PLANE EXTRACTION FROM INHOMOGENEOUS POINT CLOUDS AND PLANE-TO-PLANE ALIGNMENTS

WENZHENG FAN

PhD

The Hong Kong Polytechnic University
2020

The Hong Kong Polytechnic University

Department of Land Surveying and

Geo-Informatics

Plane Extraction from Inhomogeneous Point Clouds and Plane-to-plane Alignments

Wenzheng FAN

A thesis submitted in partial fulfillment of the requirements for the

degree of Doctor of Philosophy

July, 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, of the best of knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Wenzheng FAN (Name of Student)

Plane Extraction from Inhomogeneous Point Clouds and

Plane-to-plane Alignments

Abstract

Over the last few years, increasing demands for building interior surveying have brought the challenge of acquiring the geometric information of indoor environments effectively and efficiently. Though the development of robotic engineering had provided some preliminary solutions for building the virtual world for the robotics, their precision and accuracy could not satisfy the applications in Architecture, Engineering, and Construction (AEC). Various solutions have been introduced with limitations in robust feature extraction, working range, accessibility, detail levels, coverage, and reliability. Meanwhile, popular methods used for extracting features, especially planes, from point clouds for mobile mappings were not sufficient, accurate and robust enough for establishing 6 Degree-of-Freedom (DOF) point alignment workflows.

A novel method for plane extraction from low-resolution inhomogeneous point clouds captured by multi-line Mobile Laser Scanners (MLS), the Enhanced Line Simplification (ELS) algorithm, was proposed and developed. The method employed raw data acquisition sequence to form point grids for analyzing curvatures and identifying feature points along raw scanlines and dedicated virtual scanlines. By clustering identified line segments concerning scanline directions, patches were

detected and merged to form planes. The method eliminated the calculation of the local estimated normals and overcame the over-segmentation problem in processing noisy data.

Utilizing the plane extraction results, a plane-to-plane point cloud alignment workflow was presented for aligning point clouds captured on a mobile platform to the same frame. The workflow recovered the rotation and translation relationships between frames by identifying common planes and non-linear optimization. The implementation of the coarse-to-fine procedure and the shortest-path initialization strategy waived the use of Inertial Measurement Units (IMU) and other positioning sensors.

A backpack prototype, which adopted two multi-line laser scanners as the primary sensors, was designed to test the performance of the proposed methods in multiple scenarios. The results showed that the proposed hardware system and the processing workflow could achieve acceptable accuracy and reliability in typical indoor and specific outdoor environments. The robustness of this IMU-free point cloud alignment workflow was verified as well, which could be applied in future mobile mapping and sensor fusion applications.

Publications Arising from the Thesis

Journal Papers

- **Fan, W.**, Shi, W., Xiang, H., & Ding, K. (2019). A Novel Method for Plane Extraction from Low-Resolution Inhomogeneous Point Clouds and its Application to a Customized Low-Cost Mobile Mapping System. *Remote Sensing, 11*, 2789.

- Shi, W., Ahmed, W., Li, N., **Fan, W.**, Xiang, H., & Wang, M. (2018). Semantic Geometric Modelling of Unstructured Indoor Point Cloud. *ISPRS International Journal of Geo-Information, 8(1)*, 9.

Conference Proceeding

- **Fan, W.**, & Shi, W. (2015). Feature Point Extraction Using Extended 2D Line Simplification Algorithm for Indoor Mapping. In *The 9th International Symposium on Mobile Mapping Technology, MMT2015*. Sydney, Australia.

Patent

- Shi, W. & **Fan, W.**, *Plane extraction method, system, device and storage medium based on point cloud data.* P.R. China Patent application 201811167642.5.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Wen-zhong, John SHI, a respectable, responsible, and resourceful scholar who has provided me with valuable guidance and help. All our team members, especially Dr. Yong LIU, Dr. Zhongbin LI, Dr. Zelang MIAO, Dr. Na LI, Dr. Wenyuan ZHANG, Dr. Xing YAN, Dr. Cheng YANG, Dr. Ivan RUIZ, Ms. Yao HAN, Mr. Teng LIU, Ms. Rong KOU, Ms. Xiaolin ZHOU, Mr. Haodong XIANG, Mr. Muyang WANG, and Mr. Mingyan NIE have also provided me a profusion of help and suggestions for my study and research. Special thanks also to Prof. Han HU, Dr. Xuming GE, Dr. Lei YE, and Dr. Peichao GAO for their generous help.

I also extend my thanks to all the academic staff of LSGI: without their teaching and help during the past years of studying, I would not have enriched myself and gotten prepared for this thesis and future study and research. Special thanks to the non-academic staff without whose support and assistance I might not have been able to concentrate on my research and study.

Finally, I would like to acknowledge the support provided by my beloved wife, Juliet, my adorable daughter, Elsie, and other family members during my PhD study.

Contents

List of Figures

List of Tables

Abbreviations

2D	two-**D**imensional
3D	three-**D**imensional
AEC	**A**rchitecture, **E**ngineering, and **C**onstruction
AHRS	**A**ttitude and **H**eading **R**eference **S**ystem
ALS	**A**irborne **L**aser **S**canning
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
BIM	**B**uilding **I**nformation **M**odeling
CSSD	**C**orrected **S**ample **S**tandard **D**eviations
CSV	**C**orrected **S**ample **V**ariance
DEM	**D**igital **E**levation **M**odel
DOF	**D**egree-**o**f-**F**reedom
DR	**D**ead **R**eckoning
DSM	**D**igital **S**urface **M**odel
DSP	**D**igital **S**ignal **P**rocessors
DTM	**D**igital **T**errain **M**odel
EKF	**E**xtended **K**alman **F**ilter
ELS	**E**nhanced **L**ine **S**implification
FOV	**F**ield **O**f **V**iew
FPS	**F**rame **P**er **S**econd
GCS	**G**round **C**ontrol **P**oint
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem

g2o	**G**eneral **G**raph **O**ptimization
HKPU	The **H**ong **K**ong **P**olytechnic **U**niversity
HKSAR	**H**ong **K**ong **S**pecial **A**dministrative **R**egion
ICP	**I**terative **C**losest **P**oints
IFOV	**I**nstantaneous **F**ield **O**f **V**iew
IMU	**I**nertial **M**easurement **U**nit
INS	**I**nertial **N**avigation **S**ystem
ITC	**I**nnovation and **T**echnology **C**ommission
***k*NN**	**K**-**N**earest **N**eighbors
LADAR	**LA**ser **D**etection **A**nd **R**anging
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LoD	**L**evel **o**f **D**etails
LoS	**L**ine **o**f **S**ight
LSCM	Hong Kong R&D Centre for **L**ogistics and **S**upply **C**hain **M**anagement Enabling Technologies
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystem
MLESAC	**M**aximum **L**ikelihood **E**stimation **SA**mple and **C**onsensus
MLS	**M**obile **L**aser **S**canner
MMS	**M**obile **M**apping **S**ystem
MSAC	**M**-estimator **SA**mple and **C**onsensus
MSVC	**M**icro**s**oft **V**isual **C** or **C++**
NDT	**N**ormal **D**istribution **T**ransformation
NMEA	**N**ational **M**arine **E**lectronics **A**ssociation
Pt	**P**oin**t**
PC	**P**ersonal **C**omputer
PCA	**P**rincipal **C**omponent **A**nalysis
PDFSLAM	**P**anoramic **D**irect **F**eature **SLAM**
PIO	**P**ower/**I**nput/**O**utput
POS	**P**osition and **O**rientation **S**ystem

PPS	**P**ulse **P**er **S**econd
PROSAC	**PRO**gressive **SA**mple and **C**onsensus
RANSAC	**RAN**dom **SA**mple **C**onsensus
RGB-D	**R**ed, **G**reen, **B**lue and **D**epth
RMC	**R**ecommended **M**inimum sentence **C**
RMSE	**R**oot **M**ean **S**quare **E**rror
S^{2}DAS	**S**eamlss **S**patial **D**ata **A**cquisition **S**ystem
SDK	**S**oftware **D**evelopment **K**it
SfM	**S**tructure **f**rom **M**otion
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SSD	**S**olid **S**tate **D**rive
STL	**S**tandard **T**emplate **L**ibrary
SVD	**S**ingular **V**alue **D**ecomposition
TCHV	**T**otal **C**onvex **H**ull **V**olume
TGA	**T**otal **G**rid **A**rea
TLS	**T**errestrial **L**aser **S**canner
ToF	**T**ime-**o**f-**F**light
UKF	**U**nscented **K**alman **F**ilter
vSLAM	**v**isual **S**imultaneous **L**ocalization **A**nd **M**apping

Chapter 1

Introduction

GIS data capturing is a critical technique for modeling the real world. In the past decades, sensors have been developed for capturing geometric data of specific locations, such as levels, theodolites, total stations, and Global Navigation Satellite System (GNSS) receivers. Based on the discrete geometric data captured, math models and filtering methods were applied to generate fitted models of the real world.

To meet the demands for obtaining continuous or quasi-continuous geometry data, especially information about terrain surfaces, such as Digital Elevation Model (DEM) and Digital Surface Model (DSM) of urban and forest areas, advanced technologies using close-range photogrammetry and Light Detection And Ranging (LiDAR) have provided innovative methods for capturing quasi-continuous geometric data of surfaces of objects. Different from the data captured using the conventional surveying techniques, which adopt geometry information at specific locations to

represent regional shapes, the quasi-continuous data generated by photogrammetry and laser scanning enabled precise modeling with a nearly complete degree of freedom. In addition, the tedious surveying procedures were also simplified to a reduced number of survey stations and an automatic observation for a large area at a single time (Brilakis et al., 2010).

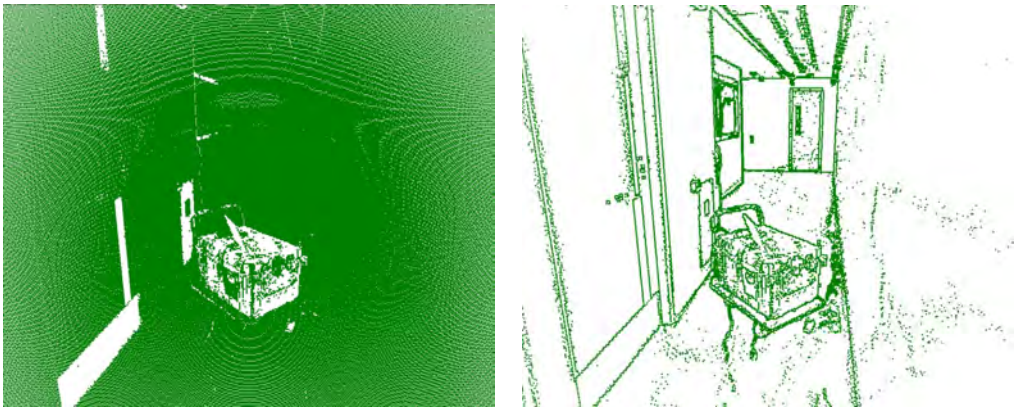
In recent years, the development of Building Information Modeling (BIM) introduced a new challenge of acquiring detailed shapes and textures of indoor environments, making the indoor mapping technology a highlighted research area globally. Under the circumstances, methods and solutions were proposed to deal with the most familiar but complex environments. The most commonly used technology for mapping in the narrowed indoor environment was laser scanning. Since pulses could be reflected by objects from any incident angles, regardless of the presence of texture, laser scanning was more convenient than photogrammetry for mapping detailed indoor scenes and generating seamless models.

Meanwhile, the existence of artificial objects makes the point clouds of indoor environments different from those of surface terrains and outdoor scenarios. Compared with the outdoor environments, the point clouds of indoor environments have the following unique characteristics, according to W. Fan (2015).

First, the feature point identification of indoor environments is crucial for indoor mapping and visualization. Feature points are defined as the points with sharp geometric changes, or points that are locally most representative in object shapes. In other words, when all non-feature points are removed from raw point clouds, the

shapes of the objects should be clear enough for human understanding, as shown in *Figure 1.1*. Since most indoor surfaces are planar surfaces, such as walls, floors and ceiling surfaces, if plane boundaries and feature points can be adequately identified, the segmentation of point clouds can accordingly be used for modeling indoor planar surfaces. Meanwhile, based on the levels of interest, some uneven surfaces can be replaced or represented by planar approximations or combinations of planar surfaces, such as curtains, ceiling ornaments, and decorations. Thus, feature points of such objects are essential for determining the plane shapes and approximating the surface of arbitrary shapes. As the approximations significantly decrease the complexity of object shapes and the point quantity, feature points can be used to visualize sharp surface curvature changes on low-performance devices, which is particularly useful in real applications. Furthermore, as the fundamental principle of Simultaneous Localization and Mapping (SLAM) is feature extraction and matching, the precise extraction of feature points and planes will improve the precision, accuracy, and reliability of indoor SLAM since they are choosing the most representative points for matching, aligning and registering.

Second, since the resolution of indoor point clouds is a function of the range, which is the distance between the sensor and the target object, and the angular resolution, which is determined by the laser scanner while manufacturing, there are severe detail loss problems in indoor spaces. Confined spaces, such as hallways, washrooms, and pantries, commonly require more scan stations than wide spaces. If not maintained properly, the point resolution might be too small and lead



(a) (b)

FIGURE 1.1: Feature points helping human understanding. (a) The point cloud with all points showing no feature is hard to understand. (b) With feature points extracted and only represented, the corridor structure is easy to identify with the doors and a trolley in front of the viewer.

to edge-loss problems, which may require prediction or approximation of the edges rather than precise measurements (P. Tang, Akinci, & Huber, 2009). Additionally, the inhomogeneous phenomenon brings a significant challenge for feature extraction since it may omit the surface features if the checking kernels were not well selected. Consequently, the robust method for extracting features from inhomogeneous point clouds is also a challenge for indoor SLAM.

The most considerable difference between indoor point clouds and other point clouds is the inevitable obstacles and specular surfaces in indoor environments. Since the distances between the sensor and the target objects are much shorter than those in the airborne LiDAR, the influences made by such obstacles and noisy points are more critical. The borders of stationary obstacles need to be clearly identified for the prediction based on the shape characteristics of continuous surfaces, while the noisy points arising from moving obstacles and specular surfaces, such as glasses,

mirrors, and extremely smooth metal surfaces, need to be detected and removed since they are reflected by objects that do not exist in real scenes. These three kinds of noisy points can cause scanline curvature changes, and they can be detected by identifying feature points representing curvature changes.

The comparison above shows that of the characteristics of indoor point clouds, feature point extraction is critical and valuable for data filtering, processing, and visualizing of indoor point clouds. Meanwhile, other problems of establishing a precise and accurate 3D GIS of indoor environments also need to be solved.

Mobile Mapping Systems (MMS) have been widely applied in mapping outdoor environments. The systems synergized various sensors and technologies to achieve continuous positioning and orientating. Researchers have tried to introduce outdoor solutions into indoor environments, but the lack of reliable positioning methods limits applications in such spaces. Though technologies and sensors have been developed, there is still no satisfying solution for mobile mapping the building interiors due to limitations in the working range, accessibility, detail levels, coverage, and reliability.

In this thesis, a novel algorithm for extracting feature points for indoor scenarios and its application to plane extraction from low-resolution inhomogeneous point clouds is introduced and discussed. For further discussion on its usage and applications, a 3D SLAM algorithm for mapping artificial indoor environments, and its carrier, the Seamless Spatial Data Acquisition System (S^2DAS), is proposed and discussed.

The rest of the thesis is organized as follows: In Chapter 2, current available sensors and solutions for indoor 3D GIS data acquisition are introduced, followed by a review of plane extraction methods and feature-based SLAM techniques. Chapter 3 discusses the experiments and discussion based on the modified Enhanced Line Simplification (ELS) algorithm together with its application to plane extraction from low-resolution inhomogeneous point clouds. As the applications of ELS algorithm did not require any hardware modifications, the design of the proposed mobile mapping backpack prototype was introduced in Chapter 4, together with the application of ELS-extracted planes, which is the plane-to-plane alignment workflow for mobile mapping. The corresponding point cloud alignment results are presented and discussed. Finally, Chapter 6 provides conclusions and future working directions.

Chapter 2

Mobile Mapping Sensors, Systems, and Technologies: A Review

Various sensors and technologies have been designed and adopted for recovering the real world in 3D GIS. For example, laser scanning is one of the most popular technologies for mapping indoor environments. Point clouds captured at different locations are registered and merged using common features, targets and pre-known positioning and orientation data. Point clouds can be used to generate 3D models, the fundamental geometric elements of 3D geodatabases.

However, such station-by-station workflow may not be fluent enough in some narrow indoor spaces, as the registrations between stations are difficult in specific indoor scenarios. For instance, in narrow spaces and complex environments with obstacles, direct line-of-sight (LoS) may be blocked, leaving remote possibility for

resection. Another example would be the lack of adequate distinguishable common features and Ground Control Points (GCP) for co-registration when trying to register two or more scan stations into the same coordinate frame. Although artificial targets, in the form of spherical and hemispherical targets, could be installed to facilitate the registration process, the lack of LoS in such environments still prevents successful alignments between scan stations. To fulfill the needs of mapping in such scenarios, indoor MMS, which are similar to outdoor MMS, were introduced and have become one of the most popular indoor mapping solutions.

In indoor environments, GNSS signals are no longer available and reliable, making the positioning and orientation of the mapping platform a challenge. The Inertial Navigation System (INS) is dedicated to providing orientation and positioning data to some extent, although improvements are still needed in precision, reliability, and costs. Consequently, researchers in the fields of robot control, computer vision, and surveying have developed SLAM algorithms to determine the position and orientation of the mobile platform in GNSS-denied environments. According to Leonard and Durrant-Whyte (1991), SLAM was the simultaneous process that combined the map building process, in which the geometric data captured by moving sensors were aligned to the global reference frame, with the localization process, in which the positions were determined using the geometric data on maps accumulated in the mapping process.

In past years, new requirements were made for higher reliability, accuracy, and robustness to facilitate the application of SLAM techniques to the mobile mapping

process.

- **Reduce the requirements on GCP**. GCP are reliable controls and constraints for noisy and unreliable SLAM positioning and orientation results, and the key for registering the final dataset to the global reference frame. Their number and accuracy determine the overall indoor mapping accuracy. The proper distribution of the GCP is determined by the degree of complexity of the indoor environments, the planned path of the mobile mapping process, and the design of the mapping solutions. For instance, staircases always require more GCP for vertical variation control compared with flat rooms. The accuracy of the GCP coordinates influences the registration accuracy and the elimination of the drift error of the IMU and the SLAM algorithms.

- **Extract valid features from the point cloud regardless of the resolution changes**. For LiDAR-based systems, extracting feature points from single-frame point clouds plays a vital role in indoor SLAM workflow. In such environments, the resolution directly affects the level of detail (LoD) the dataset can achieve. The higher the LoD, the easier it is to extract the feature points. Meanwhile, low-resolution point clouds may result in edge-loss problems, which are challenging to recover (P. Tang et al., 2009). Therefore, some of the solutions integrate multiple LiDAR to increase the resolution of point clouds. If the resolution cannot be maintained, the algorithm adopted must be prepared for processing such inhomogeneous data. Otherwise, the assistance of ancillary information might be required.

- **Integrate multiple sensors**. Different sensors used for geometric and texture data acquisition, positioning, and attitude determination are integrated organically into most of the solutions to make more relevant data available, improving the reliability of the SLAM algorithm. Sensor integration is used not only for data fusion but also as corrections for other sensors to reduce drifting errors, achieving higher accuracy and reliability. However, improving the accuracy and reliability of single-sensor SLAM procedures is still an important research topic because improving a single component would enhance the whole system.

To fulfill the requirements above, both experimental and commercial solutions are introduced for mapping indoor environments. In this chapter, specialized technologies, sensors, and solutions are presented, with discussions on their advantages and limitations.

2.1 Mobile Mapping Sensors

Sensors installed in indoor mapping solutions can be grouped into three categories: a) point cloud generation sensors responsible for geometric and texture data acquisition, b) positioning and navigation sensors and technologies designed to provide positioning information or geo-reference information for geometric and texture data, and c) the attitude-determination sensors providing the corresponding

attitude information of the sensor platform. By fusing data obtained from the sensors of the three groups, point clouds, textures, and other kinds of data that describe the real world can be generated to model the objects of interest.

2.1.1 Point Cloud Generation Sensors

Several technologies are involved in generating the point clouds of indoor mapping solutions, as shown in *Figure 2.1*. The basic working principles of each sensor are briefly introduced and compared in this section.

2.1.1.1 Terrestrial Laser Scanner (TLS)

As shown in *Figure 2.1(a)*, TLS is designed to capture point clouds of the surrounding environments by rotating its internal prism or mirror and itself against the horizontal axis and its base to make two-axis rotations of the laser beams (Shan & Toth, 2018). The distance between the laser scanner and the object can be calculated using *Equation 2.1* while the coordinate of the object reflection can be derived from *Equation 2.2* as the angle encoders inside the TLS work with the ranger to provide horizontal and vertical angles together with the travel time of the laser's round trip (Shan & Toth, 2018).

$$r = \frac{v \cdot \Delta t}{2} \tag{2.1}$$

where:

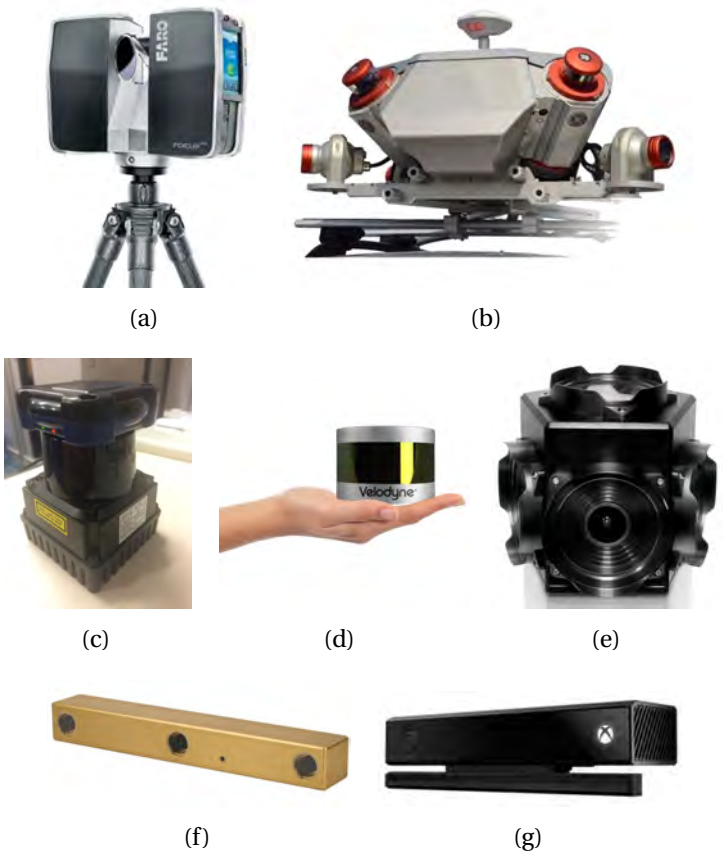


FIGURE 2.1: Popular point cloud generation sensors for indoor mobile mapping. (a) FARO Focus^{3D} S 120 TLS © FARO Copyright 2015. (b) RIEGL VMX-450 MLS © RIEGL Copyright 2015. (c) Hokuyo UTM-30LX 2D MLS. (d) Velogyne VLP-16 16-line 3D MLS © Velodyne Copyright 2019. (e) FLIR Ladybug 5 panoramic camera © FLIR Copyright 2015. (f) FLIR Bumblebee XB3 sensor © FLIR Copyright 2015. (g) Microsoft Kinect depth camera (Gen. 2) © Microsoft Copyright 2015.

r is the direct LoS distance between the laser scanner and the object,

v is the speed of the laser beams,

Δt is the round travel time of the laser beam.

$$\begin{cases} x = r\sin\theta\cos\phi \\ y = r\sin\theta\sin\phi \\ z = r\cos\theta \end{cases} \tag{2.2}$$

where:

(x, y, z) is the coordinate of the pending point in the Cartesian coordinate frame of the TLS,

θ is the vertical/elevation angle of the laser beam in the spherical coordinate frame of the TLS,

ϕ is the horizontal/azimuth angle of the laser beam in the spherical coordinate frame of the TLS.

Therefore, the accuracy of TLS point coordinates is affected by errors in the horizontal and vertical angle encoders and the distance measurements. The accuracy of static TLS could be up to millimeter-level, which makes it perfect for capturing point clouds for building the interior models that could be applied to AEC. The ranges of TLS vary from tens of meters to thousands of meters, determined by the power of the laser beams and the working principles. For those applied in AEC, the ranges are commonly less than a hundred meters. Varieties of TLS are listed with detailed information in Shan and Toth (2018), which will not be introduced here.

Most conventional spatial data acquisition solutions are based on TLS working in fully static mode, stop-and-go mode, and profile mode (Chow, 2014) (Applanix Corp., 2015). In static mode and stop-and-go mode, point clouds at scan stations can be fused using reflective targets and spherical and hemispherical targets as GCP, or Iterative Closest Point (ICP) algorithm, which uses the overall shape and point distribution as the reference. The point registration process merges the clouds into

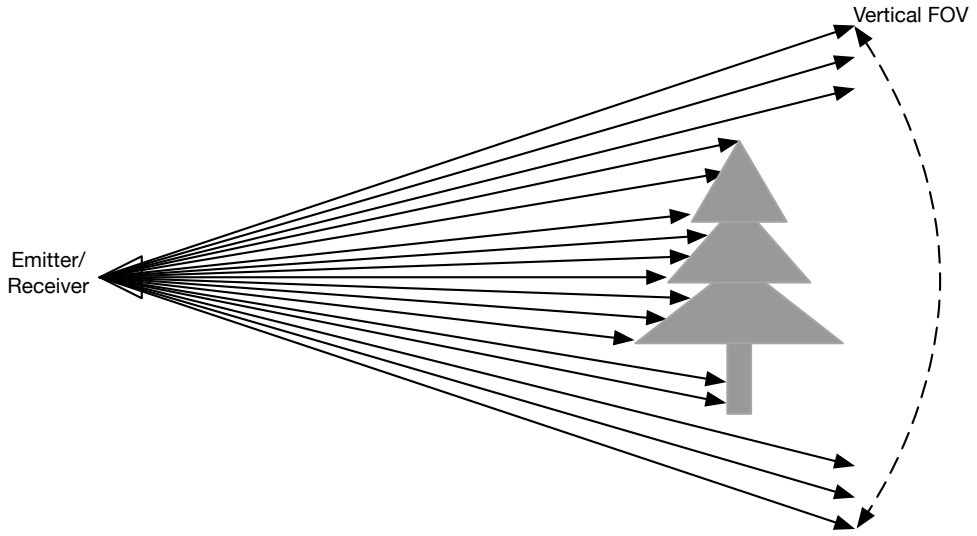
the same reference frame and forms the point cloud describing the indoor environments as a single dataset. In profile mode, the scanner works as a single-line profiler which maintains the horizontal angle unchanged while the platform keeps moving. However, the profile mode requires individual permissions from the manufacturer as most TLS would warn the users and stop work when its internal sensors sense extensive vibrations or movement.

2.1.1.2 Mobile Laser Scanner (MLS)

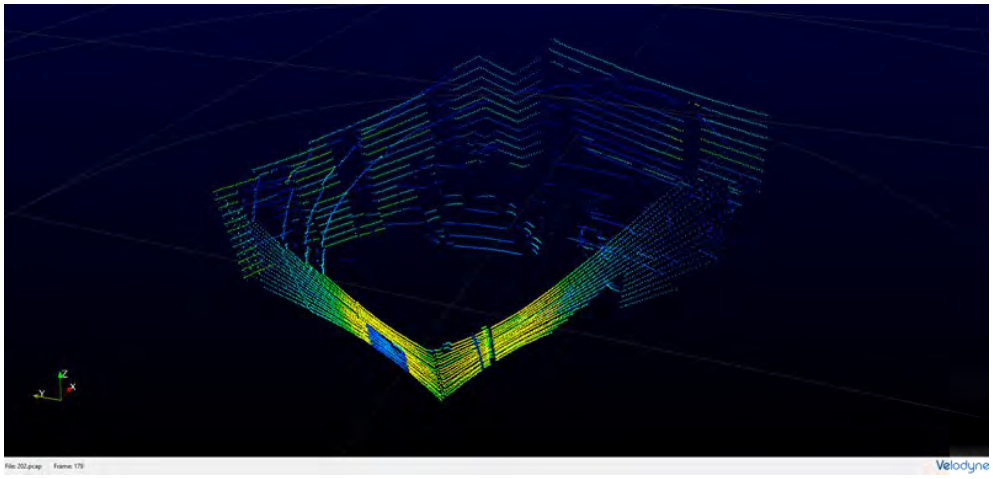
According to Shan and Toth (2018), MLS, which is also called Dynamic Terrestrial Laser Scanner, is a kind of scanner that is installed on dynamic platforms, such as airplanes, boats, and ground-based vehicles, and rotates its internal prism to capture reflections of laser beams to generate point clouds of the surrounding environments. Two kinds of MLS are available. One, the conventional MLS is similar to the TLS, and creates points by rotating the internal prism; the second, the flash LiDAR, generates LiDAR beams in the form of grids to scan onward fields.

Similar to TLS, the prism inside the conventional MLS rotates against its axis to create point clouds. However, there is only one rotation axis inside the MLS as the conventional MLS is similar to the profiler installed horizontally. For single-line MLS, such as the RIEGL VUX-1 profiler shown in *Figure 2.1(b)* and the Hokuyo UTM-30LX 2D MLS shown in *Figure 2.1(c)*, the prism rotates against the vertical axis to create a 2D point cloud covering a specific horizontal Field of View (FOV). Meanwhile, in multi-line MLS, such as the Velodyne VLP-16 shown in *Figure 2.1(d)*,

additional laser-detector pairs are installed to generate laser beams with inclination angles against the vertical axis, providing beams with the extra vertical dimension and generating 3D point clouds (Glennie, Kusari, & Facchin, 2016). The working principle is shown in *Figure 2.2(a)* while an example of the point clouds captured is shown in *Figure 2.2(b)*.



(a)



(b)

FIGURE 2.2: Working principle and point clouds of muilti-line MLS. (a) The laser beams emitted by the emitter, reflected by the object, and received by the receiver. (b) A single frame of MLS point cloud of room interior captured using VLP-16 and visualized in VeloView provided by Velodyne.

With the developing of autonomous driving technology, the previous military-specified Geiger-mode scanner introduced by Shan and Toth (2018) was redeveloped to a flash LiDAR. In such scanners, tiny emitters, which are commonly Micro-Electro-Mechanical System (MEMS) parts, are installed as plane arrays to generate laser beams illuminating a relatively wide FOV with a much higher resolution and updated frequency (M. Wang, 2018). Benefiting from modern development in the semiconductor industry, the price of such scanners is much lower than traditional scanners, and the price decreases with increasing production volume. However, the accuracy of such flash light scanners is much lower, around 5 cm or less on measurement accuracy. It can only fit needs in autonomous driving rather than in mapping applications discussed in this thesis.

The calculation of the MLS point coordinates is still based on *Equation 2.1* and *2.2*. However, as the MLS keeps generating points in high speed and there is no on-board storage in the scanner, the data are compressed for data transmission. Only horizontal angles of the first point of a group of points are explicitly listed in the output data package. Taking scanners made by Velodyne as an example, only the azimuth angle of the first point of every thirty-two points is listed (Velodyne Acoustics Inc., 2015). Therefore, the horizontal angles of the rest points in the group can only be derived using *Equation 2.3*, *2.4* and *2.5*.

$$\Delta t_i = \Delta t_S * I_S + \Delta t_{DP} * I_{DP} \tag{2.3}$$

where:

I_S is the sequence index of the pending point i, which is the group index of the sixteen-point group in the data package,

I_{DP} is the data point index of the pending point i, which is the sequence index of the point i in the sixteen-point group and corresponding to the laser beam ID,

Δt_S is the given time interval for every sixteen beams of laser fires,

Δt_{DP} is the given time interval between each of the adjacent laser fires,

Δt_i is the time offset of the pending point i in the 384-point data packet.

$$\Delta\phi_j = \phi_{j+1} - \phi_j \tag{2.4}$$

where:

ϕ_j is the horizontal angle of the j-th data block (the j-th group of thirty-two points),

$\Delta\phi_j$ is the time interval between the first point of the j-th and $(j+1)$-th data blocks.

$$\begin{cases} \phi_i = \dfrac{\Delta t_i - t_j}{\Delta t_S * 2} * \Delta\phi_j + \phi_j \\ \theta_i = \theta_{I_{DP}} \end{cases} \tag{2.5}$$

where:

t_j is the time offset of the first point in the j-th thirty-two-point data block,

Δt_S is given time interval for each sixteen beams of laser fires,

ϕ_i is the horizontal angle of the pending point i,

$\theta_{I_{DP}}$ is the vertical angle of the giving data point index, which can be got from *Table 2.1* provided by Velodyne Acoustics Inc. (2015),

θ_i is the vertical angle of the pending point i.

TABLE 2.1: The corresponding firing sequences, fire ID, and vertical angles of Velodyne VLP-16 3D scanner according to Velodyne Acoustics Inc. (2015)

Raw Sequence ID	Fire ID	Vertical Angle (°)
1	0	-15
2	1	1
3	2	-13
4	3	3
5	4	-11
6	5	5
7	6	-9
8	7	7
9	8	-7
10	9	9
11	10	-5
12	11	11
13	12	-3
14	13	13
15	14	-1
16	15	15
1 (∗)	0	-15

(∗)Note: This is the first fire of the next firing loop showing the horizontal angle change between loops.

Therefore, the accuracy of MLS point coordinates is affected by errors from the horizontal angle encoders, distance measurements, installation error of the vertical

prism (errors in vertical angle due to inaccurate installation), and variation in rotation speed. As the sizes of MLS are smaller while the mechanisms inside are more complicated, the accuracy of such MLS cannot achieve the accuracy of TLS. The commonly declared measurement accuracy of such MLS is 3-5 cm while the best could reach 2 cm (Velodyne Acoustics Inc., 2012) (Velodyne Acoustics Inc., 2015) (Suteng Innovation Technology Co Ltd, 2015). As many of such scanners are based on Time-of-Flight (ToF) mechanism and the update rate is up to 20 Hz, the valid working range is limited to around 100 m.

Meanwhile, when the manufacturer changes the vertical installation angles of the laser-detector pairs, the vertical FOV and resolution can be easily adjusted to fulfill requirements in both autonomous driving and mobile mapping (*Table 2.1*). To increase the resolution for semantic analysis, most vendors tend to reduce the interval between scanlines for close-range targets by decreasing the vertical angle intervals between scanlines whose inclination angles are close to 0 °, such as Hesai Pandar40 (*Figure 2.3*).

Although the MLS might not be a perfect solution for autonomous driving due to their relatively high expenses, they, especially the single-line models produced by Hokuyo and SICK and multi-line models launched by Velodyne, are currently the most commonly adopted mobile mapping sensors. However, the accuracy of such scanners cannot satisfy their application in the field of AEC, especially for generating as-built BIM, which requires millimeter-level accuracy.

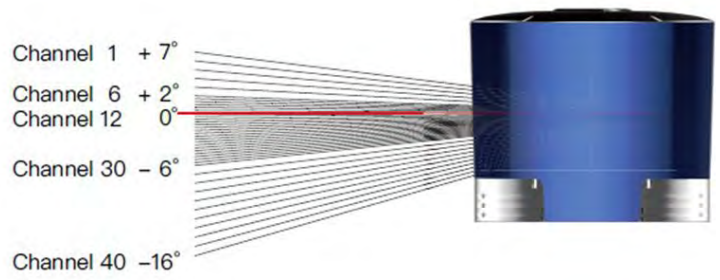


FIGURE 2.3: Vertical Channel Distribution of the laser channels in the Hesai Pandar40 Multi-line Scanner. The vertical intervals between each of the neighboring channels between channel 1 and 6 and channel 30 and 40 are 1°, while the intervals between each of the neighboring channels between channel 6 and 30 are 0.33°. The overall coverage is from 16° below the horizontal plane to 7° above the horizontal plane. © Hesai Copyright 2018

2.1.1.3 Stereo Camera

Few solutions are based on stereo cameras because the accuracy is related to the main distance between two sensors, and the resolution is limited by the quality of visible textures.

A few commercial solutions adopt close-range photogrammetry based on sequential images captured by monocular or panoramic cameras (*Figure 2.1(e)*). Although panoramic cameras are commonly used in such systems, the overlapping FOV is not large enough for building images for instant stereo images. The working principle of close-range photogrammetry based on sequential images is by knowing the interior orientation, which can be obtained by precise calibration in the laboratory, and exterior orientation, which is derived from the positioning and orientation of the mapping platform, to perform close-range photogrammetry processing based on the feature and common points extracted from images (Iwane Laboratories Ltd, 2010). Along with the increase of the baseline length, which is the distance

between the data acquisition positions of neighboring images, the accuracy could be up to tens of centimeters, fitting the needs for GIS database building.

Meanwhile, portable sensors based on binocular cameras are available (*Figure 2.1(f)*), whose accuracy is related to and restricted by the distance between the sensor and the objects. As the gaps between the two optical sensors are usually around tens of centimeters, the accuracy of binocular systems is worse than 10 cm when the distance between the sensor and the object exceeds 4 m (FLIR Integrated Imaging Solutions Inc, 2011). Intel released its new generation of RealSense devices in January 2018, and the stereo cameras based on artificial features, commonly infrared tags, became commercially available with the valid range of a few meters (Keselman, Woodfill, Grunnet-Jepsen, & Bhowmik, 2017) (Intel® Software, 2018).

2.1.1.4 Depth Camera

Originally designed for entertainment purposes, depth cameras, or RGB-Depth (RGB-D) cameras as they can acquire both RGB information and depth data, are advanced sensors based on infrared triangulation or ToF technology (Figure 2.1(g)). The output of the depth camera is an RGB image and a depth image of the Instantaneous FOV (IFOV) where the multi-sensor camera is facing. Some researchers also categorize sensors like FLIR Bumblebee as a depth camera since it can directly output RGB-D images based on binocular photogrammetry. However, this thesis still considers RGB-D cameras as a kind of sensor equipped with a dedicated emitter and a corresponding receiver, usually a pair of infrared sensors, to generate depth

maps of facing areas. Examples of such cameras include the Microsoft Kinect I and II, Google Project Tango series (discontinued), Asus Xtion sensor (discontinued), and Occipital Structure series (Haggag et al., 2013) (Darwish, Tang, Li, & Chen, 2017).

Recent developments still limit the working distance, resolution, and data acquisition rate of depth sensors though scholars and researchers have already adopted them as a convenient surveying tool. Many scholars are working on the calibration and working range extension, considering the compact sizes and low prices of RGBD sensors. Darwish et al. (2017) introduced an online method that fused on the data captured by the optical cameras of the Structure Sensor and its carrier, which is an iPad, to form stereo images to calibrate the depth sensor and increase the valid range as well. The Root Mean Square Error (RMSE) at 2.5 m distance can be reduced to around 20 mm while the overall working range can reach up to 9 m. However, the limited accuracy at the maximum range and the accuracy decreasing with the distance exponentially still limit the application of such sensors in large space mobile mapping.

2.1.1.5 Structure-from-motion (SfM) and Visual SLAM (vSLAM)

SfM and vSLAM are two similar techniques rather than hardware sensors. They can both recover the 3D structures of surrounding environments from monocular 2D images. However, the differences are pronounced.

By extracting and matching texture features, SfM connects 2D images with no pre-known interior and exterior orientation based on tie points only. It recovers geometric relationships of unordered images taken from different viewpoints (Schonberger & Frahm, 2016). In addition to its inability to provide general, complete, and robust solutions owing to the absence of texture features extracted, the most significant limitation of this technology is that the algorithm cannot regain the scale factor of the structures. Therefore, the implementation of the binocular SfM algorithms always requires control points to recover the scale factors. However, as there is no limitation on the image acquisition sequences, SfM has been widely applied in the reconstruction of 3D object models by processing images taken at locations surrounding the object. In the meantime, as single cameras are easy to carry and install on unmanned vehicles and SfM can work even with the generally available internet photos, scholars are working in this field to reconstruct ground objects using large-scale image databases (Schonberger & Frahm, 2016) (Ewertowski, Tomczyk, Evans, Roberts, & Ewertowski, 2019).

Similarly, vSLAM is a batch of algorithms which recover the 3D geometric relationship using tie points extracted from 2D images. The sensors adopted can be either monocular or binocular; the monocular solutions require the scale parameter as an input factor because the algorithm itself cannot directly recover the factor. Besides the difference in adjustment methods, the most appreciable difference between SfM and vSLAM is that the initial state between the ordered images in vSLAM is often well-known by using other sensors, such as GNSS receivers or IMU Dead

Reckoning (DR), or just zero difference from the frames ahead in the ordered image sequence. Also, the main product of vSLAM is the trajectory of the sensor, while the maps generated are often byproducts.

Although the two algorithms can both generate 3D maps and are widely used in mobile mapping, they will not be discussed in detail as they are both data processing algorithm whose points are generated as products rather than sensors through which depth maps or point clouds can be directly produced.

2.1.1.6 Comparison of the Point Cloud Generation Sensors

The key specifications of the sensors above are listed in *Table 2.2*. Listed features are considered as criteria for choosing sensors for mobile mapping based on the requirements of the applications, such as BIM generation, progress monitoring, navigation.

For BIM generation, experts from the AEC industry typically require millimeter-level accuracy, making stationary and stop-and-go TLS the only choice. Navigation applications often require lightweight solutions, accepting only small-size camera sensors, either stereo cameras or depth cameras.

The reasonable level of accuracy and working range of MLS is currently a more popular solution for most outdoor and indoor solutions. The 360 ° horizontal FOV provides the possibility of SLAM with high robustness using the surrounding environments, while the vertical FOV of tens of degrees enables 3D SLAM in most scenarios. By installing more than two multi-line MLS with angles between them, the

TABLE 2.2: Comparison of the Point Cloud Generation Sensors

Sensor Type	Working Principle	Passive / Active	H. FOV (*)	V. FOV(*)	Range	Accuracy
TLS (static)	ToF	Active	360 °	135 °	× 100 m	< 1 cm
TLS (profiler)	ToF	Active	N/A	135 °	× 100 m	< 1 cm
MLS (single-line)	ToF	Active	360 °	N/A	× 100 m	~ 3 cm
MLS (multi-line)	ToF	Active	360 °	30-40 °	~ 100 m	~ 3-5 cm
Stereo Camera	Close-range Photogrammetry	Passive	w.r.t focal lengths	w.r.t focal lengths	< 10 m	w.r.t. distance (10 cm at 4 m)
Depth Camera	ToF / Triangulation	Active	< 90 °	< 60 °	< 10 m	w.r.t. distance (3.85 cm at 3.5 m)

(∗)Note: The range for horizontal FOV (H. FOV) is 0 ° - 360 ° and for vertical FOV (V. FOV) is 0 ° - 180 °.

system would be capable of recovering 360 °+ 360 ° FOV, providing much higher reliability for 3D SLAM in building interiors.

In our dedicated system developed for indoor mobile mapping, two multiple-scanline MLS are installed as the main point cloud generating sensors to capture instant 3D point clouds used for 3D SLAM. Features extraction methods and the algorithms are discussed in *Chapter 3*, while the point cloud alignment strategies and algorithms based on such features for 3D IMU-free SLAM are presented in *Chapter 4*.

2.1.2 Positioning and Navigation Sensors and Technologies

Various technologies can be used to facilitate positioning in mobile mapping applications, such as GNSS, IMU, and INS, which are the most commonly used positioning technologies. Meanwhile, SLAM has been utilized as one of the most popular software solutions for generating trajectory and orientations.

2.1.2.1 Hardware Sensors

In outdoor mobile mapping based on vehicle platforms, the combination of GNSS, IMU, and a wheel-side odometer (*Figure 2.4*), is mandatory for applications in complex city environments. The kinematic positioning is required as well to provide centimeter-level real-time positioning accuracy. It is similar to Position and Orientation System (POS), which is the integration of GNSS and IMU, and enables the applications of airborne laser scanner technique worldwide (Shan & Toth, 2018). However, an extra odometer is provided to correct the IMU drift while GNSS solutions are unavailable, which barely happens in airborne scenarios.

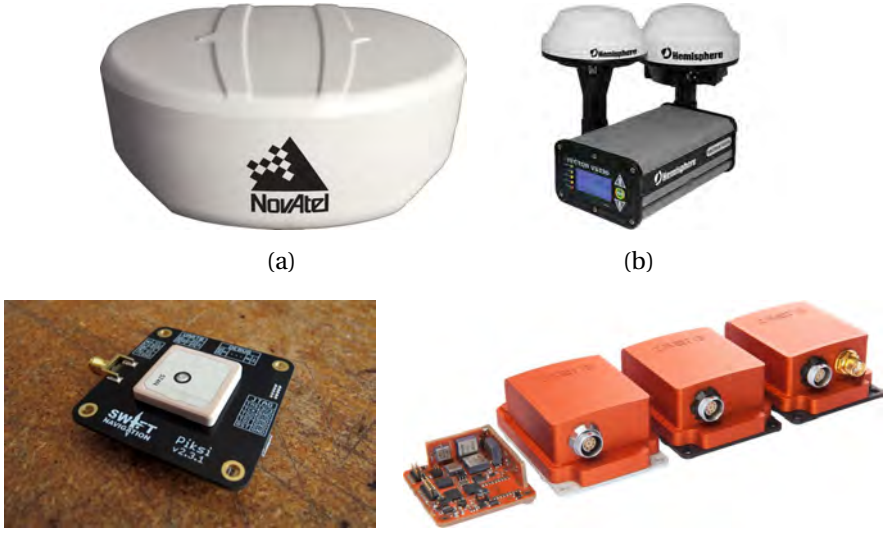
The GNSS receiver is responsible for solving the positions of the platform in 1 s interval by resectioning the satellites on the orbits. When GNSS signals are not reliable enough, i.e., due to multi-path noises and blockage of surrounding high-rise buildings, IMU data, consisting of accelerations and orientation measurements, are used to calculate movements and attitudes from the signal-loss position using DR method. As a data processing method based on high-frequency measurements and

second-order integral, drifts accumulate with the increasing distance. Therefore, the odometer is used to provide one-dimension measurements to derive the speed of the platform and correct the differences in first-order integral resulting from the small errors between the real acceleration and the measurement values (Georgy, Karamat, Iqbal, & Noureldin, 2011). In the meantime, the high-frequency IMU also enables the positioning and orientation of the platform between GNSS positioning intervals, rather than calculating the differences from Doppler speed measurements and moving directions on the trajectory.

FIGURE 2.4: Applanix POS LV system for positioning and heading applications

As shown in *Figure 2.5(a)*, it is a normal-sized antenna for survey-grade GNSS applications, while the sensor shown in *Figure 2.5(b)* is a multi-antenna GNSS system which can output both the positions and headings of the platform. In some mobile mapping solutions, either backpack solutions or vehicle-based solutions, multi-GNSS systems are used as both positioning and heading sensors, such as 3D Laser Mapping ROBIN system by 3D Laser Mapping, shown in *Figure 2.6(a)*, and MMS-A by Mitsubishi, shown in *Figure 2.6(b)*. When at least three antennas are provided, the baselines between antennas can be used to calculate the three-axis attitude of the installation platform as the Mitsubishi solution. *Figure 2.5(c)* shows a dedicated

GNSS module, which consists of the compact antenna and the receiver board module on the reverse side, designed for handheld devices. *Figure 2.5(d)* shows a MEMS IMU intended for mobile mapping applications. Conventional IMU is quite heavy and large. Although adopted in some of the solutions, especially vehicle-based solutions that are not sensitive to weights and power consumption, it is believed they will be replaced by MEMS sensors and algorithms in the future.



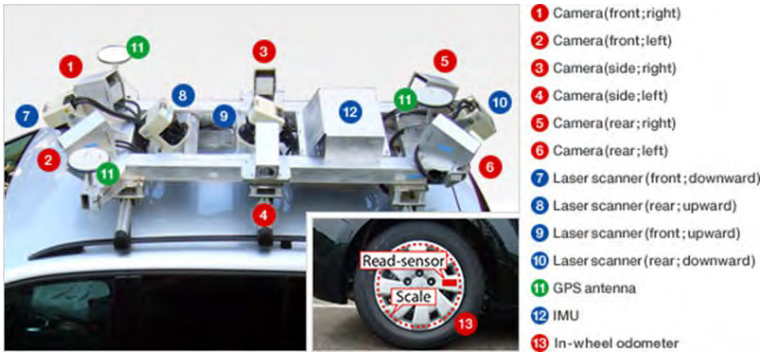
(a) (b)

(c) (d)

FIGURE 2.5: Popular positioning and navigation sensors for mobile mapping. (a) NovAtel GNSS antenna © NovAtel Copyright 2016. (b) Hemisphere multi-antenna GNSS © Hemisphere Copyright 2016. (c) Swift Piksi GNSS module © Swift Copyright 2016. (d) Series of MEMS IMU, Attitude and Heading Reference System and GPS/INS products from Xsens © Xsens Copyright 2016.

