAHALONOBIS RESULT OF UWB DATASET IN THE ENGINEERING FACULTY

The Mahalanobis distance is a multivariate statistical method used to measure how far a data point is from the mean of a distribution, considering the covariance structure. In the context of UWB, it is mainly applied for outlier detection, identifying measurements that deviate significantly due to noise, signal blockage, or non-line-of-sight (NLoS) effects.

Table A.1. Subject 1 example data sources.

Timestamp	X	Y	Z
1748432788587,00	37.54	-28.64	0.13
1748432788588,00	37.55	-28.64	0.17
1748432788588,00	37.56	-28.65	0.12
1748432788589,00	37.53	-28.65	0.19
1748432788590,00	37.6	-28.61	0.46
1748432788590,00	37.62	-28.6	0.45
1748432788803,00	37.57	-28.61	0.26
1748432788903,00	37.57	-28.61	0.27
1748432789003,00	37.54	-28.62	0.18
1748432789103,00	37.54	-28.62	0.18
1748432789203,00	37.55	-28.62	0.2
1748432789303,00	37.56	-28.63	0.15
1748432789402,00	37.59	-28.62	0.26
1748432789502,00	37.6	-28.62	0.29
1748432789603,00	37.57	-28.64	0.28
1748432789703,00	37.58	-28.63	0.34
1748432789803,00	37.57	-28.66	0.26
1748432789902,00	37.56	-28.66	0.17

Table A.1 Subject 1 example data sources. (cont.)

Timestamp	X	Y	Z
1748432790003,00	37.55	-28.65	0.15
1748432790103,00	37.56	-28.62	0.14
1748432790203,00	37.6	-28.61	0.23
1748432790303,00	37.61	-28.6	0.27
1748432790403,00	37.61	-28.61	0.25
1748432790503,00	37.6	-28.6	0.23
1748432790603,00	37.6	-28.6	0.26
1748432790703,00	37.59	-28.6	0.28
1748432790803,00	37.59	-28.6	0.25
1748432790903,00	37.59	-28.61	0.27
1748432791003,00	37.61	-28.61	0.3
1748432791103,00	37.6	-28.61	0.3
1748432791203,00	37.59	-28.61	0.27
1748432791303,00	37.6	-28.61	0.31
1748432791403,00	37.58	-28.61	0.28
1748432791503,00	37.56	-28.62	0.18
1748432791603,00	37.58	-28.62	0.24

APPENDIX B: MAHALONOBIS OUTLIER DETECTION RESULT OF UWB DATASET IN THE DATA SCIENCE AND THE ARFICIAL INTELLIGENCE INSTITUTE BUILDING

The Mahalanobis distance is a multivariate statistical method used to measure how far a data point is from the mean of a distribution, considering the covariance structure. In the context of UWB, it is mainly applied for outlier detection, identifying measurements that deviate significantly due to noise, signal blockage, or non-line-of-sight (NLoS) effects.

Table B.1. Data sources.

Timestamp	X	Y	Z	Mahalanobis Step3D²	Outlier Stepwise Mahalanobis
1748432930275,0	3.91	-5.09	1.18	1.489.006.408	TRUE
1748432914076,0	4.6	-5.38	1.19	2.298.181.948	TRUE
1748432926577,0	5.31	-5.36	1.24	1.386.658.869	TRUE
1748432934878,0	6.73	-6.85	0.98	6.474.667.375	TRUE
1748432942678,0	9.41	-7.09	0.82	1.449.242.582	TRUE
1748432960284,0	9.12	-7.32	0.56	1.287.376.815	TRUE
1748432964380,0	6.82	-5.23	1.83	1.586.636.951	TRUE
1748432979482,0	8.2	-7	0.46	1.369.164.202	TRUE
1748432992182,0	8.62	-6.61	0.97	2.073.530.008	TRUE
1748433128792,0	24.48	-15.26	1.67	2.561.950.388	TRUE
1748433128892,0	19.78	-12.55	1.78	2.646.837.588	TRUE
1748433128992,0	16.3	-10.47	1.92	173.122.296	TRUE
1748433138292,0	13.59	-10.2	0.68	1.572.926.986	TRUE

Table B.1 Data sources. (cont.)

Timestamp	X	Y	Z	Mahalanobis Step3D²	Outlier Stepwise Mahalanobis
1748433217097,0	24.6	-16.3	-0.26	1.696.567.405	TRUE
1748433217197,0	19.59	-13.4	0.08	3.390.915.781	TRUE
1748433217298,0	16.16	-11.21	0.62	2.968.668.007	TRUE
1748433217397,0	17.33	-9.47	0.85	2.521.798.294	TRUE
1748433217497,0	11.9	-8.2	1.03	1.182.199.894	TRUE
174843305701,0	24.99	-15.61	2.09	1.940.749.263	TRUE
174843305801,0	20.22	-12.82	2.16	2.879.304.856	TRUE
174843305901,0	16.65	-10.82	1.66	2.215.737.837	TRUE
174843306001,0	13.94	-9.35	1.14	1.550.542.214	TRUE
174843306701,0	7.41	-5.62	2.25	1.992.225.144	TRUE
174843306801,0	7.06	-5.61	2.61	1.292.630.911	TRUE
174843400205,0	23.98	-15.08	1.24	157.734.624	TRUE
174843400305,0	19.39	-12.37	1.15	289.158.019	TRUE
174843400405,0	15.99	-10.35	1.05	1.684.884.285	TRUE
174843400505,0	13.44	-8.8	1.36	1.228.550.342	TRUE
174843400204,0	6.15	-5.5	1.54	1.664.129.593	TRUE
1748434381019,0	15.18	-8.76	0.53	1.159.412.654	TRUE
1748434381119,0	11.82	-6.43	0.8	6.537.608.074	TRUE
1748434381219,0	9.34	-4.71	0.87	3.700.908.422	TRUE
1748434381319,0	7.44	-3.38	0.99	2.163.627.224	TRUE
1748434382019,0	2.61	0.1	1.59	2.136.126.202	TRUE
1748434382719,0	2.95	-0.85	1.52	1.584.671.559	TRUE
1748434384018,0	2.95	-0.85	1.52	1.584.671.559	TRUE
1748434394518,0	6.3	-5.44	1.43	1.649.792.906	TRUE
1748434394621,0	6.25	-5.54	1.89	1.523.173.018	TRUE
1748434394816,0	6.12	-5.78	1.37	1.340.457.893	TRUE
1748434402919,0	13.33	-10.1	-0.28	1.650.335.892	TRUE

Table B.1 Data sources. (cont.)

Timestamp	X	Y	Z	Mahalanobis Step3D²	Outlier Stepwise Mahalanobis
1748434501719,0	17.88	-11.56	0.69	1.230.347.422	TRUE
1748434501819,0	14.92	-9.72	0.99	419.333.803	TRUE
1748434501919,0	12.69	-8.33	1.22	2.387.768.579	TRUE
1748434502019,0	10.96	-7.29	1.28	1.329.139.334	TRUE
1748434510719,0	2.25	0.36	0.94	1.178.277.863	TRUE
1748434521819,0	6.51	-4.73	1.95	1.186.159.343	TRUE
1748434528119,0	9.67	-8.7	0.14	1.328.110.134	TRUE
1748434640519,0	17.8	-11.23	0.2	1.741.271.886	TRUE
1748434640621,0	14.36	-8.84	0.32	7.108.272.043	TRUE
1748434640719,0	11.67	-7.12	0.65	3.643.992.886	TRUE
1748434640819,0	9.7	-5.82	0.88	2.027.753.588	TRUE
1748434643621,0	5.08	-3.36	1.92	130.101.188	TRUE
174843465020,0	14.23	-10.27	0.66	664.931.995	TRUE
174843465420,0	14.45	-10.69	0.07	1.196.836.036	TRUE
1748434657320,0	15.12	-11.26	-0.71	1.312.536.383	TRUE
1748434770419,0	20.48	-13.18	0.35	2.005.988.539	TRUE
1748434770519,0	16.51	-10.55	0.94	8.552.882.729	TRUE
1748434770619,0	13.42	-8.63	1.2	4.497.184.317	TRUE
1748434770719,0	11.2	-7.16	1.41	2.563.135.823	TRUE
1748434770819,0	9.69	-6.03	1.51	1.687.062.487	TRUE
1748434786019,0	14.55	-10.42	0.78	6.306.362.968	TRUE

APPENDIX C: MAHALONOBIS RESULT OF GPS SENSOR LOGGER DATA

Presents the 3D Mahalanobis outlier-detection procedure on raw GPS logs, reporting the chosen threshold(s), percentage of points flagged, and before/after trajectory plots used for pre-fusion cleaning.

Table C.1. Data sources.

Timestamp	Latitude	Longitude	Altitude
45805,57	41,08	29,05	102,92
45805,57	41,08	29,05	102,89
45805,57	41,08	29,05	102,89
45805,57	41,08	29,05	102,90
45805,57	41,08	29,05	102,90
45805,57	41,08	29,05	102,90
45805,57	41,08	29,05	102,90
45805,57	41,08	29,05	102,90
45805,57	41,08	29,05	102,91
45805,57	41,08	29,05	102,91
45805,57	41,08	29,05	102,91
45805,57	41,08	29,05	102,92
45805,57	41,08	29,05	102,93
45805,57	41,08	29,05	102,93
45805,57	41,08	29,05	102,95
45805,57	41,08	29,05	102,96
45805,57	41,08	29,05	102,97
45805,57	41,08	29,05	102,98
45805,57	41,08	29,05	103,96
45805,57	41,08	29,05	103,59
45805,57	41,08	29,05	103,18

Table C.1 Data sources. (cont.)

Timestamp	Latitude	Longitude	Altitude
45805,57	41,08	29,05	103,27
45805,57	41,08	29,05	103,62
45805,57	41,08	29,05	103,98
45805,57	41,08	29,05	104,13
45805,57	41,08	29,05	104,55
45805,57	41,08	29,05	104,25
45805,57	41,08	29,05	104,06
45805,57	41,08	29,05	103,89
45805,57	41,08	29,05	103,28
45805,57	41,08	29,05	103,29
45805,57	41,08	29,05	103,20
45805,57	41,08	29,05	102,89
45805,57	41,08	29,05	103,05
45805,57	41,08	29,05	103,32
45805,57	41,08	29,05	103,32
45805,57	41,08	29,05	103,41
45805,57	41,08	29,05	103,38
45805,57	41,08	29,05	103,23
45805,57	41,08	29,05	103,01
45805,57	41,08	29,05	103,11
45805,57	41,08	29,05	102,82
45805,57	41,08	29,05	102,65
45805,57	41,08	29,05	102,77
45805,57	41,08	29,05	102,81
45805,57	41,08	29,05	103,01
45805,57	41,08	29,05	103,15
45805,57	41,08	29,05	103,17
45805,57	41,08	29,05	103,25
45805,57	41,08	29,05	103,19

Table C.1 Data sources. (cont.)

