

Field Test Bed for Evaluating Embedded Vehicle Sensors with Indiana Companies

by

Ayman Habib*

Lyles School of Civil Engineering, Purdue University

Darcy Bullock

Lyles School of Civil Engineering, Purdue University

Yi-Chun Lin

Lyles School of Civil Engineering, Purdue University

Raja Manish

Lyles School of Civil Engineering, Purdue University

Radhika Ravi

Lyles School of Civil Engineering, Purdue University

*Corresponding Author.

Phone: (765)-496-0173. Email: ahabib@purdue.edu.

Joint Transportation Research Program
SPR-4438

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views and policies of the Indiana Department of Transportation or the Federal Highway Administration. The report does not constitute a standard, specification or regulation.

May 12, 2022

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA/IN/JTRP-2020/xx	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Field Test Bed for Evaluating Embedded Vehicle Sensors with Indiana Companies		5. Report Date May 12, 2022	
		6. Performing Organization Code	
7. Author(s) Ayman Habib, Darcy Bullock, Yi-Chun Lin, Raja Manish, Radhika Ravi		8. Performing Organization Report No. FHWA/IN/JTRP-2020/xx	
9. Performing Organization Name and Address Joint Transportation Research Program Hall for Discovery and Learning Research (DLR), Suite 204 207 S. Martin Jischke Drive West Lafayette, IN 47907		10. Work Unit No.	
		11. Contract or Grant No. SPR-4438	
12. Sponsoring Agency Name and Address Indiana Department of Transportation (SPR) State Office Building 100 North Senate Avenue Indianapolis, IN 46204		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.			
16. Abstract With the advent of modern sensing technology, mapping products have begun to achieve an unprecedented precision of measurements. Considering their diverse use cases, several factors play a role in what would make the resulting measurements accurate. For light detection and ranging (LiDAR) and photogrammetry-based mapping solutions that implement vehicles outfitted with laser ranging devices, RGB cameras, and global navigation satellite system/inertial navigation system (GNSS/INS) georeferencing units, the quality of the derived mapping products is governed by the combined accuracy of the various sensors. While ranging errors associated with LiDAR systems or the imaging quality of RGB cameras are sensor-dependent and are mostly constant, the accuracy of a georeferencing unit depends on a variety of extrinsic factors, including but not limited to, availability of clear line-of-path to GNSS satellites and presence of radio interferences. The quality of GNSS signal, in turn, is affected by the grade of hardware components used and, to a great extent, obstructions to signal reception. This document reports some of the major challenges of vehicle-based mobile mapping with regards to GNSS/INS navigation. The background of GNSS/INS positioning is discussed to build a framework for trajectory enhancement as well as improvement of LiDAR mapping products. The focus is put on using available sensor data from LiDAR and/or cameras to enhance their position/orientation quality. Some of the best practices in light of potential trajectory deterioration are also recommended.			
17. Key Words Mobile mapping systems, direct georeferencing, LiDAR, registration, trajectory enhancement, semantic segmentation		18. Distribution Statement No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 139	22. Price

EXECUTIVE SUMMARY

Field Test Bed for Evaluating Embedded Vehicle Sensors with Indiana Companies

Introduction

Indiana is fortunate to have a relatively dense InCORS differential correction network that provides to high accuracy GNSS positions for a variety of use cases, such as mobile mapping, forestry and agriculture. However, when distance from the base station increases or satellite occlusions occur due to structures, vegetations, urban canons, and power lines (Figure 1), the spatial accuracy decreases.

The objectives of this project were to identify limitations in the precision/accuracy of the GNSS/INS-based geo-referencing for mobile mapping systems, evaluate the current InCORS network to ensure GNSS accuracy, and develop mitigation measures for degraded/occluded GNSS areas to provide high quality mobile mapping.



Figure 1. Examples of environmental factors that will impact the quality of the GNSS trajectory: Steel overpass, concrete bridge overpass, electromagnetic interference, and heavy canopy cover.

Findings

Without additional sensors, such as high-quality inertial navigation sensors and LiDAR sensors, situations such as those shown in Figure 1 significantly degrade the baseline GNSS positioning capability of vehicles due to signal attenuation, noise, and multi-path issues. Although inertial navigation sensors can provide short duration compensation, high quality inertial navigation systems are often prohibitively expensive. In contrast, integrating LiDAR data into the mobile mapping environment allows the vehicle to leverage well known feature characteristics to provide compensation (Figure 2). This report details several of these mathematical techniques and validates them with traditional surveying techniques (Figure 3).

Implementation

Throughout this study, these techniques were used across a number of INDOT mapping projects with degraded GNSS, including:

- Bridge deck thickness mapping ¹
- Indoor salt pile mapping (Figures 4 and 5)
- Semantic segmentation of bridge components and road elements (Figure 6)
More results are available through the following links: <https://youtu.be/5zd-L9fvFbs>; <https://youtu.be/Op3jjxn1Oj0>

In addition, the team supported the INDOT surveying department on fabricating brackets for new antennas as InCORS nodes were updated and brought back online. Beyond serving these near-term use cases, the fundamental techniques, such as those shown in Figures 2, 4, and 5 will advance basic autonomous technology.

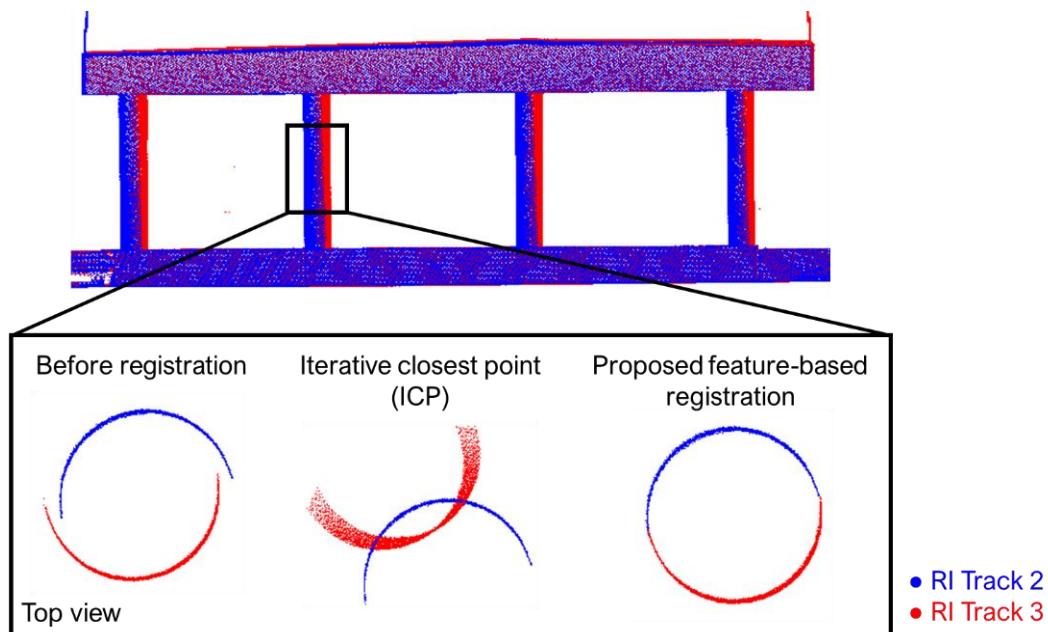
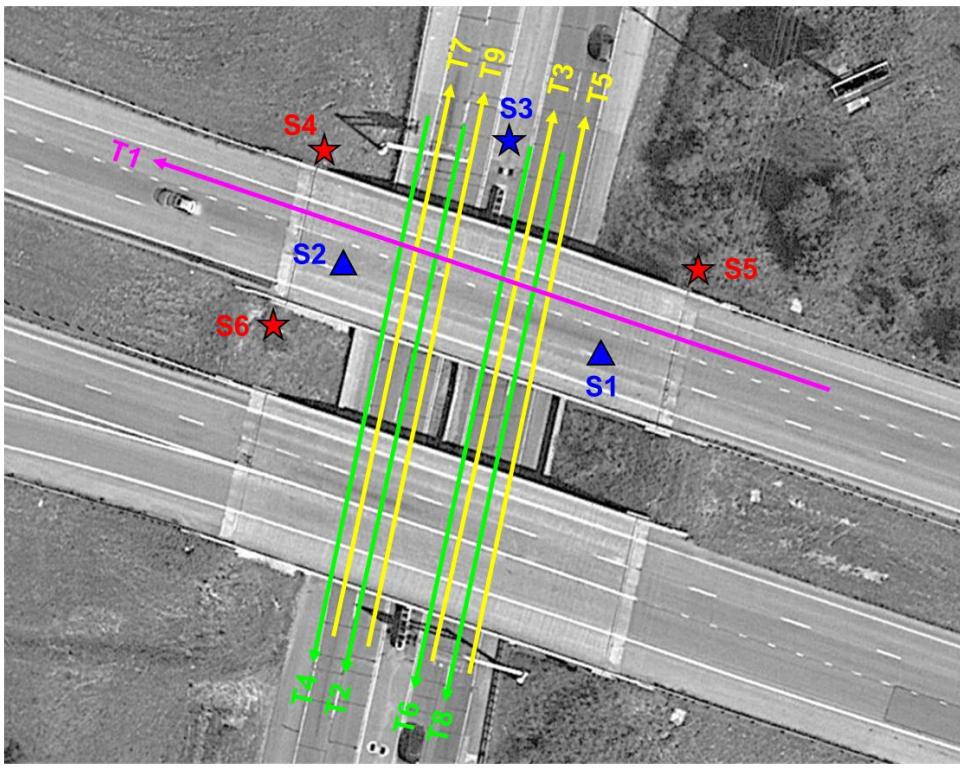


Figure 2. Sample registration results using the iterative closest point (ICP) and proposed approach, showing a cylindrical feature before and after registration.

¹ Lin Y-C, Liu J, Cheng Y-T, Hasheminasab SM, Wells T, Bullock D, Habib A. Processing Strategy and Comparative Performance of Different Mobile LiDAR System Grades for Bridge Monitoring: A Case Study. Sensors. 2021; 21(22):7550. <https://doi.org/10.3390/s21227550>



- ← Track above the bridge (westbound on I-74)
- ← Track below the bridge (southbound on US-231)
- ← Track below the bridge (northbound on US-231)
- ▲ FARO scan below the bridge on US-231
- ★ FARO scan on US-231
- ★ Trimble scan on the I-74 bridge embankment

(a)



(b)

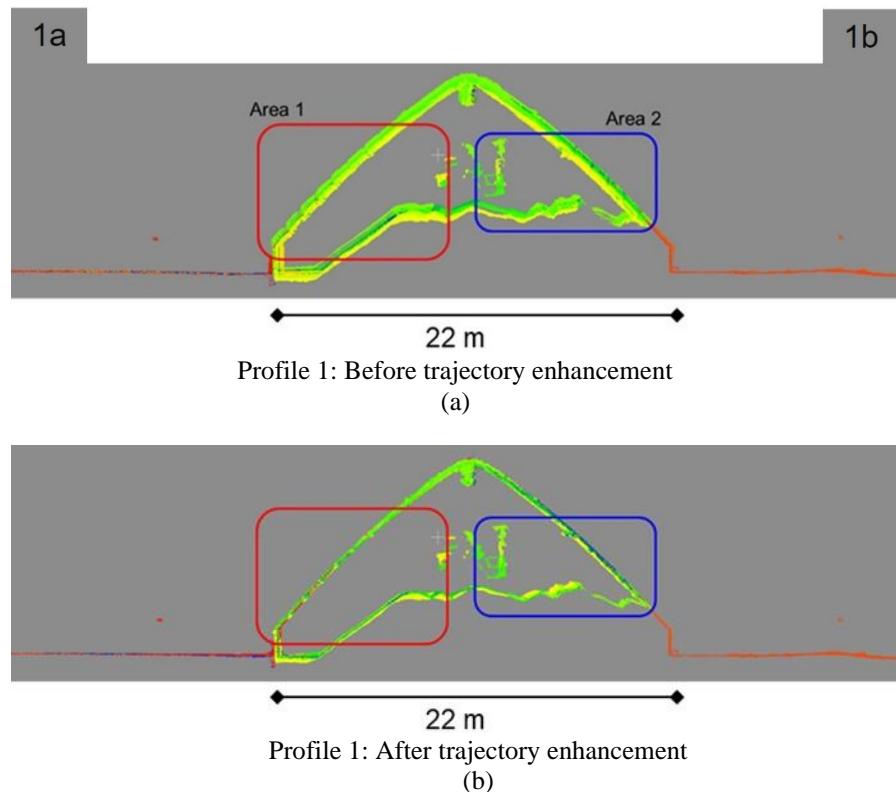


(c)

Figure 3. Data acquisition: (a) drive run configuration for the vehicles (Tracks T1–T9) and TLS scan locations (Scans S1–S6), (b) image of the Trimble station (Scan S4) atop the I-74 embankment outside the barrier rail, and (c) image of the FARO station (Scan S1) on the US-231 under the I-74 bridge.



Figure 4. Indoor mapping of dome facility using a crane bucket.



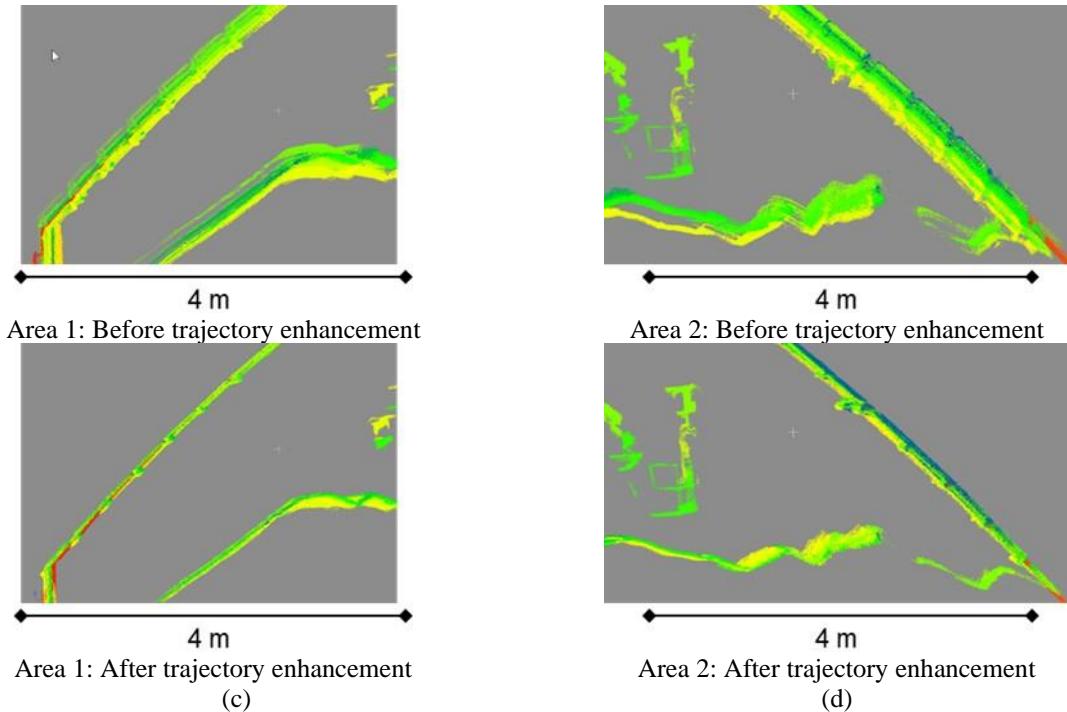


Figure 5. Sample profile (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.

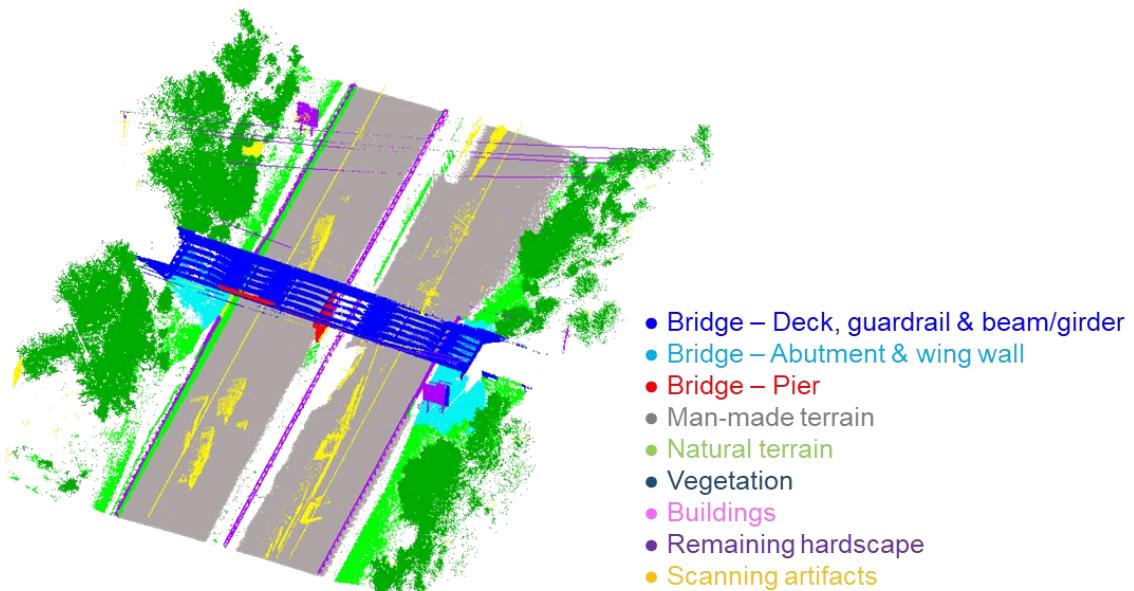


Figure 6. Sample semantic segmentation result using mobile LiDAR point clouds.

CONTENTS

1. INTRODUCTION	17
2. GNSS/INS-DERIVED POSITIONING.....	20
2.1 Components of GNSS/INS Navigation.....	20
2.1.1 GNSS Receiver-Satellite Infrastructure.....	20
2.1.2 Inertial Measurement Unit (IMU) and Inertial Navigation System (INS).....	21
2.2 Working Principle of GNSS Navigation.....	25
2.2.1 Measurement Observables	26
2.2.2 Autonomous and Relative (differential) Positioning	27
2.3 System-level Factors Influencing the Quality of GNSS/INS Trajectories.....	29
2.3.1 Selection of GNSS hardware components	29
2.3.2 Processing Scheme.....	30
2.3.3 Availability of Continuously Operating Reference Stations (CORS)	31
3. ENVIRONMENTAL FACTORS INFLUENCING GNSS/INS-DERIVED TRAJECTORY ACCURACY	33
3.1 Signal outages due to physical obstructions.....	33
3.2 Electromagnetic Interference	37
3.3 Concluding Remarks	37
4. PROCESSING STRATEGY AND COMPARATIVE PERFORMANCE OF DIFFERENT MOBILE LIDAR SYSTEM GRADES FOR BRIDGE MONITORING: A CASE STUDY	39
Abstract	39
4.1 Introduction	39
4.2 Related Work.....	41
4.2.1 LiDAR for Infrastructure Mapping.....	41
4.2.2 Point Cloud Registration.....	42
4.3 Data Acquisition Systems and Field Surveys	43
4.3.1 System Description and Calibration	43
4.3.2 Study Site and Field Surveys	45
4.4 Methodology for Point Cloud Registration and Bridge Deck Thickness Evaluation	49
4.4.1 Feature-Based Fine Registration.....	50
4.4.2 Bridge Deck Thickness Evaluation.....	55
4.4.3 Comparative Quality Assessment	56
4.5 Experimental Results and Discussion	57

4.5.1	Point Cloud Registration and Alignment.....	57
4.5.2	Point Cloud Registration and Alignment.....	64
4.6	Conclusions and Recommendations for Future Work	66
5.	LIDAR-AIDED TRAJECTORY ENHANCEMENT FOR MOBILE MAPPING SYSTEMS IN GNSS-CHALLENGING URBAN ENVIRONMENTS	67
	Abstract	67
5.1	Introduction	67
5.2	Methodology	71
	5.2.1 Automated Planar Feature Extraction and Matching.....	71
	5.2.2 Trajectory Enhancement Framework.....	76
5.3	Experimental Results.....	80
	5.3.1 Dataset 1: Underpass Data Acquisition	83
	5.3.2 Dataset 2: Indoor Mapping	89
	5.3.3 Dataset 3: Indoor Mapping of Dome Facility using Crane Bucket	95
5.4	Conclusions and Recommendations for Future Work	102
6.	SEMANTIC SEGMENTATION OF BRIDGE COMPONENTS AND ROAD ELEMENTS USING MOBILE LIDAR DATA.....	103
	Abstract	103
6.1	Introduction	103
6.2	Data Acquisition System and Dataset Description	106
	6.2.1 Mobile LiDAR mapping systems	106
	6.2.2 Dataset description.....	108
6.3	Methodology	109
	6.3.1 Point cloud preprocessing.....	110
	6.3.2 Point cloud annotation	111
	6.3.3 Cross-labeling	112
	6.3.4 Deep learning-based point cloud semantic segmentation.....	113
	6.3.5 Transfer learning.....	115
6.4	Experimental Results.....	116
	6.4.1 Cross-labeling performance assessment	117
	6.4.2 Deep learning-based semantic segmentation: baseline model.....	118
	6.4.3 Transfer learning between scenes	120
	6.4.4 Transfer learning between scenes and systems.....	123
6.5	Conclusions and Future Work.....	126

7. CONCLUSIONS.....	127
REFERENCES	128

LIST OF TABLES

Table 1-1 PWMMS-HA and PWMMS-UHA GNSS/INS specifications.....	18
Table 1-2: PWMMS-HA and PWMMS-UHA LiDAR sensor specifications	19
Table 2-1: Satellite constellations available for GNSS navigation.....	21
Table 2-2 Pros and cons of GNSS and INS systems	23
Table 2-3 PWMMS-HA/UHA GNSS/INS component classification	25
Table 4-1 Specifications of the acquired point clouds by PWMMS-HA, PWMMS-UHA, and Terrestrial laser scanners (TLSs) above and below the bridge in question.	47
Table 4-2 Weighted average of the RMSE of plane/line/cylinder fittings before and after registration for TLS, PWMMS-HA, and PWMMS-UHA.	61
Table 4-3 Estimated post-registration radii of the cylindrical columns supporting the I-74 bridge from the TLS, PWMMS-HA, and PWMMS-UHA point clouds.....	62
Table 4-4 Statistics of the difference between bridge deck thickness estimates from different systems.....	66
Table 5-1 Data acquisition specifications and parameters used for each procedure employed during trajectory enhancement.....	83
Table 5-2 Dataset 1: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 45,454,918 points)	87
Table 5-3 Dataset 1: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement	89
Table 5-4 Dataset 2: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 9,432,850 points)	93
Table 5-5 Dataset 2: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement	95
Table 5-6 Dataset 3: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 51,420,371 points)	99
Table 5-7 Dataset 3: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement	102
Table 6-1 Specifications of the georeferencing and LiDAR sensors and the expected accuracy of the point cloud for the mobile LiDAR mapping systems.	107
Table 6-2 Performance measures for the proposed cross-labeling approach.	118
Table 6-3 Performance of Model A on the testing set from the PWMMS-UHA bridge dataset.	119
Table 6-4 Performance of Models A, B, and C on the testing set from the PWMMS-UHA highway dataset.....	122
Table 6-5 Performance of Models A, D, and E on the testing set from the PWMMS-HA highway dataset.	125

LIST OF FIGURES

Figure 1-1 Summary of the report	17
Figure 1-2 Vehicle-based mobile mapping systems used in the study	18
Figure 2-1 Working mechanism of GNSS/INS navigation	20
Figure 2-2 Principal components of GNSS navigation [5]	21
Figure 2-3 A 6-DOF IMU with accelerometer/gyroscope axes.....	22
Figure 2-4 Apollo program IMU on display at Draper Labs, Cambridge, MA [6]	22
Figure 2-5 Working mechanism of a strapdown INS	23
Figure 2-6 Types of IMU (a) Silicon MEMS, (b) FOG, (c) RLG, (d) Quartz MEMS [7–9]	24
Figure 2-7 (a) Conventional trilateration using control points (P1-P3) to locate a fixed-point A; (b) GNSS positioning with satellites as control points [5]	26
Figure 2-8 GNSS observables measured by a receiver. (a) Square-wave pseudorandom code, (b) Sinewave GNSS carrier signal, and (c) Time shift between PRC and receiver's replica [5]	27
Figure 2-9 Differential GNSS positioning using two receivers [5]	28
Figure 2-10 GNSS antenna used in various applications. (Left to right) Antennas used on a UAV, land-based vehicle, and for geodetic surveying, respectively. All are from the same manufacturer: Trimble [11].....	30
Figure 2-11 (a) CORS network within and around Indiana with their nominal coverage. (b) Effective coverage evaluated for two separate base stations, and (c) A simulation of the effective coverage of base stations to achieve a position accuracy of 1 cm	32
Figure 3-1 GNSS data acquisition in agricultural and urban environments. Except for a few differences, both scenarios are analogous.....	33
Figure 3-2 Backpack and UGV-based mobile mapping platforms.....	34
Figure 3-3 (a) Data acquisition in an agricultural field, (b) Satellite availability for backpack system, and (c) Satellite availability for unmanned ground vehicle (UGV).....	34
Figure 3-4 Combined separation plot for (a) Backpack trajectory, and (b) UGV trajectory	35
Figure 3-5 Impact of roadside vegetation on GNSS trajectory and LiDAR-derived point clouds. (a) Area showing canopy cover on both sides of the road, (b) Different views of the post- processed trajectory colored by position inaccuracy, (c) Post-processed trajectory standard deviation and satellite availability, and (d) Discrepancies in the reconstructed point clouds from different tracks.	36
Figure 3-6 EM-induced misalignment in reconstructed point clouds.....	37
Figure 3-7 Examples of environmental factors that will impact the quality of the GNSS trajectory: Steel overpass, concrete bridge overpass, electromagnetic interference, and heavy canopy cover.	38
Figure 4-1 The wheel-based mobile mapping systems and onboard sensors used in this study: (a) Purdue Wheel-based Mobile Mapping System-High Accuracy (PWMMS-HA) and (b) Purdue Wheel-based Mobile Mapping System-Ultra High Accuracy (PWMMS-UHA). Both platforms are non-commercial systems designed and integrated by the research group.	44
Figure 4-2 Study site: (a) the westbound bridge at the intersection of the I-74 and US-231 near Crawfordsville in Indiana, USA (aerial photo adapted from Google Earth images), (b) image of the bridge captured by PWMMS-HA while driving westbound on the I-74, and (c) side view of the bridge captured by PWMMS-HA while driving southbound on the US-231.....	46

Figure 4-3 Data acquisition: (a) drive run configuration for the vehicles (Tracks T1–T9) and TLS scan locations (Scans S1–S6), (b) image of the Trimble station (Scan S4) atop the I-74 embankment outside the barrier rail, and (c) image of the FARO station (Scan S1) on the US-231 under the I-74 bridge.....	48
Figure 4-4 Global navigation satellite system/inertial navigation system (GNSS/INS) position accuracy charts for the (a) PWMMS-HA (north and east positions), (b) PWMMS-HA (down position), and (c) PWMMS-UHA vehicles. The highlighted eight peaks correspond to the eight southbound and northbound tracks (Tracks T2–T9) on the US-231 below the bridge, where suboptimal position accuracy can be observed.....	49
Figure 4-5 Workflow of the proposed bridge deck thickness evaluation and comparative quality assessment strategy.	50
Figure 4-6 An example of cylindrical features (in green) that have been segmented from a LiDAR point cloud (colored by intensity).	51
Figure 4-7 Different options for representing planar features showing the normal vectors to the planes (defined by the eigenvector corresponding to the smallest eigenvalue) that are mainly along the (a) Z-axis (i.e., the eigenvector component along the Z-axis is larger than those along the X and Y axes), (b) Y-axis (i.e., the eigenvector component along the Y-axis is larger than those along the X and Z axes), and (c) X-axis (i.e., the eigenvector component along the X-axis is larger than those along the Y and Z axes).	52
Figure 4-8 Different options for representing cylindrical features with directions (defined by the eigenvector corresponding to the largest eigenvalue) that are mainly along the (a) Z-axis (i.e., the eigenvector component along the Z-axis is larger than those along the X and Y axes), (b) Y-axis (i.e., the eigenvector component along the Y-axis is larger than those along the X and Z axes), and (c) X-axis (i.e., the eigenvector component along the X-axis is larger than those along the Y and Z axes).	52
Figure 4-9 Schematic illustration of the normal distance vector components that are minimized using the (a) plane-fitting model, (b) 3D line-fitting model, and (c) cylinder-fitting model.....	53
Figure 4-10 Sample registration results using the iterative closest point (ICP) and proposed approach, showing a cylindrical feature before and after registration.	55
Figure 4-11 Example of (a) a point cloud capturing the top and bottom surfaces of the bridge deck and (b) a heat map representing the estimated bridge deck thickness.....	56
Figure 4-12 Point density over the bridge deck from one track (track T1) for: (a) PWMMS-HA and (b) PWMMS-UHA.....	57
Figure 4-13 Sample cross-sectional profile of the bridge deck showing the point cloud alignment before registration for the (a) PWMMS-HA and (b) PWMMS-UHA datasets. Both datasets have nine tracks (T1 is above the bridge and T2–T9 are below the bridge).	58
Figure 4-14 Extracted planar/linear/cylindrical features (in blue/red/green, respectively) for the registration of (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA point clouds.	60
Figure 4-15 Box and whisker plots of the fine-registration transformation parameters for the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.	61
Figure 4-16 The twelve cylindrical columns and their axes derived from TLS (in blue), PWMMS-HA (in red), and PWMMS-UHA (in green) data.....	62
Figure 4-17 Sample cross-sectional profile showing the post-registration point cloud alignment for the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.	63

Figure 4-18 Bridge deck thickness estimates shown as a heat map using the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.....	64
Figure 4-19 Heat map visualization of the difference in bridge deck thickness estimates between (a) TLS and PWMMS-HA, (b) TLS and PWMMS-UHA, and (c) PWMMS-HA and PWMMS-UHA.....	65
Figure 5-1 Schematic illustration of possible wrong matches in case of large misalignment and similar appearance of planar surfaces (false matches 1, 2, and 3 show possible wrong matches between planar surfaces).....	72
Figure 5-2 (a) Flowchart for feature extraction and matching without loop closure; (b) planar feature matching flowchart; and (c) flowchart for feature extraction and matching with loop closure	74
Figure 5-3 Sample image and real point cloud along a narrow indoor corridor colored by time (total of 120 secs) and feature ID assigned after conducting feature matching without/with loop closure	75
Figure 5-4 Flowchart for trajectory enhancement model that takes matched planar features from LiDAR as input to produce enhanced GNSS/INS trajectory	76
Figure 5-5 Down sampled trajectory reference points (down sampling time interval Δt) used for trajectory enhancement: t_i to $t_i + n - 1$ and t_i' to $t_i + n - 1'$ denote the absolute and relative timestamps of the n neighboring trajectory reference points for a generic timestamp t_0	77
Figure 5-6 Range-based adaptive variance for trajectory enhancement: standard deviation (σ_{nd}) as a function of the range (ρ_i) of captured points.....	79
Figure 5-7 Study sites used for trajectory enhancement: (a) Dataset 1: underpass data acquisition; (b) Dataset 2: indoor mapping; (c) Dataset 3: indoor mapping of dome facility using a crane bucket	81
Figure 5-8 Backpack MLMS used in this study	82
Figure 5-9 Dataset 1: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results	84
Figure 5-10 Dataset 1: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure.....	86
Figure 5-11 Dataset 1: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.....	87
Figure 5-12 Dataset 1: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position ..	88
Figure 5-13 Dataset 1: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters	89
Figure 5-14 Dataset 2: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results	90
Figure 5-15 Dataset 2: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure.....	91
Figure 5-16 Dataset 2: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.....	92
Figure 5-17 Dataset 2: Profile 3 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.....	93

Figure 5-18 Dataset 2: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position ..	94
Figure 5-19 Dataset 2: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters with respect to time.....	94
Figure 5-20 Dataset 3: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results	96
Figure 5-21 Dataset 3: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure.....	97
Figure 5-22 Dataset 3: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.....	98
Figure 5-23 Dataset 3: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement.....	99
Figure 5-24 Dataset 3: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position	100
Figure 5-25 Dataset 3: Dome surface, side wall, and salt surface before and after trajectory enhancement colored by outdoor (blue) and indoor (red) trajectory portions	101
Figure 5-26 Dataset 3: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters	101
Figure 6-1 The wheel-based mobile mapping systems and onboard sensors: (a) PWMMS-HA and (b) PWMMS-UHA.....	107
Figure 6-2 Study site showing data collection route and bridge locations (aerial photo adopted from Google Earth Image).	108
Figure 6-3 Sample data from the mobile mapping system showing the images (from southbound and northbound) and point cloud (colored by intensity) corresponding to Bridge 2.....	109
Figure 6-4 Framework of the proposed deep learning-based semantic segmentation for bridge components and road elements.	109
Figure 6-5 Planimetric point density map for (a) the original point cloud and (b) down-sampled point cloud with a minimum point spacing of 5 cm (Bridge 2 in the PWMMS-UHA bridge dataset).	110
Figure 6-6 Example of point cloud down-sampling showing (a) the original point cloud and (b) down-sampled point cloud with a minimum point spacing of 5 cm.....	110
Figure 6-7 Classes for bridge components: deck, guardrail, beam, and girder in blue; abutment and wing wall in cyan; pier in red.....	111
Figure 6-8 Sample annotated point cloud colored by class (Bridge 2 in the PWMMS-UHA bridge dataset).	112
Figure 6-9 Illustration of the cross-labeling approach: (a) data alignment between datasets acquired by different MLMSs and (b) manually-annotated PWMMS-UHA data and the corresponding cross-labeled PWMMS-HA data (Bridge 2).....	113
Figure 6-10 Sample geometrically homogeneous partition result (Bridge 2 in the PWMMS-UHA bridge dataset): (a) point cloud colored by height and (b) point cloud randomly colored by partition ID.....	114
Figure 6-11 Schematic diagram of the superpoint graph (SPG) approach: (a) superpoint graph construction and (b) superpoint embedding and contextual segmentation.....	114

Figure 6-12 Schematic diagram for the training (three-fold cross-validation) and testing stage.	115
Figure 6-13 Schematic diagram of the SPG approach showing the fixed layers for transfer learning in blue.	115
Figure 6-14 Cross labeled results and manually-annotated reference (Bridge 3, Tile 25, and Tile 212 from PWMMS-HA).	117
Figure 6-15 Predictions from Model A for the testing set (Bridges 2, 5, 10, 15, 18, 21 from the PWMMS-UHA bridge dataset)	119
Figure 6-16 Predictions from Models A, B, and C for the testing set (Tiles 14, 212, 340, 376, 445, 499 from the PWMMS-UHA highway dataset).	121
Figure 6-17 Predictions from Models A, D, and E for the testing set (Tiles 14, 212, 340, 376, 445, 499 from the PWMMS-HA highway dataset).	124
Figure 7-1 Main components of vehicle-based mobile mapping with regards to GNSS/INS navigation.	127

1. INTRODUCTION

With the advent of modern sensing technologies, mapping products have begun to achieve an unprecedented precision/accuracy. Considering their diverse use cases, several factors play a role in what would impact the quality of the resulting measurements. For light detection and ranging (LiDAR) and photogrammetry-based mapping solutions that implement vehicles outfitted with laser ranging devices, RGB cameras, and global navigation satellite system/inertial navigation system (GNSS/INS) georeferencing units, the quality of the derived mapping products is governed by the combined accuracy of the various sensors. While ranging errors associated with LiDAR systems or the imaging quality of RGB cameras are sensor-dependent and are mostly constant, the accuracy of a georeferencing unit depends on a variety of extrinsic factors, including but not limited to, availability of clear line-of-path to GNSS satellites and presence of radio interferences. The quality of GNSS signal, in turn, is affected by the grade of hardware components used and, to a great extent, obstructions to signal reception.

This document reports some of the major challenges of vehicle-based mobile mapping with regards to GNSS/INS navigation. The background of GNSS and inertial measurement unit (IMU)-based positioning is discussed to build a framework for trajectory enhancement as well as improvement of LiDAR mapping products. The focus is put on using available sensor data from LiDAR and/or cameras to enhance their position/orientation quality. Some of the best practices in light of potential trajectory deterioration are also recommended. The report is structured as follows: chapter 1 introduces land-based mobile mapping systems utilized in this study along with a description of various sensors onboard these platforms; chapter 2 presents the background of GNSS/INS navigation and chapter 3 reviews some of the major environmental factors that impact the accuracy of the GNSS/INS-derived trajectory upon which the entire mapping prospective is built; considering key navigational challenges, chapters 4 discusses a data driven error mitigation approach through point cloud registration; chapter 5 introduces a system driven error mitigation approach by means of trajectory enhancement; finally, in chapter 6, a novel framework is developed to utilize the enhanced mapping products for semantic segmentation of highway elements from acquired surveys along transportation corridors. Figure 1-1 shows the structure of this report.

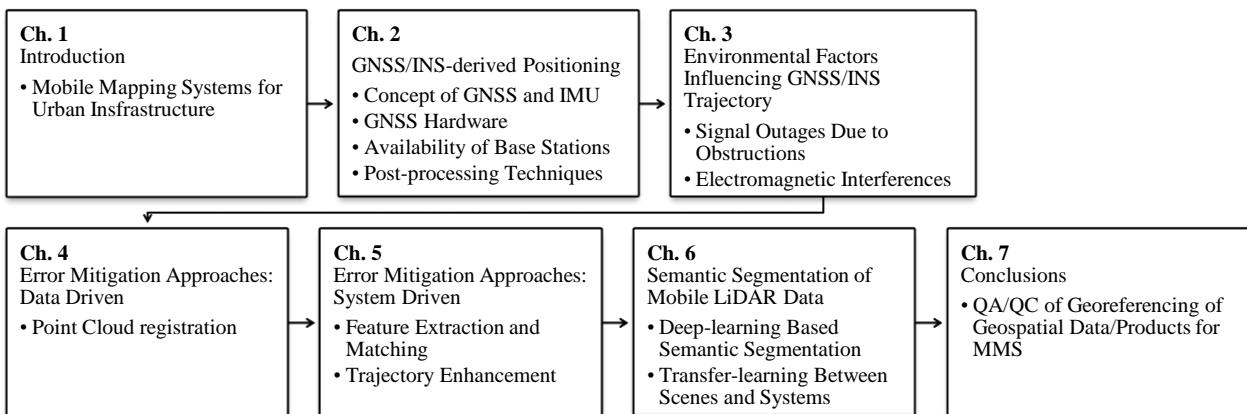
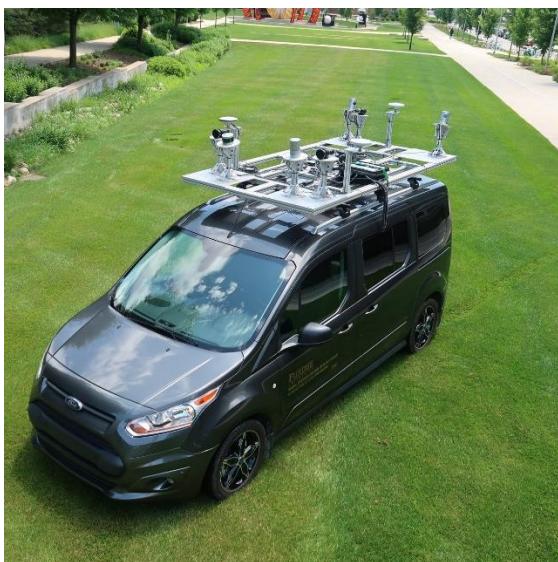


Figure 1-1 Summary of the report

Mobile Mapping Systems

Conventional surveying to map urban infrastructure such as bridges and roadways are time-consuming and require several crews skilled in handling different measurement tools. Recent developments in the field of GNSS/INS-based mapping has generated a lot of interest in leveraging LiDAR technologies that were once limited to static surveys, into broader mobile applications [1–4]. Through a careful hardware/software implementation, LiDAR and RGB sensor data can be directly geo-referenced to a global coordinate system, opening a door to numerous possibilities, including surveys of road networks. Figure 1-2 shows two vehicle-based mobile mapping systems that were developed by our research team with the primary objective of mapping urban infrastructure. The Purdue wheel-based mobile mapping systems – high accuracy and ultra-high accuracy, hereafter denoted as PWMMS-HA and PWMMS-UHA, are outfitted with high accuracy GNSS/INS units that serve to geo-reference an array of LiDAR and RGB sensors onboard the two platforms. Table 1-1 and Table 1-2 list specifications of various sensors onboard the PWMMS-HA and PWMMS-UHA.



(a) PWMMS-HA



(b) PWMMS-UHA

Figure 1-2 Vehicle-based mobile mapping systems used in the study

Table 1-1 PWMMS-HA and PWMMS-UHA GNSS/INS specifications

Platform	GNSS/INS Unit	
	PWMMS-HA	PWMMS-UHA
Sensor model	Applanix POS LV 220	NovAtel ProPak6 IMU-ISA-100C
Positional accuracy	2-5 cm	1-2 cm
Attitude accuracy (pitch/roll)	0.015°	0.003°
Attitude accuracy (heading)	0.025°	0.004°

Table 1-2: PWMMS-HA and PWMMS-UHA LiDAR sensor specifications

Platform	LiDAR Unit			
	PWMMS-HA	PWMMS-UHA		
Sensor model	Velodyne VLP-16 High-Res	Velodyne HDL-32E	RIEGL VUX 1HA	Z+F Profiler 9012
No. of channels	16	32	1	1
Max. range	100 m	100 m	135 m	119 m
Sensor weight	0.830 kg	1.0 kg	3.5 kg	13.5 kg
Range accuracy	± 3 cm	± 2 cm	± 5 mm	± 2 mm
Horizontal FOV	360°	360°	360°	360°
Vertical FOV	+10° to -10°	+10° to -30°	NA	NA
Pulse repetition rate	~300,000 pt/s (single return)	~695,000 pt/s (single return)	~1,000,000 point/s	~1,000,000 point/s
Horizontal beam divergence	3 mrad	3 mrad	0.5 mrad	0.5 mrad
Vertical beam divergence	1.5 mrad	1.2 mrad	0.5 mrad	0.5 mrad
Horizontal laser footprint @ 30 m	9.0 cm	9.0 cm	1.5 cm	1.5 cm
Vertical laser footprint @ 30 m	4.5 cm	3.6 cm	1.5 cm	1.5 cm

This study utilized the two platforms in numerous ground surveys to acquire LiDAR and RGB sensor data as well as to understand the limitations of GNSS/INS navigation throughout those missions. First, to provide some background on GNSS/INS navigation, the following chapter presents the concept of GNSS/INS georeferencing and the expectations from using them in practical land-based surveys.

2. GNSS/INS-DERIVED POSITIONING

In mobile mapping applications, accurate georeferencing of collected data is essential to ensure that the derived products are correctly referenced to a global coordinate system. In this regard, a GNSS/INS-based direct georeferencing is a valuable tool for mobile surveys. In principle, GNSS/INS navigation is the localization of a mapping platform in a global coordinate system using GNSS receivers assisted by inertial navigation systems (INS). A GNSS receiver produces intermittent position information in a global coordinate system. However, it is not precise in the short-term and lacks orientation information. The INS consists of one or more inertial measurement units (IMU) placed orthogonally to each other. These IMU units are capable of measuring 3D linear acceleration and angular velocity which when combined across all IMUs can produce incremental position and orientation updates. Nevertheless, since such computations involve mathematical integration of raw measurements, the resulting position/orientation values have errors that drift with time. When working in tandem, IMU supplements GNSS receivers whenever there are no position updates and GNSS keeps IMU drifts at check. Eventually, the GNSS position updates are coupled with the INS's incremental measurements to produce a refined trajectory. This is the fundamental principle of GNSS/INS navigation. Figure 2-1 shows the basic mechanism of this process.

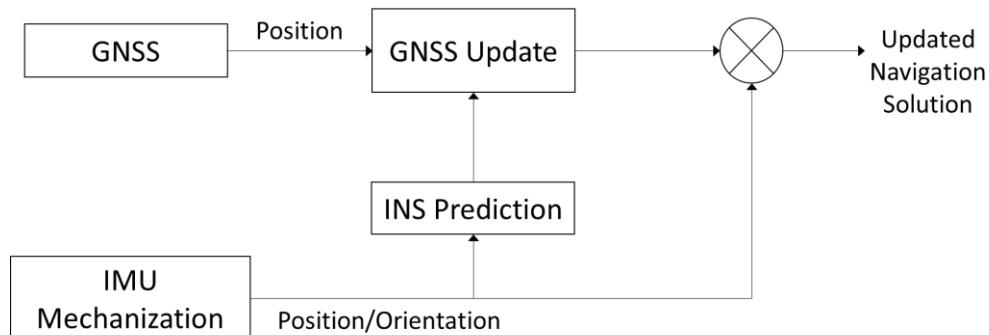


Figure 2-1 Working mechanism of GNSS/INS navigation

2.1 Components of GNSS/INS Navigation

2.1.1 GNSS Receiver-Satellite Infrastructure

The GNSS infrastructure refers to the key elements of GNSS positioning - from equipment at the end-user side consisting of an antenna and a receiver, to various satellite constellations that broadcast coded navigation data on multiple frequency bands. Figure 2-2 illustrates all the major components of a GNSS navigation system. The satellite constellations were originally established by organizations in various countries to fulfill their military and strategic needs. However, a gradual civilian access to certain frequency bands allowed manufacturers to develop multi-constellation capable receivers, which improved the reliability and wide-scale adoption of these navigation systems. Table 2-1 lists widely used satellite constellations supported by modern GNSS receivers.

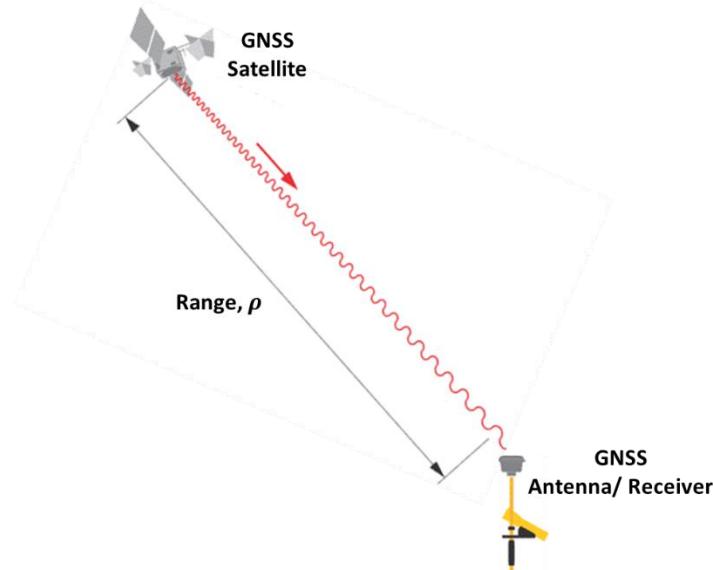


Figure 2-2 Principal components of GNSS navigation [5]

All the carrier frequency bands were chosen in the range of 1175 – 1610 MHz (except for the S-band) to minimize the impact of ionospheric delay and to simplify antenna design. The L1, E1, and B1 are high frequency carrier bands in the range of 1559 – 1610 MHz whereas all the rest are between 1175 – 1280 MHz (except for the S-band which has a frequency of about 2492 MHz). The various allotment of carrier bands evolved over time based on bandwidth and speed requirements. For example, the L5 band was added to the existing GPS L1 and L2 to improve satellite navigation accuracy and robustness in certain civilian applications such as aviation.

Table 2-1: Satellite constellations available for GNSS navigation

Satellite constellation	Year fully operational	Coverage	Carrier frequencies	Country of Origin
GPS	1993	Global	L1, L2, L5	USA
GLONASS	1995	Global	L1, L2, L3	Russia
Galileo	2021	Global	E1, E5, E6	European Union
BeiDou	2020	Global	B1, B2, B3	China
QZSS	2024	Regional	L1, L2, L5, L6	Japan
IRNSS/NavIC	2018	Regional	L5, S	India

2.1.2 Inertial Measurement Unit (IMU) and Inertial Navigation System (INS)

IMU are inertia-based sensors that detect change in its state when acted upon by external forces. These are the primary instruments used in INS and mainly consists of three accelerometers and three gyroscopes. An accelerometer outputs linear acceleration or velocity increment, whereas gyroscope outputs angular velocity. The accelerometer measurements also include gravitational components which are accounted for during mathematical calculations. Figure 2-3 depicts a 6-

degree of freedom (DOF) IMU illustrating the axes of sensitivity for the linear and rotational measurements.

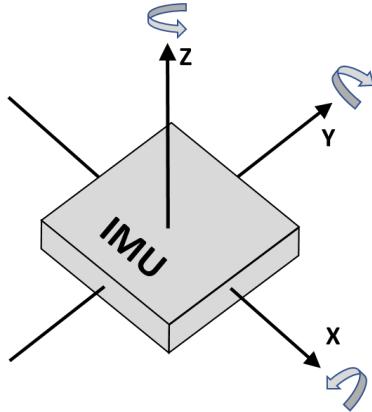


Figure 2-3 A 6-DOF IMU with accelerometer/gyroscope axes

An INS is a navigation system that uses IMU measurements to project a changing motion onto a computational frame of reference. It can also be implemented in a closed-loop algorithm to limit errors in position, velocity, and orientation. Nonetheless, an independently operating INS accumulates error over time due to sensor bias instability and other indeterministic error components. There are mainly two types of INS – gimballed and strapdown systems. The gimballed INS consists of three gyroscopes and three accelerometers attached to a gimbal system whose three orthogonal axes of rotation define the roll, pitch, and yaw of the platform on which the INS is mounted. The independently-rotating gyroscopes cancel their tendency to twist on vehicle's rotation and the mechanism allows the recording of the roll, pitch, and yaw values directly at the bearing of the gimbals. Separately, the accelerometer readings are double-integrated by the onboard computer to continuously output the current position of the platform. The main drawback of the gimballed INS is that it requires very high precision machined parts that are difficult to manufacture. Moreover, they are historically known to suffer from the problem of gimbal lock. Figure 2-4 shows a gimballed IMU developed for the Apollo spacecraft.

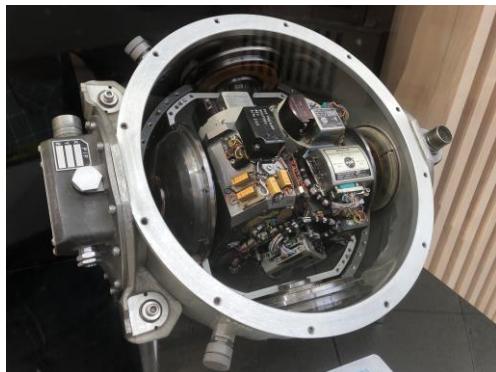


Figure 2-4 Apollo program IMU on display at Draper Labs, Cambridge, MA [6]

The strapdown INS is relatively simple in its construction and is most common nowadays. In its implementation, accelerometers and gyroscopes are mounted directly to the body of the platform. Measurements from the accelerometers are double integrated and combined with gyroscopes' instantaneous turning rate to output platform's position and velocity. Figure 2-5 shows the working mechanism of the strap-down INS.

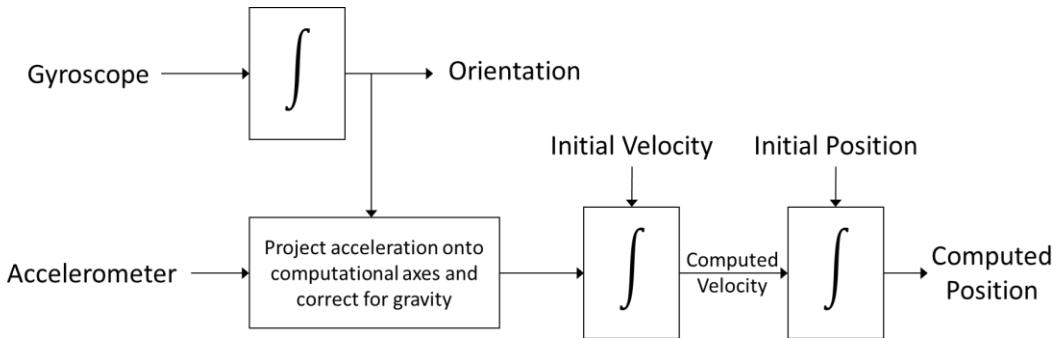


Figure 2-5 Working mechanism of a strapdown INS

In GNSS/INS navigation, INS plays an important role when dealing with periods of GNSS inactivity. The high data-rate of the INS output fills in the gap between successive GNSS updates. On the other hand, accurate GNSS updates limit error growth of the INS in the long-term. This way, the two subsystems complement each other toward a common objective. Table 2-2 enumerates the pros and cons of independent GNSS and INS systems.

Table 2-2 Pros and cons of GNSS and INS systems

	GNSS	INS
Pros	<ul style="list-style-type: none"> 1. Highly accurate in long-term 2. Errors are bounded 3. Relatively inexpensive 	<ul style="list-style-type: none"> 1. Highly precise in short-term 2. Very high data rate 3. Self-contained system or autonomous 4. Provides both position and orientation
Cons	<ul style="list-style-type: none"> 1. Prone to external obstructions/interferences 2. Low data rate 3. No orientation information 	<ul style="list-style-type: none"> 1. Unbounded errors that lead to drifts over time 2. High-performance IMU can be very expensive

Various grades of IMU exist based on its technology and intended application. The grades in-turn drive the cost of IMU units - typically, the cost of an IMU depends heavily on the quality of gyroscope used in the enclosure. Moreover, a commercial IMU module contains more than just the accelerometers and gyroscopes. Auxiliary sensors such as magnetometer and temperature transducers are useful in determining sensor orientation and in modeling temperature-dependent variations, respectively, and can therefore add to the total cost of the unit. From gyro technology point of view, an IMU can be based on one of the following.

- Silicon micro-electro-mechanical systems (MEMS)
- Fiber optic gyro (FOG)-based
- Ring laser gyro (RLG)-based
- Quartz MEMS

The Silicon MEMS IMUs are based on mass deflection principle and are a popular choice in low-cost applications as they can be fabricated into miniaturized modules. These IMUs, however, have higher bias instability and less reliable error characteristics compared to the other three types. The FOG-based IMUs detect motion by comparing light propagation times in different coils. The propagation time changes according to the rotation of the IMU, which is then encoded into angular rates. These are high-performance IMUs that are vibration and shock-resistant and have very good sensor characteristics in terms of bias instability and other error characteristics. The RLG-based IMUs are similar to FOG, but instead of fiber optic coils, RLG uses reflective mirrors in a sealed enclosure. These IMUs are considered the most accurate as well as the most expensive among all four types. Lastly, the quartz MEMS IMUs use vibrating quartz to sense motion by detecting variation in its amplitude. These IMUs feature small size, weight, and power (SWaP) and are highly reliable over a large range of temperature, making them suitable for space applications. In several tests, quartz MEMS IMUs have been shown to outperform RLG IMUs. Figure 2-6 shows examples of each of the above IMU types.



Figure 2-6 Types of IMU (a) Silicon MEMS, (b) FOG, (c) RLG, (d) Quartz MEMS [7–9]

Similar to gyro-based classification, IMUs may use one of the following accelerometer technologies, though they are not limited to these.

- Piezoelectric
- Piezoresistive
- Capacitive

Piezoelectric and piezoresistive accelerometers use a piezoelectric material, such as quartz, to measure acceleration as a change in electrical charge and resistance properties, respectively. Capacitive accelerometers detect changes in the capacitance of a moving plate with respect to a fixed one. Other less popular accelerometer types include strain gauge, fiber optic, and Hall effect accelerometers, to mention a few.

From the overall performance, IMUs can be classified into the following grades.

- Consumer
- Industrial
- Tactical
- Navigation and military

The consumer-grade units are low-cost MEMS IMU that are used in day-to-day electronics, like cell phones, cameras, and gimbals. Industrial-grade IMUs are built for higher vibration resistance and repeatability over time and temperature. Tactical grade IMUs have a very low bias instability which can be from a high-performance MEMS or FOG-based IMU. The navigation grade IMUs have extremely low in-run bias instability, allowing their use in autonomous systems such as submarines and satellites. RLG IMUs can achieve navigation-grade performance [7]. More recently, a similar performance has been claimed using a MEMS IMU [10].