However, since GNSS receivers can barely output precise positions in indoor environments, the IMU and its combination with SLAM algorithms would be a better choice for indoor mobile mapping. Alternatively, if static or stop-and-go solutions are adopted, total stations and other conventional survey instruments can be used to obtain the accurate positions of the survey stations and the registration targets.

(a)



(b)

FIGURE 2.6: Mobile mapping solutions utilizing multiple GNSS antenna for orientating. (a) 3D Laser Mapping ROBIN system © 3D Laser Mapping Copyright 2019. (b) Mitsubishi MMS-A © Mitsubishi Copyright 2019.

2.1.2.2 SLAM

SLAM is the most popular solution adopted in scenarios where GNSS signals are no longer available, such as city canyons and indoor environments. Depending on the data used, there are four kinds of SLAM methods—LiDAR SLAM, vSLAM, RGB-D SLAM, and integrated SLAM—in which both image and LiDAR data will be used in the positioning and mapping process. Despite the differences in source data, the principles of these four SLAM methods are the same. They are all based on

the registration and matching between data frames or the alignment between the features and the local maps. By calculating the respective position and orientation changes of the survey platform, the geo-referencing relationship between neighboring frames and key frames can be estimated. Through these frame-to-frame or frame-to-map alignments, an estimated relationship network can be built and adjusted, producing the estimated pose and position of every single frame. In most algorithms, IMU data are thought to eliminate the lose-lock errors while the SLAM results restrict the drifting of the SLAM trajectory. A SLAM processing result example is shown in *Figure 2.7*, whose planes are adopted as features for alignment.

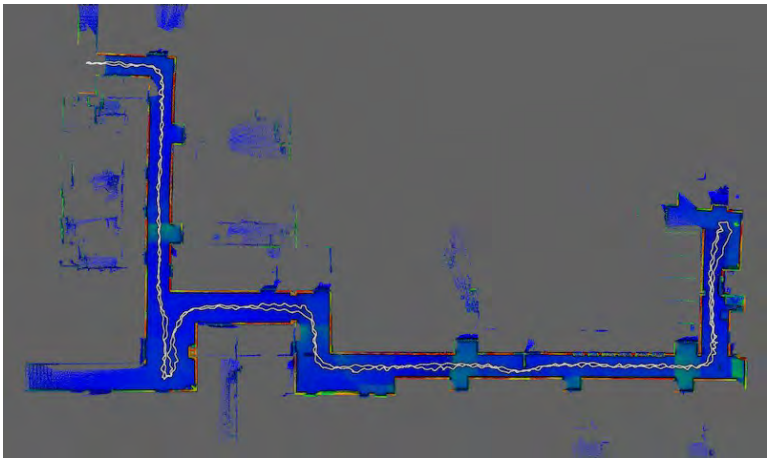


FIGURE 2.7: SLAM processing result example showing the trajectory in white and the point cloud of the floor in blue

The SLAM processing method can be adopted for both outdoor and indoor environments: its most significant advantage is that it can provide reliable positioning information in GNSS-denied environments. Most of the indoor mobile mapping solutions nowadays are based on their integration with IMU. The detailed workflows and techniques of SLAM are reviewed in *Section 2.5*.

2.1.3 Attitude Determination Sensors and Technologies

Attitude determination methods are in two kinds: direct and indirect methods. In direct methods, the corresponding sensors can output real-time attitude information of the platform by analyzing the difference between the heading direction and the magnetic north, and the angle between its downward direction and the direction of the gravity. Such sensors include, but are not limited to, 2D/3D compass, IMU, inclinometer, gyroscopes, and Attitude and Heading Reference System (AHRS). With the developing of MEMS technologies, such sensors, which used to be heavy and large, are now small and easy to integrate with other sensors, as shown in *Figure 2.5(d)*.

The second kind of attitude determination technology is based on indirect methods. Such sensors calculate the estimated attitude information based on kinematic positionings, such as the one shown in *Figure 2.5(b)*. For example, the attitude determination device based on dual GNSS antenna calculates the single-direction orientation of the platform based on the positioning differences with the known baseline length as a constraint. Meanwhile, multi-antenna devices with more than three antennas can output pitch, roll, and yaw information at the same time based on the comparison between the orientations of the sensor pointing at the grid axes.

Another example is the attitude determination method, which can estimate the attitude differences between two epochs in 3D scenarios or the heading variations in 2D scenes. Therefore, given the initial attitude information, the kinematic attitude

determination can be derived. However, in such examples, the sensors involved are the point cloud capturing sensors or the image sensors, which blurs the boundary between the categories of point cloud and image capturing sensors and the dedicated attitude determination sensors.

2.1.4 Discussion

As listed in previous sections, various sensors are implemented for mobile mapping applications. Typically, a complete mobile mapping system consists of a positioning sensor, an attitude determination sensor, and at least one point-cloud capturing sensor. In some of the solutions, a camera, either a normal camera or a panoramic camera, is included for capturing images of the surrounding environments, which could be used for both point cloud colorization and image documentation. For such applications, panoramic cameras are better than traditional cameras as they are capable of providing 360 ° FOV.

In some solutions, the boundary between the three categories of sensors could be blurred. For instance, the laser scanners, which are typically point cloud capturing sensors, are used to provide point clouds to determine position and attitude in the SLAM process instead of the dedicated sensors for localization and orientating. Meanwhile, sensors are designed to be alternatives of other sensors to enhance the reliability of the whole system, such as the adoption of IMU in outdoor POS.

Based on this fundamental structure, indoor mobile mapping systems were designed to meeting the needs of various GNSS-denied scenarios, which are introduced in the next section.

2.2 State-of-the-art Indoor Mobile Mapping Solutions

Researchers and commercial suppliers have introduced a variety of mobile mapping solutions for indoor environments. Brief introductions of these state-of-the-art systems are listed in this section.

The floorplan-only solutions are a product of outdated technology and are no longer evolving presently. However, as some of them are also based on sensors installed on mobile platforms, the data processing algorithms and workflow can still be referred to in state-of-the-art solutions. Therefore, they will be first presented, followed by the introductions of semi-mobile and fully mobile solutions.

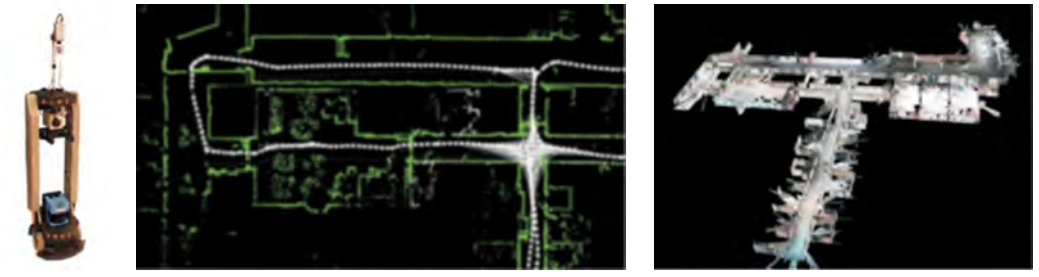
Meanwhile, solutions based on data fusion of multiple scan stations are fully static solutions. The registration and point cloud alignments are based on common features and targets extracted from overlapping areas. They will not be discussed here either.

2.2.1 Floorplan-only Solutions

A few indoor solutions were introduced in earlier years based on the assumption that there are only floors, ceilings, and walls in indoor environments, generating the

surface model of the main infrastructures, which are similar to the 2.5D representation of the floorplan. The data capturing and processing workflow is then converted to the process of intersection identification and point cloud classification.

Biber, Andreasson, Duckett, and Schilling (2004) introduced a floorplan-only solution based on an integrated pushcart system. In the system, there was a 2D laser scanner facing forward, as shown in *Figure 2.8(a)*, used for extracting positions of wall features and determining the position and orientation changes. The floorplan could, therefore, be generated using 2D point clouds of assumed vertical wall features, as shown in *Figure 2.8(b)*. There was a panoramic camera on the top of the trolley, capturing 360-degree images of the surrounding area. Based on the hypothesis that all vertical wall boundaries were perpendicular to the upper and lower borders of the image, and the intersection extracted based on image texture, the regions of the wall surfaces were identified. The texture patches were then attached directly to the wall surface models to generate 2.5D models, as shown in *Figure 2.8(c)*.



(a) (b) (c)

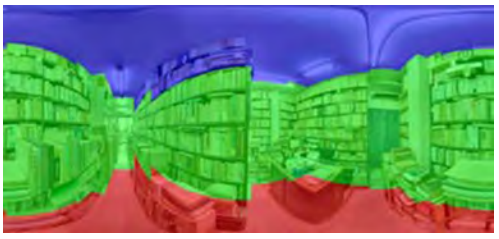
FIGURE 2.8: A floorplan-only solution creating floorplans and 2.5D models. (a) Data acquisition platform. (b) Floorplan based on 2D point clouds. (c) Generated 2.5D models. (Biber, Andreasson, Duckett, & Schilling, 2004)

Cabral and Furukawa (2014) described a piecewise planar and compact floorplan reconstruction method based on SfM and multi-view stereo. Given images as shown in *Figure 2.9(a)*, a novel structure classification was firstly performed to categorize all pixels into three regions—floor, ceiling, and wall—as shown in *Figure 2.9(b)*. Then the floorplan reconstruction problem was defined as the shortest path problem based on piecewise planarity. The geometry of the models was recovered using tied 3D points, as shown in *Figure 2.9(c)*. A significant limitation of this method was that all features, including furniture and decorations, are stitched to the main infrastructure planes as textures. Such omissions resulted from the limited locations for capturing images while they could not be entirely avoided by adding observing stations due to the complexity of indoor environments. It not only removed the indoor objects which might be of great interest but also created wrong textures for the main infrastructure.

The two solutions above were just representative floorplan-only solutions. They concentrated on getting the main structural features in artificial environments rather than capturing all intricate shapes and variations. Generally, data acquisition and processing time was much less than the detail capturing solutions. Their pros were quite obvious:

- The data volume obtained was much smaller since there was neither requirement on additional attitude determination sensors, nor dense 3D point clouds.

(a)



(b)

(c)

FIGURE 2.9: A panoramic image based floorplan-only solution and the 2.5D model created. (a) Panoramic image acquired. (b) Classified panoramic image. (c) Generated 2.5D models. (Cabral & Furukawa, 2014)

- The final outputs concentrated on the floorplans and the 2.5D models of the surrounding environments, which was the information on the underlying infrastructure itself. There was no data removal work required.

However, the limitation of such methods was the omission of the detailed information on surfaces, such as decorations, shelf, and other surface variations, which makes it applicable only to the reconstruction of empty spaces or environments

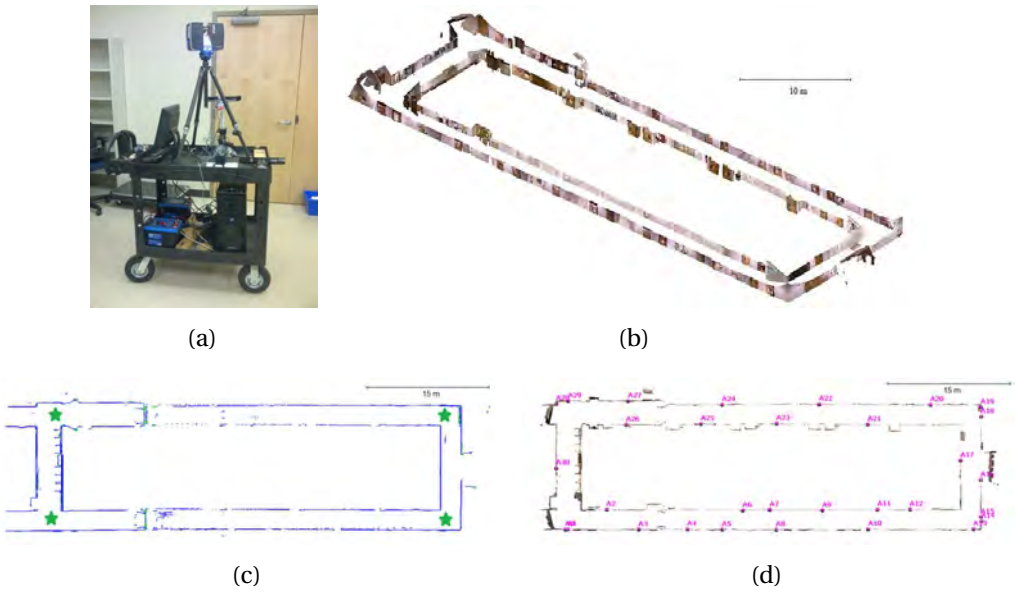
in which the detailed information can be ignored. Also, the assumption of such methods made them unreliable to work in complex environments, such as indoor environments with quite a few wall intersections obstructing objects (Cabral & Furukawa, 2014). For modern architecture with glasses and steel, floorplan-only solutions would omit most of the features as LiDAR reflections and images cannot fully represent the as-built states of the environments, while the indoor infrastructure is much more complicated as well.

2.2.2 Semi-mobile Solutions

As interim workflow before fully mobile solutions were introduced, semi-mobile solutions were a perfect alternative to the station-by-station workflow as they decrease the time used between stations while maintaining the accuracy of fully static solutions. The concept is based on the simple consideration of installing TLS on a tripod, which was fixed on a moving platform, such as a trolley, to reduce the time used for moving and installing TLS at different scan stations.

In addition to the solutions adopted by survey companies, Chow (2014) verified the possibilities and accuracy using such stop-and-go method with the integration of depth-camera-based SLAM algorithms. In this application, the TLS was installed on a pushcart, together with two self-calibrated Kinect depth cameras, a medium-precision IMU (Xsens MTi-30 IMU) and a desktop computer, as shown in *Figure 2.10(a)*. The Kinect sensors were responsible for capturing wall surface data, including both texture and point clouds, for generating another model, as shown in *Figure*

2.10(b). When either the TLS or the Kinect sensor was capturing data, the pushcart was kept still, as in a full *stop* mode. The whole trolley was moved to the next station after finishing one scan station, which is the *go* mode. Since the valid working range of the RGB-D sensor was much smaller than the TLS, the distance between RGB-D scan stations was much shorter than the distance between those of TLS, as indicated in *Figure 2.10(c)* and *2.10(d)*. After capturing all data at scan stations around the *trajectory*, the point clouds were registered into the same coordinate frame with the assistance of the IMU and mesh-based SLAM algorithm to identify the position changes between stations.



(a) (b)

(c) (d)

FIGURE 2.10: A semi-mobile solution combining TLS and RGB-D sensors on a trolley. (a) The installation of the data acquisition platform. (b) The wall surface model generated by Kinect sensor. (c) The floorplan generated using TLS point cloud fusion with the green stars indicating the survey stations of TLS. (d) The floorplan generated using Kinect mesh fusion with the purple dots indicating the survey stations of Kinect. (Chow, 2014)

The semi-mobile solution was a practical alternative for rapid data acquisition

in indoor environments. For stop-and-go workflow with TLS, much higher accuracy could be achieved if proper scan targets were provided as control points for point cloud registration to enable the system to work without IMU. ICP algorithms and reasonable distribution of features would also offer the possibility of working without scan targets, as they were just variants of the fully static workflow. Meanwhile, Chow (2014) had verified that the accuracy of indoor mapping using Kinect sensors with SLAM algorithm was acceptable compared with reference value based on Kinect with IMU sensors.

As the moving regions of such systems were restricted by the movement of the platform, which was a wheeled trolley, the valid working range of such designs could only be planar floor areas rather than staircases and slopes. The valid working range of Kinect sensor was limited as well, which made this mobile mapping solution practical only in cramped environments such as hallways and aisles. Furthermore, as only the scan station installation time was waived in the working flow, the total working time was not reduced too much.

2.2.3 Fully Mobile Solutions

To meet the increasing demands of rapid data acquisition, the fully mobile solutions in the form of integrated trolleys and backpacks were introduced by many scholars, researchers, and commercial organizations.

One of the most significant differences between these solutions was the installation platform. Since various sensors need to be installed on the same rigid frame

to make their geometrical relationship stable and measurable, most solutions were based on either trolleys or backpacks, enabling the seamless mobility of the sensor packages. The systems based on trolleys and pushcarts had restricted their moving regions since they could not be moved upstairs or downstairs. Meanwhile, floor changing via lifts was not a very reasonable choice since the movement inside the elevator would be derived using the measurements captured by the integrated IMU only. Otherwise, external control points would be required. However, the wheeled design would make more payload available, such as high-performance computers, high-precision IMU, and large-volume batteries. Although the moving region of the backpack design was more flexible than the trolleys, the moving pattern of the platform would be much more complicated since there would be continuous movement in the vertical direction, and the movement were difficult to estimate precisely.

Consequently, more companies were considering mobile mapping trolleys as a temporary solution before satisfied mobile mapping backpacks could be launched. The moving ranges of such backpack solutions were much more flexible compared with the trolleys mentioned just mentioned. Nevertheless, as the movement of the human body, the carrier of the platform, should be precisely modeled for reducing drifts and errors in IMU data processing, only a few solutions could achieve accuracy better than 0.5% of the trajectory length in the translation errors (Geiger, Lenz, & Urtasun, 2012).

Various solutions have been introduced in the past few years. Most were based on portable sensors, including stereo cameras, RGB-D sensors, and laser scanners.

They can be categorized into the following four kinds based on the point cloud acquisition sensors adopted.

2.2.3.1 Optical-sensor-based Solutions

Optical-sensor-based solutions were almost ideal choices for outdoor mobile mapping since they could not only collect RGB information of surrounding environments but also generate stereo image pairs by using sequential images to provide geometric data (Paz, Pinies, Tardos, & Neira, 2008). However, in indoor applications, the absence of GNSS signals made the high-accuracy positioning data unavailable, which made it much more challenging to generate stereo image pairs from sequential images. Although scholars were working on recovering scales from monocular vSLAM results using IMU trajectory, the accuracy was still greatly affected by the drifting error of the IMU (Lupton & Sukkarieh, 2008) (Nützi, Weiss, Scaramuzza, & Siegwart, 2011) (Concha, Loianno, Kumar, & Civera, 2016). Additionally, the absence of texture features in images, which was quite common in indoor environments, especially in corridors and empty rooms, led to both tracking loss problems and empty spaces on object surfaces. Lastly, the high cost of computing accurate and dense stereo images also prevent the technique's wide application (Jennings & Murray, 1997). Therefore, not too many solutions adopted the optical-sensor-only design for indoor mobile mapping.

In the automatic robot control field, Jennings and Murray (1997) introduced a trinocular-sensor-based image acquisition system for detecting obstacles, path

planning, and mobile mapping. The depth image generated was 128×120 in resolution with 20 disparities and computed in 350 ms using dedicated high speed Digital Signal Processors (DSP), which was relative high-performance in 1997. However, the limitation of high computational cost made the data acquisition rate lower than 2 Hz, which was not practical for applications.

Paz et al. (2008) introduced a binocular sensor, i.e., a FLIR Bumblebee 2, to recover the geometric relationship and texture information of objects. However, the accuracy of points with distances larger than 10 m was still with considerable uncertainties due to the limited baseline length between the two optical sensors. Moreover, the huge computational cost was also a critical concern for mobile mapping solutions.

Only a few indoor mobile mapping solutions which were based on optical sensors only as data acquisition sensors owing to the enormous computation load and the data uncertainty determined by the separated level of the binocular sensors. Another problem of not adopting optical-sensor-only design would be the inability to identify common points on surfaces without too much texture, which was quite common in indoor environments. However, such solutions were the simplest indoor mapping solution in most scenarios. Considering the popularity of binocular and trinocular camera systems implemented on smartphones in present days, this method was an excellent choice for navigation-grade applications, such as indoor localization and pedestrian navigation.

2.2.3.2 Depth-camera-based Solutions

The depth camera can produce RGB-D images that can be used for geometric relationship recovery and texture data acquisition, meaning it is possible to implement vSLAM and point cloud SLAM. Several solutions were based on depth camera, such as Google's Project Tango, which was a series of tablet- or smartphone-based solutions with integrated RGB-D sensors to produced 3D reconstructions of the real world and the attitude change information based on optical and depth images, as shown in *Figure 2.11* (Google, 2015). The most significant advantage of Project Tango devices was that by integrating various sensors into the same small-size handheld device, system developers were capable of developing embedded algorithms to fuse corresponding data from different sensors (depth images, RGB images, and IMU measurements) to generate a trajectory estimation or local maps. Unfortunately, Google officially shut down the project on March 01, 2018, making this kind of solutions unavailable.

(a) (b)

FIGURE 2.11: Google's Project Tango tablet and demo simulation. (a) Project Tango Developer Toolkit. (b) RGB-D image simulation. © Google Copyright 2015

Microsoft Kinect sensors were also widely used in mapping indoor environments. As analyzed by Huai, Zhang, and Yilmaz (2015), because the working range,

resolution, and accuracy of the RGB-D were affected by the distances between the sensor and the objects, a coarse-to-fine iterative ICP process needed to be performed together with Scale Invariant Feature Transform (SIFT) visual odometry and IMU to extract dense point clouds of objects. Similar indoor mapping solutions were also introduced by Henry, Krainin, Herbst, Ren, and Fox (2012), Tsai et al. (2015), and S. Tang et al. (2016) owing to the much lower prices of depth cameras compared with MLS, and the much lower computational cost compared with conventional binocular sensors. Such solutions could work either with or without an onboard IMU, while the IMU could reduce the computational load and enhance the overall reliability of the system.

However, most depth cameras required calibration before data acquisition since they were initially designed for entertainment purposes rather than surveying applications, which also made this topic an important research area (Darwish et al., 2017). Another disadvantage restricting their application area was the limited effective working range of only a few meters, which made it tedious for capturing 3D GIS data of a large room as the operator needed to approach the main structure to capture the data. S. Tang et al. (2016) and Darwish et al. (2017) introduced a method based on online calibration between the depth camera and the RGB sensor, increasing the valid working range of such sensors. Based on the precise estimation using a short-range depth camera, the exterior orientation of the RGB image can, therefore, be determined by utilizing the parameters derived from the online calibration. However, the results showed that the proposed workflow was still more useful for

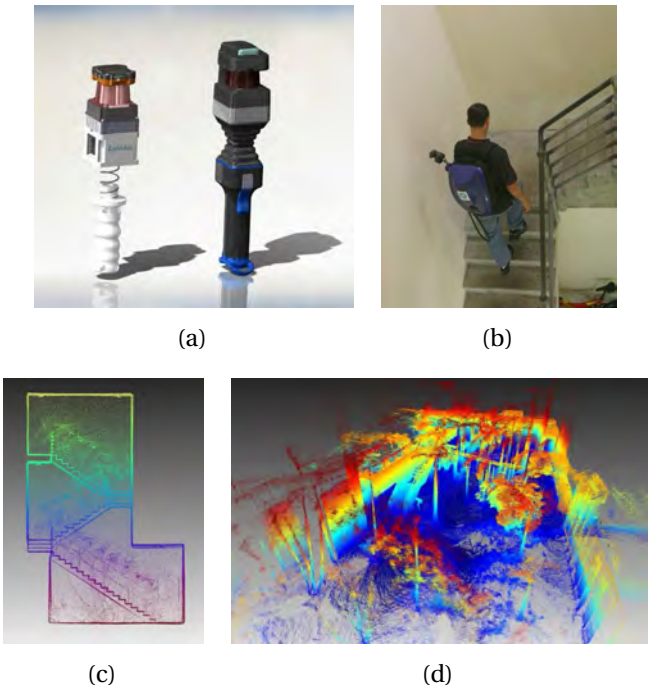
mapping small-size environments as drifting cannot be ignored in large spaces.

2.2.3.3 LiDAR-based Solutions

LiDAR-only solutions focused on the lightweight solutions for capturing 2D or 3D geometric data of the surrounding environments. Multiple solutions were introduced including the ZEB series by Bosse, Zlot, and Flick (2012) and GeoSLAM (2019); the iMS 2D by ViAmetris 3D Mapping (2015a); a backpack solution by Nüchter, Bormann, Koch, Kühn, and May (2015); and the LiBackpack DG50 by GreenValley International (2019). These solutions are based on different working principles.

As shown in *Figure 2.12*, Zebedee or ZEB1 was a LiDAR-IMU integrated solution developed by GeoSLAM. The laser scanner and IMU were installed together as a moving part and swung against the handler. Thus, in the constant moving process, the velocity drifts were modeled by utilizing the zero-velocity moments. The system could be installed on various platforms, including backpacks, handheld, and vehicles, since the only requirement was maintaining the swinging movements while it was working. The IMU mounted inside Zebedee was industrial-grade MEMS MicroStrain 3DM-GX2 with high output rate and large measurement range (Bosse et al., 2012). Subsequently, GeoSLAM released its optimized products. The successors include ZEB Revo, ZEB Revo RT, and ZEB Horizon (GeoSLAM, 2019). As shown in *Figure 2.13*, the scanner and IMU kept rotating together against the installation axis while working. Although not explicitly explained, the change of moving mechanism

was capable of providing relative attitude via the embedded angular encoder to calibrate IMU drifts.



(a) (b)

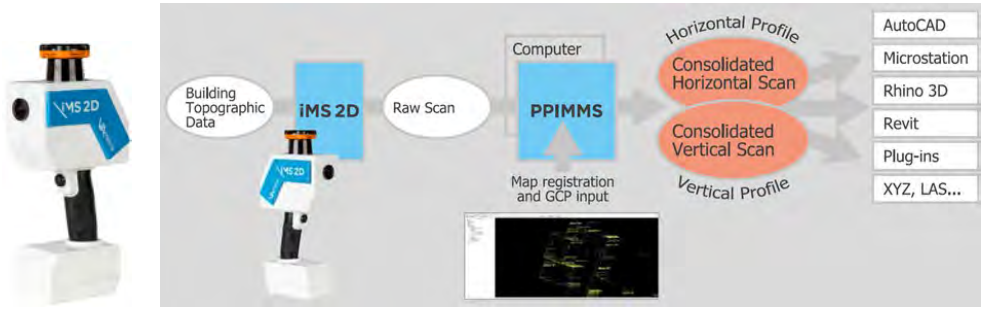
(c) (d)

FIGURE 2.12: ZEB1 handheld solution provided by GeoSLAM and sample data. (a) The design of 1st and 2nd generation of Zebedee. (b) Zebedee working in backpack mode. (c) The profile view of point clouds acquired with Zebedee mounted on the backpack (close-loop). (d) SLAM results from the outdoor courtyard environment. (Bosse, Zlot, & Flick, 2012)

(a) (b) (c)

FIGURE 2.13: GeoSLAM new product line. (a) ZEB Revo containing a single-line Hokuyo scanner, an IMU, a rotation structure, and a handler. (b) ZEB Revo RT which is an integration of ZEB Revo and a tablet controller on the handler. (c) ZEB Horizon consists of a Velodyne 16-line scanner, an IMU, a rotation structure, and a handler. © GeoSLAM Copyright 2019

Based on the 2D SLAM with 6 Degree-of-Freedom (DOF) movement, iMS 2D worked with the help of IMU because there were still attitude variations when the operator was holding the instrument and moving around, as shown in *Figure 2.14*. The system relied on the 200Hz high-accuracy AHRS IMU to provide precise attitude and DR positioning results, while the 2D SLAM results offered positioning variations in the 2D plane (ViAmetris 3D Mapping, 2015a). The system worked on the assumption that a satisfying indoor positioning and orientation result, a 2D floorplan, and the corresponding video stream could be provided by limiting and calibrating the IMU drifts in the horizontal plane using 2D SLAM results. Therefore, the integrated panoramic camera could only provide images used for documentary rather than for generating texture data, which makes the only geometric output of iMS 2D either the 2D horizontal or the vertical profile of the surrounding environments. Moreover, it might be theoretically practical to integrate the algorithm introduced by Biber et al. (2004) to generate surface models.



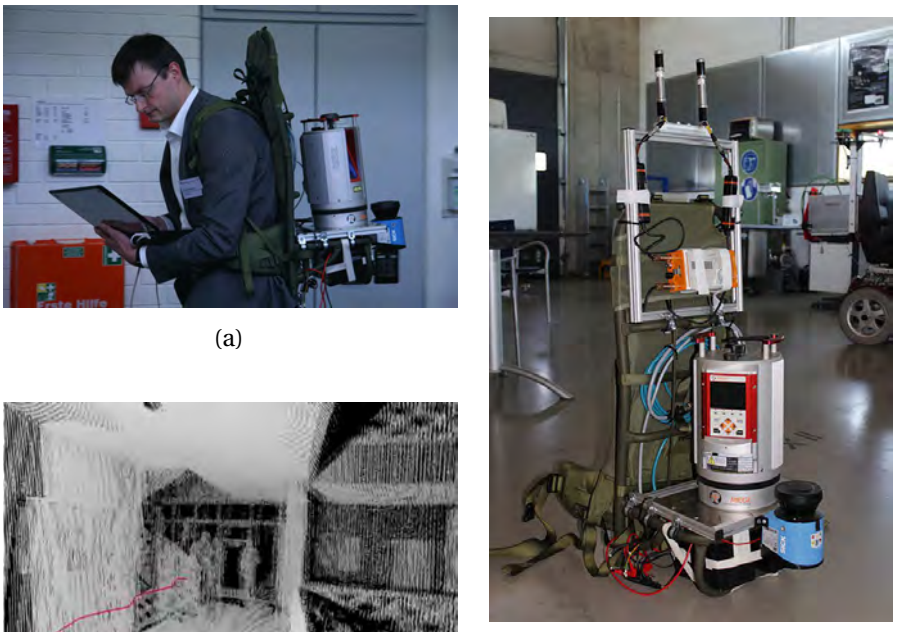
(a) (b)

FIGURE 2.14: iMS 2D scanner and its data processing workflow. (a) The compact 2D scanner with panoramic camera and AHRS installed. (b) The working flowchart of iMS 2D. © ViAmetris Copyright 2015

The solution introduced by Nüchter et al. (2015) can work without any IMU using dense 3D point cloud and 3D SLAM algorithm, as shown in *Figure 2.15*. It is based on a SICK LMS-100 2D LiDAR scanning 270 ° at 25 Hz and a RIEGL VZ-400 3D LiDAR scanning 100 ° (V) × 260 ° (H) area at the update rate of lower than 1 Hz. The solution worked based on a two-step registration process to achieve precise SLAM without the assistance of IMU. The first matching process based on RANdom SAmple Consensus (RANSAC) identified point pairs with similar distances in the scenes and aligned the 2D point cloud roughly, while the subsequent ICP process refined the alignment as it could not conduct on point clouds with large rotation differences (Nüchter et al., 2015). It should be noted that an extra camera can be integrated with VZ-400, which was originally designed as a TLS, to provide texture acquisition capability (RIEGL Laser Measurement Systems GmbH, 2015). Nevertheless, this was not verified in their solution.

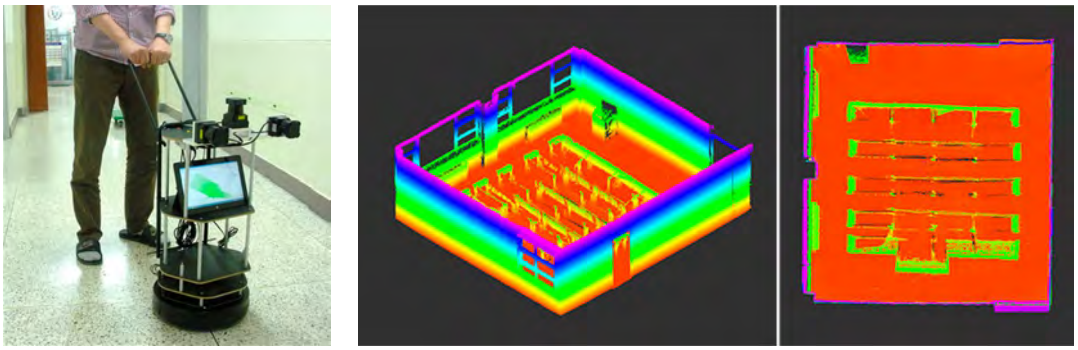
Jung, Yoon, Ju, and Heo (2015) delivered a pushcart-based triple 2D MLS solution under the constraints that all features were parallel or perpendicular and the data acquisition process was captured in a closed loop, as shown in *Figure 2.16*. However, this solution focused more on the planar indoor environment mapping, which required more work to fit real applications of 3D GIS data acquisition.

Google released its exploring project in this area named Cartographer in 2015, as shown in *Figure 2.17*, which was constructed based on dual 270 ° 2D LiDAR and at least one IMU on the backpack (Lardinois, 2014). In October 2016, Google made this project open source to share their achievements and encourage scholars to work in



(a) (b) (c)

FIGURE 2.15: A backpack-mounted TLS solution integrated with 2D LiDAR and its sample data. (a) The photo of operating the backpack system. (b) The hardware components. (c) The 3D point cloud output. (Nüchter, Borrmann, Koch, Kühn, & May, 2015)

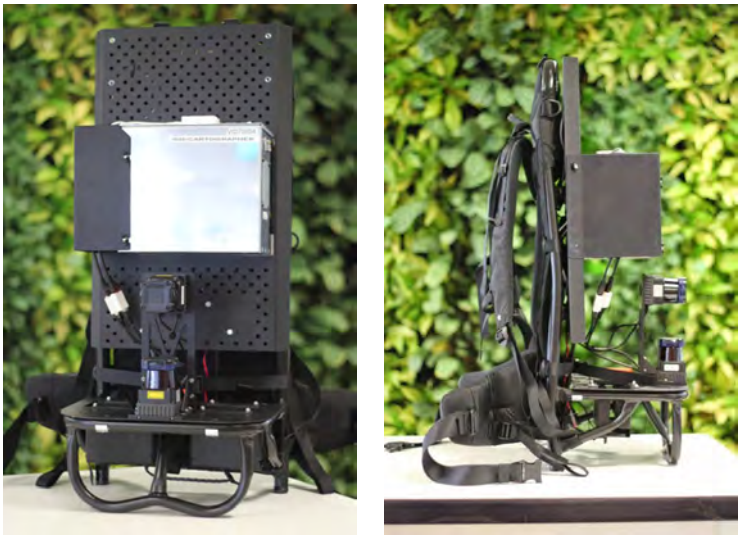


(a) (b)

FIGURE 2.16: A pushcart-based indoor mapping solution with constraints of perpendicular planes and its sample data. (a) The indoor mapping system. (b) The generated point cloud of a classroom. (Jung, Yoon, Ju, & Heo, 2015)

this area, and it released little information afterwards (Google, 2016). Although not installed in any of their public released demo versions, multi-line scanners, such as

Velodyne VLP-16, were also supported in their source codes.



(a) (b)

FIGURE 2.17: The rear view and side view of Google Cartographer. (a) The rear view of the backpack standing on the frame itself. (b) The side view showing two perpendicular single-line laser scanners. (Lardinois, 2014)

As one of the major global LiDAR system vendors, Green Valley International released several versions of their mobile mapping backpack. Recently, they announced their updated version of products. These were the LiBackpack D50 and DG50, as shown in *Figure 2.18*, designed with two Velodyne VLP-16 Lite scanners and an IMU inside the hardware frame (GreenValley International, 2019). Besides, the DG version was installed with an extra GNSS receiver, obtaining the capability of working outdoor and all data aligned to the global reference frames. Furthermore, the GNSS results could provide extra high-accuracy control points anywhere when valid GNSS solutions could be realized.

The most significant limitation of such LiDAR-only solutions was that the attitude determination devices, both the IMU and the 3D LiDAR, needed to have the

(a) (b)

FIGURE 2.18: The GVI LiBackpack D50 and DG50 dual scanner mobile mapping backpack. (a) The side view of D50 with two multi-line laser scanner and embedded IMU. (b) The rear view of DG50 with an additional GNSS receiver compared with D50. The tablet, Microsoft Surface Pro, is connected to the internal processing unit via a wireless connection to control the unit. © Green Valley International Copyright 2019

ability of high-rate data output, which made the cost of such devices much higher than other solutions. Furthermore, such solutions could only generate 2D or 3D point cloud data without texture information because there were no imagery sensor on board. Although the intensity of points could be used to represent material changes, such characteristics can be easily affected by the changes in incident angles of laser beams, which made it more difficult as the laser scanner is continuously moving around. Last but not least, as most laser scanners adopted were designed for outdoor autonomous driving applications, their distance measurement accuracy was around 2-5 cm, making the overall accuracy of the point clouds barely possible for high-accuracy applications, such as millimeter-level BIM generation.

2.2.3.4 Integrated Solutions

Most of the commercialized solutions and some of the research outcomes can be grouped as integrated solutions, which integrate multiple kinds of data acquisition sensors, which include, but are not limited to, single or multiple 2D or 3D MLS or TLS, single panoramic camera or multiple fisheye cameras, IMU, and AHRS. Their installation frame could be in the forms of either wheeled trolley or backpack. Among all trolley-based solutions, Sanborn Platform for Indoor Mapping (SPIN), TiMMS, and iMS 3D were the three major products available on the market in 2015.

SPIN was a commercial solution based on a robot platform with onboard 3-axis digital gyroscope, odometer, two 2D MLS, and a Kinect depth camera, as shown in *Figure 2.19(a)* (The Sanborn Map Company Inc., 2015). In the system, the horizontal LiDAR with longer detection range was responsible for the 2D SLAM positioning with help from the gyroscope and the odometer, while the vertical LiDAR with much shorter working distance acted as the main point cloud generator. The Kinect sensor, installed on the lower part facing forward, captured RGB-D images, to provide texture information with the help of depth data for registration and data fusion. The platform was manipulated via a joystick while moving was motorized by the robot platform to achieve higher smoothness and robustness in the data acquisition process. The system was released in 2014, but the information has hardly been updated in recent years. In addition to the obvious limitations in size, weight, and moving speed, the resolution of the texture data captured by Kinect sensor might also be a little coarse, which made the coloring of the point clouds and the extraction of

surface texture to contain more uncertainty.

(a) (b) (c)

FIGURE 2.19: Commercialized pushcart solutions. (a) SPIN © The Sanborn Map Company Inc. Copyright 2015. (b) TiMMS © Applanix Corp. Copyright 2015. (c) iMS 3D © ViAmetris Copyright 2015.

The TiMMS (Gen. 2, shown in *Figure 2.19(b)*) provided by Applanix, a Trimble company, and the iMS 3D (shown in *Figure 2.19(c)*) provided by ViAmetris were similar since they were using the same Ladybug 3 panoramic camera to collect texture information (ViAmetris 3D Mapping, 2015b) (Applanix Corp., 2015). However, the TiMMS was an IMU-only solution based on POS while the integrated 3D TLS worked in 2D profile mode most of the time. In the meantime, the TiMMS required predefined GCP to correct drift error of the IMU. On the other hand, the iMS 3D was a much more comprehensive solution based on 2D SLAM and IMU. The dual vertical 2D scanners collected dense point clouds, collaborating with panoramic images to generate colorized points and surface models.

With the development of algorithms and sensors, trolley-based systems became less available as the limited moving range narrowed down the application fields.

NavVis was one of the companies insisting on providing pushcart solutions (NavVis US Inc., 2019). As shown in *Figure 2.20(a)*, a Velodyne multi-line scanner was installed on top of the trolley, providing 6 DOF SLAM solutions. The tilted installation was designed to capture more points reflected from the ground and ceiling. Meanwhile, three single-line scanners were installed vertically with angles to capture points as profilers. In addition, five cameras were installed around the top part of the trolley, capturing 360 ° images of the environments. When moving, the upper part of the trolley could be separated from the lower wheeled platform, and both parts could be placed into cases for ease of carrying, as shown in *Figure 2.20(b)*. In scenarios such as multi-floor parking lot, slopes can be used to facilitate the movements between floors and a compact point cloud can be generated based on the utilization of IMU and 6 DOF SLAM process, as shown in *Figure 2.20(c)*.

(a) (b) (c)

FIGURE 2.20: NavVis M6 Mobile Mapping Trolley. (a) The trolley in working mode, pushed by the operator and moving around. (b) The system in carrier mode and separated in two cases. (c) The point cloud of a multi-floor parking lot generated by NavVis M6. © NavVis Copyright 2019

Another kind of rapidly developed solution was the mobile mapping backpack. Various companies and research groups released their solutions. Generally, two

multi-line laser scanners were installed, one on the top and the other in the rear. The two scanners were installed perpendicular or nearly perpendicular to each other to ensure maximum coverage of the single-frame point clouds. Most solutions were installed with Velodyne 16- or 32-line scanners, enabling the embedded precise time synchronization via Pulse-Per-Second (PPS) technology. Meanwhile, a panoramic camera consisting of multiple fisheye or wide-angle lens was installed. Such camera can either be a compact camera solution, such as Ladybug 5/5Plus and Garmin Virb 360, or a proprietary camera system with lens installed at different positions on the backpack, facing different angles. Furthermore, there was an IMU or integrated navigation system (GNSS + IMU) of reasonable accuracy, which was determined by the performance of the SLAM workflow.

As shown in *Figure 2.21*, the backpack solution introduced by Liu et al. (2010) was based on three monocular cameras and three 2D scanners, while the Pegasus: Backpack provided by Leica Geosystems AG (2015) integrated five cameras and two 3D LiDAR, and the UltraCam Panther by Vexcel Imaging GmbH (2016) was supplied with more than twenty cameras and a single 3D LiDAR.

The three solutions adopted an integrated imagery system to collect texture data. The Pegasus: Backpack and UltraCam Panther produced frame-by-frame panoramic images rather than colored points and relied on SLAM with the assistance of high-precision IMU, while a second solution provided simple surface models of planar vertical façades. The most considerable difference was that by

installing a vertical pitch LiDAR, the solution by Liu et al. (2010) could extract orientation changes in the pitch direction with higher precision. However, all three solutions cannot achieve survey-grade accuracy. The Pegasus: Backpack was currently still under development and seeking for better indoor accuracy, while the UltraCam Panther was claimed as being redesigned at the time of writing this thesis. The companies were defining such solutions as documentation tools rather than surveying instruments, considering that the professional users could still not accept such accuracy (Leica Geosystems AG, 2015) (Vexcel Imaging GmbH, 2016).

In recent years, more companies have embarked on excursions in this field by releasing their mobile mapping backpacks. As shown in *Figure 2.21(d)-2.21(f)*, such brands and models included, but were not limited to, iScan-P by Hi-Target, bMS3D LD5+ by ViAmetris, and C50 by Green Valley International. Both hardware and software designs were similar in these solutions, which were using the 6 DOF SLAM and loop closure results to calibrate the drifting errors of IMU, producing high-accuracy point clouds (ViAmetris 3D Mapping, 2019). Meanwhile, some startup companies released their shrank version of such backpacks by commercializing their research outcomes in the universities, such as Kaarta products by J. Zhang and Singh (2015) from Carnegie Mellon University and Paracosm's PX-80 Handheld LiDAR from the University of Florida (Higgins, 2018). As shown in *Figure 2.22*, the most significant difference of such solutions was that the moving freedom was much more flexible compared with their predecessors as they reduce the weight significantly. Some solutions, such as Kaarta Stencil2, also provided post-processing options that trimmed

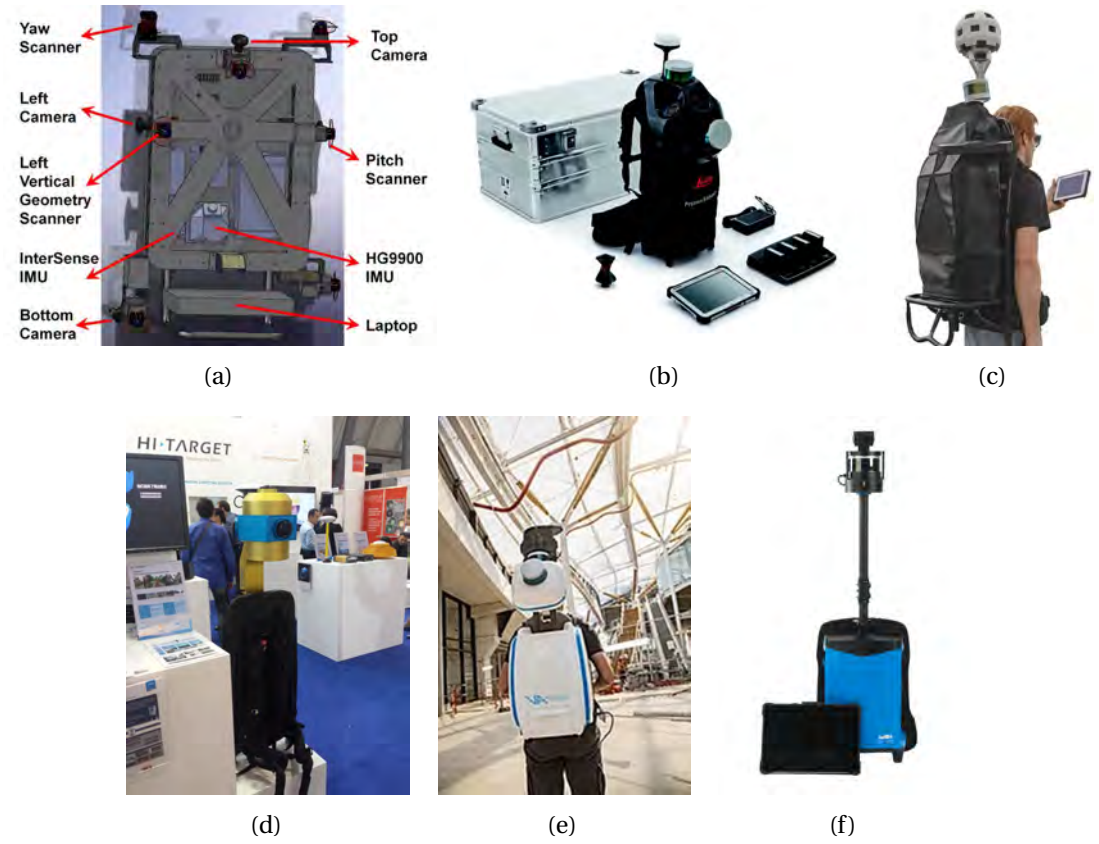


FIGURE 2.21: Backpack mobile mapping solutions. (a) The CAD drawing provided by Liu et al. (2010) with three industrial cameras. (b) Pegasus: Backpack provided by Leica with five industrial cameras © Leica Geosystems Copyright 2015. (c) Vexcel UltraCam Panther provided by Vexcel Imaging with more than twenty cameras and a single 3D laser scanner © Vexcel Imaging Copyright 2016. (d) Hi-Target iScan-P backpack demonstrating on InterGeo2015 (APOGEO, 2015). (e) bMS3D demonstration in shopping mall © ViAmetris Copyright 2019. (f) GVI C50 backpack with both a laser scanner and a panoramic camera © Green Valley International Copyright 2019.

the noisy point clouds generated to slim planes by identifying the plane first and then removing the unwanted noises automatically. This option was necessary because the SLAM algorithm can still not achieve a perfect level of accuracy so far.

Lehtola et al. (2017) and Nocerino, Menna, Remondino, Toschi, and Rodríguez-Gonzálvez (2017) compared most of the solutions available on the open market by

(a) (b)

FIGURE 2.22: Portable integrated mobile mapping solutions by startup companies. (a) Kaarta Stencil2, integrating an Intel NUC, a Velodyne VLP-16 scanner, a fisheye camera, and an IMU inside, with a tablet for control and data visualization © Kaarta Copyright 2019. (b) Paracosm PX-80 which is similar to Kaarta Stencil2 but with different installation and processing workflow © Occipital Copyright 2019.

2017 and found most of them were capable of generating point clouds of typical indoor scenarios with the vertical accuracy of tens of centimeters while the horizontal accuracy of most systems is still not satisfactory so far. However, the accuracy of such solutions is not only affected by the length of the trajectory but also by the complexity and size of the spaces.

2.3 A Summary of Indoor Mobile Mapping Sensors and System

In previous sections, relevant sensors and solutions in indoor mobile mapping applications were reviewed and discussed, and the designs and the working principles of most of the mobile mapping systems available were reviewed. In designing a

novel indoor mobile mapping system, its design and installation should be taken as references for making a reasonable combination and design of sensors.