Timestamp	Latitude	Longitude	Altitude
45805,57	41,08	29,05	103,22
45805,57	41,08	29,05	103,28
45805,57	41,08	29,05	103,09
45805,57	41,08	29,05	103,03
45805,57	41,08	29,05	103,12
45805,57	41,08	29,05	103,21
45805,57	41,08	29,05	103,22
45805,57	41,08	29,05	103,13
45805,57	41,08	29,05	103,13
45805,57	41,08	29,05	103,11
45805,57	41,08	29,05	103,26
45805,57	41,08	29,05	103,30
45805,57	41,08	29,05	103,53
45805,57	41,08	29,05	103,10
45805,57	41,08	29,05	103,07
45805,57	41,08	29,05	103,04
45805,57	41,08	29,05	103,05
45805,57	41,08	29,05	102,86
45805,57	41,08	29,05	103,00
45805,57	41,08	29,05	103,36
45805,57	41,08	29,05	103,21
45805,57	41,08	29,05	103,38
45805,57	41,08	29,05	103,17
45805,57	41,08	29,05	102,99
45805,57	41,08	29,05	102,76
45805,57	41,08	29,05	102,55
45805,57	41,08	29,05	102,40
45805,57	41,08	29,05	102,36
45805,57	41,08	29,05	102,85
45805,57	41,08	29,05	102,91
45805,57	41,08	29,05	103,02

Table C.1 Data sources. (cont.)

45805,57	41,08	29,05	102,94
45805,57	41,08	29,05	102,93
45805,57	41,08	29,05	102,89
45805,57	41,08	29,05	103,14
45805,57	41,08	29,05	103,48
45805,57	41,08	29,05	103,16
45805,57	41,08	29,05	102,95
45805,57	41,08	29,05	103,15
45805,57	41,08	29,05	103,19
45805,57	41,08	29,05	103,22
45805,57	41,08	29,05	103,37
45805,58	41,08	29,05	103,32
45805,58	41,08	29,05	103,28
45805,58	41,08	29,05	103,45
45805,58	41,08	29,05	103,47
45805,58	41,08	29,05	103,45
45805,58	41,08	29,05	103,46
45805,58	41,08	29,05	103,58
45805,58	41,08	29,05	103,69
45805,58	41,08	29,05	103,71
45805,58	41,08	29,05	103,83
45805,58	41,08	29,05	103,81
45805,58	41,08	29,05	103,78
45805,58	41,08	29,05	103,73
45805,58	41,08	29,05	103,67
45805,58	41,08	29,05	103,48
45805,58	41,08	29,05	103,76
45805,58	41,08	29,05	103,82
45805,58	41,08	29,05	103,79
45805,58	41,08	29,05	103,70

Table C.1 Data sources. (cont.)

Timestamp	Latitude	Longitude	Altitude
45805,58	41,08	29,05	103,58
45805,58	41,08	29,05	103,49
45805,58	41,08	29,05	103,49
45805,58	41,08	29,05	103,40
45805,58	41,08	29,05	103,09
45805,58	41,08	29,05	103,10
45805,58	41,08	29,05	103,13
45805,58	41,08	29,05	102,82
45805,58	41,08	29,05	102,56
45805,58	41,08	29,05	102,68
45805,58	41,08	29,05	102,99
45805,58	41,08	29,05	103,10
45805,58	41,08	29,05	103,31
45805,58	41,08	29,05	103,22

APPENDIX D: UWB SENSOR LOCATION DATA

COORDINATE TRANSFORMATION

Documents anchor/tag coordinates and the transformation from local UWB frames to global references (e.g., ENU/UTM), including transformation matrices, residual errors, and alignment accuracy.

```
import numpy as np
from pyproj import Transformer

def geodetic_to_ecef(lat, lon, h):
    """
    Convert geodetic coordinates (deg, deg, meters) to ECEF XYZ (meters).
    Uses WGS84 by default. Returns (X,Y,Z).
    """
    # pyproj expects lon, lat order

    transformer = Transformer.from_crs("EPSG:4979", "EPSG:4978", always_xy=True)
    X, Y, Z = transformer.transform(lon, lat, h)
    return X, Y, Z

def ecef_to_enu(X, Y, Z, lat0, lon0, h0):
    """
    Convert ECEF XYZ to local ENU coordinates (east,north,up) relative to origin at
    (lat0,lon0,h0).
    lat0, lon0 in degrees; (X,Y,Z) and output ENU in same units as input.
    """
    # Reference origin in ECEF
    X0, Y0, Z0 = geodetic_to_ecef(lat0, lon0, h0)
    # Convert degrees to radians
```

```

phi0 = np.deg2rad(lat0); lam0 = np.deg2rad(lon0)
# Differences
dX, dY, dZ = X-X0, Y-Y0, Z-Z0
# ENU axes
sin_lat = np.sin(phi0); cos_lat = np.cos(phi0)
sin_lon = np.sin(lam0); cos_lon = np.cos(lam0)
east = -sin_lon * dX + cos_lon * dY
north = -sin_lat*cos_lon * dX - sin_lat*sin_lon * dY + cos_lat * dZ
up = cos_lat*cos_lon * dX + cos_lat*sin_lon * dY + sin_lat * dZ
return east, north, up

# Example usage:
lat0, lon0, h0 = gps_df['Latitude'][0], gps_df['Longitude'][0], gps_df['Altitude'][0]
# Convert all GPS points to ECEF
gps_ecef = np.array([geodetic_to_ecef(lat, lon, h)
for lat, lon, h in zip(gps_df['Latitude'], gps_df['Longitude'], gps_df['Altitude'])])
# Then to ENU
gps_enu = np.array([ecef_to_enu(X, Y, Z, lat0, lon0, h0)
for X, Y, Z in gps_ecef])

def align_points(P, Q, with_scale=True):
    """
    Compute similarity transform (R, t, s) aligning point set P to Q.
    P,Q are (N×D) arrays of corresponding points. Returns (R, t, s) such that
     $Q \approx s * P @ R.T + t$ .
    """
    P = np.asarray(P); Q = np.asarray(Q)
    assert P.shape == Q.shape, "Point sets must correspond in size"
    N, D = P.shape
    mean_P = P.mean(axis=0)
    mean_Q = Q.mean(axis=0)

```

```

P0 = P - mean_P
Q0 = Q - mean_Q
# Covariance
H = P0.T @ Q0
U, S, Vt = np.linalg.svd(H)
R = Vt.T @ U.T
# Ensure a proper rotation (det=+1)
if np.linalg.det(R) < 0:
    Vt[-1,:] *= -1
    R = Vt.T @ U.T
# Scale factor
if with_scale:
    varP = (P0**2).sum()
    s = np.trace(np.diag(S)) / varP
else:
    s = 1.0
t = mean_Q - s*(mean_P @ R)
return R, t, s

# Example usage:
# Suppose uwb_pts and enu_pts are arrays of matching reference points (N×3)
# e.g. from known anchors or synchronized readings
R, t, s = align_points(uwb_pts, enu_pts, with_scale=True)
uwb_aligned = s * (uwb_pts @ R.T) +

```


APPENDIX E: PYREVIT PYTHON CODINGS

Provides the pyRevit Python scripts for data import, filtering, BIM coordinate mapping, geofencing, and visualization, with brief usage notes and dependencies.

```
# -*- coding: utf-8 -*-
```

```
“ ” “ ”
```

Hybrid GPS–UWB fusion pipeline:

- 1) Geodetic (ITRF20) -> ECEF -> ENU
- 2) UWB local XYZ -> ENU via Procrustes/Helmert (R, t, s)
- 3) Time alignment + interpolation
(handles 10 Hz vs 1 Hz or vice versa)
- 4) Mahalanobis outlier removal (3D)
- 5) Unscented Kalman Filter (constant-velocity) fusion

No external deps beyond numpy/pandas.

(matplotlib optional)

```
“ ” “ ”
```

```
from __future__ import annotations
```

```
import math
```

```
import numpy as np
```

```
import pandas as pd
```

```
from dataclasses import dataclass
```

```
def to_datetime(s):
```

```
“ ” “ ”
```

```
- -
```

Robust timestamp parser; accepts ISO, epoch seconds/ms.

```
""" """
```

```
if np.issubdtype(pd.Series(s).dtype, np.number):
    # Heuristic: treat values > 1e12 as ms,
    >1e9 as s; else seconds.
    vals = pd.Series(s).astype(float)
    if (vals > 1e12).any():
        return pd.to_datetime(vals, unit="ms", utc=True)
    elif (vals > 1e9).any():
        return pd.to_datetime(vals, unit="s", utc=True)
    else:
        return pd.to_datetime(vals, unit="s", utc=True)
    return pd.to_datetime(s, utc=True)
```

```
def _safe_cholesky(M, jitter=1e-9, max_tries=8):
```

```
    " " "
```

Cholesky with jitter for numerical robustness."""

```
A = M.copy()
for _ in range(max_tries):
    try:
        return np.linalg.cholesky(A)
    except np.linalg.LinAlgError:
        A = A + np.eye(A.shape[0]) * jitter
        jitter *= 10.0
    raise np.linalg.LinAlgError
    ("Cholesky failed; matrix not PD.")
```

```
# 1) Geodesy: ITRF20(WGS/GRS) -> ECEF -> ENU
```

```
@dataclass
```

```

class Ellipsoid:
    a: float = 6378137.0
    # semi-major axis [m]
    f: float = 1.0/298.257223563
    # WGS-84 flattening (GRS80: 1/298.257222101)
    def e2(self) -> float:
        return self.f * (2.0 - self.f)
    WGS84 = Ellipsoid() # good for ITRF/GRS as well
    (differences are negligible for this use)

    def geodetic_to_ecef(lat_deg: float,
        lon_deg: float, h_m: float,
        ell: Ellipsoid = WGS84) -> np.ndarray:
        """(lat,lon[h/m]) -> (X,Y,Z) in meters (ECEF).
        """
        lat = math.radians(lat_deg)
        lon = math.radians(lon_deg)
        a, e2 = ell.a, ell.e2()
        sinp, cosp = math.sin(lat), math.cos(lat)
        sinl, cosl = math.sin(lon), math.cos(lon)
        N = a / math.sqrt(1.0 - e2 * sinp*sinp)
        X = (N + h_m) * cosp * cosl
        Y = (N + h_m) * cosp * sinl
        Z = (N * (1.0 - e2) + h_m) * sinp
        return np.array([X, Y, Z], dtype=float)

    def ecef_to_enu(XYZ: np.ndarray,
        lat0_deg: float, lon0_deg: float, h0_m: float,
        ell: Ellipsoid = WGS84) -> np.ndarray:
        """ECEF -> ENU at local origin (lat0,lon0,h0).
        """