Table 2-3 shows the classification of various GNSS/INS components of the PWMMS-HA/UHA systems discussed in Chapter 1. For land-based mobile mapping systems, a minimum of tactical-grade IMU capable of measuring rapid changes in vehicular dynamics is crucial for consistently precise georeferencing. The quality requirement pertains to both the position as well as the orientation of the mapping platform. Having said so, while IMUs are designed as self-contained navigation systems, GNSS receivers rely on both internal and external factors that collectively facilitate GNSS positioning, as will be discussed in subsequent sections.

Table 2-3 PWMMS-HA/UHA GNSS/INS component classification

	PWMMS-HA	PWMMS-UHA
GNSS constellations supported	GPS+GLONASS +GALILEO+BeiDou	GPS+GLONASS
Gyro technology	-	FOG
Accelerometer technology	-	MEMS
IMU grade	Tactical	Tactical

Note: PWMMS-HA IMU technology information not available

2.2 Working Principle of GNSS Navigation

An easy way to understand GNSS navigation is by comparing it with trilateration. In trilateration, in order to find the position of a fixed point with respect to a reference, distances between control stations and the point of interest are measured and then used to compute that point's relative position. Similarly, in the case of GNSS navigation, distances between a point on Earth and GNSS satellites orbiting above the Earth's surface are measured and the position of that point on the Earth is estimated. Theoretically, three satellites are sufficient to compute the 3D position of a point. In reality, however, four satellites are needed to obtain an accurate position estimate where the fourth satellite enables time-related corrections. Nonetheless, more satellites are preferred for higher redundancy of the measurement data. Figure 2-7 shows the GNSS positioning technique drawing similarities to the conventional trilateration.

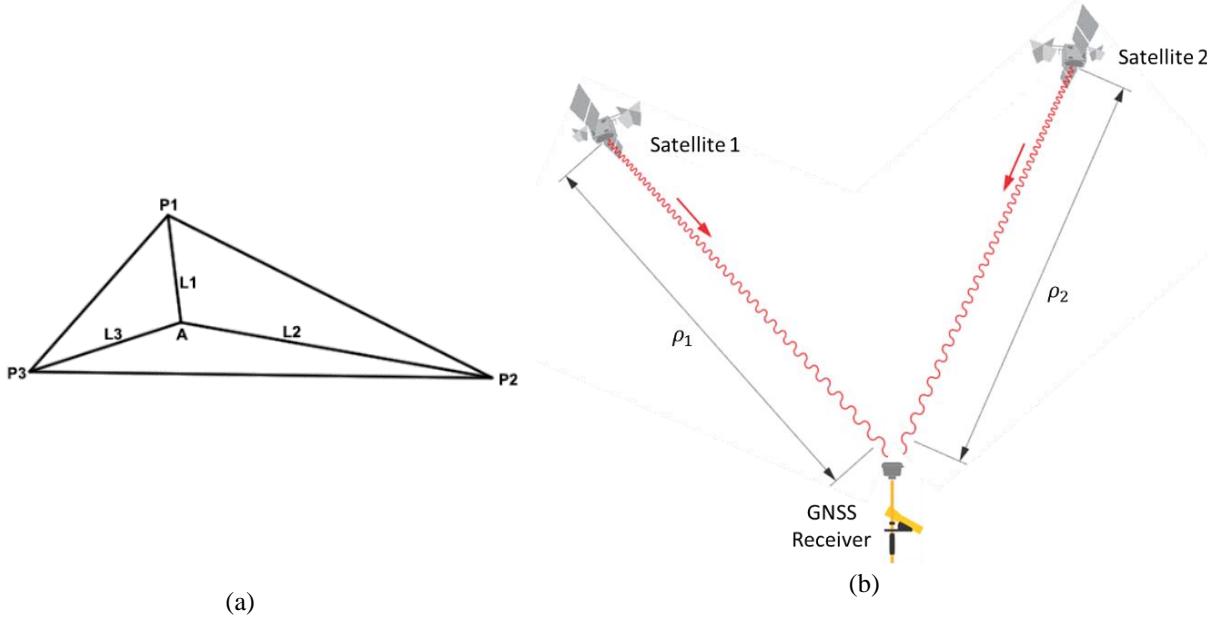


Figure 2-7 (a) Conventional trilateration using control points (P1-P3) to locate a fixed-point A; (b) GNSS positioning with satellites as control points [5]

2.2.1 Measurement Observables

When working with GNSS signals for positioning, usually two types of measurements are encountered – code-phase and carrier-phase observables. The code-phase observables consist of pseudorandom code (PRC), also known as civilian access (C/A) code, modulated onto the carrier signals. PRCs received from satellites are correlated with a receiver replica, and the time-offset that results in the highest correlation is then used to calculate a pseudorange. This satellite-to-receiver pseudorange can be used to quickly compute a point position in the 3-dimensional space. Such solutions are useful in applications where low-accuracy positioning is sufficient.

The carrier-phase observables are based on the GNSS carrier signals that have a very high frequency compared to the bitrate of pseudorandom code (1.5GHz vs. 1MHz, respectively). Thus, correlating carrier-phase observables can provide a more accurate measure of time-offset. The downside of using carrier-phase observables is that it requires longer time to produce position estimates since carrier signals are hard to correlate compared to pseudo-random codes. Figure 2-8 shows the characteristics of pseudorandom code and carrier signals, along with time-offset computation for the former.

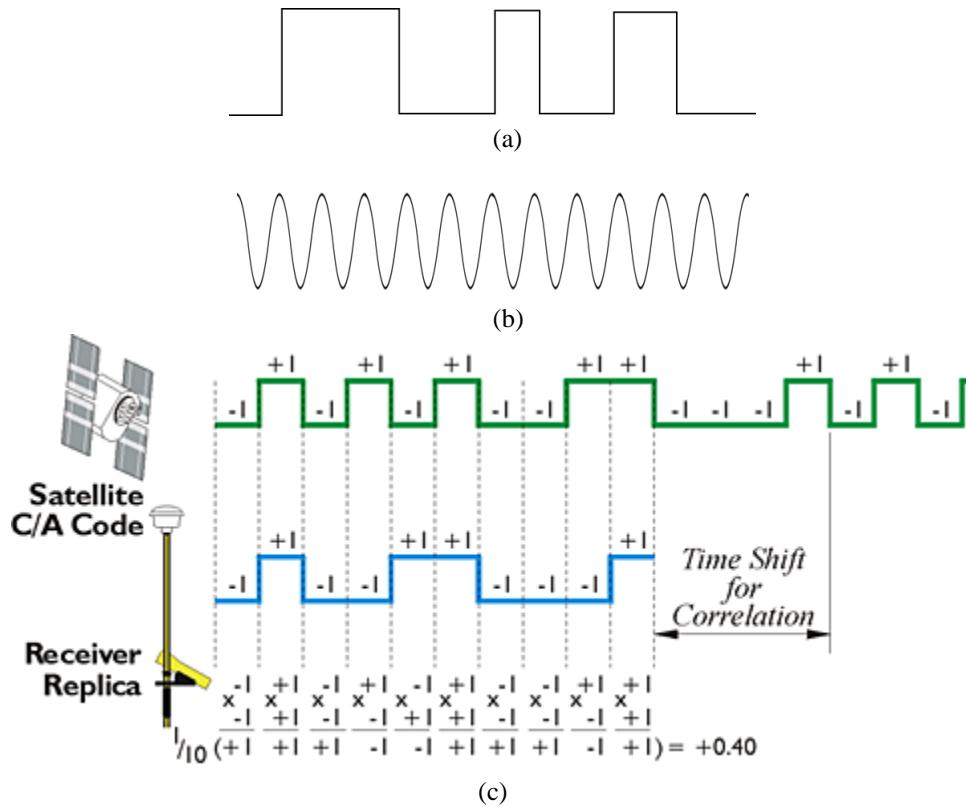


Figure 2-8 GNSS observables measured by a receiver. (a) Square-wave pseudorandom code, (b) Sinewave GNSS carrier signal, and (c) Time shift between PRC and receiver's replica [5]

2.2.2 Autonomous and Relative (differential) Positioning

The way GNSS-positioning works, each satellite broadcasts a unique PRC and a navigation message containing satellite's position almanac along with coarse positions (ephemeris) of other satellites. They also include clock offsets and atmospheric corrections. The GNSS receiver then solves for its 3D position and receiver's clock by computing pseudorange between the receiver and individual satellites. This allows for a real-time positioning of the receiver, often called autonomous single-point solution. Mathematically, the pseudorange (p) between the receiver and a satellite is given by Equation 2-1.

$$p = \rho + c(dt - dT) + \text{errors} \quad \text{Equation 2-1}$$

Where, ρ is the true range, c is the speed of light, dt is satellite clock offset, and dT is receiver clock offset. Note that the accuracy of such position estimates is low, in the order of several meters, due to various errors in measurement and clock offsets.

Another positioning method different from single-point solution is relative or differential positioning. Unlike autonomous positioning, where a receiver must rely only on available navigation messages, in differential positioning, a second receiver is used in conjunction with the

first, where one of them is fixed at a given position. If the two receiver units have a very short baseline compared to the satellite-to-receiver distance of over 19,000 km, both experience similar error due to atmospheric effects and thus have a high correlation between their measurements. Using the fact that the base receiver's position is fixed and known, the correlation between the two can be used in correcting any systematic biases or errors present in the rover receiver, hence improving its position accuracy. Such position estimates are often accurate to within a decimeter or so. A pseudorange-based differential processing produces better position estimates than single-point positioning, in the range of a decimeter. On the other hand, differential processing with carrier-phase observables can result in higher accuracies, often under a few centimeters. Through multi-frequency differential processing with a very short baseline between the receiver and a base station network, it is even possible to approach sub-centimeter accuracies. Generally, differential processing is referred to using carrier-phase observables. A schematic of differential positioning is shown in Figure 2-9.

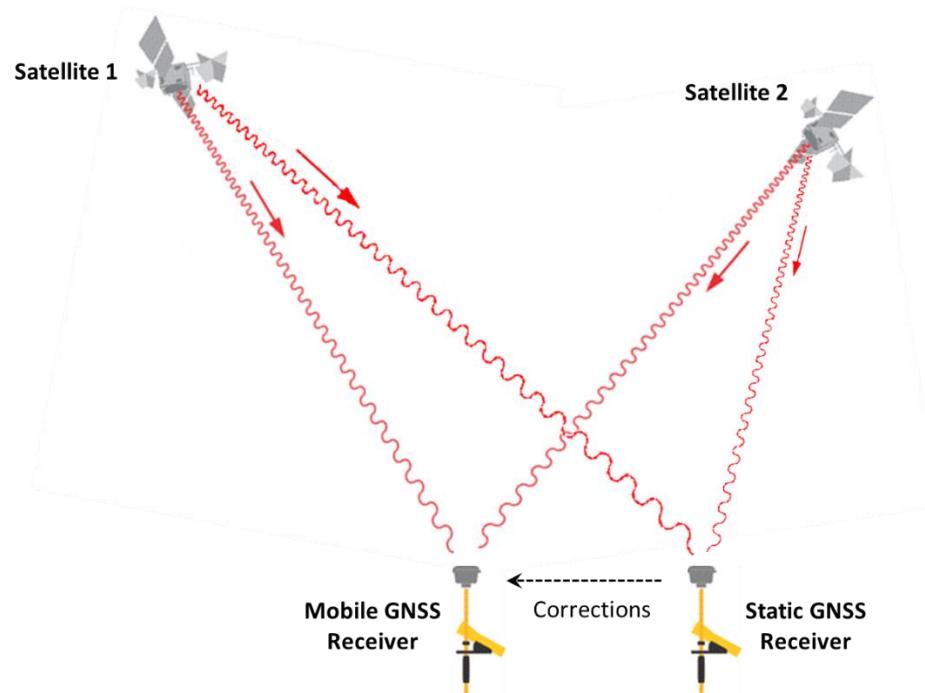


Figure 2-9 Differential GNSS positioning using two receivers [5]

In practice, GNSS receivers are designed to first use code-phase positioning techniques before transitioning to the use of a more precise carrier-phase measurements. Solving for code-phase makes the problem of matching carrier wavelengths more feasible. Additional information about GNSS satellites, such as precise orbit and clock files obtained from ground tracking stations facilitate these processes. With newer and faster processors onboard GNSS receivers, this problem of signal correlation is becoming more tractable than ever. Considering various applications, GNSS positioning is conducted through three popular techniques.

- Precise-point positioning (PPP): This is an autonomous positioning technique that only uses receiver-acquired measurements containing broadcast ephemeris and clock corrections. The

computed positions exhibit error in meters, however, in other cases, separate precise ephemeris and clock data can be obtained from ground tracking stations and added during data processing to reduce the error. Recent developments have led to even more precise orbit and clock corrections based on long-term satellite observations. Despite this, PPP is rarely used in mobile mapping applications and are usually implemented as a pre-processing check for data integrity.

- Real-time Kinematic (RTK): It is a real-time positioning technique that is based on the concept of differential positioning. The idea is to setup a communication channel between the rover receiver and nearby base stations. These base receivers calculate differences in their actual and observed positions and communicate those to the rover for an appropriate error correction. Such corrections often result in centimeter-level position accuracy. This happens in real-time and no further processing is needed. This is the main advantage of RTK surveys. The downside is that receiver noise or multipath issues cannot be handled by this technique.
- Post-processed Kinematic (PPK): This is the most widely used positioning technique used in mobile mapping surveys. It is also based on differential positioning. In this post-processing technique, error corrections are obtained from base receivers and applied to improve the position estimate of the rover receiver. An appropriate configuration of base stations is necessary to achieve an optimum performance. This includes presence of base stations in all directions around the rover. The baseline length is also an important factor when correlating signals perturbed by atmosphere, as discussed earlier. As a rule of thumb, baseline of up to 30 km are considered optimal.

2.3 System-level Factors Influencing the Quality of GNSS/INS Trajectories

Different applications have different accuracy requirements for mapping data. For infrastructure and land surveys, a differential post-processed GNSS/INS solution is preferred over autonomous single-point positioning owing to former's proven record of consistently producing high-accuracy position estimates. However, in order to maintain such a high level of performance throughout the mission, high-grade GNSS/IMU hardware systems and an efficient post-processing technique, among several other variables, are necessary. The following subsections describe in detail some of the major factors that influence the quality of GNSS/INS-derived trajectories obtained from land-based mobile mapping platforms.

2.3.1 Selection of GNSS hardware components

Assuming the inertial unit (IMU) used for navigation has already been selected to be of a project-specific grade, the succeeding requirement for an effective navigation is then governed by the quality of the GNSS sensors used, which mainly consist of an antenna and receiver pair. For a highly accurate differential positioning, a GNSS antenna must support multiple operating frequency bands as defined for various navigation satellite constellations. The shape, size, weight, cost, and sensitivity of an antenna primarily depend on its noise rejection and multi-constellation capabilities. For example, antennas used on unmanned aerial systems are usually light in weight, and since they are operated far from any overhead obstacles, do not need a sophisticated noise dissipation. On the other hand, antennas mounted on vehicles close to the ground or those used for geodetic surveying must maximize the signal-to-noise ratio by rejecting unwanted signals

bouncing back from nearby obstacles (called *multipath* effect). Achieving such characteristic requires a larger antenna enclosure to house separate specialized components, thus increasing the size, weight, and cost of the unit. Figure 2-10 compares GNSS antennas used in some popular applications.



Figure 2-10 GNSS antenna used in various applications. (Left to right) Antennas used on a UAV, land-based vehicle, and for geodetic surveying, respectively. All are from the same manufacturer: Trimble [11].

The multi-frequency, multi-constellation signals acquired by an antenna are analyzed by the receiver unit to compute raw pseudorange and doppler values, as well as a real-time trajectory for coarse positioning. For the receiver, its data throughput, capacity to handle multiple constellations, and an ability to integrate with other sensor types such as cameras/LiDAR for time synchronization purposes are what determine the cost of that unit. GNSS receiver units used for non-survey applications have fewer capabilities. For example, low-cost receivers in consumer electronics do not have the processing power and bandwidth to store raw GNSS data essential for differential post-processing. For their intended use, low accuracy position estimates are sufficient. At the same time, low-end receivers may not have the ability to time-tag or log sensor events due to limited computing resources. Modern high-performance GNSS receivers have a built-in precision oscillator which when synchronized with the GNSS clock can produce timing signals accurate to within a few nanoseconds. Such high precision is critical in mobile mapping applications to capture the rapidly changing surrounding dynamics through accurate time-synchronization of GNSS/INS, imagery, and LiDAR data.

2.3.2 Processing Scheme

The GNSS/INS post-processing involves the use of data filter (such as Kalman filter) to update INS predictions every time a GNSS solution is available. For better convergence and high accuracy, a differential post-processing is conducted in forward and reverse directions and the solutions are combined. Doing so also enables the quality control of resulting trajectory. Depending on the base station availability and satellite geometry, the forward and reverse solutions may achieve different results – fixed or float, in different parts of the mission. Fixed solution refers to successful correlation of the carrier-phase signals whereas float refers to code-phase solutions. Any difference in the forward and reverse solution can be observed on a combined separation plot, that indicates the difference in position estimates at any moment of the survey [12].

In addition to the processing direction, the way a data filter is used with the GNSS and INS data can also impact the resulting solution. Usually, two types of processing modes are implemented – Loosely coupled (LC) and tightly coupled (TC) processing. In the LC processing mode, GNSS data is processed first and then a Kalman filter is initiated for the INS to use the

already processed GNSS updates whenever they are available. On the other hand, TC processing mode utilizes both the GNSS and INS raw data in a data fusion strategy where each measurement is first used to predict and update the other, and then they are combined. The TC processing method can produce a highly precise result and is tolerant to slight inconsistencies in input GNSS measurements. For this reason, in most applications including land-based surveys, the TC processing mode is preferred.

Along with the above processing techniques, several other parameters also facilitate an efficient GNSS/INS post-processing – parameters such as satellite/constellation selection, signal-to-noise ratio cutoff, satellite elevation mask, velocity constraints, and ionospheric error computation threshold can be methodically tested to obtain the most suitable combination of settings.

2.3.3 Availability of Continuously Operating Reference Stations (CORS)

As discussed earlier, mapping surveys typically engage differential processing with base station receivers to obtain highly precise position estimates. These base receivers can be any static GNSS receiver whose position is fixed and known. In practical applications, having multiple base receivers improve data redundancy and coverage area in case missions spanning a large distance. To facilitate various public projects, departments of transportation (DOT), such as Indiana's INDOT, operate a network of strategically placed reference base stations. Referred as continuously operating reference stations (CORS), these base stations can be used in GNSS-based differential positioning as well as for real-time kinematic (RTK) surveys. To reintroduce the concept of differential positioning, raw GNSS data acquired from a mission is compared with the corresponding raw data from a local network of base stations, and any error present in the surveyed data is then mathematically corrected by evaluating the difference between the two sets. For an accurate error differencing, a well-distributed network of base stations is crucial. This is particularly important for missions that cover large ground distances since longer baselines in any direction may result in a less accurate error correction. Figure 2-11a shows the CORS network managed by INDOT and other local agencies with their nominal coverage over the state of Indiana. From a usage point of view, nominal coverage maps can be slightly misleading, since they do not indicate the actual level of precision expected in various parts of the region. A simple simulation model developed from several hour-long GNSS data reveals how the actual differential positioning performs when a particular level of precision is desired. In this process, a pair of base stations were simulated as a base-rover pair and a static processing of GNSS data was performed. Then, the baseline length error, i.e., the difference between true and estimated distances between stations, was computed. From this error, ranges for specific error levels were obtained and a contour connecting all such ranges from each pair around the base station was plotted. Figure 2-11b demonstrates the process with two base stations showing their effective coverage for a position accuracy of 1 cm. Figure 2-11c shows the same for all base stations. The figure also indicates all the blind spots left out by the insufficiency of base stations despite their nominal larger coverage. With greater utilization of GNSS-based surveys, it is therefore becoming important that the strategy of CORS network implementation is revisited and, if need be, newer base stations are established to fill gaps in the current structure.

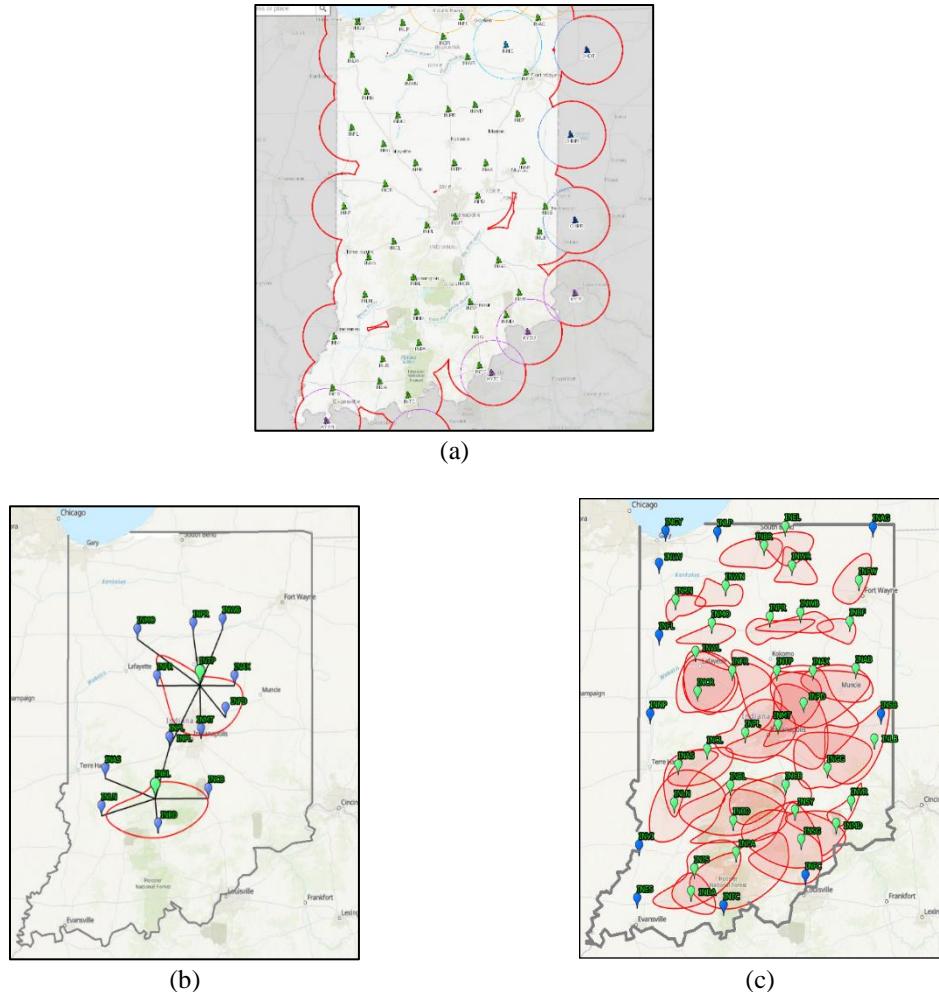


Figure 2-11 (a) CORS network within and around Indiana with their nominal coverage. (b) Effective coverage evaluated for two separate base stations, and (c) A simulation of the effective coverage of base stations to achieve a position accuracy of 1 cm.

3. ENVIRONMENTAL FACTORS INFLUENCING GNSS/INS-DERIVED TRAJECTORY ACCURACY

Unlike aerial data acquisitions, land-based mobile mapping surveys have their peculiar challenges. One of them is obstructions to GNSS signals reception due to man-made or natural structures. In addition, GNSS signals may also be disrupted due to electromagnetic interferences. The following subsections describe these challenges in detail.

3.1 Signal outages due to physical obstructions

Any partial obstruction of the sky prevents a clear line of path between the antenna and GNSS satellites. This immediately lowers the reliability of GNSS/INS positional estimates due to loss of data redundancy and/or multipath errors. Furthermore, in the event of a complete signal outage, such as under a plant canopy or bridge, the position estimation eventually switches to a complete inertial system-based predictive positioning. While inertial measurements somewhat compensate for short-term outages, the accuracy drastically degrades with time, to the point that it can be no longer relied upon. Figure 3-1 illustrates the operation of land-based mapping systems for agricultural and urban environments. When objects around these platforms, such as plants and buildings, extend beyond the antenna height, some or all of the useful GNSS signals are blocked. Effectively, these two scenarios lead to the same result, i.e., irrespective of the cause of obstruction, the resulting trajectory is affected in the same way.

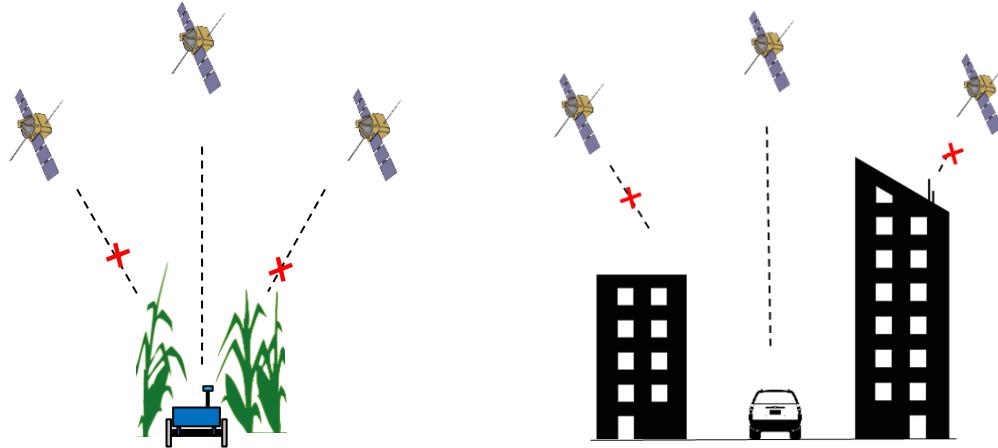


Figure 3-1 GNSS data acquisition in agricultural and urban environments. Except for a few differences, both scenarios are analogous.

The issue of signal obstruction leading to poor quality trajectory can be illustrated with real-world examples. Figure 3-2 shows two mobile mapping systems – a backpack-based and an unmanned ground vehicle (UGV)-based platforms, that were developed in-house for a variety of mapping applications [13]. Both systems have a high accuracy integrated GNSS/INS unit (NovAtel SPAN-CPT and SPAN-IGM S1, respectively) for direct georeferencing. The acquired data is post-processed through the LC processing technique.



Figure 3-2 Backpack and UGV-based mobile mapping platforms

Figure 3-3a demonstrates an example use-case of these two platforms in an agricultural environment. Looking at the corresponding plots in Figure 3-3b and Figure 3-3c that show GNSS satellite availability for each platform, one can tell that while the backpack system experienced occasional signal outages, the UGV system suffered from a frequent loss of signals, notably for prolonged durations. Figure 3-4 shows the combined separation plots for these two missions. Evidently, the UGV trajectory has much wider separation between the forward and reverse trajectories compared to the backpack. This means resulting mapping products will have an equivalent amount of variation (i.e., the UGV trajectory will be inferior to that from the backpack).

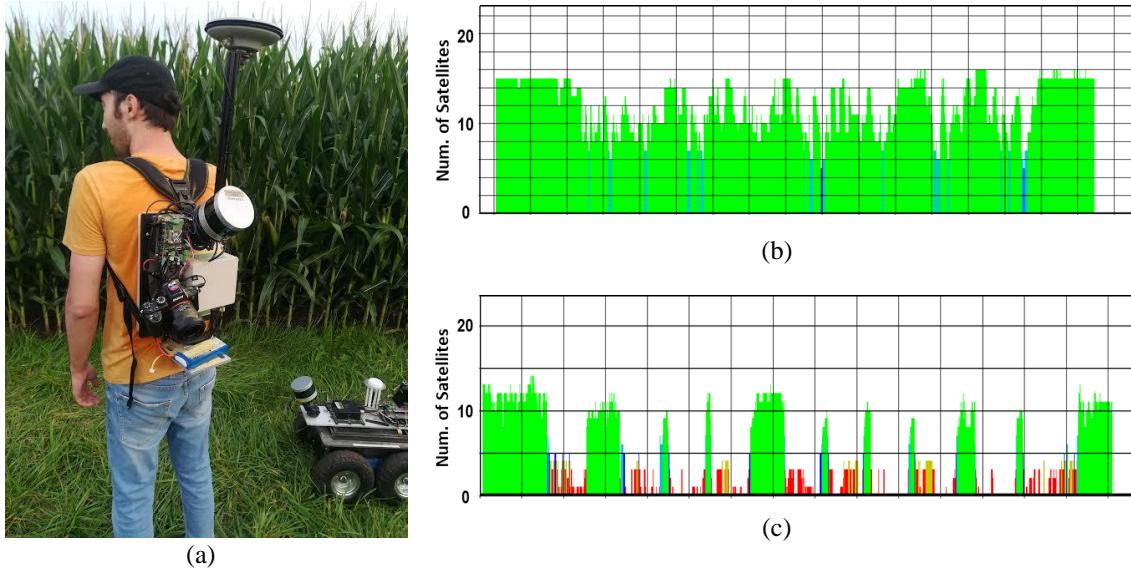


Figure 3-3 (a) Data acquisition in an agricultural field, (b) Satellite availability for backpack system, and (c) Satellite availability for unmanned ground vehicle (UGV).

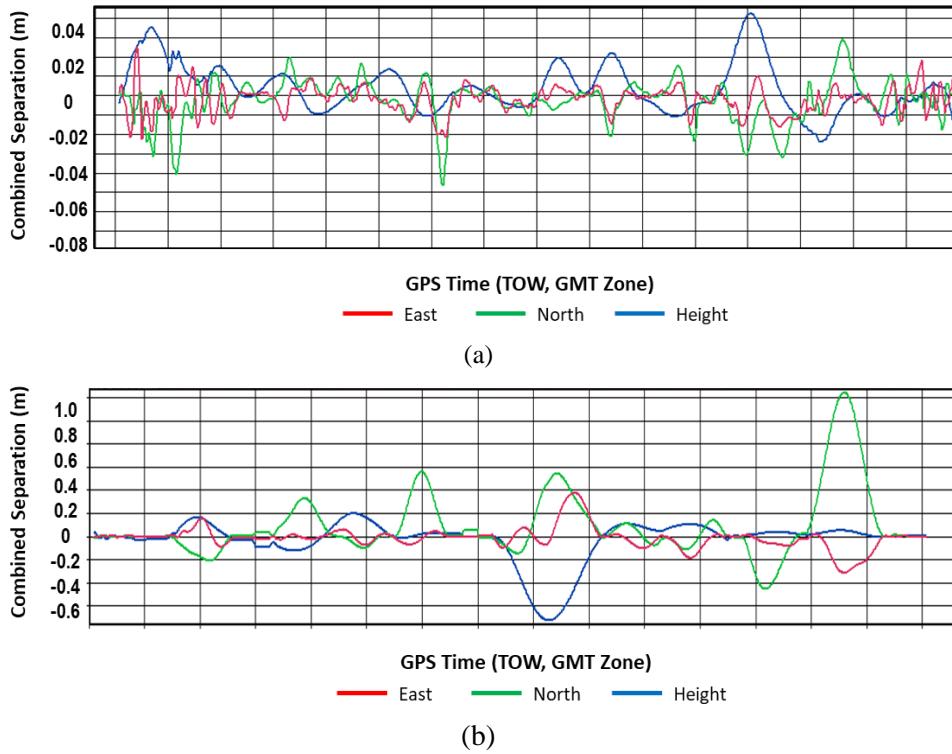


Figure 3-4 Combined separation plot for (a) Backpack trajectory, and (b) UGV trajectory

This relatively poor performance of the UGV GNSS system can be attributed to the shorter height of its antenna. Such obstructions to the GNSS signals result in the reduction of trajectory accuracy, more so for any survey with either backpack or UGV that involves scanning inside building facilities, where a complete signal outage is expected.

The vehicle-based mobile mapping systems, PWMMS-HA and PWMMS-UHA, were utilized to map highway and its surrounding infrastructure like bridges and traffic signs. As with the backpack and UGV systems mentioned above, these two vehicle-based mapping platforms experienced similar GNSS signal reception issues within the vicinity of bridges and road-side canopy. Figure 3-5 shows the outcome of one such mission with the PWMMS-UHA. As pictured in Figure 3-5a, canopy cover on both sides of the road is blocking almost the entire sky view. This blockage resulted in loss of GNSS signal lock and thus an increase in position standard deviation, as indicated in Figure 3-5b and c. As an outcome of this occurrence, reconstructed point clouds from the LiDAR units onboard PWMMS-UHA show clear misalignment among tracks from different drive passes visualized in Figure 3-5d for different profiles across the road. Similar to tree canopies, during surveys along road corridors, nearby traffic such as cargos and trucks may also inhibit signal reception. Such instances, although short-lived, cause intermittent loss of GNSS signal and require more time than the actual outage duration to resume an effective positioning.

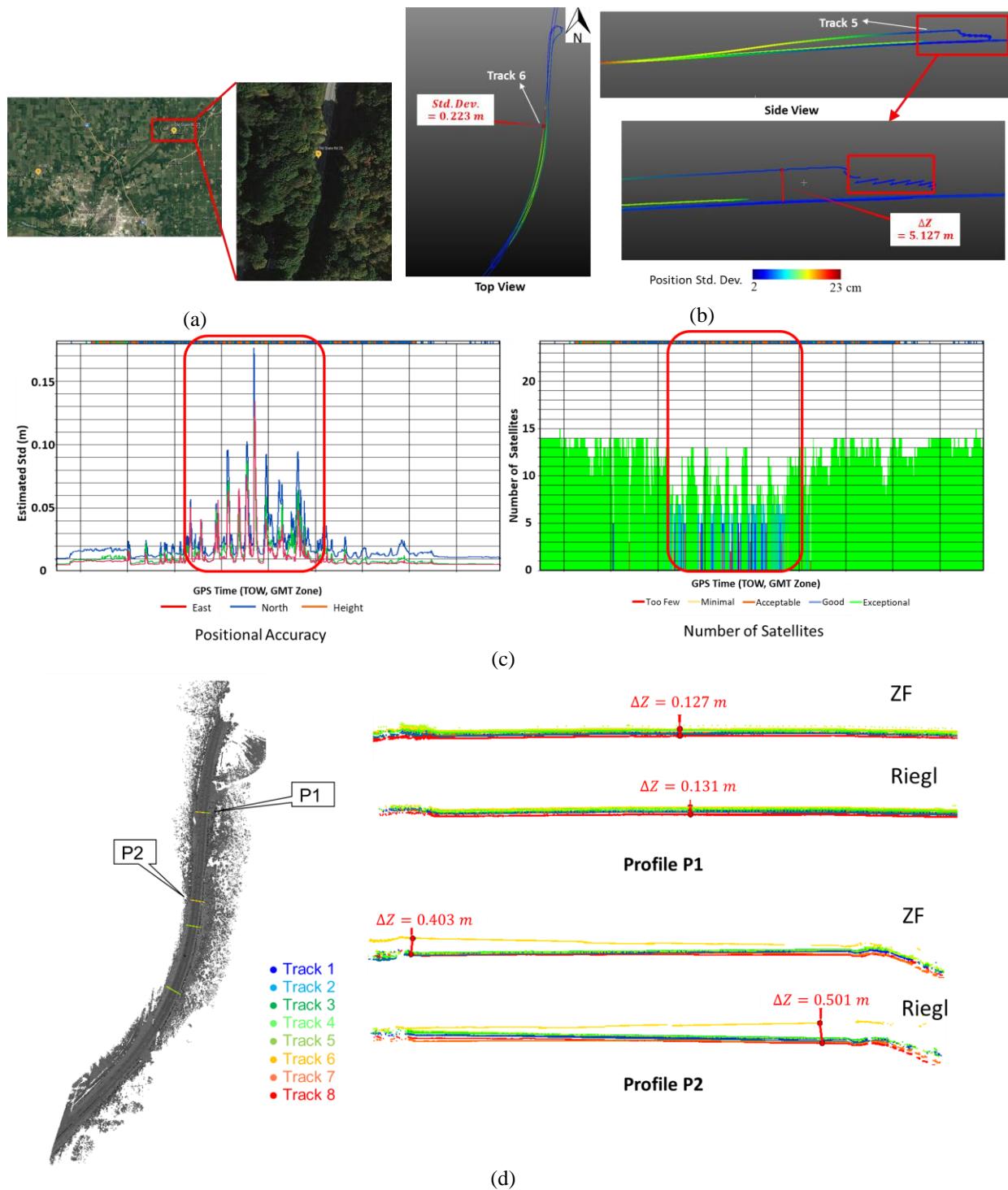


Figure 3-5 Impact of roadside vegetation on GNSS trajectory and LiDAR-derived point clouds. (a) Area showing canopy cover on both sides of the road, (b) Different views of the post-processed trajectory colored by position inaccuracy, (c) Post-processed trajectory standard deviation and satellite availability, and (d) Discrepancies in the reconstructed point clouds from different tracks.

3.2 Electromagnetic Interference

Other than physical obstructions, high amplitude electromagnetic (EM) waves can also disrupt the GNSS signals reception resulting in a similar reduction in the post-processed trajectory accuracy as well as the alignment of derived mapping products. EM interferences can originate from a variety of electronic devices working on radio spectrums, such as LiDAR units and radio transceivers. Additionally, high-tension (high voltage) power transmission lines can also introduce EM perturbations (in some rare circumstances, the GNSS positioning can also be affected by ionospheric disturbances caused by solar flares). Figure 3-6 visualizes an example from PWMMS-UHA where the presence of EM sources near a data acquisition site resulted in erroneous georeferencing. Point clouds from different LiDAR units across different drive runs can be seen as multiple versions of the same object separated by almost a decimeter, which are ideally expected to align within a few centimeters.

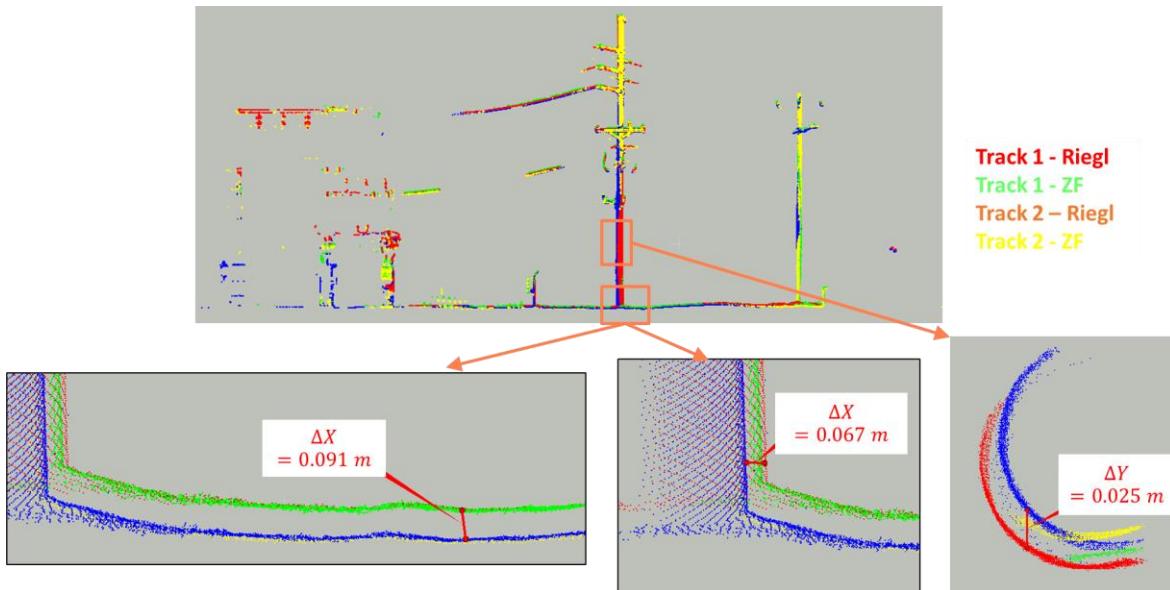


Figure 3-6 EM-induced misalignment in reconstructed point clouds

Thus, while a meticulous system design is crucial to reduce any direct exposure of sensors to unwanted interferences, it is equally important that data acquisition missions are carefully planned to account for external interference sources like power sub-stations and transmission lines.

3.3 Concluding Remarks

All the above-mentioned factors that affect GNSS/INS-based mapping are equally applicable to every ground-based mapping system. The analogy of mapping under plant canopy and inside building facilities can be easily extended to vehicles driven under bridges or on roads along-side tall trees that partially or completely block clear line of path to GNSS signals. Furthermore, the effect of electromagnetic interferences is more applicable to road vehicles since they are expected to frequently pass by power transmission lines and distribution substations. Figure 3-7 summarizes various environmental factors that can impact the quality of GNSS/INS trajectory.



Figure 3-7 Examples of environmental factors that will impact the quality of the GNSS trajectory: Steel overpass, concrete bridge overpass, electromagnetic interference, and heavy canopy cover.

4. PROCESSING STRATEGY AND COMPARATIVE PERFORMANCE OF DIFFERENT MOBILE LIDAR SYSTEM GRADES FOR BRIDGE MONITORING: A CASE STUDY

Abstract

Collecting precise as-built data is essential for tracking construction progress. Three-dimensional models generated from such data capture the as-is conditions of the structures, providing valuable information for monitoring existing infrastructure over time. As-built data can be acquired using a wide range of remote sensing technologies, among which mobile LiDAR is gaining increasing attention due to its ability to collect high-resolution data over a relatively large area in a short time. The quality of mobile LiDAR data depends not only on the grade of onboard LiDAR scanners but also on the accuracy of direct georeferencing information and system calibration. Consequently, millimeter-level accuracy is difficult to achieve. In this study, the performance of mapping-grade and surveying-grade mobile LiDAR systems for bridge monitoring is evaluated against static laser scanners. Field surveys were conducted over a concrete bridge where grinding was required to achieve desired smoothness. A semi-automated, feature-based fine registration strategy is proposed to compensate for the impact of georeferencing and system calibration errors on mobile LiDAR data. Bridge deck thickness is evaluated using surface segments to minimize the impact of inherent noise in the point cloud. The results show that the two grades of mobile LiDAR delivered thickness estimates that are in agreement with those derived from static laser scanning in the 1 cm range. The mobile LiDAR data acquisition took roughly five minutes without having a significant impact on traffic, while the static laser scanning required more than three hours.

4.1 Introduction

Securing bridges with good structural and functional conditions starts with effective quality assurance and quality control during construction to ensure that such infrastructure fully satisfies the design requirements [14]. In addition, frequent and accurate inspection of the structural and functional conditions of each bridge is required to ensure traffic safety and prioritize maintenance [15]. Traditional bridge evaluation practices rely on visual inspection and point-based measurements. Deck thickness is typically checked during and post-construction to ensure structural adequacy and conformance. For instance, the American Association of State Highway and Transportation Officials (AASHTO) requires that the minimum thickness of a concrete deck should not be less than 7 inches [16]. To evaluate bridge deck thickness during construction, measurements are typically taken from a string line pulled between the screed rails or a pole stabbed into the plastic concrete—i.e., freshly poured concrete [17]. Post-construction, ultrasonic thickness meters are usually used. Such approaches are labor-intensive and prone to human errors, as they require trained personnel to identify structurally unsound locations. Furthermore, this spot sampling is not sufficient to capture the thickness values over the entire bridge, which are necessary to provide as-built documentation that can be used for long-term asset inventory and management.

Modern remote sensing techniques provide promising non-contact alternatives for during- and post-construction bridge evaluation and inspection. Prior research has established principles and

procedures for using unmanned aerial vehicle (UAV) imagery for bridge visual inspection and 3D information reconstruction [18–22]. Moreover, with several protection mechanisms, UAVs can maneuver at a close proximity to the bridge and even perform contact inspection tasks [23–25]. In contrast to imaging sensors, light detection and ranging (LiDAR) provides direct 3D measurements that can be used for quantitative evaluation. Terrestrial laser scanners (TLSs) can acquire high-resolution point clouds with millimeter-level precision, deemed as the gold standard in construction management and infrastructure monitoring. However, the time and labor required for data acquisition and post-processing increase significantly for a large site. The scanner might have to be set up within the driving lane to acquire sufficient point density along the structure, which in turn would affect traffic flow. Therefore, monitoring large-sized infrastructure using TLSs might not be practical and/or scalable. Mobile LiDAR mapping systems (MLMSs) facilitate efficient field surveys that can cover large areas with a minimal impact on traffic flow. For MLMSs, direct georeferencing, i.e., trajectory information provided by the onboard global navigation satellite system/inertial navigation system (GNSS/INS) unit, is typically adopted to reconstruct point clouds in a common reference/mapping frame. Point cloud quality therefore depends on the ranging accuracy of LiDAR units, the grade of the onboard GNSS/INS unit, and the reliability of the system calibration procedure. Centimeter-level positional accuracy can be expected if the quality of derived georeferencing data from the GNSS/INS unit and system calibration parameters are guaranteed. However, intermittent access to a GNSS signal and issues pertaining to system calibration cloud results in discrepancy of more than several centimeters between point clouds from different drive runs/flight lines. In this case, registration is required to fine-tune the point cloud alignment.

This paper describes an assessment of alternative mobile LiDAR systems for bridge monitoring. More specifically, this study addresses the following research questions: (i) can we use MLMS data (with centimeter-level positional accuracy) to derive quantitative measures of bridge and achieve an accuracy similar to that from TLS data, and (ii) are surveying-grade MLMS systems (with millimeter- to centimeter-level accuracy) better than mapping-grade MLMS systems (with an accuracy of a few centimeters) in terms of providing quantitative assessments of bridges? The key contributions of this study can be summarized as follows:

- A semi-automated feature-based fine registration is proposed to compensate for the impact of georeferencing and system calibration errors on mobile LiDAR data. The developed registration strategy can also be used to fine-tune the point cloud alignment among different TLS scans.
- A bridge deck thickness evaluation strategy based on surface-to-surface distance is proposed to minimize the impact of inherent noise on the point clouds. The aim is to achieve an accuracy better than ± 1 cm for the derived thickness measures.
- The performance of different MLMS grades is assessed against TLS data in terms of the quality of derived thickness measures, scalability, and impact on traffic flow.

The remainder of this paper is structured as follows: Section 4.2 provides an overview of prior research; the mobile mapping systems, study site, and data collection procedure are introduced in Section 4.3; the feature-based registration and bridge deck thickness evaluation approaches are covered in Section 4.4; the results together with their analysis are discussed in Section , which is

followed by a summary of the study conclusions and recommendations for future research in Section 4.6.

4.2 Related Work

4.2.1 LiDAR for Infrastructure Mapping

LiDAR, known for its ability to directly generate accurate 3D point clouds with high density, has recently been receiving an increasing amount of interest by the construction management and infrastructure monitoring research/professional communities. Acquired point clouds by TLSs have been used for generating precise 3D models to evaluate the progression of construction processes [26–28]. The high accuracy of TLS point clouds also allows for millimeter-level displacement and deformation evaluation post-construction [29–31]. However, the point density and accuracy of TLS point clouds drop quickly as the distance from the sensor increases. To ensure full coverage of the structure in question, multiple TLS stations are required. The station locations should be carefully chosen to minimize occlusions and have sufficient overlap (i.e., common areas) for the registration of point clouds from different scans. For a large site, time, labor, and location requirements for data acquisition increase significantly, making it unrealistic to apply such a technique in a scalable manner.

Mobile LiDAR has emerged as a promising alternative that can overcome the shortcomings of TLSs. One or more LiDAR units can be mounted on various platforms, e.g., UAVs, trucks, tractors, and robots. Field surveys with mobile LiDAR are efficient and can cover large areas that are impractical to conduct with TLSs. These key benefits have stimulated the interest of the research/professional community to apply mobile LiDAR for analyzing complex road environments, such as lane marking detection and road boundary extraction [32–35], as well as mapping railroads and tunnels [36–39]. Several studies have validated and reported the accuracy of mobile LiDAR data for monitoring civil infrastructure. Puri and Turkan [40] used a wheel-based mobile LiDAR system equipped with a Velodyne HDL-64E LiDAR unit for tracking the construction progress of a bridge. They pointed out that a noise level in the range of $\pm 3\text{--}4\text{ cm}$ was present in the as-built data from MLMSs, affecting the performance of progress tracking. Lin et al. [41] evaluated the performance of a wheel-based MLMSs equipped with four Velodyne LiDAR units for mapping airfield pavement before and after a resurfacing process. In their study, the positional accuracy of LiDAR point clouds was in the $\pm 5\text{ cm}$ range, and a 9 cm increase in pavement elevation after resurfacing was detected. Another study verified the absolute accuracy of point clouds acquired by a commercially available system, the Lynx Mobile Mapper system from Optech Inc. (Toronto, ON, Canada) [42]. Their test site was a university campus, comprised of both infrastructure and vegetation. Although the range accuracy of the onboard LiDAR unit in that study was $\pm 8\text{ mm}$, the absolute accuracy of the point cloud was determined to be in the range of ± 1 to $\pm 5\text{ cm}$, mainly attributed to the trajectory quality. Moreover, in areas with limited/intermittent access to GNSS signals, the positional accuracy might deteriorate to the $\pm 0.3\text{ m}$ range. Inaccuracy of the system calibration parameters would cause additional deterioration in the derived point clouds. Overall, millimeter-level positional accuracy is difficult to achieve even with high-end LiDAR units due to GNSS reception issues and/or system calibration artifacts. Data processing and analysis strategies that reduce the impact of the above factors, as well as inherent noise in the point cloud, are required to take full advantage of potential benefits of mobile LiDAR.

4.2.2 Point Cloud Registration

Point cloud registration aims at aligning LiDAR data from different TLS scans and/or MLMS drive runs/flight lines to a common reference frame. Registration approaches can be categorized based on the used primitives; namely, direct cloud-to-cloud registration and feature-based registration. The well-known iterative closest point (ICP) [43,44] and its variants belong to the cloud-to-cloud registration category. Such algorithms assume that some initial transformation parameters exist, and their aim is to refine these parameters. Besl and McKay (1992) assumed that the closest points from two point clouds after coarse registration constitute a conjugate point pair, and used such pairs for the estimation of transformation parameters. Chen and Medoini (1992) further considered surfaces normal during point matching; more specifically, point-to-point correspondence is established by normally projecting a point in one scan onto its adjacent surface in the other scan. Several variants were introduced to improve the ICP robustness using local neighborhood characteristics [45,46]. Studies that share similar concepts, such as the iterative closest projected point [47], point-to-plane registration [48], and multiscale model-to-model cloud comparison [49] have been introduced. In addition, some registration algorithms start by generating a 3D mesh or surface model (e.g., triangulated irregular network), and then use cloud-to-surface or surface-to-surface pairings to estimate the transformation parameters [50,51]. However, creating a mesh is a complex task, especially for point clouds with vertical discontinuities and/or excessive occlusions. In general, cloud-to-cloud fine registration using the ICP or its variants has some disadvantages: (i) it requires large overlap areas between point clouds; (ii) it is sensitive to point density distribution and noise level within the point clouds; and (iii) it requires solid surfaces with good variation in orientation/slope/aspect. For TLS and mobile LiDAR data, a large overlap is not always guaranteed due to the varying sensor-to-object distance, occlusions, and constraints imposed by the scanning environment (e.g., traffic along transportation corridors). In addition, the point density and precision drop quickly as one moves away from the sensor/trajectory, and the distribution of the surface orientation variation within the study site can be unbalanced. These factors could lead to overweighting when estimating the transformation parameters. In order to meet high accuracy requirements, user intervention is unavoidable.

Another group of registration algorithms utilize common features that can be identified in point clouds captured at different locations. In general, such algorithms do not require coarse alignment of the point clouds. The major task in feature-based registration is the identification of common points/features. This task can be quickly performed semi-automatically. Special targets (e.g., highly reflective checkerboard and/or spherical targets), which can be identified in the point clouds, were commonly used to increase the level of automation within the registration process [52–55]. Among such targets, spherical targets were used more frequently since they eliminate the need to re-orient the target to face the scanner during point cloud acquisition [53]. Target-free registration is nonetheless the ultimate goal and, therefore, many studies focused on the automated identification and matching of features. In urban areas, geometric primitives such as point, linear, cylindrical, and planar features can be reliably extracted from point clouds. Such primitives have been employed to estimate the transformation parameters between two point clouds [54,56–65]. Some studies demonstrated the ability of feature-based registration in handling point clouds acquired by different platforms, including airborne LiDAR, mobile LiDAR, TLSs, and even imagery [66–71]. Habib et al. [72] registered LiDAR and photogrammetric data using linear

features. Renaudin et al. [73] utilized photogrammetric data to help in feature extraction and the registration of TLS data with minimal overlap. In addition to geometric primitives, key points based on local feature descriptors that encompass local shape geometry were used for registration [74–76]. Recently, learning-based local feature descriptors and point cloud registration framework targeting fully automated feature detection, and matching evolved rapidly [77–83]. While deep-learning-based methods have shown superior performance on several benchmark datasets, their generalization ability on unseen real datasets needs careful evaluation.

Currently, the vast majority of existing registration tasks are based on pair-wise registration (i.e., the registration process is sequentially established two scans at a time) and/or require initial segmentation to derive the feature parameters, which are used in the registration process. Pair-wise registration has two disadvantages: (i) it makes the process time-consuming when dealing with multiple scans and/or drive-runs; and (ii) the sequential registration leads to the propagation of errors, which increases as we move away from the reference scan. Furthermore, existing algorithms commonly utilize feature parameters (e.g., line endpoints, direction vector of linear/axis of cylindrical features, and normal vector of a plane) for the registration process [53,55–72,84]. Direct use of these parameters would not allow for the sufficient mitigation of inherent noise and/or point density variations in the point cloud. In response to these challenges, this study provides a strategy for the simultaneous registration of several scans from TLS and MLMS drive runs. Moreover, the individual points along the registration features, rather than their parameters, are used in the registration process, thus allowing for the mitigation of noise level/point density variation in the point cloud as well as the negative impact of the partial coverage of registration primitives in the different scans.

4.3 Data Acquisition Systems and Field Surveys

In this study, MLMS and TLS datasets were collected over and underneath a highway bridge that was suspected to have an inadequate deck thickness. This section introduces the data acquisition system and calibration strategy. We also describe the study site and provide information regarding field surveys.

4.3.1 System Description and Calibration

The MLMS data used in this research were captured by a mapping-grade system—Purdue Wheel-based Mobile Mapping System-High Accuracy (PWMMS-HA)—and a surveying-grade system—Purdue Wheel-based Mobile Mapping System-Ultra High Accuracy (PWMMS-UHA). Figure 4-1a shows the PWMMS-HA, which is equipped with four LiDAR units (three Velodyne HDL-32E and one Velodyne VLP-16 High Resolution), three cameras, and a GNSS/INS unit for direct georeferencing. The front-left, front-right, rear-left, and rear-right LiDAR units are hereafter denoted as HDL-FL, VLP-FR, HDL-RL, and HDL-RR, respectively. The range accuracy of the Velodyne HDL-32E and VLP-16 is ± 2 cm and ± 3 cm, respectively [85,86]. The post-processing positional accuracy of the GNSS/INS unit is ± 2 cm with an attitude accuracy of 0.020° and 0.025° for the roll/pitch and heading, respectively [87]. The PWMMS-UHA, shown in Figure 4-1b, is outfitted with two 2D profiler LiDAR units: a Riegl VUX-1HA (hereafter denoted as RI) and a Z+F Profiler 9012 (hereafter denoted as ZF). Two rear-looking cameras and georeferencing units are also installed on this system. The range accuracy of the RI and ZF scanners is ± 5 mm and ± 3

mm, respectively [88,89]. The GNSS/INS post-processing positional accuracy is \pm 1–2 cm with an attitude accuracy of \pm 0.003° for pitch/roll and \pm 0.004° for heading [90].

The rigorous system calibration introduced by Ravi et al. (Ravi et al., 2018) was carried out for both MLMS vehicles to determine the relative position and orientation (denoted hereafter as mounting parameters) between the onboard sensors and IMU body frame, whose position and orientation are derived through the GNSS/INS integration process. The expected post-calibration positional accuracy of the derived point cloud was estimated using the individual sensors' specifications and standard deviations of the estimated mounting parameters through the LiDAR error propagation calculator developed by Habib et al. (A. Habib et al., 2006). The calculator suggests an expected accuracy of about \pm 4 cm and \pm 2 cm at a range of 30 m from the vehicle for the PWMMS-HA and PWMMS-UHA, respectively.