Firstly, GNSS signals are no longer reliable or are even unavailable in indoor environments. Although direct line-of-sight (LoS) surveying using total stations can achieve the required precision and accuracy, the complex environments and narrow spaces would not allow such applications. Meanwhile, as workflow designed for robot control, researchers in the fields of robot control, computer vision and surveying had developed 2D and 3D SLAM algorithms, integrated SLAM algorithms, and SLAM algorithms with constraints to correct the drifting errors of IMU. Therefore, SLAM and IMU-SLAM were the most popular solutions for precise positioning and orientation in such GNSS-denied scenarios. The reliability, precision, accuracy and attitude recovery for the constant moving platform would continue to be the most highlighted research area for such indoor mapping system.

Secondly, the presence and identification of features would undoubtedly affect the precision and reliability of the SLAM algorithm. For multi-line MLS and RGB-D cameras, the resolution of the point clouds, which was determined by the design of the scanners and the distances between the objects and the scanners, decided the detail levels of the features and their identification from the low-density inhomogeneous point clouds. Low-density point clouds may have resulted in edge-loss problems that were difficult to recover (P. Tang et al., 2009). Meanwhile, for optical sensors, the absence of texture in indoor environments was the main reason causing track-loss problems. Therefore, how to design the feature extraction algorithm

to overcome the limitation of feature existence should be considered at the beginning stage of developing the mobile mapping system.

Thirdly, sensor integration was a reasonable choice for building mobile positioning and orientation solutions. Different sensors used for geometric and texture data acquisition, positioning, and attitude determination had already been integrated into most of the solutions to make more relevant data available. However, sensor integration should be used not only for data fusion but also as the extra source for other sensors to achieve higher accuracy and reliability in both positioning and orientation. As mentioned, both laser scanner and optical sensors had their limitations in continuously capturing features used for the SLAM process. If fused with IMU data, even segments of LiDAR SLAM or vSLAM results could be used to limit the drifting of IMU and produce better DR results.

Last but not least, neither SLAM nor IMU can eliminate the drifting problem in indoor positioning and orientation. The distribution and accuracy of GCP had requirements. Satisfying distributions of GCP with acceptable accuracy were reliable controls for noisy and drifting positioning results. For example, the stair rooms always required more GCP for vertical variation control than the flat rooms. The accuracy of the GCP coordinates would, on the other hand, directly affect registration accuracy and elimination of the drift error made by the onboard IMU and the SLAM algorithm.

2.4 Plane Extraction from Point Clouds

As discussed above, most solutions adopt Iterative Closest Points (ICP) as one of the main tools for 2D LiDAR SLAM. However, 2D SLAM algorithm limited the moving platform to the smoothly moving platforms, such as pushcarts and vehicle-based robotics, which barely moved or only vibrated slightly and smoothly in the vertical dimension. If there were any significant motions in the vertical dimension with unknown tilt angles, such as a backpack solution, a high-precision IMU would be an essential component for estimating the instantaneous attitude of the platform. Since there was no high-precision IMU installed in the proposed S^2DAS solution and a high DOF SLAM would improve the reliability of the whole system, using the IMU in isolation to facilitate the attitude recovering process was not considered. Therefore, developing a novel algorithm to achieve IMU-free 3D SLAM would be the first challenge in the data processing workflow.

For ordinary, high-resolution 3D point cloud alignments, ICP algorithms could be directly applied to calculate the transformation matrix and register both point clouds to the same coordinate frame (Granger & Pennec, 2002) (Niloy J Mitra, Gelfand, Pottmann, & Guibas, 2004). Some of the other methods were based on statistical histograms for rotation and translation movement (Makadia, Patterson, & Daniilidis, 2006) (Rusu, Blodow, Marton, & Beetz, 2008) (Rusu, Blodow, & Beetz, 2009). However, the most critical difference between the high-resolution TLS point

clouds and the point clouds generated by the multi-line scanners was the inhomogeneous distribution of points. This low-resolution problem also brought difficulties in modeling the regional shapes of objects, such as estimating the local normal vectors of points, which could be used to identify shapes from the point clouds. As the proposed working areas for the proposed mapping system were artificial indoor environments, planes extracted using a novel method were considered as the features used for recovering the relationship between frames of point clouds, based on the assumption that the geometric relationships between frames can be recovered with at least three pairs of planes facing different directions.

To recover shapes from quasi-continuous point clouds, modeling of smooth surfaces, especially the planes, is one of the major research highlights in point cloud processing. Such smooth surfaces include, but are not limited to, roads, walls, roofs and other surfaces which could be considered as low-noise surfaces with considerably small disturbances that could be represented using regular parametric surfaces. Meanwhile, such regions were commonly human-made objects of interest and required automatic extraction from large-volume point clouds. Architectural appearances, internal of manufacturing facilities, historic sites, and human organs were categorized as such surfaces of great interest. Various modeling workflows had been introduced by scholars and commercial software packages (Kusnoto & Evans, 2002) (P. Tang et al., 2009) (P. Tang, Huber, Akinci, Lipman, & Lytle, 2010) (Brilakis et al., 2010) (Volk, Stengel, & Schultmann, 2014).

To achieve rapid extraction of features from low-resolution inhomogeneous

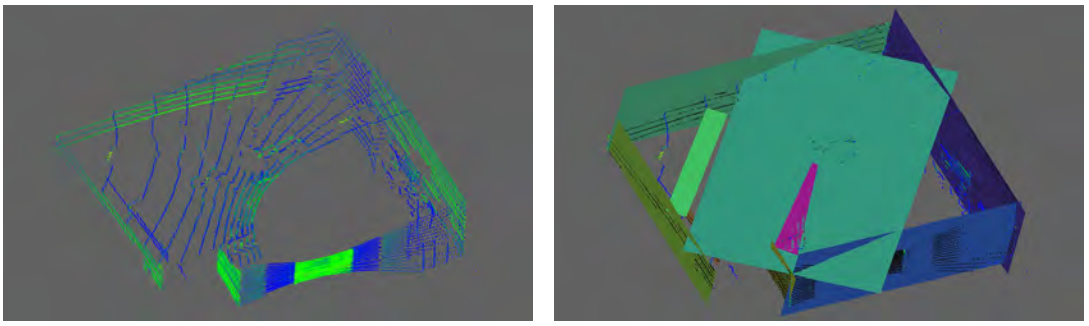
point clouds for SLAM, planes were selected as features, and their specific distributions were used as tie features between frames of point clouds. Once segmented and identified, any disturbances and noises on the surfaces would be considered as useless points, leaving only least-square fitted parametric elements representing the given subsets of point clouds. Numerous techniques had been proposed and developed for modeling planar surfaces. Some of the most typical workflows are reviewed in this section.

2.4.1 RANSAC and Its Variants

RANSAC was one of the most popular modeling fitting methods in extracting shapes from dense point clouds. Initially designed for establishing the geometric relationship between images in photogrammetry, the algorithm and its principle, which was seeking the subset with the largest number of elements fitting specific geometric patterns, had been deployed in various software libraries for extracting shapes and objects from given point clouds. Schnabel, Wahl, and Klein (2007) listed several applications for segmenting and identifying different shapes from point clouds, such as planes, spheres, cylinders, cones, and tori. Meanwhile, the variants and extensions of RANSAC—M-estimator SAmple and Consensus (MSAC), Maximum Likelihood Estimation SAmple and Consensus (MLESAC), and PROgressive SAmple and Consensus (PROSAC)—took the residuals for the estimated models into consideration to improve the likelihood of the models fitted (Grilli, Menna, & Remondino, 2017). A universal solution for applying these sample consensus modules was Point

Cloud Library (PCL) introduced by Rusu and Cousins (2011). Modules for extracting planes, lines, circles, spheres, cylinders, cones, and elements with specific requirements in directions and normal vectors were defined with estimators such as classic RANSAC, Least Median of Squares, MSAC, Randomized RANSAC, Randomized MSAC, MLESAC, and PROSAC. Meanwhile, the free toolbox CloudCompare also provided useful tools for RANSAC shape detection (Schnabel et al., 2007).

An example is used to demonstrate their application to an inhomogeneous point cloud of a simple environment, which was captured by a multi-line MLS, as shown in *Figure 2.23*. Wrong segments which could be easily spotted and removed manually, were identified. However, RANSAC method could not perfectly identify planes from low-resolution inhomogeneous point clouds of complex environments directly, such as the point clouds captured by multi-line MLS.



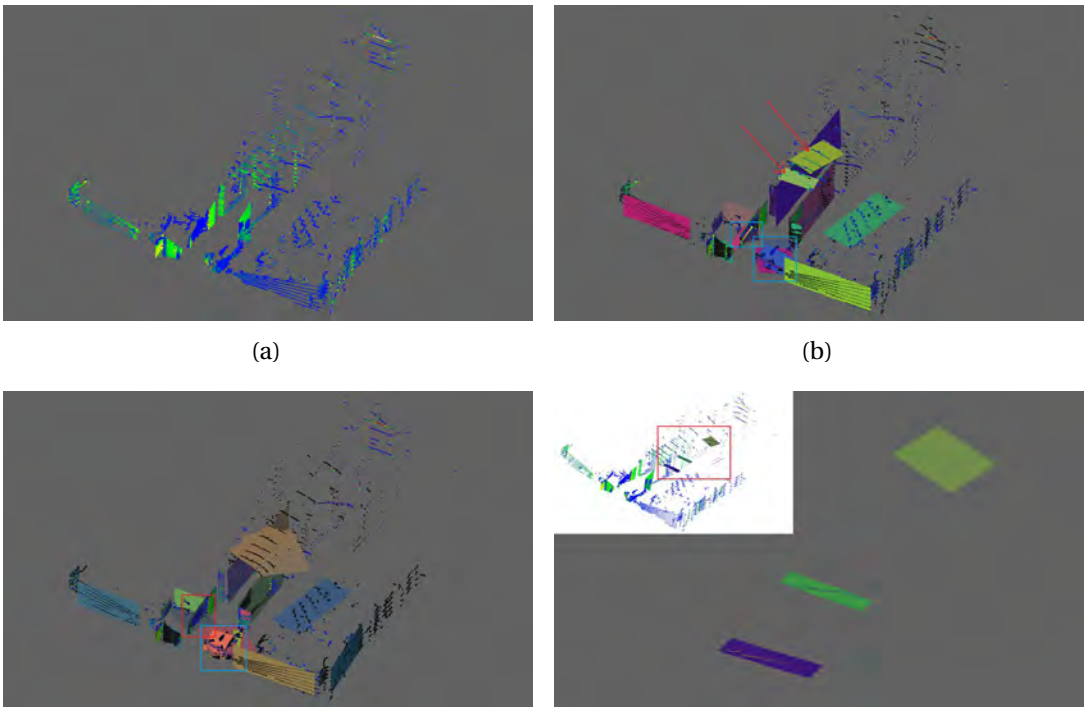
(a) (b)

FIGURE 2.23: Example of applying classic RANSAC to extract planes from a multi-line MLS point cloud of a neat room. (a) The raw point cloud that points are colorized concerning the reflected intensity, blue for low intensity while green for high intensity. (b) The classic RANSAC plane extraction result using CloudCompare with the implementation of Schnabel, Wahl, and Klein (2007). The value of the minimal support points per primitive was set to 100. The distance-to-plane threshold was set to 0.03 m. All other parameters were kept the default. Planes intersect in the ceiling area, showing the improperly identified planes.

In the meantime, as shown in *Figure 2.24*, when processing the low-resolution part of the point cloud, the estimator, which considered the size of subsets only, generated false results instead of the approximation in the vicinity. In some applications such as CloudCompare, local estimated normal vectors were introduced to the estimator to eliminate such errors, as subsets with normal vectors with great divisions would not be considered to be on the same plane.

Nevertheless, the local estimated normal vectors, which were the normal direction of the tangent plane at the pending point, were easily affected by the measurement noises. Also, those matters affecting the estimation of the normal vectors, such as the local resolution of the point cloud and kernel size of the neighborhood, also reflected the local normal estimation process (Hoppe, DeRose, Duchamp, McDonald, & Stuetzle, 1992) (Niloy J. Mitra & Nguyen, 2003). Because it is one of the main factors in region growth workflow, the problem of miscalculation is discussed in *Section 2.4.2*.

A robust algorithm was required to provide plane extraction results in the SLAM process. RANSAC and its variants were not considered as a solution in the proposed workflow due to their limitations. However, because it is a method for removing noise by filtering unwanted data from given groups of points, RANSAC could be used to provide refined estimations of models, i.e., planes, cylinders, and spheres.



(a) (b)

(c) (d)

FIGURE 2.24: Example of applying RANSAC algorithm to extract planes from a multi-line MLS point cloud of a corridor with glasses. (a) The raw point cloud in which the points are colorized to reflect intensity, blue for low intensity and green for high intensity. (b) The classic RANSAC plane extraction result using CloudCompare with implementation of Schnabel, Wahl, and Klein (2007). The value of the minimal support points per primitive was set to 100. The distance-to-plane threshold was set to 0.03 m. All other parameters were kept the default. Wrong extraction results are marked in blue boxes. Although planes marked with red arrows are points from the same plane, they were divided into two unparallel planes. (c) Result with pre-calculated normal vectors using Boulch and Marlet (2016) method in CloudCompare to eliminate wrong planes in (c) (blue box region) but still wrong results left (red box region). (d) The RANSAC detection result based on manually selected points on the floor selected from the region in the red box shown on the left-upper corner. The algorithm cannot properly group the given points into the same plane but into three separated plane segments.

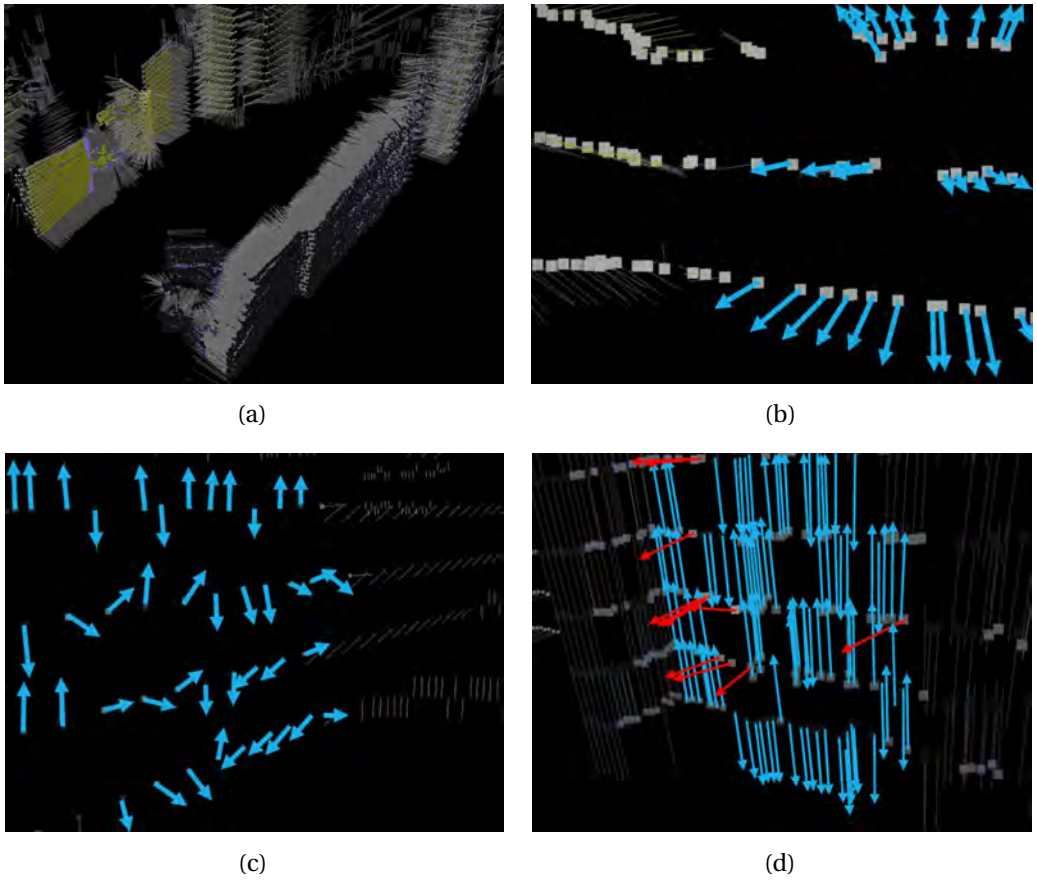
2.4.2 Classic Region Growth

Region growth was another popular solution for extracting segments from dense point clouds and also was deployed in PCL (Rusu & Cousins, 2011). This algorithm

was capable of connecting adjacent regions and points with similarity in the directions the patches were facing and in the curvature changes along with the expansion of the planes. It did these by inputting and defining local estimated normal vectors, local curvatures, neighbor finding function, curvature threshold, and angle threshold in addition to the pending point clouds. In some solutions, point-to-patch distances of pending points were considered.

Though listed as one of the main input factors, the definition of the neighborhood region affected the performance and reliability of this algorithm greatly. The default method for search in PCL is the k-Nearest Neighbors (kNN) algorithm based on k-d tree, while either the number of k or the distance threshold to the pending point can be selected (Rusu & Cousins, 2011). In other words, this searching algorithm was not considering that there might be congested points in small areas affecting the estimated normal vectors and shapes. For example, if there were 50 points existing in a region of 10 cm × 10 cm × 10 cm with other nearest neighbor (not a member of the 50 points subset) at the minimal distance of 1 m and the kNN search distance was set to 10 cm, all the 50 points would be adopted for estimating the local normal and their normal directions would be facing the same direction. Consequently, the 50 points, which could be distributed in any shapes, would be considered as a planar segment. Meanwhile, if the distribution of points was not homogeneous, which is common in MLS point clouds, the search for eligible neighbors would be conducted along the directions of the scanlines. The incapability of searching neighbors in all directions might affect the estimation of the local areas.

Using the same point cloud in *Figure 2.24*, *Figure 2.25* shows some examples of miscalculated local estimated normal vectors. The inhomogeneous distribution of points in multi-line MLS point clouds, especially the linear distribution along the rotating directions, had greatly affected the estimation of the local normal vectors (Shan & Toth, 2018).



(a) (b)

(c) (d)

FIGURE 2.25: Estimated local normal vectors of the point cloud of a hallway. All normal vectors are represented as thin gray line segments starting from the pending points and extending along the local estimated normal directions. (a) 45 °-view of the hallway with local estimated normal vectors. (b) Corrected local estimated normal vectors of points of wall surfaces with blue arrows marking local estimated normal vectors pointing to nearly the same directions. (c) Wrong normal estimation results using radius as parameters for determining local areas. The normal vectors in blue were local estimated normal vectors that should be pointing to a similar direction. (d) Errors in local normal estimation results using kNN for determining local areas. The arrows in blue should be pointing to similar directions as the arrows in red.

An alternative was to conduct a grid-based neighbor search strategy for estimating local normal vectors by using the original data acquisition sequence of the MLS. However, as the geometric resolution of the grid varies with the distance between the pending point and the scanner, in addition to the actual distance between the neighbors in the grid, there should be an extra determination of removing fake neighbors, which will affect the normal estimation result as well. For example, a grid neighbor far away from the pending point might not be a suitable point for estimating the local normal of the neighborhood with the radius of 10 cm. However, even when properly assigned, experiments by Miyazaki, Yamamoto, and Harada (2017) showed that the normal estimations of points on the edges of planes were affected, leading to the omission of plane boundary points on the identified planes. Such a problem might not affect the high-resolution point cloud. However, it would create vital failure in identifying planes from low-resolution point clouds as, indeed, there were not so many points on the small-sized planes. The defects of the grid-based normal estimation are further discussed in *Section 2.4.4*.

Another region growth workflow, which was not based on local estimated normal vectors, was realized by Vosselman, Gorte, Sithole, and Rabbani (2004). Named Bottom-up method, this process of approximating and recovering planar patches was considered a simple, easy-to-implement, and calculation-efficient method (P. Tang et al., 2010). Starting from local patches of small sizes after examining the flatness and surface curvatures of certain neighborhoods in point clouds using Hough

transform, small meshes were used as initial elements in the hierarchy building process to detect planes that were larger than the given thresholds (Vosselman et al., 2004).

In the initial patch detection process, the least-square method and statistical noise removal procedure were implemented to sort out the subset indicating a plane and fit a specific set of parameters to it. Then, by checking the point-to-patch distances along the normal direction of the patch and the projection-to-boundary distances along the expanding direction within the patch plane, candidates were added into the subset to form a larger patch if both requirements were met. The parameter of the plane would also be updated once new points were added into the patch.

This method should be one of the universal methods as a normal vector was not required in the entire bottom-up process, avoiding the error-prone normal estimation process. However, the tedious distance checking process for each single point was considered to be time-consuming on a CPU-only platform. With the developing of parallel computing using GPU computing units, this algorithm could be redeveloped.

Dong, Yang, Hu, and Scherer (2018) presented a supervoxel-based region growth method that iteratively classified multiscale supervoxels into planar and nonplanar supervoxels and connected them to form planes. However, the test was conducted using a TLS point cloud, while the calculation of the pointwise geometric features was based on kNN search and the spatial distance was selected as the only geometric feature for over-segmentation, which was already proved to be unreliable in

inhomogeneous MLS point clouds.

2.4.3 3D Hough Transform

As an extension of the 2D Hough transform in 3D spaces, 3D Hough transform could be applied in the plane detection process (Vosselman et al., 2004) (P. Tang et al., 2010). Based on *Equation 2.6* of non-vertical planes in the real space given by Vosselman et al. (2004), the corresponding mapping of planes in the real space can be defined as a point in the parameter space.

$$Z = s_x X + s_y Y + d \tag{2.6}$$

where:

(X, Y, Z) are the coordinates of the points on the plane in the real space,

s_x and s_y represent the slope of the surface with respect to the direction along the X-axis and Y-axis, and

d is the height of the surface in Z-axis at the origin point of the space.

As defined, the corresponding mapping point in the parameter space was point (s_x, s_y, d), which defined a plane in the real space. Therefore, all planes in the real space, passing the pending point (X, Y, Z), could be represented as surfaces in the parameter spaces. The detection of planes was based on the assumption that the number of planes intersecting at the same point in the parameter space equals the

number of points in the object space or the actual space that were located on the same plane which was represented by the point in the parameter space (Vosselman et al., 2004) (Tarsha-Kurdi, Landes, & Grussenmeyer, 2007) (Borrmann, Elseberg, Lingemann, & Nüchter, 2011). Consequently, the plane parameters would be estimated using the intersection point in the parameter space. In reality, due to inevitable variations, such as measurement noises and other disturbances, the corresponding planes in the parameter space rarely intersected at the same point. Therefore, a 3D bin was introduced for automatic identification of the intersection (Vosselman et al., 2004).

With local estimated normal vectors provided, the planes in the parameter space were limited to the tangent planes defined by the normal vectors. Subsequently, a Gaussian sphere was used for identifying the normal vectors with similarity in pointing directions as a 2D parameter space, with a 1D parameter space for determining the d value (Vosselman et al., 2004).

Compared with other popular methods such as RANSAC and region growth, 3D Hough transform was considered a computational-intensive algorithm, which was sensitive to the segmentation results before the estimation process, especially in the applications to certain scenarios with plenty of parallel planes in the vicinity, while the automatic determination of the parameters was also considered as a difficult job (Tarsha-Kurdi et al., 2007). In addition, its definition of planes in *Equation 2.6* also restricted the facing directions of the plane extracted, which might not be suitable for identifying planes in artificial environments.

2.4.4 Plane Extraction from MLS Point Clouds

The linear distribution of points captured by vehicle-based mobile mapping systems had become one of the most popular survey products waiting for future processing as well, though it differed from the point clouds captured by Airborne Laser Scanning (ALS) and TLS. Moreover, the extraction of attribute information and semantic modeling had been highlighted research topics in recent years. Planes, being one of the most common natural and artificial elements, were crucial for extracting structure information from point clouds of enormous sizes. Building facades, road surfaces, traffic signs, and other objects of interests were commonly in the shape of planes, and their identification was based on the prior extraction of the segments from the point clouds first. In addition to the methods mentioned above, workflows had been designed for detecting planes from point clouds captured by vehicle-based MLS.

When normal vectors are available, region growth can be performed based on Euclidean approximation, the angles between local estimated normal vectors of points, and the distance between the pending point and the patch. Based on the classic region growth method and analysis of normal vectors derived using PCA algorithm, Nurunnabi, Belton, and West (2016) integrated a robust segmentation method and a slice merging algorithm to form a plane detection algorithm for point clouds captured by TLS, ALS, and MLS. However, for sparse point clouds, the linear distribution of point clouds, which is quite common in MLS point clouds, missing points along the moving directions of the platform may result in the incapability of

estimating reliable normal vectors, which is similar to the results shown in *Figure 2.25*.

In most circumstances, the grid distribution of the multi-line MLS point clouds could be used to calculate local estimated normal vectors. However, as the grid distribution did not reflect the actual geometric distributions based on 3D Euclidean distances, the normal vectors estimated could be easily influenced by their neighbors in the grid. Miyazaki et al. (2017) showed the normal calculation results of similar data sets, linear single-line MLS data, showing the normal vectors of the knee points on the scanline would be improperly tilted as its neighboring points were actually on two different planes. Such disturbance would not significantly affect dense point clouds, such as the sample data set listed in that report, as the rest of the points were still abundant for identifying planes. However, for inhomogeneous sparse point clouds captured by multi-line MLS, in which there might be only two lines of points reflected by the pending planes, such mistakes would result in missing planes. Czerniawski, Sankaran, Nahangi, Haas, and Leite (2018) demonstrated similar results, showing that even for dense point clouds, the normal vectors of edge points would be affected, making the understanding of semantic information from such data sets error-prone. Y. Fan et al. (2018) summarized and compared most of the methods used for point cloud segmentation, presenting the importance of the normal vectors for point cloud segmentation in conventional methods, which was indispensable for recovering local geometric features from the point clouds. In

other words, the mistakes in estimating local normal vectors would result in different kinds of errors and incapability of extracting the intact planes.

To avoid using normal vectors in plane detection from point clouds captured by vehicle-based MLS, Cabo, García Cortés, and Ordoñez (2015) and H. L. Nguyen, Belton, and Helmholz (2019) designed similar methods for clustering and combining line segments extracted using the Douglas-Peucker algorithm. Cabo et al. (2015) used neighboring endpoints in the line segment searching process for clustering. However, for noisy point clouds, missing a few points from the extracted plane was quite normal, which might result in the incapability of extracting valid planes using these criteria. W. Wang, Sakurada, and Kawaguchi (2016) extended the application of this workflow to the point clouds captured by multi-line MLS, which was similar to the sensors installed on S^2DAS. The distance threshold was adopted in the searching process, which was determined by the distance between the point and the scanner and the elevation angle of the laser beam. However, checking only single points for the whole line segments may cause misclassification since the small but consist direction differences between the two adjacent line segments might be considered as the existence of two lines on the two planes that are quite close but with small angles. More representative points, such as the first and last points, and the centroid of the line segments, should be checked.

Moreover, the analysis on single-line MLS data only concentrated on the mirror rotating direction in Cabo et al. (2015) and H. L. Nguyen et al. (2019). Consequently, parallel line segments with small variations in the direction within the

pending plane, which were very common in the noisy point clouds captured by the scanners designed for outdoor autonomous driving, could not be distinguished. Another main flaw of the three methods was that no scanline curvature was considered in the merging process. As only one single dimension was considered in the solutions mentioned above, which was the direction along the horizontal scanline, the analysis on variations and shape changes was limited to the narrow scope of the original scan sequence of points captured by the same scanline with the same vertical elevation angle. The scanline curvatures, as introduced by Grant, Voorhies, and Itti (2013) and shown in *Figure 2.26*, which were unavoidable in point clouds of large-scale environments, were not considered in the three methods, while the curvature caused the direction differences in line segments on the same plane.

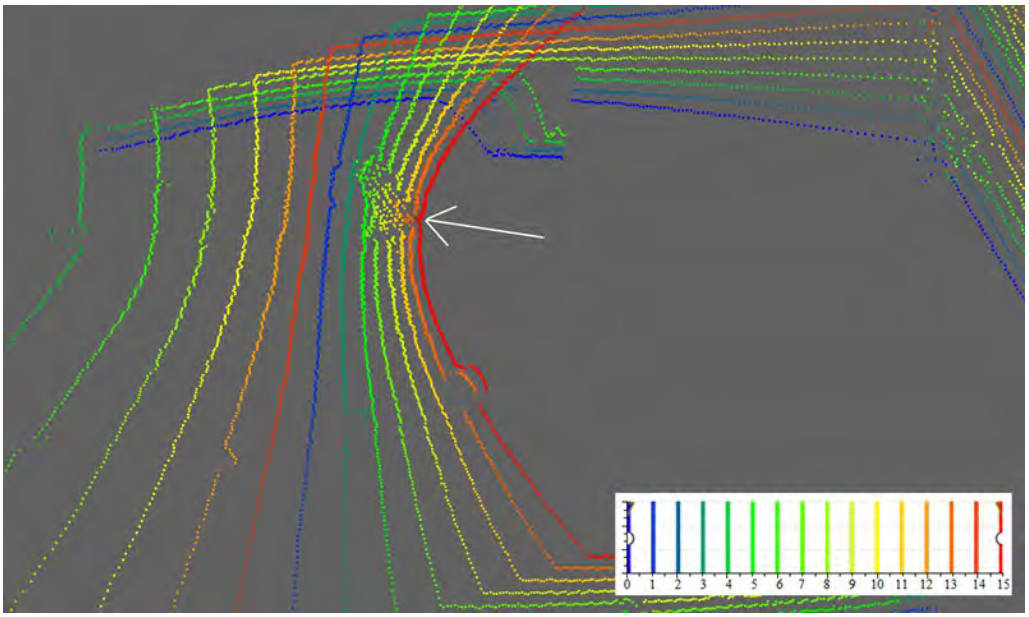


FIGURE 2.26: Curved scanlines existing in point clouds captured by a multi-line MLS scanner. Points are colored in different colors with respect to the scanlines. There is significant curvature in the scanline in red and marked with white arrow. The curvature of neighboring scanlines shows identifiable differences from the scanline marked with white arrow.

In our experiments, line clustering methods that considered only the single sequence would be affected by the so-called "over-fragment" problem, which was the same plane being separated by the scanline curvatures, as shown in Figure 2.27. Although neighboring planes could be easily combined by checking their orientation and distance, there were still risks that the missing edge points and the variations in normal directions, which was quite common in MLS point clouds captured by laser scanners with such low accuracy, would cause the miscalculation of the distance between plane edges and result in unsuccessful plane fusing.

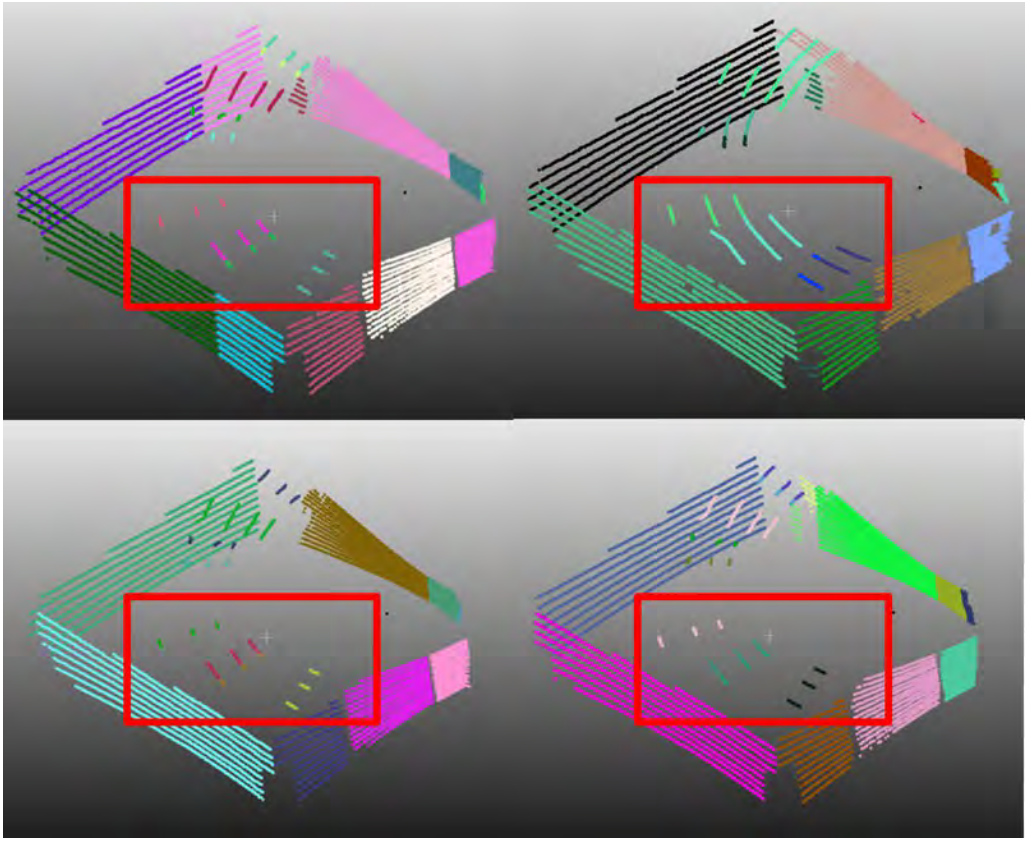


FIGURE 2.27: The over-fragment problem caused by scanline curvatures. The ceiling points in the red box should be grouped into the same plane.

In the meantime, Grant et al. (2013) utilized the curvatures of the laser scan-lines in identifying planes from the low-resolution inhomogeneous point clouds. However, for small planar patches, the occurrence of curvatures was not very identifiable compared with the vibrations of the scanner and the measurement noises, which might cause unsuccessful detection and identification.

Depth images are another kind of structured point cloud data usually generated by depth cameras. Pathak, Vaskevicius, and Birk (2009) and Pathak, Birk, Vaškevičius, and Poppinga (2010) have introduced a registration method for depth images that planes were extracted for seeking the correspondence between frames. This method made point clouds with homogeneous resolution possible for recovering the correspondence. However, as points in depth images were evenly distributed according to their sensor type and limited working range, the algorithms used for extracting planes from depth images could not be applied to the low-resolution 3D point cloud generated by the SLAM scanner in S^2DAS.

2.4.5 Previous Works on the ELS Algorithm

In W. Fan (2015), an enhanced algorithm—the ELS algorithm—was designed and verified on 3D TLS point clouds, for deploying 2D line simplification algorithms to extract breakpoints and feature points from structured point clouds. The algorithm was based on the linear distribution of the structured point clouds and acted as an extension of the 2D line simplification algorithm. The ELS algorithm considered each of the points between the two endpoints of the given line segment as

the pending points. The feature points representing the most appreciable curvatures could be reserved by checking the distance between the pending point and the straight line segments formed by the two endpoints. The checking procedure was performed in the two original scanline directions of TLS. Subsequently, the most representative points in shapes were extracted.

Similar ideas, such as the one in Woo, Kang, Wang, and Lee (2002), had already been presented that the structure and sequence hidden in the point clouds with respect to the working principle should be utilized. However, it was still the normal vectors that were calculated and used as the features for identifying breakpoints and boundaries. Although Delaunay triangulation was introduced to generate reliable local surfaces for calculating local estimated normal vectors, the distances between points, which may affect the local estimated normal vectors, were still not taken into consideration.

One of the limitations of the proposed algorithms in W. Fan (2015) was that although the algorithms were designed to omit the time-consuming normal calculation process, the efficiency of the algorithms was not preferable for real-time data parsing. Therefore, modifications and improvements were made to allow the algorithm become a high-performance workflow, which was part of the plane extraction algorithm. Moreover, considering that data processed in the real-time SLAM progress on S^2DAS was structured, the algorithm was implemented on the raw data and used for extracting planar segments from the low-resolution point clouds.

Furthermore, although the algorithm was capable of extracting feature points

which could be used for modeling planes, the final plane modeling processing was a manual process based on the selected feature points rather than a direct segmentation and plane extraction algorithm. Automatic plane extraction based on the ELS algorithm should be implemented for batch processing of point clouds captured by multi-line scanners.

2.5 Feature-based SLAM

The general working mechanism of SLAM was that by capturing point clouds or images, the corresponding features could be extracted and aligned to recover the geometric relationship between the frames. Thus the trajectory of the moving platform, including the positions and attitudes of every frame, could be derived recursively. Although the errors in trajectory positions and attitudes increased with the accumulation of the moving distances, redundant alignments between non-adjacent frames could be established for adjustment. Such alignments were used to correct the position and attitude drifts using the specific estimation and data processing method, such as EKF, Unscented Kalman Filter (UKF), and partial filter in IMU DR workflow (Montemerlo & Thrun, 2003) (J. Zhang & Singh, 2014) (J. Zhang & Singh, 2015) (Kamijo, Gu, & Hsu, 2015). Therefore, a fused trajectory of the platform could be derived in the data fusion process.

In LiDAR SLAM, three typical categories were used to recover direct coordinate relationships: point-to-point methods, feature-to-feature methods, and grid-based

likelihood methods. In addition, there were also methods based on building the relationship between the single-frame data and the accumulated maps, such as the works done by J. Zhang and Singh (2015), also called incremental SLAM methods.

For direct point-to-point matches, either the geometric distribution of every single point or the distribution functions of points could be utilized. ICP algorithm was one of the most popular algorithms in implementing either point-to-point methods or feature-based methods by conducting the iterative process to align the pair of certain subsets of the given point clouds (Besl & McKay, 1992) (C.-C. Wang & Thorpe, 2002) (Liu et al., 2010). Moreover, Iterative Matching Range Point (IMRP) and Iterative Dual Correspondence (IDC) methods worked by iteratively optimizing the position and orientation changes to align the frames (Lu & Milios, 1997) (Brenneke, Wulf, & Wagner, 2003) (Diosi & Kleeman, 2005). Normal Distribution Transformation (NDT) used the point distribution functions to identify the likelihood of point distributions between frames and estimate the corresponding position and attitude changes (Biber & Strasser, 2003) (Magnusson, 2009). Biber, Fleck, Wand, Staneker, and Straßer (2005) used scan matching based on the probabilistic density functions to establish the pairwise relationships between frames with global optimizations.

Different kinds of features, namely landmarks, were selected for building the relationships between frames of point clouds, such as endpoints by Achtelik, Bachrach, He, Prentice, and Roy (2009) and Kohlbrecher, von Stryk, Meyer, and Klingauf (2011); unevenly distributed points by J. Zhang and Singh (2014); line segments in 2D-only solutions by L. Zhang and Ghosh (2000), Pfister, Roumeliotis, and

Burdick (2003), and V. Nguyen, Gächter, Martinelli, Tomatis, and Siegwart (2007); and planes by Pathak, Birk, Vaskevicius, et al. (2010), W. Wang et al. (2016), and Dai, Nießner, Zollhöfer, Izadi, and Theobalt (2017). Corners and lines in indoor environments, and trees, which were cylinders in outdoor environments, were popular features used in the SLAM process (Y. Li & Olson, 2010). Since the points and features were limited in single-frame point clouds, most of the feature-based methods were based on feature-to-map alignment to maximize the correlation between point clouds. Meanwhile, when feature points were selected for alignments, the feature-based methods were based on either point-to-point or point-to-map alignments with only specific feature points. Thus they were considered as points in the matching process.

2D grids and 3D voxels, which are different forms of grid maps in 2D and 3D, could be used to estimate the position and attitude change by checking the occupied status in the observation spaces (Rivadeneyra & Campbell, 2011) (Lee, 2015). Such methods considered the occupancy of points in the space. Then the occupied cells were considered as points for the localization and orientation process.

2.5.1 Plane-based SLAM Algorithms

The selection of the features adopted in SLAM was changed with the evolving of MLS. In the earlier years, when only 2D MLS were available, lines and circles were selected as the features as they were the most identifiable features with low repetition. Furthermore, the specific distribution of such features was even more reliable

as it was unique in the testing environments in most cases. With the subsequent development of multi-line MLS and the invention of the dedicated spinning installation method, such as GeoSLAM products, 3D features, which include but are not limited to planes, cones, and cylinders, were used as features for estimating location and orientation changes in 3D spaces.

The use of cones and cylinders was similar to the application of line-based SLAM algorithms, as they protracted along their axes, while the planes extended along all directions which were perpendicular to their normal vectors. Meanwhile, considering the wide distribution of planes in natural and artificial spaces, plane-based SLAM methods had been introduced and discussed by scholars in the past decades.

With the extraction of plane boundaries, Poppinga, Vaskevicius, Birk, and Pathak (2008) presented a direct registration method for RGB-D sensors as the resolution was relatively more homogeneous. However, this method could not be applied for low-resolution point clouds captured by multi-line scanners, because the sparse distribution of the points introduced significant uncertainty in determining the position of the boundaries. Meanwhile, Theiler and Schindler (2012) addressed a co-registration method by aligning intersection points of planes extracted from TLS point clouds. Nevertheless, the result was limited by the accuracy of the plane extracted, as the accuracy of the TLS was much higher than that of the MLS. For planes extracted from MLS point clouds, whose parameters might be affected by the inevitable noises and disturbances in the angle and distance measurements, the positions of the intersections might be fallible and cause significant errors in alignments.

Pathak, Birk, Vaskevicius, et al. (2010) presented a robot based on a single-line scanner pitching from -90 to +90 °, providing the vertical scanlines with the interval of 0.5 °. The robustness and speed of the online region-growth-based plane extraction introduced by Poppinga et al. (2008) and the plane-based 3D SLAM had been verified as feature-based SLAM. The algorithm demonstrated better performance in both time consumption and robustness when compared with ICP. It also introduced proper procedures which could be used for solving the position and orientation changes. W. Wang et al. (2016) introduced a similar IMU-free SLAM pipeline which was adopted in the proposed workflow listed in this report. However, the segmentation results of the previous frame would be required to produce acceptable results. Meanwhile, it was considered that the method would achieve better performance with the availability of RGB panoramic images and the corresponding integration.

Ulas and Temeltas (2012) utilized RANSAC extracted planes with EKF and UKF processor. The results showed acceptable accuracy in indoor environments though it did not perform well in outdoor environments. Geneva, Eckenhoff, Yang, and Huang (2018) also proposed an integrated SLAM method where a continuous pre-integration of RANSAC plane extractions and IMU measurements was implemented together to produce reliable results in indoor environments. Meanwhile, the RANSAC method proved to be so time-consuming that the plane extraction had to be conducted offline.

By projecting point clouds to 2D image spaces to extract planar surfaces, Lenac,

Kitanov, Cupec, and Petrović (2017) realized a plane-based SLAM workflow designed for multi-line MLS by registering planes to local maps. However, the system did not perform very well without IMU integration. Grant, Voorhies, and Itti (2019) demonstrated the application of the plane detection algorithm introduced in Grant et al. (2013) on outdoor SLAM that used both planes and points. The constraints of planes in multiple directions were discussed, and points were used to facilitate the alignment process when the planes were not capable of providing both rotation and translation parameters in the three axes. The solution performed well in both outdoor and indoor scenarios. However, the poor performance of the adopted plane-extraction methods in complex environment must be considered.

2.5.2 Discussion

Various kinds of features were selected as the landmarks in SLAM workflows in the past decades. Since the proposed mobile mapping solution was based on a backpack moving in 6 DOF, the motion of the platform would not be precisely estimated with 2D SLAM techniques. Furthermore, point-to-point SLAM might not provide robust results either since the vertical resolution of most modern 3D multi-line MLS quite low. Taking the most widely adopted MLS, Velodyne Puck and Puck Hi-res as examples, since the vertical interval between scanner channels was either 2 ° or 1.33 °, the possibility that the points of different frames might not be reflected by the same positions on the surrounding objects must be considered. The laser

beams emitted by the same multi-line scanner could barely hit the precise same position twice, as shown in *Figure 2.28*. The point-to-point SLAM workflow, which was based on the hypothesis that the alignment of point clouds was equivalent to the alignment of feature points, would be affected by such position changes of feature points. Such influences were not quite apparent in 2D SLAM as it was easy to rectify for the low-DOF platform. However, when the motion in 3D was considered, such as movement on a backpack platform, the 6 DOF motions created significant uncertainty in estimating the three-axis rotations and translations, which would result in drifts in the subsequent DR process.

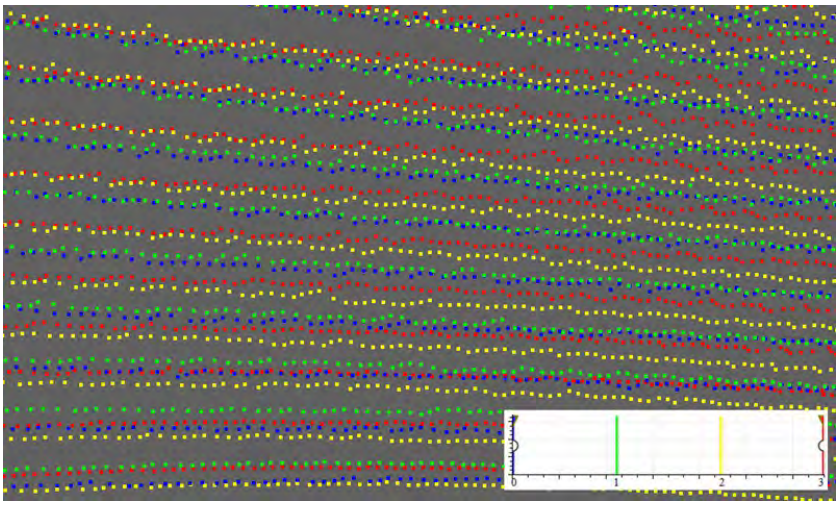


FIGURE 2.28: An illustration of the non-common-point problem on the platforms with 6 DOF motion. The low-resolution point clouds of consecutive frames were in different colors while they had been registered to the same frame. The intersectional distributions of the points show that the points could hardly hit the same location twice.

Therefore, planes were selected as the features in 3D SLAM with 6 DOF in the proposed workflow, which was the main reason why the two scanners were required

with angles between them, in addition to the purpose of capturing points from an extra dimension for constructing the point cloud with higher coverage.

Generally, planes were selected as features used in SLAM workflow, and results with acceptable accuracy had been produced for some of the IMU-free methods in indoor environments. The main limitation of most of the algorithms and methods was that the poor performance of the plane extraction workflow might degrade the reliability of the IMU-free SLAM workflow. Consequently, the applications to outdoor datasets would require integration with IMU or other sensors to improve reliability in complex environments with planes facing directions that are not enough for solving 6-DOF position and attitude changes.

2.6 A Summary of Plane Extraction and SLAM Methods for Mobile Mapping

In the sections above, various kinds of plane extraction methods and SLAM workflows, especially the plane-based SLAM techniques, were reviewed. Since the planes identified and the subsequent SLAM process were implemented to serving the applications to the mobile mapping backpack, the sensors involved must be considered as well in the discussion.

Most of the plane extraction methods were designed to segment dense point clouds, in which the local characteristics, such as the curvatures and local estimated

normal vectors, were used to detect the flatness of the patches and merge corresponding patches. Although the increasing size of the local neighborhood could reduce the possibility of wrong estimation resulting from the inhomogeneous distribution of points, the limited dimensions of low-resolution point clouds captured by multi-line scanner would restrict the identification of planes consisting of only a few scanlines, while the local shapes might be ignored as well.

Furthermore, as the limitation of adopting ICP and other point-to-point alignment methods has been discussed above, the features, especially artificial features, should be identified as the landmarks in the SLAM process. Since the feasibility of adopting planes in the SLAM processes had been proved, the poor performance of plane extraction from low-resolution inhomogeneous point clouds limited the reliability and precision of most plane-based SLAM workflows. Meanwhile, most workflows relied on the external IMU to provide the state initializations for matching the common plane pairs. The robustness of the plane extraction methods and the SLAM workflow should be improved for reliable implementation.

Therefore, a novel plane extraction method based on ELS feature point extraction algorithms is introduced in *Chapter 3*. Subsequently, a plane-to-plane IMU-free alignment workflow for low-resolution inhomogeneous point clouds captured by multi-line MLS is presented in *Chapter 4*.

Chapter 3

Plane Extraction from Low-resolution Inhomogeneous Point Clouds

As reviewed in *Section 2.4*, various plane extraction methods have been presented in the past. However, most could not produce satisfying results when given low-resolution inhomogeneous point clouds. To achieve 6 DOF alignment of point clouds, the planes were selected as an effective feature for registering point clouds to the same frame in the proposed mobile mapping workflow. The challenge in applying multi-line MLS to 6 DOF mobile mapping was in developing a novel method that could detect and identify planar segments from the low-resolution inhomogeneous point clouds in order to establish an automatic workflow of feature matching and point cloud alignments.

In this chapter, the modifications and improvements to the ELS algorithm to make it applicable for feature point extraction from low-resolution inhomogeneous

point clouds are introduced. Secondly, the ELS-based line segment clustering algorithms are presented. Finally, the test results on multiple scenarios are listed, and their pros and cons are discussed.