```

```

X0, Y0, Z0 = geodetic_to_ecef(lat0_deg,
lon0_deg, h0_m, ell)
d = XYZ - np.array([X0, Y0, Z0], dtype=float)

```

```

lat0 = math.radians(lat0_deg)
lon0 = math.radians(lon0_deg)
sl, cl = math.sin(lat0), math.cos(lat0)
sb, cb = math.sin(lon0), math.cos(lon0)

```

```

R = np.array([
[-sb, cb, 0.0],
[-sl*cb, -sl*sb, cl ],
[ cl*cb, cl*sb, sl ]
], dtype=float)
return (R @ d.reshape(3,1)).ravel()

```

```

def gps_geodetic_to_enu(gps_df: pd.DataFrame,
lat_col="lat", lon_col="lon", h_col="h",
t_col="timestamp",
origin=("auto", "auto", "auto"),
ell: Ellipsoid = WGS84) -> pd.DataFrame:
"""

```

```

Convert a GPS dataframe in geodetic
(ITRF20) to ENU relative to origin.
origin: (lat0, lon0, h0) or
("auto", "auto", "auto") to use the first row.
"""

```

```

df = gps_df.copy()
df[t_col] = _to_datetime(df[t_col])
df = df.sort_values(t_col)

```

```

if origin == ("auto", "auto", "auto"):
    lat0, lon0, h0 = float(df.iloc[0][lat_col]), float(df.iloc[0][lon_col]),
    float(df.iloc[0][h_col])
else:
    lat0, lon0, h0 = origin

```

```

# Vectorized conversion
ecef = np.vstack([
    geodetic_to_ecef(row[lat_col],
    row[lon_col], row[h_col], ell)
    for _, row in df.iterrows()
])
enu = np.vstack([
    ecef_to_enu(ecef[i], lat0, lon0, h0, ell)
    for i in range(ecef.shape[0])
])

```

```

out = pd.DataFrame(
    t_col: df[t_col].values,
    "gps_e": enu[:,0], "gps_n": enu[:,1],
    "gps_u": enu[:,2]
).set_index(t_col)
return out

```

```

# 2) Similarity (Procrustes/Helmert) for UWB->ENU

```

```

@dataclass
class Similarity:
    R: np.ndarray # 3x3
    t: np.ndarray # 3,

```

```

s: float # scalar

def procrustes_umeyama(P: np.ndarray,
Q: np.ndarray, with_scale=True)
-> Similarity:
"""
Find similarity transform s,R,t aligning P
(Nx3) to Q (Nx3):  $Q \approx s * R P^T + t$ 
Returns (R, t, s). Ensures  $\det(R)=+1$  (proper rotation).
"""
P = np.asarray(P, float)
Q = np.asarray(Q, float)
assert P.shape == Q.shape and P.shape[1] == 3
muP = P.mean(axis=0)
muQ = Q.mean(axis=0)
P0 = P - muP
Q0 = Q - muQ
H = P0.T @ Q0 / P.shape[0]
U, S, Vt = np.linalg.svd(H)
R = Vt.T @ U.T
if np.linalg.det(R) < 0:
    Vt[-1,:] *= -1
R = Vt.T @ U.T
if with_scale:
    varP = (P0**2).sum() / P.shape[0]
    s = np.trace(np.diag(S)) / varP
else:
    s = 1.0
t = muQ - s * (R @ muP)
return Similarity(R=R, t=t, s=s)

```

```
def apply_similarity(X: np.ndarray, sim: Similarity)
-> np.ndarray:
return (sim.s * (sim.R @ X.T)).T + sim.t
```

```
# 3) Time sync and interpolation
```

```
def estimate_rate_hz(times: pd.DatetimeIndex)
-> float:
    """
    Median sampling rate in Hz."""
    if len(times) < 2:
        return 1.0
    dt = np.median(np.diff(times.view(np.int64))) # ns
    if dt <= 0:
        return 1.0
    hz = 1e9 / dt
    return float(hz)
```

```
def resample_to_index(df_xyz: pd.DataFrame,
target_index: pd.DatetimeIndex,
cols=("e","n","u")) -> pd.DataFrame:
    """
```

```
    Nearest then linear interpolation to ensure
    continuous series on target_index.
```

```
    """
```

```
    tmp = df_xyz.copy().reindex
    (target_index, method="nearest")
    # linear pass to smooth within small gaps
    tmp = tmp.interpolate(method="time",
```

```
limit_direction="both")
```

```
return tmp
```

```
def build_common_index(gps_idx, uwb_idx,
```

```
target_hz=None) -> pd.DatetimeIndex:
```

```
“ ” “ ”
```

Common timeline over intersecting time

range at max(gps_hz, uwb_hz) or target_hz.

```
“ ” “ ”
```

```
start = max(gps_idx.min(), uwb_idx.min())
```

```
end = min(gps_idx.max(), uwb_idx.max())
```

```
if end <= start:
```

```
raise ValueError("No temporal overlap between
```

```
GPS and UWB streams.")
```

```
gps_hz = estimate_rate_hz(gps_idx)
```

```
uwb_hz = estimate_rate_hz(uwb_idx)
```

```
hz = target_hz if target_hz else max(gps_hz, uwb_hz)
```

```
step_ms = int(round(1000.0 / hz))
```

```
return pd.date_range(start=start, end=end,
```

```
freq=f"step_msms", tz="UTC")
```

4) Mahalanobis outlier removal

```
def mahalanobis_mask(X: np.ndarray, alpha:
```

```
float = 0.99) -> np.ndarray:
```

```
“ ” “ ”
```

Return boolean mask of inliers using squared

Mahalanobis distance vs χ^2 threshold.

DOF=3 thresholds: 0.95->7.815,

0.975->9.348, 0.99->11.345


```
“ ” “ ”
```

```
thr = 0.95: 7.815, 0.975: 9.348, 0.99: 11.345.
```

```
get(alpha, 11.345)
```

```
X = np.asarray(X, float)
```

```
mu = X.mean(axis=0)
```

```
C = np.cov(X, rowvar=False)
```

```
C += np.eye(3)*1e-9 # regularize
```

```
Ci = np.linalg.inv(C)
```

```
d = X - mu
```

```
md2 = np.sum(d @ Ci * d, axis=1) # squared MD
```

```
return md2 <= thr
```

```
def clean_xyz_df(df: pd.DataFrame, cols=("e", "n", "u"),
alpha=0.99) -> pd.DataFrame:
```

```
m = mahalanobis_mask(df.loc[:, cols].values, alpha)
```

```
return df.loc[m]
```

```
# 5) Unscented Kalman Filter (CV model, 6D state)
```

```
@dataclass
```

```
class UKFParams:
```

```
alpha: float = 1e-3
```

```
beta: float = 2.0
```

```
kappa: float = 0.0
```

```
class UKF6CV:
```

```
“ ” “ ”
```

```
6D state: [x,y,z,vx,vy,vz]
```

```
Process: constant velocity, additive Q
```

```
Measurement: position only ([x,y,z])
```

“ ” “ ”

```
def __init__(self, Q_diag=(0.02,0.02,0.02, 0.1,0.1,0.1),
R_gps_diag=(1.5,1.5,2.5), R_uwb_diag=(0.3,0.3,0.5),
params: UKFParams = UKFParams()):
    self.n = 6
    self.x = np.zeros(6, dtype=float)
    self.P = np.eye(6)*1.0
    self.Q = np.diag(Q_diag)
    self.R_gps = np.diag(R_gps_diag)
    self.R_uwb = np.diag(R_uwb_diag)
    self.prm = params
    a, b, k = self.prm.alpha, self.prm.beta, self.prm.kappa
    self.lam = a*a*(self.n + k) - self.n
    self.Wm = np.full(2*self.n+1, 1.0/(2*(self.n + self.lam)))
    self.Wc = self.Wm.copy()
    self.Wm[0] = self.lam/(self.n + self.lam)
    self.Wc[0] = self.lam/(self.n + self.lam) + (1 - a*a + b)
```

```
def _sigma_points(self, x, P):
    S = _safe_cholesky((self.n + self.lam) * P)
    X = np.zeros((2*self.n+1, self.n))
    X[0] = x
    for i in range(self.n):
        X[i+1] = x + S[i]
        X[i+1+self.n] = x - S[i]
    return X
```

```
def _predict_f(self, X, dt):
    Y = X.copy()
    Y[:,0:3] = X[:,0:3] + X[:,3:6]*dt
    # velocities unchanged in CV model
```

```
return Y
```

```
def predict(self, dt):
```

```
    Xsig = self._sigma_points(self.x, self.P)
```

```
    Xpred = self._predict_f(Xsig, dt)
```

```
    x_pred = (self.Wm[:,None] * Xpred).sum(axis=0)
```

```
    P_pred = self.Q.copy()
```

```
    for i in range(Xpred.shape[0]):
```

```
        dx = (Xpred[i]-x_pred).reshape(-1,1)
```

```
        P_pred += self.Wc[i] * (dx @ dx.T)
```

```
    self.x, self.P = x_pred, P_pred
```

```
def _update_generic(self, z, R):
```

```
    # Build sigma points around current state
```

```
    Xsig = self._sigma_points(self.x, self.P)
```

```
    # Measurement: position only
```

```
    Zsig = Xsig[:, 0:3]
```

```
    z_pred = (self.Wm[:,None] * Zsig).sum(axis=0)
```

```
    Pz = R.copy()
```

```
    for i in range(Zsig.shape[0]):
```

```
        dz = (Zsig[i]-z_pred).reshape(-1,1)
```

```
        Pz += self.Wc[i]*(dz @ dz.T)
```

```
    Pxz = np.zeros((self.n,3))
```

```
    for i in range(Zsig.shape[0]):
```

```
        dx = (Xsig[i]-self.x).reshape(-1,1)
```

```
        dz = (Zsig[i]-z_pred).reshape(-1,1)
```

```
        Pxz += self.Wc[i]*(dx @ dz.T)
```

```
    K = Pxz @ np.linalg.inv(Pz)
```

```
    self.x = self.x + (K @ (z - z_pred))
```

```
    self.P = self.P - K @ Pz @ K.T
```

```
def update_gps(self, zgps):
self._update_generic(np.asarray(zgps, float),
self.R_gps)
```

```
def update_uwb(self, zuwb):
self._update_generic(np.asarray(zuwb, float),
self.R_uwb)
```

```
# 6) End-to-end pipeline
```

```
@dataclass
class ColumnMapGPS:
t: str = "timestamp"
lat: str = "lat"
lon: str = "lon"
h: str = "h"
```

```
@dataclass
class ColumnMapUWB:
t: str = "timestamp"
```

```
x: str = "x"
y: str = "y"
z: str = "z"
```

```
def estimate_similarity_from_overlap(gps_enu:
pd.DataFrame, uwb_local: pd.DataFrame)
->
```

```
Similarity:
```

```
“ ” “ ”
```