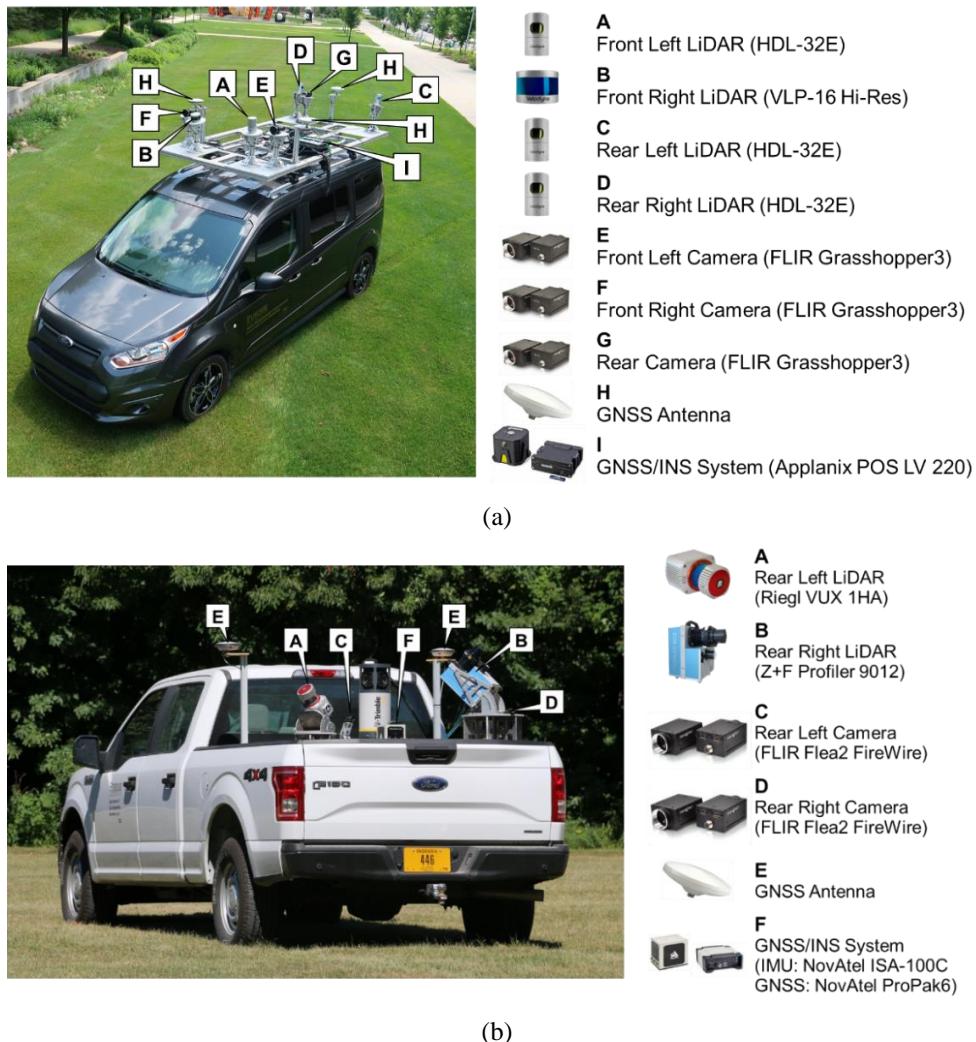
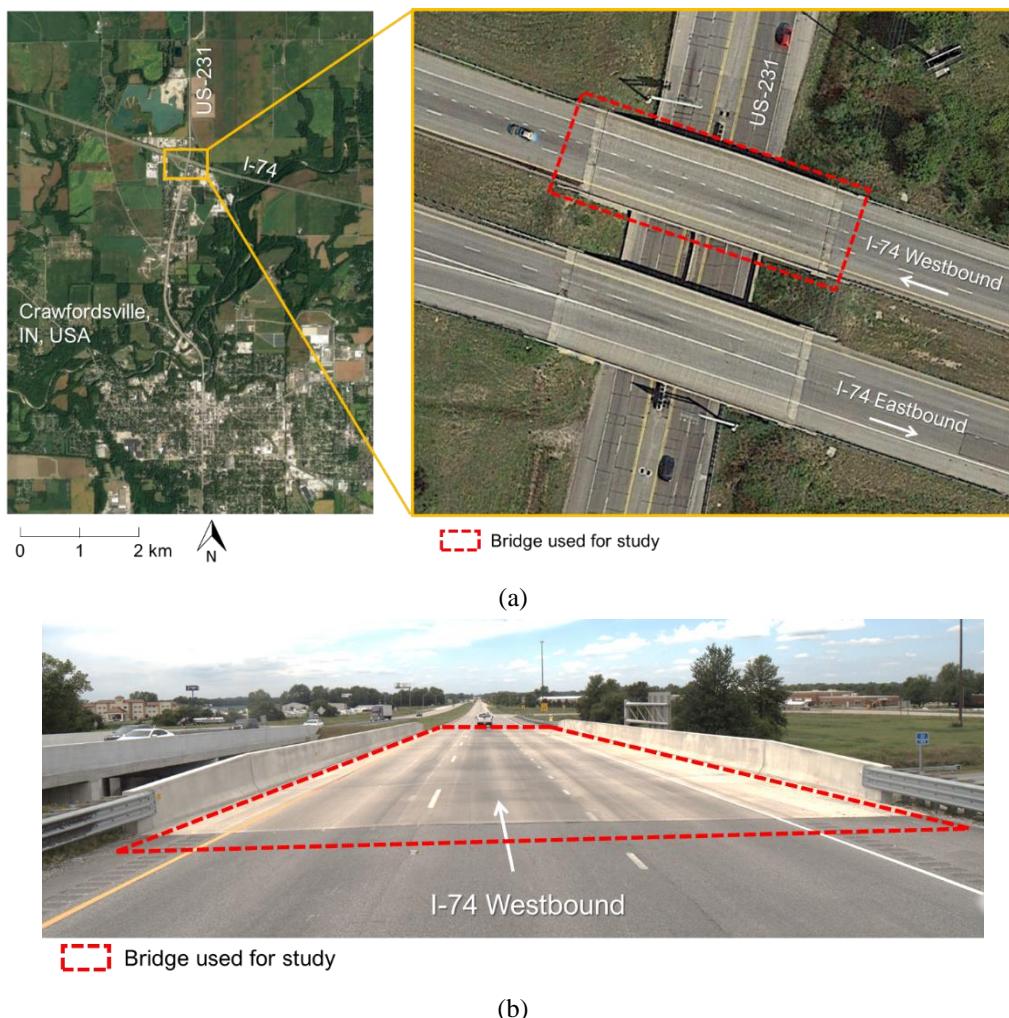


Figure 4-1 The wheel-based mobile mapping systems and onboard sensors used in this study: (a) Purdue Wheel-based Mobile Mapping System-High Accuracy (PWMMS-HA) and (b) Purdue Wheel-based Mobile Mapping System-Ultra High Accuracy (PWMMS-UHA). Both platforms are non-commercial systems designed and integrated by the research group.

The TLS point clouds were captured by FARO Focus 3D X330 and Trimble TX8b laser scanners. The FARO Focus 3D X330 laser scanner is integrated with a high-dynamic-range imaging unit. It has a range systematic error of ± 2 mm and a range noise better than ± 0.5 mm for objects 10 m to 25 m away from the scanner (one sigma). The scanning speed is up to 976,000 points per second with a maximum range of 330 m [93]. The Trimble TX8b laser scanner has a range systematic error of ± 2 mm and a range noise better than ± 2 mm for objects 2 m to 120 m away from the scanner (in standard modes at one sigma). It can scan up to one million points per second with a maximum range of 120 m [94].

4.3.2 Study Site and Field Surveys

Field surveys were conducted over and underneath a westbound bridge along an interstate highway at the intersection of the I-74 and US-231 near Crawfordsville in Indiana, USA (shown in Figure 4-2a). The bridge in question required post construction grinding to meet smoothness and ride quality standards. Once grinding was completed, it was important to perform an as-built survey to document bridge deck thickness. Figure 4-2b,c displays images acquired by one of the cameras onboard the PWMMS-HA, showing the concrete deck and side view of the bridge, respectively.





(c)

Figure 4-2 Study site: (a) the westbound bridge at the intersection of the I-74 and US-231 near Crawfordsville in Indiana, USA (aerial photo adapted from Google Earth images), (b) image of the bridge captured by PWMMS-HA while driving westbound on the I-74, and (c) side view of the bridge captured by PWMMS-HA while driving southbound on the US-231.

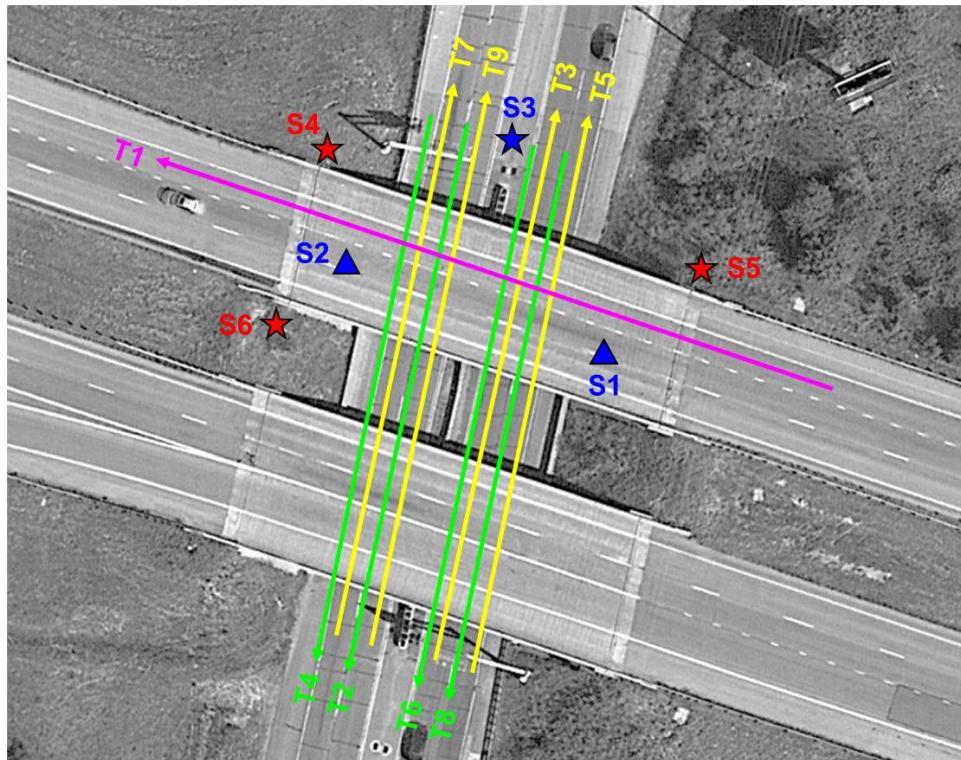
Table 4-1 lists the specifications of the three datasets collected in this study. The MLMS drive run configuration and TLS scan locations are shown in Figure 4-3a. Both MLMS vehicles started by driving westbound on the I-74 (magenta track in Figure 4-3a, T1), and then drove southbound and northbound below the bridge on the US-231 (green—T2, T4, T6, and T8—and yellow—T3, T5, T7, and T9—tracks in Figure 4-3a). Just before reaching the bridge on the I-74, a highway patrol cruiser slowed down the traffic while driving the PWMMS-UHA and PWMMS-HA. Tracks T2–T9, which were designed for investigating the impact of partial GNSS signal outage, were conducted while stopping the southbound and northbound traffic on US-231 for less than three minutes. In total, the data acquisition on the I-74 and US-231 took about five minutes. The average driving speed was 30 km/h for both MLMS vehicles. Figure 4-4 illustrates the position accuracy charts reported by the GNSS/INS integration software, providing a glimpse of the trajectory quality for the two MLMS vehicles. As can be observed in the figure, the position error of PWMMS-UHA is smaller compared to that of PWMMS-HA, owing to the higher-end IMU onboard. In Figure 4-4, the highlighted eight peaks correspond to the eight tracks below the bridge (T2–T9 in Figure 4-3a), suggesting suboptimal accuracy for the below-bridge tracks due to intermittent access to a GNSS signal. The position error, however, does not increase over time since (i) the signal was restored when the vehicles cleared the bridge, (ii) the GNSS data were processed in both forward and backward directions, and (iii) the relatively short GNSS data outage duration can be handled by the onboard IMU.

To speed up the scanning process, TLS point clouds underneath and over the bridge were simultaneously captured by FARO Focus 3D X330 and Trimble TX8b laser scanners, respectively. Figure 4-3a illustrates the scan locations with the Trimble stations set up outside the barrier rail atop of the I-74 bridge embankment to capture the top surface of the deck—Figure 4-3b. The FARO stations were located along the US-231 to capture the bottom surface of the deck—Figure 4-3c. While the below-bridge FARO scan locations ensured sufficient coverage of the bottom surface of the deck, the setup of the Trimble scans was less than optimal as the overlap region between the east (S5) and west (S4 and S6) scans, which are 55 m apart, happens at the middle of

the bridge. Therefore, one might expect less than optimal registration of the Trimble scans due to large separation, lower point density, and shallow scan angles within the overlap region. In terms of data collection time, each TLS scan took about thirty-five minutes for a total of three hours, considering the time for the scanners to move and set up between the different locations.

Table 4-1 Specifications of the acquired point clouds by PWMMS-HA, PWMMS-UHA, and Terrestrial laser scanners (TLSs) above and below the bridge in question.

Sensor	Number of Tracks/Scans	Number of Points per Track/Scan	Data Acquisition Time
PWMMS-HA	HDL-RR	9	~7 million 5 min
	HDL-RL	9	
	HDL-FL	9	
	VLP-FR	9	
PWMMS-UHA	RI	9	~15 million 5 min
	ZF	9	
TLS	FARO	3	~167 million ~199 million 3 h
	Trimble	3	



- ◀ Track above the bridge (westbound on I-74)
- ◀ Track below the bridge (southbound on US-231)
- ◀ Track below the bridge (northbound on US-231)
- ▲ FARO scan below the bridge on US-231
- ★ FARO scan on US-231
- ★ Trimble scan on the I-74 bridge embankment

(a)

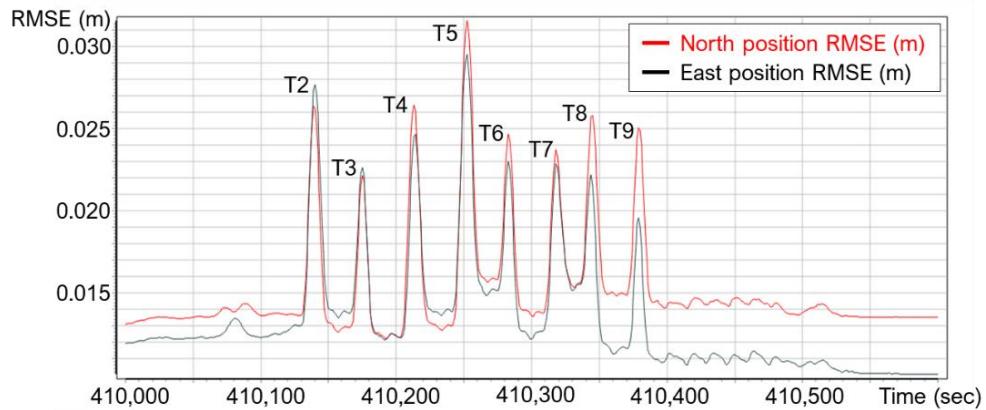


(b)

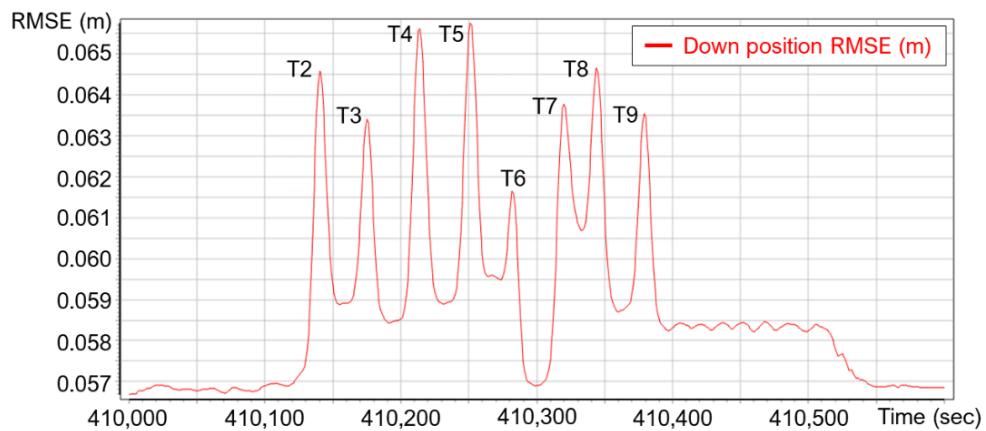


(c)

Figure 4-3 Data acquisition: (a) drive run configuration for the vehicles (Tracks T1–T9) and TLS scan locations (Scans S1–S6), (b) image of the Trimble station (Scan S4) atop the I-74 embankment outside the barrier rail, and (c) image of the FARO station (Scan S1) on the US-231 under the I-74 bridge.



(a)



(b)

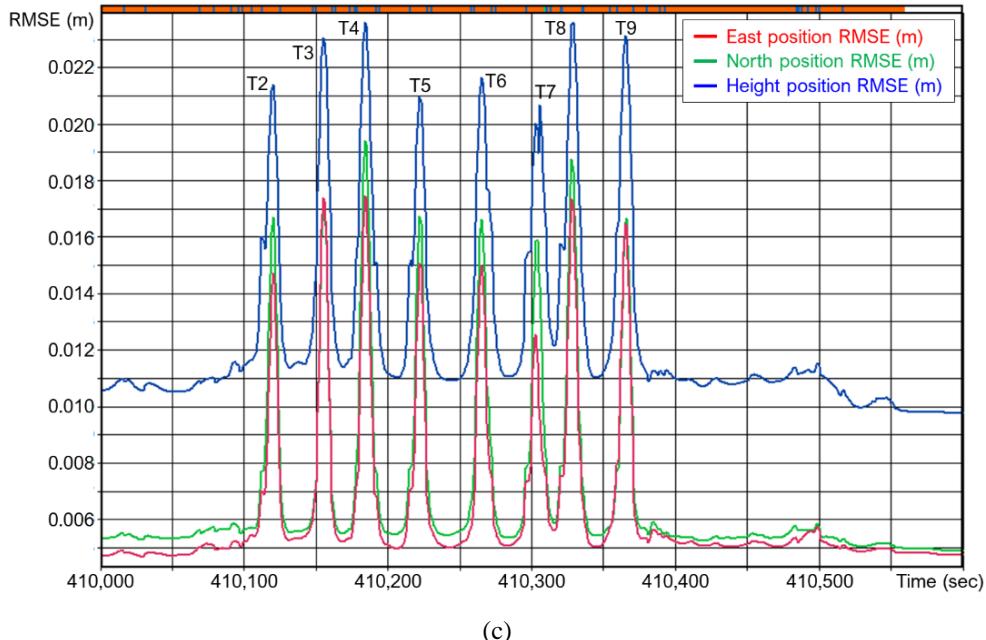


Figure 4-4 Global navigation satellite system/inertial navigation system (GNSS/INS) position accuracy charts for the (a) PWMMS-HA (north and east positions), (b) PWMMS-HA (down position), and (c) PWMMS-UHA vehicles. The highlighted eight peaks correspond to the eight southbound and northbound tracks (Tracks T2–T9) on the US-231 below the bridge, where suboptimal position accuracy can be observed.

4.4 Methodology for Point Cloud Registration and Bridge Deck Thickness Evaluation

The proposed workflow (shown in Figure 4-5) is comprised of registration, bridge deck thickness evaluation, and comparative quality assessment. For MLMS tracks, the point clouds are directly georeferenced in a global mapping frame, and coarse alignment is thus guaranteed. Feature-based fine registration is carried out to fine-tune the alignment between point clouds from different tracks. For TLSs, the point cloud acquired by each scan is available in a different local reference frame, which is defined by the scan location/setup. A coarse registration is carried out based on manually identified conjugate points in the respective point clouds, so that the feature extraction (which will be covered later) for individual scans can be conducted simultaneously. A successive feature-based fine registration is performed to align the point clouds from the different scans. One should note that whether a point cloud is in a local or global reference frame is not critical for the evaluation of bridge deck thickness. In this study, a coarse registration between the TLS and MLMS data is performed only to compare their thickness estimates. The following subsections describe the proposed feature-based fine registration, bridge deck thickness evaluation, and comparative quality assessment.

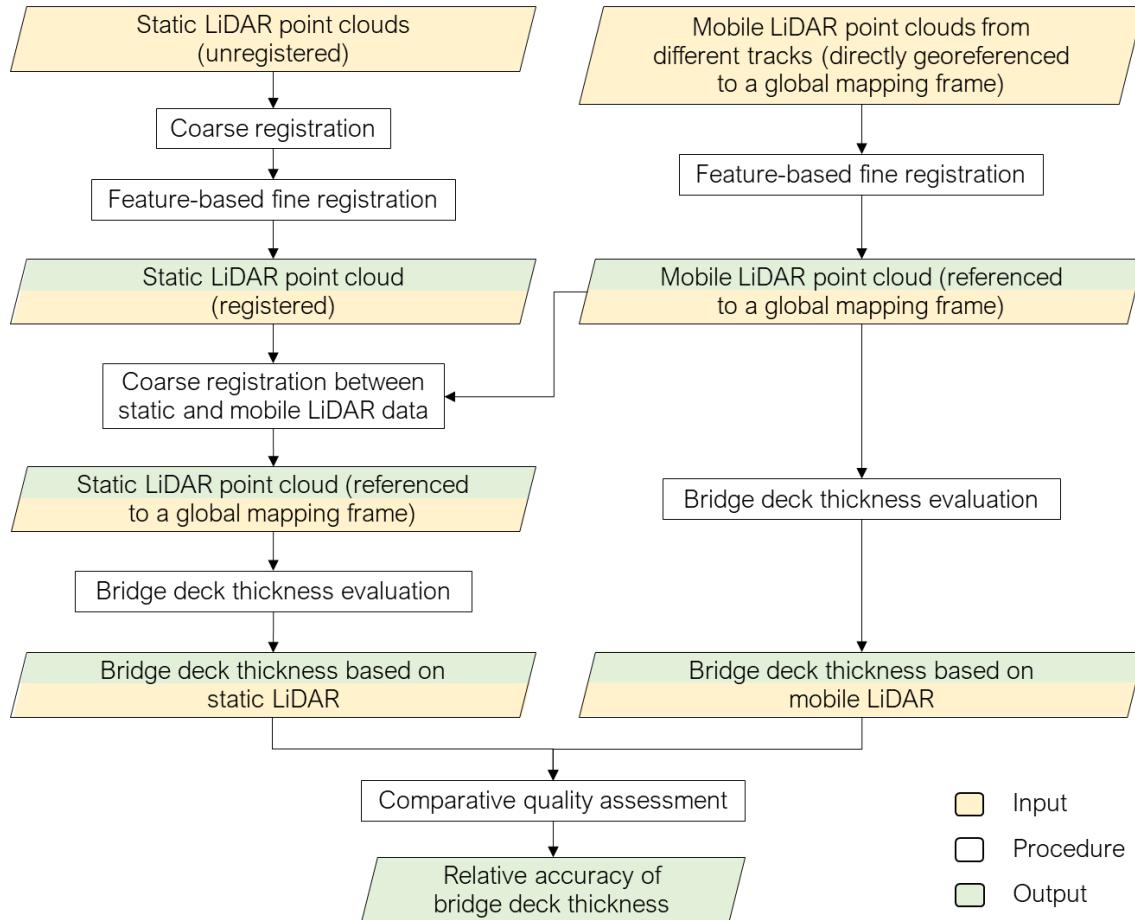


Figure 4-5 Workflow of the proposed bridge deck thickness evaluation and comparative quality assessment strategy.

4.4.1 Feature-Based Fine Registration

The key advantage of the proposed registration strategy is the simultaneous alignment of point clouds from multiple TLS scans or MLMS drive-runs using planar, linear, and cylindrical features without the setup of specific targets. Besides point cloud alignment, the parametric model of registration primitives is derived and can be used for developing an as-built 3D model as well as the subsequent monitoring of bridge elements (i.e., settlement and deformation of bridge support elements).

4.4.1.1 Semi-Automated Feature Extraction

The proposed feature extraction strategy is adapted from the multi-class simultaneous segmentation proposed by Habib and Lin (Ayman Habib & Lin, 2016). First, a seed point is manually identified in the point cloud from the individual scans/tracks. If the point clouds are roughly aligned, the same seed point can be used to extract the feature from all scans/tracks. Otherwise, the user will need to select the seed point for each scan/track. Next, a seed region is established by identifying the neighboring points of the seed point. Dimensionality analysis [96]

is then performed to classify the seed region as a linear/cylindrical, planar, or rough feature. Only planar, linear, and cylindrical seed regions are considered for feature extraction. The parameters of the best-fitting plane/line/cylinder, which will be discussed later in this section, are estimated via an iterative model fitting and outlier removal. More specifically, model parameters are estimated in an iterative manner while assigning lower weights to points that are farther from the fitted surface in the previous iteration. Next, neighboring points that belong to the current feature are augmented through region growing. Concretely, neighboring points whose normal distance are smaller than a multiplication factor times the root-mean-square error (RMSE) of the fitted model are added. The output of the feature extraction includes the segmented points (an example is shown in Figure 4-6) and parameters describing the respective feature model—both are required in the LSA registration and model parameterization.

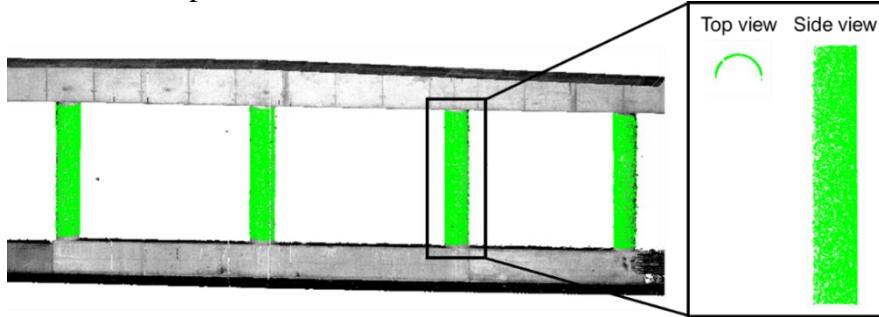


Figure 4-6 An example of cylindrical features (in green) that have been segmented from a LiDAR point cloud (colored by intensity).

In terms of the parametric model representation, a 3D plane is defined by the normal vector to the plane, $[w_x \ w_y \ w_z]^T$, and signed normal distance from the origin to the plane (d) as shown in Equation 3-1. To establish an independent set of parameters, one of the normal vector components is fixed to 1. The fixed component is chosen according to the orientation of the plane, which is defined based on the eigenvectors corresponding to the smallest eigenvalue, as illustrated in Figure 4-7. A cylindrical feature, on the other hand, is defined by a 3D line representing its axis and radius (r). The cylinder axis is represented by a point, $[x_0 \ y_0 \ z_0]^T$, and a direction vector, $[u_x \ u_y \ u_z]^T$, as can be seen in Equation 3-2, where q represents the point location along the cylinder axis. Since a line has only four degrees of freedom, one of the coordinate components is set to zero, with the respective direction vector component set to 1. The fixed parameters for the cylinder axis are chosen according to its orientation, which is defined based on the eigenvectors corresponding to the largest eigenvalue, as depicted in Figure 4-8. Finally, a linear feature is represented the same way as the axis of a cylindrical feature (i.e., four parameters, with two representing a point along the line and two defining its direction).

$$\frac{w_x x + w_y y + w_z z}{\sqrt{w_x^2 + w_y^2 + w_z^2}} + d = 0 \quad \text{Equation 4-1}$$

$$\begin{aligned} x &= qu_x + x_0 \\ y &= qu_y + y_0 \\ z &= qu_z + z_0 \end{aligned} \quad \text{Equation 4-2}$$

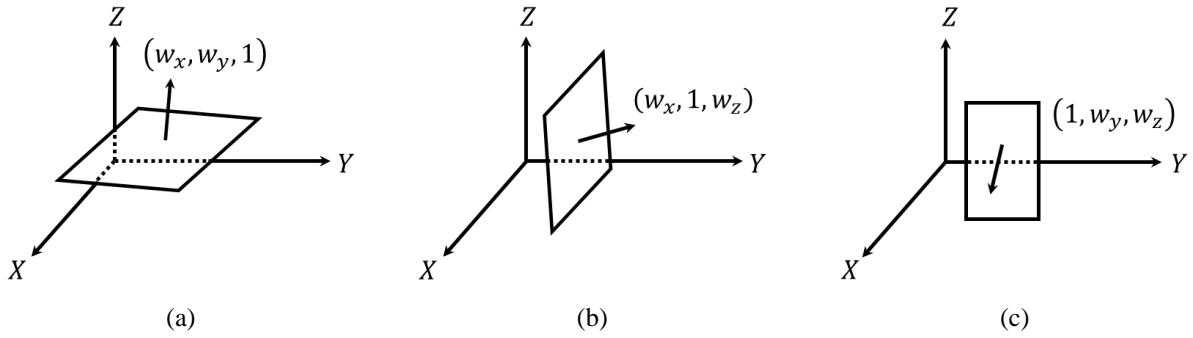


Figure 4-7 Different options for representing planar features showing the normal vectors to the planes (defined by the eigenvector corresponding to the smallest eigenvalue) that are mainly along the (a) Z-axis (i.e., the eigenvector component along the Z-axis is larger than those along the X and Y axes), (b) Y-axis (i.e., the eigenvector component along the Y-axis is larger than those along the X and Z axes), and (c) X-axis (i.e., the eigenvector component along the X-axis is larger than those along the Y and Z axes).

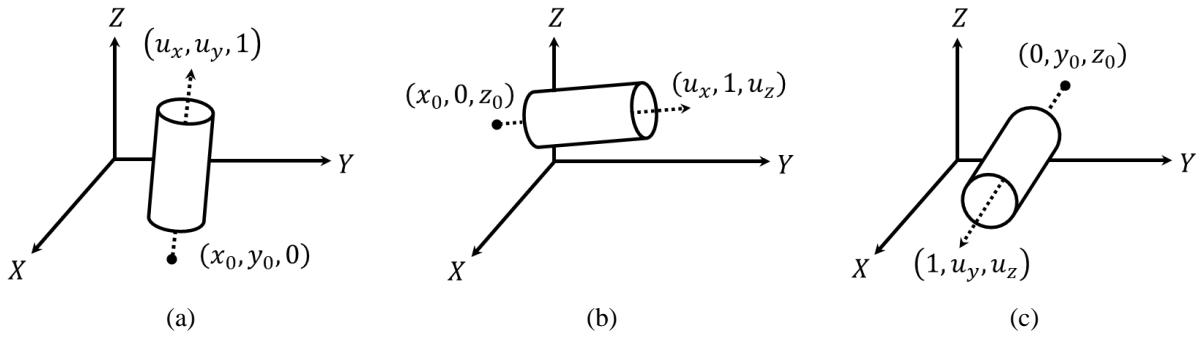


Figure 4-8 Different options for representing cylindrical features with directions (defined by the eigenvector corresponding to the largest eigenvalue) that are mainly along the (a) Z-axis (i.e., the eigenvector component along the Z-axis is larger than those along the X and Y axes), (b) Y-axis (i.e., the eigenvector component along the Y-axis is larger than those along the X and Z axes), and (c) X-axis (i.e., the eigenvector component along the X-axis is larger than those along the Y and Z axes).

4.4.1.2 Least-Squares Adjustment for Feature-Based Fine Registration

The conceptual basis of the proposed LSA model is that conjugate features would fit a single parametric model after registration. The objective function of the LSA model estimates the transformation parameters as well as feature parameters in the common reference frame. For simultaneous registration between m_s point clouds, one of them is selected to define a common reference frame, $(\vec{pc}_c = [x_c \ y_c \ z_c]^T)$, and the rest are considered as sources, $(\vec{pc}_{si} = [x_{si} \ y_{si} \ z_{si}]^T, i = 1, 2, \dots, m_s - 1)$. The 3D similarity model in Equation 3-3 is used to represent the transformation from the i^{th} source (si) to the common (c) reference frames, where $\vec{t}_{si}^c, R_{si}^c, s_{si}^c$ denote the translation vector, rotation matrix, and scale factor, respectively:

$$\overrightarrow{pc}_c = \vec{t}_{si}^c + s_{si}^c R_{si}^c \overrightarrow{pc}_{si}$$

Equation 4-3

For planar features, the normal distance vector, $[nd_x \ nd_y \ nd_z]^T$, between a transformed point, $[x_c \ y_c \ z_c]^T$, and the post-registration parametric model, defined by $[w_x \ w_y \ w_z]^T$ and d , is presented in Equation 3-4—refer to the illustration in Figure 4-9a. For a linear/cylindrical feature, the normal distance vector, $[nd_x \ nd_y \ nd_z]^T$, between a transformed point, $[x_c \ y_c \ z_c]^T$, and the post-registration parametric model representing the linear/axis of a cylindrical feature, defined by $[x_0 \ y_0 \ z_0]^T$ and $[u_x \ u_y \ u_z]^T$, is expressed in Equation 3-5. An illustration of the linear/cylindrical feature post-registration model fitting is shown in Figure 4-9b,c, respectively. For planar and linear features, the LSA aims at minimizing the squared sum of normal vector components in Equations 3-4 and 3-5, respectively, for all the points in the different scans/tracks that encompass such features. In these equations, \odot denotes the dot product of two vectors. For cylindrical features, on the other hand, the LSA aims at minimizing the squared sum of normal distances between the points that belong to such features and their surface, as given by Equation 3-6:

$$\begin{bmatrix} nd_x \\ nd_y \\ nd_z \end{bmatrix} = -\frac{d \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}}{\sqrt{w_x^2 + w_y^2 + w_z^2}} - \frac{\left(\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \odot \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \right) \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}}{w_x^2 + w_y^2 + w_z^2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 4-4}$$

$$\begin{bmatrix} nd_x \\ nd_y \\ nd_z \end{bmatrix} = \begin{bmatrix} x_c - x_0 \\ y_c - y_0 \\ z_c - z_0 \end{bmatrix} - \frac{\left(\begin{bmatrix} x_c - x_0 \\ y_c - y_0 \\ z_c - z_0 \end{bmatrix} \odot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \right) \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}}{u_x^2 + u_y^2 + u_z^2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 4-5}$$

$$nd_x^2 + nd_y^2 + nd_z^2 - r^2 = 0 \quad \text{Equation 4-6}$$

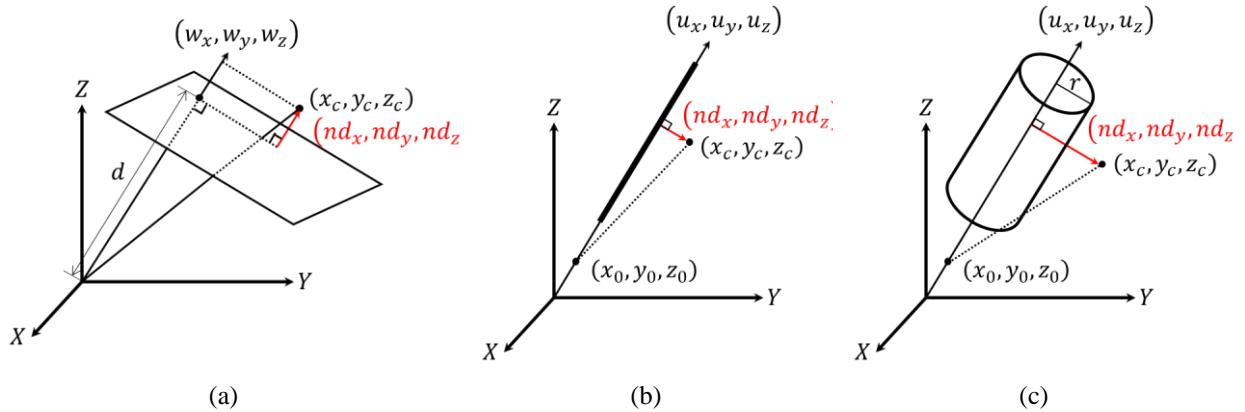


Figure 4-9 Schematic illustration of the normal distance vector components that are minimized using the (a) plane-fitting model, (b) 3D line-fitting model, and (c) cylinder-fitting model.

The linearized mathematical model corresponding to Equations 3-4–3-6 can be written as in Equation 3-7, which is commonly known as the Gauss–Helmert model [97]. Here, y is the discrepancy vector arising from the linearization process; e is the vector of random noise contaminating the point cloud coordinates in the different scans/tracks, which follows a stochastic distribution with a zero mean and a variance–covariance matrix of $\sigma_0^2 P^{-1}$, with σ_0^2 representing the a priori variance factor and the weight matrix, P , depending on the specifications of the data acquisition system; and x is the vector of unknown parameters (including transformation and feature parameters). The matrices A and B are composed of the partial derivatives of the models in Equations 3-4–3-6 with respect to the unknown parameters and point cloud coordinates, respectively. For a planar or linear feature, the respective B matrix is rank-deficient, which can be deduced by analyzing the effective contribution of Equations 3-4 and 3-5 towards the overall redundancy. Although a point on a planar feature provides three equations, it only has a net contribution of one towards the redundancy—corresponding to the minimization of normal distance to the plane. For linear features, the three equations have a net contribution of two towards the redundancy—corresponding to the minimization of normal distance to the linear feature. Lastly, a point on a cylindrical feature provides a net contribution of one towards the redundancy. In order to ensure that the resulting normal matrix is full-rank and that the transformation and feature parameters can be reliably estimated, it is critical to have features with different orientations over the area of interest. The solution vector of the LSA model can be written as per Equation 3-8, where $(BP^{-1}B^T)^+$ denotes the Moore–Penrose pseudoinverse that has to be used to compensate for the rank deficiency of the B matrix. The residuals, \tilde{e} , are given in Equation 3-9. The a posteriori variance factor, $\widehat{\sigma}_0^2$, can be computed using Equation 3-10, where m is the number of unknown parameters. The variance–covariance matrix of estimated parameters is given in Equation 3-11. Interested readers can refer to Ravi and Habib [98] for more details regarding LSA with a rank-deficient weight matrix. Figure 4-10 presents a sample registration result using the ICP and proposed feature-based registration approach. The point clouds were acquired from two tracks in opposite driving directions, and thus capture different sides of the bridge piers. While the ICP incorrectly aligned the two sides of the piers, the proposed approach produced superior results by fitting conjugate features to a single cylinder model:

$$y = Ax + Be, \quad e \sim (0, \sigma_0^2 P^{-1}) \quad \text{Equation 4-7}$$

$$\hat{x} = (A^T(BP^{-1}B^T)^+A)^{-1}(A^T(BP^{-1}B^T)^+y) \quad \text{Equation 4-8}$$

$$\tilde{e} = y - A\hat{x} \quad \text{Equation 4-9}$$

$$\widehat{\sigma}_0^2 = \frac{\tilde{e}^T(BP^{-1}B^T)^+\tilde{e}}{\text{rank}(BP^{-1}B^T) - m} \quad \text{Equation 4-10}$$

$$\Sigma = \widehat{\sigma}_0^2 (A^T(BP^{-1}B^T)^+A)^{-1} \quad \text{Equation 4-11}$$

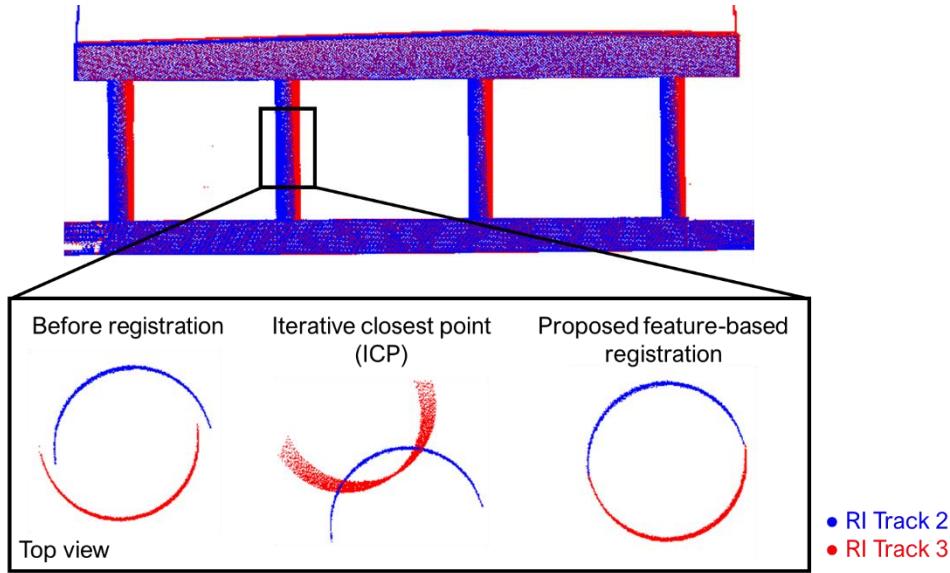


Figure 4-10 Sample registration results using the iterative closest point (ICP) and proposed approach, showing a cylindrical feature before and after registration.

4.4.2 Bridge Deck Thickness Evaluation

The proposed bridge deck thickness evaluation utilizes surface-to-surface distance in order to mitigate the impact of inherent noise on the post-registration point cloud. First, point clouds capturing the top and bottom surfaces of the deck are extracted—one is selected as the target and another as the source. The target and source point clouds are then partitioned into segments with a pre-defined size ($30\text{ cm} \times 30\text{ cm}$ in this study). For a given segment, the dimensionality analysis is carried out to test its planarity, and only planar segments are included in subsequent evaluation. An iterative plane fitting is then performed to remove potential outlier points and estimate the parameters of the plane (refer to Section 4.1.1 for detailed information). A segment is rejected from the thickness evaluation if (i) the plane-fitting RMSE exceeds a user-defined threshold (thres_{RMSE}) or (ii) the remaining inlier points after iterative plane fitting fail to reach another user-defined threshold (thres_{npt}). The first threshold (thres_{RMSE}) can be selected according to the expected noise level in the post-registration point cloud. The second threshold (thres_{npt}) can be defined based on the percentage of remaining points after iterative plane fitting—e.g., if more than 50% of the original points are removed by the outlier removal procedure, the segment is rejected from the thickness evaluation. Next, the center of a target plane and its projection on the source plane are determined. The thickness of the deck at that location is evaluated by the normal distance between the center of the target plane and the source plane. Finally, the deck thickness estimates from all the surface segments are visualized as a heat map. Figure 4-11 illustrates an example, showing the point clouds from the above- and below-bridge tracks (Figure 4-11a) as well as the corresponding thickness map (Figure 4-11b). As evident in Figure 4-11a, the point cloud from the below-bridge tracks shows some gaps because of the occlusion caused by the bridge pier caps.

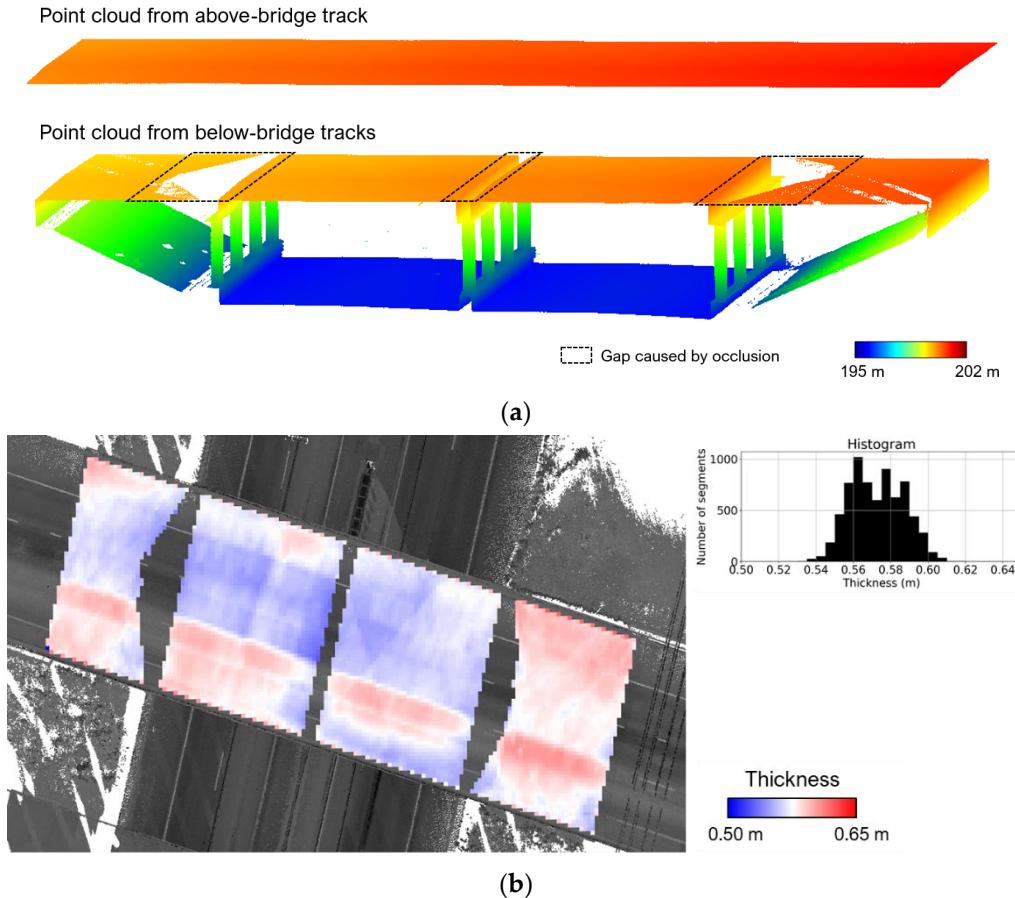


Figure 4-11 Example of (a) a point cloud capturing the top and bottom surfaces of the bridge deck and (b) a heat map representing the estimated bridge deck thickness.

4.4.3 Comparative Quality Assessment

In this study, the derived bridge deck thickness using MLMS data is compared to that estimated from TLS point clouds. The comparison is carried out by evaluating the difference between the thickness estimates for corresponding surface segments in the MLMS and TLS data. Therefore, ensuring the alignment between MLMS and TLS data is a prerequisite. A coarse registration is conducted to transform the TLS point cloud into the MLMS mapping frame. Since the proposed segment-based thickness evaluation is conducted at 30 cm intervals and the coarse registration accuracy is better than ± 10 cm, a fine registration procedure is not necessary. Once the point clouds are aligned, the correspondence is established by searching for the closest surface segment in the MLMS point cloud for each segment in the TLS point cloud. The difference between bridge deck thickness estimates for corresponding surface segments is evaluated and visualized as a heat map. Moreover, the mean, standard deviation, and RMSE of the differences are reported as quantitative measures of the comparative performance of the different sensing modalities (i.e., TLSs and MLMSs).

4.5 Experimental Results and Discussion

This section starts with reporting the registration results for the MLMS and TLS datasets. For the MLMS datasets, the alignment between point clouds from different tracks before and after registration is examined to illustrate the quality of the system calibration and the impact of GNSS outage during data acquisition. Next, the bridge deck thickness is evaluated, and a comparative analysis of the derived metrics is reported.

4.5.1 Point Cloud Registration and Alignment

As described in Section 3.2, one TLS and two MLMS datasets were collected in this study. The TLS dataset had six scans, with each having its local coordinate system. First, a coarse registration using manually identified point pairs in TLS point clouds was performed. For such registration, one of the FARO scans, Scan S1 (see Figure 4-3), was selected to define the common reference frame. In contrast to the TLS data, point clouds acquired by the MLMSs were directly georeferenced to a global mapping frame through the onboard GNSS/INS unit—the UTM coordinate system with WGS84 as the datum was chosen as the reference frame in this study. The point density over the bridge deck from one track is depicted in Figure 4-12, where the median is 7700 and 6800 points per square meter for PWMMS-HA and PWMMS-UHA, respectively. PWMMS-HA had a higher and more uniform point density across the bridge deck as compared to PWMMS-UHA.

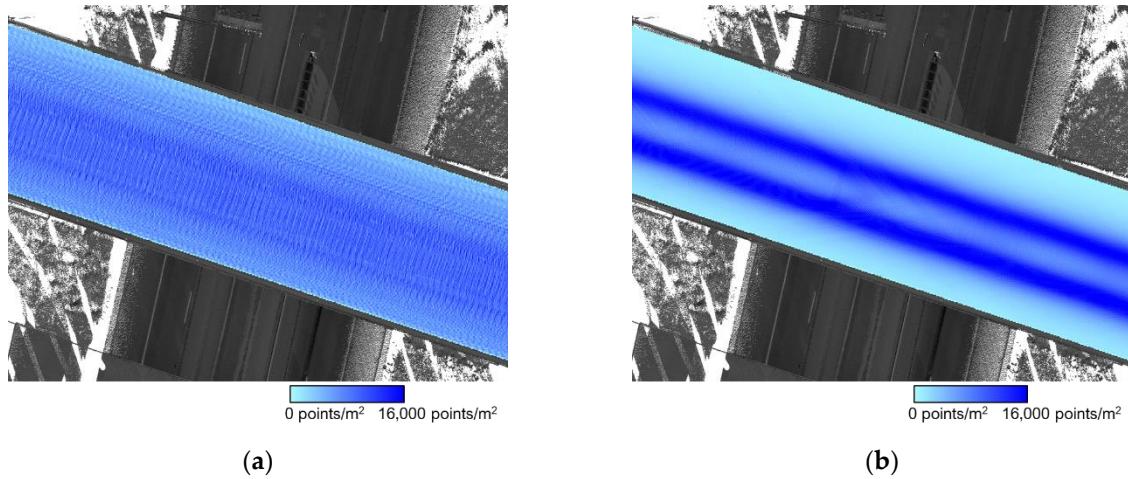


Figure 4-12 Point density over the bridge deck from one track (track T1) for: (a) PWMMS-HA and (b) PWMMS-UHA.

Prior to fine registration, the alignment of MLMS point clouds from different tracks was inspected. Figure 4-13 depicts a cross-section of the bridge deck before registration. The top surface of the bridge deck and inside surface of the barrier rails were captured by the above-bridge track (T1), while the bottom surface of the bridge deck and the outside surface of the barrier rails were scanned by the below-bridge tracks (T2–T9). To assess the agreement between point clouds acquired by the four LiDAR units onboard the PWMMS-HA, we performed plane fitting over a $1\text{ m} \times 1\text{ m}$ segment on the top surface of the bridge deck (see Figure 4-13a). The plane-fitting RMSE was 1.3 cm, indicating that the point clouds from different LiDAR units in Track T1 were in good

agreement, thus verifying the quality of the system calibration parameters (i.e., inaccurate system calibration would result in discrepancies among captured point clouds by different sensors in the same track). A segment on the bottom surface of the bridge deck, in contrast, yielded a plane-fitting RMSE of 3.0 cm. Since this area was captured by eight tracks (T2–T9), one can deduce that the alignment among the point clouds from those tracks was not as good as the alignment between point clouds captured by the PWMMS-HA LiDAR sensors within a given track. The lower trajectory quality due to intermittent access to a GNSS signal below the bridge was the cause of such systematic discrepancy among the tracks, as can be seen in Figure 4-13a. A similar analysis was conducted for the PWMMS-UHA point clouds, as can be seen in Figure 4-13b. The zoomed-in area on the top surface of the bridge deck reveals that for a given track, the precision of point clouds from the two LiDAR units onboard the PWMMS-UHA was in the range of few millimeters. However, due to the intermittent access to a GNSS signal during below-bridge data acquisition, a degradation in the RMSE to the 2.0 cm range can be seen. The barrier rails along the two sides of the bridge were used to evaluate the alignment between the above-bridge and below-bridge tracks. The misalignment was interactively quantified by evaluating the distance between two manually selected points on the top of the barrier rail from the above-bridge and below-bridge point clouds. As shown in Figure 4-13a,b, a misalignment of about 7 cm and 2 cm along the vertical direction was present between the above-bridge and below-bridge point clouds for the PWMMS-HA and PWMMS-UHA, respectively. Such misalignment would lead to, no doubt, an unreliable estimation of bridge deck thickness.

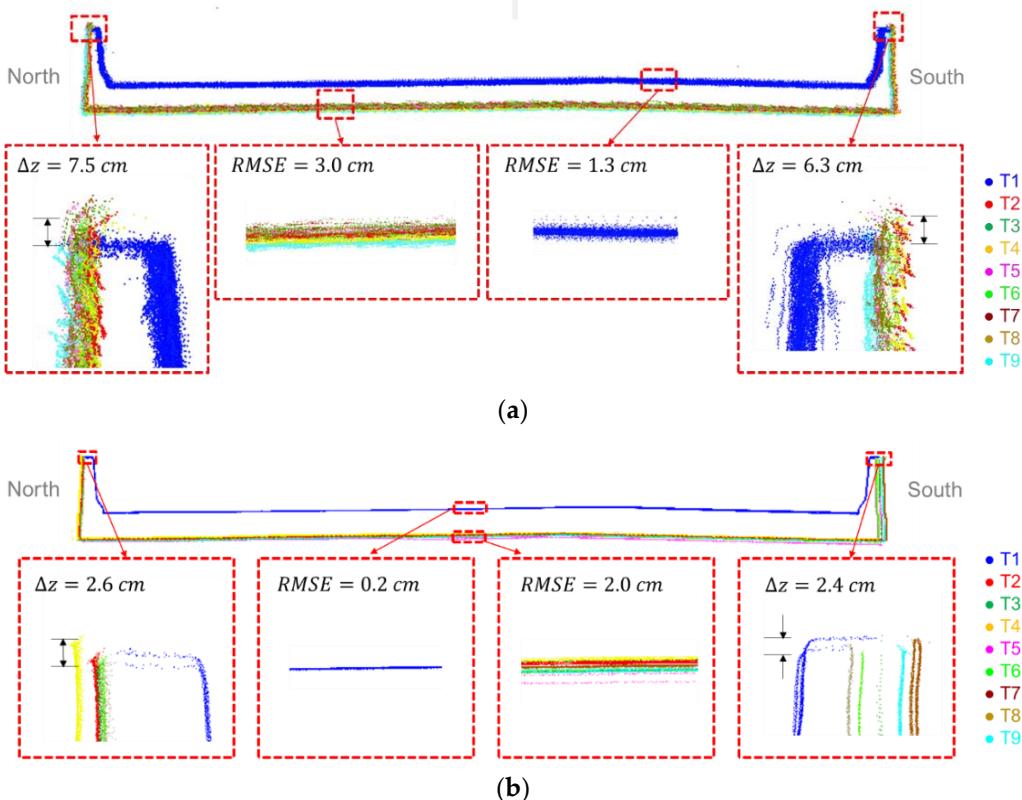


Figure 4-13 Sample cross-sectional profile of the bridge deck showing the point cloud alignment before registration for the (a) PWMMS-HA and (b) PWMMS-UHA datasets. Both datasets have nine tracks (T1 is above the bridge and T2–T9 are below the bridge).

The proposed feature-based fine registration was then performed for the TLS and MLMS datasets to refine the alignment between scans/tracks. First, planar/linear/cylindrical features were semi-automatically extracted from the point clouds, and the results are shown in Figure 4-14. In order to reliably solve for the transformation parameters, features with different orientations were extracted from the point clouds. These features should be well-distributed over the area of interest, and the number of points contributing to the estimation of different transformation parameters was of similar magnitude to prevent over-weighting in the LSA model. A total of 46, 30, and 29 features were extracted from the TLS, PWMMS-HA, and PWMMS-UHA point clouds, respectively. For the TLS dataset, the overlap between the above-bridge and below-bridge scans was adequate, as the Trimble scanner was placed on the I-74 embankments next to the barrier rails, and thus was able to capture objects on the US-231 without much occlusion. For the MLMS datasets, on the other hand, the overlap between the above- and below-bridge tracks was limited due to occlusions caused by the barrier rails. Point clouds acquired by the above-bridge track barely captured the road surface, signboards, and other objects on the US-231. Common features among the above-bridge and below-bridge tracks include signboards, light poles, and the north side of the barrier rail on the westbound I-74, as can be seen in Figure 4-14b,c.