3.1 Modification of the ELS Algorithm

To apply the ELS algorithm into point clouds captured by multi-line scanners, modifications were need for better results. These modifications included altering the definition of point grid, extra virtual scanline directions, point shifting and projection, and optimizing programming codes. The reasons, changes, and results are listed in this section.

3.1.1 Point Grid Recovery

As introduced in W. Fan (2015), the ELS algorithm was designed to extract feature points and breakpoints from indoor point clouds from the curvatures of scanlines and virtual scanlines. In conventional TLS, the single laser prism kept rotating or oscillating to generate point clouds of the surrounding environments. Consequently, the vertical direction—the visible moving directions of the pulses due to the rotation or oscillation of the prism—was defined as the scanline direction. The rotation of the scanner base in horizontal plane caused the rotation of pulses in the horizontal plane, generating a scanline consisting of all the points with the same vertical angle

but different azimuth angles. The direction of this scanline was along the direction of the rotation against the vertical axis of the scanner.

The firing sequences of most of the multi-line MLS differed from those of the TLS. Taking Velodyne VLP-16 as an example, the point clouds generated were not strictly aligned in the form of grids because the sixteen lasers were fired with intervals in horizontal and vertical directions. Consequently, the original firing sequence was not a sequenced line but a set of connected polylines along the vertical and diagonal directions, as shown in *Table 2.1* and *Figure 3.1(a)* (Velodyne Acoustics Inc., 2015). The data output from the scanner needed to be rearranged before the ELS processing according to the corresponding vertical angles of points, which were along the increasing sequence of vertical angles, as shown in *Figure 3.1(b)*.

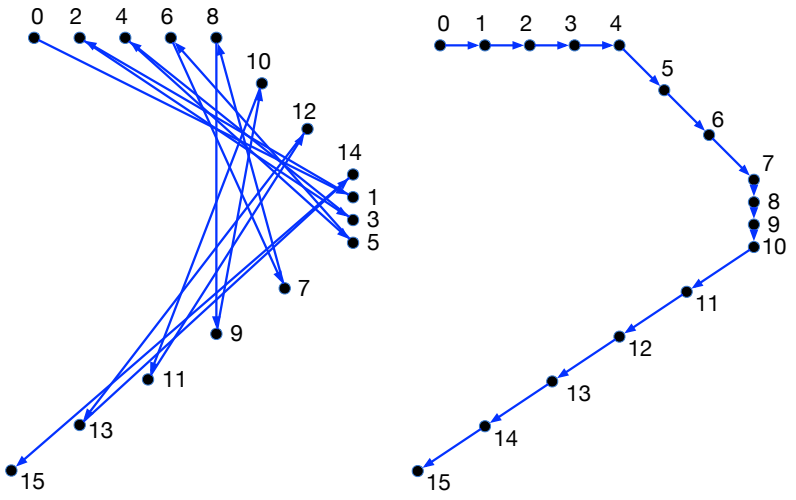


FIGURE 3.1: Profile views of a single vertical scanline demonstrating the point sequence before and after scanline rearrangement. (a) Raw point sequence recovered before scanline rearranged, the number labels beside the points are the fire sequence ID. (b) Rearranged scanline sequence, the number labels beside the points are the rearranged sequence ID.

After the initial firing sequence in each of the groups of the sixteen points had been recovered and rearranged, the assumed point grid could be recovered, as shown in *Figure 3.4*. The original raw scanline indicating the altered data acquisition sequence was along the vertical direction, although the horizontal angles were not the same. The sequence in the group of the sixteen points was changed while the sequence between the groups was maintained, generating the horizontal scanlines. Also, based on the vertical and horizontal scanline grids, the two diagonal scanline directions could be defined. Consequently, the ELS algorithm could be implemented.

As discussed in W. Fan (2015), the splitting operation would speed up the ELS processing, which was no longer needed in processing multi-line MLS point clouds. For three of the four processing directions, the length of each of the scanline segments was only the same as the number of the laser channels—sixteen if a Velodyne VLP-16 scanner were considered as the scanner. On the other hand, if the highest horizontal interval were adopted, which was 0.1 ° interval, each scanline would have 3600 points. The pre-separation would then no longer be necessary in this implementation. For easy understanding and implementation, the horizontal scanline direction was defined as the *row* while the vertical direction was defined as the *column*.

Another problem considered in the grid recovering process was the horizontal offset. The problem affected the processing of point clouds captured by some of the

multi-line scanner modules, such as RoboSense RS-LiDAR-32D and Velodyne VLP-32 (Suteng Innovation Technology Co Ltd, 2015) (Velodyne Acoustics Inc., 2018). The horizontal offsets within the group of thirty-two points could be up to a few degrees, as shown in *Table A.1* in *Appendix A* and *Figure 3.2*. Therefore, an additional operation was needed that would rearrange all points of a single frame to minimize the horizontal sequential offsets between points.

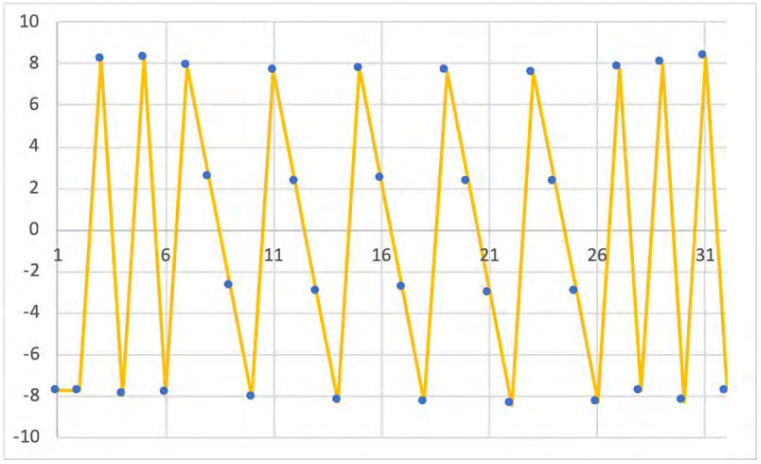


FIGURE 3.2: Horizontal distribution of installation offsets in ReboSense RS-LiDAR-32D scanner according to Suteng Innovation Technology Co Ltd (2015). The x-axis is the sequential ID of the rearranged channels, while the y-axis is the horizontal installation offsets with respect to the calculated horizontal angles versus the rotation speed and timestamp. Note that the horizontal installation offsets are different between scanners of the same model due to installation error.

Multiple methods have been implemented and compared, including methods based on minimal neighboring offsets, sharing points as public neighbors, direct shifting in the installation offsets, and nearest neighbor to the azimuths of the middle-column points. The fourth method was the best choice as less misalignment and sharing was generated, considering the corresponding time synchronization. In other words, the sixteenth point in the rearranged point column, which was the

middle point in the column, was selected as the reference point in the column. Regardless of the original data capturing sequence, the search for nearest neighbor with the minimal azimuth difference was conducted from the middle to the two endpoints in the column. Such neighbors were selected as points on the same column. The process is shown in *Figure 3.3*. After this process, a few points with azimuth differences that were too large could still not be properly reorganized as they were out of the major azimuth range of the rest points. They would be considered as feature points in most cases as they were identified by the ELS algorithm.

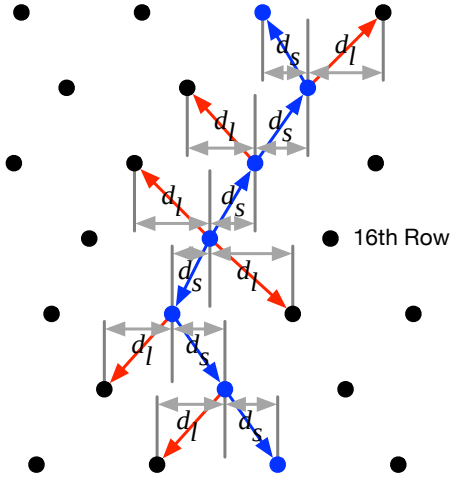


FIGURE 3.3: The rearrangement of point grid with horizontal installation offsets. The search starts from the only blue point in the sixteenth row, and the blue and red arrows indicate the candidates. The points with smaller angular differences *d_s* are selected as the nearest neighbors in the rearranged column and marked in blue. The points in blue are the final rearranged vertical scanline column.

3.1.2 Virtual Scanline Directions

In addition to checking the existence of polylines along the original scanline direction, which was the vertical data acquisition sequence in most of the TLS, W. Fan (2015) performed an extra check along the assumed scanline direction, which was

the horizontal direction against the rotation axis of the TLS base. In this thesis, the two diagonal directions were also considered in the feature point checking process. As shown in *Figure 3.4*, the azimuth and elevation angles of the laser beams were considered as grid coordinates. As the horizontal and vertical intervals in TLS point cloud acquisition were alterable, the grid resolution could be changed while still evenly distributed, which was different from the situation on a typical multi-line MLS, such as the Velodyne VLP-16 scanner. However, the change in angular resolution would not affect implementation of the ELS algorithm because not only the grid neighborhood relationship but also the distances between points in the clouds were considered in the feature extraction process.

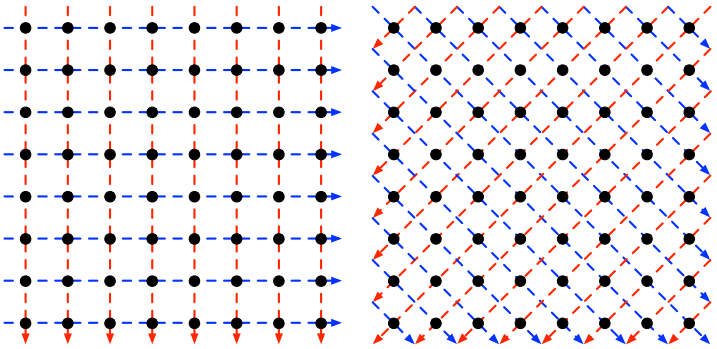


FIGURE 3.4: The applicable scanlines in an assumed point cloud captured by TLS. Points in black represent an evenly distributed point grid in both figures. (a) Original scanline distribution in the point grids. Dashed lines in red show the original data acquisition sequence while the dashed lines in blue represent virtual scanlines along the rotation direction of the TLS, which is the horizontal sequence of TLS point acquisition. (b) The distribution of the two diagonal virtual scanline directions in the point grids. The dashed lines in red and blue represent the two directions which are diagonal in the point grid.

3.1.3 Point Shifting and Projection

Because the beam emitter kept rotating during the period of firing the sixteen laser beams, the horizontal angles of the points were not the same as what happened on a TLS. According to Velodyne Acoustics Inc. (2015), if the highest spinning speed were configured, which was 20Hz for a Velodyne VLP-16 scanner, a 7200 °/*s* angular speed would be achieved. In other words, given the firing interval of 2.304 μs, the horizontal offsets between each of the laser beams and the first beam in the sixteen-point group could be calculated using *Equation 2.5*, and the results are listed in *Table 3.1*.

TABLE 3.1: The rearranged sequences, raw fire ID, vertical angles, and horizontal offsets at the rotation speed of 20 Hz / 1200 rpm from the first point of the group of Velodyne VLP-16 3D scanner according to Velodyne Acoustics Inc. (2015)

Rearranged Sequence ID	Fire ID	Vertical Angle (°)	Time Offset (μs)	Horizontal Offset (")	Sequential Offset (")
1	0	-15	N/A	N/A	N/A
2	2	-13	4.608	1.991	1.991
3	4	-11	9.216	3.981	1.991
4	6	-9	13.824	5.972	1.991
5	8	-7	18.432	7.963	1.991
6	10	-5	23.040	9.953	1.991
7	12	-3	27.648	11.944	1.991
8	14	-1	32.256	13.935	1.991
9	1	1	2.304	0.995	-12.939
10	3	3	6.912	2.986	1.991
11	5	5	11.520	4.977	1.991
12	7	7	16.128	6.967	1.991
13	9	9	20.736	8.958	1.991
14	11	11	25.344	10.949	1.991
15	13	13	29.952	12.939	1.991
16	15	15	34.560	14.930	1.991
1 (∗)	0	-15	55.296	23.888	8.958

(∗)Note: This is the first fire of the next firing loop showing the horizontal angle change between loops.

Given a maximum sequential horizontal offset of 12.939 " at the rotation speed of 1200 rpm, the corresponding offset in Euclidean distance could be estimated using the arc length, which was 0.038 m at 10 m range and 0.188 m at 50 m range. The offsets would be significant enough to affect the ELS results when the distance between the object and the scanner were larger than 10 m if the selected threshold value in ELS workflow was set to 0.045 m. Besides, this misalignment affects extraction along the two diagonal directions as well. Although this offset could be eliminated by shifting the lower or upper part of eight points into the neighboring column, there was no need for such implementation as this operation would, therefore, cause more severe misalignment problems. To reduce the curvature changes resulting from the uneven distribution in horizontal angles, a point shifting and projection operation was performed to enable the algorithm to concentrate on the variation analysis of the polyline within the scan plane.

Firstly, the projection plane was determined by the three selected points on the same scanline, which were the origin of the scanner and the two endpoints on the pending scanline. On an ideal scanline, the endpoints of the scanline were the first point in the rearranged sequence with the lowest elevation angle (-15 °) and the last point in the rearranged sequence with the highest elevation angle (15 °). In some scenarios, if there were either no first point or no last point as the points of the object might be too close or too far to the scanner and void points were generated, the neighboring point beside the endpoint would be used. If the neighboring point remained void, the replacement would keep going until there would not even be

three points forming a plane. Thus there would be no need for projection if only two or fewer points were valid in the group of the scanline. The process is shown in *Figure 3.5.*

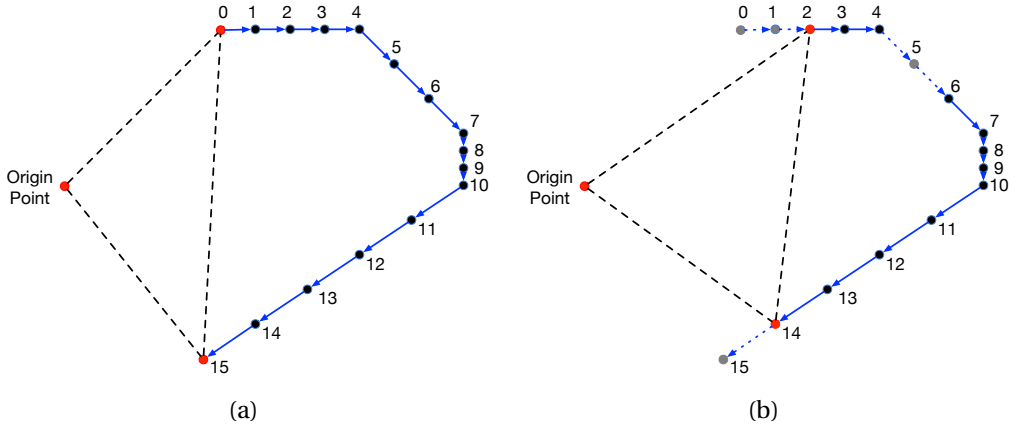


FIGURE 3.5: Profile views of the single column of points demonstrating the determination of the scanline plane. (a) In ideal scenarios, the origin point and the two endpoints in red are selected to determine the scanline plane. (b) When points in gray are omitted for certain reasons, neighbor points would be selected to build the scanline plane. The missing of points between endpoints will not affect the process of determining the scanline plane.

Once the scanline plane was determined, all points were projected to the scanline plane against the normal vector of the plane, and the projected points were used to extract feature points instead of the raw points. Consequently, any variations or components of variations against the directions of the normal vectors of the projection plane would be ignored in the ELS processing after the projection. The algorithm would therefore concentrate on analyzing the changes of shapes within the scanline planes. Such projections were performed on the group of the points on every single scanline, namely sixteen points for the vertical and diagonal scanlines. Meanwhile, there was no need for conducting such projection on them as the elevation angles of the points on the same horizontal scanlines were the same.

3.1.4 Double Feature Points in the Last Segments

Due to the inevitable noisy disturbance in the distance measurements and the estimated dynamic azimuth angles, the most representative feature points extracted might be second-degree feature points, as shown in *Figure 3.6*. In such circumstances, the false selection of the feature points might result in wrong line segmentation sections and diverted directions, which might affect the plane generation results as well. Consequently, a double feature point selection mechanism was added to the last segment generation process.

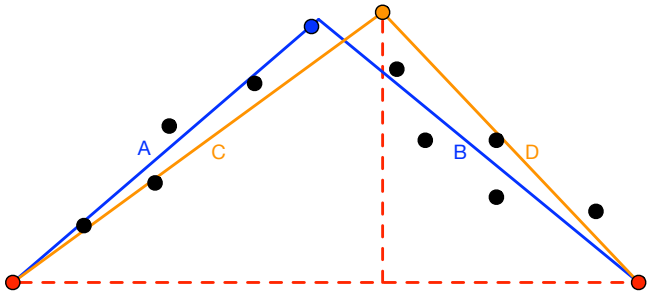


FIGURE 3.6: The feature point selected accidentally and the double feature points selected. In the original process, because the distance between the orange point and the line segments ending in the two red points is the largest, it is taken as the feature point. However, it is appreciated that the extracted segments C and D are less representative than segments A and B. Consequently, by selecting both blue and orange points as feature points, segments A and D would be selected as final line segments.

In the recursive feature extraction process, the lowest-level line segment consisted of two segments with no more subdivisions, such as the segment containing A and B in *Figure 3.6*. An extra selection process was conducted to check for the additional feature points. The points with the largest and second largest vertical distances were selected as the double feature points. Two groups of line segments of the two segments in each group were checked. The squared sum of distances

between each of the points between new endpoints and new sub-line segments, which could be understood as the squared sum of the residuals of the two groups of the new segments, were calculated. If the squared sum of the group of the second appreciable feature point, which was the point with the second largest vertical distance, were larger than the square sum of the group of the largest feature point, the second appreciable feature point would be taken as a feature point as well. Therefore, both points were selected as feature points in the ELS workflow.

The introduction of the double feature point mechanism decreased the possibility of wrong segmentation while at the same time probably reduced the length of the line segments. For segments with both feature points, there was a possibility that the segments were affected by the largest appreciable feature point as well, as the D segment in *Figure 3.6*. However, the directions of the segments were determined using the Singular Value Decomposition (SVD) analysis rather than the direction from one of the endpoints to the other, and the direction of the final segment would, therefore, be less affected. In the example given in *Figure 3.6*, the direction of the segment D would be close to the direction of segment B.

3.1.5 Significant Feature Points

Some points could not be distinguished as feature points in the ELS workflow, though they needed to be considered as feature points in reality. The points with zero coordinate—zero points—and the points that were far from their nearest neighboring points were considered significant feature points.

As the raw point grids were adopted in the ELS process, zero points were existing due to the presence of unmeasurable objects because the detector received no valid reflection. Such points were defined as the first kind of significant feature points as they could not be used to form any objects or segments. Moreover, points with significant distance to their neighboring points in the grid also exist in certain environments. The ELS workflow may not identify them as shape change points as the vertical distances between these points and the corresponding neighboring points might be smaller than the given threshold. Consequently, if the distance from the pending point to its neighbor points in the grid were larger than a given threshold, such as 1 m, it would also be considered a significant feature point. The determination of the value is subject to the resolution changes in the indoor environments and the extent of the corresponding environment. For typical indoor environments, the value could be set to triple the resolution as there might be one or two points missing due to the errors in scanners and changes in reflectance. For outdoor environments, the value may be set to 5 m, as there would be more points missing due to the variety of the object surfaces.

Significant feature points were introduced to exclude points with certain ambiguity from the group of the normal points because specific reasons made those points distinguishable. However, such features could not be identified using the criteria for identifying feature points in the ELS workflow. Consequently, those points were added as the significant feature points at the end of the ELS workflow.

3.1.6 Code Implementation

The implementation structure of the codes was altered to speed up the ELS process. The detailed revisions include (1) data format revising for more compact variables, especially the bitwise variable defining the feature characteristics, (2) iterator retrieving and searching algorithms by using point ID and data offset in Standard Template Library (STL) containers, (3) initialing and optimizing memory to run smoothly on low-RAM computers, (4) revising logical flow from simple loops to recursive calls of the kernel function, and (5) skipping large size of blocks in the enhanced Lang algorithms.

The optimizations succeeded in reducing processing time on the mentioned platform. The time used for extracting feature points from a point cloud of 1,612,070 points, shown in *Figure A.1(a)*, was reduced from 42'26" to less than 2", as the result shown in *Figure A.1(b)*. The optimization shorten the time used for ELS algorithm by about 95%, making the algorithm applicable for real-time applications. Consequently, the processing time for extracting feature points from the four directions of scanlines was reduced to milliseconds for the given frames of points, consisting of around 25,000 raw points per frame. The possibility of implementing GPU computing had been verified as well, and the results were that the parallel calculation of distance in processing point clouds could not be faster than CPU logic process with optimized spatial indexing method. Alternatively, the CPU-based parallel processing, OpenMP library, was still implemented in the processing codes (OpenMP ARB, 2015).

The overall flowchart of the implemented ELS algorithm is listed in *Figure A.2*, with the corresponding modification and revision facilitating it. The extraction results can be used to extract plane segments from low-resolution structured point clouds acquired by 3D SLAM scanner in S^2DAS. The detailed processing procedures and problems are discussed in *Section 3.2*.

3.2 Plane Extraction Based on ELS Extraction Results

As discussed in W. Fan (2015), regardless of the inhomogeneous resolution of the point clouds, the feature point extraction results can be used for plane segment extraction. This section discusses the plane extraction workflow based on the ELS extraction results. Firstly, some line segments were identified in the four given scanline directions by extracting feature points using the ELS algorithm. Next, plane fractions were generated by clustering line segments pointing to similar directions with limited distance between them. The plane fractions for different scanline directions were merged to generate large planes, which were the final outputs.

3.2.1 ELS Feature Extraction

The first step of the proposed plane extraction workflow was the extraction of feature points using ELS algorithm, introduced in detail in the previous section. The corresponding feature points and their indices in the respective scanline sequence were recorded for the following procedures. After this process, all feature points and

future procedures were conducted in the four parallel sections with respect to the horizontal and vertical scanline directions and the two diagonal virtual scanline directions until explicitly noted.

3.2.2 Scanline Segment Seeking

The scanline segment seeking procedure was along the sequence of the individual scanline. When the seeker came across a directional non-feature point, the point was selected as the first point in the line segment group. The seeking process continued until the next directional feature point was met, with all non-feature points added into the segment group. Then the segment was extended in both directions with the neighboring feature points added to the group as the endpoints of the segment. The seeking process is illustrated in *Figure 3.7*. After all the points were checked in this scanline segment seeking process along the four scanline directions, the process was concluded with the centroids and the direction vectors of each of the line segments estimated using SVD.

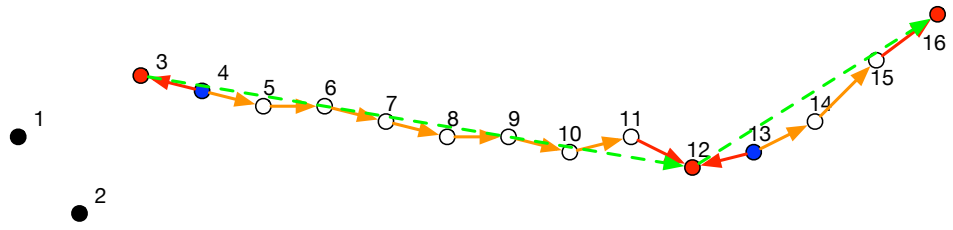


FIGURE 3.7: An example of the scanline segment seeking process. The numbers beside points label the rearranged scanline sequence while points in red and black indicate feature points and points in white for non-feature points. The seeking process starts at the blue points along the scanline direction marked with orange arrows and terminated at the last non-feature points. Then the segment is extended along the red arrow and ends at the feature points. The dashed arrows represent the directional vector of the scanline segments.

When a feature point was selected as the endpoints of neighboring segments, such as the middle red point labeled with 12 in *Figure 3.7*, an approximating selection process was performed to eliminate ambiguity. In this process, the pending points were grouped into segments with the smaller distance between the pending point and either of the line segments without the pending point, namely the subsets consisting of points 3-11 and points 13-16. If the distances to the two segments were the same, the pending points would be grouped into the segments with more points in its group. The pending point would be removed from the other segment, and the direction vectors and centroid would be updated as well.

3.2.3 Scanline Segment Clustering

The segments identified cluster together to form plane fractions in the clustering process. In this procedure, an initial scanline segment was selected as the seed segment. The direction of the pending line segment next to the seed segment was checked, and the difference between them was calculated. With the difference smaller than a certain given threshold, the two segments would be merged as the initial segment, while all points on the two segments would be used to calculate the local estimated normal $\mathbf{n_{pln}}$. The corresponding threshold value was set to 90 ◦ for excluding gross differences.

The search continued with an extra displacement vector $\mathbf{v_{dis}}$ in addition to the direction vector $\mathbf{v_{dir}}$, as shown in *Figure 3.8*. $\mathbf{v_{dis}}$ was defined as the unit vector pointing to the centroid of the pending line segment from the current line segment, which

was already a part of the plane. The cross product of $\mathbf{v_{dis}}$ and $\mathbf{v_{dir}}$ was calculated and defined as normal vector of the newly added planar segments, i.e., $\mathbf{n_{new}}$. If there were any possibility that the pending line segment was not on the same plane as the current line segment as they already pointed to the same direction, $\mathbf{n_{new}}$ would not be parallel to $\mathbf{n_{pln}}$. The purpose of introducing $\mathbf{n_{new}}$ defined by the point direction and the displacement vector was to eliminate misclassifying parallel line segments that were on a plane parallel but not overlapping, or a plane that was not parallel to the current plane but on a plane that was intersecting with the current plane, as shown in *Figure 3.9*.

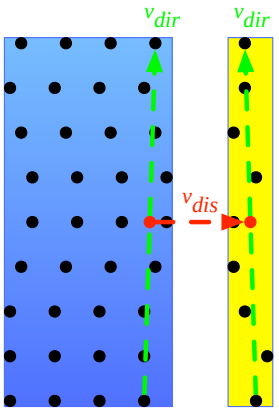


FIGURE 3.8: The definitions of the direction vector and the displacement vector of the scanline segments. The points in the blue box represent identified points on the same plane while points in the yellow box show a pending scanline segment. The points in red are the virtual centroid points of the two neighboring scanline segments. The dashed arrows in green show the direction vectors ($\mathbf{v_{dir}}$) of the neighboring scanline segments while the dashed arrow in red demonstrates the displacement vector ($\mathbf{v_{dis}}$) pointing from the centroid of the current scanline segment to the centroid of the pending scanline segment.

If $\mathbf{n_{new}}$ and $\mathbf{n_{pln}}$ were not parallel, there was an alternative check considering that the unavoidable noise might have affected the comparison. The displacement vector defined by each of the grid neighboring points on the current line segments

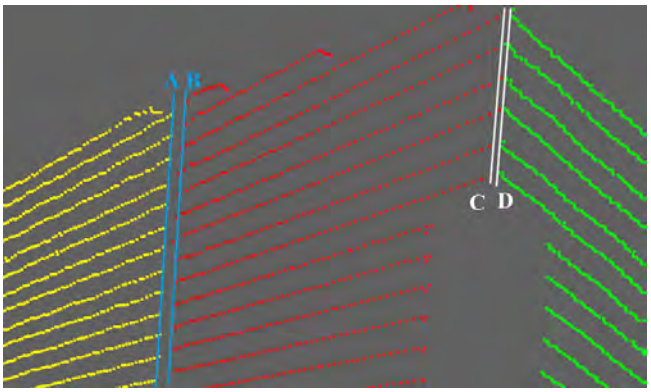


FIGURE 3.9: Parallel scanline segments which are not on the same plane. The points in different colors represent the three planes. The planes in yellow and red are parallel planes with a distance of 2 cm, while planes in red and green are perpendicular to each other. The blue line segments *A* and *B* are parallel segments, and the white segments *C* and *D* are parallel as well.

and the pending line segments would be calculated, i.e., $\mathbf{v}^{\mathbf{i}}_{\mathbf{dis}}$, as well as $\mathbf{n}^{\mathbf{i}}_{\mathbf{new}}$. If any of $\mathbf{n}^{\mathbf{i}}_{\mathbf{new}}$ was parallel to $\mathbf{n}_{\mathbf{pln}}$, $\mathbf{n}_{\mathbf{new}}$ is considered to be parallel to $\mathbf{n}_{\mathbf{pln}}$.

Furthermore, the neighboring rate would be checked if the two neighboring line segments were considered to be parallel and facing the same direction. The neighboring rate was defined as the number of points on the pending line segments which were grid neighbors of the current line segments versus the total number of points on the pending line segments, or vice versa, whichever was larger. Only if the neighboring rate were larger than a given empirical threshold would the two line segments be considered as eligible for clustering.

After all parallel line segments within a certain distance added into the group of points of the pending plane, the search process paused, and the plane was generated. As Principal Component Analysis (PCA) was sensitive to measurement noises, especially when the size of the subset used for estimating normal directions was not very large, the parameters of Hessian formula representing the pending plane were

then estimated using a RANSAC process with all disturbances filtered out as noises (Nurunnabi et al., 2016). The plane fragments were then reserved in the pending fragment group of the respective scanline directions.

3.2.4 Multi-direction Fragment Merging

As all the directional scanlines had been handled in parallel sections, the planar fragments were listed in the four groups. Their overlapping relationships were further analyzed. Some of the patches were merged in the fragment merging process. Merging was based on the examination of their normal directions: if two planes from different groups were facing the same direction, or if the angular difference between their normal directions was smaller than a given empirical threshold.

Once the fragments were considered as facing the same direction, the overlapping status was checked. Because the normal directions had been checked and the planes had been selected from different sequences of the same point cloud, the planes sharing at least one point in common were considered as sharing the same region and could be combined. This common point was identified in a binary search based on the sorted lists of point indices and was terminated once a common point was spotted.

Subsequently, the merged planes were reserved in a pending plane group and checked with the planes from other groups. The merging procedure was an iterative process. It kept running until neither common point nor parallel fragment could be

identified. In this process, neighboring planes which were separated due to unnecessary feature points identified in ELS workflow, such as the misclassified feature points from scanline curvature, would be merged as the possibility was low that the falsely identified feature points would affect more than one direction.

Furthermore, in case of false merging, the appearance times of the fragments in different groups were also checked. Only planes that were identified in more than two different directions were considered as valid planes. The overall flowchart of the plane extraction process is presented in *Figure A.3* and *A.4*.

3.3 Sample Tests and Results

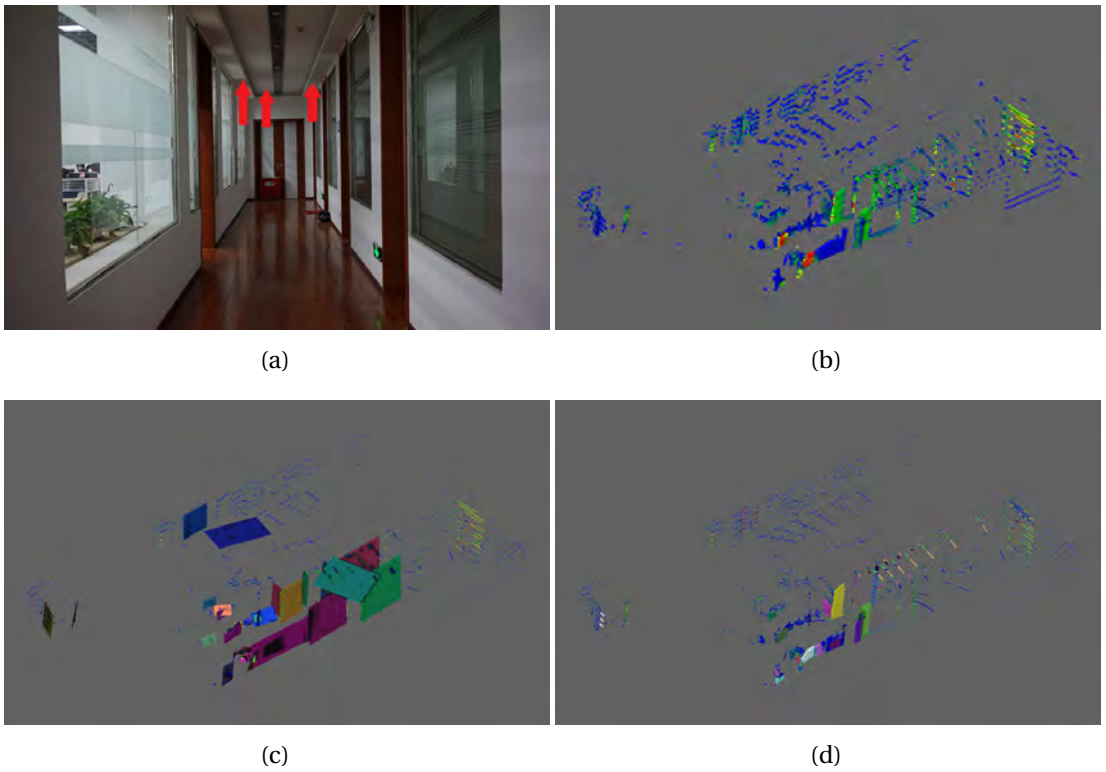
Typical indoor environments were used to test the performance and reliability of the proposed workflow, including corridors (simple and complicated), laboratories, large lecture halls, and stairwells. There are appreciable disadvantages in applying the state-of-the-art algorithms in extracting planes from the low-resolution inhomogeneous point clouds and some of the false extraction results had been listed in *Section 2.4*, and only results produced by the RANSAC workflow were compared in this section. Inevitable measurement noises affected the distribution of points, so the distance thresholds of RANSAC used in the direct RANSAC plane extraction and ELS-based method were set to the same value of 3 cm for comparison.

3.3.1 Corridor

Two corridors were selected as the testing site to test the effect of glasses on the two methods. The corridor with glasses and the dual-tier ceiling was more complicated for identifying all planes at the same time, while the simpler T-junction corridor was larger in scale.

Corridor with glasses for separating partitions was a typical office environment, as shown in *Figure 3.10*. The presence of glasses replaced the sidewalls, and the left parts were only low walls. Meanwhile, the dual-tier structure of the ceiling introduced long and narrow strips in the ceiling as well. Furthermore, the horizontally installed scanner could only capture points of ceilings and floors with long distances between neighboring rows of points, causing difficulties in understanding the point cloud. Three types of extraction results were selected for comparison, showing the differences in the results.

For the parts with high resolution, the two methods did not produce too many differences between the extracting results. However, significant differences exist in the extracting results of long strips and parallel planes. *Figure 3.11* gives an example of long strips of parallel ceiling sections which could not be extracted properly using the RANSAC method. One possible cause was the miscalculated normal vectors. As the normal vectors estimated were corresponding to the points in the spherical region with given distances, narrow strips might result in diverted normal vectors, generating the plane extraction results that only the center part of the ceiling could



(a) (b)

(c) (d)

FIGURE 3.10: The photo and point cloud of the corridor dataset, with the processing results of RANSAC and the proposed method. (a) Photo of the corridor dataset showing the position of the doors, sidewalls, and glasses, as well as the dual-tier ceiling (red arrows). (b) The single frame point cloud captured by Velodyne VLP-16 multi-line scanner. (c) RANSAC processing result with colored blocks showing the identified planes. (d) The result produced by the proposed method of which points in different colors represent groups of points extracted as planes.

be extracted as planes.

Extracting the Z-shape areas correctly was another challenge spotted. The RANSAC method leaned toward direct plane extraction results from the maximum subset, making the complete Z-shape area extracted with the two tilted planes wrongly extracted, as shown in *Figure 3.12(a)*. As for the connected Z-shape regions, there were possibilities that all the connected areas would be identified as planes with high levels of noises, as shown in *Figure 3.12(c)*. In the meantime, the proposed method showed acceptable results that only the respective planar regions

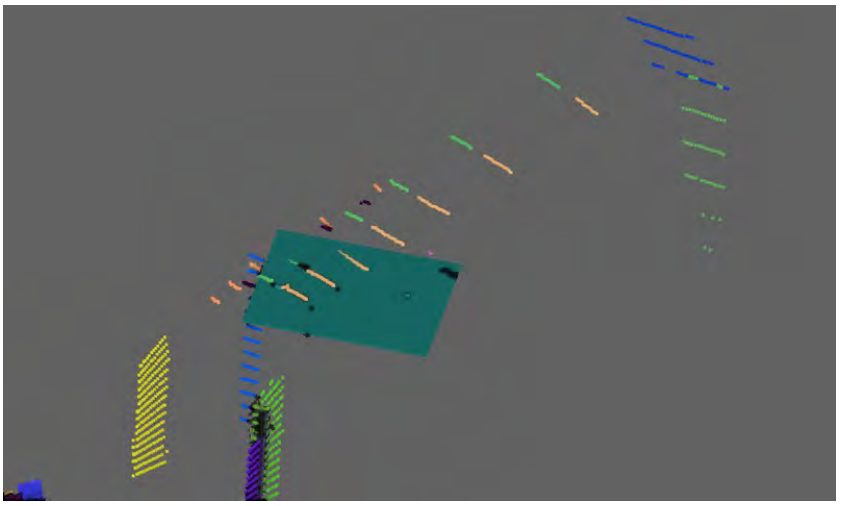
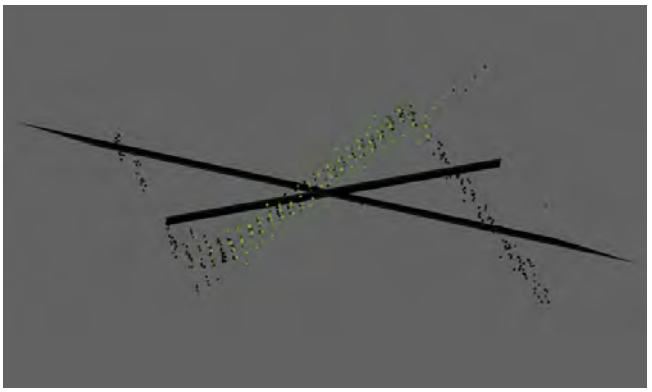


FIGURE 3.11: A part of the point cloud showing the RANSAC result and the result produced by the proposed method. The points in different colors show extraction results of the proposed method while the block in green is showing the only plane identified by the RANSAC method.

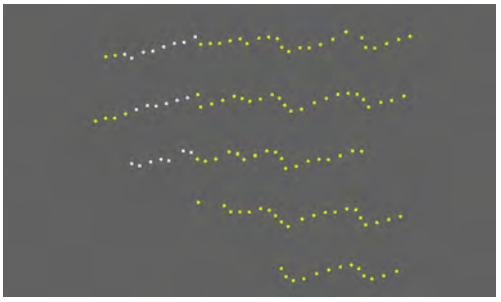
were selected as planes.

Adjacent parallel sidewalls with distances between them were not amendable for plane extraction using the RANSAC method as well, especially when the distances between them or the sizes of the neighboring planes were not too large. *Figure 3.13(a)* shows an example of the RANSAC method extracting adjacent parallel regions as a single plane, with all distance between adjacent planes being considered acceptable disturbances during the subset selection process. Meanwhile, *Figure 3.13(b)* shows the results using the proposed method, in which separated planes were extracted.

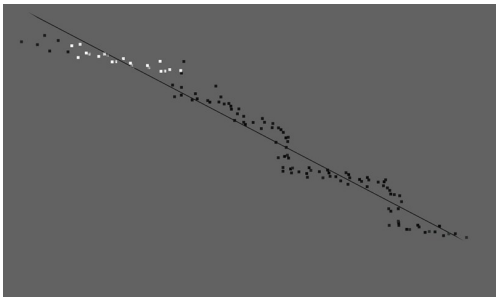
The comparison of the T-junction corridor data of a residence site did not show too many differences like the previous comparison, as shown in *Figure A.5* in *Appendix A*. However, because the testing site consisted of long, narrow corridors, the distance variations introduced changes in resolution along the vertical distribution



(a)



(b)



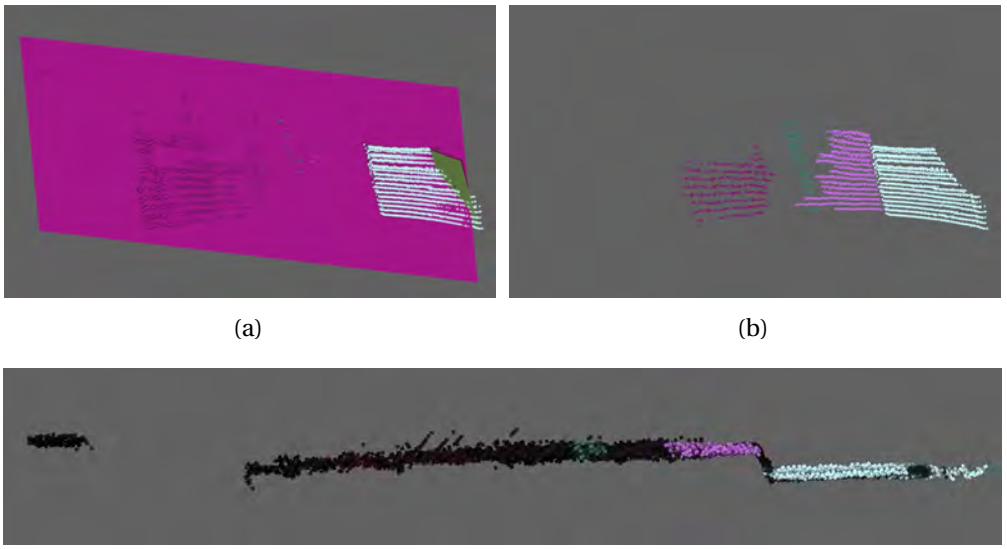
(c)

FIGURE 3.12: The point clouds and extraction results using the two methods of Z-shape point clouds. (a) The points and extraction results of both methods. The points in green represent the only plane extracted by the proposed method while all the points were used for extracting and estimating plane parameters of the two planes in RANSAC and two wrong results, represented by black lines, were generated. (b) A side view of the connected Z-shape region where only white points are extracted by the proposed method for estimating plane parameters while points in both green white and yellow are extracted as elements of the plane. (c) The top view of the points showing in (b) and the line representing the plane estimated by the RANSAC method.

of the scanner channels. It was clearly shown that the reduced resolution led to unsuccessfully plane extracting in RANSAC results, as shown in *Figure A.5(e)*.

3.3.2 Laboratory

The extraction results of a regulated room, which was a laboratory, showed differences in extracting specific planes as well. The RANSAC method extracted most of



(a) (b)

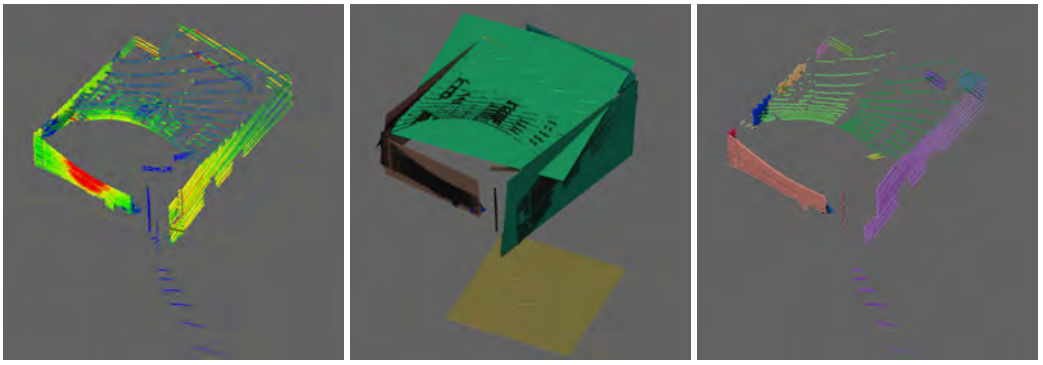
(c)

FIGURE 3.13: Extraction results of adjacent parallel planes. (a) The RANSAC extraction result shows the block area in claret using all visible points. (b) The extraction results using the proposed method in which the isolated planes were extracted. (c) The top view of the points identified as a single plane. All points, regardless of color, were used to estimate RANSAC parameters; while distance is significant between adjacent parallel planes in pink and white.

the planes from the point clouds. In addition, the proposed method could extract more small-sized planes in addition to the planes extracted by the RANSAC workflow, as shown in *Figure 3.14*.

As for differences in details, since there were motions within the point cloud due to the movements of the sensors, there were motion differences between the first and last captured points in a single frame of the point cloud. As mentioned above, the RANSAC method may consider adjacent parallel planes as a single plane with disturbances. The same problem would exist were the problem of corresponding motion difference to be spotted, as indicated in *Figure 3.14(c)* and *3.14(d)*. The proposed method produced no such failure since it already showed its capability in

(a)



(b) (c) (d)

FIGURE 3.14: The photo and the plane extraction results of the laboratory data using both the RANSAC method and the proposed method. (a) The panoramic photo of the laboratory. (b) The raw data for plane extraction with red indicating highest intensity and blue indicating lowest intensity. (c) The RANSAC extracting results with various blocks in different colors indicating the results. (d) The proposed plane extraction results with points in different colors indicating groups of points representing planes.

distinguishing such adjacent planes in the previous comparison. Meanwhile, the proposed method considered the fringe area as the two sides of the point grid, and there was no possibility for merging according to the working mechanism of the proposed method.

However, when planar objects reflect only two scanlines of points, the extracted

plane might be incomplete. The limitation was due to the working mechanism that the successful extraction of scanline segments was based on the identification of non-feature points between the two endpoints, requiring at least three points on a single scanline segment. An example is provided in *Figure 3.15*, showing that no non-feature point could be identified from three of the four scanline directions, resulting in no non-feature point for generating a valid scanline segment. Therefore, results indicate that the planes extracted using the proposed method might not be as universal as the RANSAC methods.

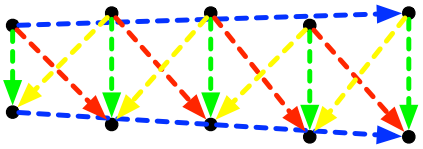
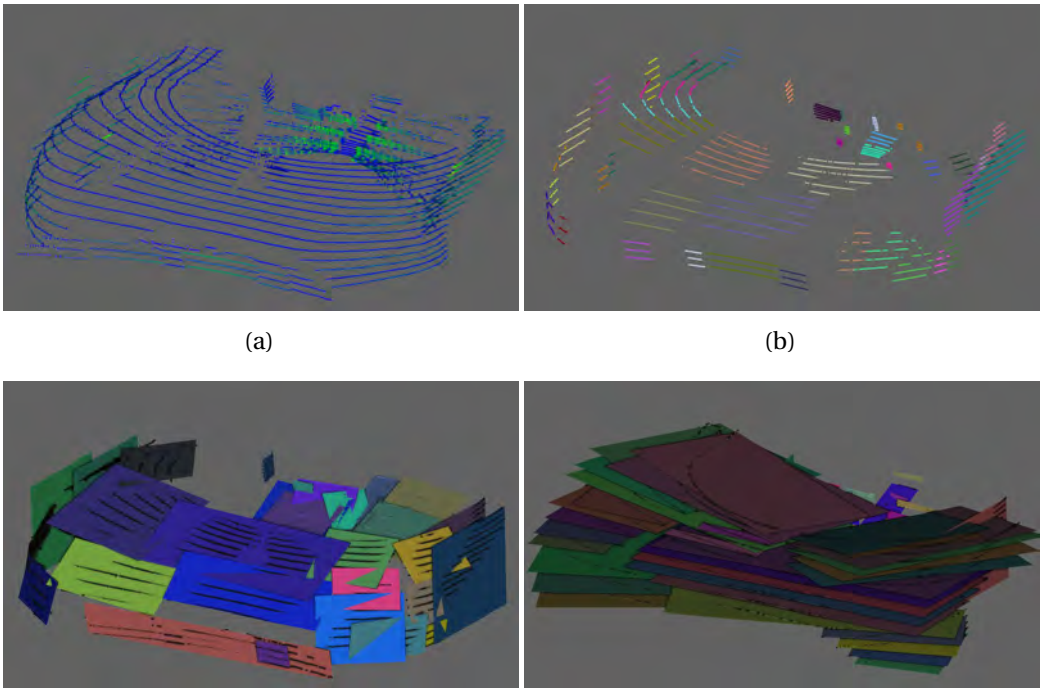


FIGURE 3.15: An example of two-line planes that cannot be identified by the proposed method. Three of the four scanline directions, in green, red, and yellow, cannot produce non-feature point for valid scanline segments.