Use overlapping samples (after rough nearest reindex)
to estimate UWB->ENU similarity.

```
“ ” “ ”
```

```
common = gps_enu.join(uwb_local,
how="inner", lsuffix="", rsuffix="_uwb")
# Remove NA rows
common = common.dropna()
if len(common) < 6:
raise ValueError("Not enough overlap to
estimate similarity (need >=6 points).")

# Robust pre-clean: Mahalanobis on GPS ENU
to avoid degenerate alignment
gps_pts = common[["gps_e", "gps_n", "gps_u"]].
to_numpy()
uwb_pts = common[["uwb_x", "uwb_y", "uwb_z"]].
to_numpy() if
"uwb_x", "uwb_y", "uwb_z".issubset(common.columns)
else common[["x", "y", "z"]].to_numpy()

# If UWB columns came as x,y,z already:
if "uwb_x" not in common.columns:
common = common.rename(columns=
"x":"uwb_x", "y":"uwb_y", "z":"uwb_z")

# A quick MD clean on both:
keep_gps = mahalanobis_mask(gps_pts, 0.99)
keep_uwb = mahalanobis_mask(uwb_pts, 0.99)
keep = keep_gps and keep_uwb
gps_pts = gps_pts[keep]
uwb_pts = uwb_pts[keep]
if len(gps_pts) < 6:
raise ValueError
```

```
("Too few clean pairs to estimate similarity.")
```

```
return procrustes_umeyama(uwb_pts, gps_pts, with_scale=True)
```

```
def run_pipeline(
    gps_df: pd.DataFrame, uwb_df: pd.DataFrame,
    gps_cols: ColumnMapGPS = ColumnMapGPS(),
    uwb_cols: ColumnMapUWB = ColumnMapUWB(),
    origin=("auto", "auto", "auto"),
    target_hz: float — None = None,
    md_alpha: float = 0.99,
    ukf.Q=(0.02,0.02,0.02, 0.2,0.2,0.2),
    R_gps=(2.0,2.0,3.0),
    R_uwb=(0.3,0.3,0.6)
) -> pd.DataFrame:
    """
```

Returns a dataframe indexed by time with columns:

```
'gps_e', 'gps_n', 'gps_u', 'uwb_e', 'uwb_n',
'uwb_u', 'fused_e', 'fused_n', 'fused_u'
“ ” “ ”
```

```
# — GPS -> ENU
```

```
gps = gps_geodetic_to_enu(gps_df.
    rename(columns=gps_cols.t:"timestamp",
    gps_cols.lat:"lat",
    gps_cols.lon:"lon",
    gps_cols.h:"h"),
    origin=origin)
gps.columns = ["gps_e", "gps_n", "gps_u"]
```

```
# — UWB: parse timestamps and keep local XYZ
```

```
uwb = uwb_df.copy()
```

```

uwb["timestamp"] = _to_datetime(uwb[uwb_cols.t])
uwb = uwb.sort_values("timestamp").set_index("timestamp")
uwb_local = uwb.rename
(columns=uwb_cols.x:"x", uwb_cols.y:"y",
uwb_cols.z:"z")[[ "x", "y", "z" ]]

# — Common index
(fastest stream or given target_hz)
common_index =
build_common_index(gps.index, uwb_local.index, target_hz=target_hz)

# — Rough resample
(for overlap to estimate similarity)
gps_r = resample_to_index(gps, common_index, cols=("gps_e", "gps_n", "gps_u"))
uwb_r_local =
resample_to_index(uwb_local,
common_index, cols=("x", "y", "z"))

# — Estimate similarity transform
UWB_local -> ENU using overlap pairs
sim = estimate_similarity_from_overlap(gps_r, uwb_r_local)

# — Apply similarity to all UWB local points
uwb_aligned_full = pd.DataFrame(index=uwb_local.index,
columns=["uwb_e", "uwb_n", "uwb_u"], dtype=float)
uwb_aligned_full.loc[:, ["uwb_e", "uwb_n", "uwb_u"]] =
apply_similarity(uwb_local.to_numpy(), sim)

# — Resample both to the common index
(now in same ENU frame)
gps_rs = resample_to_index(gps, common_index, cols=("gps_e", "gps_n", "gps_u"))

```

```

uwb_rs = resample_to_index(uwb_aligned_full,
common_index,
cols=("uwb_e","uwb_n","uwb_u"))

# — Outlier removal (Mahalanobis)
on the resampled streams
gps_clean = clean_xyz_df(gps_rs.rename(columns=
"gps_e":"e","gps_n":"n","gps_u":"u"),
alpha=md_alpha) uwb_clean = clean_xyz_df(uwb_rs.rename(columns=
"uwb_e":"e","uwb_n":"n","uwb_u":"u"),
alpha=md_alpha)
# Re-join (missing rows -> NaN; UKF can
handle by skipping updates)
all_idx = common_index
gps_c = gps_clean.reindex(all_idx)
uwb_c = uwb_clean.reindex(all_idx)

# — UKF fusion
dt = (all_idx[1] - all_idx[0]).total_seconds()
ukf = UKF6CV(Q_diag=ukf_Q, R_gps_diag=R_gps, R_uwb_diag=R_uwb)

# Initialize at first available measurement
(prefer UWB, else GPS)
i0 = 0
while i0 < len(all_idx) and
pd.isna(uwb_c.iloc[i0][["e","n","u"]]).any() and
pd.isna(gps_c.iloc[i0][["e","n","u"]]).any():
i0 += 1
if i0 == len(all_idx):
raise ValueError
("No usable measurements to initialize the filter.")

```



```

if not pd.isna(uwb_c.iloc[i0][["e","n","u"]]).any():
    ukf.x[0:3] = uwb_c.iloc[i0][["e","n","u"]].to_numpy()
else:
    ukf.x[0:3] = gps_c.iloc[i0][["e","n","u"]].to_numpy()

fused_e, fused_n, fused_u = [], [], []
last_t = all_idx[0]

for t in all_idx:
    dt = (t - last_t).total_seconds() if t != last_t else dt
    last_t = t
    ukf.predict(dt)

    # GPS update if present
    grow = gps_c.loc[t]
    if not pd.isna(grow[["e","n","u"]]).any():
        ukf.update_gps(grow[["e","n","u"]].to_numpy())

    # UWB update if present
    urow = uwb_c.loc[t]
    if not pd.isna(urow[["e","n","u"]]).any():
        ukf.update_uwb(urow[["e","n","u"]].to_numpy())

    fused_e.append(ukf.x[0]); fused_n.append(ukf.x[1]); fused_u.append(ukf.x[2])

# — Output frame
out = pd.DataFrame(index=all_idx, data=
    "gps_e": gps_rs["gps_e"],
    "gps_n": gps_rs["gps_n"],
    "gps_u": gps_rs["gps_u"],
    "uwb_e": uwb_rs["uwb_e"],

```

```

"uwb_n": uwb_rs["uwb_n"],
"uwb_u": uwb_rs["uwb_u"],
"fused_e": fused_e, "fused_n":
fused_n, "fused_u": fused_u )
out.attrs["similarity"] = sim
# keep R,t,s for reporting
return out

```

```

# 7) Example usage (pseudo)

```

```

if name == " main ":
# Example column names – adjust
to your CSV/Excel schema.
# GPS: columns ['timestamp','lat','lon','h']
# UWB: columns ['timestamp','x','y','z']
import os

gps path = "Cleaned GPS Data.xlsx"
# or CSV
uwb path = "cleaned location data no outliers.csv"

# Load data
if gps path.lower().endswith(".xlsx"):
gps df = pd.read_excel(gps path)
else:
gps df = pd.read_csv(gps path)

uwb df = pd.read_csv(uwb path)

# Run pipeline (auto-origin = first GPS fix)

```

```

fused = run_pipeline(gps_df, uwb_df,
gps_cols=ColumnMapGPS(t="timestamp",
lat="lat", lon="lon", h="h"),
uwb_cols=ColumnMapUWB(t="timestamp",
x="x", y="y", z="z"),
origin=("auto", "auto", "auto"),
target_hz=None, # None -> fastest stream
md.alpha=0.99,
ukf.Q=(0.02,0.02,0.02, 0.3,0.3,0.3),
R_gps=(3.0,3.0,5.0),
R_uwb=(0.25,0.25,0.5))

fused.to_csv("fused_enu.csv", index=True)
print("Saved fused trajectory -> fused_enu.csv")
sim = fused.attrs["similarity"]
print("UWB->ENU similarity:
n scale:", sim.s, "n R:n", sim.R, "n t:", sim.t)

```

PyRevit plug in codings for generic component:

```

import clr, os, csv, math
clr.AddReference('RevitAPI')
clr.AddReference('RevitServices')
from Autodesk.Revit.DB import *
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
from System.Collections.Generic import List
from pyrevit import forms, script

doc = DocumentManager.Instance.CurrentDBDocument

```

```

uidoc = DocumentManager.Instance.CurrentUIApplication.
ActiveUIDocument
logger = script.get logger()