Once the features were extracted, the proposed LSA strategy was carried out to estimate the transformation and post-registration feature parameters. For the MLMS datasets, the LSA estimated the transformation parameters among point clouds acquired by each sensor from individual tracks (i.e., the registration was simultaneously conducted for a total of thirty-six and eighteen point clouds for PWMMS-HA and PWMMS-UHA, respectively). Track T1 from HDL-RR and Track T1 from RI were selected as target tracks for PWMMS-HA and PWMMS-UHA, respectively. For TLSs, Scan S1 was used as the target scan when registering the six point clouds captured by the FARO and Trimble scanners. Figure 4-15 shows box and whisker plots of the transformation parameters for the TLS and MLMS datasets, where one can observe that the magnitude and variance of the parameters for each system were at the same level. The square root of a posteriori variance factor after registration (which represents the noise level in the data as well as the quality of post-registration alignment) is 0.60 cm, 1.47 cm, and 0.74 cm for TLSs, PWMMS-HA, and PWMMS-UHA, respectively. The weighted average of the RMSE of normal distances between the LiDAR points and the best-fitted plane/line/cylinder before and after registration for each dataset is listed in Table 4-2. The reduction in the post-registration RMSE reflects an improvement in the alignment of the features after registration. As expected, the post-registration RMSE values for the different features are in agreement with the square root of a posteriori variance.

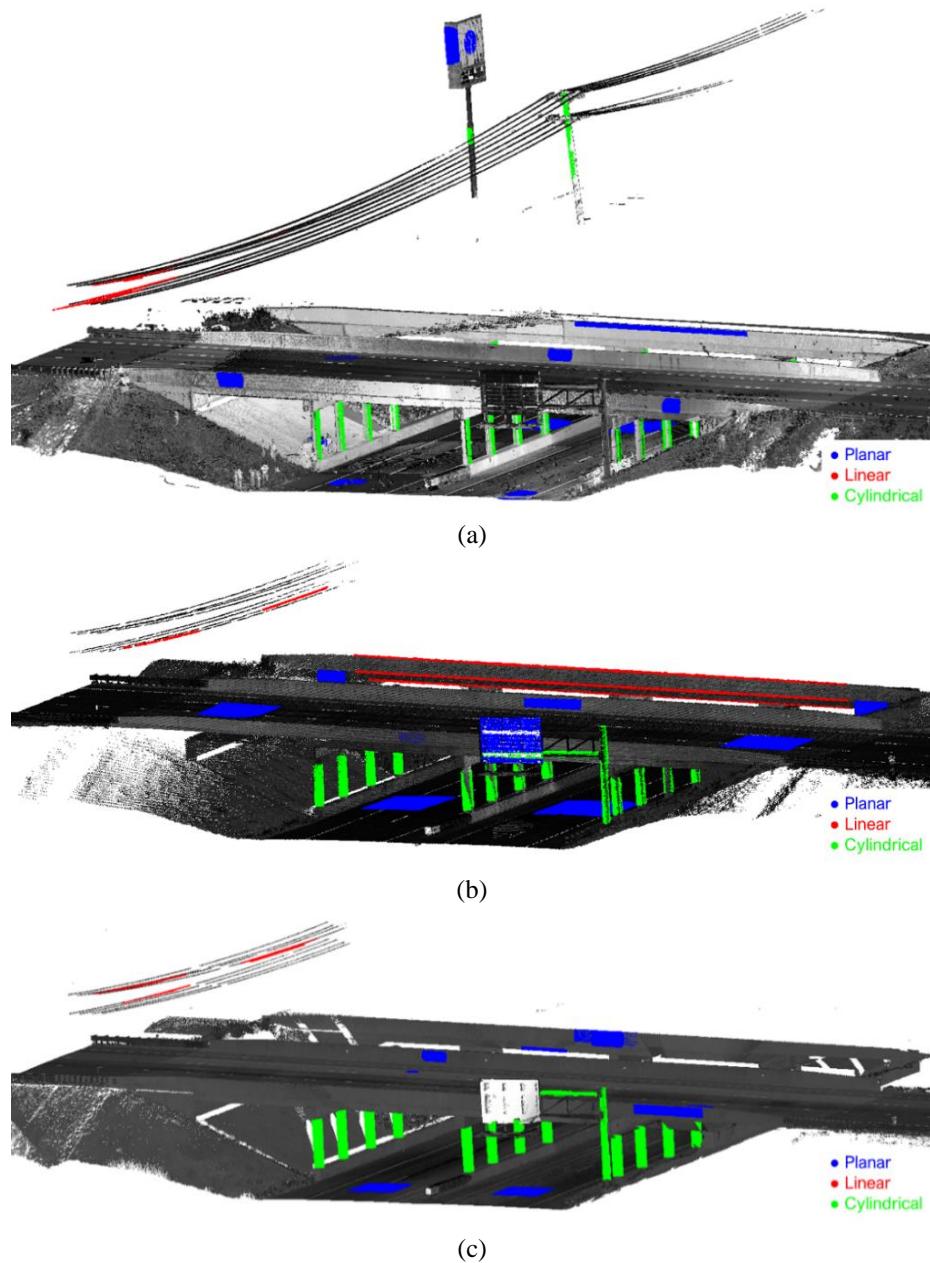


Figure 4-14 Extracted planar/linear/cylindrical features (in blue/red/green, respectively) for the registration of (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA point clouds.

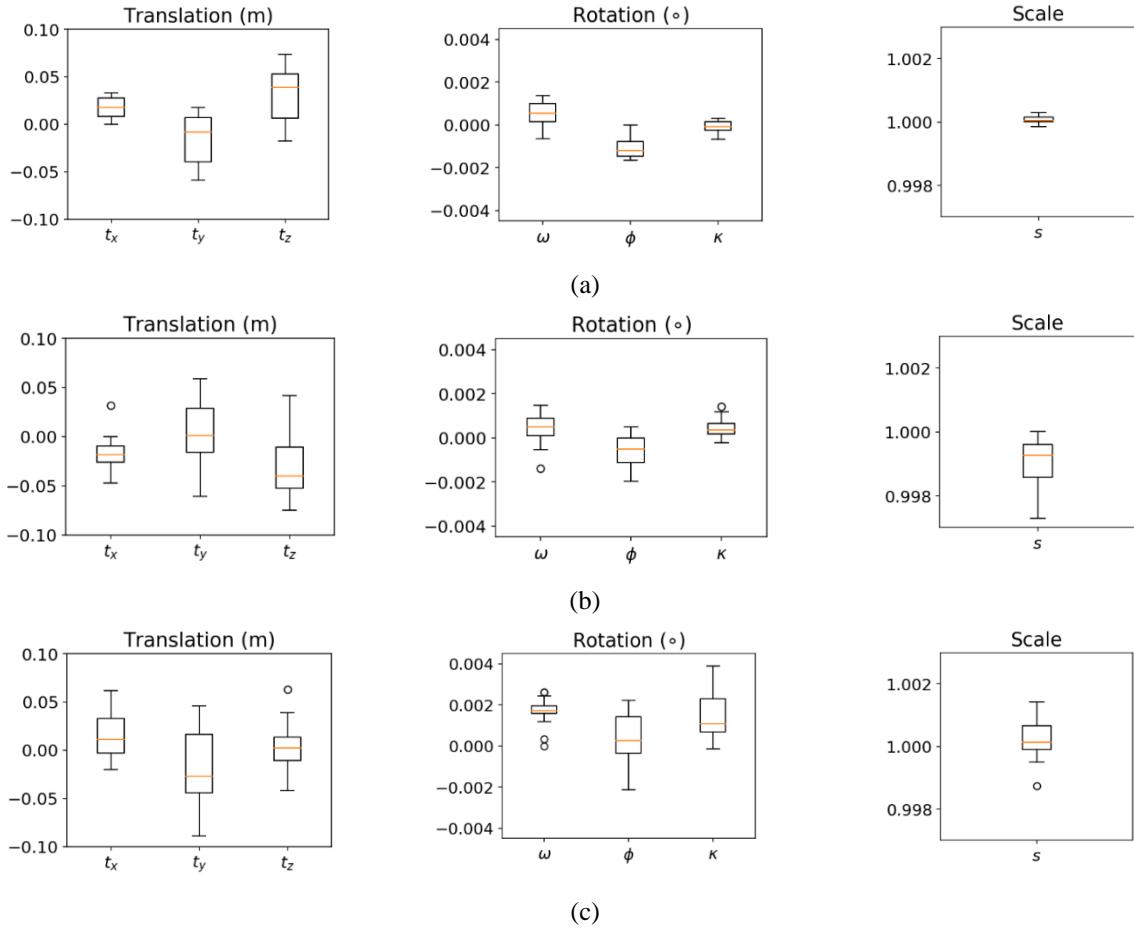


Figure 4-15 Box and whisker plots of the fine-registration transformation parameters for the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.

Table 4-2 Weighted average of the RMSE of plane/line/cylinder fittings before and after registration for TLS, PWMMS-HA, and PWMMS-UHA.

	TLS	PWMMS-HA	PWMMS-UHA
Number of features	Planar	19	11
	Linear	6	5
	Cylindrical	21	14
	Total	46	30
Weighted average of RMSE (m)	Before	0.014	0.026
	After	0.004	0.014

One of the advantages of the proposed alignment strategy is the production of a parametric model representation of the registration primitives, which could be used for bridge monitoring over time (e.g., the cylindrical columns supporting the bridge). To show the comparative performance of TLS and MLMSs in terms of the similarity of derived post-registration parametric models for cylindrical features, Figure 4-16 depicts a top view of the twelve columns supporting the I-74 bridge together with their axes. As evident from the figure, the cylindrical columns from different sensing modalities are well-aligned. Table 4-3 reports the estimated radii of the twelve columns where the estimates from different systems are in agreement within the 1 cm range. Moreover, the

relative planimetric discrepancies between the TLS and MLMS datasets were evaluated using the derived horizontal locations of the cylindrical columns. The results show that the relative horizontal locations are compatible within a 1 cm range. Once again, these values are in agreement with the post-registration estimates of the square root of a posteriori variance factors reported earlier.

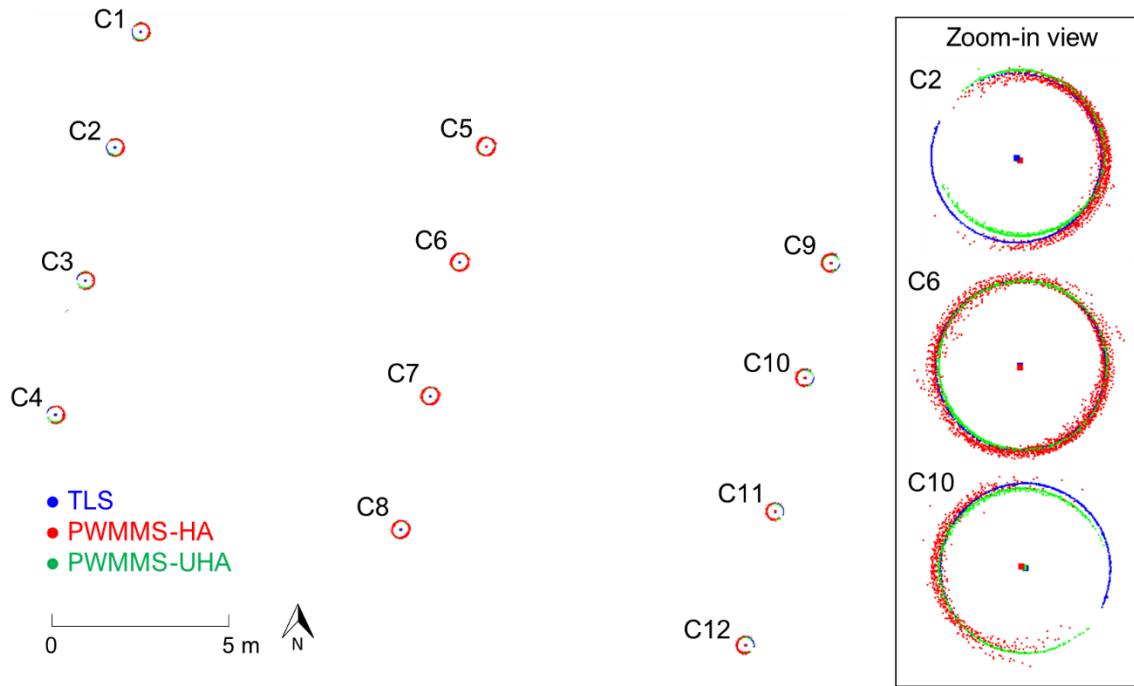


Figure 4-16 The twelve cylindrical columns and their axes derived from TLS (in blue), PWMMS-HA (in red), and PWMMS-UHA (in green) data.

Table 4-3 Estimated post-registration radii of the cylindrical columns supporting the I-74 bridge from the TLS, PWMMS-HA, and PWMMS-UHA point clouds.

ID	Number of Points			Radius (m)			Difference (m)		
	TLS	HA	UHA	TLS	HA	UHA	TLS vs. HA	TLS vs. UHA	HA vs. UHA
C1	24,752	155,483	86,355	0.314	0.314	0.308	0.000	-0.006	-0.006
C2	25,764	150,622	80,216	0.305	0.305	0.297	0.000	-0.008	-0.008
C3	25,840	149,090	78,308	0.306	0.304	0.294	-0.002	-0.012	-0.010
C4	25,366	149,641	78,516	0.305	0.303	0.291	-0.002	-0.014	-0.012
C5	30,775	242,075	158,403	0.316	0.317	0.313	0.001	-0.003	-0.003
C6	20,143	234,633	156,096	0.305	0.306	0.302	0.001	-0.004	-0.004
C7	20,267	233,235	152,566	0.305	0.306	0.301	0.001	-0.004	-0.005
C8	20,123	231,872	149,291	0.305	0.306	0.301	0.001	-0.004	-0.005
C9	25,985	140,673	113,636	0.315	0.310	0.297	-0.005	-0.018	-0.012
C10	26,942	136,115	108,052	0.305	0.302	0.291	-0.003	-0.014	-0.011
C11	27,124	137,503	103,772	0.305	0.303	0.295	-0.002	-0.010	-0.008
C12	25,314	136,477	102,548	0.305	0.303	0.298	-0.002	-0.007	-0.006
			Mean	0.308	0.308	0.308	-0.001	-0.009	-0.008
			Std. Dev.	0.002	0.002	0.002	0.002	0.005	0.003
			RMSE	0.002	0.002	0.002	0.010	0.008	

As another verification of the impact of the fine registration of the MLMS point clouds, a cross-section of the bridge deck (at the same location shown in Figure 4-13) that was extracted from the point is depicted in Figure 4-17, where a significant improvement can be observed. Zoomed-in areas at the barrier rails show that the above-bridge and below-bridge tracks/scans are in agreement within a 1 cm range for all systems. Moreover, zoomed-in areas at the top and bottom surfaces of the bridge deck verify that the point clouds from different tracks/scans are in good agreement along the vertical direction. According to the plane-fitting RMSE shown in Figure 4-17, the precision of the point cloud from PWMMS-HA is in the ± 1.5 cm range, which is better than the expected value of ± 4 cm. Moreover, both PWMMS-UHA and TLS point clouds achieve millimeter-level precision (i.e., in the ± 0.3 cm and ± 0.4 cm range, respectively). In summary, the results show that the proposed feature-based fine registration can effectively minimize the impact of trajectory and system calibration errors, and thus improve the point cloud quality.

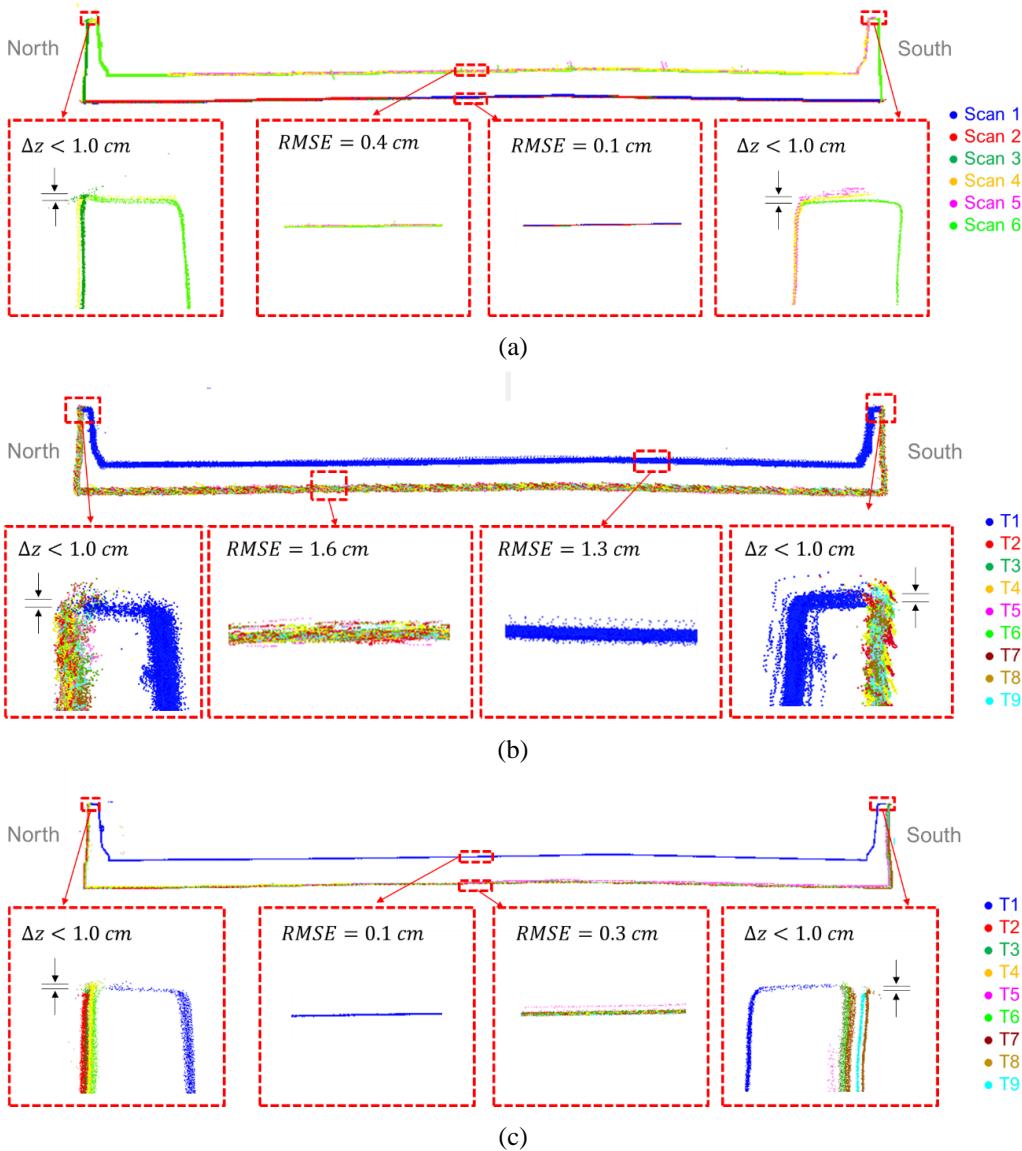


Figure 4-17 Sample cross-sectional profile showing the post-registration point cloud alignment for the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.

4.5.2 Point Cloud Registration and Alignment

Having examined the point cloud alignment, bridge deck thickness was then evaluated using the proposed surface segment-based approach. The segment size was set to 30 cm × 30 cm. The thresholds $thres_{RMSE}$ and $thres_{npt}$ were defined as three times the square root of a posteriori variance factor after registration and 50% of it, respectively. The thickness estimates are visualized as a heat map and shown in Figure 4-18. The spatial patterns from the three datasets are similar as they all indicate a smaller thickness value over the right two lanes towards the west side of the bridge.

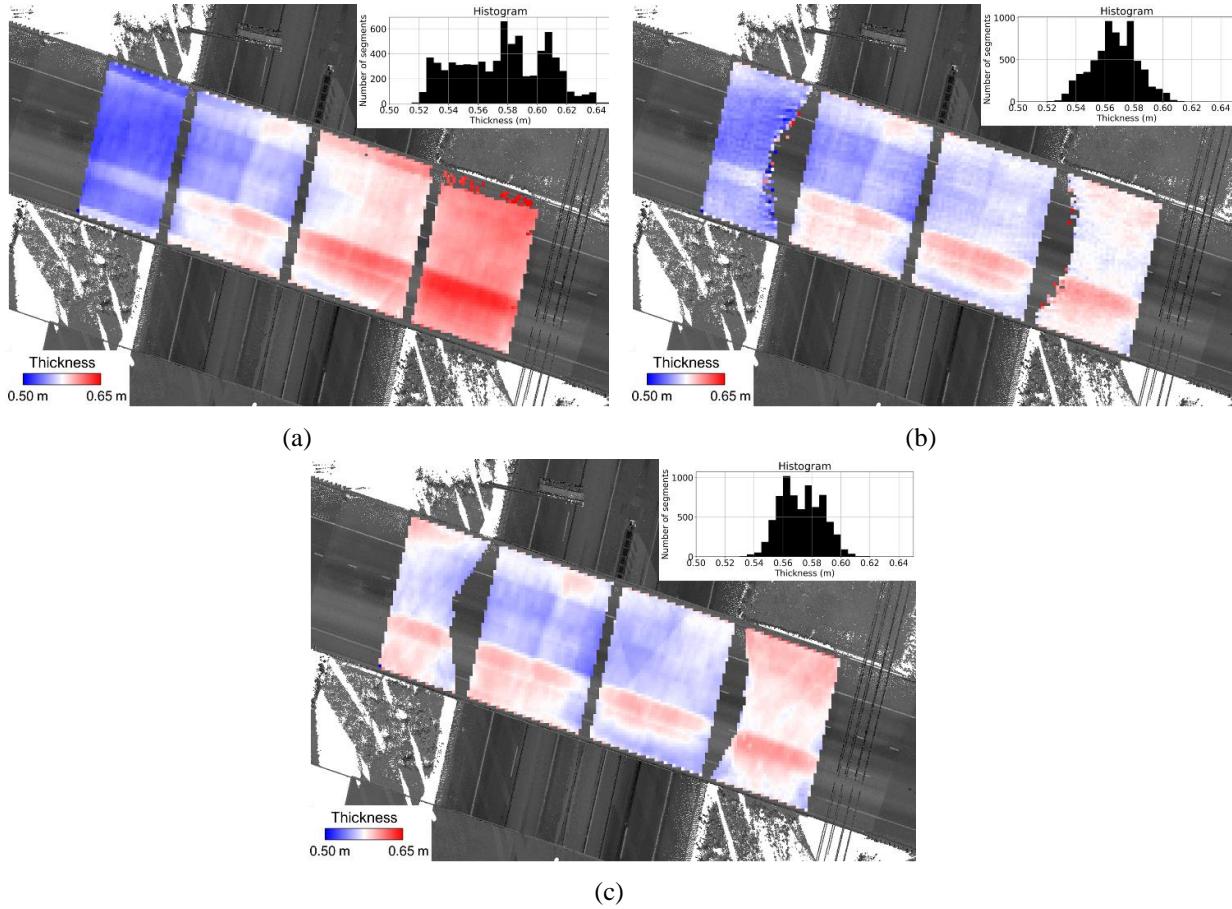


Figure 4-18 Bridge deck thickness estimates shown as a heat map using the (a) TLS, (b) PWMMS-HA, and (c) PWMMS-UHA datasets.

To quantify the similarity of thickness estimates using different modalities, a coarse registration between the TLS and MLMS datasets was performed to align the former to the reference frame of the latter. The registration accuracy was better than ± 10 cm, which is good enough for identifying corresponding surface segments. The difference between the thickness estimates at each segment were evaluated and visualized as a heat map, as can be seen in Figure 4-19. The mean, standard deviation, and RMSE of the thickness differences are reported in Table 4-4. One should note that although the PWMMS-HA point cloud is less accurate, it has a higher point density, which in turn

provides larger redundancy for plane fitting in bridge deck thickness evaluation. Therefore, the thickness evaluation accuracy for PWMMS-HA is similar to that for PWMMS-UHA. The results show that the bridge deck thickness estimates from different systems are in agreement within the 1–3 cm range. A closer investigation of the results in Figure 4-19 and Table 4-4 reveals that there is a higher level of compatibility in the thickness estimates from the PWMMS-HA and PWMMS-UHA systems. When compared to the TLS-based thickness estimates, one can observe a trend in the difference (i.e., underestimation/overestimation of thickness at the west side/east side of the bridge, respectively). This trend is the result of the Trimble scan locations leading to less-than-optimal overlap between the east and west above-bridge scans. Therefore, it is believed that the variation in the thickness estimate is in the 1 cm range. In summary, the PWMMS-HA, although with a centimeter-level accuracy LiDAR unit, has a similar performance as the PWMMS-UHA. The derived thickness from the MLMS units is comparable to that derived using TLS, with the latter being more sensitive to the scan locations. The results also reveal that the proposed segment-based thickness evaluation can handle inherent noise in the point cloud and provide reliable thickness estimates.

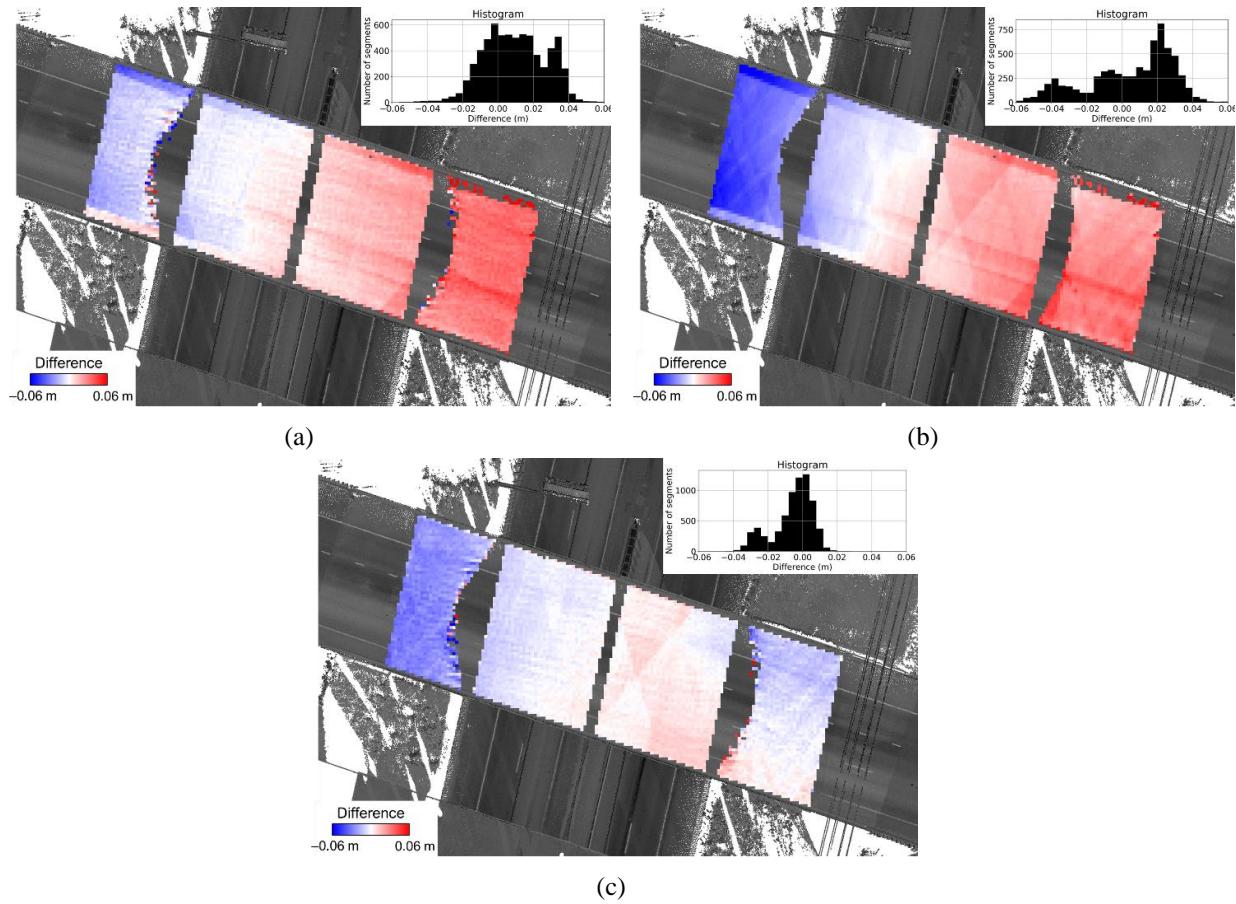


Figure 4-19 Heat map visualization of the difference in bridge deck thickness estimates between (a) TLS and PWMMS-HA, (b) TLS and PWMMS-UHA, and (c) PWMMS-HA and PWMMS-UHA.

Table 4-4 Statistics of the difference between bridge deck thickness estimates from different systems.

	TLS vs. PWMMS-HA	TLS vs. PWMMS-UHA	PWMMS-HA vs. PWMMS-UHA
Mean (m)	0.010	0.005	-0.006
Std. Dev. (m)	0.018	0.025	0.012
RMSE (m)	0.021	0.025	0.013

4.6 Conclusions and Recommendations for Future Work

This paper presented an evaluation of the performance of mapping-grade and surveying-grade mobile LiDAR systems for bridge monitoring. The performance of these systems was assessed against static laser scanners. To take full advantage of MLMS-based point clouds, a semi-automated feature-based fine registration was proposed to mitigate the negative impact of georeferencing and system calibration errors. The proposed procedure can simultaneously estimate the necessary transformation parameters for the alignment of all the derived point clouds by various sensors onboard the MLMS from different tracks. In addition, the post-alignment parametric model of the registration primitives (planar, linear, and cylindrical features) is also estimated. Bridge deck thickness was evaluated using surface segments while minimizing the impact of inherent noise in the point clouds. Field surveys were carried out over a representative bridge that had grinding conducted on it to achieve desired pavement smoothness and ride quality. The results show that the proposed feature-based fine registration effectively mitigated the impact of intermittent accessibility to a GNSS signal below the bridge. The post-registration alignment quality for the point clouds captured by the mapping-grade MLMS, surveying-grade MLMS, and TLS units is ± 1.5 cm, ± 0.7 cm, and ± 0.6 cm, respectively. Although point clouds from the mapping-grade system had a higher noise level, the evaluated bridge deck thickness was compatible to the one derived from the surveying-grade system in the range of 1 cm. The thickness estimates from both MLMS units were compatible with that derived from the TLSs. The MLMS data acquisition was conducted in five minutes, while TLSs took more than three hours. The proposed fine-registration strategy also delivered a parametric model of bridge elements, which can be used for monitoring the bridge elements over time. MLMS-based parametric models are in agreement with those from the TLSs in the 1 cm range.

Future research will focus on improving the automation level of feature extraction, as well as deriving other quantitative measures for the identification of structural issues. Moreover, larger infrastructure will be inspected to evaluate the impact of extended outage in GNSS signal reception on derived MLMS point clouds. Finally, we will be focusing on automated segmentation and parametric model representation of different structural elements of the infrastructure to aid the periodic monitoring process.

5. LIDAR-AIDED TRAJECTORY ENHANCEMENT FOR MOBILE MAPPING SYSTEMS IN GNSS-CHALLENGING URBAN ENVIRONMENTS

Abstract

Mobile LiDAR mapping systems (MLMS) carrying a sensor suite of LiDAR, GNSS, and INS are being used extensively to obtain high accuracy georeferenced point clouds. Achieving this objective is contingent on ensuring the best possible accuracy of the GNSS/INS-based vehicle trajectory as well as the system calibration parameters relating the LiDAR to GNSS/INS unit. This paper focuses on the former while assuming that the MLMS is accurately calibrated. We propose a fully automated framework to generate accurate 3D point clouds using MLMS in GNSS-challenging urban environments. Data from different sensors – GNSS, INS, and LiDAR – onboard MLMS are fused to leverage their respective information to improve the trajectory (position and orientation parameters) quality compared to that obtained from a GNSS/INS. Misalignment within LiDAR point cloud caused by trajectory quality deterioration is leveraged in this study to enhance the trajectory in GNSS-challenging environments. A two-stage approach is proposed where the first stage utilizes matched features without loop closure to perform a preliminary trajectory enhancement followed by a second stage that incorporates loop closure to mitigate any trajectory drift accumulating over time. The proposed approach is experimentally validated using three Backpack MLMS datasets acquired under different surroundings that result in varying levels of GNSS signal outages. The results demonstrate that the proposed approach enhances the trajectory to improve the point cloud alignment quality from about 2 m to about 4-8 cm.

5.1 Introduction

In recent years, mobile LiDAR mapping systems (MLMS) have gained wide popularity in their utility for several mapping applications, including agricultural field mapping, infrastructure monitoring, heritage site documentation, and asset management. The study sites and mapping applications often dictate the type of MLMS platform – space-borne, manned/unmanned aerial vehicle, terrestrial wheel-based platform (cars/trucks/unmanned ground vehicles), or pedestrian backpack platform – utilized to attain the intended mapping goal. All applications require accurate mapping products that can be ascertained based on: (a) system calibration accuracy and (b) GNSS/INS trajectory accuracy. The former encompasses intrinsic sensor(s) calibration as well as estimation of mounting parameters relating the different onboard sensors. Extensive prior research has been conducted to propose strategies that consistently improve the ease, efficiency, and performance of MLMS calibration [99–107]. The research presented in this paper focuses on the problem of improving the accuracy of GNSS/INS trajectory. For a well-calibrated MLMS, the point cloud accuracy is largely affected by environmental factors during data acquisition that cause intermittent or extended periods of GNSS signal outages. Data acquisitions prone to deteriorated GNSS/INS trajectory quality include, but are not limited to, in-canopy mapping in forests and agricultural fields, transportation corridor mapping in the presence of roadside vegetation and overhead bridges, mapping urban canyons with tall buildings, and indoor building/corridor mapping under complete obstruction of GNSS signal [108–111].

Research conducted in the domain of odometry and mapping can be classified based on the utilized sensors (cameras and/or LiDAR), integration with INS, and integration with georeferencing information from GNSS. In the remainder of this section, we will first provide a quick overview of odometry techniques relying only on cameras followed by research conducted on its fusion with INS and/or GNSS. Next, we will review odometry techniques relying solely on LiDAR sensors followed by existing literature on integrating it with INS, GNSS, and/or cameras. The shortcomings of the existing literature will be used to form the basis of the motivation for the research proposed in this paper.

Visual odometry (VO) techniques that rely solely on cameras have been researched for vehicle localization in recent years. Nister et al. [112] proposed a frame-by-frame matching of sparse point features to yield point correspondences in different images. The camera motion was then computed by minimizing the reprojection error. This approach of extracting and tracking features over a continuous stream of images was later used in other works such as MonoSLAM [113] and PTAM [114] to conduct real-time localization and mapping. Recently, ORB-SLAM was developed by Mur-Artal et al. [115] to track camera motion and create a map while utilizing trajectory loop closure information. Visual odometry has since been extended to integrate information from INS to propose visual-inertial odometry (VIO) techniques. Loosely coupled fusion of inertial data from IMU and pose (position and orientation) information from VO is generally conducted using extended and unscented Kalman filters [116–119]. Recently, tightly coupled integration for VIO has also been explored wherein the raw data from the camera and IMU are used to jointly estimate a set of state variables that define the platform dynamics [120–122]. While VO and VIO techniques can obtain locally accurate pose estimates, they are prone to accumulating large drift in long-term navigation. Furthermore, due to the absence of information from GNSS, the resultant products are referenced in a local frame instead of a global frame. In order to reduce the accumulated drift and to define a global reference frame, GNSS information is incorporated into VIO techniques. Loosely coupled integration of GNSS solution with VIO has been proposed by Mascaro et al. [123], Chu et al. [124], and Chiang et al. [125] using different optimization frameworks. Mascaro et al. [123] proposed a sliding window graph-based optimization scheme that continuously realigns the VIO pose estimates with the global reference frame from GNSS to attain global platform positioning. Chu et al. [124] utilized an Extended Kalman Filter (EKF) framework to integrate monocular camera-based navigation parameters with those from INS and translation magnitude obtained from GNSS. Chiang et al. [125] utilized an EKF to perform a loosely coupled integration of camera-derived trajectory from ORB-SLAM; position, attitude, and velocity information from INS; and absolute position from GNSS. Very recently, Cioffi and Scaramuzza [126], Liu et al. [127] and Cao et al. [128] have pursued a tightly coupled integration of GNSS, camera, and INS raw information for localization based on visual reprojection errors, IMU pre-integration errors, and raw GNSS measurement errors.

Apart from cameras, LiDAR is another sensor that has been researched for odometry and mapping. Zhang and Singh [129] proposed a novel two-stage algorithm for LiDAR odometry and mapping (LOAM) without the assistance of GNSS and INS, which would later form the basis for several improvements proposed under similar framework. Their approach used points belonging to linear and planar features extracted and matched from one LiDAR frame to the next. They reported an average positioning error of 1–2% of distance traveled as compared to 11–16% in case of trajectory

estimated using INS only. They later proposed a two-stage algorithm for visual-LiDAR odometry and mapping (V-LOAM) using image features and LiDAR range measurements, which resulted in a positioning accuracy of 0.75% of distance traveled [130]. Shan and Englot [131] developed a two-step computationally lightweight LiDAR odometry and mapping (LeGO-LOAM) that leveraged the presence of ground plane within each LiDAR frame. They reported that with a lesser computation time, LeGO-LOAM could achieve an accuracy similar to the LOAM algorithm. Chen et al. [132] integrated semantic information within the standard LOAM algorithm (SLOAM), wherein LiDAR scan frames were first labeled to train a semantic segmentation framework and then, the resultant semantically labeled point cloud was used to identify relevant features for LiDAR odometry. The resultant average positioning error was reported as 0.58% of distance traveled. Du et al. [133] utilized point-wise semantic labels to improve feature extraction and corresponding point matching for LiDAR odometry (or, S-ALOAM). They employed a deep learning-based semantic segmentation framework for point-wise semantic labeling of 3D point cloud. They achieved an average absolute position error of 0.5–1 m per 100 m and average rotational drift of about 1.10 degree per 100 m. Ji et al. [134] proposed a two-stage method for LOAM incorporated with loop-closure (L-LOAM). They conducted loop closure using global descriptors to find matching pairs of LiDAR feature segments. They achieved an average position error of 1.30 m and an average rotational drift of 1.20 degree per meter. Zhang et al. [135] proposed a two-stage feature extraction approach for LiDAR odometry and mapping. Linear and planar features were extracted according to (a) smoothness criterion in the first stage and (b) surface normal vector constraint in the second stage. However, their study did not provide quantitative measures of the resultant performance of the proposed strategy. Cong et al. [136] developed a LiDAR-SLAM approach for estimating trajectory position and orientation parameters for unmanned ground vehicle (UGV) mobile mapping system navigating in dynamic scenes. They reported trajectory estimation accuracy in the range of about 2–5 m. The literature reviewed above for trajectory estimation using only LiDAR suffers from the following drawbacks: (a) without the incorporation of loop closure, errors accumulate over time, thus rendering these approaches unreliable for longer durations of data acquisition, (b) the L-LOAM approach that integrates loop closure results in an accuracy of over 1 m, which is not suitable for high accuracy applications, (c) manual efforts are required in creating training data for approaches that rely on deep learning frameworks for semantic labeling of point clouds, and (d) in the absence of GNSS, the point clouds are not georeferenced, thus rendering it impossible to allow multi-temporal, multi-platform data fusion.

LiDAR-based odometry has been improved by fusing it with information from cameras, INS, and/or GNSS. Tang et al. [137] integrated INS and LiDAR-SLAM into a single navigation frame with a loosely coupled EKF to enable stable long-term navigation for a UGV platform. They attained an average positioning and mapping accuracy of 8–10 cm. Qian et al. [138] integrated GNSS/INS trajectory with LiDAR-SLAM technique to obtain highly accurate positioning in forest mapping applications. Their study attained a trajectory positioning accuracy of 6–13 cm, which was demonstrated to be about 70–86% improvement compared to traditional GNSS/INS integration. Chang et al. [139] proposed an integration of GNSS, INS, and LiDAR-SLAM based on graph optimization. They reported the relative position error to be 0.26% of distance traveled during GNSS outage periods of 60 seconds. Chang et al. [140] proposed an integration scheme for GNSS, INS, and LiDAR grid-based SLAM using an EKF with motion constraints. Their study

reported a positioning and mapping accuracy of 1.4–2.2 m for underground mapping with a GNSS signal outage of 320 seconds. The approaches reviewed above that rely on LiDAR-SLAM utilize frame-to-frame matching within LiDAR point clouds for trajectory estimation. When mapping scenes that are monotonic or dynamic in nature, such frame-to-frame matching becomes prone to inaccurate matches, thus resulting in inaccurate trajectory estimation.

Aboutaleb et al. [141] explored the benefits of integrating LiDAR with GNSS and INS in order to bridge GNSS outages in challenging urban environments. They proposed a 3D reduced inertial sensor system (3D-RISS) algorithm to integrate LiDAR and INS information with GNSS using a Kalman filter. For a test conducted on a total distance of 2 km with simulated GNSS outage of 80 seconds (covering 225 m), they reported the drift in trajectory position during the outage to be 3.38 m (1.5% of distance travelled) and the RMS position error of the entire trajectory was 1.7 m. In another test for a total distance of 24.4 km with an outage of about 10 minutes (covering ~2 km), they reported the trajectory position drift during the outage to be 40 m (2% of distance traveled) and the RMS position error for the whole trajectory was 16.28 m. Kukko et al. [142] proposed a graph optimization-based method to correct the post-processed GNSS/INS trajectory for positional drift encountered while collecting mobile LiDAR data under boreal forest canopy conditions. Tree stem location data was utilized in the process, which was seen to improve the tree positioning accuracy from 0.7 m to 6 cm. Li et al. [143] investigated switching navigation solutions between indoor and outdoor environments during data acquisition – GNSS/INS integration is utilized for outdoors whereas INS/LiDAR integrated navigation is employed indoors. It was observed that INS/LiDAR attained an improvement of 50% over the dead reckoning algorithm of INS in indoor environment. This review of existing literature integrating GNSS and INS with LiDAR for trajectory estimation, especially those targeted towards urban environment, indicates a decimeter to meter-level accuracy with some approaches resulting in trajectory drift as a function of the traveled distance.

The study presented in this paper aims to overcome the listed shortcomings by proposing a fully automated two-stage GNSS/INS trajectory enhancement strategy to mitigate the impact of GNSS-challenging urban environments on the accuracy of mapping products derived from MLMS. Urban surroundings are usually rich in planar surfaces, such as building façades, rooftops, beams, pillars, and ground surfaces. This study proposes a feature extraction and matching strategy that can leverage existing planar features in the surroundings to enhance the trajectory accuracy. In areas that lack in planar features, such as areas with dome structures, the proposed strategy divides the curved surface into piecewise planar segments to perform trajectory enhancement. The major contributions of the research presented in this paper are as follows:

1. The proposed approach for feature matching and extraction followed by trajectory enhancement is fully automated with no requirement for manual intervention at any stage.
2. The approach incorporates loop closure to eliminate any drift accumulating over time, thus resulting in accurate trajectory for longer duration of data acquisition. Apart from ensuring an accurate trajectory, the major focus of this paper is to enhance the accuracy of point clouds and mitigate any misalignment caused by mapping in areas with GNSS signal outages.

3. A numbered list. Based on extensive experimental testing spanning varied GNSS-challenging scenarios (signal outages ranging from 40 to 150 seconds), the proposed trajectory enhancement framework demonstrates the ability to produce point clouds with alignment precision of 4 – 8 cm, which is better than the accuracy attained by any state-of-the-art techniques dealing with similar durations of GNSS signal outages.

5.2 Methodology

The conceptual basis of the approach is that any inaccuracy in trajectory position and attitude parameters would manifest in the point cloud as discrepancies among conjugate features. We propose a two-stage approach that is broadly based on preliminary trajectory enhancement without loop closure to reduce short-term misalignment followed by a final enhancement with loop closure that eliminates any drift accumulating over time. In this section, we start by presenting the automated feature extraction and matching strategy without/with loop closure followed by the trajectory enhancement framework that takes matched features as input and produces enhanced GNSS/INS trajectory and point clouds.

5.2.1 Automated Planar Feature Extraction and Matching

In this section, we present an automated approach for feature extraction and matching that is used for drift corrections within the trajectory enhancement optimization model. As mentioned earlier in Section 4.1, since this study focuses on urban environments, we rely on planar features in the surroundings to achieve our objective. The main challenge encountered while extracting and matching adequate features for trajectory enhancement stems from the variable level of misalignment within derived 3D point clouds. A data acquisition mission spanning open sky areas with intermittent dense canopy cover, overhead bridges, indoor segments, and other similar GNSS-challenging segments results in variable accuracy of trajectory parameters and thus, a variable level of misalignment within derived 3D point clouds. Point cloud misalignment in areas with GNSS signal outages are further impacted by the trajectory drift accumulating over time elapsed between revisits of the same location.

The planar feature extraction and matching strategy proposed in this paper is designed under the hypothesis that planar segments that are spatially close to each other and have similar direction of normal vectors represent the same feature. The spatial and directional proximity thresholds for planar feature matching are user-defined based on the level of misalignment expected in the point cloud because of trajectory inaccuracies. Using constant spatial/directional proximity thresholds for the entire point cloud might result in incorrect matches owing to the variable trajectory quality in different areas – strict thresholds would result in omitted conjugate feature matches whereas relaxed thresholds would result in wrongly matched features in areas with distinct features that satisfy the similarity criteria. A schematic illustration of a possible case that might result in incorrectly matched features is shown in Figure 5-1, where the pairs of yellow, blue, and gray highlighted planar segments indicate the incorrect matches. Omitted conjugate feature matches hinder the ability to enhance the trajectory since the misalignment information is not portrayed in the observation equations while wrongly matched features result in incorrect trajectory enhancement results. To overcome the stated challenges with feature matching, a first round of feature extraction and matching is conducted without loop closure to be utilized for preliminary

trajectory enhancement. Next, feature matching with loop closure is conducted on intermediately enhanced point cloud to estimate the final trajectory. In the remainder of this section, we discuss the planar feature extraction strategy followed by matching strategies without/with loop closure.

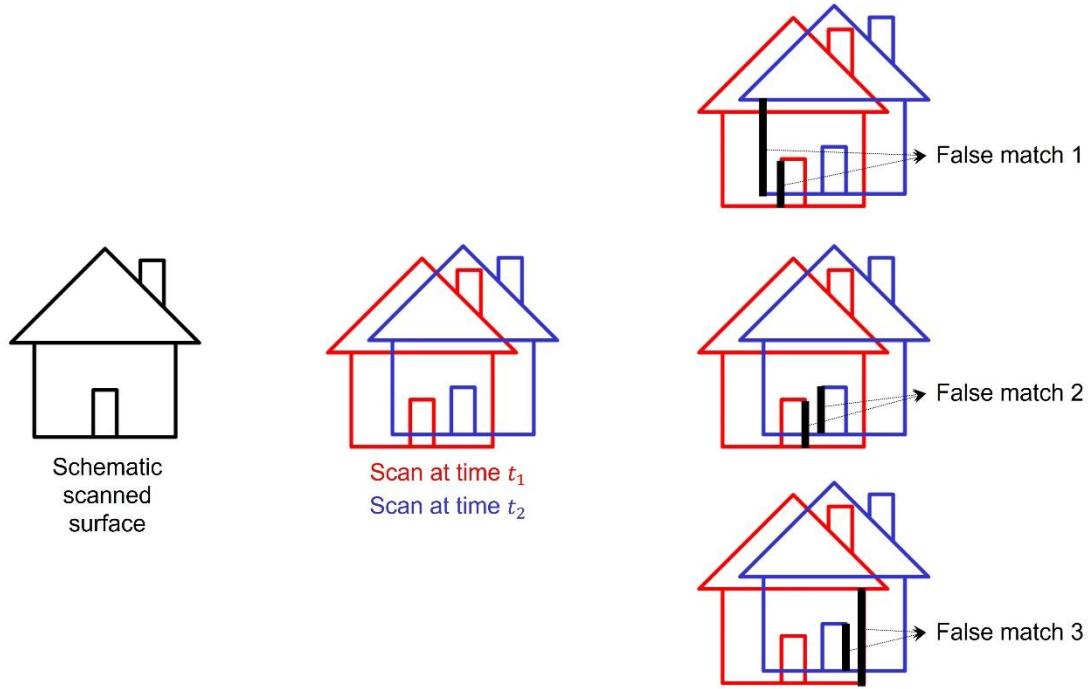


Figure 5-1 Schematic illustration of possible wrong matches in case of large misalignment and similar appearance of planar surfaces (false matches 1, 2, and 3 show possible wrong matches between planar surfaces)

Planar Feature Extraction and Matching Without Loop Closure

The flowchart for planar feature extraction and matching without loop closure for trajectory enhancement is provided in Figure 5-2a. Planar feature extraction for trajectory enhancement is designed under the hypothesis that the trajectory drift and thereby, misalignment within the point cloud over short intervals of time is not substantial. The point cloud is first partitioned into short time intervals (say, N -second intervals) within which there is minimal misalignment due to limited trajectory drift. Planar features are then extracted within each point cloud partition. Starting from randomly generated seed points for the point cloud partition, a seed region is first established corresponding to each seed point by extracting the closest N_{seed} points. A principal component analysis (PCA) is carried out for the seed region to determine whether it is a planar, linear, or rough region, as proposed by Habib and Lin [144]. Each planar seed region is then used to obtain the best-fitting plane and finally, conduct a region growing to include neighboring points as part of the planar feature if their normal distance from the best-fitting plane is less than a threshold.

The planar features extracted from point cloud partitions are then matched to identify conjugate features without loop closure. High platform dynamics within N -second durations might result in discrepancy within each point cloud partition that could result in a planar surface segmented as two different planar features during the extraction. So, an intra-partition feature matching is first conducted to identify conjugate planar features within each partition, as depicted in Figure 5-2b. Starting with the most reliable planar feature (based on the number of points and RMSE of plane fitting) within a point cloud partition (hereby denoted as template plane), the remaining planar features within the same partition (or, candidate planes) are tested for their spatial and directional proximity to the template. Spatial proximity of a template and candidate plane is checked by identifying whether the intersection of the normal vector of the former with the latter lies within a distance threshold. The directional proximity is checked by computing the angle between the normal vectors of the two planes. This process is repeated for all the extracted planar features within each partition to establish intra-partition feature matches. Next, an inter-partition feature matching is conducted between each pair of *consecutive* point cloud partitions. It follows the same spatial and directional proximity criteria as discussed previously for intra-partition feature matching with the only difference being that a template plane is matched with candidate planes that belong to the point cloud partition that is in succession to that of the template plane. A sample of feature extraction and matching results without loop closure is shown in Figure 5-3 for a narrow straight indoor corridor, which appears to be irregular in the point cloud due to trajectory drift. The point cloud is colored by the feature ID assigned after feature matching without loop closure, wherein it can be seen that planar wall façades and ceiling surfaces that are distorted from planarity over time due to drifting trajectory are identified and designated as separate features to assist in trajectory enhancement. However, different versions of the same planar surface with substantial misalignment between location revisits are not matched in this step. The resultant feature matches without loop closure are incorporated in the optimization model for trajectory enhancement to mitigate short-term trajectory drift. One should note that this approach for feature matching could also cause an undersegmentation of undulating/curved surfaces. However, in urban areas, such occurrences are minimal and furthermore, this issue can be mitigated through a careful choice of spatial and directional proximity thresholds. One of the experimental datasets in this paper will focus on demonstrating the performance of the algorithm in the presence of undulating/curved surfaces.

Planar Feature Extraction and Matching with Loop Closure

Feature extraction and matching with loop closure is conducted on the point cloud derived after the first stage of trajectory enhancement. The flowchart for the proposed approach is shown in Figure 5-2c. Planar features are first extracted from the partitioned intermediate point cloud using the same approach as detailed earlier. Next, an intra-partition feature matching is conducted for the extracted planar features similar to the case without loop closure. Finally, an inter-partition feature matching is conducted among *all possible point cloud partition pairings* unlike the case without loop closure where the inter-partition matching was conducted only among *consecutive point clouds partitions*. A sample of feature extraction and matching results with loop closure is shown in Figure 5-3, where the points are colored by feature ID. Different versions of the same wall between location revisits (that were earlier not matched due to misalignment ranging up to 2.5 m) are now matched as the same feature.