3.3.3 Large Lecture Hall

The crucial challenges in extracting planes properly from the single-frame point cloud of a large lecture theater included the estimation of the local estimated normal vectors, the identification and separation of theater seats, and the division of the quasi-curved planes on the large theater ceiling. As the dimensions of the point cloud had been enlarged greatly, the distances between neighboring scanlines had been increased as well. Therefore, the correct estimation of the local normal vectors required the proper selection of the radius used for searching for local neighbors.

Otherwise, the planes extracted would be a series of parallel planes facing wrong directions, as shown in *Figure 3.16(d)*. Such results proved the necessity of gird normal estimation for such low-resolution inhomogeneous point clouds as it would avoid the improperly selected searching radius. Once normal vectors were estimated correctly, the planes with small directional differences, which formed the quasi-curved ceiling, could be identified and divided respectively using the RANSAC method, as shown in *Figure 3.16(c)*.



(a) (b)

(c) (d)

FIGURE 3.16: Raw points of the theater and the processing results using RANSAC and the proposed methods. (a) The raw points of the theater. (b) Extraction results produced by the proposed method with the groups of points in different colors representing different planes. (c) The RANSAC extraction result produced by local normal estimated with a long radius. (d) The RANSAC extraction result produced by local normal estimated with a short radius. The ceiling part was separated as multiple planes. Every plane segment consisted of one scanline segment.

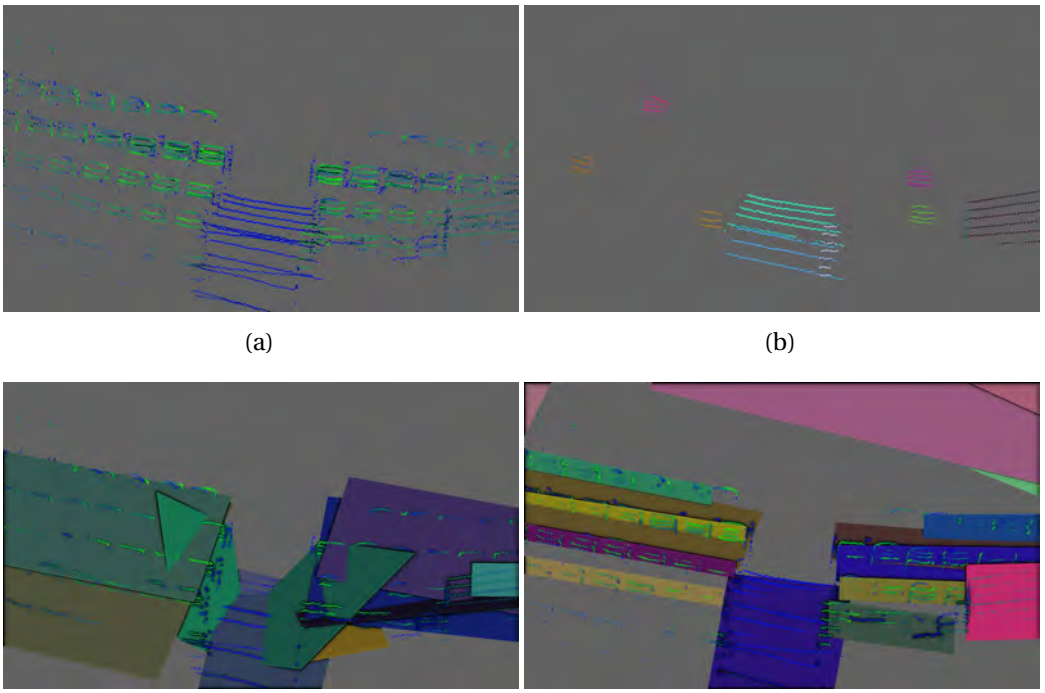
Meanwhile, the local normal estimation was affected by the radius problem as

well. As shown in *Figure 3.17(c)*, too large radius values would result in wrong normal estimations which turned the grouping of points on seats into a process of estimating an inclined plane with all seat points considered as disturbances and noises. When proper normal vectors could be estimated, as shown in *Figure 3.17(d)*, the working mechanism of the RANSAC method would connect all points reflected by the seats on the same row as the same plane, while the proposed method would only be capable of extracting some of the chairs, as shown in *Figure 3.17(b)*. Therefore, both methods did not produce satisfying results of extracting such small planes from large-scale point clouds.

3.3.4 Stairwell

A single-frame point cloud of the stairwell was used to test the proposed workflow as planes would be used for point cloud alignment in such environments. As the testing environment was quite small, the resolution of the point cloud was much higher than the resolution in other cases. The inhomogeneous distribution of the points on the sidewalls was not severe enough to affect the estimation of the local normal vectors, making them more precise as well. As shown in *Figure 3.18*, there was no great difference in extracting such large and appreciable planes except the aforementioned problems in extracting the two adjacent planes with distances and the small planes, as shown in *Figure 3.18(c)*.

However, both methods were not capable of identifying each of the steps of stairs instead of a plane across multiple steps, as shown in *Figure 3.19*. The proposed

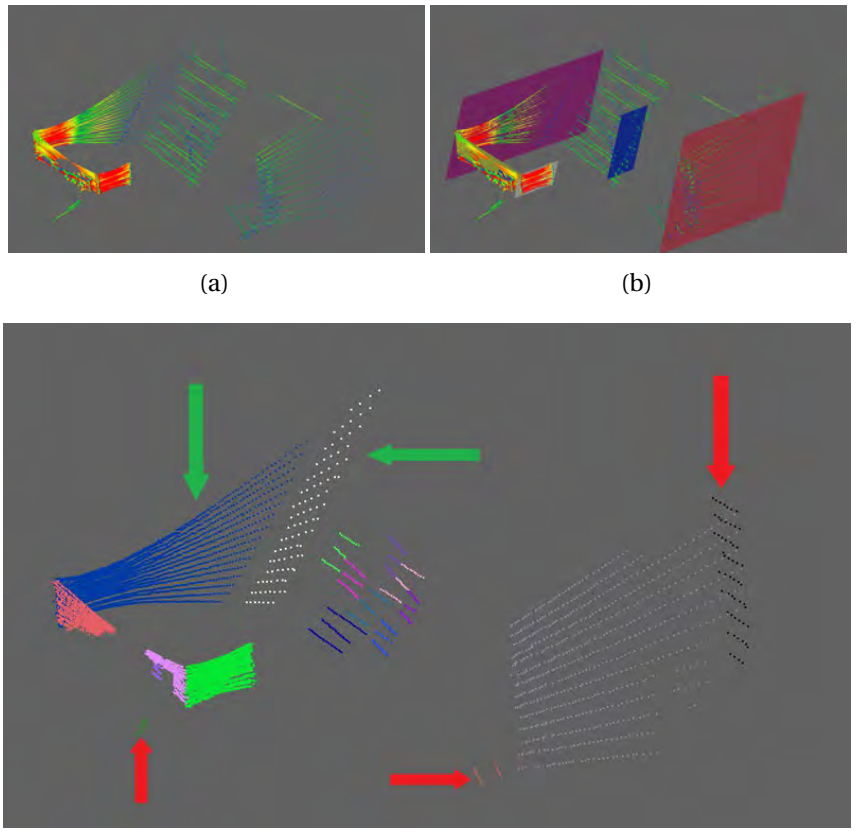


(a) (b)

(c) (d)

FIGURE 3.17: The raw points of the theater seats and the processing results using RANSAC and the proposed methods. (a) The raw points of the theater seat area. (b) The extraction results produced by the proposed method, with only five of the seats successfully extracted. The two larger planes in light blue show the successful extraction of two steps on the aisle. (c) The RANSAC extraction result produced by local normal estimated with a long radius. Points of seats of neighboring rows are grouped as plane points with noises and disturbances. (d) The RANSAC extraction result produced by local normal estimated with a short radius. Points of seats of the same row are grouped as plane points together rather than as individual seats.

method extracted small fragments instead of the large plane in *Figure 3.19(b)*, while the results were not completely correct. However, as the changes in point positions between frames would result in minor changes in plane parameters and cause alignment errors in the SLAM process, the small fractions might be easier to exclude as they cannot be aligned, which was easier for the SLAM process.



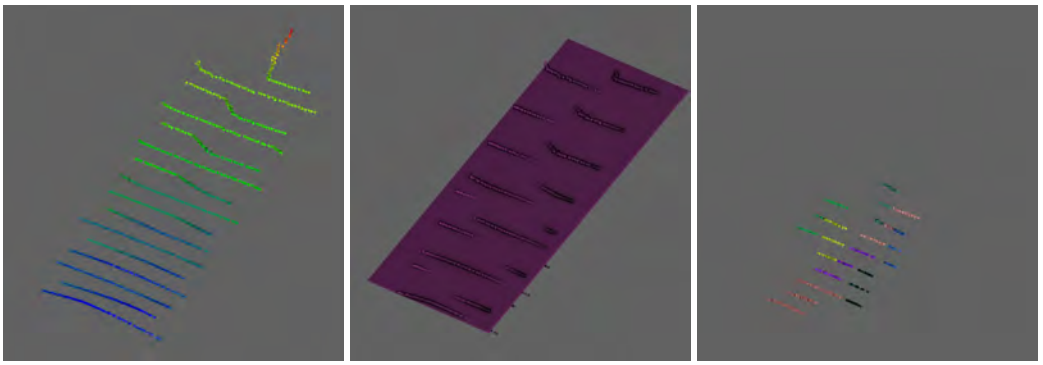
(a) (b)

(c)

FIGURE 3.18: The raw points of the stairwell and the processing results using RANSAC and the proposed methods. (a) The raw points of the stairs. (b) The RANSAC extraction result in which all stair points were grouped into the same plane. (c) The result produced using the proposed method. The planes labeled with green arrows show the planes that were successfully distinguished while the planes labeled with red arrows indicate the small planes extracted.

3.4 Summary

By comparing extraction results and examining limitations, the proposed ELS-based plane extraction algorithm demonstrated its contribution in processing low-resolution inhomogeneous point clouds. The algorithm showed its advantages in processing point clouds of long strips with small height differences, individual or connected Z-shape areas, adjacent parallel planes, planes with inhomogeneous



(a) (b) (c)

FIGURE 3.19: The raw points of the staircase and the processing results using RANSAC and the proposed methods. (a) The raw points of the stairs. (b) The RANSAC extraction result in which all stair points were grouped into the same plane. (c) The result produced using the proposed method where small fractions of planes were extracted.

point resolution, and small-size planes. The limitations which were not properly handled, such as the false feature points due to scanline curvatures, biased feature points resulting from the inflexible segment slicing rather than an adaptive threshold, incapability of extracting planes consisting of scanlines with only feature points, and the respective non-parallel parameter estimation process in code implementations, have been discussed.

With these pros and cons, the proposed algorithm was a more reliable workflow in processing low-resolution inhomogeneous point clouds. Since the scanners installed on S^2DAS were multi-line MLS, which produced such point clouds, the extraction of planes based on ELS feature point results had been implemented in the form of C++ codes and packaged as a compact library for applications to the plane-based SLAM process, which was discussed in *Chapter 4*. Indeed, the ELS-based plane extraction algorithm could be implemented in a lightweight FPGA platform

for easy and rapid implementation in any corresponding platforms.

Chapter 4

ELS-based 3D Point Cloud Alignments

Various feature-based SLAM workflows were reviewed in *Section 2.5*, and the plane-to-plane point cloud alignment methods discussed. However, the unreliable planar segmentation limited the performance of most methods, and these methods did not show acceptable results, given the low-resolution inhomogeneous point clouds.

Based on the plane extraction methods designed for point clouds captured by multi-line MLS, a new point cloud alignment process was proposed to register the point clouds captured on a mobile backpack platform to the same reference frame. The process was similar to the SLAM process, but it was not a real-time processing workflow. Multiple versions of dual-scanner systems were designed to test the performance of the proposed method, and several on-site tests were conducted in typical scenarios on the university campus.

In this chapter, the design of the mobile mapping backpack is first introduced,

and the dedicated time synchronization methods between the scanners explained. Secondly, the ELS-based point cloud alignment workflow is presented. A few sample tests are conducted, and the results shown in the third section. The overall discussions on the results and the performance of the workflow are listed in the final section.

4.1 S^{2}DAS: A Seamless Mobile Mapping Backpack

With the continuous urbanization in the world, the complexity and heterogeneity of spatial data in the vertical dimension raised new challenges to the traditional surveying techniques. As shown in *Figure 4.1*, conventional technologies developed for 2D and 3D spatial data acquisition provided remote data acquisition methods for collecting both geometry and attribute information of ground objects and enabling data manipulations in GIS. These technologies include satellite remote sensing and photogrammetry, airborne remote sensing and photogrammetry, and oblique photogrammetry. Such remote sensing methods offered geographic data in various scales from different platforms on varying altitudes. In addition, conventional surveying techniques, such as leveling, control and detail surveying, close-range photogrammetry, and vehicle-based mobile mapping system, made it possible for close-range surveying on the ground.

However, for more than 80% of human activity environments, such as indoor environments and congested areas in cities, effective and efficient methods and

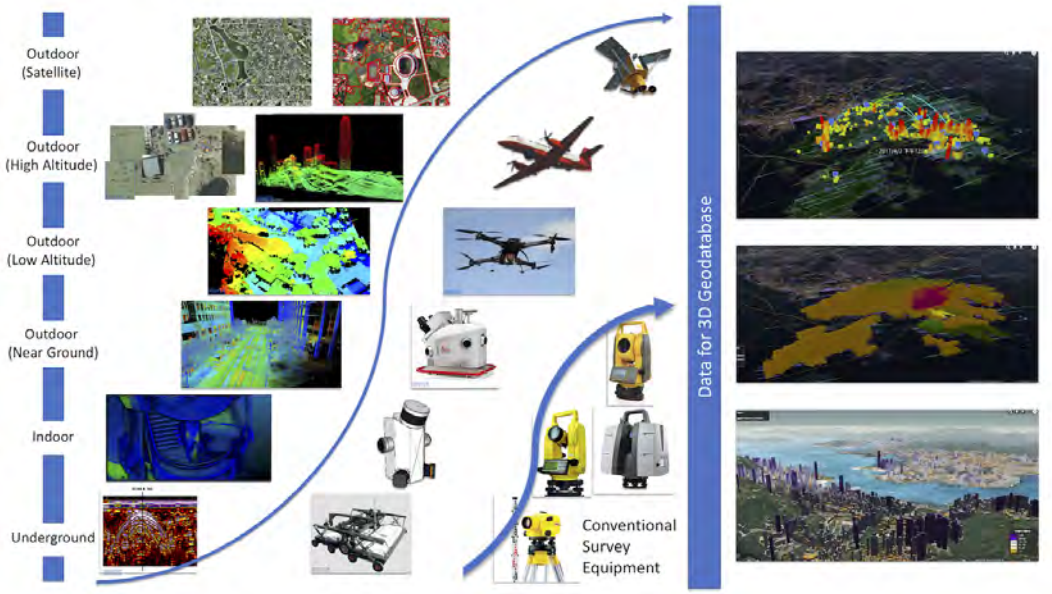


FIGURE 4.1: Data capturing techniques for building 3D geodatabases. The figures on the left side show the data captured at different altitudes. The pictures in the middle part illustrate the various kinds of equipment and platforms for the data on the left side. The figures on the right present a few examples of the 3D geodatabase applications.

solutions for acquiring and recovering the real world in virtual environments are still being developed. We proposed and designed S^2DAS, a backpack mobile mapping system to capture geometry data and texture data for 3D GIS modeling in indoor environments, to fulfill the requirements of indoor 3D spatial data acquisition and support the development of 3D GIS. By synergizing advanced spatial information technologies, the multi-sensor integrated system captures location and textural information for indoor and congested 3D city environments, and provides a preprocessing toolkit for fusing multi-source data as an effective and reliable data capture device. As shown in *Figure 4.1*, S^2DAS and other mobile mapping solutions are essential parts of modern spatial data acquisition workflow.

Based on the degree of complexity of the working environments, the developments and the long term roadmap of S^2DAS were divided into three stages: 1) **building-wide stage** with backpack mapping platform and data processing algorithm and toolkits for indoor environments, 2) **citywide stage** with seamless backpack mapping platform for both indoor and outdoor environments, concentrating more on the algorithms and methodologies for complicated cross regions of city canyons and open areas, and 3) **smart city stage** with vehicle- and UAV-based seamless mobile mapping platform enabling multi-platform seamless mapping technologies for various urban environments. Implementing the three phases would enable effective and efficient 3D seamless geodatabase producing, processing, management, and geo-visualization.

The development of S^2DAS was part of the project entitled *Develop 3D Geodatabase Framework for Hong Kong—A Lightweight 3D Seamless Spatial Data Acquisition System (SSDAS)*, which was supported by the *Innovation and Technology Support Programme (ITSP)* of Hong Kong SAR.

4.1.1 Hardware Design for Testing the LiDAR Point Alignment Workflow

As discussed in *Chapter 2*, multiple sensor integration was the most popular and appropriate choice for generating indoor point clouds. The adoption of a single sensor

usually comes with omission and inevitable errors. In the proposed platform, multiple sensors not only facilitated the use of initial values and constraints for improving the processing efficiency and accuracy but also provided various kinds of data for building the 3D models of the objects.

Taking SLAM methods as an example, dual systems were implemented in S^2DAS for SLAM data acquisition. The primary system was based on two multi-line MLS generating low-resolution point clouds, and the LiDAR SLAM workflows were conducted in real-time and post processing. Meanwhile, optical images captured during motion were used as the data source for vSLAM, the secondary SLAM system. Algorithms were designed and modified for cooperative working with laser scanners and improving localization and orientation precision in post processing with the assistance of MEMS IMU to accomplish the integrated multi-sensor combination.

To test the performance of the proposed point cloud alignment workflow, a simplified version of S^2DAS was designed. The system consisted of two multi-line laser scanner, a Velodyne Puck with 2 ° vertical interval and a Velodyne Puck Hi-res with 1.33 ° vertical interval, a simulator that kept outputting timing signals similar to a GNSS module in indoor environments, a switch for Ethernet communication, a data logging PC, and a dedicated battery unit.

Initially, the two scanners were named SLAM scanner and PCD scanner as it was considered that the former would be responsible for 6 DOF SLAM and the latter would be used purely for gathering point clouds of surrounding environments. The

SLAM scanner was installed over the top of the operator's head, generating point clouds covering large areas horizontally. The PCD scanner was installed nearly perpendicular to the ground to capture point clouds of vertical spaces. *Figure 4.2* shows the point clouds captured by the two scanners. However, it was soon discovered that both scanners should be adopted for the SLAM process to generate satisfying results because the SLAM scanner could not capture horizontal planes in some environments. Meanwhile, the results of the PCD scanner were always better: slight attitude errors would be amplified with the wider distribution of point clouds captured by the SLAM scanner. Therefore, the names were maintained as point clouds captured by both scanners were used for SLAM while only the points captured by the PCD scanner were used for merging and generating the final results.

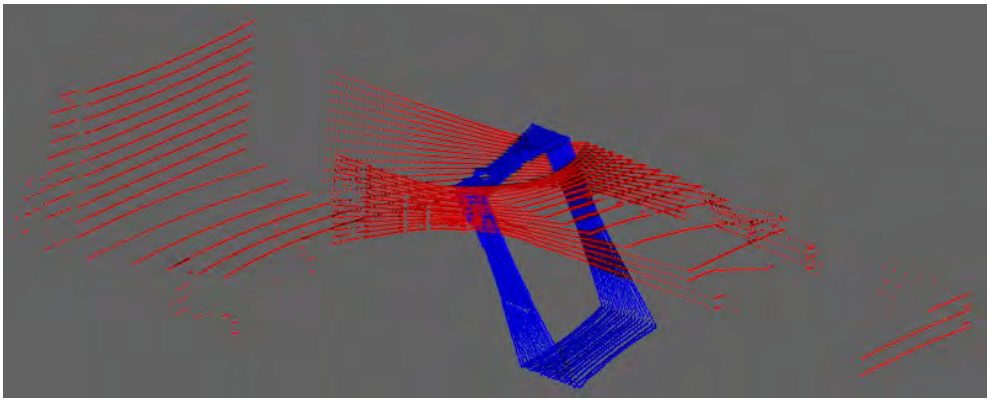


FIGURE 4.2: A single frame of point cloud captured by two laser scanners installed with angles between them. The points in red were captured by the horizontally installed 16-line scanner, the SLAM scanner, while the points in blue were captured by the vertically installed 16-line PCD scanner. The model of SLAM scanner is Velodyne VLP-16 with 2 ° vertical resolution while the PCD scanner is Velodyne Puck Hi-Res with 1.33 ° vertical interval.

4.1.2 Data Processing Workflow and Software Design

Given the plan of implementing LiDAR SLAM, vSLAM, and IMU DR on S^2DAS, the workflow of S^2DAS is shown in *Figure B.1* in *Appendix B*. The workflow was designed at the beginning stage of the project and updated whenever modifications were needed with respect to any specified requirements and design changes.

The basic concept of the workflow was a sequence of processing procedures. The dual laser scanners were adopted as main SLAM positioning and orientation data source. The vSLAM process was conducted offline after all data were captured. The SLAM results were passed to Extended Kalman Filter (EKF), correcting IMU drifts and generating more reliable results. The results were used to correct the motion changes and deformations in the point clouds. Subsequently, the second-pass LiDAR SLAM was conducted. The rectified SLAM results were passed to the EKF, with vSLAM results and IMU readings, for the final trajectory and the georeferenced point clouds and images.

Meanwhile, in the point cloud merging process, the point cloud generated in the vSLAM process would be combined with the LiDAR point cloud colorized by the geo-referenced images. Consequently, the colorized point clouds were provided in the modeling process in which the 3D models of indoor environments were generated.

4.1.3 Time Synchronization

When combining point clouds, either mobile mapping or static point cloud, the alignment and co-registration of the sensor coordinate frames was a vital process. This process recovers the geometric relationships between coordinate frames of each scan stations and aligns geometric data of stationary objects to the same coordinate frame. This co-registration process is inevitable and more important for moving platforms and sensors as data acquired by various sensors need to be fused into the same coordinate frame. However, there may not be enough features to align different kinds of data.

Consequently, time alignment is a critical requirement for system integration because all sensors are installed on a moving platform whose attitudes and positions keep changing while their positions and orientations keep the same as a rigid body. Provided no external references, time references could be facilitated as the initial reference for recovering the alignments, with the known geometric relationships. Therefore, if there was any misalignment in time, the time differences between different sensors would cause misalignment and errors. Two kinds of synchronizations were carried out in the system: synchronizations based on a dedicated hardware device providing timing signals and the frame slicing and synchronization of laser scanners.

4.1.3.1 Dedicated Hardware GNSS Simulator

According to Velodyne Acoustics Inc. (2015), the Velodyne Puck series scanners adopted GNSS timing signals as the external reference for time synchronization. The scanners captured the pulse signal from the GNSS module once per second, i.e., 1 PPS, to achieve the precise synchronization with the external reference. When the external signals were absent, the internal clock with a drift of 5 s for every 24 hours, namely 57 ns per second, would provide the internal time reference for the points captured. However, the drifts between the internal clocks of scanners could not be the same as they were random drifts, while such amounts of time were appreciable in calculating the position of the points. Therefore, a GNSS simulator was implemented in the system, providing a consistent time reference for the two laser scanners in the forms of 1 PPS and the corresponding time tags.

4.1.3.2 Synchronization between Laser Scanners

The synchronization between the two laser scanners was different from other sensors, as it was a section splitting problem rather than direct time alignment. Originally, the continuous point cloud capturing process was split into sections with respect to the 0 ° horizontal angles. Consequently, the time duration of the point clouds cannot be automatically aligned even when their internal time frames were aligned.

Time differences between sensor frames were always unavoidable, and the data

captured with time tags can be considered as corresponding data with constant time offsets once the stable time reference could be maintained, as shown in *Figure 4.3(a)*. Meanwhile, this principle could also be applied to every point in the point clouds captured by laser scanners as points were indeed time tagged with the data capturing time.

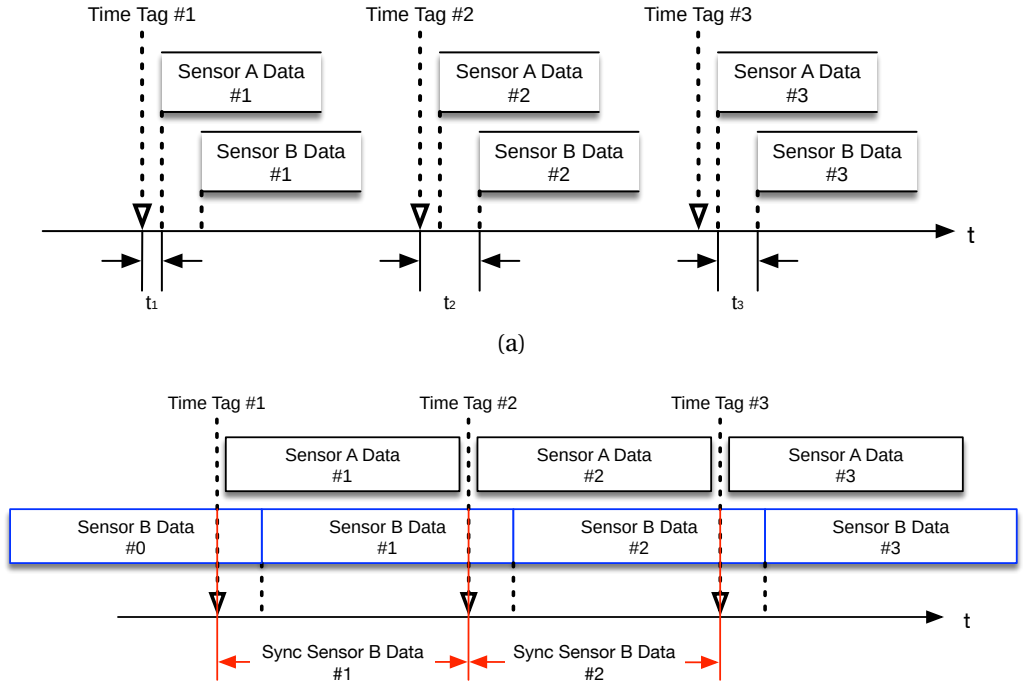


FIGURE 4.3: Time synchronization based on time tags. (a) Time synchronization between normal sensors based on single time tags. The time offsets between the real time given by sensors and the given time frame (t_1 and t_2) and between sensors (t_3) kept changing within a small range, which was considered as synchronization errors and drifts. (b) Time synchronization between point clouds captured by different scanners. Instead of recovering time-to-time correspondence between each single points, synchronization was based on section slicing on continuous data streams. The block with blue boundaries shows the original stream slices based on horizontal angles of points (0-360 °) while red separations illustrate the time intervals used for slicing when aligned to the corresponding time slots of sensor A.

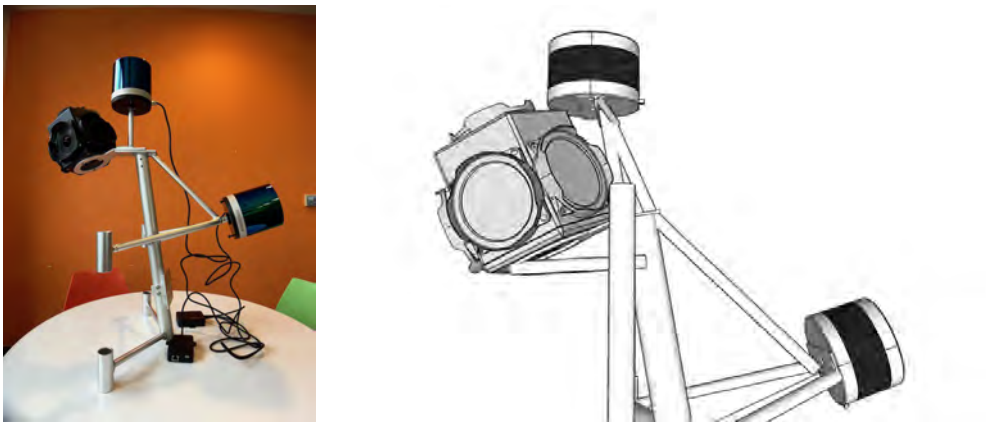
Consequently, the synchronization between laser scanners was converted into a section slicing problem that points captured in the same time slots were considered

as corresponding frames of points rather than splitting the continuous point capturing process with respect to the horizontal angles of each single point, as shown in *Figure 4.3(b)*. Therefore, the two point clouds captured by different scanners during the same time slots were considered as synchronized point clouds of the same frame.

4.1.4 Prototypes for Algorithm Verification

The conceptual hardware design of the initial verification prototype is shown in *Figure B.2* in *Appendix B*. This prototype could barely meet the requirements of lightweight and compact. However, the large-sized hardware frame could provide plenty of installation positions and angles for adjustments and modifications for developing a system with higher feasibility and better performances. Based on this hardware design, an updated version of installation frame based on a hiking backpack frame was designed, as shown in *Figure 4.4(a)* and *B.3* in *Appendix B*, while the photos of an operator carrying the backpack are given in *Figure B.4*.

Six camera lenses were integrated into the Ladybug camera, the optical sensor installed on the platform for testing the vSLAM workflow, and a significant part of FOV was blocked by the base plate of the camera, which was similar to the bottom blocked area of TLS. Therefore, as shown in *Figure 4.4(b)*, there was a titled angle between the base plate of the panoramic camera and the assumed horizontal plane, making one of the six camera lens to face downwards, to capture features near the



(a) (b)

FIGURE 4.4: The side view of the installation frame with only the scanners and the panoramic camera installed. (a) The complete frame showing the positions of the scanners and the camera. (b) The detail of the higher part shows that the camera was installed obliquely for better coverage in the front direction.

carrier and in front of the backpack, and fulfill the data capturing requirements in narrowed spaces, such as stairwells and corridors.

In addition to the self-obstructing problem, the blocked FOV caused by other sensors and the backpack installation frame needed to be considered as well. As the sensors, i.e., the laser scanners and the panoramic camera, were installed on the same frame, they needed to be distributed in a compact space to reduce displacement and errors due to the deformation of the installation frames. However, the congested installation space may cause severe obstruction since they would appear in the FOV of other sensors, and they might be considered as tie points between frames to generate false geometric relationships. Obstruction to the MLS would result in the blocked FOV, which further reduced coverage of the scanners which was already incomplete. *Figure 4.5* provides an example of the FOV of the panoramic camera, which was partly blocked by the laser scanner above it.

FIGURE 4.5: The example of neighboring sensors blocking part of the panoramic FOV of the camera. For the six cameras numbered from 0-5, the top right and bottom right corners of camera 2, the top left and bottom left corners of camera 3, and the lower part of camera 5, which are all marked with red circles, are blocked by the two laser scanners beside the camera lens.

In the proposed design, the FOV of both the camera and the scanners were plotted, and their installation positions were adjusted to minimize the obstructed zones, as shown in *Figure 4.6*. Eventually, considering the size of the backpack and that height of the backpack and the carrier might be too high to move around in indoor environments, there were still obstructions in the camera FOV in the final design.

A simplified version of a mobile mapping helmet was designed to conduct the initial LiDAR SLAM algorithm test. Only the two laser scanners were installed on the helmet, with the same installation angle as designed on the backpack, as shown in *Figure 4.7*. The time synchronization signals were provided by either an external GPS or the dedicated synchronizer.

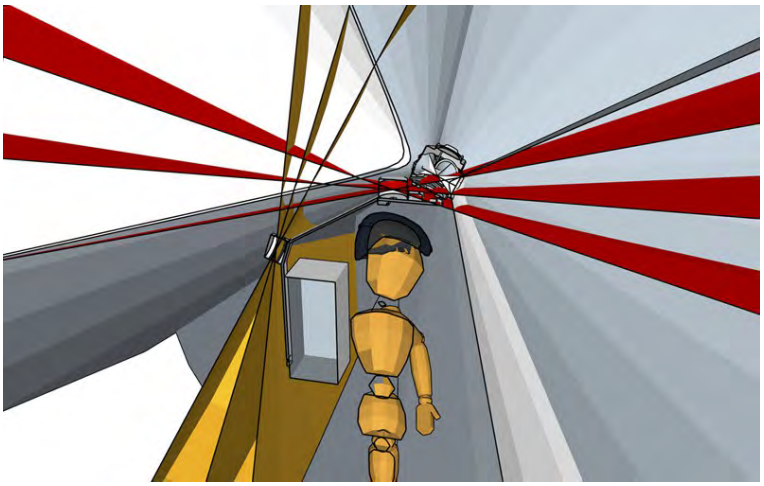


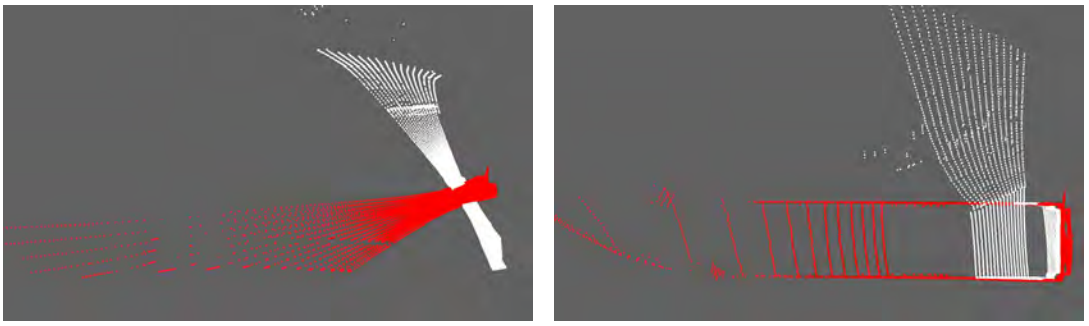
FIGURE 4.6: The example of FOV-based sensor position adjustment. The region in red is the FOV of the horizontally installed laser scanner while the region in yellow is for the oblique scanner. The region in white and gray is the FOV of the six camera lens of FLIR Ladybug 5Plus. It is shown that the camera has obstructed the right part of FOV of the horizontal laser scanner while the horizontal laser scanner also blocked small parts of Ladybug's FOV.

(a) (b)

FIGURE 4.7: The compact helmet design used for verifying LiDAR SLAM algorithm. (a) Rear view with 45 ° angle. (b) Front view with 45 ° angle.

As shown in *Figure 4.4(b), 4.6* and *4.7*, the angle between the axes of the two scanners is nearly 76 °, increasing the probability of capturing horizontal planes in narrow spaces, reducing the possibility of capturing the moving human heels, and making the two overlapped point clouds with extended coverage at the same time. *Figure 4.8* provides an example on the FOV of the proposed installation method,

in which it is clearly shown planes facing multiple directions could be captured, enabling the 3 DOF estimation of the 3-axis rotation and the 3 DOF estimation of the 3-axis translation between consecutive frames.



(a) (b)

FIGURE 4.8: The side view and top view of a single frame of the point cloud captured by the scanners installed on S^2DAS. The points captured by the horizontal scanner are in red, while the points captured by the other scanner are in white. (a) The side view showing the difference in coverage. (b) The top view showing the complementary coverage.

4.2 ELS-based Point Cloud Alignment Workflow

In the proposed method, plane-to-plane registrations were implemented to solve alignment problems between consecutive frames, while extra alignments between frames, which were not adjacent and used as redundant observations, were added in the proposed workflow to enhance the network building for adjustment, which was similar to the architecture introduced in Dai et al. (2017).

4.2.1 General Alignment Process

In this report, the successful alignments established between frames of points were called observations. In other words, if the relationship was built between two given point clouds, the estimated rotation and translation parameters, in the form of a 4×4 matrix as listed in *Equation 4.1* and *4.2*, were defined as an observation representing the position and orientation difference between the two frames.

$$\boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\to i}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad \boldsymbol{T}^{\boldsymbol{j}}_{\boldsymbol{j\to i}} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{4.1}$$

where:

$\boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\to i}}$ is the 3×3 rotation matrix of which the target frame is the $i-$th frame, the source frame is the $j-$th frame and the current coordinate frame before transformation is in the $j-$th frame, and

$\boldsymbol{T}^{\boldsymbol{j}}_{\boldsymbol{j\to i}}$ is the 3×1 translation vector of which the target frame is the $i-$th frame, the source frame is the $j-$th frame and the current coordinate frame before transformation is in the $j-$th frame.

$$\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j\to i}} = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\to i}} & \boldsymbol{T}^{\boldsymbol{j}}_{\boldsymbol{j\to i}} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

where:

$\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j\rightarrow i}}$ is the 4×4 transformation matrix of which the target frame is the $i-$th frame while the source frame is the $j-$th frame and the current coordinate frame before transformation is in the $j-$th frame.

Consequently, given the correspondence between the planes extracted from two frames of point clouds, the rotation matrix and the translation vector can be estimated. To reduce the number of parameters invoked in the process, axis-angle parameters was introduced to maintain the unique transformation and the correspondence, rather than the Euler-angle parameters which would easily be affected by the rotating sequence, as shown in *Equation 4.3*. Besides, the unique correspondence between the rotation matrix and the axis-angles could be facilitated for the derivation-based non-linear optimization in the pose estimation process. Given the parameters representing all the corresponding planes in the pairwise combination in *Equation 4.4*, solving the rotation matrix between the two frames of point clouds could be converted to solving the axis-angles, considering the two combinations of plane normal directions. It was the question of how the normal directions of the planes in frame j could be rotated to coincide with the corresponding normal directions of the planes in frame i, as shown in *Equation 4.5*.

$$\boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\rightarrow i}} \leftrightarrow \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{4.3}$$

where:

θ_1, θ_2 and θ_3 are the corresponding axis-angles with respect to the rotation matrix $\boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\rightarrow i}}$.

$$\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{k}} = \begin{bmatrix} a^i_k \\ b^i_k \\ c^i_k \end{bmatrix}, \quad \boldsymbol{N_i} = \begin{bmatrix} \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{1}} & \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{2}} & \dots & \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{l}} \end{bmatrix} = \begin{bmatrix} a^i_1 & a^i_2 & \dots & a^i_l \\ b^i_1 & b^i_2 & \dots & b^i_l \\ c^i_1 & c^i_2 & \dots & c^i_l \end{bmatrix} \tag{4.4}$$

where:

$\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{k}}$ is the normal direction of the identified plane k in frame i,

a^i_k, b^i_k and c^i_k are normalized parameters of the plane according to the Hessian formula, and

$\boldsymbol{N_i}$ is the parameter matrix consisting of all normal vectors of the corresponding planes in the frame i.

$$\boldsymbol{N_i} = \boldsymbol{R}^{\boldsymbol{j}}_{\boldsymbol{j\rightarrow i}}\, \boldsymbol{N_j} \tag{4.5}$$

In the estimation progress, the residual vector could be defined as the angular distances between the normal vectors of the planes in the target frame and the normal vectors of the corresponding planes after rotation, as shown in *Equation 4.6* and *4.7*. These residuals were converted to the difference between the dot product of the two normal vectors and 1, which was the cosine value of the maximum angle between the vectors after ignoring the facing direction. The problem was then defined as a non-linear process of optimizing the residual to the minimum values.

$$\boldsymbol{n}_{\boldsymbol{kR}}^{\boldsymbol{j}} = \boldsymbol{R}_{\boldsymbol{j\rightarrow i}}^{\boldsymbol{j}}\, \boldsymbol{n}_{\boldsymbol{k}}^{\boldsymbol{j}} \tag{4.6}$$

where:

$\boldsymbol{n}_{\boldsymbol{kR}}^{\boldsymbol{j}}$ is the rotated normal direction of the $k-$th plane in the $j-$th frame.

$$\epsilon_{kR} = 1 - \boldsymbol{n}_{\boldsymbol{k}}^{\boldsymbol{i}} \cdot \boldsymbol{n}_{\boldsymbol{kR}}^{\boldsymbol{j}} \tag{4.7}$$

where:

ϵ_{kR} is the residual between the normal directions of the $k-$th plane in the $i-$th frame, which is the target frame, and the normal directions of the corresponding rotated planes of the $j-$th frame.

Once the rotation matrix was solved, the translation was estimated with respect to the 3 DOF. In this progress, the pairwise vertical distances between the planes in the target frame and the corresponding rotated planes in the source frame were considered the residuals. The vertical distances were calculated using the projected distance of the vector starting at the geometric center points of the planes on the $k-$th plane in the $i-$th frame and the center points of the corresponding rotated planes on the $j-$th plane after the translation, which were calculated using *Equation 4.8*.

$$\begin{aligned} \epsilon_{kT} &= \boldsymbol{v}_{\boldsymbol{kT}}^{\boldsymbol{j\rightarrow i}} \cdot \boldsymbol{n}_{\boldsymbol{k}}^{\boldsymbol{i}} \\ &= (\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{j\rightarrow i}} - \boldsymbol{T}_{\boldsymbol{j\rightarrow i}}^{\boldsymbol{iR}}) \cdot \boldsymbol{n}_{\boldsymbol{k}}^{\boldsymbol{i}} \end{aligned} \tag{4.8}$$

where:

ϵ_{kT} is the residual of the translation between the $k-$th plane in the $i-$th frame and the corresponding rotated and translated plane in the $j-$th frame,

$\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{j\rightarrow i}}$ is the vector starting at the center point of the $k-$th plane in the $i-$th frame and pointing at the center point of the corresponding rotated plane in the $j-$th frame with no translation,

$\boldsymbol{v}_{\boldsymbol{kT}}^{\boldsymbol{j\rightarrow i}}$ is the vector starting at the center point of the $k-$th plane in the $i-$th frame and pointing at the center point of the corresponding rotated and translated plane in the $j-$th frame, and

$\boldsymbol{T}_{\boldsymbol{j\rightarrow i}}^{\boldsymbol{iR}}$ is the translation vector in the coordinate frame defined by the rotated $j-$th frame.

The weight values were considered in the estimation progress that the difference in point numbers between corresponding planes and the ratio of the number of points on the plane to the total number of points extracted as points on planes, as shown in *Equations 4.9-4.11*. Consequently, the weighted residuals were calculated using *Equations 4.12* and *4.13*.

$$w_{dk} = \frac{N_k^{less}}{N_k^{more}} \tag{4.9}$$

where:

w_{dk} is the weight determined by the difference in the number of points on the corresponding plane pairs,

N_k^{less} is the number of points on the corresponding $k-$th planes, of which frame the number of points is the smaller, and

N_k^{more} is the number of points on the corresponding $k-$th planes, of which frame the number of points is the larger.

$$w_{pk} = \frac{N_{ki}}{N_{pln,i}} \tag{4.10}$$

where:

w_{pk} is the weight determined by the ratio of the number of points on the plane to the total number of all the points extracted as the points on all the planes,

N_{ki} is the number of points on the $k-$th plane in the $i-$th frame, and

$N_{pln,i}$ is the total number of points on all the planes which are used for the plane matching progress.

$$w_k = w_{dk} \cdot w_{pk} \tag{4.11}$$

where:

w_k is the weight defined for estimating the transformation process.

$$\epsilon_{kwR} = w_k \cdot \epsilon_{kR} \tag{4.12}$$

where:

ϵ_{kwR} is weighted rotation residual of the $k-$th plane.

$$\epsilon_{kwT} = w_k \cdot \epsilon_{kT} \tag{4.13}$$

where:

ϵ_{kwT} is weighted translation residual of the $k-$th plane.

The axis-angle conversion and the non-linear optimization were done by the library provided by Google, namely Ceres Solver (Sameer, Keir, et al., 2010). Either sharing planes of different frames or the plane pairs, which were not reflected by the same segments of the plane but the correspondence was explicitly assigned, could be used for the estimation. Thus the estimation could be used for both intra-scanner calibration, in which no same segment of the sharing plane could be identified, and the alignment between frames of point clouds, in which the planes could be automatically matched.

Consequently, the estimated rotation matrix and the translation vector could be combined in the form of *Equation 4.2* and the relationship between the two frames could be represented in *Equation 4.14*.

$$\boldsymbol{X}_j^i = \boldsymbol{R}\boldsymbol{T}_{j\rightarrow i}^j \boldsymbol{X}_j^j \tag{4.14}$$

where:

$\boldsymbol{X}_j^j$ is the coordinates of the $j-$th frame points in its original coordinate frame defined by the scanner coordinate frame, and

$\boldsymbol{X}_j^i$ is the coordinates of the $j-$th frame points in the coordinate frame defined by the $i-$th frame.

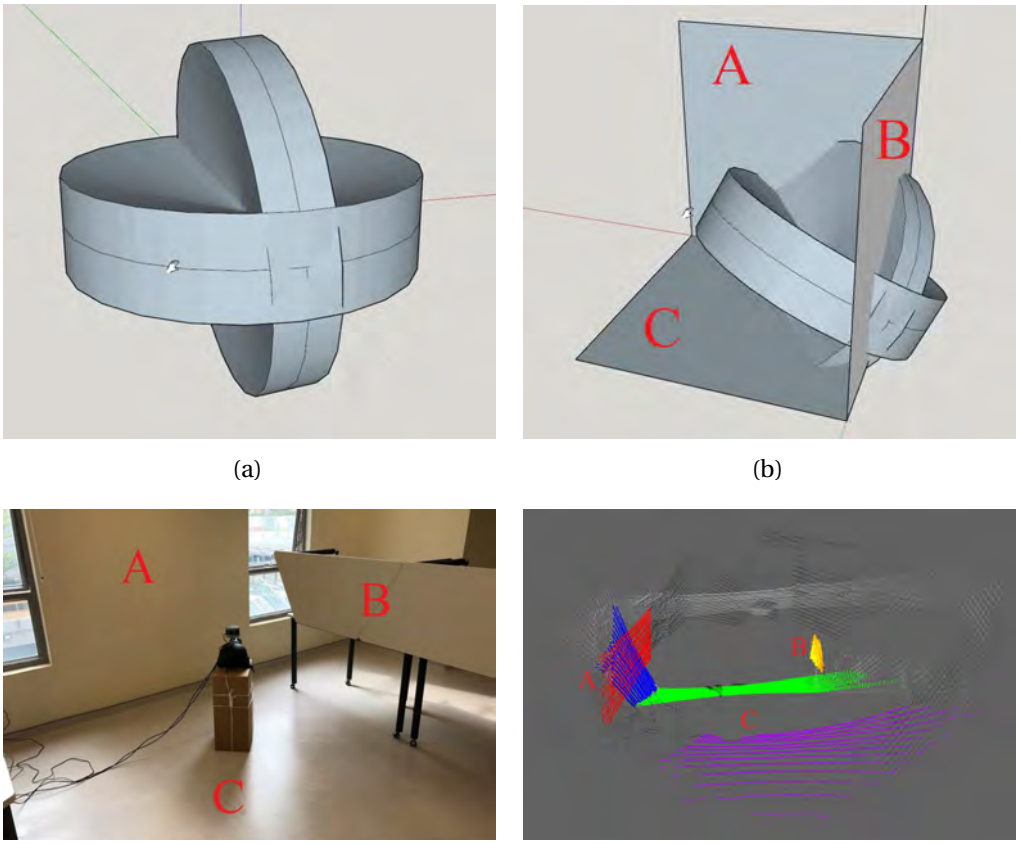
4.2.2 Intra-scanner Calibration

To determine the coordinate relationship between the two scanners, a temporal calibration site was designed based on the assumption that the 6 DOF coordinate relationship could be estimated with the sharing planes facing the three directions which were perpendicular to each other. As shown in *Figure 4.9(a)*, the FOV of the two scanners could be illustrated using connected cones. Therefore, if three planes were provided in the form of three perpendicular planes, as shown in *Figure 4.9(b)*, three pairs of planes could be identified.

In practice, there was no need for the three planes to be perfectly perpendicular to each other. By implementing the plane-to-plane alignments over manually assigning the correspondence to them, the rotation and translation between the two coordinate frames could be identified. Since the horizontally installed scanner was chosen as the primary scanner and its coordinate frame was selected as the main coordinate frame, the lever arm of the other scanner was estimated using the process mentioned above in *Section 4.2.1*.

4.2.3 Frame-to-frame Alignment Based on Plane Matching

The automatic frame-to-frame alignment was based on the plane matching results that could be used in the alignment process listed in *Section 4.2.1*. The alignment process is following a coarse-to-fine procedure that a coarse alignment was performed before the general alignment process to provide an initial relationship for



(a) (b)

(c) (d)

FIGURE 4.9: Calibration between the two multi-line scanners. The two connected cones are used to illustrate the 30 ° FOV of the SLAM scanners and the 20 ° FOV of the PCD scanner. (a) The overlapped coverage of the two scanners installed on S^2DAS. (b) The three planes, A, B, and C, show the three common planes that are used for calibration as sharing planes. (c) The photo of the calibration site with the assumed planes marked as A, B and C. (d) The calibrated point clouds with the horizontally scanned point cloud in black and the other point cloud in white. The corresponding A planes are in blue and red, while the corresponding B planes are in yellow and orange and the corresponding C planes in green and purple.

the plane matching process.