```

```

M TO FT = 3.280839895
CSV FILTER = 'CSV Files—*.csv'
DEFAULT MARK EVERY = 10
# Hēr 10 noktada bir işaret koy
APPLY ROT DEG = 0.0
# ENU
- -
ORIGIN OFFSET ENU M = (0.0, 0.0, 0.0)
# ENU ofset (m)
LINE SUBSAMPLE = 1
# 1 = ;
2 = . -

```

```

def pick csv(): path = forms.pick file(file filter=CSV FILTER,
multi file=False, title='Pick
fused enu.csv')
if not path:
- - -
forms.alert('No file selected.', exitscript=True)
-
return path
-

```

```

def enu to revit xyz(e m, n m,)
- - - - -
u_m, rot_deg=0.0, offset_enu_m=(0,0,0)):
“ ” “ ”

```

```

ENU (m) -> [ops rotasyon + ofset] -> feet, XYZ."""
# Yatay rotasyon (Up eksenini etrafında)
th = math.radians(rot_deg)
e2 = e_m*math.cos(th) - n_m*math.sin(th)
n2 = e_m*math.sin(th) + n_m*math.cos(th)
u2 = u_m
e2 += offset_enu_m[0]
n2 += offset_enu_m[1]
u2 += offset_enu_m[2]
return XYZ(e2*M.TO_FT, n2*M.TO_FT, u2*M.TO_FT)

def read_fused_csv(path):
    pts = [] # (XYZ, timestamp_str)
    with open(path, 'r') as f:
        rdr = csv.DictReader(f)
        need = 'fused_e','fused_n','fused_u'
        if not need.issubset(set([c.strip()
            for c in rdr.fieldnames])):
            raise Exception('CSV must contain columns: fused_e,fused_n,fused_u
            (+ timestamp optional)')
        for row in rdr:
            try:
                e = float(row['fused_e']); n =
                float(row['fused_n']);
                u = float(row['fused_u'])
            except:
                continue
            ts = row.get('timestamp') or row.get('time') or ''
            xyz = enu_to_revit_xyz(e, n, u,
            rot_deg=APPLY_ROT_DEG,
            offset_enu_m=ORIGIN_OFFSET_ENU_M)

```

```

pts.append((xyz, ts))
if len(pts) < 2:
    raise Exception('Not enough points in CSV.')
return pts

def get_generic_category_id():
    return Category.GetCategory(doc, BuiltInCategory.OST_GenericModel).Id

def ensure_material(name, color):
    """
    Create or fetch a material."""
    mats = FilteredElementCollector(doc).
    OfClass(Material).ToElements()
    for m in mats:
        if m.Name == name:
            return m.Id
    # create
    t = Transaction(doc, "Create Material: " + name)
    t.Start()
    mat = Material.Create(doc, name)
    mat.Color = color
    t.Commit()
    return mat.Id

def create_polyline_directshape(xyz_list,
    name="RTLS_Path"):
    """
    Create a 3D polyline DirectShape under Generic Model.
    """
    cat_id = get_generic_category_id()
    ds = DirectShape.CreateElement(doc, cat_id)

```

```

ds.ApplicationId = "RTLS"
ds.ApplicationDataId = name

# PolyLine requires IList[XYZ]
pts_ilst = List[XYZ]([xyz for xyz in xyz_list])
poly = PolyLine.Create(pts_ilst)

t = Transaction(doc, "Create RTLS Polyline")
t.Start()
ds.SetShape(List[GeometryObject]([poly]))
ds.Name = name
t.Commit()
return ds

def create_point_marker(xyz, name=
"RTLS_Point", color=Color(0, 180, 255)):
    " " " "

    Create a tiny cube DirectShape as a point marker.
    " " " "

    cat_id = get_generic_category_id()
    ds = DirectShape.CreateElement(doc, cat_id)
    ds.ApplicationId = "RTLS"
    ds.ApplicationDataId = name
    sz = 0.05 * M TO FT

    def box_edges(center, half):
    x, y, z = center.X, center.Y, center.Z

```

```

hx = hy = hz = half
corners = [
XYZ(x-hx,y-hy,z-hz), XYZ(x+hx,y-hy,z-hz),
XYZ(x+hx,y+hy,z-hz), XYZ(x-hx,y+hy,z-hz),
XYZ(x-hx,y-hy,z+hz), XYZ(x+hx,y-hy,z+hz),
XYZ(x+hx,y+hy,z+hz), XYZ(x-hx,y+hy,z+hz), ]
quads = [
    [0, 1, 2, 3, 0], [4, 5, 6, 7, 4],
    [0, 1, 5, 4, 0], [1, 2, 6, 5, 1],
    [2, 3, 7, 6, 2], [3, 0, 4, 7, 3]
]
lines = []
for q in quads:
    pts = List[XYZ]([corners[i] for i in q])
    lines.append(PolyLine.Create(pts))
return lines

material_id = ensure_material
("RTLS.Point_Mat", color)

```

```

t = Transaction(doc, "Create RTLS Point")
t.Start()
shape = List[GeometryObject]()
for pl in box.edges(p, sz/2.0):
    shape.Add(pl)
ds.SetShape(shape)
ds.Name = name
ds.SetMaterialId(material_id)
t.Commit()
return ds

```

```

def main():
    path = pick_csv()

```



```

pts_ts = read_fused_csv(path)

# subsample for polyline and markers
poly_pts = [pt for i,(pt,-)
in enumerate(pts_ts) if i% LINE_SUBSAMPLE == 0]
mark_pts = [pt for i,(pt,-)
in enumerate(pts_ts) if i% DEFAULT_MARK_EVERY == 0]

create_polyline_directshape(poly_pts,
name="RTLS_Path")
for i, p in enumerate(mark_pts):
create_point_marker(p, name="RTLS_P%04d"% i)

forms.alert('Imported %d points. Polyline
and %d markers created.' %
(len(pts_ts), len(mark_pts)), ok=True)

if __name__ == "__main__":
main()

```

APPENDIX F: UNSCENTED KALMAN FILTER FUSION OF GPS AND UWB POSITION DATA

Details the UKF data (state vector, motion/measurement models, noise settings, and time-sync) and summarizes fusion outputs with representative error statistics and trajectory comparison.

Table F.1. Fused data sources.

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:29:50	0.914151	-311.995	2.251.354	-3,00E-12	2.9E-12
2025-05-28 15:29:51	1.717.392	-39.154	0.280899	0.328974	-0.32578
2025-05-28 15:29:52	1.861.174	-272.488	-0.3306	0.270846	0.150155
2025-05-28 15:29:53	0.931969	-0.10072	0.346704	-0.03083	0.772094
2025-05-28 15:29:54	1.577.453	2.152.215	0.545601	0.111273	1.083.247
2025-05-28 15:29:55	121.321	2.906.299	-0.88741	0.024822	1.023.403
2025-05-28 15:29:56	2.974.161	3.031.255	-106.904	0.305821	0.877986
2025-05-28 15:29:57	2.910.754	4.398.054	-114.765	0.25131	0.950152
2025-05-28 15:29:58	3.263.215	4.003.963	-0.55782	0.265223	0.765263
2025-05-28 15:29:59	4.375.642	4.397.868	-0.1819	0.375796	0.716795
2025-05-28 15:30:00	6.850.248	405.785	-0.65596	0.639841	0.583841
2025-05-28 15:30:01	6.595.115	4.626.891	0.281312	0.529898	0.582022
2025-05-28 15:30:02	7.038.606	3.994.285	-0.45197	0.519428	0.43484
2025-05-28 15:30:03	8.083.185	4.686.316	-0.02136	0.582669	0.465812
2025-05-28 15:30:04	8.084.656	4.912.011	0.032891	0.512753	0.436927
2025-05-28 15:30:05	8.819.948	5.586.576	0.162331	0.53958	0.465574
2025-05-28 15:30:06	9.864.247	5.601.232	0.324055	0.600652	0.411012
2025-05-28 15:30:07	1.080.473	4.545.524	0.031499	0.641952	0.232763
2025-05-28 15:30:08	1.090.506	4.246.879	-0.18452	0.57585	0.167908
2025-05-28 15:30:09	1.245.506	3.866.596	0.515221	0.695183	0.100754
2025-05-28 15:30:10	116.154	395.688	0.552243	0.506591	0.099468
2025-05-28 15:30:11	1.256.267	4.820.112	1.027.271	0.560865	0.193533

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:30:12	1.279.685	4.778.943	1.478.435	0.520563	0.164579
2025-05-28 15:30:13	1.318.413	4.206.225	0.451075	0.504103	0.073521
2025-05-28 15:30:14	1.307.965	5.007.396	0.458132	0.428892	0.163447
2025-05-28 15:30:15	1.420.657	5.090.138	0.476845	0.515186	0.15347
2025-05-28 15:30:16	1.521.278	5.284.079	0.704813	0.575896	0.158474
2025-05-28 15:30:17	152.407	5.452.761	0.307657	0.508147	0.159736
2025-05-28 15:30:18	1.495.484	6.238.615	-0.09786	0.409989	0.237138
2025-05-28 15:30:19	1.563.808	6.947.895	0.204691	0.443764	0.295497
2025-05-28 15:30:20	1.676.834	7.178.114	-0.14467	0.528605	0.28743
2025-05-28 15:30:21	1.728.833	6.276.486	-116.722	0.52754	0.140501
2025-05-28 15:30:22	1.692.278	6.016.948	-0.73185	0.417198	0.091076
2025-05-28 15:30:23	1.697.125	6.271.848	0.295359	0.371645	0.111315
2025-05-28 15:30:24	1.747.778	7.324.151	-0.39395	0.388307	0.227555
2025-05-28 15:30:25	1.809.862	7.074.499	-0.56871	0.41703	0.168609
2025-05-28 15:30:26	1.944.826	6.340.209	-0.16688	0.532227	0.057083
2025-05-28 15:30:27	2.025.304	6.547.255	-114.717	0.565892	0.075606
2025-05-28 15:30:28	2.088.736	6.816.268	-0.80049	0.574344	0.099495
2025-05-28 15:30:29	2.145.269	859.894	-0.82567	0.57323	0.307394
2025-05-28 15:30:30	2.125.267	9.163.327	-0.51655	0.477722	0.339136
2025-05-28 15:30:31	228.207	1.013.272	-0.1493	0.612392	0.416984
2025-05-28 15:30:32	2.430.196	9.550.013	-0.53876	0.719712	0.293505
2025-05-28 15:30:33	2.393.003	9.624.818	-139.861	0.584876	0.266491
2025-05-28 15:30:34	2.475.258	9.828.537	-220.255	0.614233	0.258738
2025-05-28 15:30:35	2.433.892	1.040.562	-124.817	0.487269	0.298059
2025-05-28 15:30:36	2.616.028	1.276.466	-0.71031	0.652054	0.552629
2025-05-28 15:30:37	2.566.635	1.123.824	-0.34758	0.510504	0.295827
2025-05-28 15:30:38	2.540.671	1.113.758	-0.66106	0.415376	0.246853
2025-05-28 15:30:39	2.571.147	121.182	-0.39226	0.401713	0.337488
2025-05-28 15:30:40	2.602.068	114.957	-145.992	0.390287	0.218911