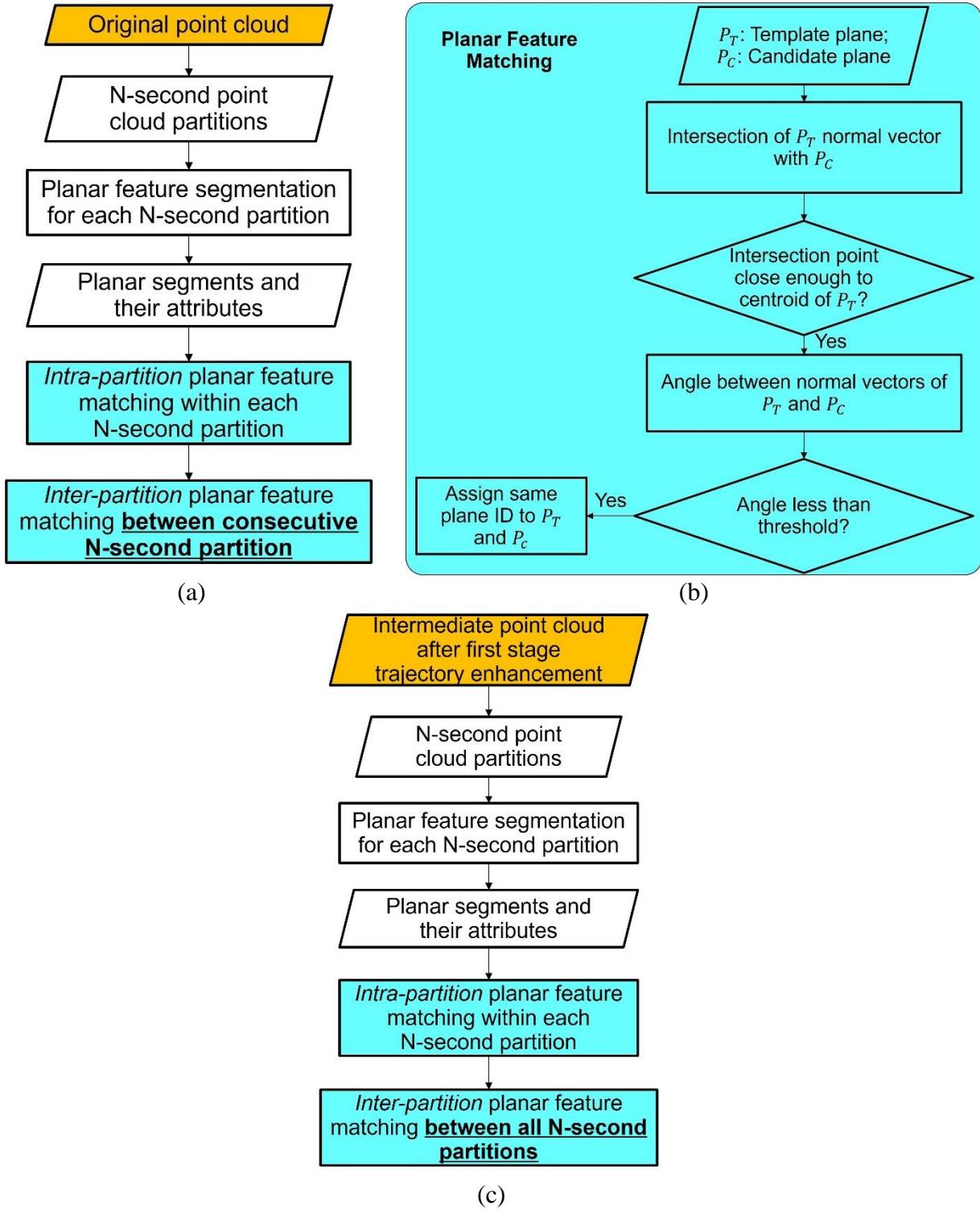


Figure 5-2 (a) Flowchart for feature extraction and matching without loop closure; (b) planar feature matching flowchart; and (c) flowchart for feature extraction and matching with loop closure

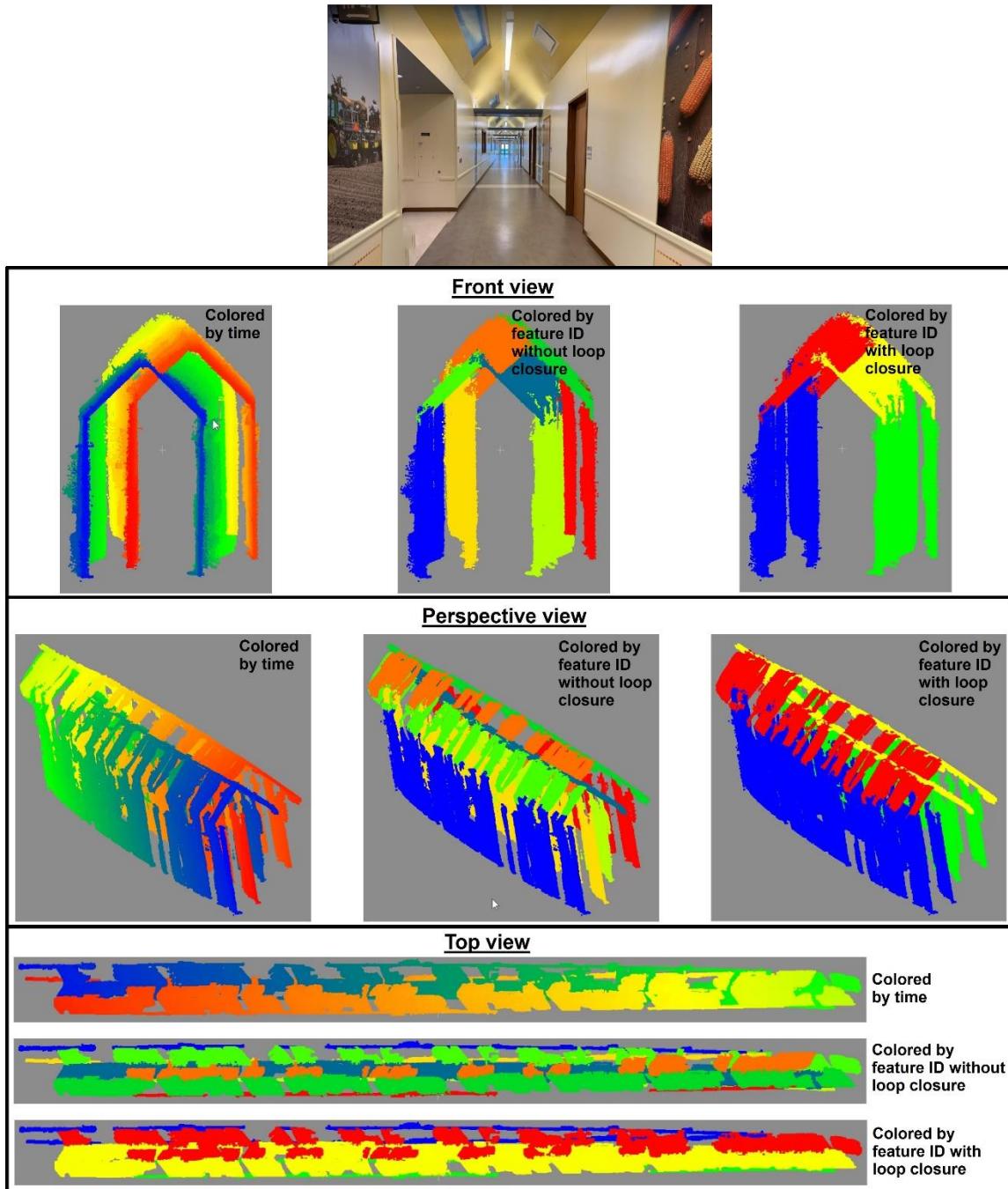


Figure 5-3 Sample image and real point cloud along a narrow indoor corridor colored by time (total of 120 secs) and feature ID assigned after conducting feature matching without/with loop closure

5.2.2 Trajectory Enhancement Framework

The conceptual basis of the proposed trajectory enhancement framework is to estimate corrections to position and orientation parameters using a least squares adjustment (LSA) model wherein the extracted and matched planar features are used as input. We start by introducing the point positioning equation to derive the mapping frame coordinates for any LiDAR point using the system mounting and trajectory parameters, as given in Equation 5-1. For any LiDAR point, I , captured at time, t , its mapping (m) frame coordinates are a function of: (i) trajectory position and orientation parameters relating the IMU body frame at the corresponding time, $b(t)$, and mapping frame – where the position is denoted by $r_{b(t)}^m$ and rotation matrix based on the orientation angles is denoted by $R_{b(t)}^m$; (ii) LiDAR mounting parameters relating the laser unit (lu) and body (b) frames – where the lever arm is denoted by r_{lu}^b and boresight matrix is denoted by R_{lu}^b ; and (iii) the laser unit frame coordinates of the point, denoted by $r_I^{lu(t)}$. The trajectory enhancement procedure estimates corrections to position and orientation parameters that results in refined coordinates of the LiDAR points to produce well-aligned point clouds based on the given point positioning equation.

$$r_I^m(t) = f(r_{b(t)}^m, R_{b(t)}^m, r_{lu}^b, R_{lu}^b, r_I^{lu(t)}) = r_{b(t)}^m + R_{b(t)}^m r_{lu}^b + R_{b(t)}^m R_{lu}^b r_I^{lu(t)} \quad \text{Equation 5-1}$$

Figure 5-4 shows the flowchart for the proposed trajectory enhancement procedure. Block 1 in Figure 5-4 deals with the mathematical modeling of trajectory reference points to be used in the trajectory enhancement optimization model, as will be discussed in Section 4.2.1. Block 2 then takes the trajectory reference points and matched features as input within the optimization model that minimizes the discrepancy between conjugate features to produce an enhanced trajectory leading to well-aligned point clouds, as detailed further in Section 4.2.2.

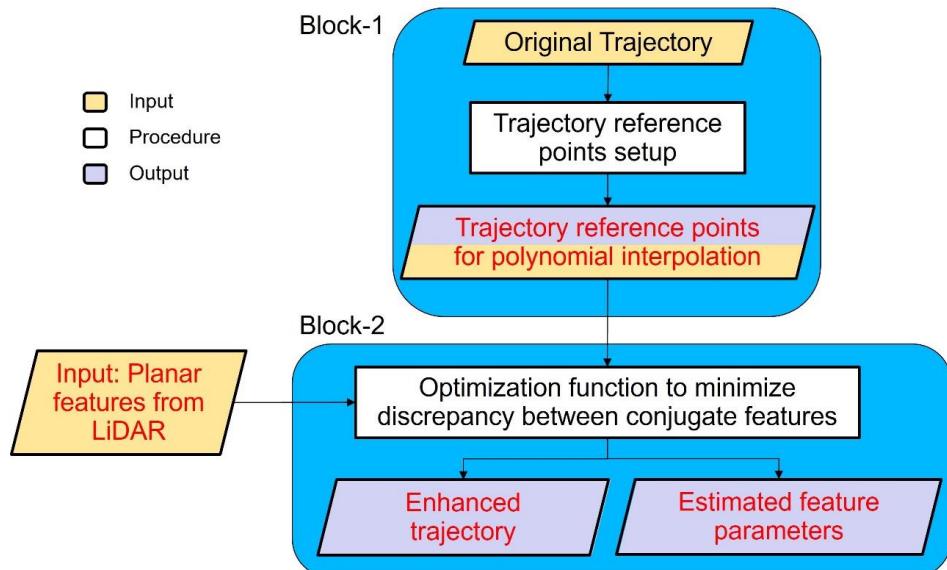


Figure 5-4 Flowchart for trajectory enhancement model that takes matched planar features from LiDAR as input to produce enhanced GNSS/INS trajectory

5.2.2.1 Trajectory Corrections Modeling

In order to avoid over-parametrization in the LSA model caused by trying to solve for the trajectory parameters correction at each laser beam firing timestamp, we deal with a down sampled trajectory – denoted henceforth as trajectory reference points – within the LSA model while assuming that the MLMS platform has a relatively smooth trajectory with moderate dynamics. The trajectory reference points are obtained by down sampling the original high frequency (say, 100–200 Hz) trajectory parameters to a user-defined lower frequency (or, a down sampling time interval Δt). Thus, the proposed trajectory enhancement optimization model estimates corrections to the trajectory reference points, which are then utilized to model corrections to the trajectory parameters at specific timestamps corresponding to laser beam firings. In this study, the trajectory position/attitude parameters are denoted as $(X_0, Y_0, Z_0, \omega, \phi, \kappa)$, where (X_0, Y_0, Z_0) define the vector $r_{b(t)}^m$ and (ω, ϕ, κ) define the rotation matrix $R_{b(t)}^m$. The corresponding corrections to trajectory parameters are denoted as $(\delta X_0, \delta Y_0, \delta Z_0, \delta \omega, \delta \phi, \delta \kappa)$. Corrections to each of the six trajectory position/attitude parameters are modeled separately as p^{th} -order polynomial in time, where the coefficients of this polynomial are expressed as a function of corrections for the n neighboring trajectory reference points. The down sampling time interval, polynomial order, and number of neighboring trajectory reference points are user-defined based on the nature of platform dynamics. Figure 5-5 shows neighboring trajectory reference points, where one should note that each trajectory point at any timestamp, t , has an associated relative timestamp, t' , which is designated to avoid numerical instability. The relative timestamp is evaluated with respect to the first of the set of n neighboring trajectory reference points.

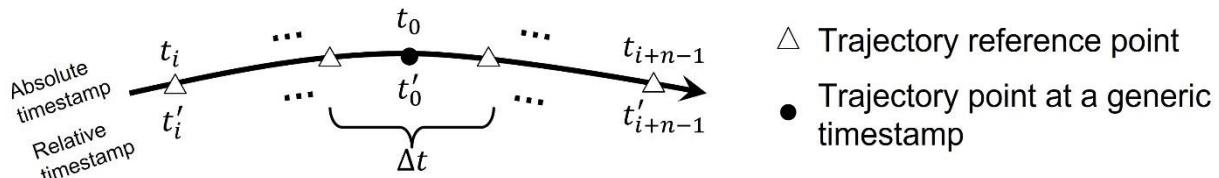


Figure 5-5 Down sampled trajectory reference points (down sampling time interval Δt) used for trajectory enhancement: t_i to t_{i+n-1} and t'_i to t'_{i+n-1} denote the absolute and relative timestamps of the n neighboring trajectory reference points for a generic timestamp t_0

For n neighboring trajectory reference points, the position/attitude corrections, denoted generically as $\delta\theta(t'_i)$, are modeled as p^{th} -order polynomial in time as given in Equation 5-2. Here, to avoid possible numerical instabilities, t' denotes the timestamp relative to the first trajectory reference point in the n -tuple. $C_{\theta_0}, C_{\theta_1}, \dots, C_{\theta_p}$ denote the coefficients of the correction polynomial for the parameter θ and are treated as unknowns that are expressed such that the corrections at the trajectory reference points satisfy the assumed correction polynomials. Thus, the correction polynomial coefficients can be derived using least squares adjustment (LSA) as given in Equation 5-3, if $n \geq p + 1$. Then, the correction to the position/attitude parameter at any generic relative timestamp, t'_0 , can be derived using Equation 5-4 based on its n neighboring trajectory reference points (indexed from i to $i + n - 1$) with corresponding relative timestamps t'_i to t'_{i+n-1} . In summary, the corrections to any position/attitude parameter can be modeled as a p^{th} -order

polynomial function of its own timestamp and the coefficients of this polynomial are expressed as a function of the corrections and timestamps for the n neighboring trajectory reference points. The corrections for the trajectory reference points are the unknowns in the trajectory enhancement process. One should note that p and n could be selected differently for each of the six trajectory position and orientation parameters.

$$\begin{pmatrix} \delta\theta(t'_i) \\ \delta\theta(t'_{i+1}) \\ \vdots \\ \delta\theta(t'_{i+n-1}) \end{pmatrix} = \underbrace{\begin{bmatrix} 1 & t'_i & t'^2_i & \cdots & t'^p_i \\ 1 & t'_{i+1} & t'^2_{i+1} & \cdots & t'^p_{i+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t'_{i+n-1} & t'^2_{i+n-1} & \cdots & t'^p_{i+n-1} \end{bmatrix}}_A \begin{pmatrix} C_{\theta_0} \\ C_{\theta_1} \\ \vdots \\ C_{\theta_p} \end{pmatrix} \quad \text{Equation 5-2}$$

$$\begin{pmatrix} C_{\theta_0} \\ C_{\theta_1} \\ \vdots \\ C_{\theta_p} \end{pmatrix} = \underbrace{(A^T A)^{-1} A^T}_M \begin{pmatrix} \delta\theta(t'_i) \\ \delta\theta(t'_{i+1}) \\ \vdots \\ \delta\theta(t'_{i+n-1}) \end{pmatrix} \quad \text{Equation 5-3}$$

$$\delta\theta(t_0) = [1 \quad t'_i \quad t'^2_i \quad \cdots \quad t'^p_i] \begin{pmatrix} C_{\theta_0} \\ C_{\theta_1} \\ \vdots \\ C_{\theta_p} \end{pmatrix} \quad \text{Equation 5-4}$$

$$= [1 \quad t'_i \quad t'^2_i \quad \cdots \quad t'^p_i] M \begin{pmatrix} \delta\theta(t'_i) \\ \delta\theta(t'_{i+1}) \\ \vdots \\ \delta\theta(t'_{i+n-1}) \end{pmatrix}$$

5.2.2.2 Optimization Model for Trajectory Enhancement

The proposed optimization model for trajectory enhancement estimates the corrections to the trajectory reference points using LSA that minimize the normal distance of LiDAR points to their corresponding features. The optimization model for trajectory enhancement is comprised of two types of observation equations – (a) equations based on LiDAR feature points and (b) equations incorporating prior trajectory information. The first type of observation equations, as given in Equation 5-5, minimizes the normal distance, $\text{nd}(I, t, \Phi_k)$, for each LiDAR point, I , captured at time, t , and belonging to the k^{th} planar feature whose parameters are represented as Φ_k . Planar feature parameters (Φ) constitute the normal vector (w_x, w_y, w_z) and distance of the plane from the origin (d). The normal distance of LiDAR point from the corresponding plane is a function of the feature parameters, Φ_k and mapping frame coordinates of point, I , which in turn are a function of the corrections for n neighboring trajectory reference points with respect to the time of capture, t . Each observation equation is assigned an a-priori variance, denoted by σ_{nd}^2 , to incorporate range-based adaptive weights to each of the LiDAR observations since points captured close to the LiDAR unit are expected to conform to the planar surface more precisely than further points that are prone to more noise. Assuming that σ_{ref} denotes the nominal expected accuracy for the LiDAR points up to ρ_{max} distance from the LiDAR unit based on the manufacturer specifications of the involved sensors, the adaptive variance is given by Equation 5-6, indicating that points with range less than ρ_{max} have a constant variance of σ_{ref}^2 whereas those with larger range have a linearly

increasing standard deviation, which results in a decreasing weight. A plot of standard deviation σ_{nd} with respect to the range of captured point ρ_i is shown in Figure 5-6. While solving for Φ_k , three out of the four parameters used to define a plane are designated to be independent during LSA depending on the orientation of the planar surface – w_x , w_y , or w_z is fixed to unity for a planar feature whose normal vector is predominantly parallel to X -axis, Y -axis, or Z -axis, respectively.

$$\underset{\delta\theta_b^m(t_{ref}), \Phi_k}{\operatorname{argmin}} \sum_{\text{LiDAR feature points}} \frac{(nd(I, t, \Phi_k))^2}{\sigma_{nd}^2} \quad \text{Equation 5-5}$$

$$\sigma_{nd}^2 = \left(\frac{\max(\rho_{max}, \rho_i)}{\rho_{max}} \times \sigma_{ref} \right)^2 \quad \text{Equation 5-6}$$

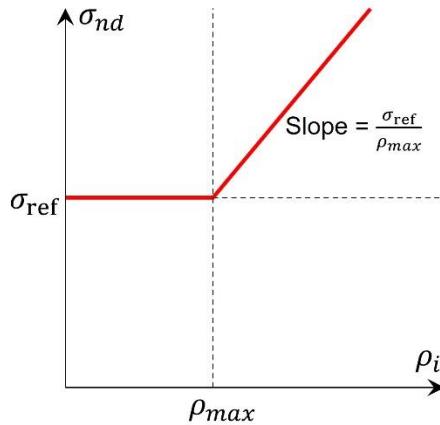


Figure 5-6 Range-based adaptive variance for trajectory enhancement: standard deviation (σ_{nd}) as a function of the range (ρ_i) of captured points

A second type of observation equations are included within the LSA model to incorporate prior information about trajectory parameters. The initial trajectory along with the estimation accuracy for the parameters is generated by post-processing GNSS and INS information. These reported initial accuracy values are used to assign adequate weights to the estimated corrections to the trajectory parameters while conducting trajectory enhancement. This is achieved by including observation equations, as given in Equation 5-7, to minimize the change in position and orientation parameters for a total of N_t trajectory reference points depending on the reported standard deviation for each parameter. Additionally, the change in instantaneous linear velocity (v_i) – as computed according to Equation 5-8 between two consecutive trajectory reference points – is minimized based on the velocity accuracy (σ_v) reported while generating the original post-processed trajectory. This is expressed in Equation 5-9, where v_i and $v_{i_{\text{corrected}}}$ denote the instantaneous velocity computed using the i^{th} and $(i + 1)^{th}$ trajectory reference points before and after applying the estimated corrections to the position parameters. One should note that the inclusion of a-priori trajectory knowledge (Equation 5-7 and Equation 5-9) to the LSA model ensures that corrections to trajectory reference points are not estimated if there are no LiDAR feature points to assist in their estimation. In the proposed approach, the correlations between

position, attitude, and velocity parameters are ignored due to the absence of its information from GNSS/INS post-processing. However, this assumption is not expected to hinder the performance of the proposed algorithm, as demonstrated widely by existing loosely coupled integration techniques for navigation that have been developed while ignoring any correlations.

$$\underset{\delta\theta_{b(t_i)}^m}{\operatorname{argmin}} \sum_{i=1}^{N_t} \frac{(\delta\theta_{b(t_i)}^m)^2}{\sigma_{\theta_i}^2} \quad \text{Equation 5-7}$$

$$v_i(r_{b(t_i)}^m, r_{b(t_{i+1})}^m, \Delta t) = \frac{|r_{b(t_{i+1})}^m - r_{b(t_i)}^m|}{\Delta t} \quad \text{Equation 5-8}$$

$$\underset{\delta r_{b(t_i)}^m, \delta r_{b(t_{i+1})}^m}{\operatorname{argmin}} \sum_{i=1}^{N_t-1} \frac{1}{\sigma_v^2} \left(v_{i,\text{corrected}}(r_{b(t_i)}^m, \delta r_{b(t_i)}^m, r_{b(t_{i+1})}^m, \delta r_{b(t_{i+1})}^m, \Delta t) - v_i(r_{b(t_i)}^m, r_{b(t_{i+1})}^m, \Delta t) \right)^2 \quad \text{Equation 5-9}$$

With the knowledge of the LSA model, we can determine the total number of unknowns and observation equations involved in trajectory enhancement for dataset(s) with N_t as the number of trajectory reference points and np_p as the number of LiDAR points captured over N_p planar surfaces. The unknowns include $6N_t$ trajectory reference point corrections to position and orientation parameters and $3N_p$ planar feature parameters. The first set of observation equations minimizing the normal distance of LiDAR points from corresponding planar features results in one equation per LiDAR point and thus, a total of np_p equations. Additionally, incorporating prior trajectory information results in a total of $6N_t + (N_t - 1)$ observation equations. LSA for trajectory enhancement is conducted iteratively till the change in root-mean-square (RMS) of normal distances of LiDAR points to corresponding planar features is less than a pre-defined threshold. After convergence, a RANSAC-based outlier points removal [145] is conducted for each planar feature and a last adjustment iteration is conducted to arrive at the final results. One should note that RANSAC-based outlier removal is not conducted before convergence since trajectory drifts result in substantial misalignments within the planar features that might result in the omission of meaningful feature points as outliers. Furthermore, the feature matching process could be repeated after each LSA iteration during trajectory enhancement. However, this would be computationally expensive and time-consuming. The two stages of feature matching are deemed sufficient to ensure adequate feature matching for accurate trajectory enhancement.

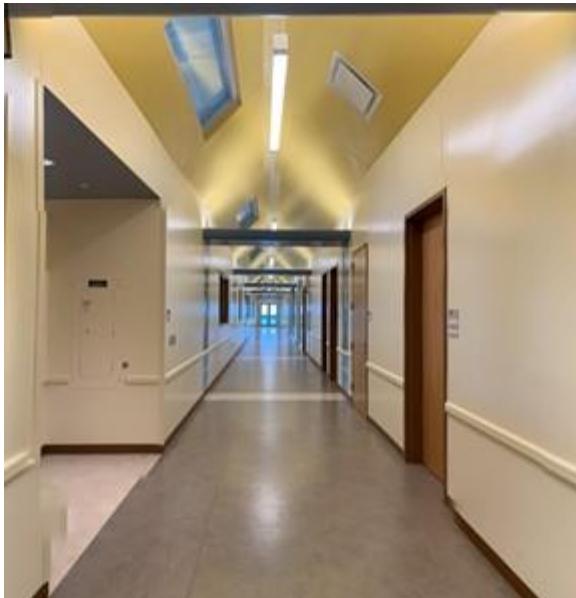
5.3 Experimental Results

This section presents trajectory enhancement results obtained for three datasets acquired using a Backpack MLMS in different GNSS-challenging urban scenarios as shown in Figure 5-7 – (a) underpass data acquisition, (b) indoor mapping, and (c) indoor mapping of dome facility using a crane bucket. A custom-built Backpack MLMS (shown in Figure 5-8) is used in this research, which comprises a Velodyne VLP-16 Hi-Res LiDAR and a Sony α7R II 43.6 MP full-frame camera with a 35 mm lens. A Novatel SPAN-CPT GNSS/INS is used for direct georeferencing of the LiDAR and camera. The Velodyne VLP-16 Hi-Res LiDAR consists of 16 laser beams with horizontal and vertical fields of view of 360° and 20°, respectively. It captures 300,000 points per

second and has a range accuracy of ± 3 cm [146]. In case of no GNSS signal outages, the Novatel SPAN-CPT GNSS/INS provides a post-processing accuracy of $\pm 2\text{--}5$ cm for position, 0.025° for roll/pitch angles, and 0.080° for attitude angle [147]. In this study, the GNSS/INS trajectory post-processing and accuracy report was derived using Inertial Explorer – a commercial software by Novatel [148]. The expected accuracy of the point cloud was estimated based on the individual sensor specifications using the LiDAR Error Propagation Calculator developed by Habib et al. [149]. The calculator suggests an accuracy of ± 5 cm at a range of 100 m in the absence of GNSS signal outages. In order to reconstruct accurate point clouds from the MLMS, rigorous system calibration is required to estimate the mounting parameters – lever arm and boresight angles – relating the onboard LiDAR sensor with the GNSS/INS unit. In this study, the Backpack MLMS was calibrated using the approach proposed by Ravi et al. [107].



(a)



(b)



(c)

Figure 5-7 Study sites used for trajectory enhancement: (a) Dataset 1: underpass data acquisition; (b) Dataset 2: indoor mapping; (c) Dataset 3: indoor mapping of dome facility using a crane bucket

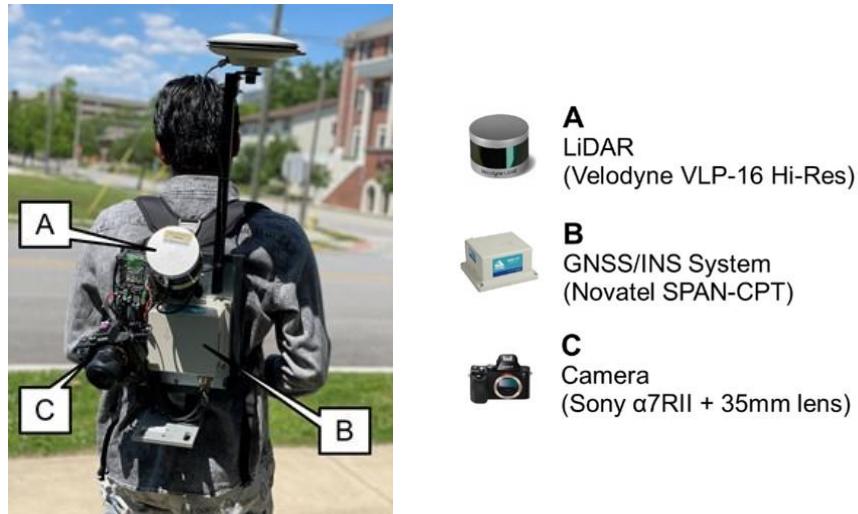


Figure 5-8 Backpack MLMS used in this study

Table 5-1 provides the characteristics of the three datasets along with the environmental factors causing trajectory quality deterioration. The feature extraction and matching thresholds as well as trajectory enhancement parameters used for each dataset are also provided in Table 5-1. The spatial and directional proximity thresholds for feature matching without loop closure are determined based on an inspection of the original point clouds whereas those for feature matching with loop closure are determined based on the inspection of the point cloud after preliminary trajectory enhancement. The trajectory enhancement results presented in this section are assessed both qualitatively and quantitatively. Narrow profiles (0.5 m wide) oriented in different directions are extracted from the point cloud to qualitatively illustrate the improvement in point cloud alignment after trajectory enhancement. While all the profiles are portrayed before and after trajectory enhancement, one profile within each dataset is used as a sample to demonstrate the alignment quality at three stages – before trajectory enhancement, after trajectory enhancement without loop closure, and after trajectory enhancement with loop closure – to illustrate the performance of the proposed two-stage approach. The improvement in point cloud alignment quality is quantified by reporting the statistical measures of normal distances of the LiDAR points to their corresponding best-fitting planes before and after trajectory enhancement. Next, the magnitude of estimated corrections to trajectory parameters is visualized for the enhanced high-frequency trajectory to illustrate the introduced trajectory modification in different areas to improve point cloud alignment. Further, a plot of the magnitude of trajectory positional parameters corrections with respect to the reported standard deviation during post-processing is also shown to highlight the relationship between the two. Finally, statistical measures describing the difference between original and enhanced trajectory position/attitude parameters are reported to quantify the modification.

Table 5-1 Data acquisition specifications and parameters used for each procedure employed during trajectory enhancement

		Dataset 1	Dataset 2	Dataset 3
Duration		18 minutes	8 minutes	12 minutes
Factors causing GNSS/INS trajectory quality deterioration		<ul style="list-style-type: none"> Underpass data acquisition Intermittent periods (~40 secs) of GNSS signal outages 	<ul style="list-style-type: none"> Indoor mapping Prolonged periods (~140 secs) of obstructed GNSS signal 	<ul style="list-style-type: none"> Indoor mapping of dome facility using crane bucket Non-trivial trajectory dynamics compared to walking Prolonged periods (~150 secs) of obstructed GNSS signal
Feature matching without loop closure	N-second point cloud interval		2 secs	
	Spatial proximity threshold	0.30 m	0.30 m	0.20 m
	Directional proximity threshold	10°	5°	2°
Feature matching with loop closure	N-second point cloud interval		2 secs	
	Spatial proximity threshold	0.30 m	0.80 m	0.30 m
	Directional proximity threshold	10°	5°	2°
Preliminary and final trajectory enhancement	Down sampling time interval for trajectory reference points		1 sec	
	Order of trajectory parameters correction polynomial		2	
	Number of neighboring trajectory reference points		3	

5.3.1 Dataset 1: Underpass Data Acquisition

The entire point cloud colored by height is shown in Figure 5-9 along with the Backpack trajectory overlaid in pink. While most of the trajectory is under open sky, the portion under the bridge has obstructed GNSS signals, thus resulting in lower quality trajectory. The profiles used for qualitative assessment of point cloud alignment before and after trajectory enhancement are also shown in Figure 5-9. Out of these, Profile 1 is used to demonstrate the point cloud alignment before trajectory enhancement and after each stage of trajectory enhancement (without/with loop closure). The points for Profile 1 are shown in Figure 5-10 colored by time of capture. This profile spans a

cross-section of the underpass area with mechanically stabilized earth (MSE) walls on either side and a pillar in the center. There is a road surface and curb on either side of the pillar, as depicted in Figure 5-10. Areas 1, 2, and 3 within Profile 1 are closely inspected to illustrate the improvement in point cloud alignment through the two stages of trajectory enhancement. For Areas 1 and 3, the MSE walls have a large misalignment in the original point cloud, which is reduced to produce better plane definitions after trajectory enhancement without loop closure. Finally, trajectory enhancement with loop closure produces a well-aligned point cloud for these walls. Area 2 shows the central pillar, where the opposite vertical faces of the pillar in the original point cloud are close within the range of misalignment of other points captured over the same surface. Further, the bridge deck appears to be out of level on either side of the pillar due to trajectory inaccuracy. After trajectory enhancement without loop closure, the points belonging to the two vertical faces of the pillar are better delineated and the level nature of the bridge deck is better represented than in the original point cloud. After trajectory enhancement with loop closure, the points along the pillar faces as well as bridge deck are well-aligned.

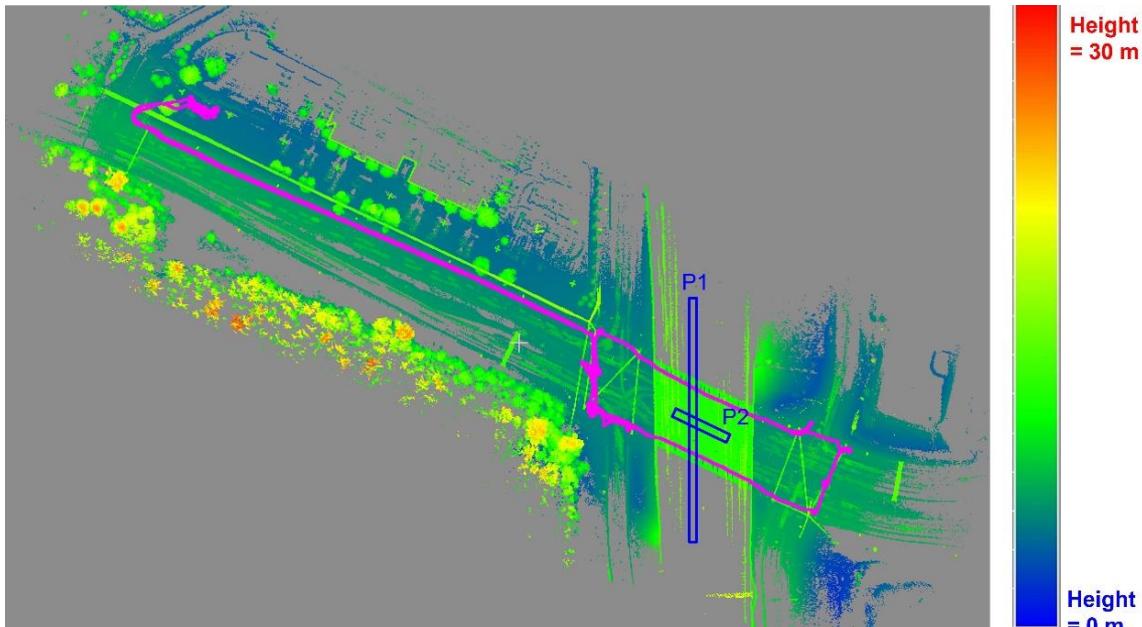
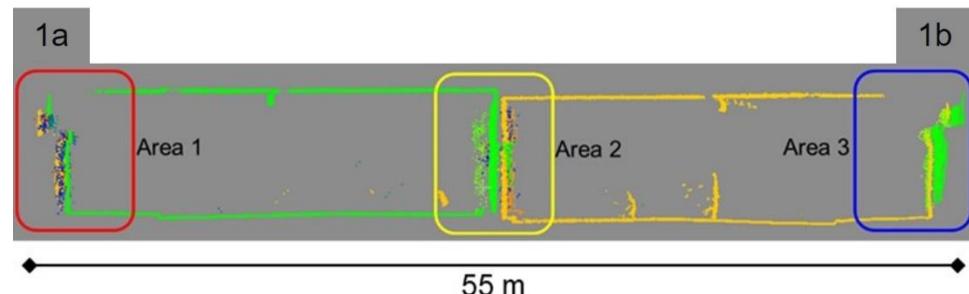
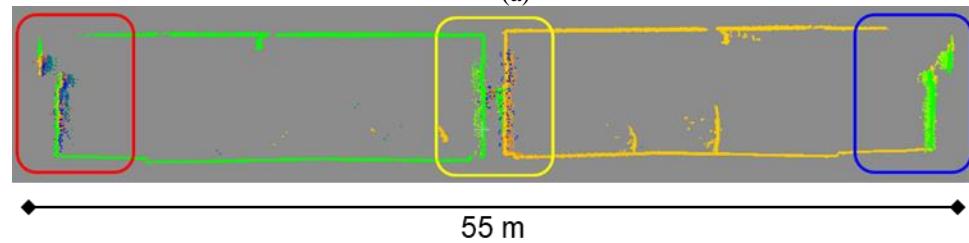


Figure 5-9 Dataset 1: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results



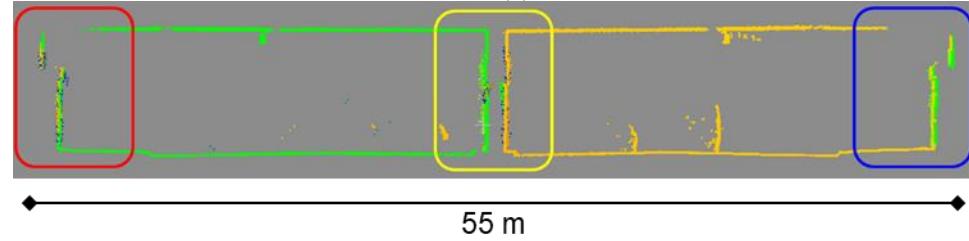
Profile 1: Before trajectory enhancement

(a)



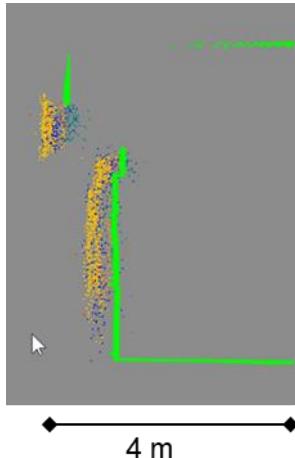
Profile 1: After trajectory enhancement without loop closure

(b)

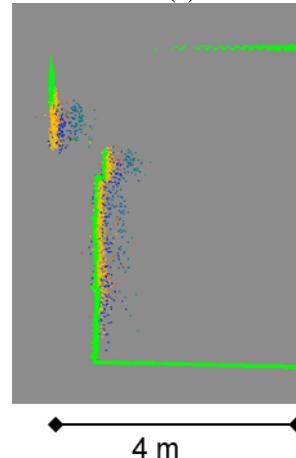


Profile 1: After trajectory enhancement with loop closure

(c)



Area 1: Before trajectory enhancement



Area 1: After trajectory enhancement without loop closure



Area 1: After trajectory enhancement with loop closure

(d)

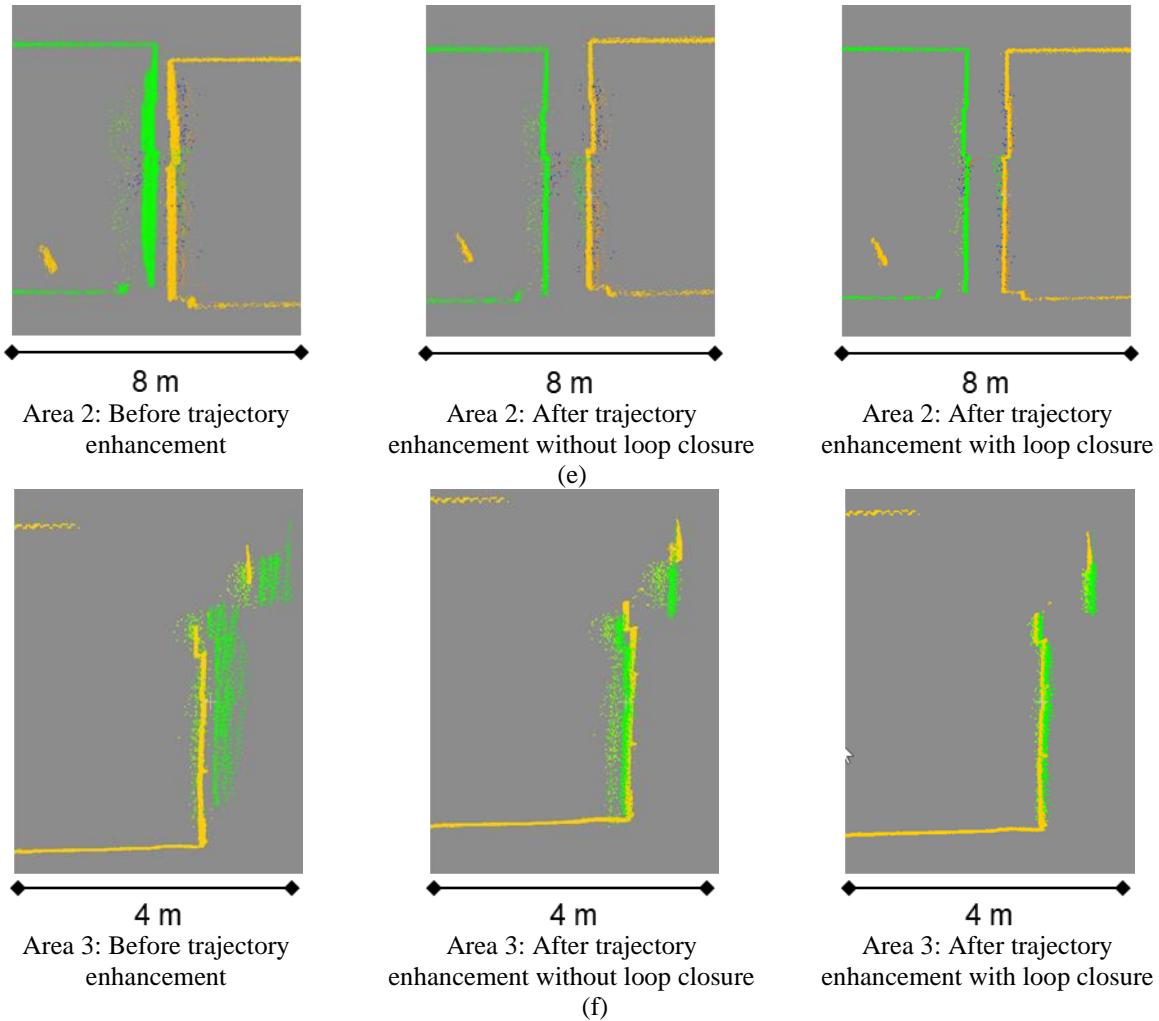


Figure 5-10 Dataset 1: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure

The point cloud for Profile 2 (D1-P2 in Figure 5-9) is depicted in Figure 5-11 to assess the alignment quality along the walking direction. The side view indicates a substantial improvement in alignment of the pillars after trajectory enhancement. The bridge beams on the top (oblique with respect to the original side view) are zoomed-in to show the impact of trajectory enhancement on defining the beam structure adequately within the resultant point cloud. Table 5-2 reports the statistical measures (mean, standard deviation, RMSE, and maximum) of normal distances of the LiDAR feature points to their corresponding best-fitting plane before and after trajectory enhancement. The RMSE of normal distances after trajectory enhancement shows that the point cloud attains a final overall accuracy of 5.7 cm.

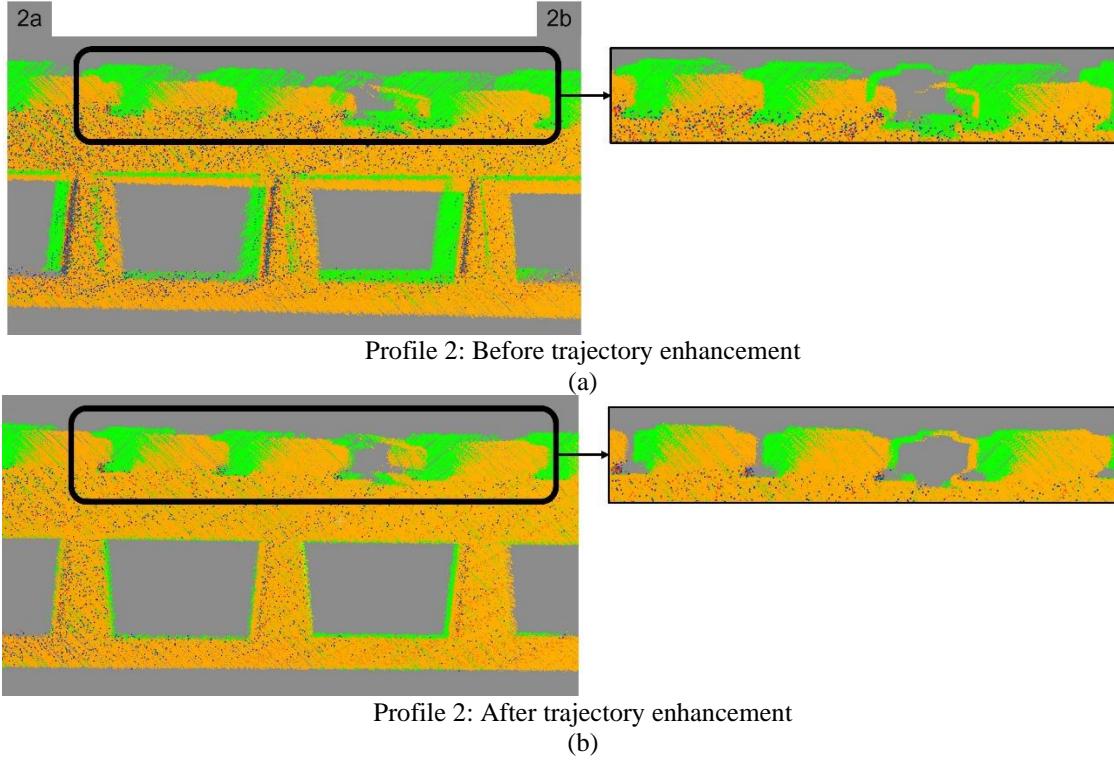


Figure 5-11 Dataset 1: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement

Table 5-2 Dataset 1: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 45,454,918 points)

	Mean (m)	STD (m)	RMSE (m)	Max (m)
Before Trajectory Enhancement	0.071	0.101	0.123	0.384
After Trajectory Enhancement	0.038	0.046	0.057	0.112

To investigate the estimated change in trajectory, Figure 5-12 shows the enhanced trajectory colored according to the magnitude of 3D corrections to positional parameters and it is overlaid on the point cloud colored by height (in shades of green). The figure indicates that the corrections are minimal (less than 5 cm) for portions of the trajectory under open sky whereas the magnitude of corrections reach the maximum value close to the middle of the bridge where there is restricted GNSS signal reception. Figure 5-13 shows the plot of the norm of post-processing standard deviations for the trajectory position parameters as reported by Inertial Explorer (blue) as well as the magnitude of corrections estimated for the same (red). The segments corresponding to the data acquisition while under the bridge are highlighted in the figure, where the post-processed trajectory accuracy is seen to deteriorate significantly. It can be seen that, as expected, the estimated corrections are more in these segments where the estimated post-processing standard deviations

were higher. The plot also depicts that the estimated corrections are generally more than the reported post-processing accuracy of position parameters, which indicates that the latter is more optimistic than the true accuracy of post-processed trajectory. In other words, there is substantial misalignment within point clouds in such areas (possibly due to accumulating trajectory drift from other areas with deteriorated trajectory accuracy) that warrants corrections to the trajectory that exceed the estimation accuracy of the parameters during post-processing. Table 5-3 reports the statistical measures for the estimated corrections to the trajectory position and attitude parameters recorded at 100 Hz frequency along with the change in velocity after enhancement. The maximum absolute correction values indicate the modification in the worst portion of the trajectory for each dataset. These measures are a direct indication of the modification to the initial trajectory to attain better point cloud alignment. The maximum positional correction to the initial trajectory is 33.4 cm (X direction) and the maximum attitude correction is 0.81° (roll angle).

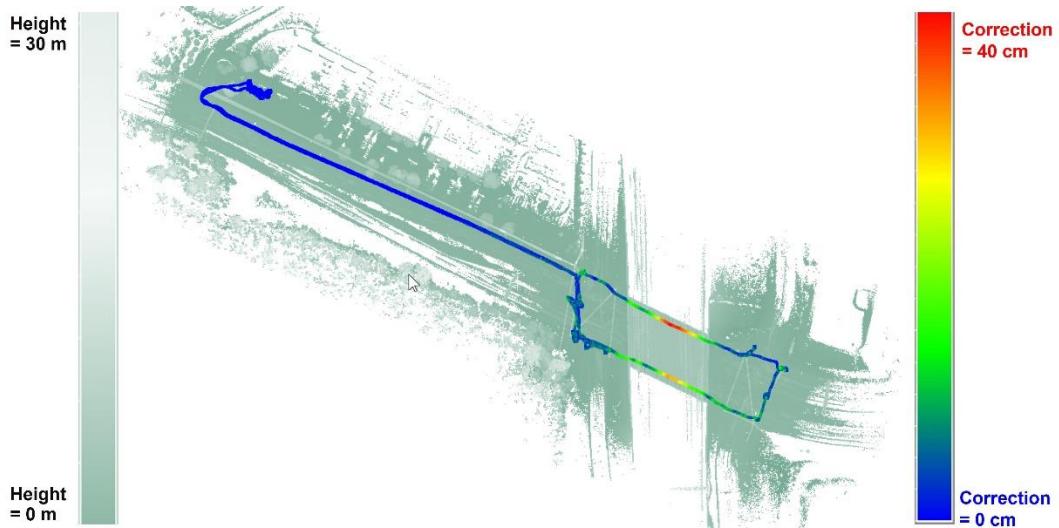


Figure 5-12 Dataset 1: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position

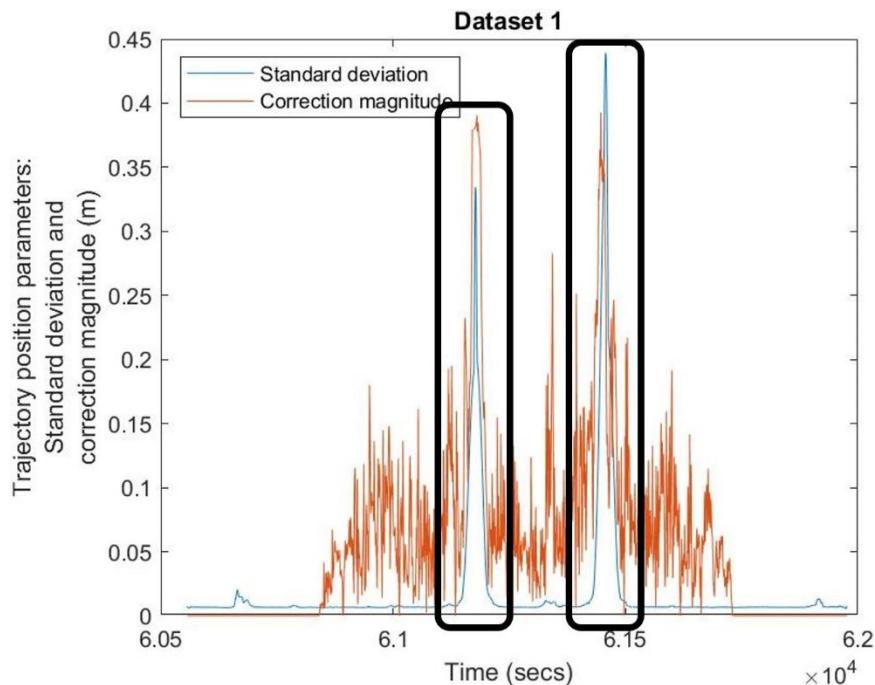


Figure 5-13 Dataset 1: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters

Table 5-3 Dataset 1: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement

# Adjusted Trajectory Points		X (m)	Y (m)	Z (m)	ω (°)	ϕ (°)	κ (°)	v (m/s)
106,158 (100 Hz; ~18 mins)	Mean	0.007	0.005	0.001	0.001	-0.009	0.170	0.021
	STD	0.089	0.056	0.037	0.076	0.117	0.182	0.035
	RMSE	0.089	0.057	0.037	0.076	0.117	0.249	0.040
	Max of Absolute Values	0.334	0.179	0.267	0.496	0.813	0.474	0.257

5.3.2 Dataset 2: Indoor Mapping

The entire point cloud colored by height is shown in Figure 5-14 along with the Backpack trajectory overlaid in pink. The dataset was acquired in an indoor facility with beginning and end of the data acquisition are under open sky whereas the remaining trajectory is indoor with no GNSS signal reception. Profile 1 in Figure 5-15 illustrates the point cloud quality before and after the two stages of trajectory enhancement. In indoor narrow corridors as the one surveyed in this dataset,

the initial misalignment within the point cloud makes them prone to wrong matches depending on the spatial proximity threshold used for feature matching. Figure 5-15 shows that trajectory enhancement without loop closure reduces the misalignment, thus allowing feature matching with loop closure. Profile 2 and Profile 3 are depicted in Figure 5-16 and Figure 5-17, respectively, where zoomed-in views of several segments clearly indicate highly accurate point cloud after trajectory enhancement. The point cloud alignment quality is quantified in Table 5-4, which indicates a final overall accuracy of 4.1 cm.

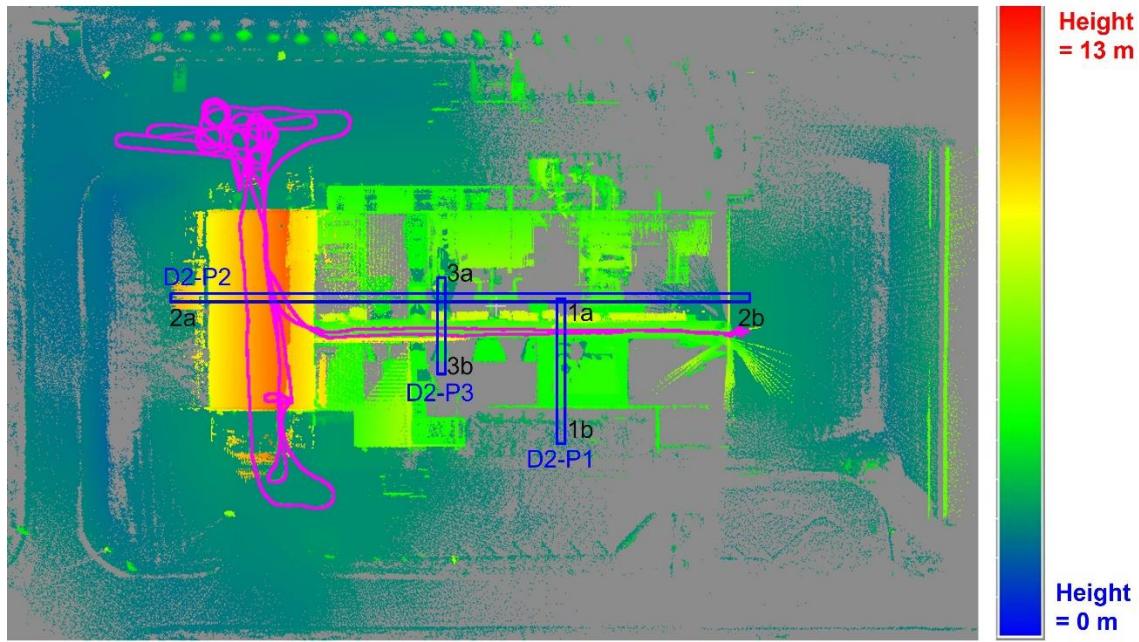
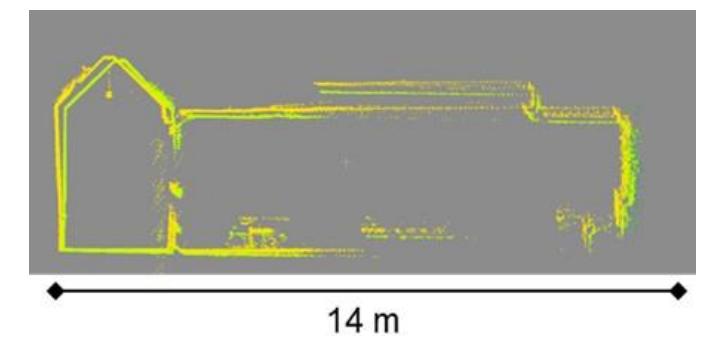
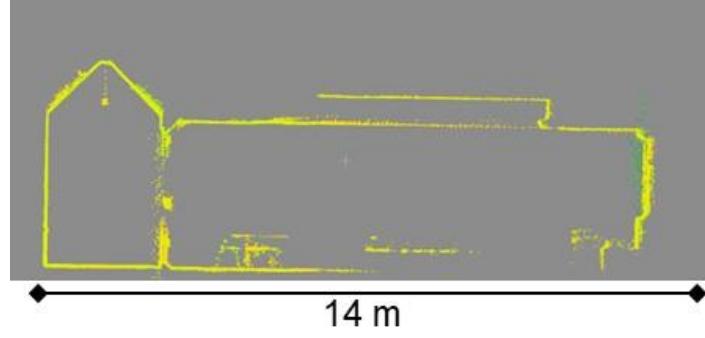


Figure 5-14 Dataset 2: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results





Profile 1: After trajectory enhancement without loop closure
(b)

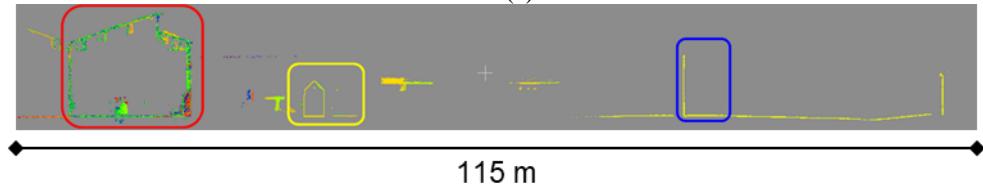


Profile 1: After trajectory enhancement with loop closure
(c)

Figure 5-15 Dataset 2: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure



Profile 2: Before trajectory enhancement
(a)



Profile 2: After trajectory enhancement
(b)

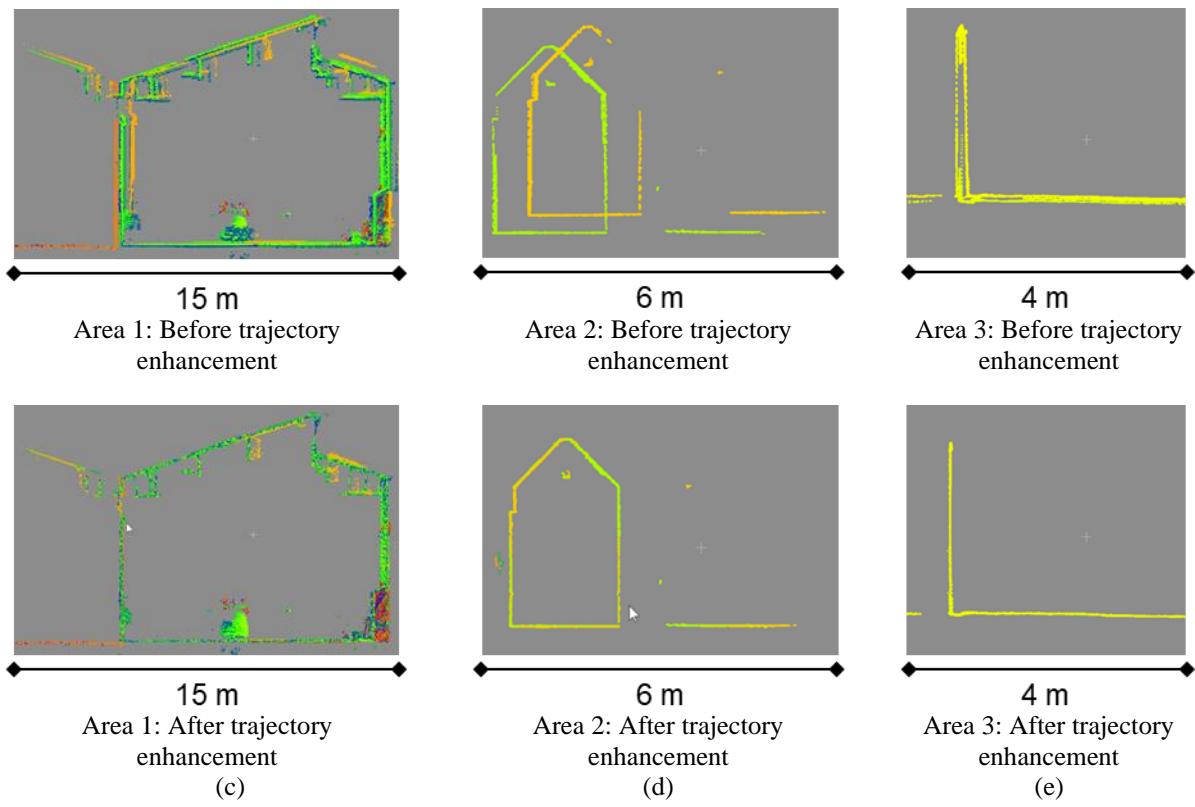
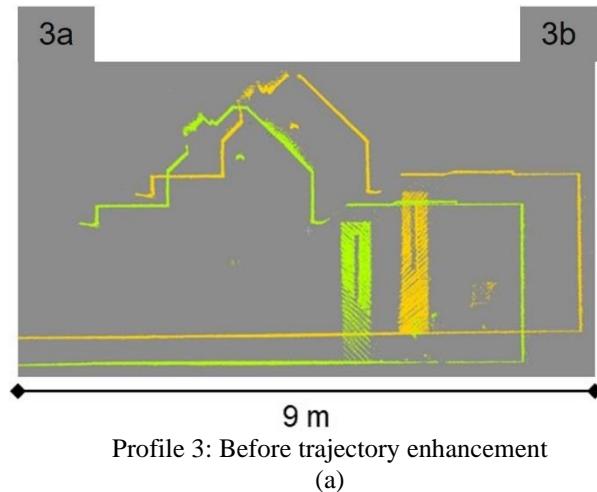


Figure 5-16 Dataset 2: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement



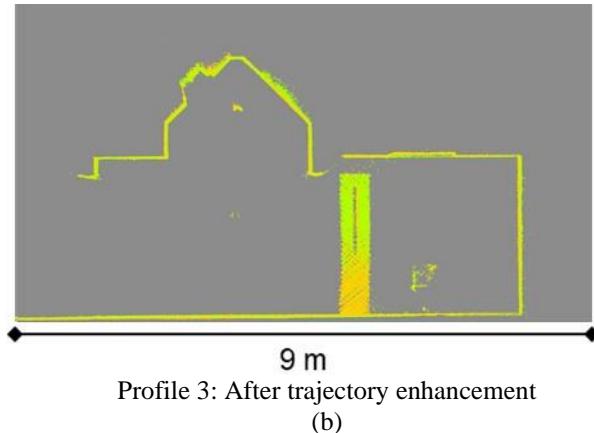


Figure 5-17 Dataset 2: Profile 3 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement

Table 5-4 Dataset 2: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 9,432,850 points)

	Mean (m)	STD (m)	RMSE (m)	Max (m)
Before Trajectory Enhancement	0.178	0.275	0.326	0.725
After Trajectory Enhancement	0.035	0.032	0.041	0.081

Figure 5-18 shows the enhanced trajectory colored according to the magnitude of 3D corrections to positional parameters, which indicates large correction magnitudes throughout the indoor trajectory with relatively higher corrections while walking through the narrow corridor in E-W direction. Figure 5-19 shows the plot of the norm of post-processing standard deviations (blue) and the magnitude of corrections estimated (red) for the trajectory position parameters, where the four shorter peaks in the post-processing standard deviation plot correspond to the indoor N-S trajectory portion and the higher peak corresponds to the long duration of indoor mapping of the E-W corridor in Figure 5-18. The plot indicates that areas with larger standard deviation have higher estimated corrections while also depicting that the post-processing accuracies are more conservative than that indicated by their impact on the point cloud accuracy. Table 5-5 reports the statistical measures for the estimated trajectory corrections with the maximum position and attitude modification of 1.284 m (X direction) and 2.32° (pitch angle), respectively.