4.2.3.1 Coarse Alignment

Coarse alignment was performed over the NDT process, implemented via the PCL library. The corresponding frames of points captured by the two scanners during

the same period were marked as frame i. Given two frames of points, i and j, their alignment could be achieved, marked as $\boldsymbol{RT}_{\boldsymbol{NDT}}^{\boldsymbol{j\rightarrow i}}$. In most of the NDT processes, the results were not capable of achieving perfect alignments in which significant misalignments were spotted due to the low-resolution distributions of the inhomogeneous point clouds, as shown in *Figure 4.10*.

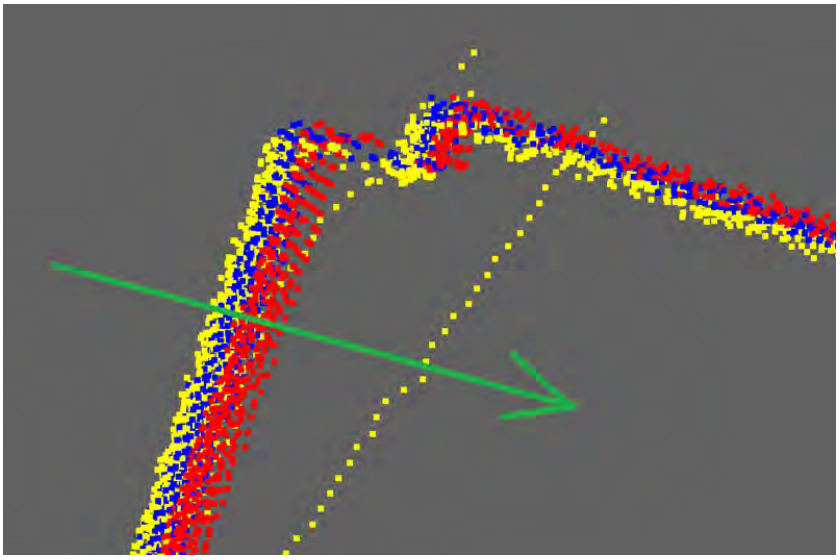


FIGURE 4.10: The example of the coarse-to-fine alignment results. The points in yellow are points from the target frame while the points in red are the source frame points aligned using the NDT result, and the points in blue are the points aligned using the proposed coarse-to-fine method. The NDT result shows significant misalignments along the direction of the green arrow.

In certain circumstances, the NDT alignment might be unsuccessful and significant errors might be produced, leading fruitless plane matching results. Therefore, an alternative process was implemented to introduce the position and attitude estimated using the state of the last frame and the motion calculated using *Equation 4.14-4.16* and *4.18*, which is *Equation 4.19*.

$$\boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{0}} = \boldsymbol{RT}_{\boldsymbol{j\rightarrow 0}}^{\boldsymbol{j}}\, \boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{j}} \quad \leftrightarrow \quad \boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{j}} = \boldsymbol{RT}_{\boldsymbol{j\rightarrow 0}}^{\boldsymbol{j}}{}^{-1}\, \boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{0}} \tag{4.15}$$

where:

the 0−th frame is defined as the target coordinate frame.

$$\boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{i}} = \boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}}\, \boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{j}} \tag{4.16}$$

where:

$\boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{j}}$ is the 4×4 position and orientation matrix of the $j-$th frame defined by the raw coordinate frame of the $j-$th frame, which is the identity matrix,

$\boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{i}}$ is the 4×4 position and orientation matrix of the $j-$th frame defined by the raw coordinate frame of the $i-$th frame, and

$\boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}}$ is the 4×4 rotation and translation matrix used to transform the coordinates from the $j-$th frame to the $i-$th frame.

$$\begin{aligned} \boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}}\, {\boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{0}}^{\boldsymbol{j}}}^{-1}\, \boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{0}} &= \\ \boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}}\, \boldsymbol{X}_{\boldsymbol{j}}^{\boldsymbol{j}} &= \boldsymbol{X}_{\boldsymbol{i}}^{\boldsymbol{i}} \\ &= {\boldsymbol{RT}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}^{\boldsymbol{i}}}^{-1}\, \boldsymbol{X}_{\boldsymbol{i}}^{\boldsymbol{0}} \end{aligned} \tag{4.17}$$

consequently,

$$\boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}} = {\boldsymbol{RT}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}^{\boldsymbol{i}}}^{-1}\, \boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{0}}^{\boldsymbol{j}} \tag{4.18}$$

$$\begin{aligned} \widehat{\boldsymbol{RT}_{\boldsymbol{i}}^{\boldsymbol{0}}} &= \widehat{\boldsymbol{RT}_{\boldsymbol{i}\rightarrow\boldsymbol{i\text{-}1}}^{\boldsymbol{i}}}\, \boldsymbol{RT}_{\boldsymbol{i\text{-}1}}^{\boldsymbol{0}} \\ &= \boldsymbol{RT}_{\boldsymbol{i\text{-}1\rightarrow i\text{-}2}}^{\boldsymbol{i\text{-}1}}\, \boldsymbol{RT}_{\boldsymbol{i\text{-}1}}^{\boldsymbol{0}} \\ &= {\boldsymbol{RT}_{\boldsymbol{i\text{-}2\rightarrow}\boldsymbol{0}}^{\boldsymbol{i\text{-}2}}}^{-1}\, \boldsymbol{RT}_{\boldsymbol{i\text{-}1\rightarrow}\boldsymbol{0}}^{\boldsymbol{i\text{-}1}}\, \boldsymbol{RT}_{\boldsymbol{i\text{-}1}}^{\boldsymbol{0}} \end{aligned} \tag{4.19}$$

where:

$\widehat{\boldsymbol{RT}_{\boldsymbol{i}}^{\boldsymbol{0}}}$ is estimation of the 4×4 position and orientation matrix of the $i-$th frame in the coordinate frame of the initial frame, and

$\widehat{\boldsymbol{RT}_{\boldsymbol{i\rightarrow i\text{-}1}}^{\boldsymbol{i}}}$ is the estimation of the 4×4 rotation and translation matrix used to transform the coordinates from the $i-$th frame to the $(i-1)-$th frame.

4.2.3.2 Fine Alignment

The fine alignment was conducted based on the successful plane matching and optimization process. Certain definitions were used in determining the quality of the plane pairs:

- The vertical distance between the planes, DP_{ctr}, which was estimated using the vertical distance between the center point of the corresponding plane,

- The average vertical distance, ADP_{each}, which was the average value of the distances between each of the point pairs consisting of points selected from the two planes,

- The average resolution of points on the planes, $ARes_v$, which was the average value of the Euclidean distance between neighboring points on the same vertical scanline,

- When the distance between points from the two pending plane pairs was smaller than $ARes_v$ or the given threshold TH_{MinRes}, which was larger, the point would be considered as an overlap point,

- The overlap rate, $R_{OverlapPln}$, which was the ratio of the number of the overlap points to the number of points on the plane,

- The overlap rate of the pair, $R_{OverlapPair}$, which was the larger ratio of the pairwise values of $R_{OverlapPln}$,

- The weight for checking overlap rate, $\omega_{overlap}$, which was corresponding to the ratio of the number of points on one of the planes (N_{PlnA}) to the number of points on the other plane (N_{PlnB}), as calculated using *Equation 4.20*, and

$$\omega_{overlap} = \frac{1}{\left| \frac{N_{PlnA}}{N_{PlnB}} - 1 \right|} \tag{4.20}$$

- The weighted overlap rate, $WR_{overlap}$, which was the product of $R_{OverlapPair}$ and $\omega_{overlap}$.

Once the initial state was determined using the NDT or the state and the motion of the previous frames, the correspondence between the planes of the given two frames could be identified using the following criteria, which were empirical values determined in the experiments:

- DP_{ctr} should be smaller than the given threshold value $TH_{Coplanarity}$,

- ADP_{each} should be smaller than the given threshold value $TH_{PerDist}$,

- $R_{OverlapPair}$ should be larger than the given threshold $TH_{overlap}$,

- $WR_{overlap}$ should be larger than the given threshold $TH_{overlap}$.

When multiple planes met the requirements of corresponding to the same plane, the one with the largest weighted overlap rate was selected as the best corresponding plane. Consequently, the correspondences were generated for estimating the relationship between frames using the method listed in *Section 4.2.1*.

In the alignment process, if any of the unweighted residuals were larger than the given threshold values, TH_{rot} and TH_{transl} determined by the empirical values, the corresponding plane pairs would be removed, and the remaining pairs would be used for alignment estimation in the next iteration until a converged solution could be provided.

Moreover, the main directions of the plane pairs were checked at the end of every iteration. The main directions were defined as the three normal directions of the pairwise planes, with the largest total angular differences between each pair. The angular differences between every pair of the main directions should be larger than the given empirical threshold value of $TH_{MainDirDiff}$. If the valid main directions were not selected, the matching would be considered as a failed matching.

Furthermore, the main directions were used to evaluate the plane matching process by checking the number of planes in every group concerning the differences between the normal directions of the planes and each of the main directions, namely the Pairing Score ($\boldsymbol{v_{PS}}$). The matching was considered as qualified only when the number of corresponding plane pairs in each group, which were stored in a 3D vector named as the pairing score vector, was larger than the given empirical threshold TH_{Group}.

When the matching relationships were successfully established, it was considered a valid observation. The observations were reserved in a list for further processing and invoking.

4.2.3.3 Plane Matching Failure

In certain circumstances, there might not be enough planes matching or enough directions facing, which would cause unsuccessful position and orientation solving. Consequently, previous frames were used for establishing possible links with the pending frames. For example, if the observation between frame i and $i-1$ was not established, the processor would try to establish the observation between frame i and $i-2$. Moreover, if no linkage was built, the NDT solutions would be used as the only solutions. In other words, the coarse aligned NDT solutions provided alternatives when no optimum results could be estimated, which increased the reliability of the overall alignment process.

4.2.4 Redundant Observations

The ordinary frame-to-frame alignments were first conducted between adjacent frames, building the initial trajectory. To provide redundant observations as the architecture introduced in Dai et al. (2017), key frames were identified, and extra alignments were established.

As introduced in the previous section, the Pairing Score ($\boldsymbol{v_{PS}}$) was used to check the quality of plane matching. The plane numbers in every group were then adopted

for checking the need for new key frames, which were the most representative frames in the local sections consisting of a few frames. If any of the elements in the current Pairing Score ($\boldsymbol{v}^{i}_{\boldsymbol{PS}}$) were smaller than the corresponding threshold, the new key frame would be required. The seeking and the identification of the key frames and the redundant alignments are listed in *Algorithm 4.1*.

Algorithm 4.1 Observation Establishment

Notation:
i: frame ID
i_{max}: maximum frame ID
i_{KF}: key frame ID
Ar_{KF}: array of key frame ID
j: key frame ID in the key frame array
j_{max}: number of key frames
A_a: alignment between adjacent frames
A_{KF}: alignment between normal frame and the previous key frame
Ar_{Ob}: array of observations
Ar_{pln}: extracted plane features
$\boldsymbol{v}_{\boldsymbol{PS_{KF}}}$: the pairing score between the current frame and the previous key frame
$\boldsymbol{v}_{\boldsymbol{PS}}$: the user-defined key frame pairing score vector threshold

Input: $Ar_{\boldsymbol{pln}}$
Output: Ar_{Ob}

procedure OBSERVATION ESTABLISHMENT
  $i_{KF} \leftarrow$ i
  $i \leftarrow i+1$
  insert i_{KF} into Ar_{KF}
  $j_{max} \leftarrow$ size of Ar_{KF}
  while $i \neq i_{max}$ **do** Ordinary Observation Establishing
    $A_a \leftarrow$ alignment from i to $i-1$
    insert A_a into Ar_{Ob}
    $A_{KF} \leftarrow$ alignment from i to i_{KF}
    insert A_{KF} into Ar_{Ob}
    $\boldsymbol{v}_{\boldsymbol{PS_{KF}}} \leftarrow A_{KF}$
    if $\boldsymbol{v}_{\boldsymbol{PS_{KF}}} < \boldsymbol{v}_{\boldsymbol{PS}}$ **then**
      $i_{KF} \leftarrow i$
      $j \leftarrow j_{max}$
      while $j \neq 0$ **do** Redundant Key Frame Observation Establishing
        $A_{KF_j} \leftarrow$ alignment from i_{KF} to $Ar_{KF}[j]$
        insert A_{KF_j} into Ar_{Ob}
        $j \leftarrow j-1$
      insert i_{KF} into Ar_{KF}
      $j_{max} \leftarrow j_{max}+1$
    $i \leftarrow i+1$
  return Ar_{Ob}

Consequently, all the observations built between the pending frames and the

previous key frames and the alignments between the key frames were stored as extra observations, which were used for the adjustment in addition to the ordinary alignments between each of the pairwise adjacent frames.

4.2.5 Alignment Transferring and Shortest Path

A significant drawback of applying NDT for the coarse alignment was that successful estimation requires a roughly aligned initial state. Meanwhile, since the mobile mapping process was a DR process, both the position and attitude were drifting along the accumulation of alignment errors.

With all the adjacent frames aligned to each of their preceding neighbors, the alignment results would be passed on between frames if no error or misalignment were considered. The equations for direct transferring can be derived based on *Equation 4.16* and *4.19*, as shown in *Equation 4.21*.

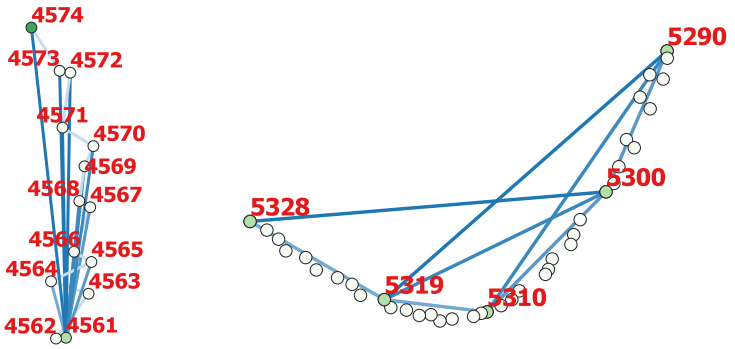
$$\begin{aligned} \boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{i}} &= \widehat{\boldsymbol{RT}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}^{\boldsymbol{j}}}\, \boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{j}} \\ &= \boldsymbol{RT}_{\boldsymbol{j\text{-}1\rightarrow i}}^{\boldsymbol{j\text{-}1}}\, \boldsymbol{RT}_{\boldsymbol{j\rightarrow j\text{-}1}}^{\boldsymbol{j}}\, \boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{j}} \\ &= \dots \\ &= \underbrace{\boldsymbol{RT}_{\boldsymbol{i+1\rightarrow i}}^{\boldsymbol{i+1}}\, \boldsymbol{RT}_{\boldsymbol{i+2\rightarrow i+1}}^{\boldsymbol{i+2}} \dots \boldsymbol{RT}_{\boldsymbol{j\rightarrow j\text{-}1}}^{\boldsymbol{j}}}_{j-i+1}\, \boldsymbol{RT}_{\boldsymbol{j}}^{\boldsymbol{j}} \end{aligned} \tag{4.21}$$

where:

$\widehat{\boldsymbol{RT}^{j}_{j\rightarrow i}}$ is the estimated value of the rotation and transformation from the $j-$th frame to the $i-$th frame.

As aforementioned in the previous sections, extra observations between the current frames and the previous key frames were built to check the pairing scores and identify new key frames. Therefore, if there were at least one frames between the current frame and the preceding key frame, more than one route would exist, which transferred the rotation and translation relationship from the current frame to the key frame if all transformations were considered to be unidirectional, as the example given in *Figure 4.11(a)*. When estimating the relationship between key frames, the similar multi-path problem existed, as shown in *Figure 4.11(b)*.

To reduce the drifts, errors, and misalignments in initializing the rotation and translation relationships using transferred estimations, the problem was defined as a shortest-path problem based on the assumption that all transformations were unidirectional pointing from the latter frames with the larger frame ID to the previous frames with the smaller frame ID. The problem was solved when Dijkstra's Shortest Paths in boost C++ library was implemented (Siek, Lee, & Lumsdaine, 2002). When the length of the successful alignment observation was defined as 1, the shortest path between any two frames could be identified with the lowest accumulative misalignment errors introduced. Therefore, given the corresponding shortest-path transformation matrices provided, as in *Equation 4.21*, the initial value of the transformation matrix between any two frames could be estimated via the shortest paths, namely $\boldsymbol{RT}^{\boldsymbol{0}}_{\boldsymbol{j\rightarrow i}}$ in *Equation 4.22*.



(a) (b)

FIGURE 4.11: The routes of transferring the rotation and translation with redundant observations. With the frame ID beside, the dots in green represent the positions of the key frames, and those in white indicate other regular frames. (a) The line segments between dots show the observations established. Multiple paths can be identified for estimating the relationship between the pending frame and the preceding key frame. For example, given frame 4565 as the pending frame, the available paths are 4565 → 4564 → 4563 → 4562 → 4561, 4565 → 4564 → 4563 → 4561, and 4565 → 4564 → 4561. (b) The existence of multi-path problem for estimating the initial relationship between key frames with only observations between key frames are shown in blue line segments. The available paths from 5290 to 5328 include 5290 → 5300 → 5328, 5290 → 5300 → 5319 → 5328, 5290 → 5300 → 5310 → 5319 → 5328, 5290 → 5310 → 5319 → 5328, and 5290 → 5319 → 5328.

$$\boldsymbol{X}_i^0 = \boldsymbol{R}\boldsymbol{T}_{j\to i}^0 \boldsymbol{X}_j^0 \tag{4.22}$$

Moreover, the raw observation between the two key frames, which was based on the point clouds in their raw coordinate frame corresponding to the scanner center, could be derived for building the edges in the adjustment process, as shown in *Equation 4.24.*

$$
\begin{aligned}
\boldsymbol{RT}^{\boldsymbol{i}}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}\,\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}\,\boldsymbol{X}^{\boldsymbol{j}}_{\boldsymbol{j}} &= \\
\boldsymbol{RT}^{\boldsymbol{i}}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}\,\boldsymbol{X}^{\boldsymbol{i}}_{\boldsymbol{i}} &= \boldsymbol{X}^{\boldsymbol{0}}_{\boldsymbol{i}} \\
&= \boldsymbol{RT}^{\boldsymbol{0}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}\,\boldsymbol{X}^{\boldsymbol{0}}_{\boldsymbol{j}} \\
&= \boldsymbol{RT}^{\boldsymbol{0}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}\,\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{0}}\,\boldsymbol{X}^{\boldsymbol{j}}_{\boldsymbol{j}}
\end{aligned}
\tag{4.23}
$$

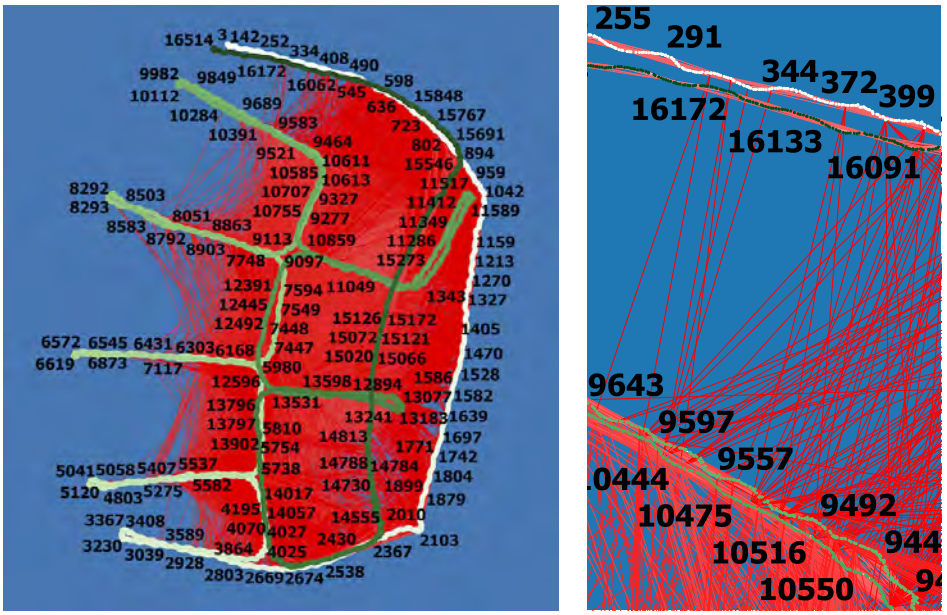
which is equivalent to

$$
\boldsymbol{RT}^{\boldsymbol{i}}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}\,\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}} = \boldsymbol{RT}^{\boldsymbol{0}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}\,\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{0}} \quad \leftrightarrow \quad \boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}} = {\boldsymbol{RT}^{\boldsymbol{i}}_{\boldsymbol{i}\rightarrow\boldsymbol{0}}}^{-1}\,\boldsymbol{RT}^{\boldsymbol{0}}_{\boldsymbol{j}\rightarrow\boldsymbol{i}}\,\boldsymbol{RT}^{\boldsymbol{j}}_{\boldsymbol{j}\rightarrow\boldsymbol{0}} \tag{4.24}
$$

Indeed, although the tedious searching of successful alignments between key frames significantly slowed down the whole aligning process, the redundant observations provided profuse edges connecting the key frames and formed small loops used for reducing the position and orientation drifts. *Figure 4.12* shows an example that of redundant observations built in mapping the interior of a large lecture hall that could hold up to 400 people.

4.2.6 Motion Rectification

When point clouds were captured on a mobile platform, they were distorted in the movement during the time of capturing this specific frame of data. For profiler installed vertically on the mobile systems that were not based on SLAM but on GNSS/IMU integrated system, the distortions could be rectified automatically as



(a) (b)

FIGURE 4.12: An example of the redundant observations during mapping a lecture hall. The dots are representing positions of frames with gradient green while the red line segments are showing the successful alignment between frames. (a) The overall trajectory and observation edges. (b) A selected part of (a) showing the redundant alignment between 300th-400th frames and 9400th-9650th frames.

the positions and orientations of the scanner at the specific acquisition time were interpolated from the trajectory and attitude changing data stream. However, for SLAM-based solutions with either single-line scanners or multi-line scanners, the point clouds were divided into frames concerning the horizontal azimuth angle loops. Therefore, the time for capturing the single-frame point cloud would result in inevitable distortion when the platform was moving.

Given the position and orientation change in the form of 4×4 matrix, the distortion in the time change could be derived with the given positions and attitudes. Given *Equation 4.1*, *4.2*, *4.3* and *4.16* with the timestamp of the first point in the current frame and the next frame, t_{c0} and t_{n0}, and the timestamp of the pending point

t_{pt}, the rectified rotation and translation matrix of the point is shown in *Equation 4.28*.

$$p_{pt} = \frac{t_{pt} - t_{c0}}{t_{n0} - t_{c0}} \tag{4.25}$$

where:

p_{pt} is the portion of the time elapsed over the total time interval of the current frame.

$$\boldsymbol{R_{pt}} \leftrightarrow \theta_{\boldsymbol{pt}} = p_{pt} \begin{bmatrix} \theta_{i1} & \theta_{i2} & \theta_{i3} \end{bmatrix}^T \tag{4.26}$$

where:

$\boldsymbol{R_{pt}}$ is the rotation matrix that is used for estimating the coordinate changes with respect to the platform movement, and

$\theta_{\boldsymbol{pt}}$ is the vector of angle-axis that is used for representing the rotation of the point cloud concerning the time when the pending time is captured.

$$\boldsymbol{T_{pt}} = p_{pt}\ \boldsymbol{T_i} \tag{4.27}$$

where:

$\boldsymbol{T_{pt}}$ is the translation vector used for estimating the coordinate changes along the platform movement, and

$\boldsymbol{T_i}$ is the translation vector of the whole point cloud within the total time interval.

$$\begin{aligned}\begin{bmatrix} x_{pt^T} & y_{pt^T} & z_{pt^T} & 1 \end{bmatrix}^T &= \boldsymbol{X_{pt^T}} \\ &= \boldsymbol{RT_{pt}}\, \boldsymbol{X_{pt}} \\ &= \boldsymbol{RT_{pt}} \begin{bmatrix} x_{pt} & y_{pt} & z_{pt} & 1 \end{bmatrix}^T\end{aligned} \tag{4.28}$$

where:

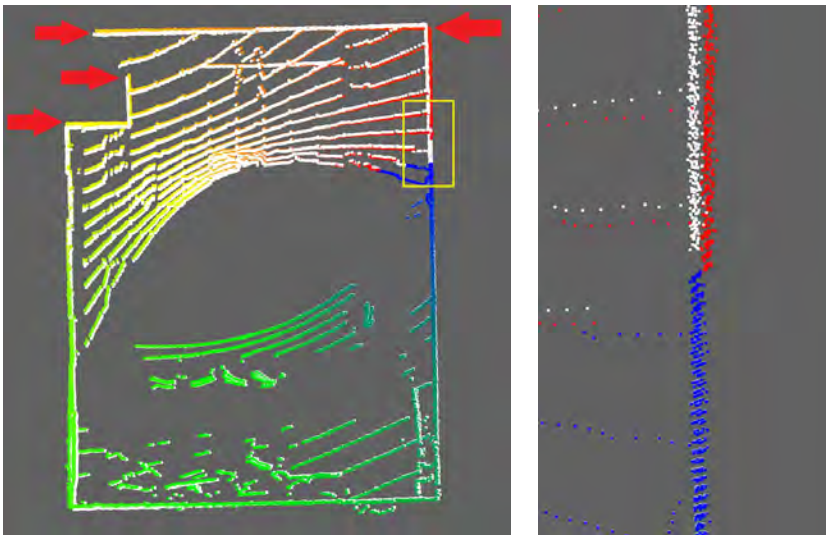
$\begin{bmatrix} x_{pt^T} & y_{pt^T} & z_{pt^T} & 1 \end{bmatrix}^T$ and $\boldsymbol{X_{pt^T}}$ are the rectified coordinates of the pending point, and

$\boldsymbol{RT_{pt}}$ is the transformation matrix derived using *Equation 4.2*, *4.26*, and *4.27*.

The motion during the capturing of the last frame could not be estimated and updated because the last frame did not have a next frame. Therefore, for a data stream consisting of N frames, only the points of the first $N-1$ frames would be rectified. However, if the update rate of the laser scanner were high enough or the moving of the platform was not quite obvious, the motion rectification would not be quite considerable. An example of motion rectification is shown in *Figure 4.13*.

4.2.7 Overall Procedure

Given these processes above for building ordinary observations between each adjacent-frame pair, the normal-to-key observations between every frame and the nearest preceding key frames, and the key-to-key observations between any two of the key frames, all successful alignments achieved were saved as edges connecting the two frames.

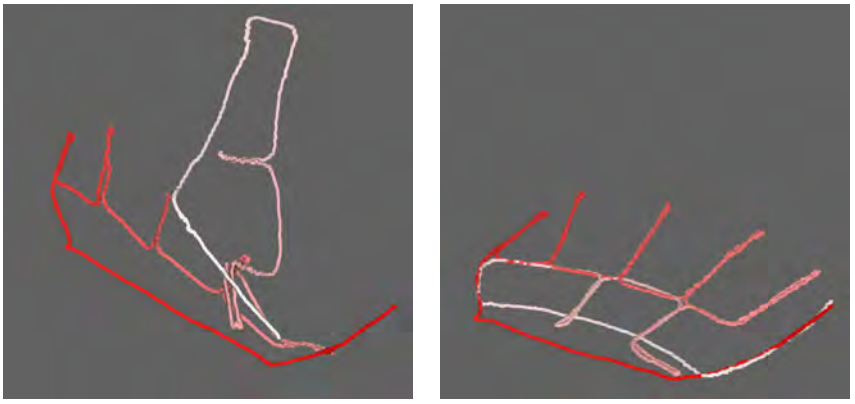


(a) (b)

FIGURE 4.13: Comparison between the raw point cloud and the rectified point cloud. The raw point cloud is in Blue → Green → Yellow → Red depending on the time stamp of the points, while the rectified point cloud is colored in white for comparison. (a) The overall comparison of the whole point cloud shows appreciable differences between rectified points and raw points. The differences increase with the ascent of the time stamp of points and are identifiable at the positions marked with arrows. (b) A part of the point cloud located at the position in (a) marked in a red box. The points are the pulses reflected from the sidewall. The points in red, which are not rectified and captured at the ending time of the frame, show significant misalignments with blue points, which are captured at the initial time of the frame. As opposed to the distorted points, the rectified points in white, which correspond to the red points, show great compliance with the positions of the blue points and form a straight line rather than fractures.

All observations were adopted in the adjustment process, which was based on the General Graph Optimization (g2o) library by Kummerle, Grisetti, Strasdat, Konolige, and Burgard (2011). The library was initially designed for adjusting vSLAM results while the frame-to-frame structure for alignments in the proposed workflow is similar to the workflow required by the library. In the implementation, the raw trajectory derived from the frame-to-frame alignments between adjacent frames was

used as the initial position and attitude values of the vertices in the adjustment process, with all the observations inputted as edges with the same weight and quality. An example of the trajectory before and after the adjustment is given in *Figure 4.14* to show the differences.



(a) (b)

FIGURE 4.14: The unadjusted and adjusted trajectory of mapping the lecture hall. The points in red are representing the positions of captured frames in the coordinate frame determined by the scanner-centered coordinate frame of the first frame. The preceding frames are in red while colors are fading with the increasing frame ID. (a) The raw trajectory with no G2O adjustment shows a twisted result. (b) The adjusted trajectory shows great coincidence with the real trajectory.

After adjustments, point clouds would be registered to the same coordinate frame, which is the coordinate frame of the first frame centering at the origin of the scanner coordinate frame. The motion rectification could be performed, and it might be necessary for performing the whole alignment and adjustment process once more. However, there would not be too many differences, considering the accuracy of the MLS. Then all the point clouds of different frames would be merged to form a point cloud of the mapping area.

The overall aligning and mobile mapping process adopted is shown in *Algorithm C.2* in *Appendix C*. The listed procedure is the complete offline processing workflow. For the online process, only alignments between adjacent frames are conducted with neither redundant observations nor g2o adjustments.

4.3 Sample Tests and Result Analysis

The proposed workflow of IMU-free 3D LiDAR mobile mapping was applied on S^2DAS. Several typical indoor scenarios and an outdoor terrace were used as test sites to verify the feasibility and reliability of the proposed methods. In the point cloud merging process, only the points captured by the vertically installed scanner were used to limit the influences of the errors in the attitudes of the horizontal scanner.

A TLS was also implemented in the test sites with target balls to capture point clouds used as reference point clouds, checking the accuracy of S^2DAS. The scanner was a FARO Focus M70 designed for indoor and short-range applications. The declared ranging accuracy of the scanner was ±3 mm while the angular and 3D accuracy were not provided (FARO Technologies Inc, 2019). The point clouds captured at every scan station were merged using ICP provided in the FARO Scene processing software while all target spheres and hemispheres were extracted using the RANSAC shape detection function provided by CloudCompare (Schnabel et al., 2007).

The point clouds captured by S^2DAS were automatically processed in the toolkit implemented using C++ with the dependent libraries of PCL and ELSExt, which is a C++ implementation of the ELS-based plane extraction method, as introduced in *Chapter 3*. The parameters for processing all the data were kept the same with no manual intervention during the whole process.

4.3.1 Large Lecture Hall

A large lecture hall holding up to 400 peoples, which is Room Z209, Block Z, The Hong Kong Polytechnic University (HKPU), was selected as the test site representing the typical large space scenarios. The inside of the hall was modified with small planar boards which formed large curved surfaces in order to provide excellent stereo effects, which was believed to be an extra challenge for S^2DAS as every small planar board needs to be distinguished for precise alignments. The merged TLS point clouds and the mobile mapping results are shown in *Figure 4.15*, with all the positions of the scan stations and the target spheres and hemispheres, and the trajectory.

The two methods showed their limits in covering the floor and stairs. In *Figure 4.15(c)*, the laser beams emitted by the TLS could not reach most of the floor areas between the neighboring row of seats due to the obstructions of the cinema chairs, while similar problems occurred only in the two high areas on the top of *Figure 4.15(d)*. In other words, the TLS was not capable of capturing every detail in the congested areas with obstructions if the scan stations did not distribute densely, while the mobile mapping solution required direct accessibility for detail capturing.

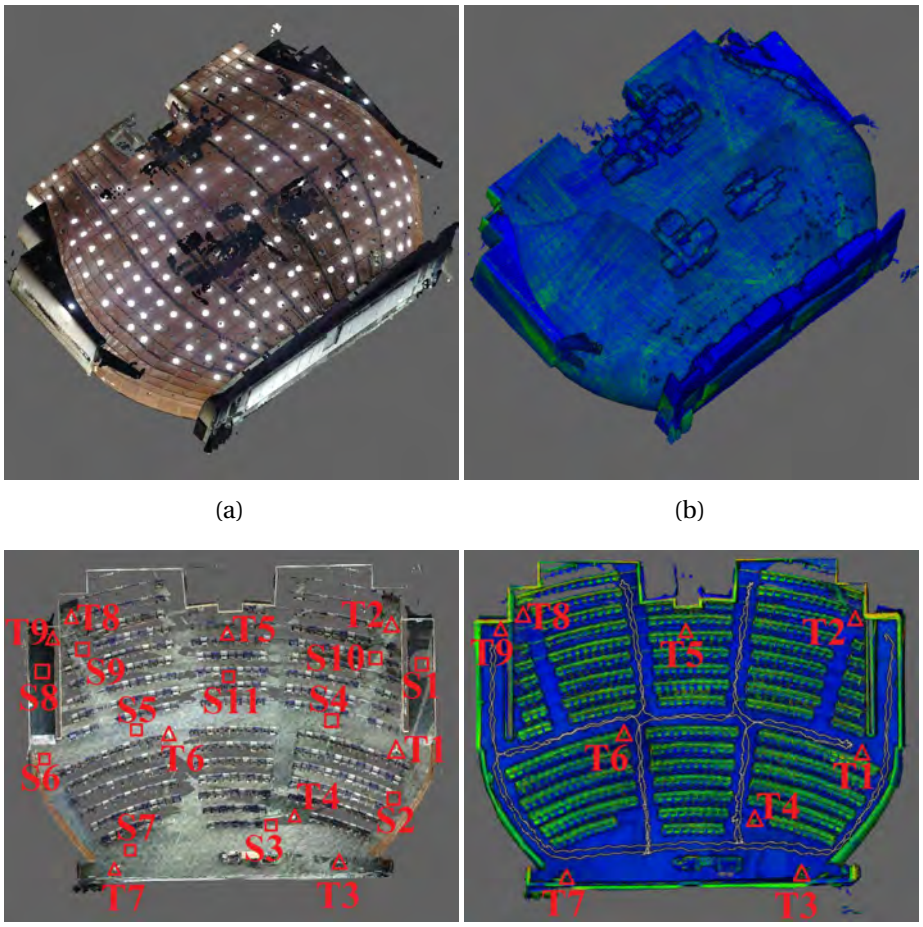


FIGURE 4.15: The point clouds of the lecture hall captured by TLS and S^2DAS. The positions of the target spheres and hemispheres are marked with triangles labeled "T1" to "T9". (a) The point cloud captured by TLS and colorized using the images captured by the embedded camera. (b) The point cloud captured by S^2DAS and colorized with respect to the point intensity provided by the multi-line scanner installed. Points in blue are with low intensities while those in green show high intensities. (c) The red squares show the positions of the eleven scan stations labeled "S1" to "S11". (d) The moving trajectory of S^2DAS shown in white with the positions of targets.

The nine target spheres and hemispheres were installed on the tripods distributed in the lecture hall to compare the distances between each of the two targets and the angles of the triangles formed by each of the three targets. The coordinates of the extracted targets in the three datasets are shown in *Table 4.1*. The hemisphere

target labeled "T8" could not be extracted from the mobile mapping point cloud because the distance between the target and the nearest data capturing position was too far. The noises and the misalignment errors showed significant disturbances on the target surfaces, and no model with substantial compliance could be fitted.

TABLE 4.1: Target coordinates in both point clouds (the Lecture Hall dataset, not aligned)

Target ID	Type	Coordinates by TLS (m)	Coordinates by S^2DAS (m)	Radius (cm)
T1	Hemisphere	(-4.298, 4.415, 3.893)	(6.073, 3.794, 2.268)	7.5
T2	Hemisphere	(3.490, 0.560, 4.953)	(0.191, -2.712, 0.995)	7.5
T3	Hemisphere	(-9.112, 11.619, 4.111)	(13.802, 6.341, 5.223)	7.5
T4	Sphere	(-4.878, 12.753, 3.191)	(14.051, 1.997, 4.070)	10.0
T5	Hemisphere	(8.255, 10.622, 4.220)	(8.499, -10.170, 2.338)	7.5
T6	Sphere	(4.140, 17.355, 3.882)	(15.809, -7.998, 4.633)	10.0
T7	Hemisphere	(-2.092, 25.100, 3.286)	(24.638, -4.210, 7.414)	7.5
T8	Hemisphere	(14.349, 19.426, 5.101)	unable to extract	7.5
T9	Hemisphere	(13.777, 21.073, 3.801)	(16.756, -18.224, 4.370)	7.5

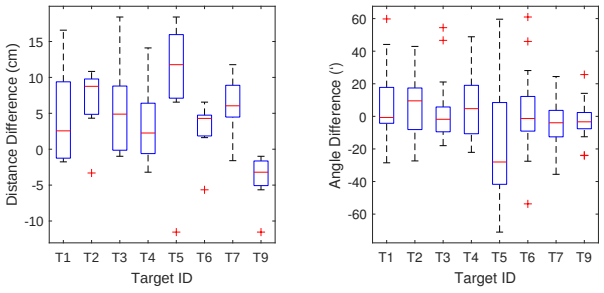
Furthermore, as the radius of the hemisphere targets was only 7.5 cm, which is not significantly large considering the distance measurement accuracy of the laser scanner (3 cm) and the possible positioning and orientation errors, the extraction of the hemisphere targets was quite difficult when the distance between the targets and the nearest data capturing positions was too long. Therefore, larger and whole spherical targets are suggested in future tests.

The distances between the centers of the targets and the angles of the triangles formed by every three targets were used to check the accuracy of the mobile mapping results. The detailed results are listed in *Table C.1, C.13* and *C.14* in *Appendix C* while the distributions are shown in *Figure 4.16(a)* and *4.16(b)*. The distribution

TABLE 4.2: Target coordinates in both point clouds (the Lecture Hall dataset, aligned)

Target ID	Coordinates by TLS (m)	Coordinates by S^2DAS (m)
T1	(-4.298,4.415,3.893)	(-4.355 4.424 3.931)
T2	(3.490,0.560,4.953)	(3.528 0.526 5.029)
T3	(-9.112,11.619,4.111)	(-9.162 11.623 4.079)
T4	(-4.878,12.753,3.191)	(-4.896 12.747 3.185)
T5	(8.255,10.622,4.220)	(8.390 10.623 4.104)
T6	(4.140,17.355,3.882)	(4.144 17.354 3.787)
T7	(-2.092,25.100,3.286)	(-2.099 25.155 3.339)
T8	(14.349,19.426,5.101)	unable to extract
T9	(13.777,21.073,3.801)	(13.731 21.044 3.883)

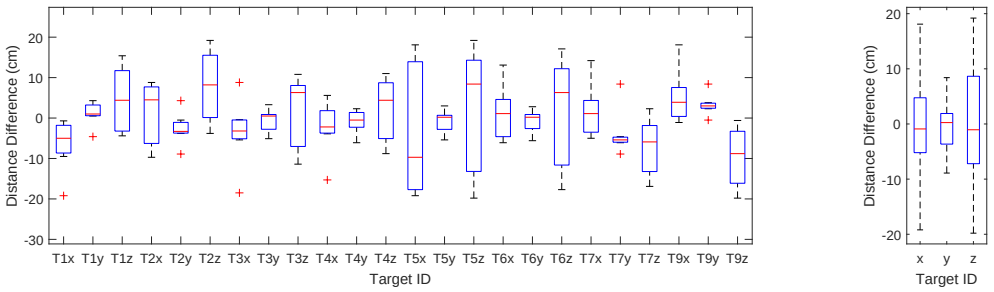
of coordinate variations between targets along the axes of the TLS coordinate frame were calculated and are illustrated in *Figure 4.17*.



(a) (b)

FIGURE 4.16: The distribution of distance and angle differences between the targets in the TLS point cloud and the S^2DAS point cloud (the Lecture Hall dataset). (a) Distribution of distance differences. (b) Distribution of angle differences.

The distance, coordinate, and angular accuracy of T2 and T5 are significantly worse than the accuracy of the other targets. Although both the extreme differences along the x-axis and the z-axis are larger than the value along the y-axis, the overall difference in the horizontal plane is smaller than the value in the vertical direction.



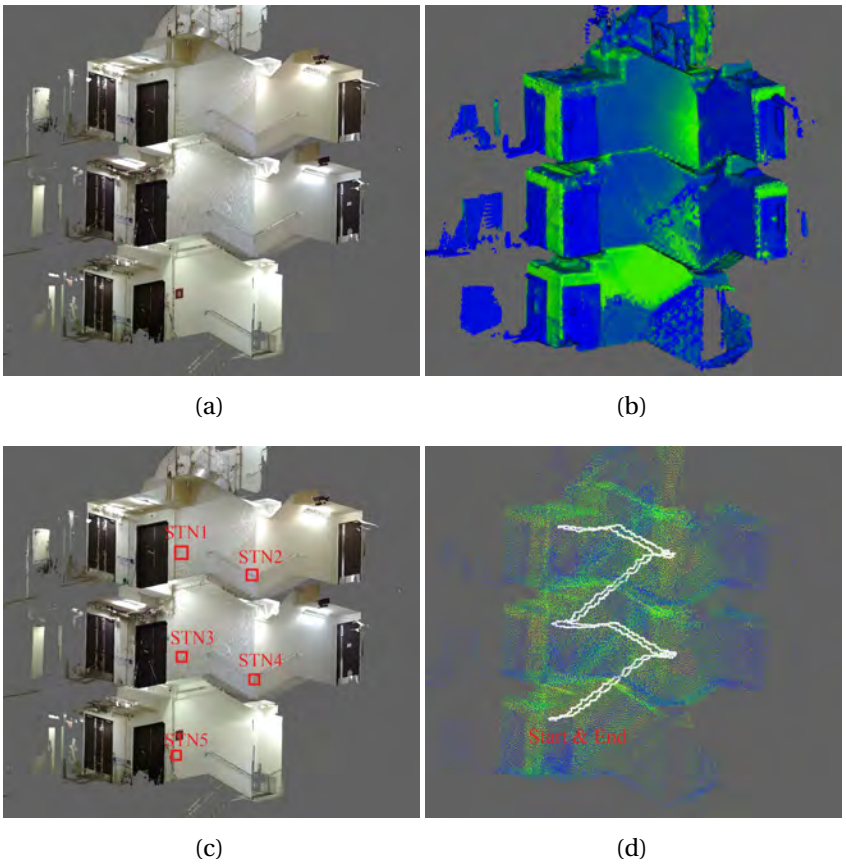
(a) (b)

FIGURE 4.17: The distribution of distance differences along the Cartesian axes between the targets in the TLS point cloud and the S^2DAS point cloud (the Lecture Hall dataset). (a) The distribution of the differences with respect to each of the targets along the Cartesian axes. (b) The distribution of all the differences with respect to the Cartesian axes.

4.3.2 Sealed Stairwell

The indoor stairwell of North Wing, Block Z, HKPU was selected as the test site for the multistory scenarios. The stairs from 6/F to 8/F were used to verify the performance of the proposed system and the algorithms. Nine target spheres and hemispheres were used as control points to compare results. The corresponding terrestrial laser scanning was performed at five scan stations. The overall merged point cloud was colorized using the images captured by the TLS and is shown in *Figure 4.18*.

The points indicating thin surfaces, i.e., planes, in the point cloud captured by S^2DAS are not as thin as the planes in the reference point cloud captured by TLS. The main reason was that the low-accuracy MLS generated inevitable random measurement noises which resulted in the distribution of points in a range of 3 cm around the real distance, while the corresponding measurement accuracy of FARO



(a) (b)

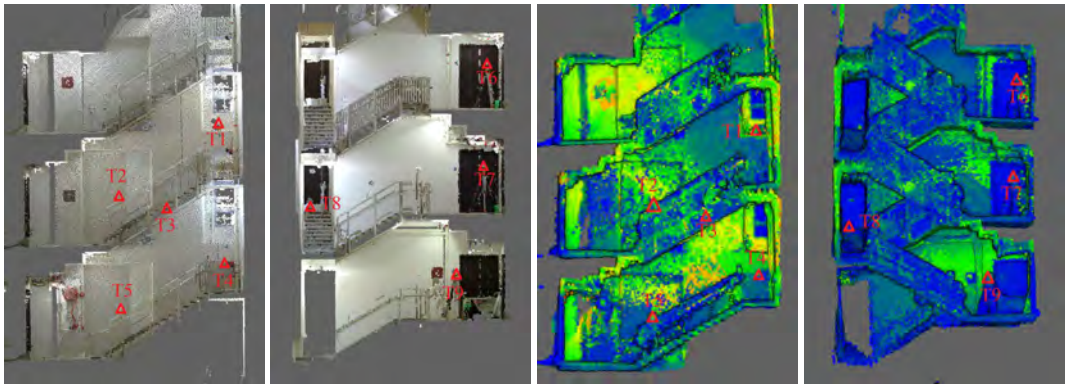
(c) (d)

FIGURE 4.18: The point clouds of the stairwell captured by TLS and S^2DAS. (a) The point cloud captured by TLS is colorized using the images captured by the embedded camera. (b) The point cloud captured by S^2DAS and colorized with respect to the point intensity provided by the multi-line scanner installed. Points in blue are with low intensities while those in green show high intensities. (c) The red squares show the positions of the five scan stations. (d) The moving trajectory of S^2DAS is shown in white.

Focus M70 was 3 mm. The uncertainty would additionally create misalignment, which increased the thickness of the planes as well.

As mentioned above, nine target spheres and hemispheres were distributed in the stairwell for comparison. The locations of the targets are shown in *Figure 4.19*. Their corresponding coordinates in the respective coordinate frame are shown in *Table 4.3*. Consequently, the detailed distance and the angle comparison between

target centers of the two results are shown in *Table C.3, C.15,* and *C.16* in *Appendix C,* while the distribution of the comparison results is shown in *Figure 4.20.* The distribution of difference in the variations along the TLS axes are presented in *Figure 4.21.*

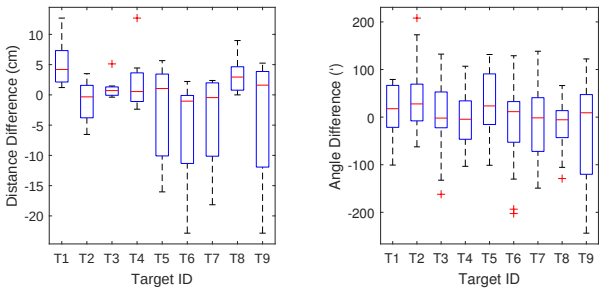


(a) (b) (c) (d)

FIGURE 4.19: The locations and labels of the nine target spheres and hemispheres in the two results. (a) The upstairs part of the TLS point cloud with the labeled triangles showing the positions of the targets. (b) The downstairs part of the TLS point cloud with the labeled triangles showing the positions of the targets. (c) The upstairs part of the S^2DAS point cloud with the labeled triangles showing the positions of the targets. (d) The downstairs part of the S^2DAS point cloud with the labeled triangles showing the positions of the targets.