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:30:41	2.611.503	1.167.016	0.054635	0.353733	0.21342
2025-05-28 15:30:42	2.669.359	1.257.826	-0.24188	0.381503	0.299226
2025-05-28 15:30:43	2.632.356	1.200.431	-0.66733	0.288674	0.191373
2025-05-28 15:30:44	2.841.163	1.162.975	0.067702	0.510935	0.121469
2025-05-28 15:30:45	2.812.806	1.257.499	0.377029	0.412798	0.223221
2025-05-28 15:30:46	2.846.047	1.273.082	-0.10255	0.402868	0.214897
2025-05-28 15:30:47	2.922.894	1.259.395	-0.93012	0.448027	0.171446
2025-05-28 15:30:48	2.872.576	1.294.347	0.339366	0.330536	0.193441
2025-05-28 15:30:49	2.936.403	1.306.405	0.519224	0.368546	0.184443
2025-05-28 15:30:50	3.140.518	1.356.502	-0.5513	0.575144	0.223539
2025-05-28 15:30:51	3.151.815	1.361.485	-0.12588	0.050348	0.0263
2025-05-28 15:30:52	311.525	1.344.399	-0.05055	-0.10921	-0.04932
2025-05-28 15:30:53	3.129.081	1.374.161	0.047495	-0.03508	0.054585
2025-05-28 15:30:54	3.180.747	1.397.181	0.105408	0.129289	0.106901
2025-05-28 15:30:55	3.234.321	1.444.879	0.0619	0.253798	0.220272
2025-05-28 15:30:56	3.287.192	141.777	-0.06567	0.339558	0.066989
2025-05-28 15:30:57	3.304.136	1.452.879	-0.13953	0.286026	0.156389
2025-05-28 15:30:58	3.334.107	1.502.131	-0.16706	0.290348	0.262557
2025-05-28 15:30:59	3.384.614	1.492.777	-0.11767	0.358271	0.149913
2025-05-28 15:31:00	3.431.699	1.509.547	0.078197	0.393903	0.155544
2025-05-28 15:31:01	3.493.766	1.528.322	0.088007	0.465691	0.16574
2025-05-28 15:31:02	3.502.659	1.546.424	0.090767	0.346406	0.170577
2025-05-28 15:31:03	3.559.484	1.561.178	0.097073	0.416645	0.163284
2025-05-28 15:31:04	3.603.852	1.588.298	-0.00015	0.425206	0.197453
2025-05-28 15:31:05	3.633.613	1.609.602	-0.44499	0.384808	0.202386
2025-05-28 15:31:06	367.863	1.554.251	-0.53023	0.4055	-0.03694
2025-05-28 15:31:07	3.724.116	1.614.318	-0.34453	0.421129	0.164937
2025-05-28 15:31:08	3.749.354	1.654.011	-0.31661	0.3677	0.238388
2025-05-28 15:31:09	3.834.584	1.654.036	-0.09939	0.521134	0.162991
2025-05-28 15:31:10	3.878.098	1.670.581	-0.29879	0.493907	0.163769

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:31:11	3.892.291	1.693.189	-0.39402	0.382463	0.183498
2025-05-28 15:31:12	3.926.516	1.698.665	0.108988	0.369731	0.142737
2025-05-28 15:31:13	3.972.518	1.708.618	-0.00161	0.398318	0.129058
2025-05-28 15:31:14	4.033.406	173.354	-0.20038	0.464987	0.167104
2025-05-28 15:31:15	4.060.387	1.765.252	-0.17441	0.403191	0.214599
2025-05-28 15:31:16	4.091.006	1.767.492	-0.02118	0.372479	0.153746
2025-05-28 15:31:17	4.130.761	1.781.926	0.057226	0.380415	0.15077
2025-05-28 15:31:18	4.208.248	1.806.265	-0.21434	0.505308	0.180093
2025-05-28 15:31:19	4.263.811	1.805.773	-0.1592	0.52124	0.121516
2025-05-28 15:31:20	431.728	184.434	0.208617	0.5255	0.20515
2025-05-28 15:31:21	4.385.361	1.847.575	0.123508	0.574673	0.150439
2025-05-28 15:31:22	4.419.021	1.850.925	0.096267	0.499294	0.113415
2025-05-28 15:31:23	4.468.208	1.866.473	-0.16488	0.496942	0.126733
2025-05-28 15:31:24	4.501.605	1.892.357	-0.09963	0.445344	0.168561
2025-05-28 15:31:25	4.552.445	191.047	-0.07902	0.46531	0.17254
2025-05-28 15:31:26	4.610.667	1.920.266	-0.20999	0.502325	0.148927
2025-05-28 15:31:27	4.676.543	1.926.821	-0.40443	0.551856	0.122529
2025-05-28 15:31:28	4.692.876	1.925.967	-0.07189	0.42884	0.081028
2025-05-28 15:31:29	4.745.437	1.938.811	0.125507	0.459478	0.09604
2025-05-28 15:31:30	481.693	1.964.622	0.176186	0.540361	0.147354
2025-05-28 15:31:31	4.871.462	1.952.516	-0.00954	0.541929	0.062369
2025-05-28 15:31:32	4.882.592	199.812	0.058843	0.405585	0.187012
2025-05-28 15:31:33	4.981.896	2.039.932	0.109347	0.591585	0.260185
2025-05-28 15:31:34	5.027.004	2.048.909	0.032152	0.547098	0.206229
2025-05-28 15:31:35	5.077.722	2.067.651	-0.05747	0.534459	0.200273
2025-05-28 15:31:36	514.948	207.366	-0.07776	0.592437	0.155891
2025-05-28 15:31:37	5.207.684	2.096.221	0.08191	0.589145	0.177964
2025-05-28 15:31:38	5.246.232	2.088.159	-0.06145	0.524661	0.096093
2025-05-28 15:31:39	5.309.243	2.087.106	0.090396	0.558048	0.062334
2025-05-28 15:31:40	5.347.709	2.117.358	0.081852	0.503151	0.138379

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:31:41	5.449.137	2.117.304	0.316953	0.664985	0.094397
2025-05-28 15:31:42	548.581	2.155.738	0.183416	0.571093	0.185673
2025-05-28 15:31:43	5.541.103	2.170.353	0.342751	0.565382	0.173245
2025-05-28 15:31:44	5.584.671	2.205.466	0.069727	0.524605	0.22917
2025-05-28 15:31:45	5.644.809	2.193.476	0.15138	0.548745	0.119408
2025-05-28 15:31:46	5.727.529	2.215.817	0.080296	0.636315	0.152115
2025-05-28 15:31:47	5.754.794	2.231.965	-0.15491	0.521942	0.15506
2025-05-28 15:31:48	5.806.972	2.255.394	0.221797	0.521892	0.179979
2025-05-28 15:31:49	592.177	2.261.644	0.361379	0.718799	0.143032
2025-05-28 15:31:50	5.959.391	2.281.707	0.245582	0.611052	0.161144
2025-05-28 15:31:51	6.008.626	2.303.699	0.074176	0.57372	0.17963
2025-05-28 15:31:52	6.042.682	2.296.857	0.181879	0.500391	0.101617
2025-05-28 15:31:53	6.112.043	2.321.071	0.117294	0.561159	0.145814
2025-05-28 15:31:54	6.192.034	2.352.843	-0.00679	0.636248	0.199877
2025-05-28 15:31:55	6.265.042	2.396.259	0.130728	0.665759	0.273563
2025-05-28 15:31:56	6.324.881	2.416.318	0.271484	0.644571	0.250611
2025-05-28 15:31:57	6.378.826	2.424.098	0.217451	0.611509	0.19626
2025-05-28 15:31:58	6.410.354	2.439.846	-0.09361	0.518345	0.184066
2025-05-28 15:31:59	6.466.122	2.466.226	0.143481	0.530717	0.209142
2025-05-28 15:32:00	6.560.296	2.462.648	-0.08696	0.659986	0.132113
2025-05-28 15:32:01	660.453	2.479.739	0.026427	0.591533	0.144313
2025-05-28 15:32:02	6.680.022	2.490.997	0.103498	0.642921	0.134334
2025-05-28 15:32:03	6.711.819	2.520.064	0.069792	0.540722	0.183502
2025-05-28 15:32:04	6.765.955	2.531.797	0.024872	0.54092	0.162692
2025-05-28 15:32:05	6.831.216	2.579.313	0.019477	0.576047	0.260964
2025-05-28 15:32:06	6.892.752	2.547.904	-0.10811	0.588411	0.080105
2025-05-28 15:32:07	6.952.003	257.276	0.197924	0.589703	0.133085
2025-05-28 15:32:08	701.206	2.624.922	0.087564	0.593118	0.255283
2025-05-28 15:32:09	7.073.727	2.633.759	0.03519	0.600526	0.202787
2025-05-28 15:32:10	7.158.323	2.615.471	-0.29427	0.677719	0.081492