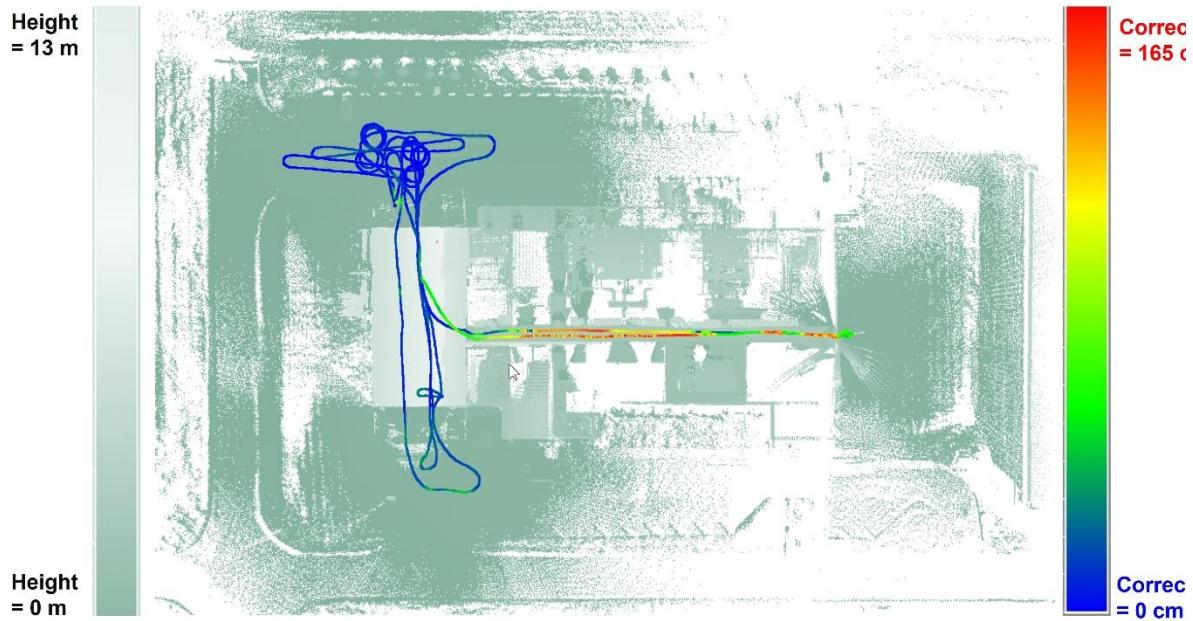


Figure 5-18 Dataset 2: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position

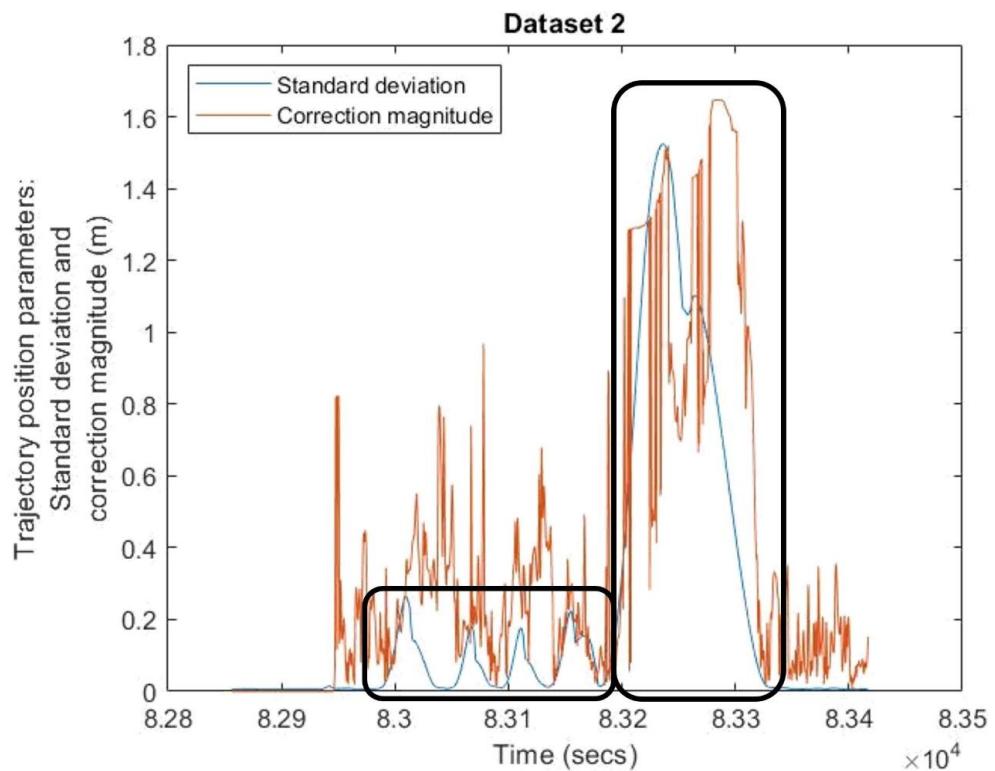


Figure 5-19 Dataset 2: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters with respect to time

Table 5-5 Dataset 2: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement

# Adjusted Trajectory Points		X (m)	Y (m)	Z (m)	ω (°)	ϕ (°)	κ (°)	v (m/s)
45,468 (~8 mins)	Mean	-0.182	0.119	-0.161	-0.032	-0.024	0.001	0.210
	STD	0.546	0.309	0.345	0.582	0.350	0.518	0.672
	RMSE	0.576	0.331	0.381	0.583	0.351	0.518	0.703
	Max of Absolute Values	1.284	0.813	0.632	2.320	2.246	2.258	1.155

5.3.3 Dataset 3: Indoor Mapping of Dome Facility using Crane Bucket

This dataset was acquired for salt dome mapping by walking outdoor around a dome followed by an indoor data acquisition with the help of a crane bucket to capture salt pile within the dome. The entire point cloud colored by height is shown in Figure 5-20 along with the Backpack trajectory overlaid in pink. The trajectory portion inside the facility is prone to complete GNSS signal obstruction. This dataset is used as an example to demonstrate the feasibility of the proposed approach to deal with non-trivial trajectory dynamics (as in the case of using crane bucket) and undulating/curved surfaces in the surroundings. Figure 5-21 shows Profile 1 before and after trajectory enhancement without/with loop closure. The dome and salt pile surface indicate improvement in alignment after the two stages of trajectory enhancements. Figure 5-22 and Figure 5-23 show zoomed-in views of different areas within Profile 1 and Profile 2, respectively, before and after trajectory enhancement. The figures clearly demonstrate that the proposed trajectory enhancement strategy can produce well-aligned point clouds for datasets comprised of areas with curved and undulating surfaces as encountered in this dataset. Table 5-6 reports the quantitative measures of LiDAR point cloud alignment. The RMSE after trajectory enhancement indicates a final overall accuracy of 8.4 cm. The relatively higher RMS of normal distance for this dataset compared to the others can be attributed to the curved and undulating surfaces in this dataset.

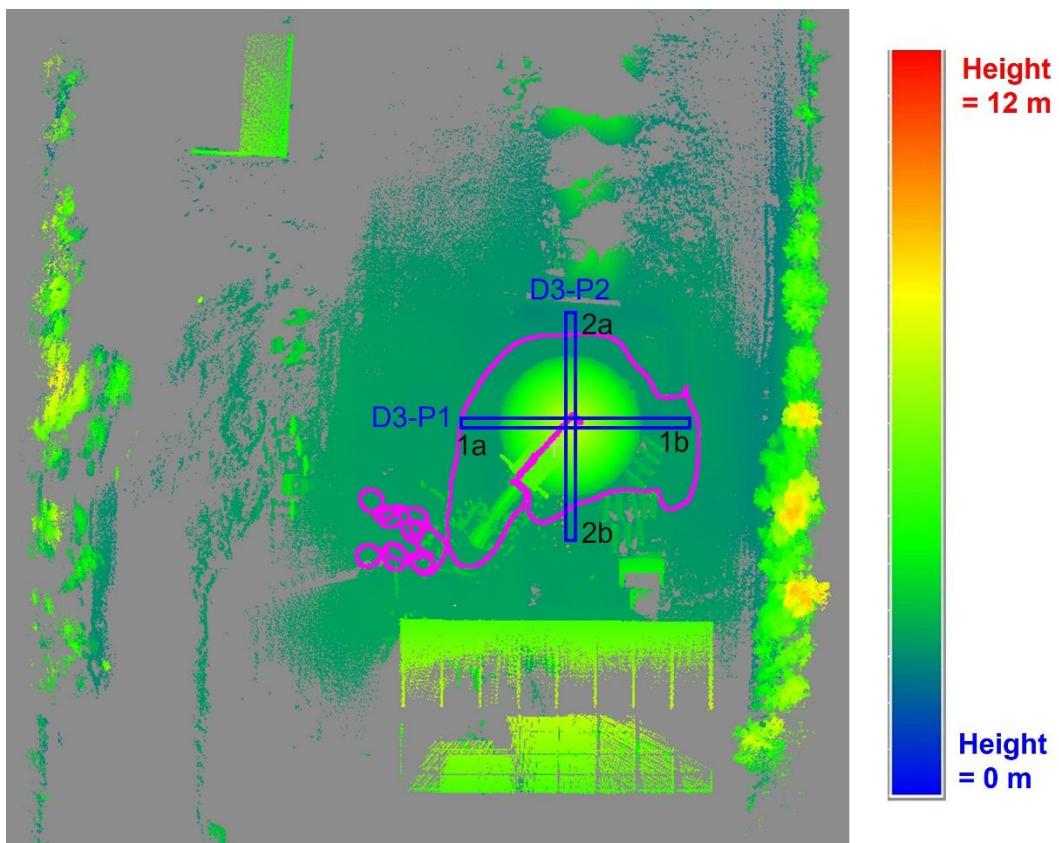
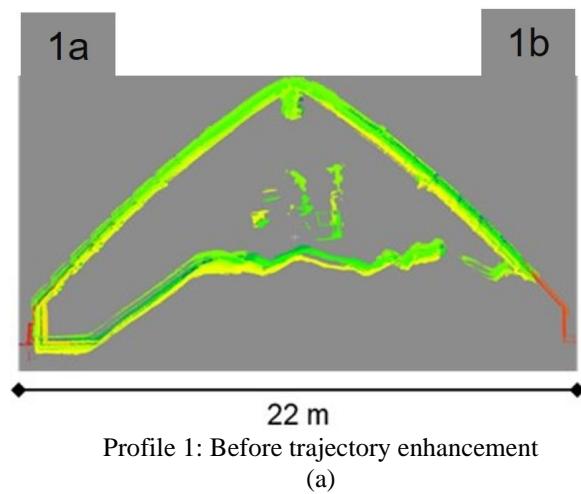
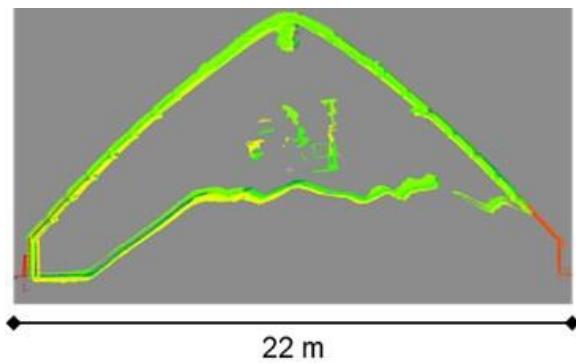
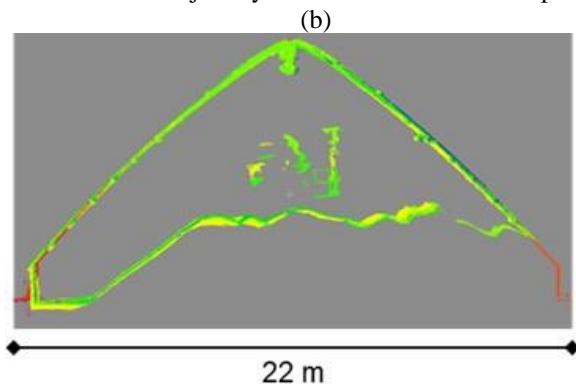


Figure 5-20 Dataset 3: Entire point cloud colored by height, overlaid Backpack MLMS trajectory (pink), and profiles (blue boxes) used for qualitative assessment of trajectory enhancement results





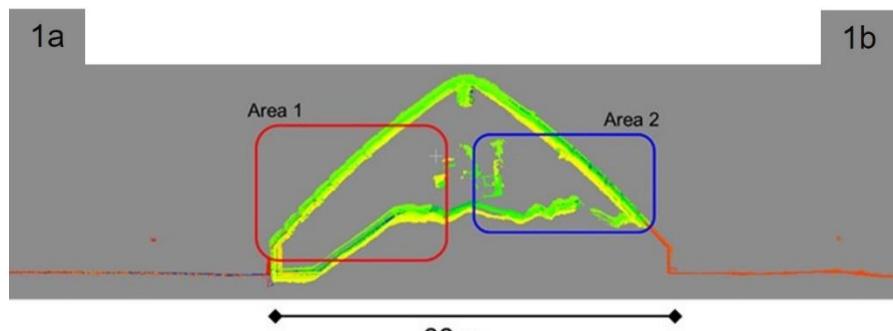
Profile 1: After trajectory enhancement without loop closure



Profile 1: After trajectory enhancement with loop closure

(c)

Figure 5-21 Dataset 3: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement without/with loop closure



Profile 1: Before trajectory enhancement

(a)

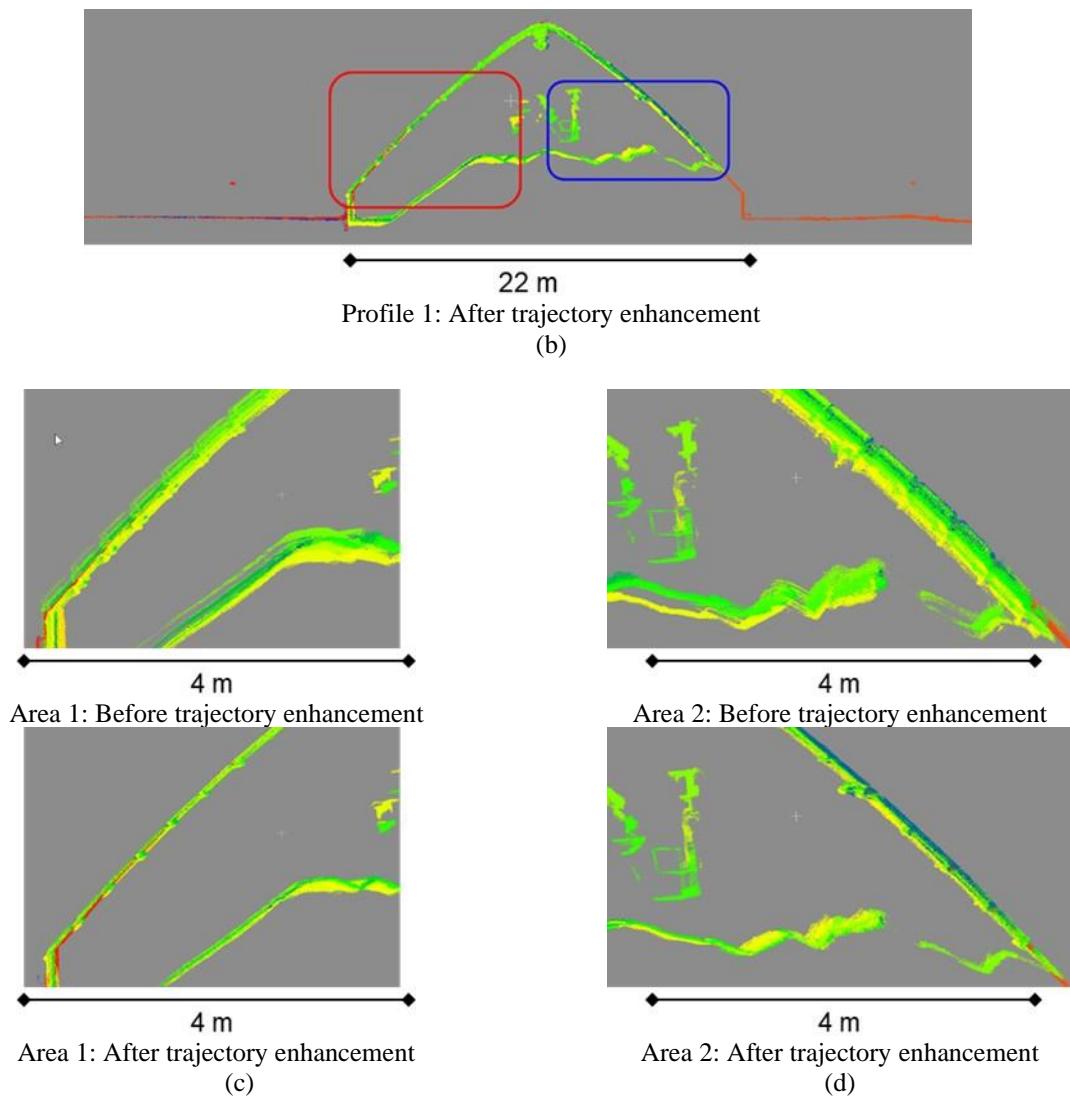
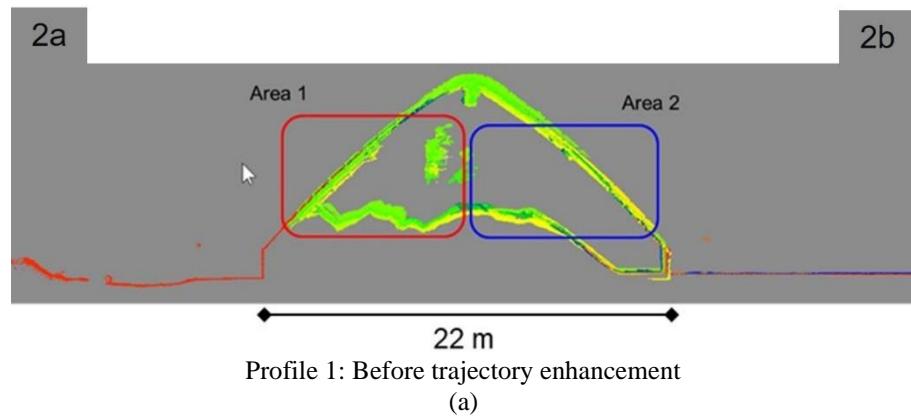


Figure 5-22 Dataset 3: Profile 1 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement



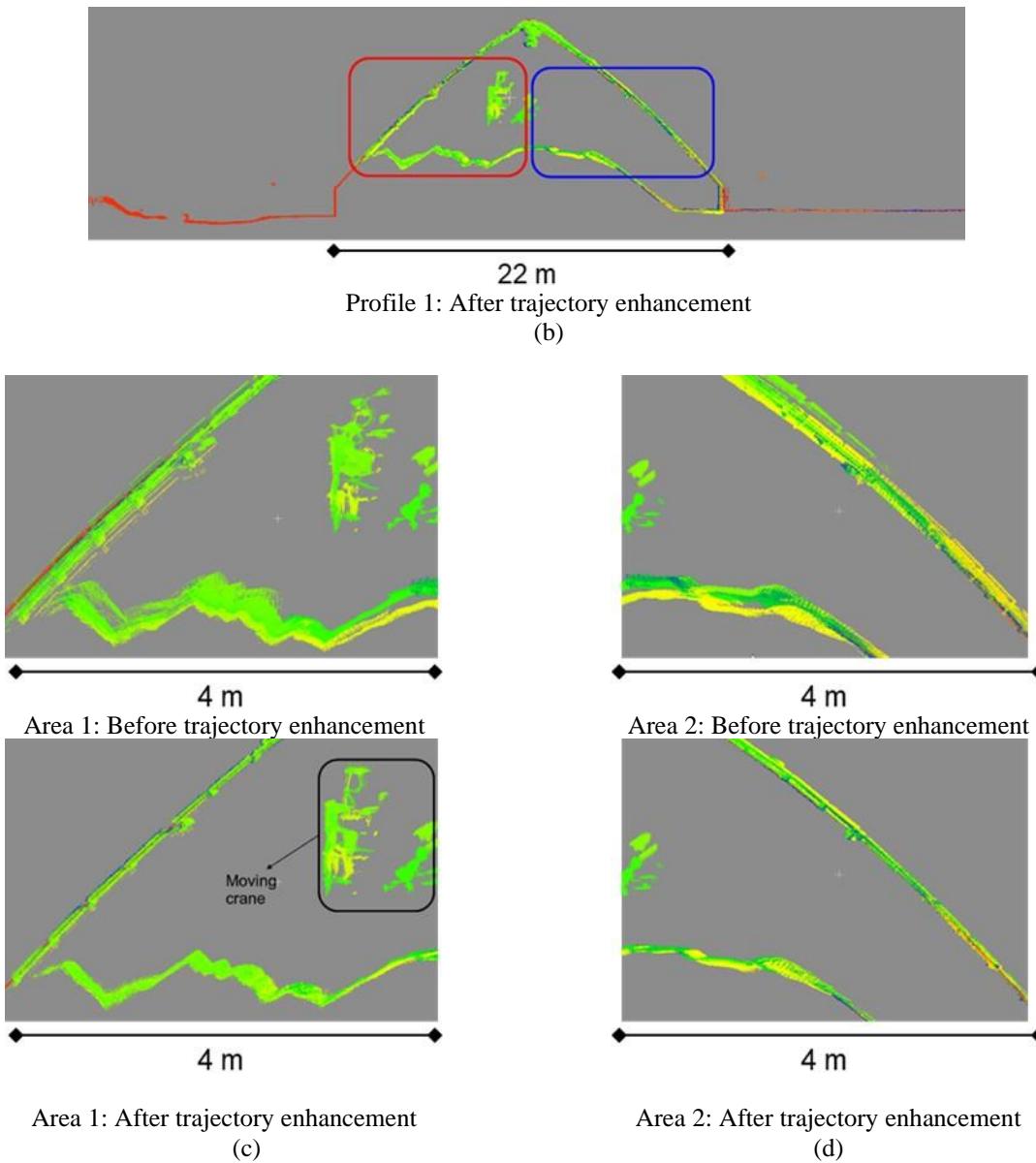


Figure 5-23 Dataset 3: Profile 2 (colored by time) illustrating point cloud alignment quality before and after trajectory enhancement

Table 5-6 Dataset 3: Quantitative evaluation of point cloud alignment quality before and after trajectory enhancement (based on a total of 51,420,371 points)

	Mean (m)	STD (m)	RMSE (m)	Max (m)
Before Trajectory Enhancement	0.116	0.120	0.165	0.423
After Trajectory Enhancement	0.055	0.073	0.084	0.251

The change in trajectory is qualitatively depicted in Figure 5-24, which depicts substantial modification for the outdoor as well as indoor portions. This can be attributed to the following characteristics of the dataset: (a) curved dome surfaces being treated as piecewise planar surfaces and (b) inability to separate outer and inner surfaces of the dome during feature matching. Figure 5-25 shows a zoomed-in view of dome surface before and after trajectory enhancement while depicting the versions captured by the outdoor and indoor trajectory portions in blue and red, respectively. It can be seen that the side wall is captured only by the outdoor portion since the indoor portion captures the salt pile surface that obstructs the side wall. The figure clearly demonstrates that the outdoor and indoor dome surface are aligned after trajectory enhancement due to the inability to delineate outer (blue) and inner (red) surfaces, thus supporting the above claim. Figure 5-26 shows the plot of the norm of post-processing standard deviations and the magnitude of corrections estimated (red) for the trajectory position parameters with the indoor trajectory portion highlighted in black. The higher corrections in areas with low standard deviations (outdoor) is caused by the same reasons stated above. It is worth noting that similar behavior was observed for attitude parameters. Table 5-7 reports the statistical measures for the estimated corrections to the high-frequency trajectory position and attitude parameters along with the velocity corrections. The maximum position and attitude modification to the initial trajectory is 51.4 cm (Y direction) and 2.04° (heading angle).

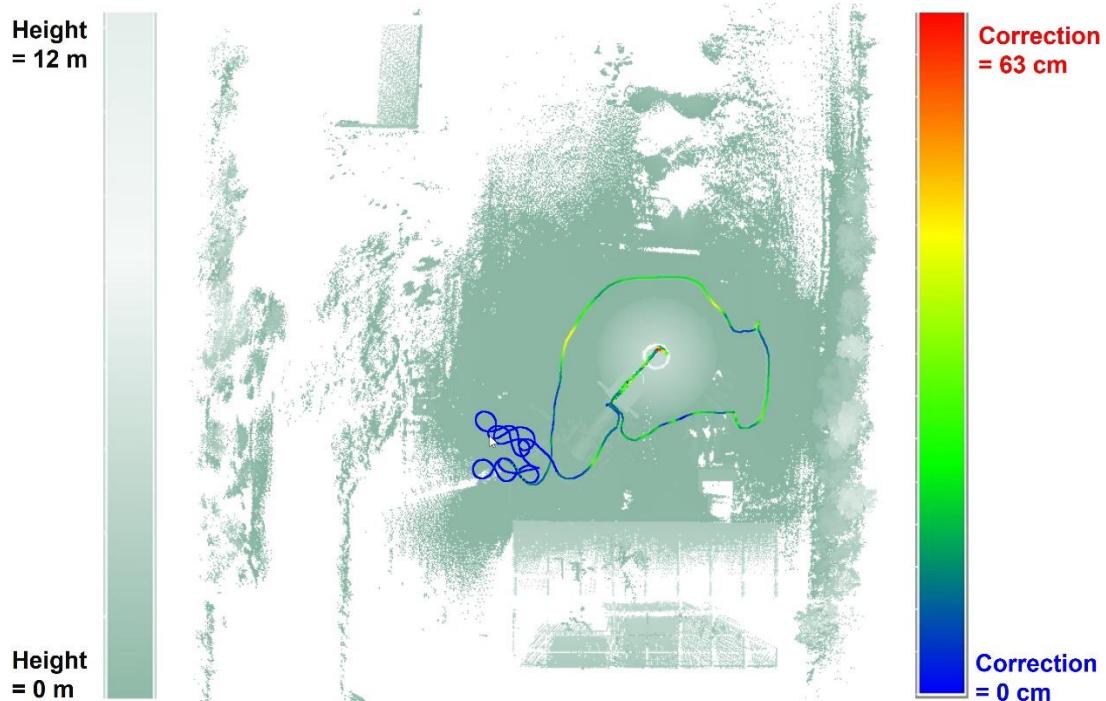


Figure 5-24 Dataset 3: Quantitative assessment of estimated position corrections to the trajectory: enhanced trajectory colored by magnitude of estimated corrections to 3D position

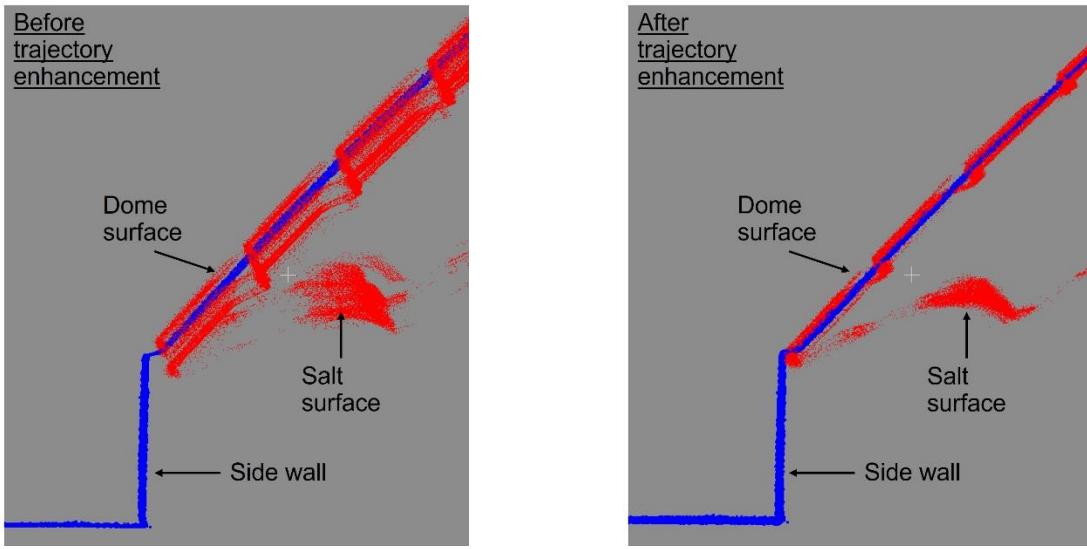


Figure 5-25 Dataset 3: Dome surface, side wall, and salt surface before and after trajectory enhancement colored by outdoor (blue) and indoor (red) trajectory portions

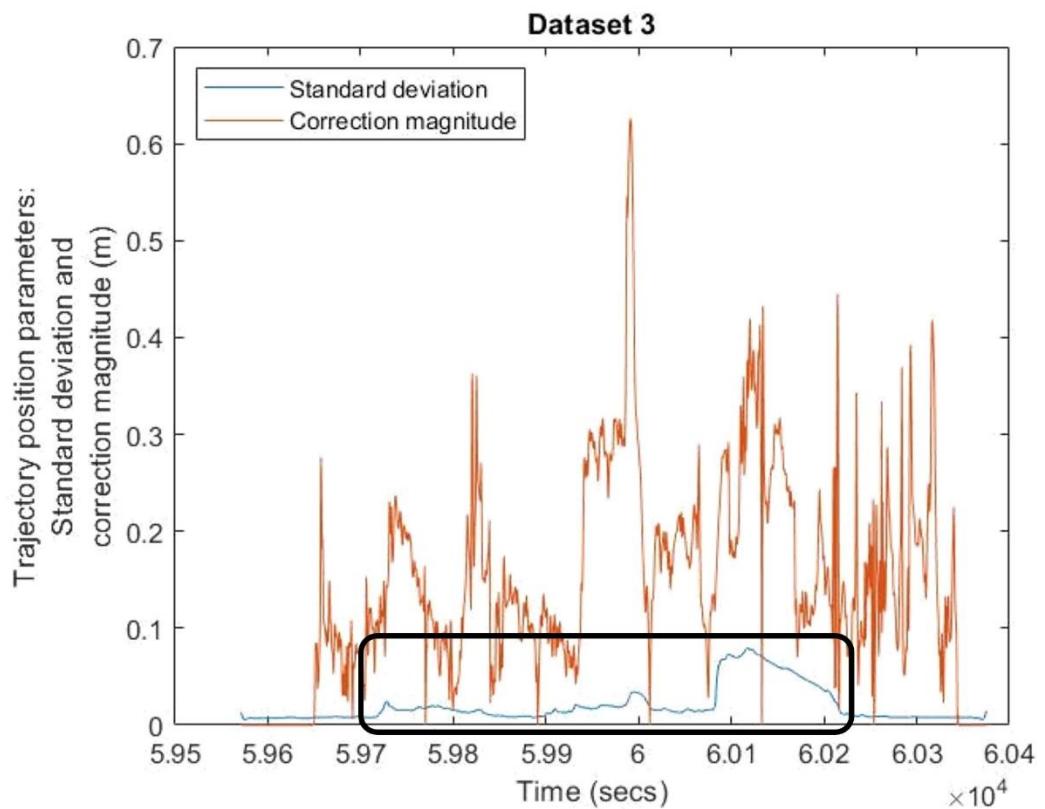


Figure 5-26 Dataset 3: Post-processing standard deviation and magnitude of estimated corrections for trajectory position parameters

Table 5-7 Dataset 3: Quantitative evaluation of the change in trajectory position (X, Y, Z), attitude (ω, ϕ, κ), and velocity (v) parameters as a result of trajectory enhancement

# Adjusted Trajectory Points		X (m)	Y (m)	Z (m)	ω (°)	ϕ (°)	κ (°)	v (m/s)
69,104 (100 Hz; ~12 mins)	Mean	0.007	0.047	-0.014	0.002	0.007	0.228	0.030
	STD	0.094	0.124	0.102	0.257	0.224	0.549	0.053
	RMSE	0.094	0.133	0.103	0.257	0.224	0.595	0.061
	Max of Absolute Values	0.370	0.514	0.352	1.681	1.108	2.041	0.582

5.4 Conclusions and Recommendations for Future Work

This paper presented a two-stage framework for GNSS/INS trajectory enhancement with the help of planar features extracted from point clouds captured by LiDAR sensors onboard a Backpack MLMS. The approach was designed to successively eliminate short-term and long-term trajectory drifts. Planar features were extracted and matched without loop closure in the original point cloud to be used for the first stage of trajectory enhancement to mitigate short-term drift. The resultant intermediate enhanced point cloud was used for feature extraction and matching with loop closure, which then assisted in conducting the final stage of trajectory enhancement. The approach was tested on three datasets with varying duration of GNSS signal outages. The results demonstrated the feasibility of the proposed approach to produce an accurate trajectory and well-aligned point clouds. For the different datasets, the proposed approach is seen to produce point clouds with an overall alignment precision of 4–8 cm irrespective of the nature of the surroundings being surveyed and initial GNSS/INS trajectory quality behavior (segments of ~40–150 secs of GNSS signal outages). The approach eliminates initial inaccuracies within the trajectory position and orientation parameters ranging up to about 1.30 m and 2.50°, respectively.

Future work in this domain will focus on devising an approach to automatically extract and match linear and/or cylindrical primitives in the surroundings to allow trajectory enhancement for complex environments, such as forests and ravines, that are not conducive to the use of solely planar features and would require the need to leverage information from surrounding trees/vegetation, power cables, and so on. The proposed approach will also be tested for longer duration of GNSS signal outages. Furthermore, a tightly coupled integration of LiDAR with GNSS and INS information will be explored by trying to utilize the enhanced trajectory parameters estimated using the proposed approach during GNSS/INS integration as prior information for producing accurate trajectory in GNSS-challenging portions of data acquisition. Finally, while the current implementation is executed on a CPU with 64 GB RAM, future work will explore the ability to utilize GPU to speed up the processing in order to facilitate trajectory enhancement for data acquisitions resulting in significantly larger point clouds.

6. SEMANTIC SEGMENTATION OF BRIDGE COMPONENTS AND ROAD ELEMENTS USING MOBILE LIDAR DATA

Abstract

Emerging mobile LiDAR mapping systems exhibit great potential as an alternative for mapping road environments. Such systems can acquire high-quality, dense point clouds that capture detailed information over an area of interest through efficient field surveys. However, recognizing and semantically segmenting different components from the point clouds automatically with efficiency and high accuracy remains a challenge. Towards this end, this study proposes a deep learning-based semantic segmentation framework to simultaneously classify bridge components and road elements using mobile LiDAR point clouds. A cross-labeling technique leveraging the good geolocation quality of mobile LiDAR data is developed to reduce the need of manual annotation. The transferability of the deep learning-based approach across data acquired from different systems or capturing varying scenes is explored. The proposed framework is evaluated using data from surveying-grade and mapping-grade systems along an interstate highway with 27 highway bridges. The results show that the proposed cross-labeling approach can effectively transfer labels from one mobile mapping system to another for all classes except scanning artifacts. The deep learning-based semantic segmentation framework can handle mobile LiDAR data efficiently and achieve an overall accuracy of 84%. Bridge decks and piers can be classified with high accuracy while abutments are difficult to predict. The proposed transfer leaning technique can boost the deep learning model performance when dealing with data from different systems and scenes.

6.1 Introduction

Precise, periodic, real-world measurements that capture the as-is conditions of structures lay the foundation for the maintenance and management of transportation infrastructure. The ability to efficiently map miles of highway and road networks is the key for applications like transportation asset management, high-definition map generation, and autonomous driving. Bridges, one of the essential elements in transportation infrastructure, require frequent and accurate inspections to ensure structural soundness and prioritize maintenance [150]. Current highway asset management practices rely primarily on visual inspection and manual measurements conducted by trained inspectors, which is time-consuming and expensive. Remote sensing technologies have stimulated various non-contact approaches for monitoring existing infrastructure. Unmanned aerial vehicles (UAV) imagery has been utilized to detect structural defects and reconstruct 3D information for bridges [151–155]. Terrestrial laser scanners (TLS) that provide high-resolution point clouds with millimeter-level precision have been deployed to track construction progress [156–158] and monitor structural health post-construction [159–161]. Despite being effective, UAV and TLS surveys are limited to a local region, thus not scalable for mapping miles of highway and road networks. Ground mobile LiDAR mapping systems (MLMS) can capture large areas with efficient field surveys and therefore are widely adopted for mapping urban environments [162–164]. Such systems typically utilize direct georeferencing, i.e., trajectory information provided by the onboard global navigation satellite system/inertial navigation system (GNSS/INS) unit, to reconstruct LiDAR data in a common mapping frame. Centimeter-level positional accuracy can be expected if the quality of trajectory from the GNSS/INS unit and system calibration parameters are

guaranteed. While MLMS exhibits great potential for mapping transportation corridors, bottlenecks in automated extraction of geometric and semantic information from mobile LiDAR data limit our ability to leverage the detailed information captured by such systems. As pointed out by prior research, semantic segmentation of mobile LiDAR data is extremely challenging due to occlusions, irregular distribution and high volume of data, and varying point density [165,166].

At an early stage, point cloud segmentation aims at clustering points into subgroups that share similar characteristics. Such segmentation approaches typically utilize the spatial proximity and similarity of local features derived from a neighborhood. Properties like gradient and surface normal were derived based on raster digital surface model (DSM) or triangulated irregular network (TIN) and used for clustering [167,168]. For irregular point clouds, dimensionality features (e.g., planar, linear, and rough) based on eigenvalue analysis in a local neighborhood proposed by Demantke et al. [96] are commonly used. Points sharing similar characteristics are grouped together using different algorithms [95,169,170]. A major drawback of these approaches is that they do not link the segments to any contextual information.

Having the ability to cluster points sharing similar characteristics into segments, researchers then focus on the extraction of semantic segmentation, i.e., assigning a semantic class label to each point or cluster in the point cloud. One trend of this research is to use supervised machine learning algorithms. For example, support vector machine (SVM), random forests, Bayesian classifier, and AdaBoost are frequently used [171–175]. Despite the good efficiency, these algorithms work on individual points or clusters and do not consider neighborhood relationships. To model the interactions between neighboring points or clusters, graphical models including Markov networks and conditional random fields (CRF) are introduced [176–180]. However, using graphical models to represent the relationship of 3D points is limited by the low-dimensional feature space. Recently, researchers have explored deep learning on point clouds, following the successful convolutional neural network architectures used for image segmentation [181]. The irregular structure and huge volume of data are the main challenges for applying neural networks on point clouds. Some studies proposed imagery and point cloud data fusion to take advantage of existing network architectures and benchmark datasets for 2D images [182,183]. Other attempts like SEGCloud [184] and VoxelNet [185] convert the point cloud to regular voxels. However, these approaches suffer from information loss in the projection or voxelization process, and thus cannot retain detailed information captured by the original point clouds. PointNet [186] is a well-known deep learning architecture that directly takes irregular 3D point clouds. PointNet++ [187] exploited local geometric structures by hierarchically applying PointNet in small neighborhoods. PointCNN [188] used an X-transformation to learn spatially local correlation between points. To model the relationship between neighboring points, Wang et al. [189] proposed EdgeConv that acts on graphs. In EdgeConv, each point in the point cloud is represented as a node and it is connected to its neighboring points by a directed graph. Although the above-mentioned work achieved good segmentation results, they can only handle a small number of points and are incapable of considering long-range contextual information (i.e., contextual information for distant objects). Landrieu et al. [190] proposed point cloud semantic segmentation with superpoint graph (SPG). Their approach partitions point clouds into geometrically homogeneous clusters (hereafter, denoted as superpoint), constructs superpoint graphs to encode the neighborhood relationship, and uses graph convolution networks for contextual segmentation. The strength of the SPG framework

lies in its ability to handle a large volume of data and model long-range contextual information while considering local geometric attributes. This approach is tested using point cloud segmentation benchmarks over large outdoor scenes, Semantic3D [191] and Stanford Large-Scale 3D Indoor Spaces (S3DIS) [192], and demonstrates superior performance.

While extensive effort has been devoted to point cloud semantic segmentation in general outdoor scenes and road environments, studies that focus on bridge components are rare in the literature. Existing work for semantic segmentation of bridge components can be classified into geometric-based strategies [193–196] and learning-based approaches [197–199]. Geometric-based strategies require manual background removal and domain knowledge about the geometric and topological constraints among the bridge components. Such approaches are sensitive to noise and occlusions in the point cloud, and thus not practical for complex real-world scenes. Learning-based approaches do not require manual background removal. Instead, they have several classes for bridge components (e.g., bridge deck, girder, pier, and abutment) and one class for background. Kim and Kim [198] compared PointNet, PointCNN, and EdgeConv for their ability to classify bridge components and concluded EdgeConv outperformed the other two architectures. Xia et al. [199] proposed a multi-scale local descriptor based on geometric features of bridges and developed a machine learning pipeline for classification. Although learning-based approaches demonstrated great potential towards versatile solutions, lack of training data is a major concern since current publicly available benchmarks do not provide label information regarding bridge components.

Insufficient training data is a universal obstacle that hinders the application of deep learning [200]. Deep learning works under the assumption that the training and testing data have the same feature space and distribution, and it relies strongly on massive training data [201,202]. When the distribution changes, the models need to be trained from scratch using new training data to ensure the best performance. Collecting and constantly updating a large amount of annotated data, however, is extremely expensive and thus impractical for many real-world applications. For instance, data acquired from different systems may present varying characteristics owing to the different noise level and scanning mechanisms of the LiDAR units begin used. Complexity of scanned scenes, e.g., different building heights and diverse shapes or sizes of road-side objects, is another factor that contributes to the varying characteristics. Consequently, new strategies that reduce the need and effort for collecting annotated data would greatly facilitate the application of deep learning to real-world problems.

Transfer learning aims to leverage knowledge from a source domain to improve the learning process in a target domain. It relaxes the assumption that the training data has the same feature space and distribution as the testing data [203]. By applying transfer learning techniques, a model in the target domain does not need to be built from scratch, thus significantly reducing the need for training data in the target domain. A good summary of past works dealing with transfer learning can be found in Pan and Yang [204] and Weiss et al. [205], with a focus on machine learning methods. Recent surveys that discuss knowledge transfer in deep neural networks can be found in Tan et al. [203] and Zhuang et al. [206]. However, existing studies on transfer leaning mainly focuses on 2D imagery data. Transfer learning in 3D, especially on point clouds, is less explored and more challenging. Most of the existing research tried to take advantage of traditional convolution neural networks and benchmark imagery datasets [207–210]. In Imad et al. [207], the

model was pre-trained on 2D object detection image datasets and fine-tuned with birds-eye-view images created from the point clouds. Zhao et al. [208] extracted several features from point clouds, including normalized height, intensity, surface curvature, and organized them into images. Balado et al. [209] and Piewak ei al. [210] projected 3D point clouds to 2D images. The latter also developed a cross-labeling technique that transfers labels in the image space to the object space. He et al. [166] proposed an extended multiclass TrAdaBoost algorithm that can handle training data from complementary sources, i.e., images and point clouds. They adopted VoxelNet in their framework to handle 3D point clouds. Zhao et al. [211] proposed a domain adaption framework to bridge the gap between synthetic and real LiDAR point clouds. To the best of the authors' knowledge, no current study has explored the transferability of deep neural network that take irregular point sets as input.

This study aims to investigate: 1) the performance of using mobile LiDAR data and a deep learning-based approach to classify bridge components as well as road elements and 2) the transferability of the deep learning-based approach across data acquired from different systems or capturing varying scenes. The SPG framework [190] is adopted for point cloud semantic segmentation because it: 1) considers the spatial context at an object level instead of points, 2) has the ability to learn rich contextual information owing to the less constrained feature space, and 3) handles large-scale point clouds without the need for down-sampling while maintaining acceptable efficiency. Two strategies that help the deep learning-based approach adapting to the varying characteristics of the acquired data are developed: cross-labeling and transfer learning. The former transfers the labels from one mobile mapping system to another; the latter fine-tunes a model trained on the source domain using a minimal amount of data from the target domain. The proposed framework is evaluated using data acquired by surveying-grade and mapping-grade MLMS units along an interstate highway.

6.2 Data Acquisition System and Dataset Description

6.2.1 Mobile LiDAR mapping systems

The data used in this study were acquired by two wheel-based mobile mapping systems developed by the research team: Purdue Wheel-based Mobile Mapping System-High Accuracy (PWMMS-HA) and Purdue Wheel-based Mobile Mapping System-Ultra High Accuracy (PWMMS-UHA), as shown in Figure 6-1. The PWMMS-HA (Figure 6-1a) is outfitted with four Velodyne LiDAR units (three HDL-32E and one VLP-16 High Resolution), three cameras, and a GNSS/INS unit for direct georeferencing. The PWMMS-UHA (Figure 6-1b) is equipped with two 2D profiler LiDAR units (a Riegl VUX-1HA and a Z+F Profiler 9012), two cameras, and a georeferencing unit. Table 6-1 reports the specifications of the LiDAR [212–215] and georeferencing [8,216] units onboard the two systems. System calibration was conducted for both vehicles using an in-situ calibration procedure [107]. The post-calibration accuracy of the point cloud from each MLMS at a specific sensor-to-object distance was estimated using a LiDAR Error Propagation calculator [217]. The results suggest an expected accuracy of $\pm 2\text{--}3$ cm and $\pm 1\text{--}2$ cm at a sensor-to-object distance of 30 meters for the PWMMS-HA and PWMMS-UHA, respectively.

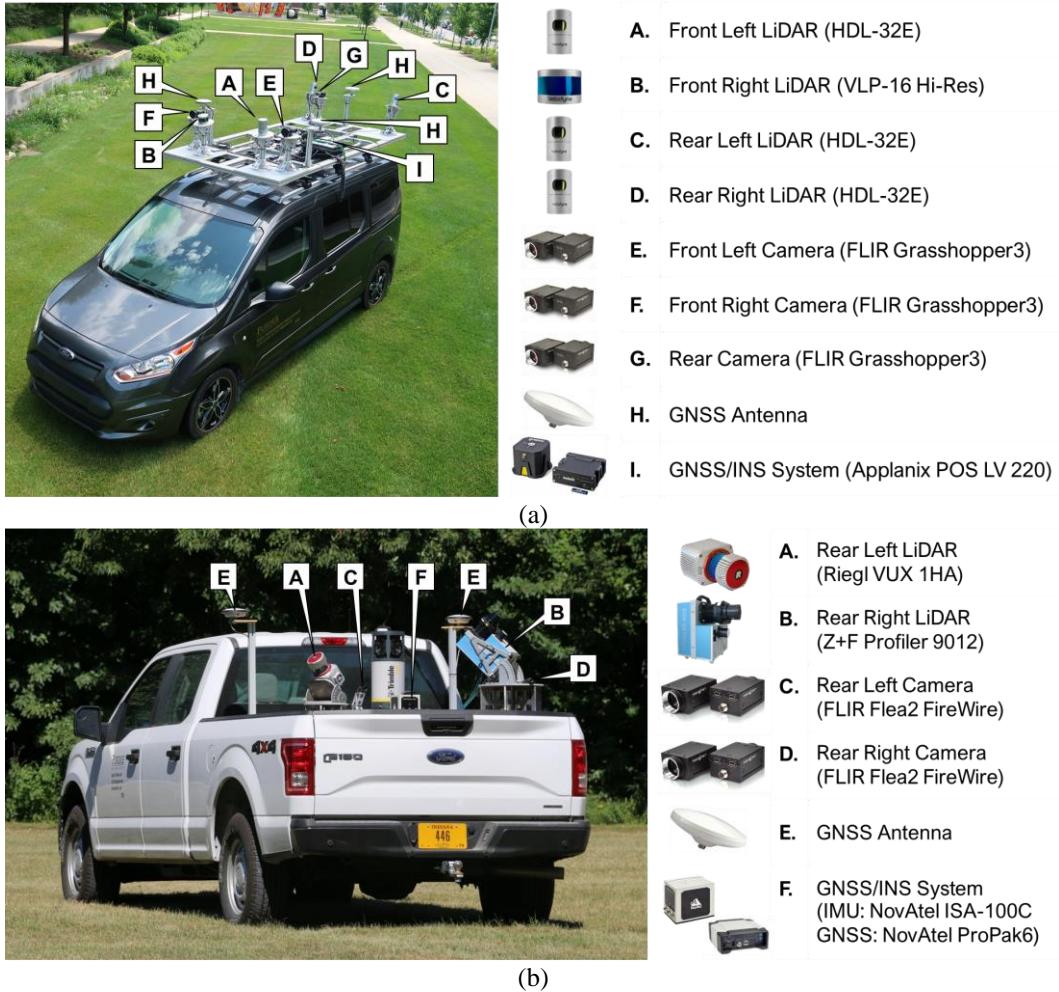


Figure 6-1 The wheel-based mobile mapping systems and onboard sensors: (a) PWMMS-HA and (b) PWMMS-UHA.

Table 6-1 Specifications of the georeferencing and LiDAR sensors and the expected accuracy of the point cloud for the mobile LiDAR mapping systems.

	PWMMS-HA		PWMMS-UHA	
GNSS/INS sensors	Applanix POS LV 220		NovAtel ProPak6; IMU-ISA-100C	
Sensor weight	$2.40 + 2.50$ kg		$1.79 + 5.00$ kg	
Positional accuracy	$\pm 2\text{--}5$ cm		$\pm 1\text{--}2$ cm	
Attitude accuracy (roll/pitch)	0.015°		0.003°	
Attitude accuracy (heading)	0.025°		0.004°	
LiDAR sensors	Velodyne VLP-16 High-Res	Velodyne HDL-32E	Riegl VUX 1HA	Z+F Profiler 9012
Sensor weight	0.830 kg	1.0 kg	3.5 kg	13.5 kg
No. of channels	16	32	1	1
Pulse repetition rate	$\sim 300,000$ point/s (single return)	$\sim 695,000$ point/s (single return)	Up to 1,000,000 point/s	Up to 1,000,000 point/s
Maximum range	100 m	100 m	135 m	119 m
Range accuracy	± 3 cm	± 2 cm	± 5 mm	± 2 mm

6.2.2 Dataset description

The field survey was conducted along an interstate highway, I65, from West Lafayette, Indiana to Whitestown, Indiana (shown in Figure 6-2). Both MLMS vehicles drove southbound and then northbound with an average speed of 50 mph. LiDAR and imagery data were acquired simultaneously and directly georeferenced by the onboard GNSS/INS unit. A total of four datasets were created: the PWMMS-UHA bridge, PWMMS-HA bridge, PWMMS-UHA highway, and PWMMS-HA highway datasets. For the PWMMS-UHA and PWMMS-HA bridge datasets, 27 overpassing bridges (Bridges 1 to 27 in Figure 6-2) along the highway were identified. The bridges together with their surrounding areas were manually extracted from the point clouds. The length of the region of interest for a bridge (dimension along the driving direction) is in the range of 100 meters to 150 meters. Figure 6-3 shows a sample point cloud and images from the MLMS capturing Bridge 2 (with column piers). For the PWMMS-UHA and PWMMS-HA highway datasets, the area of interest (I65 from West Lafayette to Whitestown) covering approximately 42 miles was tiled at a regular interval of 120 meters along the driving direction, resulting in a total of 554 tiles. The length of the tiles was set to 120 meters so that the tile size for the highway and bridge datasets would be compatible.