TABLE 4.3: Target coordinates in both point clouds (the Stairwell dataset, not aligned)

Target ID	Type	Coordinates by TLS (m)	Coordinates by S^2DAS (m)	Radius (cm)
T1	Hemisphere	(1.302, 4.957, 7.005)	(2.880, 2.847, 7.018)	7.5
T2	Sphere	(1.103, 1.046, 4.500)	(0.777, 0.255, 3.765)	10.0
T3	Hemisphere	(1.827, 2.672, 3.961)	(2.499, 0.974, 3.626)	7.5
T4	Sphere	(1.541, 4.939, 2.067)	(4.116, 2.935, 2.101)	10.0
T5	Hemisphere	(1.071, 1.082, 0.488)	(1.526, 0.435, -0.106)	7.5
T6	Hemisphere	(1.061, -1,791, 8.509)	(-1.941, -2.145, 7.050)	7.5
T7	Hemisphere	(1.225, -1.737, 4.617)	(-0.958, -1.914, 3.356)	7.5
T8	Hemisphere	(6.015, 3.334, 3.032)	(6.192, -1.291, 3.318)	7.5
T9	Hemisphere	(1.370, -0.670, 0.557)	(0.575, -1.084, -0.265)	7.5



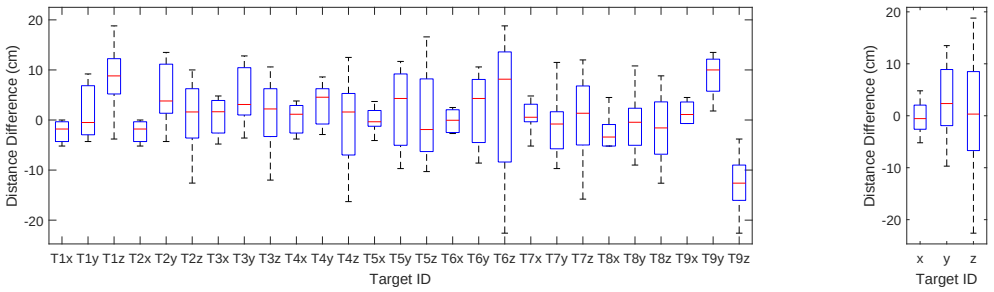
(a) (b)

FIGURE 4.20: The distribution of distance and angle differences between the targets in the TLS point cloud and the S^2DAS point cloud (the Stairwell dataset). (a) The distribution of distance differences. (b) The distribution of angle differences.

The nine targets were used to estimate the corresponding transformation parameters and the transformed coordinates are listed in *Table 4.4*. The differences in the three axes were calculated and are presented in *Table C.4*. The referenced coordinates were used to calculate the differences between coordinate variations along the Cartesian axes. The results are shown in *Figure 4.21*. The accuracy in the horizontal plane, i.e., along the x-axis and y-axis, were much better than the vertical direction, which might be due to the complex multistory structure of this scenario.

TABLE 4.4: Target coordinates in both point clouds (the Stairwell dataset, aligned)

Target ID	Coordinates by TLS (m)	Coordinates by S^2DAS (m)
T1	(1.302, 4.957, 7.005)	(1.282, 4.970, 7.080)
T2	(1.103, 1.046, 4.500)	(1.083, 1.102, 4.487)
T3	(1.827, 2.672, 3.961)	(1.855, 2.721, 3.954)
T4	(1.541, 4.939, 2.067)	(1.559, 4.946, 2.017)
T5	(1.071, 1.082, 0.488)	(1.062, 1.021, 0.541)
T6	(1.061, -1.791, 8.509)	(1.066, -1.841, 8.396)
T7	(1.225, -1.737, 4.617)	(1.205, -1.701, 4.572)
T8	(6.015, 3.334, 3.032)	(6.047, 3.363, 3.019)
T9	(1.370, -0.670, 0.557)	(1.357, -0.749, 0.670)



(a) (b)

FIGURE 4.21: The distribution of distance differences along the Cartesian axes between the targets in the TLS point cloud and the S^2DAS point cloud (the Stairwell dataset). (a) The distribution of the differences with respect to each of the targets along the Cartesian axes. (b) The distribution of all the differences with respect to the Cartesian axes.

4.3.3 Corridor

The corridor test site was located on the LG2/F, Block Z, HKPU, as shown in *Figure 4.22*. The corridor consisted of multiple right-angle turns and corners with tidy side walls. It was divided into two parts for testing the performance of the proposed solution in similar environments with different restrictions. As shown in *Figure 4.23*, the longer part of the corridor with a L-shape turn and a C-shape turn was used to test the non-loop mapping performance by comparing the two single-pass mobile mapping results with the TLS point cloud, while the other part was used to test the performance of the round-trip mapping method in narrow spaces.

The test of the single-pass mapping method was conducted in both directions along the corridor to eliminate the possible influences caused by the incident angles. The two datasets were called *Departing Trip* and *Returning Trip* to indicate the difference in moving directions. As they were independent mapping processes

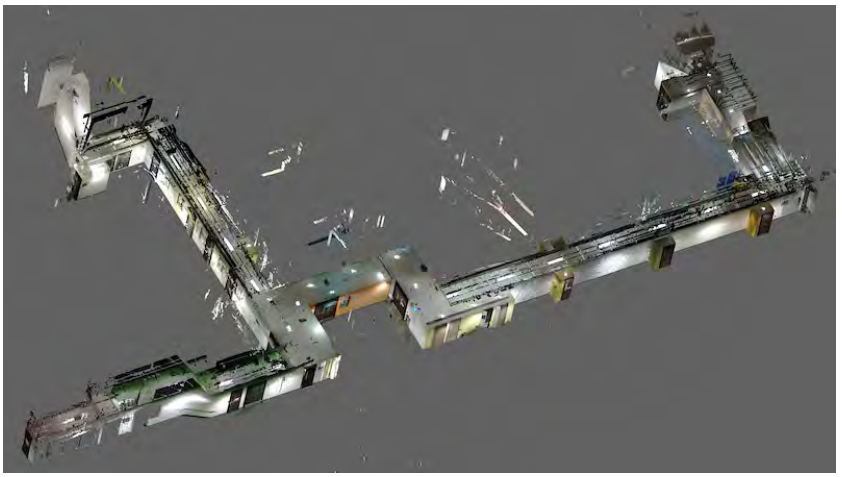
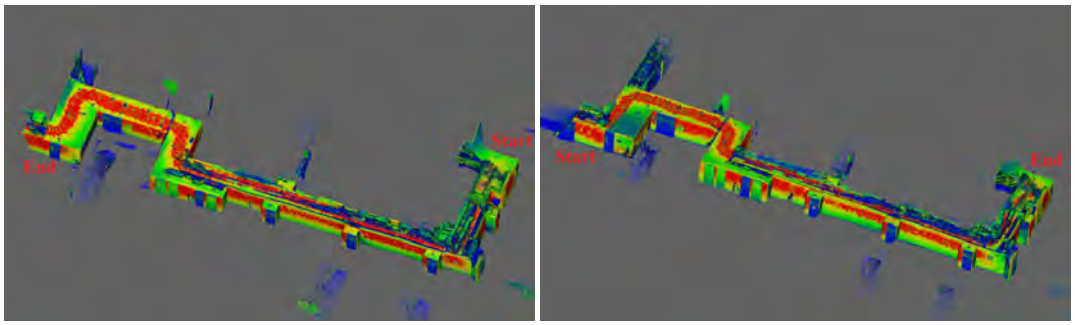


FIGURE 4.22: The point cloud of the long corridor captured by TLS.

FIGURE 4.23: The top view of the point clouds captured by TLS and the mapping regions of the two corridor datasets. The red box illustrates the region of the two-way single-pass trip while the blue one shows the region of the round trip for mapping the L-shape region.

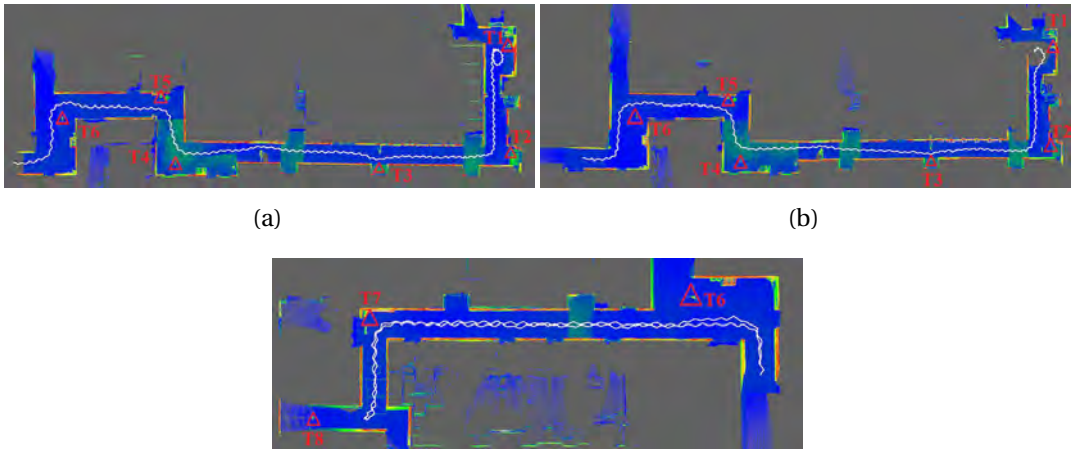
rather than round-trip mapping with adjustment, the repeatability and reliability of the proposed system could also be checked. In the meantime, the round-trip mapping of the L-shape part was called *Round Trip*. The result point clouds are shown in *Figure 4.24*, while the moving trajectories of the three point clouds are illustrated in *Figure 4.25*.

The same TLS point cloud, as shown in *Figure 4.22*, was used as the reference point cloud and multiple stations were implemented in case there was any detail



(a) (b)

FIGURE 4.24: The two point clouds captured by S^2DAS via routes in opposite directions. (a) The point cloud captured along the departing trip with the start and end positions indicated. (b) The point cloud captured along the returning trip with the start and end positions indicated.



(a) (b)

(c)

FIGURE 4.25: The moving trajectories of the two single-trip moving routes and the round-trip moving trajectory. (a) The point cloud and the trajectory along the departing trip, moving from T1 to T6. (b) The point cloud and the trajectory along the returning trip, moving from T6 to T1. (c) The point cloud captured in the round trip with white points indicating the trajectory, moving along the sequence as T6→T7→T8→T7→T6.

loss. The positions of the scan stations and the spherical and hemispherical targets are marked in *Figure 4.26*. The coordinates of the spherical and hemispherical targets are shown in *Table 4.5*.

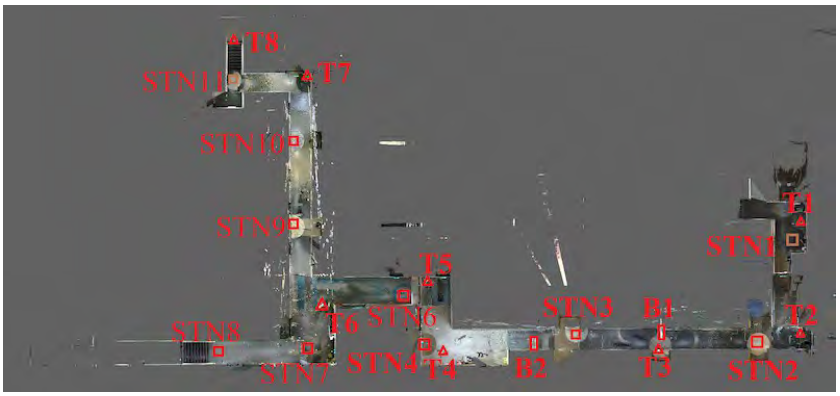


FIGURE 4.26: The distribution of the scan stations, which are rectangles labeled "STN1" to "STN11", targets, which are triangles labeled "T1" to "T9", and vertical reference boards used as extra references labeled "B1" and "B2".

TABLE 4.5: Target Coordinates in the TLS Point Cloud (the Corridor dataset)

Target ID	Type	Coordinates by TLS (m)	Radius (cm)
T1	Hemisphere	(7.258, -26.104, 0.252)	7.5
T2	Hemisphere	(-1.918, -27.142, -0.194)	7.5
T3	Hemisphere	(-4.625, -15.641, -1.273)	7.5
T4	Sphere	(-6.790, 2.233, 0.155)	10.0
T5	Hemisphere	(-0.887, 4.147, 0.003)	7.5
T6	Sphere	(-3.847, 12.548, 0.073)	10.0
T7	Hemisphere	(15.060, 15.550, -0.120)	7.5
T8	Hemisphere	(17.170, 22.119, 0.073)	7.5

In addition, multiple retractable banners were erected in the middle of the corridor to provide vertical planes for point cloud alignments as past experiments showed that this part of the corridor was too long for direct 6 DOF plane-based alignments, as shown in *Figure 4.27*. Though not completely flat, the banners were considered as planes concerning the measurement accuracy of the MLS. The distribution of banners is shown in *Figure 4.26*.

Consequently, the corresponding reference targets were extracted and their coordinates in the respective coordinate frames are listed in *Table 4.6*. Respectively, the coordinates of the three point clouds captured by the S^2DAS could be transformed

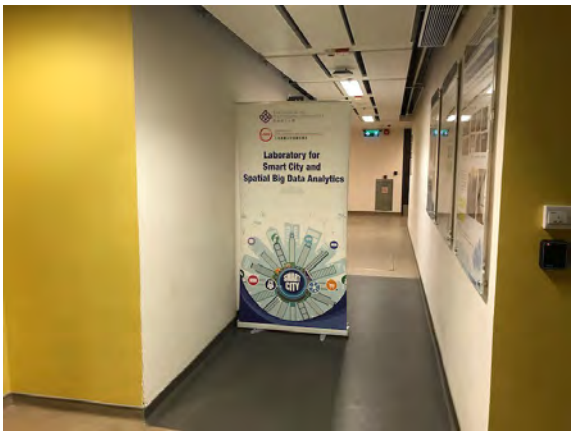


FIGURE 4.27: The vertical banners in the middle of the corridor as the reference planes.

to the coordinate frame of the TLS point cloud. The results are listed in *Table 4.7*.

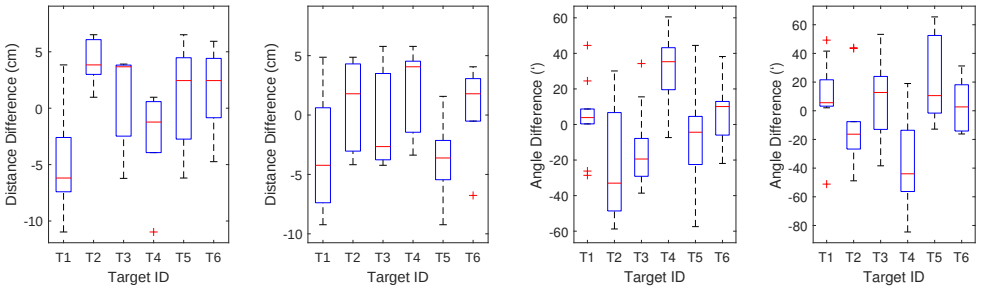
TABLE 4.6: Target Coordinates in the S^2DAS Point Clouds (the Corridor dataset, not aligned)

Target ID	Coordinates in Departing Trip (m)	Coordinates in Returning Trip (m)	Coordinates in Round Trip (m)
T1	(-0.949, 0.967, -0.407)	(12.154, 3.837, 27.823)	N/A
T2	(6.810, 5.962, 0.610)	(20.661, 7.193, 26.166)	N/A
T3	(14.298, -3.283, 0.298)	(24.584, -2.079, 19.939)	N/A
T4	(23.588, -18.554, 2.799)	(29.567, -18.156, 13.231)	N/A
T5	(19.201, -22.830, 1.902)	(24.139, -21.096, 13.316)	N/A
T6	(25.440, -29.122, 3.027)	(27.925, -27.906, 8.967)	(27.090, -26.763, 4.631)
T7	N/A	N/A	(8.924, -32.893, 3.836)
T8	N/A	N/A	(7.833, -39.637, 4.760)

The differences in distance and angles are listed in *Tables C.5-C.10* and *C.17-C.19*, while the distributions of errors are shown in *Figure 4.28 - 4.30*. As the edges and triangles formed in the *Round Trip* dataset were not enough, the distribution plot of that dataset was not listed. The comparison showed a similar pattern as the results in previous sections: errors in the vertical direction was larger than errors in the horizontal plane.

TABLE 4.7: Target Coordinates in the S^2DAS Point Clouds (the Corridor dataset, aligned)

Target ID	Coordinates in Departing Trip (m)	Coordinates in Returning Trip (m)	Coordinates in Round Trip (m)
T1	(7.252, -26.035, 0.219)	(7.280, -26.042, 0.281)	N/A
T2	(-1.954, -27.192, -0.079)	(-1.928, -27.155, -0.308)	N/A
T3	(-4.623, -15.661, -1.333)	(-4.623, -15.665, -1.242)	N/A
T4	(-6.734, 2.210, 0.050)	(-6.838, 2.252, 0.298)	N/A
T5	(-0.859, 4.161, -0.067)	(-0.948, 4.092, 0.121)	N/A
T6	(-3.892, 12.557, 0.225)	(-3.753, 12.559, -0.134)	(-3.879, 12.530, 0.073)
T7	N/A	N/A	(15.054, 15.563, -0.119)
T8	N/A	N/A	(17.208, 22.124, 0.073)

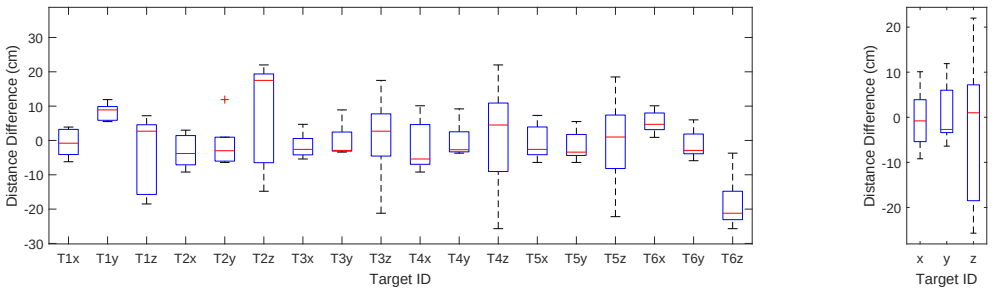


(a) (b) (c) (d)

FIGURE 4.28: The distribution of distance and angle differences between the targets in the TLS point cloud and the S^2DAS point cloud (the Single-trip Corridor dataset). (a) Departing Trip, distance differences. (b) Returning Trip, distance differences. (c) Departing Trip, angle differences. (d) Returning Trip, angle differences.

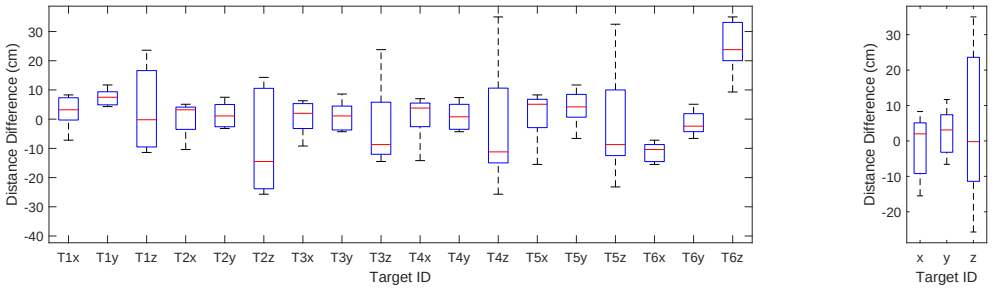
In addition to the numerical comparison results, the two single-pass results were plotted in the same viewer to check the repeatability of the proposed system, as shown in *Figure 4.31*. Contrary to the results shown in *Table 4.7*, *C.17* and *C.18*, where there were barely significant differences spotted, the visual differences were appreciable in specific view angles, which, on the other hand, showed the importance of the closed loop and control points and their role in overall adjustment. Some of the differences are shown in *Figure 4.32*.

Generally, the two passes from opposite directions produced similar results in



(a) (b)

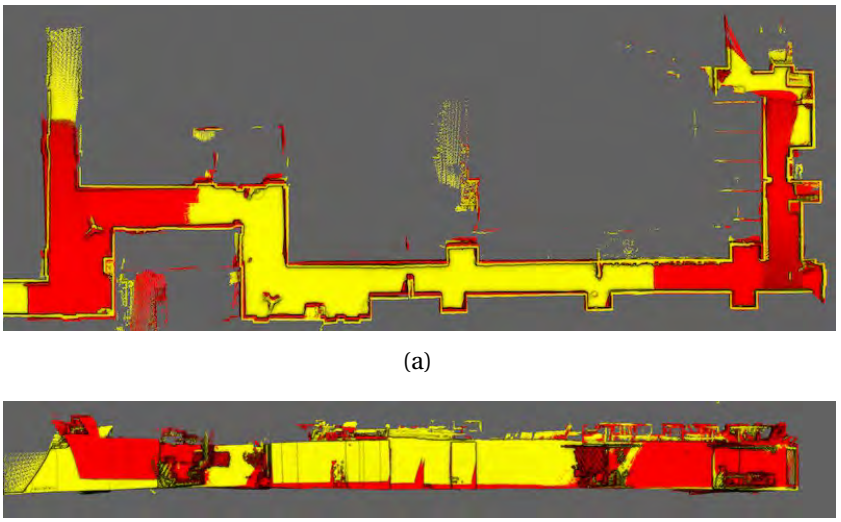
FIGURE 4.29: The distribution of distance differences along the Cartesian axes between the targets in the TLS point cloud and the S^2DAS point cloud (the Departing Trip, Single-trip Corridor dataset). (a) The distribution of the differences with respect to each of the targets along the Cartesian axes. (b) The distribution of all the differences in the Cartesian axes.



(a) (b)

FIGURE 4.30: The distribution of distance differences along the Cartesian axes between the targets in the TLS point cloud and the S^2DAS point cloud (the Returning Trip, Single-trip Corridor dataset). (a) The distribution of the differences with respect to each of the targets along the Cartesian axes. (b) The distribution of all the differences in the Cartesian axes.

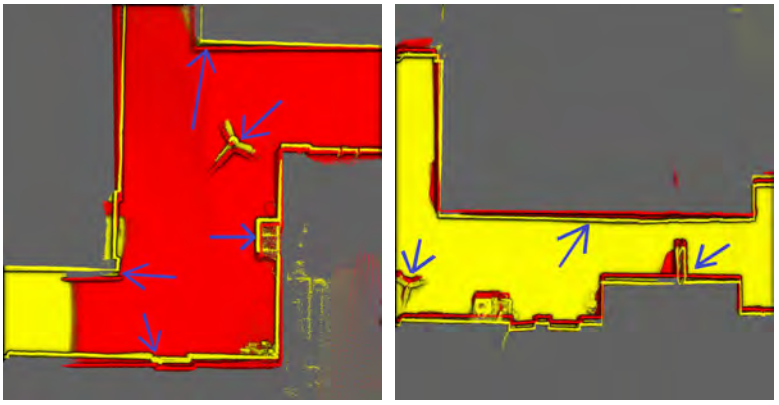
the distribution of errors concerning the differences in the coordinate changes. However, there were still appreciable differences between the two single-pass results, which proved the importance of the closed loops in the mapping procedure. In the meantime, the errors distributed along the vertical direction were much larger than errors along the two horizontal axes, which was similar to the previous comparison results.



(a)

(b)

FIGURE 4.31: The visual comparison of the two single-pass point clouds captured by S^2DAS (the Corridor dataset). The Departing Trip cloud is in red while the other cloud in yellow. (a) Vertical view. (b) Horizontal view. Significant differences can be spotted on the left side and the middle part.



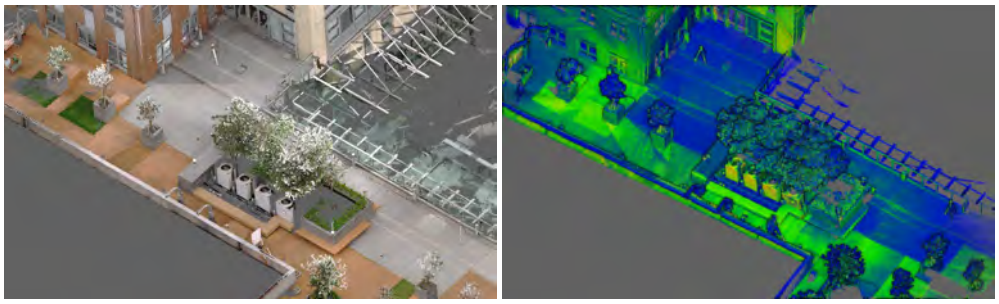
(a) (b)

FIGURE 4.32: Some of the appreciable differences between the two point clouds in the detailed vertical view. The arrows in blue marked the significant differences between the sidewalls and the tripods.

4.3.4 Outdoor Terrace

Although defined as an indoor mobile mapping solution, the backpack was used to capture point clouds of an outdoor terrace on 6/F, Block Z, HKPU, as shown in

Figure 4.33(a). The planar surfaces, which were the facade of the buildings on one side, the concrete walls of the flower beds and surrounding barrier, and the tiles and boardwalks, were used as the references for point cloud alignments.



(a) (b)

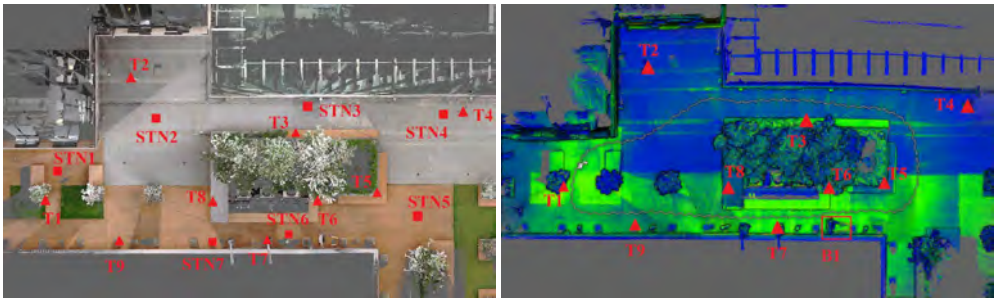
FIGURE 4.33: The point clouds of the outdoor terrace captured by TLS and S^2DAS. (a) The point cloud captured by TLS and colorized using the images captured by the embedded camera. (b) The point cloud captured by S^2DAS and colorized with respect to the point intensity provided by the multi-line scanner installed. Points in blue are with low intensities while those in green show high intensities.

Figure 4.33 shows the data captured using the FARO Focus M70 scanner and the S^2DAS mobile mapping backpack. *Table 4.8* lists the coordinates of the centers of the targets used for comparison.

TABLE 4.8: Target coordinates in both point clouds (the Outdoor Terrace dataset, not aligned)

Target ID	Type	Coordinates by TLS (m)	Coordinates by S^2DAS (m)	Radius (cm)
T1	Hemisphere	(1.839, 1.472, 9.055)	(1.198, -2.212, -0.475)	7.5
T2	Hemisphere	(-7.319, -4.174, 9.044)	(-5.515, 6.174, -0.550)	7.5
T3	Hemisphere	(-16.736, 4.099, 8.120)	(1.446, 16.628, 0.145)	7.5
T4	Hemisphere	(-28.396, 7.397, 9.152)	(3.199, 28.494, 2.240)	7.5
T5	Sphere	(-20.376, 10.419, 9.067)	(7.160, 20.899, 2.035)	10.0
T6	Hemisphere	(-16.53, 9.116, 8.682)	(6.360, 16.972, 1.258)	7.5
T7	Hemisphere	(-11.856, 10.430, 9.170)	(8.222, 12.393, 1.571)	7.5
T8	Sphere	(-9.375, 6.347, 9.060)	(4.518, 9.480, 0.782)	10.0
T9	Hemisphere	(-2.009, 6.385, 7.938)	(5.646, 2.277, -0.794)	7.5

Figure 4.34 shows the distribution of the scan stations, the targets, and the trajectory of the mobile mapping. An additional reference board was also placed on the ground, in case the number of the vertical surfaces might not be enough for providing sufficient alignment. The installation of the board is shown in *Figure 4.35* while the position is shown in *Figure 4.34*.



(a) (b)

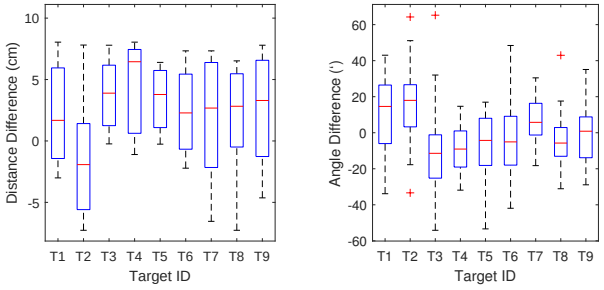
FIGURE 4.34: The point clouds of the outdoor terrace captured by TLS and S^2DAS. The positions of the target sphere and hemispheres are marked with triangles labeled "T1" to "T9". The rectangle marked with "B1" is the position of the reference board. (a) The red squares show the positions of the seven scan stations labeled "STN1" to "STN7". (b) The moving trajectory of S^2DAS shown in gray, with the labeled triangles showing the locations of the targets. The rectangle box in red labeled "B1" shows the location of the erect reference board.

The detailed differences in the distances and angles are listed in *Table C.11*, *C.12*, *C.20* and *C.21* in *Appendix C*, while the distribution of errors is listed in *Figure 4.36*.

Similar to the comparison in the previous sections, the targets were used to transform the S^2DAS point cloud to the coordinates in the reference system of the TLS point cloud. The coordinates after transformation are listed in *Table 4.9* and the distance and angle comparison results are listed in *Table C.11* and *C.12*. The results were used to calculate the distance differences along the Cartesian axes, and the boxplots are illustrated in *Figure 4.37*.

(a) (b)

FIGURE 4.35: The reference board installed beside the mapping path. (a) The front view of the board facing the opposite direction as the moving direction of the S^2DAS mobile mapping process. (b) The back view of the board facing the same direction as the moving direction of the S^2DAS mobile mapping process.



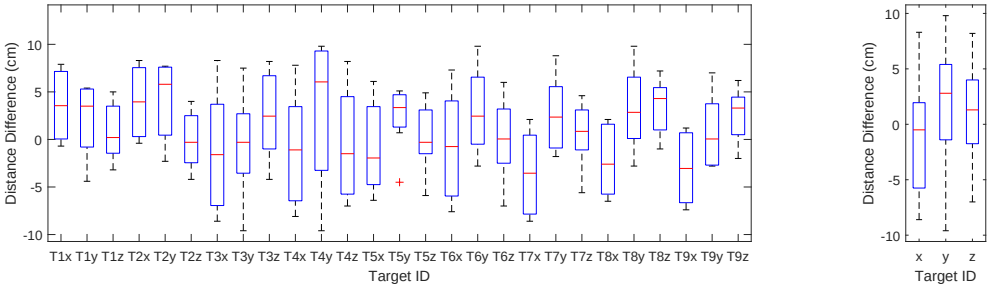
(a) (b)

FIGURE 4.36: The distribution of distance and angle differences between the targets in the TLS point cloud and the S^2DAS point cloud (the Outdoor Terrace dataset). (a) The distribution of distance differences. (b) The distribution of angle differences.

Accuracy along different axes showed no significant difference. All the absolute values of the difference were smaller than 10 cm, which might be resulting from the low complexity of the dataset. As assumed, the results showed the proposed solution was completely capable of working in such environments.

TABLE 4.9: Target coordinates in both point clouds (the Outdoor Terrace dataset, aligned)

Target ID	Coordinates by TLS (m)	Coordinates by S^2DAS (m)
T1	(1.839 1.472 9.055)	(1.871, 1.491, 9.062)
T2	(-7.319 -4.174 9.044)	(-7.283, -4.132, 9.041)
T3	(-16.736 4.099 8.120)	(-16.783, 4.066, 8.159)
T4	(-28.396 7.397 9.152)	(-28.438, 7.460, 9.109)
T5	(-20.376 10.419 9.067)	(-20.401, 10.431, 9.083)
T6	(-16.53 9.116 8.682)	(-16.567, 9.081, 8.709)
T7	(-11.856 10.43 9.170)	(-11.817, 10.405, 9.183)
T8	(-9.375 6.347 9.060)	(-9.357, 6.312, 9.027)
T9	(-2.009 6.385 7.938)	(-1.982, 6.378, 7.915)



(a) (b)

FIGURE 4.37: The distribution of distance differences along the Cartesian axes between the targets in the TLS point cloud and the S^2DAS point cloud (the Outdoor Terrace dataset). (a) The distribution of the differences with respect to each of the targets along the Cartesian axes. (b) The distribution of all the differences with respect to the Cartesian axes.

4.4 Summary

A plane-based point cloud alignment was proposed based on the plane extracted using the ELS algorithm presented in *Chapter 3*. Data captured using the prototype had been used to validate the proposed workflow. Both narrow corridors and large halls were used to test the feasibility, accuracy, and robustness of the hardware system and the software workflow.

The results showed that the proposed integrated hardware system and the data processing workflow were capable of mapping structured indoor environments and specific outdoor spaces when enough planes facing multiple directions could be identified for point cloud registration and alignments.

Chapter 5

Discussions

Methods used for extracting planes from low-resolution inhomogeneous point clouds and plane-to-plane alignments between such point clouds have been introduced and the results have been demonstrated and compared qualitatively and quantitatively. In this chapter, the methods and their advantages and disadvantages will be discussed.

5.1 Plane Extraction Based on ELS Extraction Results

By extracting geometric feature points along multiple scanline directions, generating line segments, merging parallel segments, and fusing the plane patches, planes are extracted from low-resolution inhomogeneous point clouds. The method is used in extracting planes from point clouds captured by a sixteen-line MLS. It shows great reliability in accomplishing the demanded jobs.

5.1.1 Applications of ELS-based Plane Extractions

The proposed method generated significantly better results when the resolution of the point cloud was not too high or when the distribution of points was not homogeneous, while there were not too many differences in plane extraction results for dense or homogeneous point clouds. The processing along the point grid reduced the effects made by the resolution change with respect to the distance between objects and the scanner, resulting in more planes extracted, especially floors, ceilings, and sidewalls in large spaces. The algorithm eliminated misinterpretations of short curvature segments as straight line segments and reduced the possibility of misclassifying the individual or connected Z-shape regions as planes with disturbances. The proposed algorithm also produced better results in congested areas as it did not require the estimation of the local normal vectors, which were fallible in such irregular dense regions.

The topological relationship between the points were considered based on the definition of the raw and virtual scanlines. Therefore, the small differences between planes can be identified. However, their topological relationships were not considered in the plane matching and alignment process, as the current implementation did not consider the relationships between planes as a factor.

5.1.2 Limitations of ELS Algorithms

The ELS algorithm had been proved to be an effective and efficient method for extracting feature points and planes, according to the test results above. The reliability and robustness of applying it on extracting planes from low-resolution inhomogeneous point clouds were significantly better than other state-of-the-art algorithms, which was essential for the application to the plane-based SLAM workflow. Nevertheless, there were still limitations and room for improvements.

5.1.2.1 Organized Point Clouds

As mentioned above, the successful implementation of the ELS algorithm requires the foregone knowledge of the grid distribution of points. All the tests of the proposed algorithm were conducted on the point clouds captured by TLS and MLS with known scanning sequence. When the sequence of data capturing conflicted with the geometric distribution sequence, the point grid would need to be reorganized to recover the geometric distribution of points, such as the shifting and projection operations mentioned in previous sections.

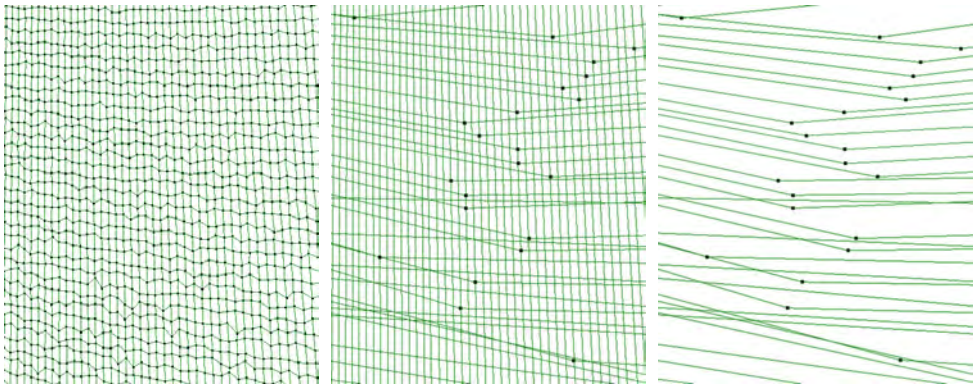
In other words, when given a point cloud with no information on the organized distribution of points, the point cloud would need to be reorganized. This is also the reason why downloaded open datasets may not be used to test the performance of the algorithm. Meanwhile, when the point clouds were formed by merging two point clouds and could not be perfectly reorganized, they could not be directly

processed using the proposed algorithm. Otherwise, a pre-process procedure with down-sampling and reorganized with respect to the assumed origin point would be required.

5.1.2.2 False Feature Points

The false identification of feature points was due to the unwanted scanline curvature with respect to the working mechanism of the horizontally rotating scanners and the feature point determination mechanism of the ELS algorithm. As introduced in W. Fan (2015), the ELS workflow examined scanlines and virtual scanlines as straight lines and identified breakpoints according to the distance between pending points and the corresponding line segments. However, as shown in *Figure 2.26* and *5.1*, the scanlines were not entirely straight lines in certain circumstances. Such curvatures existed in most of the scanlines. It was the length of the scanline segment that may have made the curvature not significant enough for identification. These curvatures became appreciable for understanding when non-feature points were removed from the grid view, leaving only horizontal feature points, as shown in *Figure 5.1(c)*. If the scanline segments were not long enough or the plane segments were not large enough, it would not have been possible for these curvatures to make variations significant enough for the ELS algorithms to extract feature points falsely. Nevertheless, for large line segments such as those on walls and ceiling surfaces, as in the example in *Figure 2.26*, there was a possibility that the ELS algorithms would

identify false breakpoints along the horizontal scanline direction, slicing the scanlines into shorter segments.



(a) (b) (c)

FIGURE 5.1: The recovered grid showing the curved scanlines in a TLS point cloud. The three subfigures are showing the same region of the TLS point cloud. The black dots illustrate the points while the line segments are connecting the neighboring points and feature points. (a) the grid view of the raw TLS point cloud (b) the grid view of the processed TLS point cloud in which feature points with respect to their directions are connected. (c) the grid view of the processed TLS point cloud with only horizontal feature points connected sequentially.

Generally, such drawbacks amplified the curvature and direction changes between neighboring points to the directional differences of line segments. In the proposed clustering process, the direction differences in neighboring scanlines were overcome by the reduced difference between the sharing neighbor, as shown in *Figure 5.2*.

However, if the unwanted curvatures could be rectified or modeled before the feature point extraction process, the possibility of such feature points would be eliminated, resulting in fewer scanline segments. As the existence of such curvatures indeed converts the 2D scanline to a 3D shape, an improper process would cause the dimension loss problem.

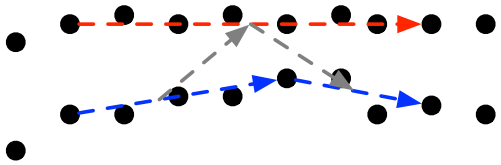


FIGURE 5.2: Ignoring the direction difference in scanline segments by introducing the sharing neighbor segment. The two segments with blue dashed arrows showing they are pointing at different directions while the segment, which is labeled with the red dashed arrow showing its direction, is their sharing neighbor segment. The search and clustering direction is along the gray dashed arrows and, therefore, the three segments could be segmented to the same group regardless of the existence of the feature point due to scanline curvature.

Meanwhile, another possible solution was that by identifying each of such curved scanlines as a piece of the planar fraction instead of several 2D line segments and merging the fractions based on the planar features. The idea is similar to the work done by Grant et al. (2013), whose idea was considered as not universally applicable.

5.1.2.3 Adaptive Threshold and Feature Point

The ELS algorithm was based on the well-known 2D line simplification algorithms, which were designed to simplify polylines by using most representative feature points to omit unnecessary detailed curvature changes (Z. Li, 2006) (Shi & Cheung, 2006). Consequently, limitations of conventional line simplification algorithms were also affecting the performance of ELS-based plane extraction workflow.

Firstly, the threshold values used in most line simplification algorithms were

constant values. In other words, the determinations of the thresholds were generally based on empirical values and experiences, or on an interactive selection process. In most cases, especially for the algorithms suitable for 3D applications, the thresholds were not affected by the geometric approximability or resolution of the point clouds, or the length of the pending line segments, which brought the problem, as introduced in *Section 3.1.5*. Therefore, an adaptive threshold determination process would hopefully reduce the possibility of such problems since the adaptive value would be adjusted with the distribution of the points in three dimensions.

Secondly, selecting the feature points would also be an adaptive process in which the extra feature points indicating better slices of scanline segments could be introduced. As shown in *Figure 5.3*, there are always differences in the geometric representations and the reality of the objects. The introduction of the extra feature points for scanline segmentation could generate better end point positions rather than the method proposed in *Section 3.2.2*. Meanwhile, the extra feature points may enable the identification of the line segments with no non-feature points, which would benefit the extraction of small-sized planes. However, this process would be time consuming and computation dependent, which would not be appropriate for real-time processing as this process would require pre-processing and understanding of the curvature patterns.

(a) (b)

FIGURE 5.3: An example of the adaptive feature point selection process. (a) The geometric representation in the form of dashed lines by connecting scanned points in blue. (b) The possible reality that the two scanline segments intersect at the position of the red point. The red point, which does not exist in the raw point cloud, should be the actual position of the feature point.

5.1.2.4 Scanline with Only Feature Points

Given the working mechanism of the proposed method, there should be at least one non-feature point existing. Consequently, the proposed method could not identify scanline segments without non-feature points, even if they were part of a plane. As four directions were defined in the workflow, for two-line strips as shown in *Figure 3.15*, it might be possible to identify non-feature points only when the extension direction was along the horizontal direction. Moreover, for the remaining three directions, all points would be identified as feature points and no corresponding line segment would be detected. Therefore, the unsuccessful identification of non-feature points resulted in the unproductive extracting results, as mentioned before and shown in *Figure 3.15*.

5.1.2.5 Time Consumption

The algorithm and workflow proposed were designed to be applied to the batch processes of extracting planes from such low-resolution inhomogeneous point clouds, and time was important for both real-time processing and post-processing as the

accumulation of short process span of thousands of frames would be plenty of time that could not be ignored. In most of our experiments, there were more than 6,000 frames of LiDAR point clouds, consisting of 12,000 frames of point clouds consisting of 25,000 points on average.

Most of the algorithms in the workflow were implemented based on PCL, such as SVD for segment directions, RANSAC for noise removal and parameter determination, and k-d tree indexing for nearest neighbor search, thus some of the processes could not be implemented in parallel sessions. The most notable example here was the RANSAC process. A sharing random number generator was implemented in the library and implementing a parallel RANSAC process would have caused the unwanted inconsistent random number, which would cause the unreliable subset selection results that there might be difference existing randomly between the two processes of the same point cloud. Nevertheless, the RANSAC processes were the most time-consuming part of the complete workflow, taking up to 50% of the total span. The unsuccessful implementation of a parallel RANSAC process indeed restrains the performance improvement of the whole workflow.

5.2 ELS-based 3D Point Cloud Alignments

Based on the plane extraction results utilizing the proposed ELS-based plane extraction method, the low-resolution inhomogeneous point clouds are aligned into the

same reference frame for mobile mapping. Experiments were performed in multiple scenarios and the results show the proposed method is capable of IMU-free point cloud registrations with the given mobile mapping platform, S^{2}DAS.

5.2.1 Accuracy Assessment Results

The accuracy assessment results based on the distance and angle checking using the extracted spherical and hemispherical targets are listed in *Table C.22*. As shown in *Figure 5.4*, the accuracy in the horizontal plane was better than the value in the vertical direction, while the average absolute distance differences in both horizontal directions were smaller than 10 cm. The distance measurement accuracy was relevant to the length of the distance measurements, which was better than 1%. The performance of the proposed system met the requirements of the system design, as listed in the ITSP proposal.

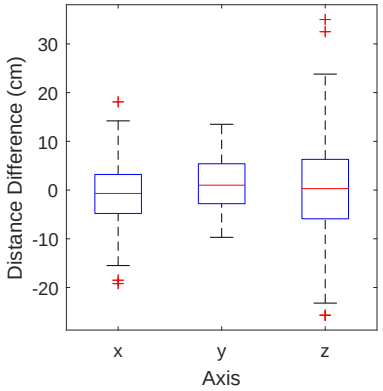


FIGURE 5.4: The distribution of all the distance differences and percentages in the Cartesian axes.

On the other hand, as an IMU-free system was not available on the open market yet, a comparison with similar products could not be made. The proposed system can achieve the same accuracy level as the commercial products, which declared that the accuracy of tens of centimeters could be achieved in indoor environments (Leica Geosystems AG, 2015) (Nocerino et al., 2017) (Lehtola et al., 2017). Referring to Geiger et al. (2012), which is currently one of the most popular benchmarks for SLAM evaluation, the translation accuracy of the proposed system was better than most of the methods listed, except the recent new champion by Dimitrievski, Van Hamme, Veelaert, and Philips (2016). Moreover, as the rotation accuracy listed in the benchmark was corresponding to the attitude of the platform, it could not be directly compared with the angle accuracy evaluated in this report. Notably, most of the solutions listed on the KITTI benchmark website were integrated with IMU and GPS while the proposed system was an IMU-free system working in GPS-denied environments.

5.2.2 Further Improvements

As some of the failure cases showed, the working mechanism of the proposed method was also its main drawback. The proposed method was not capable of providing ultimate solutions for position and orientation changes when adequate planes were not identified or differentiation in the facing directions was not adequate. Therefore, the IMU-free solution itself did not fit the applications to mapping

natural environments, such as caves, cylinder tunnels, and every scenario with neither flat floor nor planar ceiling. Also, the size of the boards used as premeditated landmarks should not be too small. Otherwise, they would not be identified by the multi-line MLS, which only covered limited FOV.

IMU, which was already installed on S^2DAS but not used in the proposed method, had been proven to be a useful DR sensor in most academic and commercial solutions (Ulas & Temeltas, 2012) (J. Zhang & Singh, 2014) (J. Zhang & Singh, 2015) (Leica Geosystems AG, 2015) (Geneva et al., 2018) (ViAmetris 3D Mapping, 2019) (GreenValley International, 2019). They were designed to provide continuous data streams of accelerations and were barely affected by the environments. Although the position and orientation readings, which were integral values of accelerations, drifted with time, multiple methods have been introduced to reduce such errors, which made it possible to use the high-accuracy plane-based mobile mapping result to calibrate IMU drifting, which was similar to the visual odometer in J. Zhang and Singh (2014) and J. Zhang and Singh (2015).

On the other hand, the IMU readings could be fused with the proposed methods and would result in (1) providing initial alignments rather than the inaccurate NDT alignments, as the preceding drifts had already been calibrated, which would reduce the total time as well since the tedious, time-consuming NDT process has been skipped, (2) assisting the alignment process to achieve better quality and smaller residuals to reduce the number of key frames as they were corresponding to the unsuccessful alignments, and (3) generating more reliable alternative position and

orientation solutions if there were no valid plane matching and alignment results rather than using the NDT alignments or motion extensions of previous frames. As a designed feature of S^2DAS, the integration between the proposed method and IMU could be either online or offline. Both procedures are shown in *Figure C.1* in *Appendix C*. Furthermore, other SLAM methods, such as vSLAM, were also introduced and planned in the proposal of S^2DAS, which was designed as a future commercial solution.

Chapter 6

Conclusions

In this thesis, recent developments in indoor mobile mapping solutions have been reviewed and discussed. To fulfill the requirements of IMU-free alignment of point clouds captured in indoor environments, an ELS-based plane extraction method was designed to detect and identify planes from single frames of low-resolution inhomogeneous point clouds captured by multi-line MLS. A mobile mapping backpack was presented as the testing platform for verifying the proposed IMU-free point cloud alignment workflow, which used the planes extracted to build point clouds of the desired environments. The results show that the proposed sensor integration and data processing methods were capable of generating point clouds with acceptable accuracy in certain environments consisting of enough planes with the differentiation in facing directions. As a mobile mapping solution package seeking commercial opportunities, the necessity of introducing IMU and other SLAM techniques into the proposed plane-to-plane point cloud alignment method has

also been discussed.

6.1 Contributions

The works presented in this thesis included plane extraction from low-resolution inhomogeneous point clouds, the prototype design for testing the performance, and the plane-based 6 DOF IMU-free point cloud alignment method, forming the complete data capturing and processing workflow and the corresponding hardware system using the proposed methods. The main innovations in the presented work include a plane extraction method utilizing the linear distribution of points in low-resolution inhomogeneous point clouds, a dedicated coarse-to-fine procedure for IMU-free initialization between adjacent frames, and a shortest-path initialization strategy to alleviate the accumulation of drifts in alignments. Meanwhile, the corresponding contributions of the research are introduced in this section.

Robust Plane Extraction from Low-Resolution Inhomogeneous Point Clouds. Although many scholars had proposed plane-based SLAM in the past decades, no commercial solution was based on these algorithms. One of the most critical reasons was the lack of a robust plane extraction algorithm designed for low-resolution inhomogeneous point clouds captured by multi-line MLS. Indeed, considering the limitations of the ELS-based plane extraction method listed, it was not a perfect and universal solution for extracting planes from all kinds of 3D point clouds. However, the proposed algorithm fulfilled the plane extraction requirements as desired, and

it was proved to be reliable for mapping indoor environments and certain outdoor environments.