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:32:11	7.189.127	2.659.432	0.013512	0.56145	0.194123
2025-05-28 15:32:12	7.252.931	2.681.532	0.161173	0.585538	0.202577
2025-05-28 15:32:13	7.320.363	2.711.082	0.191082	0.613462	0.2318
2025-05-28 15:32:14	7.368.267	2.722.819	0.150086	0.571184	0.195812
2025-05-28 15:32:15	7.418.821	2.749.091	0.247721	0.550539	0.216857
2025-05-28 15:32:16	7.433.366	2.749.796	-0.13706	0.423135	0.15087
2025-05-28 15:32:17	7.508.044	2.782.264	-0.31215	0.524926	0.205535
2025-05-28 15:32:18	7.560.155	2.797.917	0.244431	0.523723	0.190122
2025-05-28 15:32:19	7.656.568	2.824.339	-0.01405	0.662235	0.213426
2025-05-28 15:32:20	7.678.162	2.824.445	-0.06667	0.521873	0.146635
2025-05-28 15:32:21	772.308	2.867.381	-0.31338	0.49901	0.235554
2025-05-28 15:32:22	7.777.096	2.899.645	-0.08917	0.511951	0.262942
2025-05-28 15:32:23	7.848.169	2.901.988	0.057703	0.57447	0.187615
2025-05-28 15:32:24	7.887.215	2.947.601	-0.03396	0.516596	0.272066
2025-05-28 15:32:25	7.925.463	2.962.596	-0.01121	0.474416	0.233659
2025-05-28 15:32:26	8.002.872	2.984.965	0.171352	0.568666	0.230523
2025-05-28 15:32:27	8.067.725	3.005.522	0.103806	0.593783	0.222675
2025-05-28 15:32:28	8.135.819	306.145	-0.22754	0.621194	0.328542
2025-05-28 15:32:29	814.624	3.087.058	-0.30673	0.4586	0.30575
2025-05-28 15:32:30	8.176.236	3.079.769	-0.09946	0.408707	0.186667
2025-05-28 15:32:31	8.237.831	3.090.404	-0.19917	0.473888	0.161404
2025-05-28 15:32:32	8.281.009	3.106.507	0.22219	0.460644	0.161286
2025-05-28 15:32:33	8.342.553	315.518	-0.07281	0.509327	0.263641
2025-05-28 15:32:34	8.386.536	3.162.265	-0.50376	0.487471	0.203009
2025-05-28 15:32:35	8.413.556	3.163.687	-0.57308	0.419136	0.143633
2025-05-28 15:32:36	8.456.281	322.885	-0.24509	0.421688	0.303402
2025-05-28 15:32:37	8.539.307	3.251.914	-0.00837	0.550187	0.280516
2025-05-28 15:32:38	8.562.547	3.276.762	0.055241	0.450243	0.27044
2025-05-28 15:32:39	8.604.818	3.301.216	0.044532	0.441583	0.262297
2025-05-28 15:32:40	8.642.166	3.297.853	-0.23009	0.420163	0.169224

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:32:41	8.700.126	3.340.672	-0.24048	0.470616	0.251171
2025-05-28 15:32:42	8.750.176	3.368.343	0.000763	0.480079	0.25926
2025-05-28 15:32:43	8.822.305	3.403.166	-0.24547	0.556481	0.287438
2025-05-28 15:32:44	8.874.042	3.406.228	0.090161	0.544094	0.206112
2025-05-28 15:32:45	8.886.066	343.286	-0.29373	0.409883	0.225174
2025-05-28 15:32:46	8.917.707	3.441.257	-0.00631	0.380287	0.180467
2025-05-28 15:32:47	897.722	3.464.373	0.019635	0.448312	0.196516
2025-05-28 15:32:48	8.987.412	3.496.205	0.058248	0.338638	0.235084
2025-05-28 15:32:49	9.041.184	3.546.697	-0.13114	0.40167	0.320519
2025-05-28 15:32:50	9.090.306	3.560.701	0.053293	0.430022	0.263375
2025-05-28 15:32:51	9.132.409	3.585.911	0.079925	0.427177	0.259804
2025-05-28 15:32:52	9.176.721	3.602.055	0.175524	0.431534	0.232924
2025-05-28 15:32:53	9.230.373	3.632.387	0.306011	0.456396	0.249596
2025-05-28 15:32:54	9.281.075	3.711.128	0.316279	0.466945	0.361663
2025-05-28 15:32:55	9.311.427	3.712.122	-0.04952	0.436494	0.296128
2025-05-28 15:32:56	9.293.694	3.695.973	-0.09545	0.332619	0.218687
2025-05-28 15:32:57	935.164	3.723.202	-0.5116	0.37112	0.227047
2025-05-28 15:32:58	9.445.651	3.792.307	-0.57541	0.454065	0.294688
2025-05-28 15:32:59	9.439.515	384.846	-0.39777	0.382921	0.33152
2025-05-28 15:33:00	9.382.845	3.860.612	-0.16938	0.257317	0.303744
2025-05-28 15:33:01	9.373.618	3.890.639	-112.972	0.21253	0.303298
2025-05-28 15:33:02	952.671	3.992.737	-114.048	0.377646	0.393182
2025-05-28 15:33:03	95.993	3.829.323	-176.471	0.420613	0.143059
2025-05-28 15:33:04	962.183	3.783.552	-0.98403	0.39673	0.0696
2025-05-28 15:33:05	9.806.921	374.127	-137.842	0.573756	0.009654
2025-05-28 15:33:06	9.848.686	3.905.022	-19.797	0.554776	0.207571
2025-05-28 15:33:07	9.889.718	4.066.267	-278.628	0.5372	0.378505
2025-05-28 15:33:08	9.822.337	4.051.768	-267.223	0.389575	0.314689
2025-05-28 15:33:09	9.915.482	4.088.071	-342.827	0.455789	0.320597
2025-05-28 15:33:10	9.980.859	406.471	-192.249	0.48004	0.25271

Table F.1 Fused data sources. (cont.)

timestamp	fused _e	fused _n	fused _u	fused _{vx}	fused _{vy}
2025-05-28 15:33:11	9.962.406	4.043.156	-111.383	0.398451	0.195224
2025-05-28 15:33:12	1.004.946	4.104.124	-197.582	0.456514	0.246203
2025-05-28 15:33:13	1.011.471	4.056.235	-137.148	0.480669	0.156879
2025-05-28 15:33:14	1.003.842	412.264	-157.861	0.327291	0.219429
2025-05-28 15:33:15	9.988.236	4.202.848	-101.955	0.224955	0.291345
2025-05-28 15:33:16	9.999.335	4.327.465	-101.899	0.21088	0.409263
2025-05-28 15:33:17	1.006.266	4.354.046	-115.056	0.263057	0.391541
2025-05-28 15:33:18	1.015.651	4.321.976	-112.168	0.346512	0.303538
2025-05-28 15:33:19	1.030.818	4.260.491	-251.422	0.491111	0.190058
2025-05-28 15:33:20	1.046.337	4.464.252	-229.395	0.622184	0.418349
2025-05-28 15:33:21	103.518	4.440.787	-200.686	0.407456	0.337665
2025-05-28 15:33:22	1.038.451	4.365.441	-11.385	0.397526	0.202856
2025-05-28 15:33:23	1.047.023	427.613	-0.30348	0.454318	0.067455
2025-05-28 15:33:24	1.065.926	4.320.948	-0.32606	0.631713	0.114489
2025-05-28 15:33:25	1.067.313	4.395.641	-134.665	0.570804	0.192615
2025-05-28 15:33:26	1.071.203	4.392.967	0.296252	0.548346	0.165519
2025-05-28 15:33:27	1.072.747	4.539.936	-0.38184	0.499687	0.326613
2025-05-28 15:33:28	1.080.094	4.574.505	-0.81705	0.528723	0.328969
2025-05-28 15:33:29	1.084.193	4.669.659	-0.81386	0.514041	0.405867
2025-05-28 15:33:30	1.075.158	4.666.513	-122.363	0.338952	0.351851
2025-05-28 15:33:31	1.077.654	4.660.739	-214.673	0.327914	0.301259
2025-05-28 15:33:32	1.071.664	4.697.602	-2.1	0.213424	0.309581
2025-05-28 15:33:33	1.060.706	475.305	-148.483	0.051707	0.33983
2025-05-28 15:33:34	1.065.416	4.654.179	-0.54789	0.103505	0.175732
2025-05-28 15:33:35	1.078.169	483.304	-0.03619	0.248238	0.374951
2025-05-28 15:33:36	1.090.535	4.809.613	0.67391	0.370321	0.299701
2025-05-28 15:33:37	1.103.853	4.891.429	0.321767	0.489077	0.363741
2025-05-28 15:33:38	1.108.698	4.841.384	0.926551	0.488517	0.256997
2025-05-28 15:33:39	1.127.359	4.801.417	0.570574	0.658673	0.175886
2025-05-28 15:33:40	1.135.825	4.784.717	1.218.975	0.681889	0.133533

Table F.1 Fused data sources. (cont.)

timestamp	fused_e	fused_n	fused_u	fused_{vx}	fused_{vy}
2025-05-28 15:33:41	1.125.612	4.732.525	0.534251	0.471511	0.052572
2025-05-28 15:33:42	1.117.024	4.681.648	-0.16748	0.307191	-0.01677
2025-05-28 15:33:43	1.125.522	4.789.273	-0.30542	0.374216	0.118243
2025-05-28 15:33:44	1.127.984	4.803.833	0.491526	0.358399	0.121623
2025-05-28 15:33:45	1.131.469	4.759.569	0.8294	0.357176	0.051926
2025-05-28 15:33:46	1.140.807	4.835.406	0.705841	0.428396	0.139185
2025-05-28 15:33:47	1.156.373	4.785.631	0.589131	0.567751	0.060511
2025-05-28 15:33:48	1.164.236	4.733.434	-0.618	0.594752	-0.01144
2025-05-28 15:33:49	1.178.163	4.753.869	-127.804	0.693317	0.015217

APPENDIX G: MAHALONBIS CODINGS

Contains the source code implementing the 3D Mahalanobis-distance outlier detection: covariance estimation, threshold selection, and iterative gating with to reproduce the pre-fusion cleaning.

```
# Load the Engineering Faculty UWB dataset (assumed to be the 15 29 50 dataset)
df = pd.read_csv("/mnt/data/location data tag D504 28-05-2025-15 29 50.csv")
```

```
# Stepwise 3D vector calculation
```

```
coords = df[['X', 'Y', 'Z']].values
```

```
deltas = np.diff(coords, axis=0)
```

```
# Mean and covariance of stepwise movement
```

```
mean_step = np.mean(deltas, axis=0)
```

```
cov_step = np.cov(deltas.T)
```

```
inv_cov_step = np.linalg.inv(cov_step)
```

```
# Mahalanobis distances and outlier flagging
```

```
mahal_dists = [mahalanobis
```

```
(d, mean_step, inv_cov_step) for d in deltas]
```

```
mahal_squared = np.insert(np.square
```

```
(mahal_dists), 0, 0) # Pad with 0 to match
```

```
row count
```

```
df['Mahalanobis Step3D2'] = mahal_squared
```

```
# Chi-squared threshold at 99% confidence, df = 3
```

```
threshold = chi2.ppf(0.99, df=3)
```

```
df['Outlier 3D Mahalanobis'] =
```

```

df['Mahalanobis Step3D2'] > threshold

# Calculate final statistics
total points = len(df)

outlier count = df['Outlier 3D Mahalanobis'].sum()
outlier percent = (outlier count / total points) * 100
mean vector = mean step
cov matrix = cov step

total points, outlier count,
outlier percent, mean vector, cov matrix
import pandas as pd
import numpy as np
from scipy.spatial.distance import mahalanobis
from scipy.stats import chi2

# Load the Engineering Faculty dataset
file path = "/mnt/data/location data tag
D504 28-05-2025-15 29 50.csv"
df = pd.read_csv(file path)

# Compute stepwise deltas
coords
= df[['X', 'Y', 'Z']].values
deltas = np.diff(coords, axis=0)

# Compute mean and covariance of the step vectors