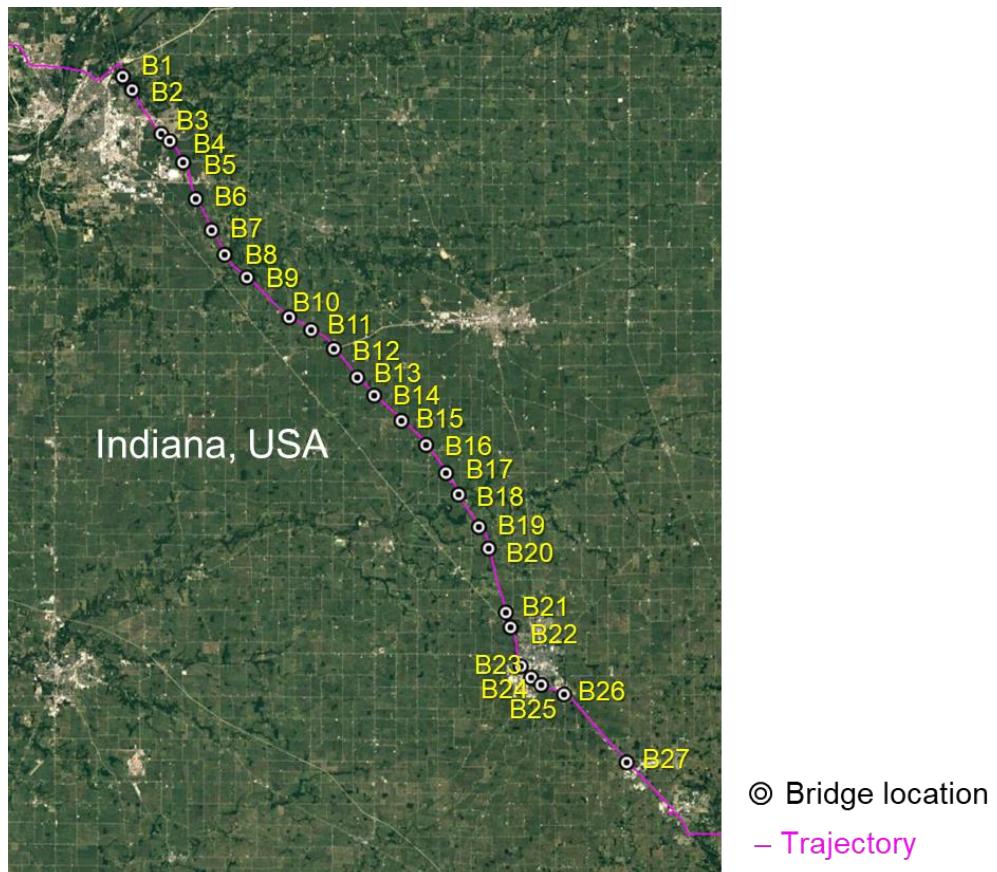


Figure 6-2 Study site showing data collection route and bridge locations (aerial photo adopted from Google Earth Image).

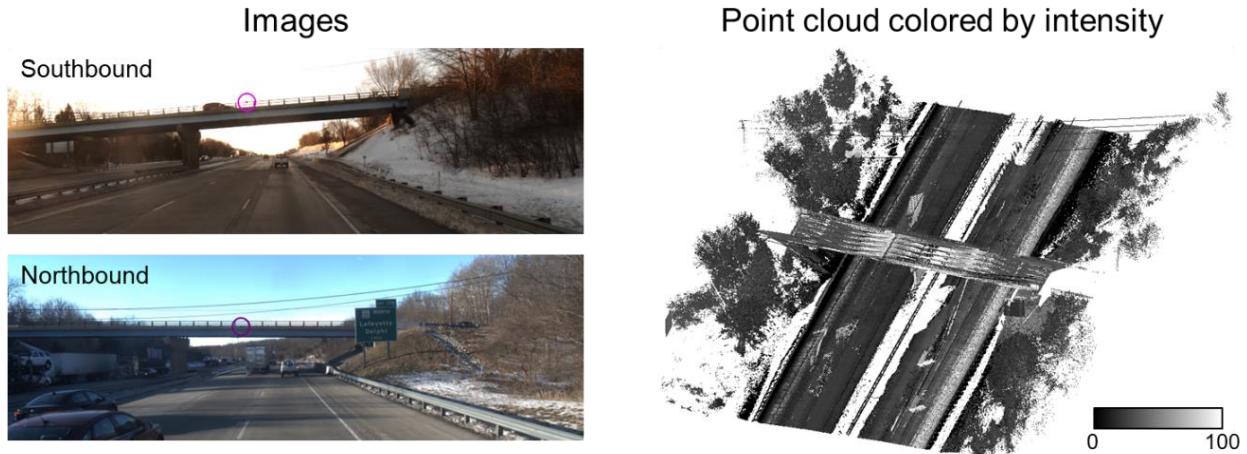


Figure 6-3 Sample data from the mobile mapping system showing the images (from southbound and northbound) and point cloud (colored by intensity) corresponding to Bridge 2.

6.3 Methodology

This section describes the main steps of the proposed framework of using mobile LiDAR data and deep learning to classify bridge components as well as road elements. The main steps include point cloud preprocessing, manual annotation, cross-labeling, deep learning-based point cloud semantic segmentation, and transfer learning, as outlined in Figure 6-4.

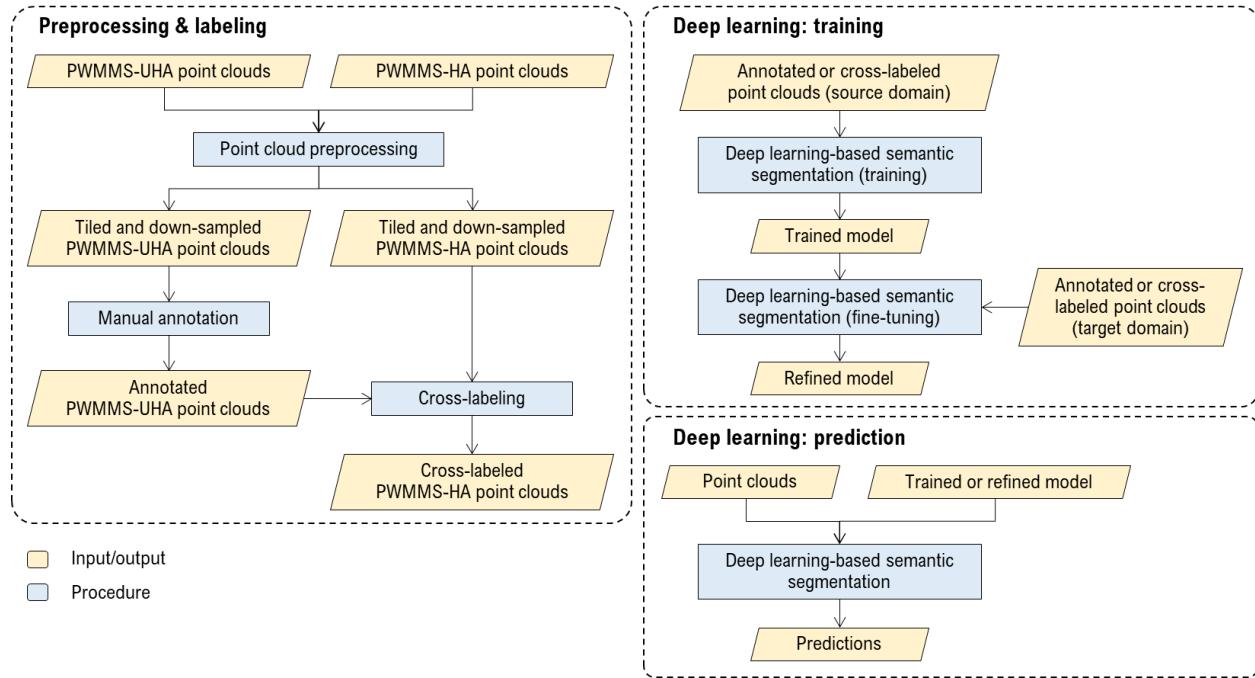


Figure 6-4 Framework of the proposed deep learning-based semantic segmentation for bridge components and road elements.

6.3.1 Point cloud preprocessing

Point clouds acquired by ground MLMS typically have a high point density near the trajectory. The point density drops drastically as the distance from the trajectory increases. The varying point density could lead to an unbalanced number of points among different classes, which harms the performance of the deep learning network. To mitigate potential issues caused by varying point density, point clouds are down-sampled to keep a minimum point distance of n_s cm, where n_s is a user-defined value. Figure 6-5 shows the planimetric point density (points per square meter, ppsm) along with the trajectory of a sample PWMMS-UHA point cloud before and after down-sampling. The minimum point spacing for down-sampling is 5 cm. While substantial variation in point density can be observed in the original point cloud (Figure 6-5a), the point density becomes much more homogeneous after down-sampling. A close view of the point cloud over a traffic sign before and after down-sampling is shown in Figure 6-6. While the number of points on the sign board and poles reduces, the shape of the sign board is retained.

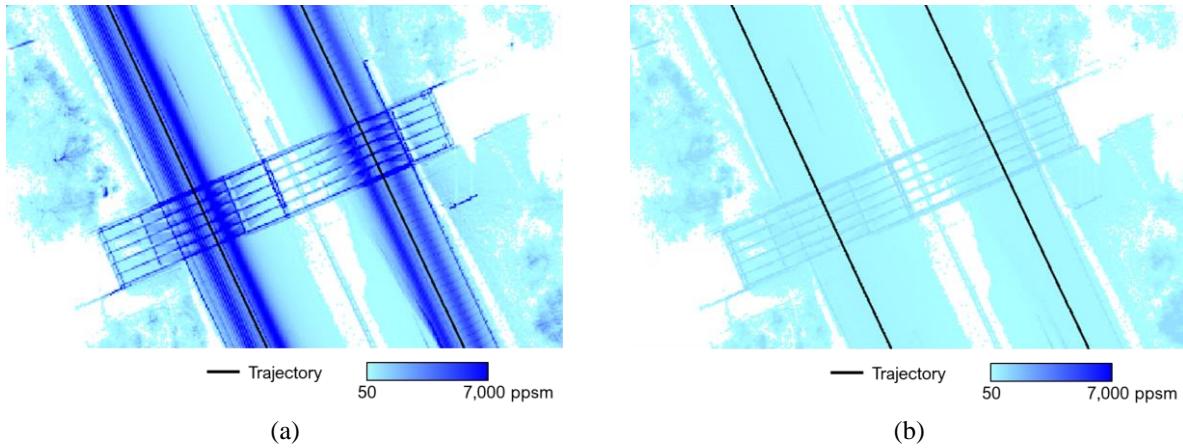


Figure 6-5 Planimetric point density map for (a) the original point cloud and (b) down-sampled point cloud with a minimum point spacing of 5 cm (Bridge 2 in the PWMMS-UHA bridge dataset).

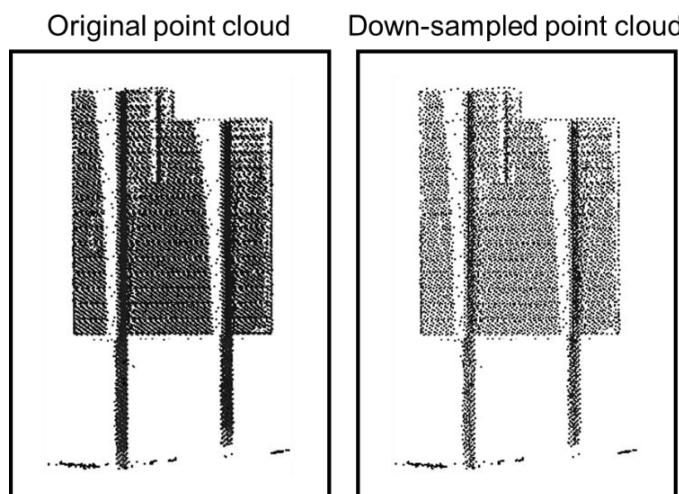


Figure 6-6 Example of point cloud down-sampling showing (a) the original point cloud and (b) down-sampled point cloud with a minimum point spacing of 5 cm.

6.3.2 Point cloud annotation

The aim of the semantic segmentation in this study is to simultaneously classify bridge components and road elements. We divide a bridge into three main parts (illustrated in Figure 6-7): 1) horizontal components – the deck, guardrail, and beam/girder, 2) vertical components on the two ends – the abutment and wing wall, and 3) vertical supports at the middle – the pier. The following nine classes are used for bridge and road elements:

- Bridge – deck, guardrail, and beam/girder
- Bridge – abutment and wing wall
- Bridge – pier
- Man-made terrain: roads
- Natural terrain: grass and ditches
- Vegetation: trees and bushes
- Buildings: buildings, retaining walls, noise barriers, etc.
- Remaining hardscape: guardrails, traffic signs, traffic lights, utility poles, etc.
- Scanning artifacts: artifacts caused by dynamically moving objects

Point cloud annotation is carried out using the procedure described in Hackel et al. [191]. The user performs rotation and cropping of the point cloud repeatedly until all points in the remaining segment belong to the same class. The open-source software – CloudCompare [218] – is used for this procedure. A sample manual annotation result for Bridge 2 in the PWMMS-UHA bridge dataset is shown in Figure 6-8.

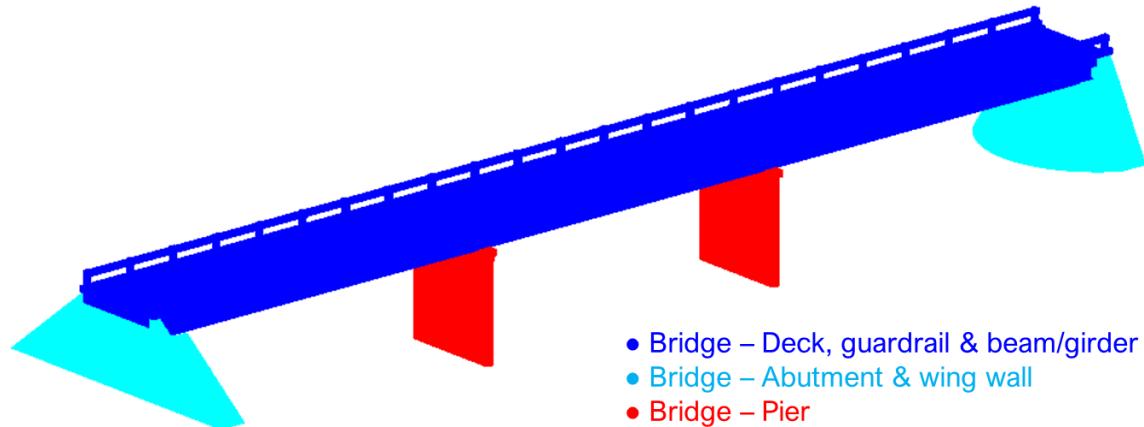


Figure 6-7 Classes for bridge components: deck, guardrail, beam, and girder in blue; abutment and wing wall in cyan; pier in red.

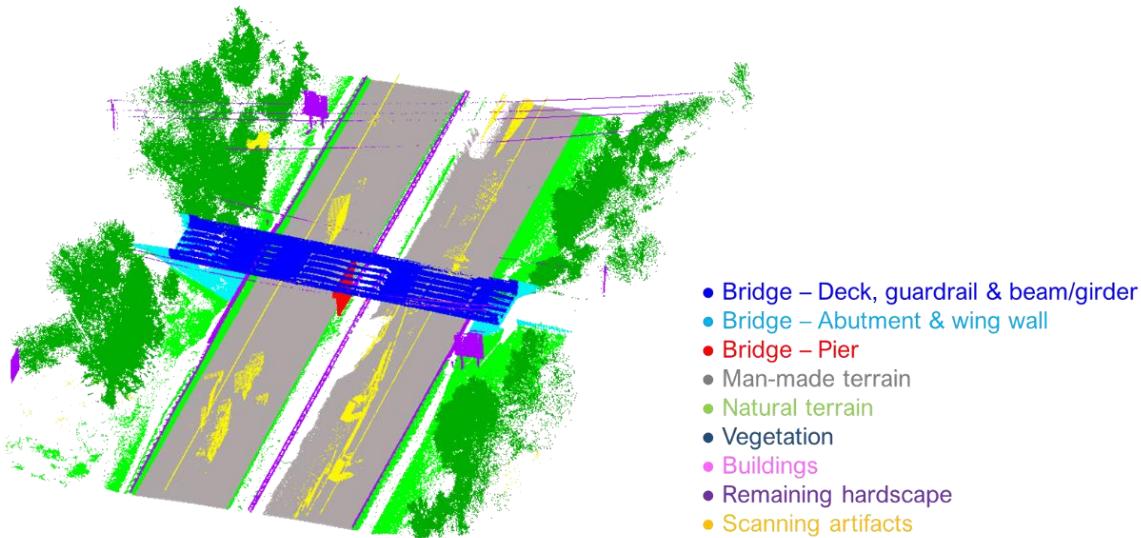


Figure 6-8 Sample annotated point cloud colored by class (Bridge 2 in the PWMMS-UHA bridge dataset).

6.3.3 Cross-labeling

For MLMSs that utilize direct georeferencing, centimeter-level positional accuracy can be expected if the quality of trajectory from the GNSS/INS unit and system calibration parameters are guaranteed. Taking advantage of the good geolocation quality and low noise level, manual annotation conducted on a source dataset can easily be transferred to a target dataset (i.e., dataset from a different system or date) as long as the two datasets cover the same area. Starting with a point in the target dataset, its closest point in the source dataset is identified. If the distance between the two points is smaller than a user-defined threshold, the point in the target dataset is assigned the same label as the point in the source dataset. Figure 6-9 demonstrates the geolocation quality of data from the two MLMSs used in this study and shows sample cross-labeling results. In Figure 6-9b, the manual annotation is conducted on the source PWMMS-UHA dataset and transferred to the target PWMMS-HA dataset. One should note that the coverage of the cross-labeled data is inherently limited by the coverage of the source data. More specifically, an area in the target dataset cannot be cross-labeled if it is not captured in the source dataset. In addition, the underlying assumption of cross-labeling is that the scene remains unchanged across data acquisitions. In this study, the source and target datasets were captured at the same date and thus the highway scenes remain the same. The scanning artifacts in the two datasets, on the other hand, show a clear disparity since the artifacts are mainly caused by moving vehicles during data acquisition and their characteristics are subject to the scanning mechanism and noise level of the LiDAR units. As a result, proper transfer of labels can be expected for all classes except scanning artifacts.

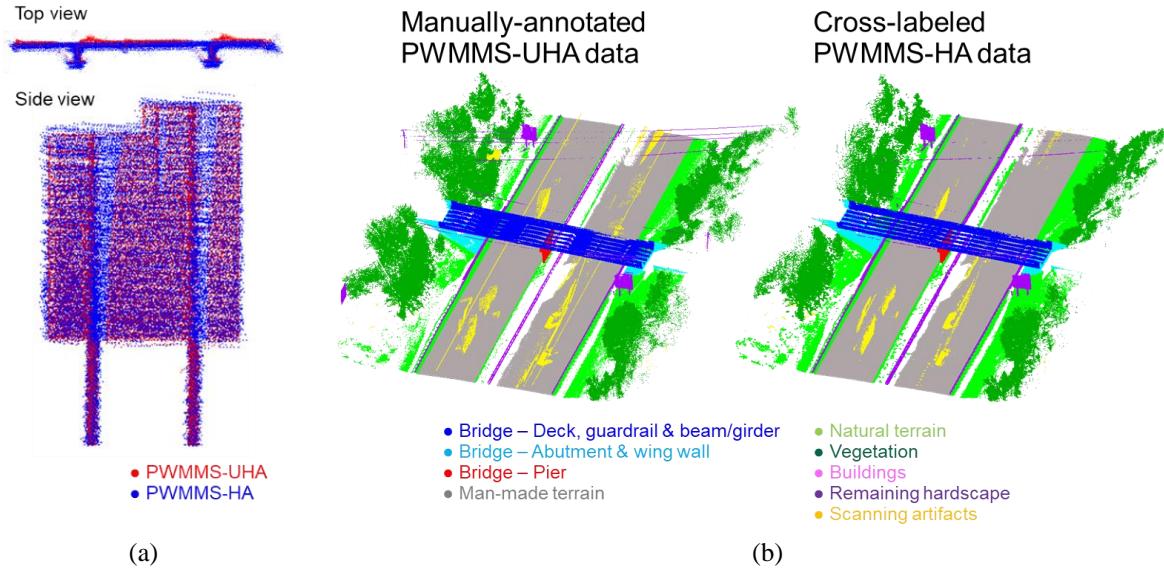


Figure 6-9 Illustration of the cross-labeling approach: (a) data alignment between datasets acquired by different MLMSs and (b) **manually-annotated** PWMMS-UHA data and the corresponding cross-labeled PWMMS-HA data (Bridge 2).

6.3.4 Deep learning-based point cloud semantic segmentation

Point cloud semantic segmentation aims at linking each point in the point cloud to a class label. The SPG framework [190] is adopted in this study for its ability to consider the spatial context at an object level, learn rich contextual information, and handle point clouds over large spatial extents. The SPG framework has four main steps: geometrically homogeneous partition, superpoint graph construction, superpoint embedding, and contextual segmentation [190].

The first step partitions the point cloud into geometrically simple yet meaningful clusters (hereafter, denoted as superpoints). For every point in the point cloud, a local neighborhood is defined based on its n_m nearest neighbors. A principal component analysis (PCA) is conducted and the following geometric features are evaluated using the eigenvalues and eigenvectors: linearity, planarity, scattering, and verticality, as introduced in Guinard and Landrieu [219]. The optimization model for geometrically homogeneous partition is defined based on the geometric features and the nearest neighbor adjacency graph. Figure 6-10 shows a sample partition results for Bridge 2 in the PWMMS-UHA bridge dataset. The second step constructs the adjacency relationship of the superpoints (as illustrated in Figure 6-11a). A Voronoi adjacency graph is defined using the entire point cloud. The SPG is a directed graph whose nodes are the superpoints and edges represent the adjacency relationship of the superpoints. Each superpoint is a geometrically homogeneous partition. Two superpoints SP1 and SP2 are adjacent if there is at least one edge in the Voronoi adjacency graph with one end in SP1 and one end in SP2. The last two steps, superpoint embedding and contextual segmentation, are the deep learning components (see Figure 6-11b for an illustration). PointNet is used for superpoint embedding, i.e., to compute a descriptor for each superpoint. In PointNet, the input points and features are aligned by a spatial transformer network (STN), individually processed by multi-layer perceptron (MLP), and max-pooled to aggregate the information from all the points. In this study, each point is represented by

its coordinates, geometric features, and planimetric distance to the trajectory (i.e., the 2D distance between the point and its closest trajectory point). While geometric features (i.e., linearity, planarity, scattering, and verticality) characterize properties in a local neighborhood, planimetric distance to the trajectory describes the global position of points with respect to the road. Finally, the contextual segmentation classifies each superpoint based on its embedding and its local surroundings within the SPG. The architecture utilizes graph convolutions, including gated graph neural networks [220] and edge-conditioned convolutions [221].

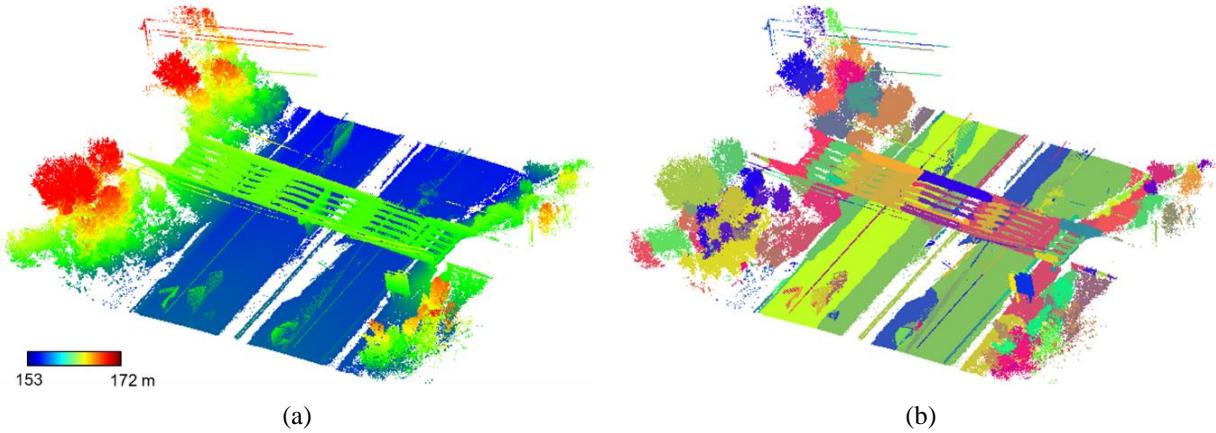


Figure 6-10 Sample geometrically homogeneous partition result (Bridge 2 in the PWMMS-UHA bridge dataset): (a) point cloud colored by height and (b) point cloud randomly colored by partition ID.

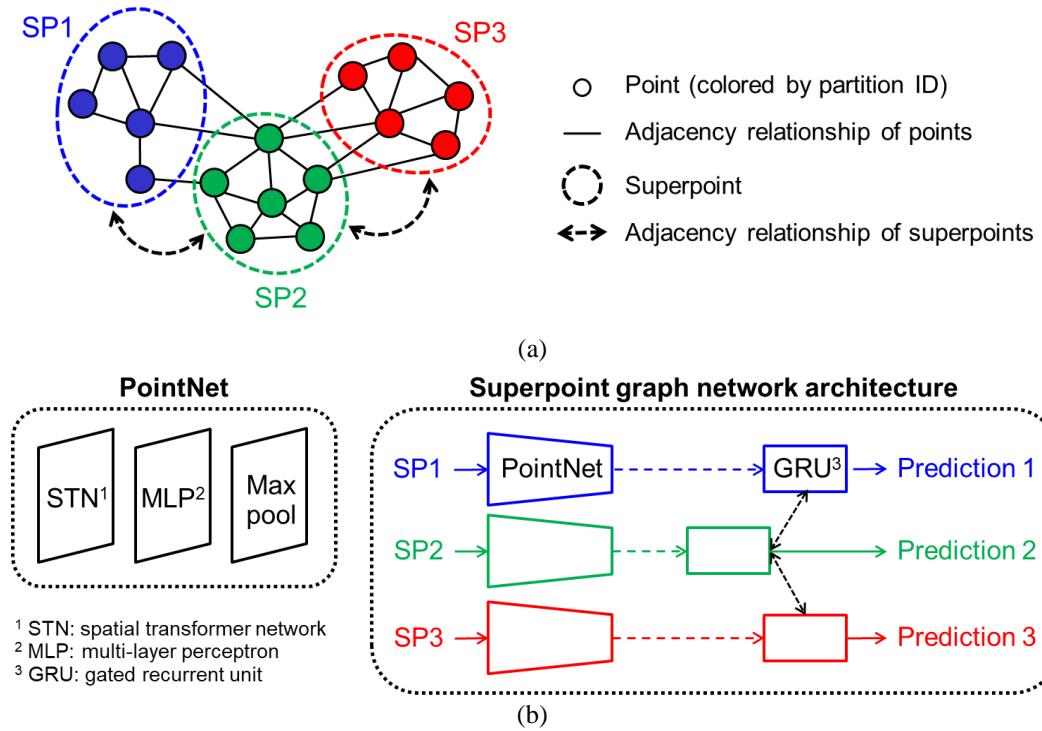


Figure 6-11 Schematic diagram of the superpoint graph (SPG) approach: (a) superpoint graph construction and (b) superpoint embedding and contextual segmentation.

In this study, each dataset is divided into the following: a training and validation set and an independent testing set. Three-fold cross-validation [222] is adopted for the training stage, as depicted in Figure 6-12. Here, the goal is to find the best model and number of epochs. First, the training and validation set is divided into 3 subsets. The training process is repeated three times. In each trial, two of the subsets are used for training and the remaining one is used for validation. The best epoch (i.e., the epoch that has the lowest validation loss) in each trial, k_i , is recorded, where i denotes the i^{th} trial ($i = 1, 2, 3$). Finally, the best model is trained using the full training and validation set with k^* epochs, where k^* is the average of k_i . Using this strategy, all the data in the training and validation set can contribute to the training of the best model. The performance of the best model is evaluated using the independent testing set in the testing stage.

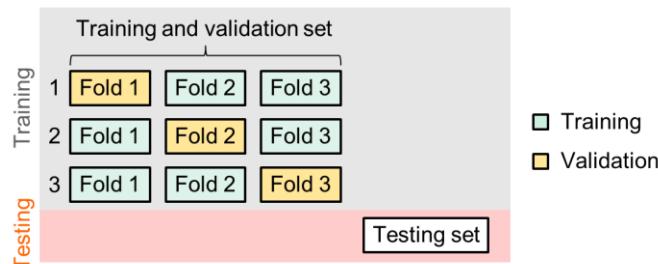


Figure 6-12 Schematic diagram for the training (three-fold cross-validation) and testing stage.

6.3.5 Transfer learning

Transfer learning aims to leverage knowledge from a source domain to improve the learning process in a target domain by minimizing the amount of training data required. The conceptual basis of transfer learning is that a deep neural network can be divided into two parts: the first part acts as a feature extractor and the features are versatile; the second part focuses on the given tasks (e.g., classification and segmentation) [203]. Therefore, a model trained on source domain can be fine-tuned with minimal data from the target domain. The fine-tuning process adjusts the parameters in the second part of the model while fixing the parameters in the first part. In this study, the SPG model consists of PointNet and graph convolutions. The transfer learning approach fine-tunes the parameters in the fully connected layers after max pooling in PointNet and the parameters in the graph convolutions, as illustrated in Figure 6-13.

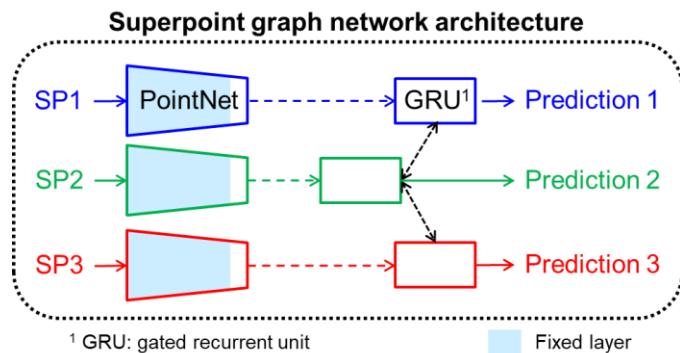


Figure 6-13 Schematic diagram of the SPG approach showing the fixed layers for transfer learning in blue.

6.4 Experimental Results

Several experiments were carried out to evaluate the performance of the deep learning-based semantic segmentation framework, as well as the proposed cross-labeling and transfer learning approaches. The PWMMS-UHA bridge, PWMMS-HA bridge, PWMMS-UHA highway, and PWMMS-HA highway datasets were used for the experiments.

In this study, manual annotation was conducted on the PWMMS-UHA datasets because point clouds from PWMMS-UHA have a very low noise level. The high point cloud quality makes it easy for a user to identify and annotate individual objects, and thus less prone to human errors. Nevertheless, manual annotation is a time-consuming process – it typically took one to several hours for a user to label one bridge/tile, depending on the complexity of the scene. For the PWMMS-UHA bridge dataset, all the 27 bridges were manually annotated. For the PWMMS-UHA highway dataset, 23 out of 554 tiles (Tiles 2, 14, 25, 30, 34, 80, 121, 169, 212, 227, 268, 319, 340, 376, 442, 445, 454, 468, 491, 497, 499, 518, and 544) were selected and manually annotated. The selected tiles cover a variety of highway scenes, including different types of guardrails, traffic signs, utility poles, noise barriers, retaining walls, and buildings. The manually-annotated PWMMS-UHA datasets were then used to cross-label their respective PWMMS-HA datasets. In addition, one bridge (Bridge 3) and two highway tiles (Tiles 25 and 212) from the PWMMS-HA datasets were selected and manually annotated for the cross-labeling performance assessment (as will be discussed in Section 6.4.1).

Several performance measures are used in this study: percentage of true positive (TP), false positive (FP), and false negative (FN), precision (Equation 6-1), recall (Equation 6-2), F1-score (Equation 6-3), and intersection of union (IoU) as defined per Equation 6-4, where A is the cluster of prediction and B is the cluster of ground truth. The above-mentioned metrics are evaluated for each class. In addition, the overall accuracy, as given in Equation 6-5, is evaluated for the entire point cloud.

$$Precision = \frac{TP}{TP + FP} \quad \text{Equation 6-1}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{Equation 6-2}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{Equation 6-3}$$

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad \text{Equation 6-4}$$

$$Overall\ accuracy = \frac{TP}{Number\ of\ points} \quad \text{Equation 6-5}$$

6.4.1 Cross-labeling performance assessment

In this experiment, the performance of the proposed cross-labeling approach is evaluated. One bridge (Bridge 3) and two highway tiles (Tiles 25 and 212) from the PWMMS-HA datasets were selected and manually annotated for this experiment. The cross-labeled results were compared against the manually-annotated ground truth to assess the quality of the transferred labels.

Figure 6-14 presents the cross-labeled results and manually-annotated reference for Bridge 3, Tile 25, and Tile 212. The reduced coverage over the ditches and vegetation in the cross-labeled point clouds is attributed to the limited spatial coverage of the PWMMS-UHA data. Another noticeable difference between the cross-labeled results and manually-annotated reference is the scanning artifacts, which is expected as explained in Section 6.3.3. Table 6-2 reports the quantitative performance measures. The proposed cross-labeling strategy achieved a precision higher than 0.97 for all classes except scanning artifacts. The slightly higher false negative rates (and thus lower recall) for the natural terrain and vegetation are mainly related to the difference in the spatial coverage between the two systems. The poor performance for scanning artifacts is not surprising as the two systems conducted the scans at different times.

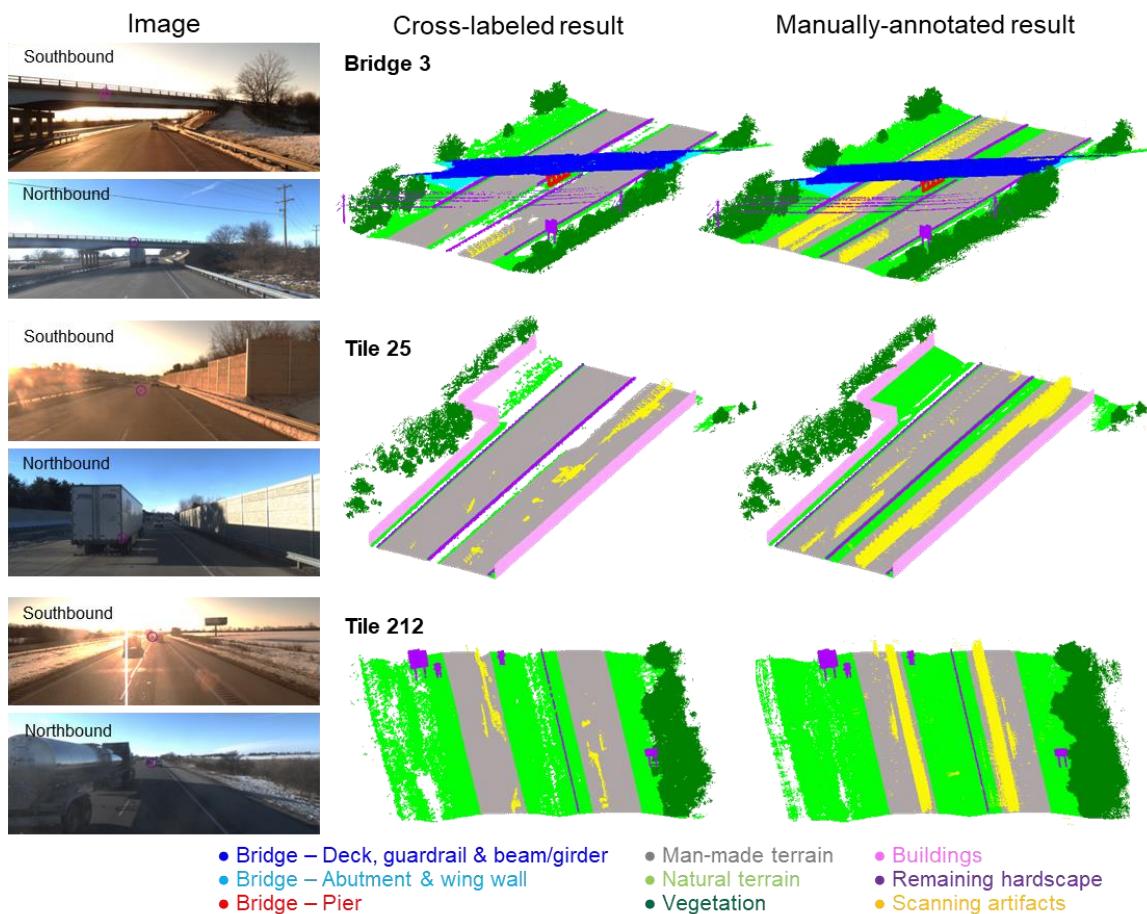


Figure 6-14 Cross labeled results and manually-annotated reference (Bridge 3, Tile 25, and Tile 212 from PWMMS-HA).

Table 6-2 Performance measures for the proposed cross-labeling approach.

	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.899	0.000	0.101	1.000	0.899	0.947	0.899
Bridge – Abutment	0.979	0.000	0.021	1.000	0.979	0.989	0.979
Bridge – Pier	0.990	0.009	0.001	0.991	0.999	0.995	0.990
Man-made terrain	0.972	0.005	0.023	0.995	0.977	0.986	0.972
Natural terrain	0.715	0.016	0.269	0.978	0.727	0.834	0.715
Vegetation	0.705	0.018	0.276	0.974	0.719	0.827	0.705
Buildings	0.994	0.005	0.001	0.995	0.999	0.997	0.994
Remaining hardscape	0.945	0.025	0.031	0.974	0.969	0.972	0.945
Scanning artifacts	0.050	0.016	0.934	0.752	0.051	0.095	0.050
Overall accuracy	85% (89% while excluding scanning artifacts)						

6.4.2 Deep learning-based semantic segmentation: baseline model

In this experiment, the performance of the deep learning-based semantic segmentation approach to classify bridge components and road elements is evaluated. The PWMMS-UHA bridge dataset was divided into the following: a training and validation set (21 bridges – Bridges 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 19, 20, 22, 23, 24, 25, 26, 27) and an independent testing set (6 bridges – Bridges 2, 5, 10, 15, 18, 21). The 6 bridges selected for the testing set cover different types of bridges presented in the area of interest. A baseline model (hereafter, denoted as Model A) was developed:

- Model A (baseline):
 - Trained and validated on the PWMMS-UHA bridge dataset (21 bridges)

The performance of Model A was evaluated using the independent testing set (6 bridges). Figure 6-15 shows the prediction from Model A for the six bridges. As evident in the figure, the SPG model was able to classify the objects as a whole, e.g., all the points belonging to a signboard would have a single label – hardscape. Focusing on the bridge components, the model successfully predicted most of the bridge decks and piers whereas it had some difficulty distinguishing abutments from the surrounding natural terrain. Misclassification commonly happens along the boundaries between man-made terrain, natural terrain, and vegetation. This problem is mainly related to the geometric partition since the boundaries between these classes are sometimes ill-defined. For Bridge 18, the overhead traffic sign was misclassified as bridge components. The performance metrics, including the percentage of TP, FP, and FN, precision, recall, F1-score, IoU, together with the overall accuracy, is reported in Table 6-3. Model A achieved an overall accuracy of 84% on the PWMMS-UHA bridge dataset, which will serve as a baseline for evaluating transfer learning performance. The results suggest that bridge decks and piers can be classified with high accuracy. The abutment, on the other hand, has low accuracy. The results are consistent with previous research findings as reported in Kim and Kim [198].

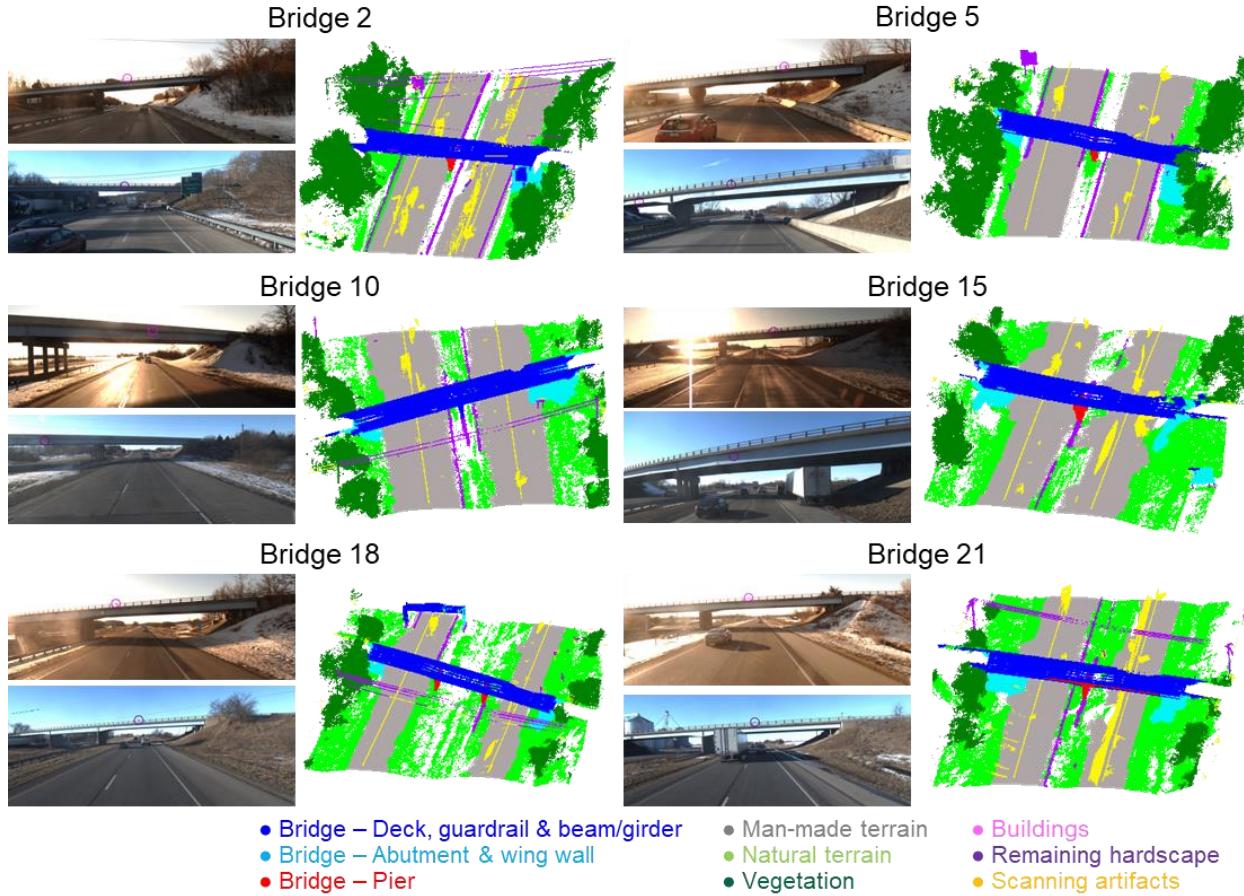


Figure 6-15 Predictions from Model A for the testing set (Bridges 2, 5, 10, 15, 18, 21 from the PWMMS-UHA bridge dataset).

Table 6-3 Performance of Model A on the testing set from the PWMMS-UHA bridge dataset.

	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.896	0.086	0.018	0.912	0.980	0.945	0.914
Bridge – Abutment	0.400	0.407	0.193	0.495	0.674	0.571	0.412
Bridge – Pier	0.819	0.061	0.120	0.931	0.872	0.901	0.804
Man-made terrain	0.785	0.213	0.002	0.787	0.997	0.880	0.704
Natural terrain	0.530	0.035	0.435	0.938	0.549	0.693	0.457
Vegetation	0.897	0.081	0.023	0.918	0.975	0.946	0.888
Buildings	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.590	0.122	0.288	0.829	0.672	0.742	0.582
Scanning artifacts	0.757	0.142	0.101	0.842	0.882	0.862	0.717
Overall accuracy				84%			

6.4.3 Transfer learning between scenes

Upon verifying the performance of the baseline model, this experiment explores the transferability of the deep learning-based semantic segmentation for handling point clouds capturing different scenes. In this experiment, the source domain is the PWMMS-UHA bridge dataset; the target domain is the PWMMS-UHA highway dataset. The PWMMS-UHA highway dataset was divided into the following: a training and validation set (17 tiles – Tiles 2, 25, 30, 34, 80, 121, 169, 227, 268, 319, 442, 454, 468, 491, 497, 518, 544) and an independent testing set (6 tiles – Tiles 14, 212, 340, 376, 445, 499). The 6 tiles selected for the testing set contain different road elements. Three models were developed:

- Model A (baseline):
 - Trained and validated on the PWMMS-UHA bridge dataset (21 bridges)
- Model B:
 - Trained and validated on the PWMMS-UHA bridge dataset (21 bridges)
 - Fine-tuned with the PWMMS-UHA highway dataset (6 tiles – Tiles 25, 80, 442, 454, 468, 497)
- Model C:
 - Trained and validated on the PWMMS-UHA highway dataset (17 tiles)

The three models were tested on the testing set from the PWMMS-UHA highway dataset (6 tiles). Model A (the baseline model) was trained with the source domain data and thus was expected to have the lowest accuracy. Model C was trained with the target domain data and thus was expected to have the highest accuracy. Figure 6-16 shows the predictions from Models A, B, and C for the six tiles and the performance metrics are reported in Table 6-4. According to Figure 6-16, common errors include: misclassifying roads as natural terrain (Tile 340), misclassifying overhead signs as bridge components (Tile 445), and misclassifying buildings as hardscape, bridge components, or natural terrain (Tiles 14 and 499). Looking into the metrics reported in Table 6-4, the performance of Model A is only slightly lower than that of Model C. This result indicates that the model trained with the PWMMS-UHA bridge dataset has acceptable performance on the PWMMS-UHA highway dataset, which is reasonable since most highway components are present in the bridge dataset. As a result, the improvement after applying transfer learning is marginal. Another observation is that all the models have poor ability for identifying buildings, which is most likely related to the lack of training samples for this class.

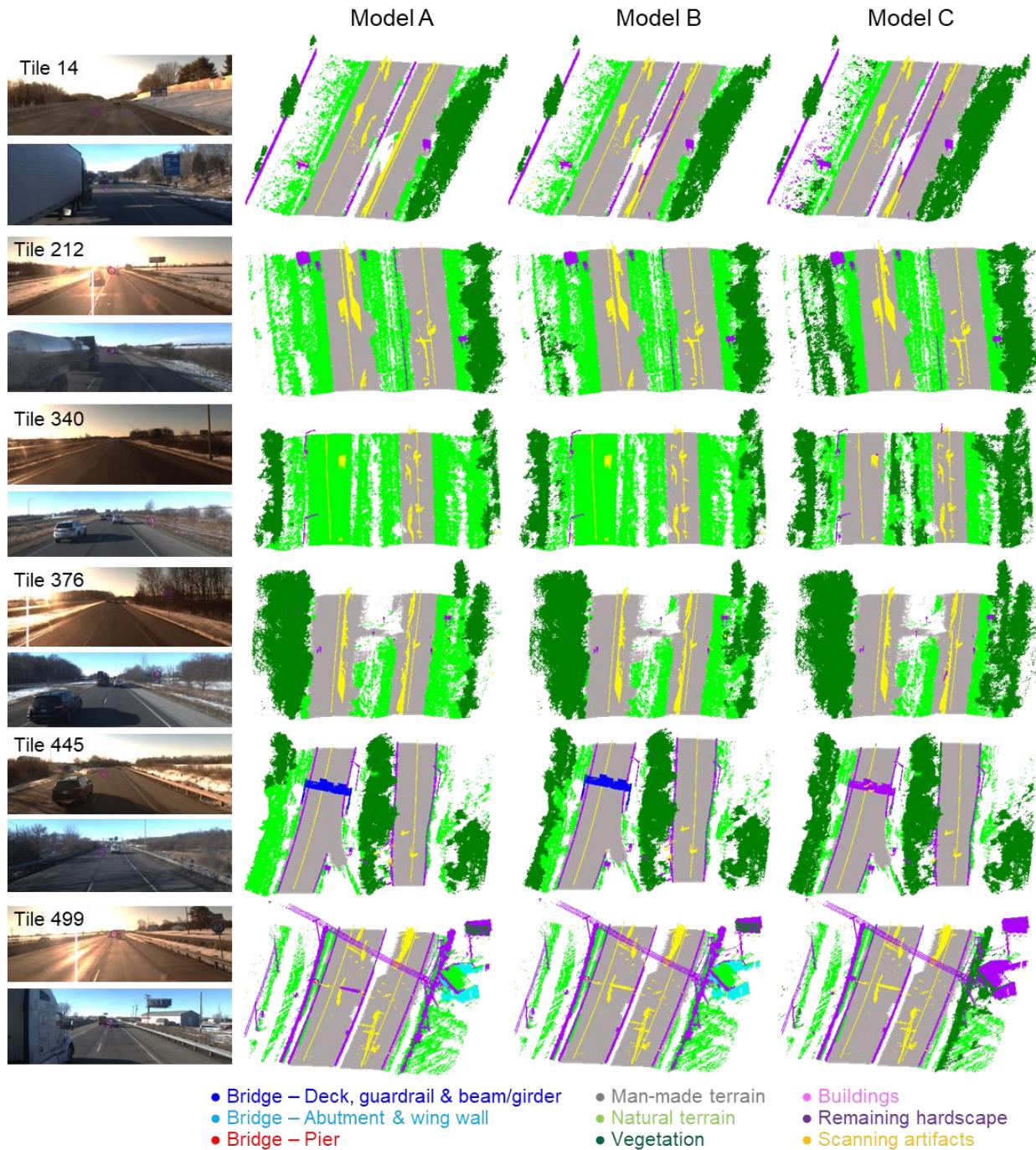


Figure 6-16 Predictions from Models A, B, and C for the testing set (Tiles 14, 212, 340, 376, 445, 499 from the PWMMS-UHA highway dataset).

Table 6-4 Performance of Models A, B, and C on the testing set from the PWMMS-UHA highway dataset.

Model A							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.789	0.128	0.083	0.860	0.905	0.882	0.731
Natural terrain	0.492	0.279	0.230	0.638	0.682	0.659	0.446
Vegetation	0.836	0.014	0.150	0.983	0.848	0.911	0.811
Buildings	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.481	0.433	0.086	0.526	0.848	0.649	0.559
Scanning artifacts	0.946	0.023	0.031	0.976	0.968	0.972	0.964
Overall accuracy	83%						
Model B							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.790	0.128	0.082	0.860	0.906	0.882	0.731
Natural terrain	0.498	0.222	0.279	0.692	0.641	0.665	0.449
Vegetation	0.873	0.038	0.090	0.959	0.907	0.932	0.851
Buildings	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.454	0.464	0.082	0.494	0.847	0.624	0.521
Scanning artifacts	0.846	0.028	0.126	0.968	0.870	0.916	0.842
Overall accuracy	84%						
Model C							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.860	0.137	0.003	0.863	0.997	0.925	0.800
Natural terrain	0.391	0.021	0.588	0.948	0.399	0.562	0.357
Vegetation	0.827	0.153	0.020	0.844	0.976	0.905	0.807
Buildings	0.000	0.000	1.000	1.000	0.000	0.000	0.000
Remaining hardscape	0.487	0.459	0.053	0.515	0.901	0.655	0.551
Scanning artifacts	0.801	0.016	0.183	0.980	0.814	0.889	0.792
Overall accuracy	85%						

6.4.4 Transfer learning between scenes and systems

The last experiment tackles the most challenging scenario: transfer learning between different scenes and systems. In this experiment, the source domain is the PWMMS-UHA bridge dataset; the target domain is the PWMMS-HA highway dataset. The PWMMS-HA highway dataset was divided into the following: a training and validation set (17 tiles – Tiles 2, 25, 30, 34, 80, 121, 169, 227, 268, 319, 442, 454, 468, 491, 497, 518, 544) and an independent testing set (6 tiles – Tiles 14, 212, 340, 376, 445, 499). Three models were developed:

- Model A (baseline):
 - Trained and validated on the PWMMS-UHA bridge dataset (21 bridges)
- Model D:
 - Trained and validated the PWMMS-UHA bridge dataset (21 bridges)
 - Fine-tuned with the PWMMS-HA highway dataset (6 tiles – Tiles 25, 80, 442, 454, 468, 497)
- Model E:
 - Trained and validated the PWMMS-HA highway dataset (17 tiles)

All the models were tested on the testing set from the PWMMS-HA highway dataset (6 tiles). One should note that the cross-labeled PWMMS-HA dataset was used for training and testing in this experiment, i.e., no manual work was involved for the annotation of the PWMMS-HA data. Model A (the baseline model) was trained with the source domain data and thus was expected to have the lowest accuracy. Model E was trained with the target domain data and thus was expected to have the highest accuracy. Figure 6-17 shows the predictions from Models A, D, and E for the six tiles and the performance metrics are reported in Table 6-5. As shown in Figure 6-17, Model A tends to misclassify man-made terrain as natural terrain. One possible reason is that the PWMMS-HA data has a higher noise level as compared to the PWMMS-UHA data, and such difference is particularly noticeable along the road surface. Other problems include misclassifying overhead signs as bridge components (Tile 445) and misclassifying buildings as hardscape, bridge components, or natural terrain (Tile 499). According to Table 6-5, the performance of Model A is significantly lower than the performance of Model E (the overall accuracy of Model A is 10% lower than that of Model E). Particularly, the low performance of Model A on the man-made terrain is the source of such difference. This result indicates that the model trained with the PWMMS-UHA bridge dataset does not generalize well on the PWMMS-HA highway dataset. Model D, in contrast, achieved slightly better performance as compared to Model E (the overall accuracy of Models D and E is 84% and 82%, respectively). Based on the IoU, the better performance is mainly from the natural terrain and vegetation, which could be attributed to the geometric partitions and large number of points in the two classes. One should note that Model D used only 6 training sets from the target domain while Model E used 17 training sets. This result suggests that the proposed transfer learning strategy can effectively boost the performance of a pre-trained model on a target domain with a small amount of training data.

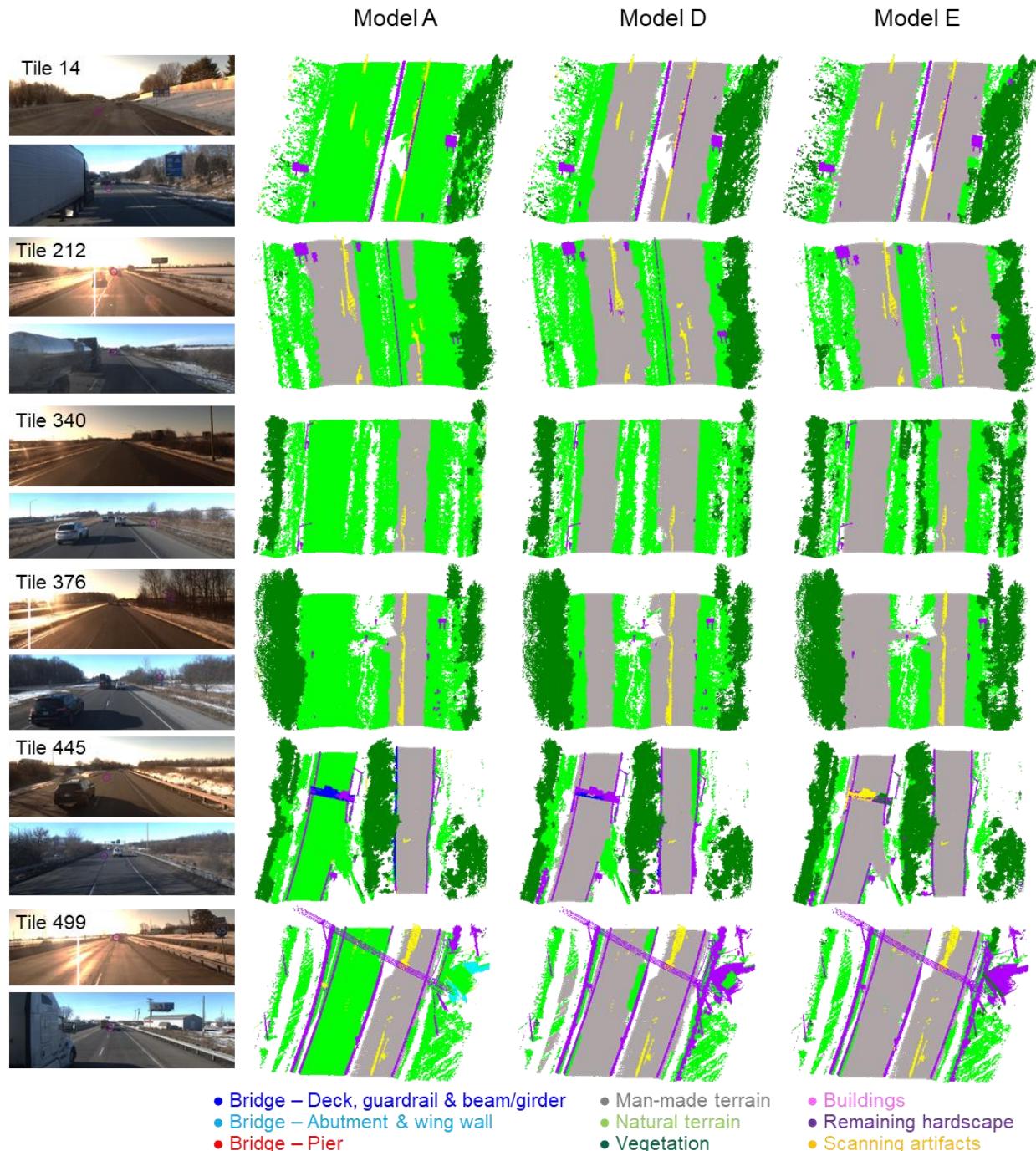


Figure 6-17 Predictions from Models A, D, and E for the testing set (Tiles 14, 212, 340, 376, 445, 499 from the PWMMS-HA highway dataset).

Table 6-5 Performance of Models A, D, and E on the testing set from the PWMMS-HA highway dataset.

Model A							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.368	0.069	0.562	0.841	0.396	0.538	0.366
Natural terrain	0.498	0.445	0.057	0.528	0.897	0.664	0.443
Vegetation	0.841	0.019	0.140	0.978	0.857	0.913	0.818
Buildings	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.587	0.263	0.150	0.691	0.797	0.740	0.693
Scanning artifacts	0.585	0.061	0.354	0.906	0.623	0.738	0.415
Overall accuracy	72%						
Model D							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.775	0.190	0.034	0.803	0.958	0.874	0.792
Natural terrain	0.612	0.121	0.266	0.835	0.697	0.760	0.590
Vegetation	0.846	0.045	0.109	0.950	0.886	0.917	0.827
Buildings	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.506	0.445	0.049	0.532	0.911	0.672	0.676
Scanning artifacts	0.587	0.011	0.402	0.982	0.593	0.740	0.446
Overall accuracy	84%						
Model E							
	TP	FP	FN	Precision	Recall	F1-score	IoU
Bridge – Deck	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Bridge – Abutment	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Bridge – Pier	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Man-made terrain	0.769	0.229	0.002	0.771	0.997	0.869	0.786
Natural terrain	0.513	0.058	0.429	0.899	0.545	0.679	0.492
Vegetation	0.819	0.113	0.069	0.879	0.923	0.900	0.799
Buildings	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Remaining hardscape	0.516	0.392	0.092	0.569	0.849	0.681	0.664
Scanning artifacts	0.568	0.089	0.344	0.865	0.623	0.724	0.400
Overall accuracy	82%						

6.5 Conclusions and Future Work

Although several factors come into play in GNSS/INS navigation, mobile LiDAR data can achieve centimeter-level positional accuracy if the quality of the resulting trajectory and system calibration parameters are accurately established. This study exploited the good geolocation quality and developed a deep learning-based framework for semantic segmentation of bridge components and road elements using mobile LiDAR data. Cross-labeling and transfer learning strategies were proposed to aid the network in adapting to varying characteristics of the acquired data and **easing** the pain of manual annotation. Experimental results showed that the proposed cross-labeling approach can effectively transfer labels from one mobile mapping system to another for all classes except for scanning artifacts. The baseline model trained with the PWMMS-UHA bridge dataset achieved an overall accuracy of 84% for classifying bridge components and road elements. Bridge decks and piers can be classified with high accuracy while abutments are difficult to predict. In terms of generalization, the baseline model had good performance on the PWMMS-UHA highway dataset (overall accuracy: 83%) but not on the PWMMS-HA highway dataset (overall accuracy: 72%). By fine-tuning the baseline model with six training sets from the PWMMS-HA highway dataset, its overall accuracy increased from 72% to 84% and reached the same level as compared to a model trained on the PWMMS-HA dataset.

Future research will focus on developing geometrically-based quality control strategies to detect, report, and fix, if possible, misclassifications as well as assess the quality of segmentation results. Moreover, potential approaches for automated identification of critical training samples will be explored. Finally, the framework will be implemented to utilize data from consumer-grade mobile LiDAR systems.

7. CONCLUSIONS

Mobile mapping systems carrying a sensor suite of camera, LiDAR, and GNSS/INS units have emerged as a prominent tool for collecting high-quality georeferenced data in an efficient manner. The quality of the mapping products is contingent on the grade of the onboard sensors, the reliability of system calibration, and the accuracy of GNSS/INS trajectory. This document discussed the main components of vehicle-based mobile mapping with regards to GNSS/INS navigation, as illustrated in Figure 7-1.

- Chapter 2 presented the background concept of GNSS/INS navigation and reasoned how GNSS hardware and data processing techniques impact the overall post-processed trajectory quality.
- Chapter 3 reviewed some of the major environmental factors that impact the accuracy of the GNSS/INS-derived trajectory upon which the entire mapping prospective is built. Factors such as bridges, tree canopy, neighboring traffic, and electromagnetic interferences independently or in combination may pose a serious challenge to the positioning efficiency of GNSS/INS systems.
- Considering potential trajectory deterioration, Chapter 4 presented a **data-driven** error mitigation approach through point cloud registration. The results showed that post-registration alignment quality for mobile LiDAR point clouds is in the range of 1 cm to 2 cm.
- Chapter 5 introduced a system-driven error mitigation approach by means of trajectory enhancement. The approach eliminates initial inaccuracies within the trajectory position and orientation parameters ranging up to 1.30 m and 2.50°, respectively.
- Finally, once the trajectory-related issues were accounted for, Chapter 6 developed a novel framework for semantic segmentation of highway elements using the enhanced mapping products. The framework can handle large volume of data acquired through long-corridor road surveys and achieved an overall accuracy of 84% for point cloud semantic segmentation.

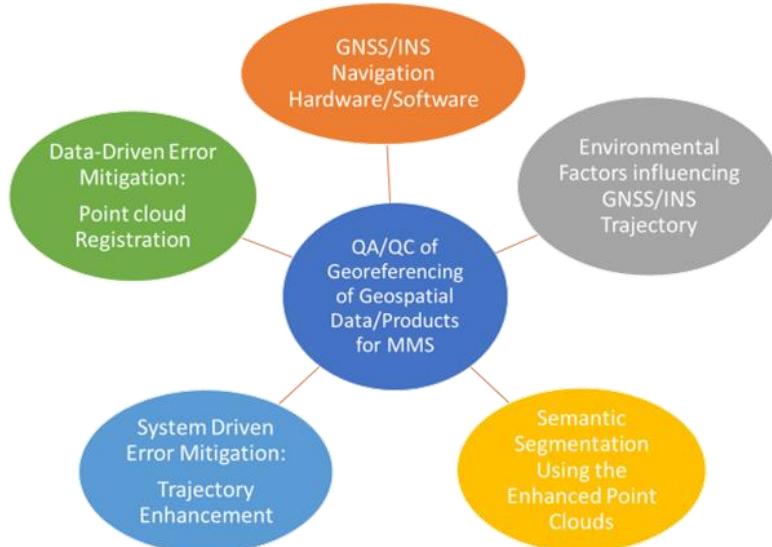


Figure 7-1 Main components of vehicle-based mobile mapping with regards to GNSS/INS navigation.

REFERENCES

- [1] C.K. Toth, R&D of mobile LiDAR mapping and future trends, in: American Society for Photogrammetry and Remote Sensing Annual Conference 2009, ASPRS 2009, 2009.
- [2] I. Puente, H. González-Jorge, J. Martínez-Sánchez, P. Arias, Review of mobile mapping and surveying technologies, Measurement: Journal of the International Measurement Confederation. 46 (2013). <https://doi.org/10.1016/j.measurement.2013.03.006>.
- [3] H. Guan, J. Li, S. Cao, Y. Yu, Use of mobile LiDAR in road information inventory: A review, International Journal of Image and Data Fusion. 7 (2016). <https://doi.org/10.1080/19479832.2016.1188860>.
- [4] F. di Stefano, S. Chiappini, A. Gorreja, M. Balestra, R. Pierdicca, Mobile 3D scan LiDAR: a literature review, Geomatics, Natural Hazards and Risk. 12 (2021). <https://doi.org/10.1080/19475705.2021.1964617>.
- [5] J. Sickle, GPS for Land Surveyors, CRC Press, 2008.
- [6] Apollo PGNCS - Wikipedia, (2022). https://en.wikipedia.org/wiki/Apollo_PGNCS (accessed May 12, 2022).
- [7] Honeywell Inertial Measurement Units, (2022). <https://aerospace.honeywell.com/us/en/learn/products/sensors/inertial-measurement-units> (accessed May 7, 2022).
- [8] Novatel IMU-ISA-100C, (2022). https://docs.novatel.com/OEM7/Content/Technical_Specs_IMU/ISA_100C_Overview.htm (accessed April 25, 2020).
- [9] SDI500 Tactical Grade IMU | EMCORE, (2022). <https://emcore.com/products/sdi500-tactical-grade-imu-inertial-measurement-unit/> (accessed May 8, 2022).
- [10] W. Zhao, Y. Cheng, S. Zhao, X. Hu, Y. Rong, J. Duan, J. Chen, Navigation Grade MEMS IMU for A Satellite, Micromachines (Basel). 12 (2021) 1–12. <https://doi.org/10.3390/MI12020151>.
- [11] Trimble, Antennas - GNSS Solutions for OEMs, (2022). <https://oemgnss.trimble.com/products-solutions/antennas/> (accessed April 29, 2022).
- [12] Waypoint Software Overview, (2022). https://docs.novatel.com/Waypoint/Content/Overview/Waypoint_Software.htm (accessed April 29, 2022).
- [13] Y.C. Lin, R. Manish, D. Bullock, A. Habib, Comparative analysis of different mobile LiDAR mapping systems for ditch line characterization, Remote Sensing. 13 (2021). <https://doi.org/10.3390/rs13132485>.
- [14] M.A. Grubb, K.E. Wilson, C.D. White, W.N. Nickas, Load and Resistance Factor Design (LRFD) for Highway Bridge Superstructures - Reference Manual, 2015. <https://doi.org/10.5194/isprs-archives-XLII-2-W5-649-2017>.
- [15] AASHTO, The manual for bridge evaluation, American Association of State Highway and Transportation Officials, Washington, DC, 2011.
- [16] AASHTO, AASHTO LRFD Bridge Design Specifications, 9th ed., American Association of State Highway and Transportation Officials, Washington, DC, 2020.
- [17] Bridge Deck Construction Manual, The State of California, Department of Transportation, Division of Engineering Services, 2015.

- [18] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, V. Rodehorst, Framework for automated UAS-based structural condition assessment of bridges, *Automation in Construction*. 97 (2019) 77–95.
<https://doi.org/10.1016/j.autcon.2018.10.006>.
- [19] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring, *Engineering*. 5 (2019) 199–222.
<https://doi.org/10.1016/j.eng.2018.11.030>.
- [20] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Automation in Construction*. 94 (2018) 112–126.
<https://doi.org/10.1016/j.autcon.2018.06.006>.
- [21] S. Chen, D.F. Laefer, E. Mangina, S.M.I. Zolanvari, J. Byrne, UAV Bridge Inspection through Evaluated 3D Reconstructions, *Journal of Bridge Engineering*. 24 (2019) 05019001. [https://doi.org/10.1061/\(asce\)be.1943-5592.0001343](https://doi.org/10.1061/(asce)be.1943-5592.0001343).
- [22] A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell’Andrea, M. Riley, Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling, *Structure and Infrastructure Engineering*. 14 (2018) 124–136.
<https://doi.org/10.1080/15732479.2017.1330891>.
- [23] C.J.O. Salaan, Y. Okada, S. Mizutani, T. Ishii, K. Koura, K. Ohno, S. Tadokoro, Close visual bridge inspection using a UAV with a passive rotating spherical shell, *Journal of Field Robotics*. 35 (2018) 850–867. <https://doi.org/10.1002/rob.21781>.
- [24] P.J. Sanchez-Cuevas, G. Heredia, A. Ollero, Multirotor UAS for bridge inspection by contact using the ceiling effect, *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017.* (2017) 767–774. <https://doi.org/10.1109/ICUAS.2017.7991412>.
- [25] P.J. Sanchez-Cuevas, P. Ramon-Soria, B. Arrue, A. Ollero, G. Heredia, Robotic system for inspection by contact of bridge beams using UAVs, *Sensors (Switzerland)*. 19 (2019). <https://doi.org/10.3390/s19020305>.
- [26] C. Zhang, D. Ardit, Advanced progress control of infrastructure construction projects using terrestrial laser scanning technology, *Infrastructures (Basel)*. 5 (2020) 1–18.
<https://doi.org/10.3390/infrastructures5100083>.
- [27] H. Son, C. Kim, Y. Kwon Cho, Automated Schedule Updates Using As-Built Data and a 4D Building Information Model, *Journal of Management in Engineering*. 33 (2017) 04017012. [https://doi.org/10.1061/\(asce\)me.1943-5479.0000528](https://doi.org/10.1061/(asce)me.1943-5479.0000528).
- [28] Z. Pučko, N. Šuman, D. Rebolj, Automated continuous construction progress monitoring using multiple workplace real time 3D scans, *Advanced Engineering Informatics*. 38 (2018) 27–40. <https://doi.org/10.1016/j.aei.2018.06.001>.
- [29] N. Ham, S.H. Lee, Empirical study on structural safety diagnosis of large-scale civil infrastructure using laser scanning and BIM, *Sustainability (Switzerland)*. 10 (2018). <https://doi.org/10.3390-su10114024>.
- [30] J. Lee, K.C. Lee, S. Lee, Y.J. Lee, S.H. Sim, Long-term displacement measurement of bridges using a LiDAR system, *Structural Control and Health Monitoring*. 26 (2019) 1–15. <https://doi.org/10.1002/stc.2428>.
- [31] G. Cha, S. Park, T. Oh, A Terrestrial LiDAR-Based Detection of Shape Deformation for Maintenance of Bridge Structures, *Journal of Construction Engineering and Management*. 145 (2019) 04019075. [https://doi.org/10.1061/\(asce\)co.1943-7862.0001701](https://doi.org/10.1061/(asce)co.1943-7862.0001701).

- [32] Y.T. Cheng, A. Patel, C. Wen, D. Bullock, A. Habib, Intensity thresholding and deep learning based lane marking extraction and lanewidth estimation from mobile light detection and ranging (LiDAR) point clouds, *Remote Sensing.* 12 (2020). <https://doi.org/10.3390/RS12091379>.
- [33] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, A. Habib, A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing.* 147 (2019) 178–192. <https://doi.org/10.1016/j.isprsjprs.2018.10.007>.
- [34] S. Lee, J. Kim, J.S. Yoon, S. Shin, O. Bailo, N. Kim, T.H. Lee, H.S. Hong, S.H. Han, I.S. Kweon, VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition, *Proceedings of the IEEE International Conference on Computer Vision.* 2017-Octob (2017) 1965–1973. <https://doi.org/10.1109/ICCV.2017.215>.
- [35] Z. Luo, J. Li, Z. Xiao, Z.G. Mou, X. Cai, C. Wang, Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments, *ISPRS Journal of Photogrammetry and Remote Sensing.* 150 (2019) 44–58. <https://doi.org/10.1016/j.isprsjprs.2019.01.024>.
- [36] M. Arastounia, Automated recognition of railroad infrastructure in rural areas from LIDAR data, *Remote Sensing.* 7 (2015) 14916–14938. <https://doi.org/10.3390/rs71114916>.
- [37] I. Puente, B. Akinci, H. González-Jorge, L. Díaz-Vilariño, P. Arias, A semi-automated method for extracting vertical clearance and cross sections in tunnels using mobile LiDAR data, *Tunnelling and Underground Space Technology.* 59 (2016) 48–54. <https://doi.org/10.1016/j.tust.2016.06.010>.
- [38] A. Sánchez-Rodríguez, B. Riveiro, M. Soilán, L.M. González-deSantos, Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets, *Automation in Construction.* 96 (2018) 171–179. <https://doi.org/10.1016/j.autcon.2018.09.014>.
- [39] A. Karunathilake, R. Honma, Y. Niina, Self-organized model fitting method for railway structures monitoring using lidar point cloud, *Remote Sensing.* 12 (2020) 1–15. <https://doi.org/10.3390/rs12223702>.
- [40] N. Puri, Y. Turkan, Bridge construction progress monitoring using LiDAR and 4D design models, *Automation in Construction.* 109 (2020) 102961. <https://doi.org/10.1016/j.autcon.2019.102961>.
- [41] Y.C. Lin, Y.T. Cheng, Y.J. Lin, J.E. Flatt, A. Habib, D. Bullock, Evaluating the Accuracy of Mobile LiDAR for Mapping Airfield Infrastructure, *Transportation Research Record.* 2673 (2019) 117–124. <https://doi.org/10.1177/0361198119835802>.
- [42] I. Puente, H. González-Jorge, B. Riveiro, P. Arias, Accuracy verification of the Lynx Mobile Mapper system, *Optics and Laser Technology.* 45 (2013) 578–586. <https://doi.org/10.1016/j.optlastec.2012.05.029>.
- [43] P.J. Besl, N.D. McKay, Method for registration of 3-D shapes, *Sensor Fusion IV: Control Paradigms and Data Structures.* 1611 (1992) 586–606. <https://doi.org/10.1117/12.57955>.
- [44] C. Yang, G. Medioni, Object modelling by registration of multiple range images, *Image and Vision Computing.* 10 (1992) 145–155. [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C).

- [45] G.C. Sharp, S.W. Lee, D.K. Wehe, ICP registration using invariant features, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24 (2002) 90–102. <https://doi.org/10.1109/34.982886>.
- [46] A. Gressin, C. Mallet, J.-Ô. Demantké, N. David, Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge, *ISPRS Journal of Photogrammetry and Remote Sensing*. 79 (2013) 240–251. <https://doi.org/10.1016/j.isprsjprs.2013.02.019>.
- [47] M. Al-Durgham, I. Detchev, A. Habib, Analysis of Two Triangle-Based Multi-Surface Registration Algorithms of Irregular Point Clouds, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XXXVIII-5/ (2012) 61–66. <https://doi.org/10.5194/isprarchives-xxxviii-5-w12-61-2011>.
- [48] D. Grant, J. Bethel, M. Crawford, Point-to-plane registration of terrestrial laser scans, *ISPRS Journal of Photogrammetry and Remote Sensing*. 72 (2012) 16–26. <https://doi.org/10.1016/j.isprsjprs.2012.05.007>.
- [49] D. Lague, N. Brodu, J. Leroux, Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z), *ISPRS Journal of Photogrammetry and Remote Sensing*. 82 (2013) 10–26. <https://doi.org/10.1016/j.isprsjprs.2013.04.009>.
- [50] A. Habib, I. Detchev, K. Bang, A comparative analysis of two approaches for multiple-surface registration of irregular point clouds, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 38 (2010) 1–6.
- [51] A. Gruen, D. Akca, Least squares 3D surface and curve matching, *ISPRS Journal of Photogrammetry and Remote Sensing*. 59 (2005) 151–174. <https://doi.org/10.1016/j.isprsjprs.2005.02.006>.
- [52] D. Akca, Full automatic registration of laser scanner point clouds, *BRISK Binary Robust Invariant Scalable Keypoints*. (2009) 12–19.
- [53] M. Franaszek, G.S. Cheok, C. Witzgall, Fast automatic registration of range images from 3D imaging systems using sphere targets, *Automation in Construction*. 18 (2009) 265–274. <https://doi.org/10.1016/j.autcon.2008.08.003>.
- [54] F. Bosché, Plane-based registration of construction laser scans with 3D/4D building models, *Advanced Engineering Informatics*. 26 (2012) 90–102. <https://doi.org/10.1016/j.aei.2011.08.009>.
- [55] W.I. Liu, Novel method for sphere target detection and center estimation from mobile terrestrial laser scanner data, *Measurement: Journal of the International Measurement Confederation*. 137 (2019) 617–623. <https://doi.org/10.1016/j.measurement.2019.02.025>.
- [56] J.Y. Han, A noniterative approach for the quick Alignment of multistation unregistered LiDAR point clouds, *IEEE Geoscience and Remote Sensing Letters*. 7 (2010) 727–730. <https://doi.org/10.1109/LGRS.2010.2046876>.
- [57] P. Kim, J. Chen, Y.K. Cho, Automated Point Cloud Registration Using Visual and Planar Features for Construction Environments, *Journal of Computing in Civil Engineering*. 32 (2018) 04017076. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000720](https://doi.org/10.1061/(asce)cp.1943-5487.0000720).
- [58] J.Y. Han, J.J. Jaw, Solving a similarity transformation between two reference frames using hybrid geometric control features, *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A*. 36 (2013) 304–313. <https://doi.org/10.1080/02533839.2012.730262>.

- [59] T. Huang, Registration method for terrestrial LiDAR point clouds using geometric features, *Optical Engineering*. 51 (2012) 021114. <https://doi.org/10.1117/1.oe.51.2.021114>.
- [60] I. Stamos, M. Leordeanu, Automated Feature-Based Range Registration of Urban Scenes of Large Scale, in: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003: pp. 0–6.
- [61] C. Dold, C. Brenner, Registration of terrestrial laser scanning data using planar patches and image data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 36 (2006) 78–83. <https://doi.org/10.15488/3750>.
- [62] K. Al-Durgham, A. Habib, Association-matrix-based sample consensus approach for automated registration of terrestrial laser scans using linear features, *Photogrammetric Engineering and Remote Sensing*. 80 (2014) 1029–1039. <https://doi.org/10.14358/PERS.80.11.1029>.
- [63] K. Al-Durgham, A. Habib, M. Mazaheri, Solution frequency-based procedure for automated registration of terrestrial laser scans using linear features, *ASPRS 2014 Annual Conference: Geospatial Power in Our Pockets, Co-Located with Joint Agency Commercial Imagery Evaluation Workshop, JACIE 2014*. (2014).
- [64] H. Fangning, H. Ayman, A Closed-Form Solution for Coarse Registration of Point Clouds Using Linear Features, *Journal of Surveying Engineering*. 142 (2016) 04016006. [https://doi.org/10.1061/\(asce\)su.1943-5428.0000174](https://doi.org/10.1061/(asce)su.1943-5428.0000174).
- [65] K. Al-Durgham, A. Habib, E. Kwak, RANSAC approach for automated registration of terrestrial laser scans using linear features, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2 (2013) 13–18. <https://doi.org/10.5194/isprsaannals-II-5-W2-13-2013>.
- [66] T.Y. Chuang, J.J. Jaw, Multi-feature registration of point clouds, *Remote Sensing*. 9 (2017). <https://doi.org/10.3390/rs9030281>.
- [67] H. Wu, M. Scaioni, H. Li, N. Li, M. Lu, C. Liu, Feature-constrained registration of building point clouds acquired by terrestrial and airborne laser scanners, *Journal of Applied Remote Sensing*. 8 (2014) 083587. <https://doi.org/10.1117/1.jrs.8.083587>.
- [68] J. Li, B. Yang, C. Chen, A. Habib, NRLI-UAV: Non-rigid registration of sequential raw laser scans and images for low-cost UAV LiDAR point cloud quality improvement, *ISPRS Journal of Photogrammetry and Remote Sensing*. 158 (2019) 123–145. <https://doi.org/10.1016/j.isprsjprs.2019.10.009>.
- [69] I. Stamos, P.K. Allen, Geometry and texture recovery of scenes of large scale, *Computer Vision and Image Understanding*. 88 (2002) 94–118. <https://doi.org/10.1006/cviu.2002.0963>.
- [70] A. Al-Rawabdeh, H. Al-Gurrani, K. Al-Durgham, I. Detchev, F. He, N. El-Sheimy, A. Habib, A robust registration algorithm for point clouds from UAV images for change detection, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 2016-Janua (2016) 765–772. <https://doi.org/10.5194/isprsaarchives-XLI-B1-765-2016>.
- [71] W. Von Hansen, H. Gross, U. Thoennesen, Line-Based Registration of Terrestrial and Airborne Lidar Data, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XXXVII (2008) 161–166.

- [72] A.F. Habib, M. Ghanma, M. Morgan, R. Al-ruzouq, Photogrammetric and Lidar Data Registration Using Linear Features, *Photogrammetric Engineering Remote Sensing*. 71 (2005) 699–707. <https://doi.org/10.14358/PERS.71.6.699>.
- [73] E. Renaudin, A. Habib, A.P. Kersting, Featured-based registration of terrestrial laser scans with minimum overlap using photogrammetric data, *ETRI Journal*. 33 (2011) 517–527. <https://doi.org/10.4218/etrij.11.1610.0006>.
- [74] J. Yang, Z. Cao, Q. Zhang, A fast and robust local descriptor for 3D point cloud registration, *Information Sciences*. 346–347 (2016) 163–179. <https://doi.org/10.1016/j.ins.2016.01.095>.
- [75] M. Bueno, H. González-Jorge, J. Martínez-Sánchez, H. Lorenzo, Automatic point cloud coarse registration using geometric keypoint descriptors for indoor scenes, *Automation in Construction*. 81 (2017) 134–148. <https://doi.org/10.1016/j.autcon.2017.06.016>.
- [76] S. Barnea, S. Filin, Keypoint based autonomous registration of terrestrial laser point-clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*. 63 (2008) 19–35. <https://doi.org/10.1016/j.isprsjprs.2007.05.005>.
- [77] C. Choy, J. Park, V. Koltun, Fully Convolutional Geometric Features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.
- [78] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, C.L. Tai, D3Feat: Joint learning of dense detection and description of 3D local features, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020: pp. 6358–6366. <https://doi.org/10.1109/CVPR42600.2020.00639>.
- [79] C. Choy, W. Dong, V. Koltun, Deep global registration, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020: pp. 2511–2520. <https://doi.org/10.1109/CVPR42600.2020.00259>.
- [80] K. Fu, S. Liu, X. Luo, M. Wang, Robust Point Cloud Registration Framework Based on Deep Graph Matching, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021: pp. 8893–8902.
- [81] S. Ao, Q. Hu, B. Yang, A. Markham, Y. Guo, SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021: pp. 11753–11762.
- [82] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, K. Schindler, PREDATOR: Registration of 3D Point Clouds with Low Overlap, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021: pp. 4267–4276.
- [83] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, C.-L. Tai, PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021: pp. 15859–15869.
- [84] A. Wendt, A concept for feature based data registration by simultaneous consideration of laser scanner data and photogrammetric images, *ISPRS Journal of Photogrammetry and Remote Sensing*. 62 (2007) 122–134. <https://doi.org/10.1016/j.isprsjprs.2006.12.001>.
- [85] Velodyne HDL32E Datasheet, (n.d.). <https://velodynelidar.com/products/hdl-32e/> (accessed May 26, 2021).
- [86] Velodyne Puck Hi-Res Datasheet, (n.d.). <https://velodynelidar.com/products/puck-hi-res/> (accessed May 26, 2021).

- [87] Applanix POSLV 220 Datasheet, (n.d.). online: <https://www.applanix.com/products/poslv.htm> (accessed April 26, 2020).
- [88] Riegl VUX-1HA, (n.d.). <http://www.riegl.com/products/newriegl-vux-1-series/newriegl-vux-1ha> (accessed April 26, 2020).
- [89] Z+F Profiler 9012, (n.d.). https://www.zf-laser.com/Z-F-PROFILER-R-9012.2d_laserscanner.0.html (accessed April 26, 2020).
- [90] Novatel IMU-ISA-100C, (n.d.).
https://docs.novatel.com/OEM7/Content/Technical_Specs_IMU/ISA_100C_Overview.htm (accessed April 26, 2020).
- [91] R. Ravi, Y.J. Lin, M. Elbahnaawy, T. Shamseldin, A. Habib, Simultaneous System Calibration of a Multi-LiDAR Multicamera Mobile Mapping Platform, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 11 (2018) 1694–1714. <https://doi.org/10.1109/JSTARS.2018.2812796>.
- [92] A. Habib, J. Lay, C. Wong, LIDAR Error Propagation Calculator, (2006). <https://engineering.purdue.edu/CE/Academics/Groups/Geomatics/DPRG/files/LIDARErrorPropagation.zip> (accessed October 10, 2021).
- [93] FARO Focus 3D X330, (n.d.).
https://www.connectcec.com/sites/cec/uploads/TechSheet_Focus3D_X_330.pdf (accessed February 7, 2021).
- [94] Trimble TX8, (n.d.). <https://fieldtech.trimble.com/resources/product-guides-brochures-data-sheets/datasheet-trimble-tx8-3d-laser-scanner> (accessed February 7, 2021).
- [95] A. Habib, Y.J. Lin, Multi-class simultaneous adaptive segmentation and quality control of point cloud data, *Remote Sensing*. 8 (2016) 1–23. <https://doi.org/10.3390/rs8020104>.
- [96] J. Demantké, C. Mallet, N. David, B. Vallet, Dimensionality based scale selection in 3D LiDAR point clouds, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XXXVIII-5/ (2012) 97–102.
<https://doi.org/10.5194/isprsarchives-xxxviii-5-w12-97-2011>.
- [97] E.M. Mikhail, F.E. Ackermann, *Observations and Least Squares*, Univ Pr of Amer, 1983.
- [98] R. Ravi, A. Habib, Least Squares Adjustment with a Rank-deficient Weight Matrix and its Applicability towards Image/LiDAR Data Processing., *Photogrammetric Engineering & Remote Sensing*. 87 (2021) 717–733.
- [99] R. Ravi, Y.-J. Lin, M. Elbahnaawy, T. Shamseldin, A. Habib, Bias impact analysis and calibration of terrestrial mobile LiDAR system with several spinning multibeam laser scanners, *IEEE Transactions on Geoscience and Remote Sensing*. 56 (2018) 5261–5275.
<https://doi.org/10.1109/TGRS.2018.2812782>.
- [100] R. Ravi, A. Habib, Fully Automated Profile-based Calibration Strategy for Airborne and Terrestrial Mobile LiDAR Systems with Spinning Multi-beam Laser Units, *Remote Sensing*. 12 (2020) 401. <https://doi.org/10.3390/rs12030401>.
- [101] N. Muhammad, S. Lacroix, Calibration of a rotating multi-beam LiDAR, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010: pp. 5648–5653. <https://doi.org/10.1109/IROS.2010.5651382>.
- [102] M. He, H. Zhao, F. Davoine, J. Cui, H. Zha, Pairwise LiDAR calibration using multi-type 3D geometric features in natural scene, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013: pp. 1828–1835.

- [103] S. Chen, J. Liu, T. Wu, W. Huang, K. Liu, D. Yin, X. Liang, J. Hyppä, R. Chen, Extrinsic calibration of 2d laser rangefinders based on a mobile sphere, *Remote Sensing*. 10 (2018) 1176. <https://doi.org/10.3390/rs10081176>.
- [104] M. He, H. Zhao, J. Cui, H. Zha, Calibration method for multiple 2d lidars system, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014: pp. 3034–3041. <https://doi.org/10.1109/ICRA.2014.6907296>.
- [105] T.O. Chan, D.D. Lichten, C.L. Glennie, Multi-feature based boresight self-calibration of a terrestrial mobile mapping system, *ISPRS Journal of Photogrammetry and Remote Sensing*. 82 (2013) 112–124. <https://doi.org/10.1016/j.isprsjprs.2013.04.005>.
- [106] E. Heinz, C. Holst, H. Kuhlmann, L. Klingbeil, Design and evaluation of a permanently installed plane-based calibration field for mobile laser scanning systems, *Remote Sensing*. 12 (2020) 555. <https://doi.org/10.3390/rs12030555>.
- [107] R. Ravi, Y.-J. Lin, M. Elbahna, T. Shamseldin, A. Habib, Simultaneous System Calibration of a Multi-LiDAR Multicamera Mobile Mapping Platform, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 11 (2018) 1694–1714. <https://doi.org/10.1109/JSTARS.2018.2812796>.
- [108] A.S. Aguiar, F.N. dos Santos, J.B. Cunha, H. Sobreira, A.J. Sousa, Localization and mapping for robots in agriculture and forestry: A survey, *Robotics*. 9 (2020) 97. <https://doi.org/10.3390/robotics9040097>.
- [109] B. Alsadik, S. Karam, The simultaneous localization and mapping (SLAM)-An overview, *Surveying and Geospatial Engineering Journal*. 2 (2021) 1–12. <https://doi.org/10.38094/jastt204117>.
- [110] G. Bresson, Z. Alsayed, L. Yu, S. Glaser, Simultaneous localization and mapping: A survey of current trends in autonomous driving, *IEEE Transactions on Intelligent Vehicles*. 2 (2017) 194–220. <https://doi.org/10.1109/TIV.2017.2749181>.
- [111] R. Manish, Y.-C. Lin, R. Ravi, S.M. Hasheminasab, T. Zhou, A. Habib, Development of a miniaturized mobile mapping system for in-row, under-canopy phenotyping, *Remote Sensing*. 13 (2021). <https://doi.org/10.3390/rs13020276>.
- [112] D. Nistér, O. Naroditsky, J. Bergen, Visual odometry, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004: pp. I–I. <https://doi.org/10.1109/CVPR.2004.1315094>.
- [113] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 29 (2007) 1052–1067. <https://doi.org/10.1109/TPAMI.2007.1049>.
- [114] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007: pp. 225–234. <https://doi.org/10.1109/ISMAR.2007.4538852>.
- [115] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, *IEEE Transactions on Robotics*. 31 (2015) 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>.
- [116] S. Weiss, M.W. Achtelik, S. Lynen, M. Chli, R. Siegwart, Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments, in: 2012 IEEE International Conference on Robotics and Automation, 2012: pp. 957–964. <https://doi.org/10.1109/ICRA.2012.6225147>.

- [117] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, Pixhawk: A system for autonomous flight using onboard computer vision, in: 2011 IEEE International Conference on Robotics and Automation, 2011: pp. 2992–2997. <https://doi.org/10.1109/ICRA.2011.5980229>.
- [118] M. Li, A.I. Mourikis, High-precision, consistent EKF-based visual-inertial odometry, *The International Journal of Robotics Research.* 32 (2013) 690–711. <https://doi.org/10.1177/0278364913481251>.
- [119] J. Engel, J. Sturm, D. Cremers, Camera-based navigation of a low-cost quadrocopter, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012: pp. 2815–2821. <https://doi.org/10.1109/IROS.2012.6385458>.
- [120] K. Eckenhoff, Y. Yang, P. Geneva, G. Huang, Tightly-coupled visual-inertial localization and 3-d rigid-body target tracking, *IEEE Robotics and Automation Letters.* 4 (2019) 1541–1548. <https://doi.org/10.1109/LRA.2019.2896472>.
- [121] Y. He, J. Zhao, Y. Guo, W. He, K. Yuan, PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features, *Sensors.* 18 (2018) 1159. <https://doi.org/10.3390/s18041159>.
- [122] J. Jiang, J. Yuan, X. Zhang, X. Zhang, DVIO: An optimization-based tightly coupled direct visual-inertial odometry, *IEEE Transactions on Industrial Electronics.* 68 (2020) 11212–11222. <https://doi.org/10.1109/TIE.2020.3036243>.
- [123] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, M. Chli, GOMSF: Graph-optimization based multi-sensor fusion for robust uav pose estimation, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: pp. 1421–1428. <https://doi.org/10.1109/ICRA.2018.8460193>.
- [124] T. Chu, N. Guo, S. Backén, D. Akos, Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments, *Sensors.* 12 (2012) 3162–3185. <https://doi.org/10.3390/s120303162>.
- [125] K.-W. Chiang, D.T. Le, T.T. Duong, R. Sun, The performance analysis of INS/GNSS/V-SLAM integration scheme using smartphone sensors for land vehicle navigation applications in GNSS-challenging environments, *Remote Sensing.* 12 (2020) 1732. <https://doi.org/10.3390/rs12111732>.
- [126] G. Cioffi, D. Scaramuzza, Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: pp. 5089–5095. <https://doi.org/10.1109/IROS45743.2020.9341697>.
- [127] J. Liu, W. Gao, Z. Hu, Optimization-based visual-inertial SLAM tightly coupled with raw GNSS measurements, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021: pp. 11612–11618. <https://doi.org/10.1109/ICRA48506.2021.9562013>.
- [128] S. Cao, X. Lu, S. Shen, GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation, *IEEE Transactions on Robotics.* (2022). <https://doi.org/10.1109/TRO.2021.3133730>.
- [129] J. Zhang, S. Singh, LOAM: LiDAR Odometry and Mapping in Real-time., in: *Robotics: Science and Systems*, 2014: pp. 1–9. <https://doi.org/10.15607/RSS.2014.X.007>.
- [130] J. Zhang, S. Singh, Visual-LiDAR odometry and mapping: Low-drift, robust, and fast, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015: pp. 2174–2181. <https://doi.org/10.1109/ICRA.2015.7139486>.

- [131] T. Shan, B. Englot, LeGO–LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: pp. 4758–4765.
<https://doi.org/10.1109/IROS.2018.8594299>.
- [132] S.W. Chen, G. v Nardari, E.S. Lee, C. Qu, X. Liu, R.A.F. Romero, V. Kumar, SLOAM: Semantic LiDAR odometry and mapping for forest inventory, *IEEE Robotics and Automation Letters.* 5 (2020) 612–619. <https://doi.org/10.1109/LRA.2019.2963823>.
- [133] S. Du, Y. Li, X. Li, M. Wu, LiDAR Odometry and Mapping Based on Semantic Information for Outdoor Environment, *Remote Sensing.* 13 (2021) 2864.
<https://doi.org/10.3390/rs13152864>.
- [134] X. Ji, L. Zuo, C. Zhang, Y. Liu, LLOAM: LiDAR odometry and mapping with loop-closure detection based correction, in: 2019 IEEE International Conference on Mechatronics and Automation (ICMA), 2019: pp. 2475–2480.
<https://doi.org/10.1109/ICMA.2019.8816388>.
- [135] S. Zhang, L. Xiao, Y. Nie, B. Dai, C. Hu, LiDAR Odometry and Mapping Based on Two-stage Feature Extraction, in: 2020 39th Chinese Control Conference (CCC), 2020: pp. 3966–3971. <https://doi.org/10.23919/CCC50068.2020.9188810>.
- [136] Y. Cong, C. Chen, J. Li, W. Wu, S. Li, B. Yang, Mapping without dynamic: Robust LiDAR-slam for ugv mobile mapping in dynamic environments, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.* 43 (2020) 515–520. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-515-2020>.
- [137] J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, J. Hyppä, LiDAR scan matching aided inertial navigation system in GNSS-denied environments, *Sensors.* 15 (2015) 16710–16728. <https://doi.org/10.3390/s150716710>.
- [138] C. Qian, H. Liu, J. Tang, Y. Chen, H. Kaartinen, A. Kukko, L. Zhu, X. Liang, L. Chen, J. Hyppä, An integrated GNSS/INS/LiDAR-SLAM positioning method for highly accurate forest stem mapping, *Remote Sensing.* 9 (2017). <https://doi.org/10.3390/rs9010003>.
- [139] L. Chang, X. Niu, T. Liu, J. Tang, C. Qian, GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization, *Remote Sensing.* 11 (2019) 1009.
<https://doi.org/10.3390/rs11091009>.
- [140] K.-W. Chiang, G.-J. Tsai, H.W. Chang, C. Joly, N. Ei-Sheimy, Seamless navigation and mapping using an INS/GNSS/grid-based SLAM semi-tightly coupled integration scheme, *Information Fusion.* 50 (2019) 181–196. <https://doi.org/10.1016/j.inffus.2019.01.004>.
- [141] A. Aboutaleb, A.S. El-Wakeel, H. Elghamrawy, A. Noureldin, LiDAR/RISS/GNSS dynamic integration for land vehicle robust positioning in challenging GNSS environments, *Remote Sensing.* 12 (2020) 2323. <https://doi.org/10.3390/rs12142323>.
- [142] A. Kukko, R. Kaijaluoto, H. Kaartinen, V. v Lehtola, A. Jaakkola, J. Hyppä, Graph SLAM correction for single scanner MLS forest data under boreal forest canopy, *ISPRS Journal of Photogrammetry and Remote Sensing.* 132 (2017) 199–209.
<https://doi.org/10.1016/j.isprsjprs.2017.09.006>.
- [143] N. Li, L. Guan, Y. Gao, S. Du, M. Wu, X. Guang, X. Cong, Indoor and outdoor low-cost seamless integrated navigation system based on the integration of INS/GNSS/LiDAR system, *Remote Sensing.* 12 (2020) 3271. <https://doi.org/10.3390/rs12193271>.
- [144] A. Habib, Y.-J. Lin, Multi-class simultaneous adaptive segmentation and quality control of point cloud data, *Remote Sensing.* 8 (2016) 104. <https://doi.org/10.3390/rs8020104>.

- [145] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun ACM.* 24 (1981) 381–395. <https://doi.org/10.1145/358669.358692>.
- [146] Velodyne, Puck Hi-Res Datasheet, (n.d.). <https://velodynelidar.com/vlp-16-hi-res.html>.
- [147] Novatel, SPAN-CPT, (n.d.). <https://novatel.com/support/previous-generation-products-drop-down/previous-generation-products/span-cpt>.
- [148] Novatel, Inertial Explorer, (n.d.). <https://novatel.com/products/waypoint-post-processing-software/inertial-explorer>.
- [149] A. Habib, J. Lay, C. Wong, Specifications for the Quality Assurance and Quality Control of LiDAR Systems, *Base Mapping and Geomatic Services of British Columbia.* (2006).
- [150] AASHTO, The manual for bridge evaluation, American Association of State Highway and Transportation Officials, Washington, DC, 2011.
- [151] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, V. Rodehorst, Framework for automated UAS-based structural condition assessment of bridges, *Automation in Construction.* 97 (2019) 77–95. <https://doi.org/10.1016/j.autcon.2018.10.006>.
- [152] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring, *Engineering.* 5 (2019) 199–222. <https://doi.org/10.1016/j.eng.2018.11.030>.
- [153] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Automation in Construction.* 94 (2018) 112–126. <https://doi.org/10.1016/j.autcon.2018.06.006>.
- [154] S. Chen, D.F. Laefer, E. Mangina, S.M.I. Zolanvari, J. Byrne, UAV Bridge Inspection through Evaluated 3D Reconstructions, *Journal of Bridge Engineering.* 24 (2019) 05019001. [https://doi.org/10.1061/\(asce\)be.1943-5592.0001343](https://doi.org/10.1061/(asce)be.1943-5592.0001343).
- [155] A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell’Andrea, M. Riley, Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling, *Structure and Infrastructure Engineering.* 14 (2018) 124–136. <https://doi.org/10.1080/15732479.2017.1330891>.
- [156] C. Zhang, D. Arditì, Advanced progress control of infrastructure construction projects using terrestrial laser scanning technology, *Infrastructures (Basel).* 5 (2020) 1–18. <https://doi.org/10.3390/infrastructures5100083>.
- [157] H. Son, C. Kim, Y. Kwon Cho, Automated Schedule Updates Using As-Built Data and a 4D Building Information Model, *Journal of Management in Engineering.* 33 (2017) 04017012. [https://doi.org/10.1061/\(asce\)me.1943-5479.0000528](https://doi.org/10.1061/(asce)me.1943-5479.0000528).
- [158] Z. Pučko, N. Šuman, D. Rebolj, Automated continuous construction progress monitoring using multiple workplace real time 3D scans, *Advanced Engineering Informatics.* 38 (2018) 27–40. <https://doi.org/10.1016/j.aei.2018.06.001>.
- [159] J. Lee, K.C. Lee, S. Lee, Y.J. Lee, S.H. Sim, Long-term displacement measurement of bridges using a LiDAR system, *Structural Control and Health Monitoring.* 26 (2019) 1–15. <https://doi.org/10.1002/stc.2428>.
- [160] G. Cha, S. Park, T. Oh, A Terrestrial LiDAR-Based Detection of Shape Deformation for Maintenance of Bridge Structures, *Journal of Construction Engineering and Management.* 145 (2019) 04019075. [https://doi.org/10.1061/\(asce\)co.1943-7862.0001701](https://doi.org/10.1061/(asce)co.1943-7862.0001701).

- [161] N. Ham, S.H. Lee, Empirical study on structural safety diagnosis of large-scale civil infrastructure using laser scanning and BIM, *Sustainability* (Switzerland). 10 (2018). <https://doi.org/10.3390/su10114024>.
- [162] C. Wang, C. Wen, Y. Dai, S. Yu, M. Liu, Urban 3D modeling with mobile laser scanning: a review, *Virtual Reality and Intelligent Hardware*. 2 (2020) 175–212. <https://doi.org/10.1016/j.vrih.2020.05.003>.
- [163] E. Che, J. Jung, M.J. Olsen, Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review, *Sensors* (Switzerland). 19 (2019). <https://doi.org/10.3390/s19040810>.
- [164] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile laser scanned point-clouds for road object detection and extraction: A review, *Remote Sensing*. 10 (2018) 1–33. <https://doi.org/10.3390/rs10101531>.
- [165] Y. Li, B. Wu, X. Ge, Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density, *ISPRS Journal of Photogrammetry and Remote Sensing*. 153 (2019) 151–165. <https://doi.org/10.1016/j.isprsjprs.2019.05.007>.
- [166] H. He, K. Khoshelham, C. Fraser, A multiclass TrAdaBoost transfer learning algorithm for the classification of mobile lidar data, *ISPRS Journal of Photogrammetry and Remote Sensing*. 166 (2020) 118–127. <https://doi.org/10.1016/j.isprsjprs.2020.05.010>.
- [167] G. Forlani, C. Nardinocchi, M. Scaioni, P. Zingaretti, Complete classification of raw LIDAR data and 3D reconstruction of buildings, *Pattern Analysis and Applications*. 8 (2006) 357–374. <https://doi.org/10.1007/s10044-005-0018-2>.
- [168] G. Vosselman, B.G.H. Gorte, G. Sithole, T. Rabbani, Recognising structure in laser scanner point clouds, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 46 (2004) 33–38.
- [169] B. Yang, Z. Dong, A shape-based segmentation method for mobile laser scanning point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*. 81 (2013) 19–30. <https://doi.org/10.1016/j.isprsjprs.2013.04.002>.
- [170] Z. Lari, A. Habib, An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data, *ISPRS Journal of Photogrammetry and Remote Sensing*. 93 (2014) 192–212. <https://doi.org/10.1016/j.isprsjprs.2013.12.001>.
- [171] Z. Wang, L. Zhang, T. Fang, P.T. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li, D. Chen, A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification, *IEEE Transactions on Geoscience and Remote Sensing*. 53 (2015) 2409–2425. <https://doi.org/10.1109/TGRS.2014.2359951>.
- [172] J. Zhang, X. Lin, X. Ning, SVM-Based classification of segmented airborne LiDAR point clouds in urban areas, *Remote Sensing*. 5 (2013) 3749–3775. <https://doi.org/10.3390/rs5083749>.
- [173] K. Khoshelham, S.O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, *Sensors*. 12 (2012) 1437–1454. <https://doi.org/10.3390/s120201437>.
- [174] N. Chehata, L. Guo, C. Mallet, Airborne lidar feature selection for urban classification using random forests, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. XXXVIII-3/ (2009) 207–212.
- [175] M. Weinmann, B. Jutzi, S. Hinz, C. Mallet, Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers, *ISPRS Journal of*

- Photogrammetry and Remote Sensing. 105 (2015) 286–304.
<https://doi.org/10.1016/j.isprsjprs.2015.01.016>.
- [176] A. Schmidt, F. Rottensteiner, U. Sörgel, Classification of airborne laser scanning data in Wadden sea areas using conditional random fields, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XXXIX-B3 (2012) 161–166. <https://doi.org/10.5194/isprarchives-xxxix-b3-161-2012>.
- [177] G. Vosselman, M. Coenen, F. Rottensteiner, Contextual segment-based classification of airborne laser scanner data, *ISPRS Journal of Photogrammetry and Remote Sensing*. 128 (2017) 354–371. <https://doi.org/10.1016/j.isprsjprs.2017.03.010>.
- [178] J. Niemeyer, F. Rottensteiner, U. Soergel, Contextual classification of lidar data and building object detection in urban areas, *ISPRS Journal of Photogrammetry and Remote Sensing*. 87 (2014) 152–165. <https://doi.org/10.1016/j.isprsjprs.2013.11.001>.
- [179] S. Guinard, L. Landrieu, Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 42 (2017) 151–157. <https://doi.org/10.5194/isprarchives-XLII-1-W1-151-2017>.
- [180] Q. Zhu, Y. Li, H. Hu, B. Wu, Robust point cloud classification based on multi-level semantic relationships for urban scenes, *ISPRS Journal of Photogrammetry and Remote Sensing*. 129 (2017) 86–102. <https://doi.org/10.1016/j.isprsjprs.2017.04.022>.
- [181] D. Wu, Z. Liang, G. Chen, Deep learning for LiDAR-only and LiDAR-fusion 3D perception : a survey, *Intelligence & Robotics*. 2 (2022) 105–129. <https://doi.org/10.20517/ir.2021.20>.
- [182] R. Zhang, G. Li, M. Li, L. Wang, Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning, *ISPRS Journal of Photogrammetry and Remote Sensing*. 143 (2018) 85–96. <https://doi.org/10.1016/j.isprsjprs.2018.04.022>.
- [183] A. Boulch, B. le Saux, N. Audebert, Unstructured point cloud semantic labeling using deep segmentation networks, *Eurographics Workshop on 3D Object Retrieval*, EG 3DOR. 2017-April (2017) 17–24. <https://doi.org/10.2312/3dor.20171047>.
- [184] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, S. Savarese, SEGCloud: Semantic segmentation of 3D point clouds, *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017.* (2018) 537–547. <https://doi.org/10.1109/3DV.2017.00067>.
- [185] Y. Zhou, O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, n.d.
- [186] C.R. Qi, H. Su, K. Mo, L. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *IEEE CVPR*. (2017). <https://doi.org/10.1109/3DV.2016.68>.
- [187] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Adv Neural Inf Process Syst*. (2017). [https://doi.org/10.1016/S0039-9140\(96\)02179-0](https://doi.org/10.1016/S0039-9140(96)02179-0).
- [188] Y. Li, R. Bu, X. Di, PointCNN : Convolution On X -Transformed Points, (2018).
- [189] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, *ACM Transactions on Graphics*. 38 (2019) Article 146.

- [190] L. Landrieu, S. Martin, Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: pp. 4558–4567.
- [191] T. Hackel, N. Savinov, L. Ladicky, J.D. Wegner, K. Schindler, M. Pollefeys, SEMANTIC3D.NET: A New Large-Scale Point Cloud Classification Benchmark, (2017).
- [192] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: pp. 1534–1543. <https://doi.org/10.1109/CVPR.2016.170>.
- [193] R. Lu, I. Brilakis, Generating bridge geometric digital twins from point clouds, Proceedings of the 2019 European Conference on Computing in Construction. 1 (2019) 367–376. <https://doi.org/10.35490/ec3.2019.182>.
- [194] R. Lu, I. Brilakis, C.R. Middleton, Detection of Structural Components in Point Clouds of Existing RC Bridges, Computer-Aided Civil and Infrastructure Engineering. 34 (2019) 191–212. <https://doi.org/10.1111/mice.12407>.
- [195] L. Truong-Hong, R. Lindenbergh, Extracting Bridge Components from a Laser Scanning Point Cloud, Lecture Notes in Civil Engineering. 98 (2021) 721–739. https://doi.org/10.1007/978-3-030-51295-8_50.
- [196] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Automation in Construction. 125 (2021) 103582. <https://doi.org/10.1016/j.autcon.2021.103582>.
- [197] H. Kim, J. Yoon, S.H. Sim, Automated bridge component recognition from point clouds using deep learning, Structural Control and Health Monitoring. 27 (2020) 1–13. <https://doi.org/10.1002/stc.2591>.
- [198] H. Kim, C. Kim, Deep-learning-based classification of point clouds for bridge inspection, Remote Sensing. 12 (2020) 1–13. <https://doi.org/10.3390/rs12223757>.
- [199] T. Xia, J. Yang, L. Chen, Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning, Automation in Construction. 133 (2022) 103992. <https://doi.org/10.1016/j.autcon.2021.103992>.
- [200] B. Gao, Y. Pan, C. Li, S. Geng, H. Zhao, Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods, IEEE Transactions on Intelligent Transportation Systems. (2021). <https://doi.org/10.1109/TITS.2021.3076844>.
- [201] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2018: pp. 270–279. https://doi.org/10.1007/978-3-030-01424-7_27.
- [202] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering. 22 (2010) 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>.
- [203] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2018: pp. 270–279. https://doi.org/10.1007/978-3-030-01424-7_27.
- [204] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering. 22 (2010) 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>.

- [205] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, Springer International Publishing, 2016. <https://doi.org/10.1186/s40537-016-0043-6>.
- [206] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A Comprehensive Survey on Transfer Learning, *Proceedings of the IEEE*. 109 (2021) 43–76. <https://doi.org/10.1109/JPROC.2020.3004555>.
- [207] M. Imad, O. Doukhi, D.J. Lee, Transfer learning based semantic segmentation for 3d object detection from point cloud, *Sensors*. 21 (2021) 1–15. <https://doi.org/10.3390/s21123964>.
- [208] C. Zhao, H. Guo, J. Lu, D. Yu, D. Li, X. Chen, ALS Point Cloud Classification with Small Training Data Set Based on Transfer Learning, *IEEE Geoscience and Remote Sensing Letters*. 17 (2020) 1406–1410. <https://doi.org/10.1109/LGRS.2019.2947608>.
- [209] J. Balado, R. Sousa, L. Díaz-Vilariño, P. Arias, Transfer Learning in urban object classification: Online images to recognize point clouds, *Automation in Construction*. 111 (2020) 103058. <https://doi.org/10.1016/j.autcon.2019.103058>.
- [210] F. Piewak, P. Pinggera, M. Schäfer, D. Peter, B. Schwarz, N. Schneider, M. Enzweiler, D. Pfeiffer, M. Zöllner, Boosting LIDAR-based semantic labeling by cross-modal training data generation, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 11134 LNCS (2019) 497–513. https://doi.org/10.1007/978-3-030-11024-6_39.
- [211] S. Zhao, Y. Wang, B. Li, B. Wu, Y. Gao, P. Xu, T. Darrell, K. Keutzer, ePointDA: An End-to-End Simulation-to-Real Domain Adaptation Framework for LiDAR Point Cloud Segmentation, (2020). <http://arxiv.org/abs/2009.03456>.
- [212] Velodyne HDL32E Datasheet, (n.d.). <https://velodynelidar.com/products/hdl-32e/> (accessed May 25, 2021).
- [213] Velodyne Puck Hi-Res Datasheet, (2022). <https://velodynelidar.com/products/puck-hi-res/> (accessed May 25, 2021).
- [214] Riegl VUX-1HA, (n.d.). <http://www.riegl.com/products/newriegl-vux-1-series/newriegl-vux-1ha> (accessed April 25, 2020).
- [215] Z+F Profiler 9012, (n.d.). https://www.zf-laser.com/Z-F-PROFILER-R-9012.2d_laserscanner.0.html (accessed April 25, 2020).
- [216] Applanix POSLV 220 Datasheet, (n.d.). online: <https://www.applanix.com/products/poslv.htm> (accessed April 25, 2020).
- [217] A. Habib, J. Lay, C. Wong, LIDAR Error Propagation Calculator, (2006). <https://engineering.purdue.edu/CE/Academics/Groups/Geomatics/DPRG/files/LIDARErrorPropagation.zip> (accessed October 9, 2021).
- [218] CloudCompare. 3D point cloud and mesh processing software, (n.d.). <https://www.danielgm.net/cc/> (accessed February 7, 2021).
- [219] S. Guinard, L. Landrieu, Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. XLII-1/W1 (2017) 151–157. <https://doi.org/10.5194/isprs-archives-XLII-1-W1-151-2017>.
- [220] Y. Li, R. Zemel, M. Brockschmidt, D. Tarlow, Gated graph sequence neural networks, *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. (2016) 1–20.

- [221] M. Simonovsky, N. Komodakis, Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs, Cvpr. (2017) 3693–3702.
http://openaccess.thecvf.com/content_cvpr_2017/papers/Simonovsky_Dynamic_Edge-Conditioned_Filters_CVPR_2017_paper.pdf.
- [222] L. and L.H. Refaeilzadeh Payam and Tang, Cross-Validation, in: M.T. Liu Ling and Özsü (Ed.), Encyclopedia of Database Systems, Springer New York, New York, NY, 2016: pp. 1–7. https://doi.org/10.1007/978-1-4899-7993-3_565-2.