```

```

mean step = np.mean(deltas, axis=0)
cov step = np.cov(deltas.T)
inv covstep = np.linalg.inv(cov step)

#ComputeMahalanobisdistanceandsquareit mahal dists =
[mahalanobis(d, mean step, inv cov step)fordindeltas]
mahal squared =np.insert
(np.square(mahal dists), 0, 0)
#padf irstrowforalignment

df ['MahalanobisStep3D'] = mahal squared threshold = chi2.ppf
(0.99, df = 3)
# Chi-squaredthresholdf or99% confidence, df = 3

df ['Outlier 3DM ahalanobis'] = df ['Mahalanobis Step3D'] >
threshold

#Computestatistics
total points = len(df )
outlier count = df
Outlier 3DM ahalanobis.sum()
outlierpercent = (outlier count/totalpoints)* 100
meanvector = meanstep
covmatrix = cov step
totalpoints, outlier count, outlierpercent, mean vector, cov matrix

```

APPENDIX H: UNSCENTED KALMAN FILTER CODINGS

Includes the source code implementing the UKF for GPS-UWB fusion (state definition, process/measurement models, sigma-point generation, and example run scripts).

```
def ukf_fuse(gps_data, uwb_data, dt):
```

```
    """
```

```
    Perform UKF fusion on synchronized GPS and UWB ENU data.
```

```
    gps_data, uwb_data: arrays of shape
```

```
    (N,3) with columns (east,north,up).
```

```
    dt: time step (constant assumed).
```

```
    Returns fused trajectory (N,3).
```

```
    """
```

```
    # State = [x, y, z, vx, vy, vz]
```

```
    x = np.hstack((gps_data[0], [0,0,0]))
```

```
    # init at first position, zero velocity
```

```
    P = np.eye(6) * 1.0
```

```
    Q = np.eye(6) * 0.01
```

```
    # process noise
```

```
    R = np.eye(6) * 0.5
```

```
    # measurement noise (3 for GPS, 3 for UWB)
```

```
    # UKF weights (for example)
```

```
    n = 6; alpha=1e-3; beta=2; kappa=0
```

```
    lam = alpha**2*(n+kappa)-n
```

```
    Wm = np.full(2*n+1, 1/(2*(n+lam)))
```

```
    Wc = np.copy(Wm)
```

```
    Wm[0] = lam/(n+lam); Wc[0] = lam/(n+lam)
```

```
    + (1-alpha**2+beta)
```

```

fused = []
for (gx, gy, gz), (ux, uy, uz)
in zip(gps_data, uwb_data):
    # 1) Generate sigma points
    sqrtP = LA.cholesky((n+lam)*P)
    Xsig = np.zeros((2*n+1, n))
    Xsig[0] = x
    for i in range(n):
        Xsig[i+1] = x + sqrtP[i]
        Xsig[i+1+n] = x - sqrtP[i]
    # 2) Predict sigma points through
    motion model (constant vel)
    Xpred = np.zeros_like(Xsig)
    for i in range(2*n+1):
        xi = Xsig[i]
        # motion model
        Xpred[i,0:3] = xi[0:3] + xi[3:6]*dt
        Xpred[i,3:6] = xi[3:6]
    # 3) Compute predicted state mean and covariance
    x_pred = np.dot(Wm, Xpred)
    P_pred = Q.copy()
    for i in range(2*n+1):
        diff = (Xpred[i] - x_pred).reshape(-1,1)
        P_pred += Wc[i] * (diff @ diff.T)
    # 4) Compute predicted measurement (stack GPS+UWB)
    # Both sensors measure [x,y,z] with identity
    Zsig = np.zeros((2*n+1, 6))
    for i in range(2*n+1):
        Zsig[i, 0:3] = Xpred[i, 0:3]
    # GPS portion
    Zsig[i, 3:6] = Xpred[i, 0:3]

```

```

# UWB portion (same state part)
z_pred = np.dot(Wm, Zsig)
Pz = R.copy()
for i in range(2*n+1):
    dz = (Zsig[i] - z_pred).reshape(-1,1)
    Pz += Wc[i] * (dz @ dz.T)
# 5) Cross covariance
Pxz = np.zeros((n,6))
for i in range(2*n+1):
    dx = (Xpred[i] - x_pred).reshape(-1,1)
    dz = (Zsig[i] - z_pred).reshape(-1,1)
    Pxz += Wc[i] * (dx @ dz.T)
# 6) Kalman gain and update
K = Pxz @ np.linalg.inv(Pz)
z = np.array([gx, gy, gz, ux, uy, uz])
x = x_pred + K @ (z - z_pred)
P = P_pred - K @ Pz @ K.T
fused.append(x[0:3])
return np.array(fused)

# Example usage:
# fused_traj = ukf_fuse(gps_clean
['east','north','up'
].values,
uwbclean[['east','north','up']].
values, dt=0.1)

```


APPENDIX I: CAD DRAWINGS

The CAD (DWG/DXF) of the Faculty of Engineering were standardized at the layer level and purged of extraneous elements for BIM environment integration, then registered to the project's shared coordinate system; subsequently, they were linked/imported into Revit, where level and grid references were used to model the primary architectural and structural components at LOD 300.



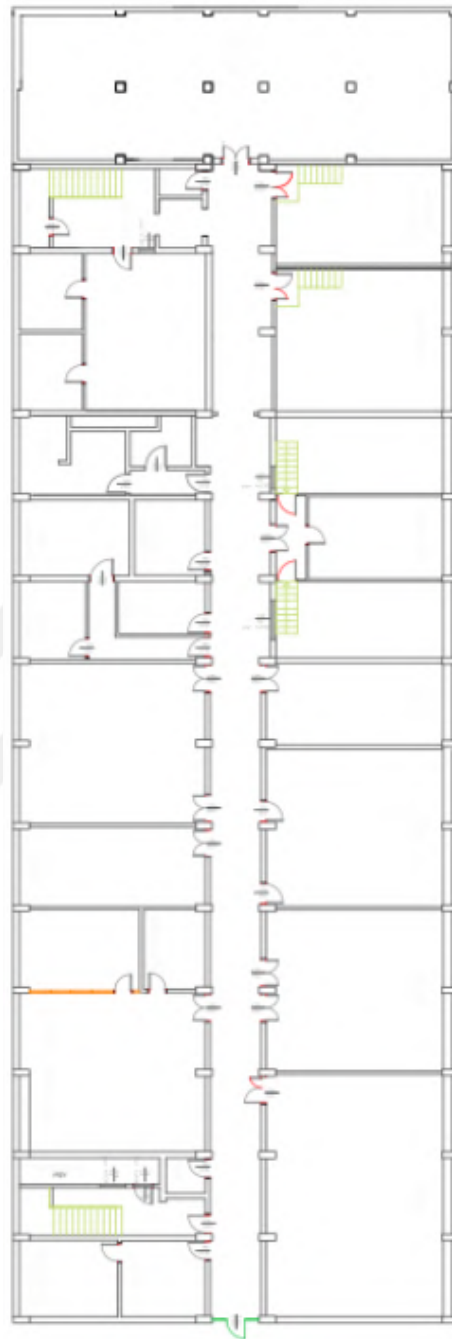


Figure I.1. Faculty of engineering basement floor plan.



Figure I.2. Faculty of engineering ground floor plan.

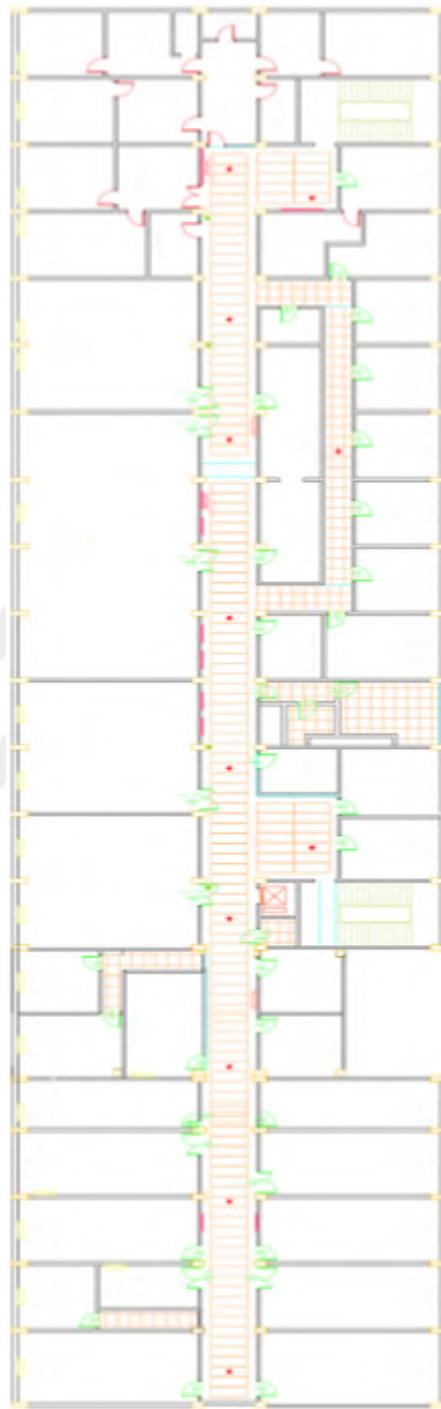


Figure I.3. Faculty of engineering first floor plan.



Figure I.4. Faculty of engineering second floor plan.