Hardware Design Using Planes for 6 DOF Point Cloud Alignment. Although most academic and commercial solutions integrate two laser scanners to cover larger FOV, the purpose of the more vertically installed scanners was similar to the profiler in outdoor mobile mapping systems, while the horizontal scanners were designed for SLAM positioning. In some of the solutions, the data captured by the vertical scanner had also been used for SLAM and improved accuracy and reliability in 6 DOF. In the proposed system, the vertical scanner was designed to enlarge the FOV for capturing planes that could be used to determine the motion in vertical directions, such as ceilings and floors. It was also one of the main reasons the system could be used for mapping stairwells without IMU, which were the most challenging scenarios for other solutions.

Coarse-to-fine Alignment for Planes Matching. The introduction of the coarse-to-fine alignment process solved two problems in IMU-free point cloud alignments: it provided (1) the initial estimation for better plane matching and (2) the alternative solution when plane matching failed. For frame-to-frame alignments between key frames, the drift errors of the position and orientation were inevitable. NDT alignments were conducted to provide coarse alignments that were better than the drifted initialization poses and more reliable for matching corresponding planes. On the other hand, when planes were not enough, or were not facing different directions that facilitated progress in 6 DOF estimation, the NDT

solution provided another choice in addition to extending the previous motion between the preceding frames.

Shortest-path for Alignment Initialization. In addition to the coarse-to-fine alignment process, the redundant alignments were adopted to generate better initial pose for NDT rather than the direct frame-to-frame transformation passing. Such implementations reduced the possibility of NDT coarse alignment failure and increased the chances of successful plane matching. Besides, the coarser initialization could reduce the time used for NDT alignment as well.

An IMU-free 6 DOF LiDAR point alignment method that can work in stairwells. Based on the algorithms and hardware mentioned above, several typical indoor scenarios were used to test the accuracy and reliability of the proposed system. The system had been proved to be a practical indoor mobile mapping solution without the presence of IMU. Unlike most of the solutions which required shape or distance changes in the point clouds to work, the proposed system could work in any scenario if a notable number of planes facing multiple directions could be spotted. The proposed IMU-free solution could work in stairwells as well, which was one of the most challenging environments for most of the other mobile mapping solutions, waiving the extra GCP in capturing the reality of multiple floors for registration.

6.2 Limitations and Open Problems

The proposed solution generated promising results. However, the limitations of the proposed hardware system and data processing workflow were appreciable. There were open problems left for solving.

Adaptive Threshold in Feature Point Extraction. Multiple thresholds had been implemented in the data processing workflow, reducing errors and misclassifications. However, most of the thresholds were empirical values based on numerous tests and comparisons, regardless of the given testing environments. If these thresholds were determined in an automatic process based on data processing results of numerous scenarios, such as a machine learning process, the thresholds would be universal. For example, if an unsatisfied result were produced, human operators would check the processing results manually and spot the errors while the algorithm could analyze and learn these results to generate a set of refreshed thresholds. Then the data could be reprocessed using the updated parameters to generate more satisfying results.

Small Plane Detection. As mentioned in *Chapter 3*, because they could only be identified in one of the four checking directions, the small planes consisting of parallel scanlines with only feature points were not correctly identified as planes by the proposed methods. However, in complex environments, such as corridors and office rooms with partition boards, such planes were commonly captured by the multi-line MLS because their vertical angular resolutions were relatively low. In

such complex scenarios, the successful detection of such planar stripes is crucial as they might be the only planar regions facing the particular direction and used for alignments. Therefore, a better detection strategy would be necessary for such applications.

Planes as Features and Landmarks. In the proposed workflow, planes were used as features for recovering the geometric alignments between frames of point clouds. However, there were two more possibilities for applying plane features in the SLAM process. The first was introducing the planes as features in addition to those feature points in the archived algorithms, such as LOAM. The extra features might be possible for improving accuracy and reliability in the SLAM process. The second one, all planes used in alignments in the proposed methods were unconnected entities rather than merged and updated planes. Therefore, planes with no overlapped areas, which were different parts of the same plane and extracted from different frames, could not be identified as corresponding planes, reducing the success rate of plane matching. Moreover, as there were inevitable misalignment errors and residuals in each of the successful alignments, the merging and update process would contain ambiguity of plane positions and thickness, and require certain refining process. Otherwise, the disturbances around the plane would result in high uncertainty in the alignment process.

Key Frame Detection. As introduced in *Chapter 4*, identification of the key frames was based on the number of matched planes between the current pending

frame and the nearest preceding key frame. Consequently, not only the changes between the point clouds but also the misalignment errors would affect the number of key frames. In other words, when the misalignment errors between the pending frame and the preceding key frame accumulated to a certain level with respect to the corresponding candidate planes for matching, an unnecessarily key frame might be identified. Furthermore, the ascending number of key frames would decrease the time efficiency of the whole workflow as a massive number of pairs of extra key frames would be asking for redundant observation building. A better implementation of key frame detection and identification, such as reduced misalignment errors based on the IMU-LiDAR data fusion, might be a possible improvement in the future.

Integration with IMU and Loop Detection. The possibility and feasibility of integrating the proposed method with IMU has been discussed in *Chapter 4*. In addition to the precise estimations for initial plane matching, the IMU could also be used to provide trajectory with an acceptable level of accuracy for detecting loops online. Also, the IMU, together with other SLAM techniques if available, could be used to provide position and pose changes when the proposed alignment method cannot generate reliable estimations in addition to providing the initial estimation of point cloud alignment.

6.3 Future Works

In addition to the aforementioned open problems, there are possible working directions in the future, to improve the unsatisfactory specifications of the current version of S^2DAS in precision, accuracy, and reliability.

Integrating Multiple SLAM Techniques. The feasibility of the proposed workflow based on pure point cloud alignments with no IMU have been proved with a certain level of accuracy and reliability. It was not a perfect solution for all the artificial environments because there were always spaces that did not have enough planes for registration. However, it did prove that the proposed method could be used as a part of the SLAM workflow to enhance robustness. To make S^2DAS a commercial-grade mobile mapping solution, which would be used in multiple scenarios, the proposed method would have to be organically combined with other SLAM techniques and positioning and orientation sensors to form a comprehensive solution for both outdoor and indoor mobile mapping. As introduced in *Chapter 4*, the next-step integration plan included data fusion with the panoramic vSLAM, which was Panoramic Direct Feature SLAM (PDFSLAM), and IMU DR techniques. For example, 7 DoF ICP can be directly applied for fusing vSLAM and LiDAR mobile mapping trajectories, while IMU results may be used as either an initializer to replace NDT results as it is too slow or an error detector to remove the wrong alignment results, improving the overall reliability.

High-accuracy Indoor Mobile Mapping Platform. In outdoor mobile mapping based on vehicle platforms, the positioning accuracy of the onboard GPS/IMU system was around ten times worse than the measurement accuracy of the laser scanner. However, benefiting from the adjustment process, the relative accuracy of the final point clouds was acceptable, considering the distance measurement accuracy of the laser scanner adopted on such platforms was usually a few millimeters. In the meantime, the accuracy of the multi-line MLS adopted in most indoor mobile mapping solutions was around 3-5 cm, while the overall positioning accuracy of the indoor platform could achieve a similar level of accuracy as the outdoor solutions. Therefore, if the current mobile mapping backpack, S^2DAS, were to be used as a positioning and orientation platform on which a millimeter-accuracy laser scanner was installed as the profiler, the possibility of such a system generating high-accuracy point clouds would be practical. The weight and synchronizing problem must be fixed to enable its wide applications.

Time Efficiency. The time efficiency of the proposed C++ implementation was still not practical for real-time processing. For a dataset consisting of around 16,000 frames of point clouds, which was captured in a large lecture hall, nearly 1,300 frames were identified as key frames, and it took almost 13 hours to process the data. As previously discussed, the considerable number of key frames was one of the main reasons that made the whole procedure time-consuming. This time could be improved with the following aspects: (1) revising the key frame identification criteria, (2) introducing IMU estimations to reduce the misalignment errors

and residuals and the time cost on NDT coarse alignments, (3) improving the loop closure and key frame relationship building process by a pre-check process based on corrected IMU trajectories, and (4) introducing parallel processing in building key frame relationships.

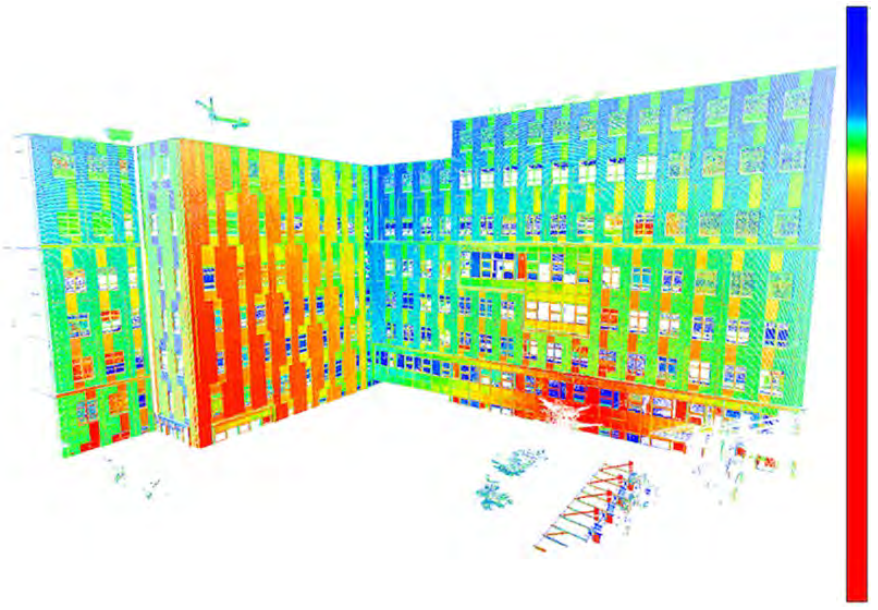
Appendix A

Plane Extraction Flowcharts and Results

TABLE A.1: The rearranged sequences, raw fire ID, vertical angles, time offsets and horizontal time and installation offsets at the rotation speed of 20 Hz / 1200 rpm from the first point of the 32-point group of RoboSense RS-LiDAR-32 3D scanner according to Suteng Innovation Technology Co Ltd (2015) (∗)

Rearranged Sequence ID	Fire ID	Vertical Angle (°)	Time Offset (μs)	H. Time Offset ($'$)	H. Installation Offset (°)
1	16	-24.9706	50.0	21.600	-7.73545
2	18	-14.6883	56.0	24.192	-7.74978
3	0	-10.3157	N/A	N/A	8.19589
4	20	-7.9978	62.0	26.784	-7.91365
5	2	-6.4594	6.0	2.592	8.25322
6	22	-5.4780	68.0	29.376	-7.83230
7	30	-4.6670	92.0	39.744	7.83511
8	28	-4.4042	86.0	37.152	2.56844
9	26	-4.0713	80.0	34.560	-2.69507
10	24	-3.7027	74.0	31.968	-8.03434
11	23	-3.3152	71.0	30.672	7.63674
12	21	-2.9643	65.0	28.080	2.30018
13	19	-2.6849	59.0	25.488	-2.98962
14	17	-2.3687	53.0	22.896	-8.25363
15	31	-1.9821	95.0	41.040	7.71533
16	29	-1.6670	89.0	38.448	2.42453
17	27	-1.3330	83.0	35.856	-2.82134
18	25	-0.9642	77.0	33.264	-8.29783
19	7	-0.6312	21.0	9.072	7.61699
20	5	-0.3330	15.0	6.480	2.28952
21	3	0.0000	9.0	3.888	-3.06551
22	1	0.3151	3.0	1.296	-8.42088
23	15	0.7028	45.0	19.440	7.54721
24	13	1.0000	39.0	16.848	2.31369
25	11	1.3330	33.0	14.256	-2.96384
26	9	1.6670	27.0	11.664	-8.29716
27	4	2.3330	12.0	5.184	7.76623
28	6	3.3330	18.0	7.776	-7.73591
29	8	4.6136	24.0	10.368	8.01336
30	10	7.0000	30.0	12.960	-8.27124
31	12	10.2983	36.0	15.552	8.32025
32	14	15.0334	42.0	18.144	-7.74762

(∗) Note: the horizontal offsets between scanners are different due to assembly errors.



(a)

(b)

FIGURE A.1: The raw data and the ELS processing result used for examining the algorithm optimization result. (a) The rainbow-color point cloud of the Z block building of HKPU containing 1,612,070 points (989 × 1630). The color is defined by the intensity values of the points with the color ramp showing at the right side of the figure. (b) The feature point extraction result using the enhanced Douglas-Peucker algorithm with the threshold of 0.03 m. Only 254,755 points were extracted (15.8% maintaining rate) while 163,026 points were extracted along the vertical scanline direction and 150,588 points were extracted along the horizontal scanline direction.

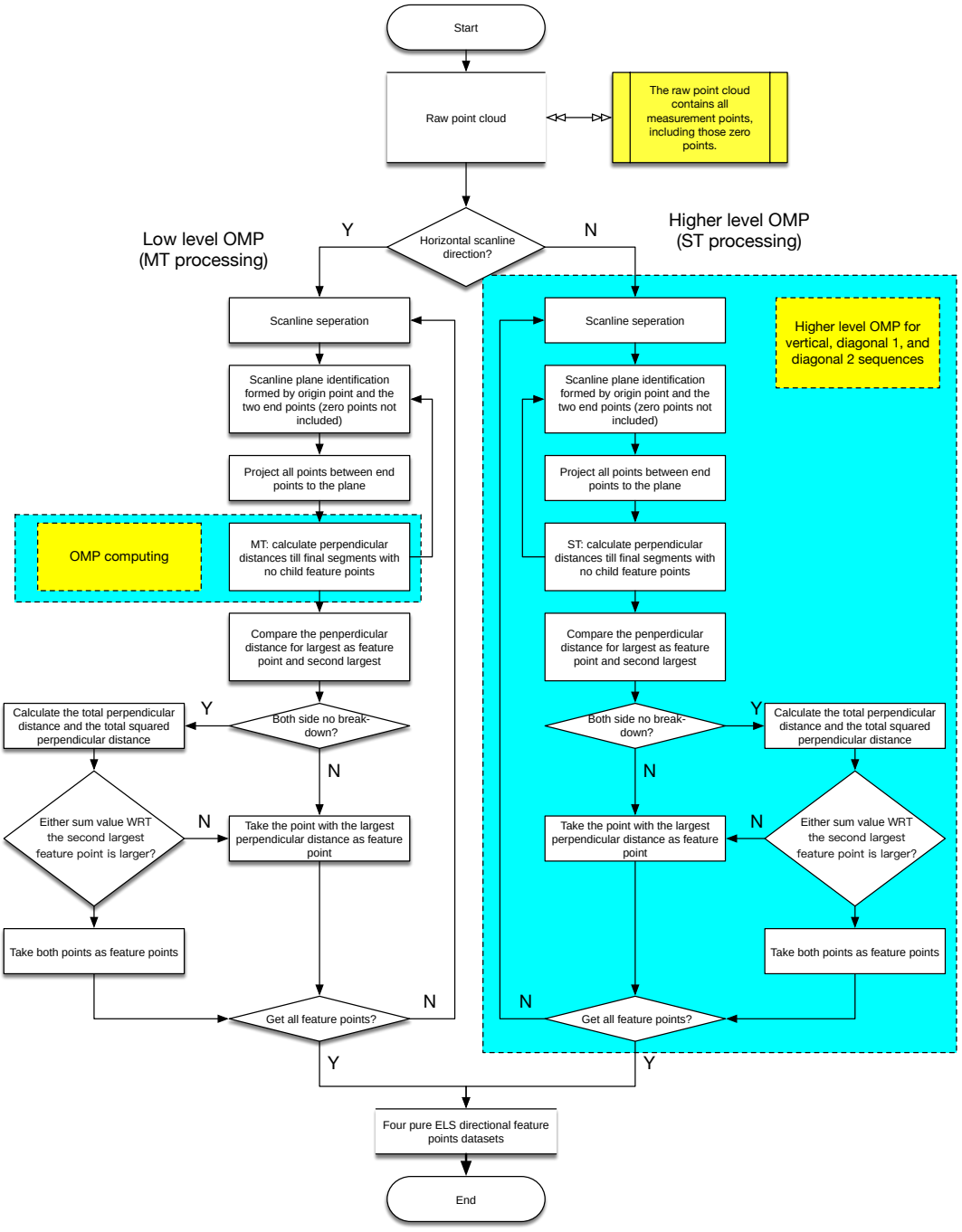


FIGURE A.2: The overall flowchart of the ELS algorithm for feature point extraction.

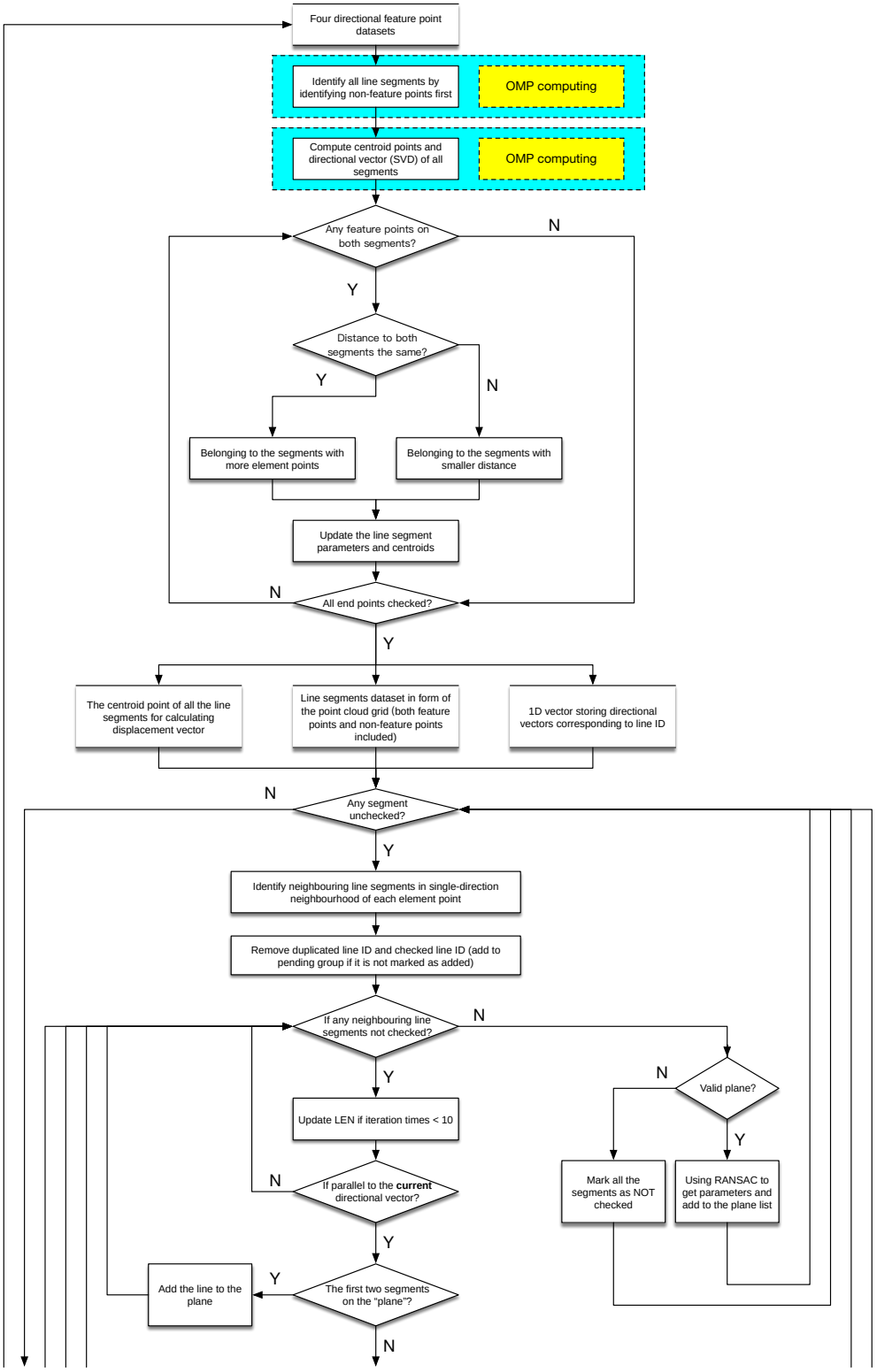


FIGURE A.3: The overall flowchart of the plane extraction workflow based on ELS results (Part I).

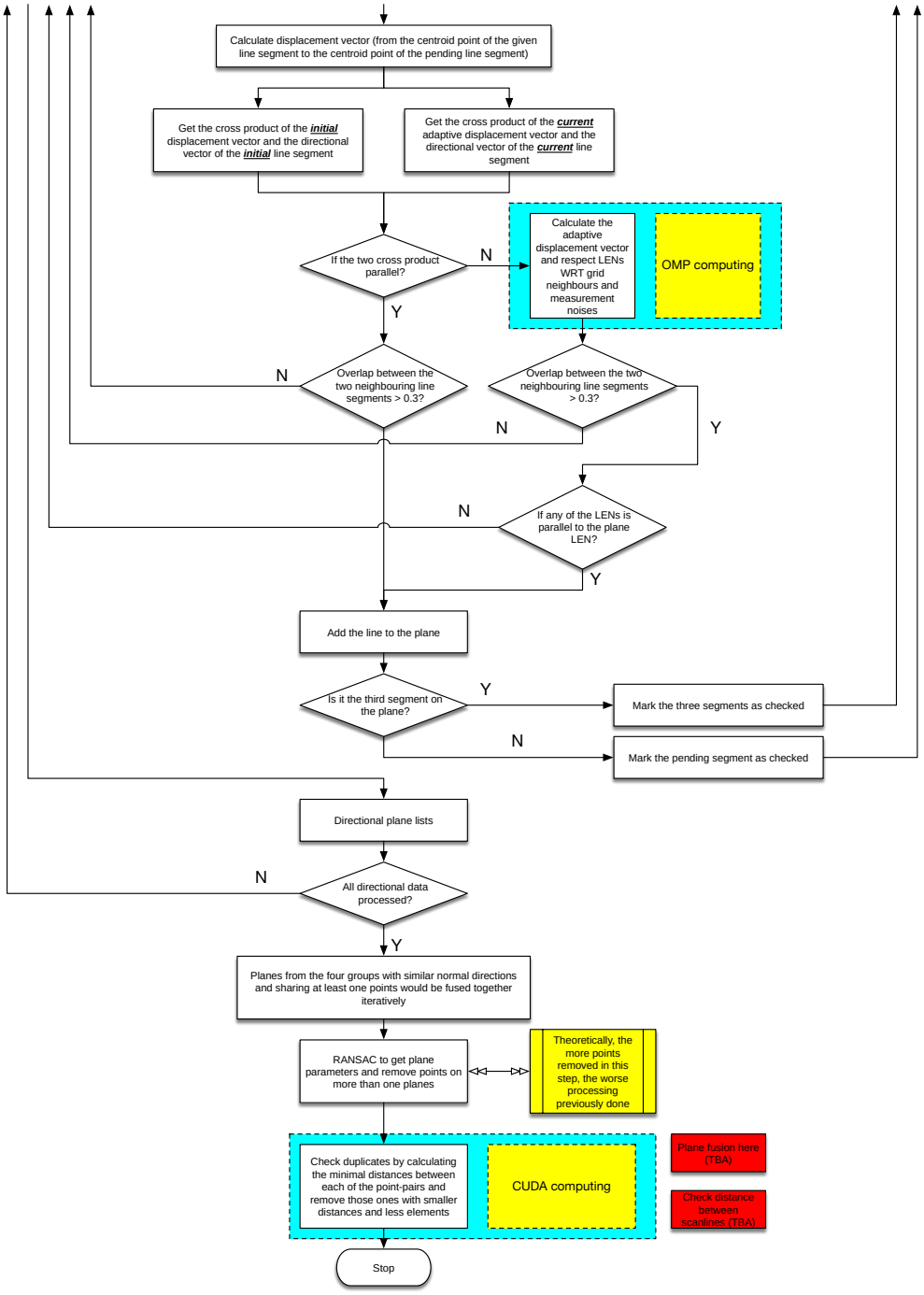
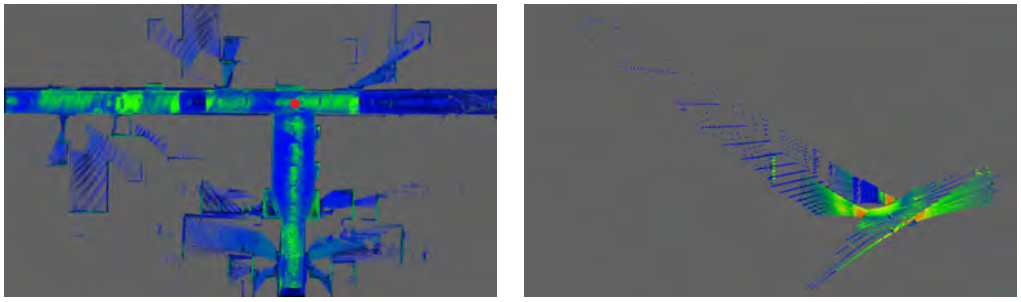
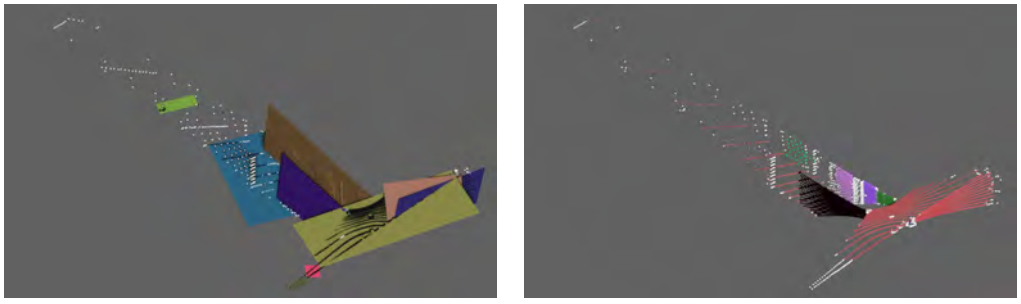


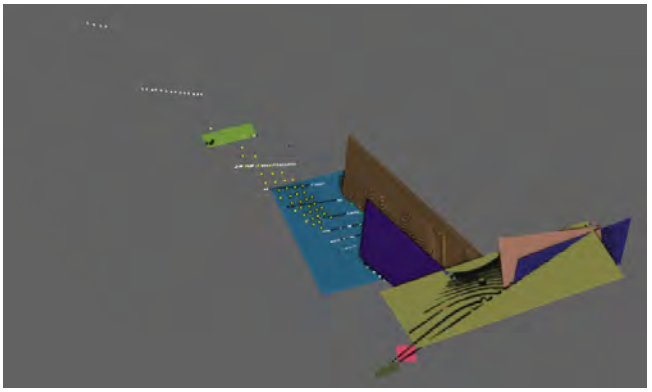
FIGURE A.4: The overall flowchart of the plane extraction workflow based on ELS results (Part II).



(a) (b)



(c) (d)



(e)

FIGURE A.5: Plane extraction results of the T-junction corridor dataset using both the RANSAC method and the proposed method. (a) The top view of the point cloud of the testing area with the red point indicating the data acquisition position. (b) The raw point cloud captured. The points are rendered with respect to the point intensity with blue as the lowest and red as the highest. (c) The RANSAC extraction results with white points showing points not extracted as planes while the colored blocks indicating extracted planes. (d) The points in different colors show various extracted planes using the proposed method. (e) Point clouds with planes extracted using both methods. The points in white indicate the incomplete extracting result of the floor while the points in yellow shows the unsuccessful extraction of the side-wall.

Appendix B

S^2DAS Design Diagrams

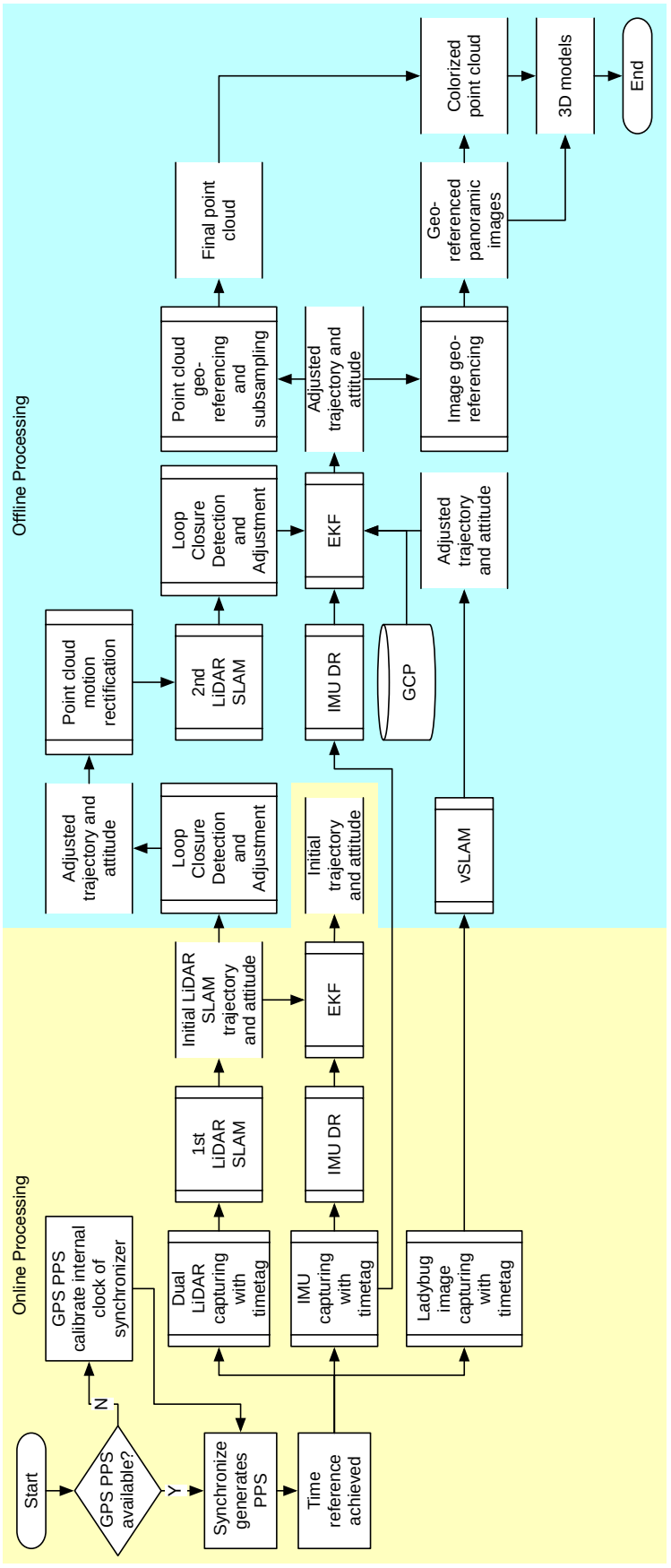
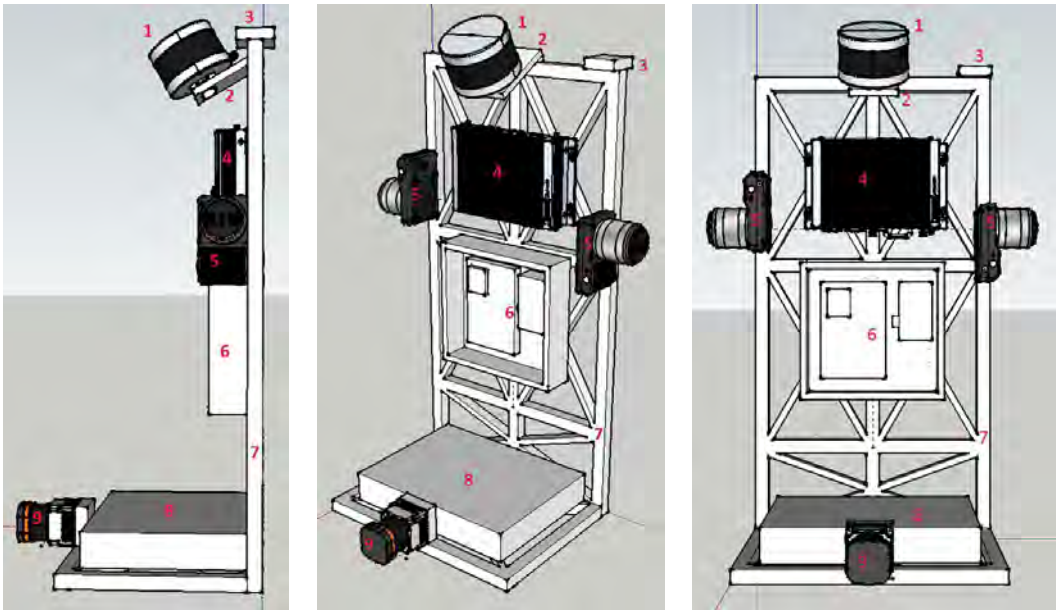
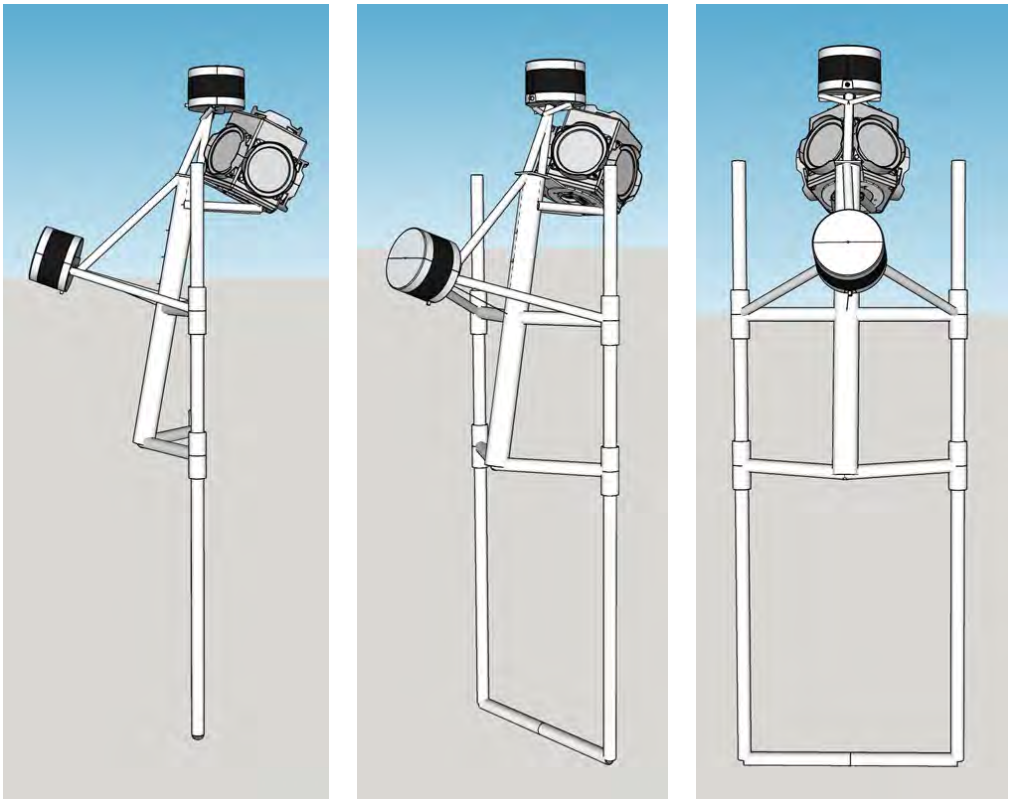


FIGURE B.1: The working flow of 3D spatial data acquisition in S^2DAS, consisting the online part with LiDAR SLAM and IMU DR and the offline part with motion-rectified LiDAR SLAM with loop closure check, vSLAM, and IMU DR.



(a) (b) (c)

FIGURE B.2: The initial prototype design of S^2DAS. The components labeled in the diagram are (1) 3D SLAM laser scanner, (2) adjustable scanner brace, (3) GPS antenna, (4) industrial PC, (5)optical cameras, (6) Power/Input/Output (PIO) box with synchronizer, IMU, 3D compass, GPS module, and connectors installed, (7) hardware frame, (8) battery, and (9) 2D point cloud scanner. (a) The right-side view. (b) The 45 °-angle view. (c) The front-side view.



(a) (b) (c)

FIGURE B.3: The finalized prototype hardware design of S^2DAS. Only the key sensors, the two Velodyne VLP-16 multi-line scanners and the FLIR Ladybug 5Plus panoramic camera, are shown. The IMU not shown explicitly is pasted beside the camera system while all other components are placed into a backpack hanging at the bottom of the backpack frame. (a) The right-side view. (b) The 45 °-angle view. (c) The front-side view.

(a)

(b)

(c)

FIGURE B.4: The finalized prototype frame of S^2DAS with the operator carrying it. (a) The front view of S^2DAS and the operator. (b) The rear view of S^2DAS and the operator. (c) The side view of S^2DAS and the operator.

Appendix C

ELS-based Plane-to-plane Point Cloud Alignment Pseudocodes and Testing Results

Algorithm C.2 Overall Alignment and Mapping Procedure

Notation:

i: frame ID; i_{max}: maximum frame ID
i_{KF}: key frame ID; Ar_{KF}: array of key frame ID
j: key frame ID in the key frame array; j_{max}: number of key frames
A_a: alignment between adjacent frames
A_{KF}: alignment between normal frame and the previous key frame
Ar_{Ob}: array of observations; Ar_{Ver}: array of G2O vertices
Ar_{Edge}: array of G2O edges
PO_i: the position and orientation of the i-th frame
Ar_{PO}: array of frame position and orientation; Ar_{pln}: extracted plane features
M_i: motion within frame
PCD_i: the i-th frame of point cloud data
PCD_i^R: the transformed and motion rectified i-th frame of point cloud data
PCD^M: the merged point cloud data
$\boldsymbol{v_{PS_{KF}}}$: the pairing score between the current frame and the previous key frame
$\boldsymbol{v_{PS}}$: the user-defined key frame pairing score vector threshold

Input: PCD_i and $Ar_{\boldsymbol{pln}}$
Output: PCD^M

procedure OVERALL POINT CLOUD ALIGNMENT PROCESS
  $i_{KF} \leftarrow i$
  $i \leftarrow i+1$
  insert i_{KF} into Ar_{KF}
  $j_{max} \leftarrow$ size of Ar_{KF}
  while $i \neq i_{max}$ **do** *Ordinary Observation Establishing*
    $A_a \leftarrow$ alignment from i to $i-1$
    insert A_a into Ar_{Ob}
    $PO_i \leftarrow PO_{i-1}$ and A_a
    insert PO_i into Ar_{PO}
    initial $\widehat{A_{KF}} \leftarrow$ shortest-path from i to $Ar_{KF}[j_{max}]$ with Ar_{Ob}
    $A_{KF} \leftarrow$ alignment from i to i_{KF} with $\widehat{A_{KF}}$
    insert A_{KF} into Ar_{Ob}
    $\boldsymbol{v_{PS_{KF}}} \leftarrow A_{KF}$
    if $\boldsymbol{v_{PS_{KF}}} < \boldsymbol{v_{PS}}$ **then**
      $i_{KF} \leftarrow i$
      $j \leftarrow j_{max}$
      while $j \neq 0$ **do** *Abundant Key Frame Observation Establishing*
        initial $\widehat{A_{KF_j}} \leftarrow$ shortest-path from i_{KF} to $Ar_{KF}[j]$ with Ar_{Ob}
        $A_{KF_j} \leftarrow$ alignment from i_{KF} to $Ar_{KF}[j]$ with $\widehat{A_{KF_j}}$
        insert A_{KF_j} into Ar_{Ob}
        $j \leftarrow j-1$
      insert i_{KF} into Ar_{KF}
      $j_{max} \leftarrow j_{max}+1$
    $i \leftarrow i+1$
  $Ar_{Ver} \leftarrow Ar_{PO}$
  $Ar_{Edge} \leftarrow Ar_{Ob}$
  Adjust positions and orientations $Ar_{PO} \leftarrow g2o$ adjustment
  for $i = 1$ to $i_{max}-1$ **do** *Motion Correction and Registration*
    $M_{i-1} \leftarrow Ar_{PO}[i]$ and $Ar_{PO}[i-1]$
    $PCD_{i-1}^R \leftarrow M_{i-1}$, $Ar_{PO}[i-1]$ and PCD_i
  $PCD^M \leftarrow PCD_i (i = 1, 2, \ldots, i_{max}-1)$
  return PCD^M

TABLE C.1: Distance Comparison in Both Point Clouds (the Lecture Hall dataset, not aligned) (*)

	T1	T2	T3	T4	T5	T6	T7	T8	T9
T1	N/A	8.754	8.667	8.388	14.008	15.448	20.811	23.969	24.581
		8.863	8.658	8.374	14.173	15.474	20.862	N/A	24.563
		1.24%	-0.11%	-0.16%	1.18%	0.17%	0.24%	N/A	0.07%
T2	-	N/A	16.788	14.893	11.157	16.842	25.222	21.768	22.977
			16.885	14.958	11.245	16.885	25.320	N/A	22.944
			0.58%	0.43%	0.78%	0.26%	0.39%	N/A	-0.14%
T3	-	-	N/A	4.479	17.396	14.442	15.222	24.746	24.767
				4.501	17.580	14.491	15.282	N/A	24.757
				0.50%	1.06%	0.34%	0.40%	N/A	-0.04%
T4	-	-	-	N/A	13.345	10.148	12.658	20.441	20.435
					13.486	10.164	12.720	N/A	20.403
					1.06%	0.16%	0.49%	N/A	-0.16%
T5	-	-	-	-	N/A	7.898	17.820	10.744	11.828
						7.964	17.938	N/A	11.712
						0.83%	0.66%	N/A	-0.98%
T6	-	-	-	-	-	N/A	9.959	10.488	10.330
							10.002	N/A	10.273
							0.43%	N/A	-0.55%
T7	-	-	-	-	-	-	N/A	17.487	16.380
								N/A	16.364
								N/A	-0.10%
T8	-	-	-	-	-	-	-	N/A	2.175
									N/A
									N/A
T9	-	-	-	-	-	-	-	-	N/A

(*) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.2: Distance Comparison in Both Point Clouds (the Lecture Hall dataset, aligned) (*)

		T1	T2	T3	T4	T5	T6	T7	T8	T9
	Δx		-9.5	-0.7	-3.9	-19.2	-6.1	-5.0		-1.1
T1	Δy	N/A	4.3	0.5	1.5	0.8	1.0	-4.6	N/A	3.8
	Δz		-3.8	7.0	4.4	15.4	13.3	-1.5		-4.4
	Δx			8.8	5.6	-9.7	3.4	4.5		8.4
T2	Δy	-	N/A	-3.8	-2.8	-3.5	-3.3	-8.9	N/A	-0.5
	Δz			10.8	8.2	19.2	17.1	2.3		-0.6
	Δx				-3.2	-18.5	-5.4	-4.3		-0.4
T3	Δy	-	-	N/A	1.0	0.3	0.5	-5.1	N/A	3.3
	Δz				-2.6	8.4	6.3	-8.5		-11.4
	Δx					-15.3	-2.2	-1.1		2.8
T4	Δy	-	-	-	N/A	-0.7	-0.5	-6.1	N/A	2.3
	Δz					11.0	8.9	-5.9		-8.8
	Δx						13.1	14.2		18.1
T5	Δy	-	-	-	-	N/A	0.2	-5.4	N/A	3.0
	Δz						-2.1	-16.9		-19.8
	Δx							1.1		5.0
T6	Δy	-	-	-	-	-	N/A	-5.6	N/A	2.8
	Δz							-14.8		-17.7
	Δx									3.9
T7	Δy	-	-	-	-	-	-	N/A	N/A	8.4
	Δz									-2.9
T8		-	-	-	-	-	-	-	N/A	N/A
T9		-	-	-	-	-	-	-	-	N/A

(*) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.3: Distance Comparison in Both Point Clouds (the Staircase dataset) (*)

	T1	T2	T3	T4	T5	T6	T7	T8	T9
T1	N/A	4.649	3.842	4.944	7.586	6.918	7.108	6.374	8.558
		4.661	3.893	5.071	7.642	6.940	7.128	6.464	8.591
		0.26%	1.33%	2.57%	0.75%	0.32%	0.29%	1.41%	0.38%
T2	-	N/A	1.860	4.612	4.012	4.911	2.788	5.614	4.309
			1.871	4.593	3.947	4.893	2.808	5.649	4.251
			0.62%	-0.39%	-1.63%	-0.38%	0.69%	0.62%	-1.32%
T3	-	-	N/A	2.968	3.894	6.418	4.498	4.341	4.792
				2.964	3.947	6.416	4.513	4.343	4.804
				-0.13%	0.01%	-0.03%	0.33%	0.06%	0.24%
T4	-	-	-	N/A	4.194	9.329	7.153	4.850	5.811
					4.222	9.327	7.130	4.863	5.856
					0.68%	0.02%	-0.33%	0.27%	0.77%
T5	-	-	-	-	N/A	8.520	5.002	5.999	1.779
						8.360	4.866	6.039	1.799
						-1.88%	-2.73%	0.67%	1.15%
T6	-	-	-	-	-	N/A	3.896	8.989	8.037
							3.830	8.989	7.808
							-1.70%	0.00%	-2.84%
T7	-	-	-	-	-	-	N/A	7.153	4.200
								7.177	4.019
								0.33%	-4.32%
T8	-	-	-	-	-	-	-	N/A	6.613
									6.666
									0.79%
T9	-	-	-	-	-	-	-	-	N/A

(*) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.4: Distance Comparison in Both Point Clouds (the Staircase dataset, aligned) (*)

		T1	T2	T3	T4	T5	T6	T7	T8	T9
	Δx		0.0	-4.8	-3.8	-1.1	-2.5	0.0	-5.2	-0.7
T1	Δy	N/A	-4.3	-3.6	0.6	7.4	6.3	-2.3	-1.6	9.2
	Δz		8.8	8.2	12.5	2.2	18.8	12.0	8.8	-3.8
	Δx			-4.8	-3.8	-1.1	-2.5	0.0	-5.2	-0.7
T2	Δy	-	N/A	0.7	4.9	11.7	10.6	2.0	2.7	13.5
	Δz			-0.6	3.7	-6.6	10.0	3.2	0.0	12.6
	Δx				1.0	3.7	2.3	4.8	-0.4	4.1
T3	Δy	-	-	N/A	4.2	11.0	9.9	1.3	2.0	12.8
	Δz				4.3	-6.0	10.6	3.8	-1.4	3.1
	Δx					2.7	1.3	3.8	-1.4	3.1
T4	Δy	-	-	-	N/A	6.8	5.7	-2.9	-2.2	8.6
	Δz					-10.3	6.3	-0.5	-3.7	-16.3
	Δx						-1.4	1.1	-4.1	0.4
T5	Δy	-	-	-	-	N/A	-1.1	-9.7	-9.0	1.8
	Δz						16.6	9.8	6.6	-6.0
	Δx							2.5	-2.7	1.8
T6	Δy	-	-	-	-	-	N/A	-8.6	-7.9	2.9
	Δz							-6.8	-10.0	-22.6
	Δx								-5.2	-0.7
T7	Δy	-	-	-	-	-	-	N/A	0.7	11.5
	Δz								-3.2	-15.8
	Δx									4.5
T8	Δy	-	-	-	-	-	-	-	N/A	10.8
	Δz									-12.6
T9		-	-	-	-	-	-	-	-	N/A

(*) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.5: Distance Comparison in Both Point Clouds (the Departing Corridor dataset) (*)

	T1	T2	T3	T4	T5	T6
T1	N/A	9.245	15.906	31.628	31.329	40.216
		9.284	15.844	31.518	31.267	40.169
		0.42%	-0.39%	-0.35%	-0.20%	-0.12%
T2	-	N/A	11.864	29.778	31.307	39.738
			11.901	29.788	31.372	39.797
			0.31%	0.03%	0.21%	0.15%
T3	-	-	N/A	18.061	20.178	28.232
				18.049	20.216	28.271
				-0.07%	0.19%	0.14%
T4	-	-	-	N/A	6.207	10.727
					6.191	10.731
					-0.26%	0.04%
T5	-	-	-	-	N/A	8.907
						8.932
						0.27%
T6	-	-	-	-	-	N/A

(*) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.6: Distance Comparison in Both Point Clouds (the Departing Corridor dataset, aligned) (*)

		T1	T2	T3	T4	T5	T6
	Δx		3	-0.8	-6.2	-3.4	3.9
T1	Δy	N/A	11.9	8.9	9.2	5.5	6
	Δz		-14.8	2.7	7.2	3.7	-18.5
	Δx			-3.8	-9.2	-6.4	0.9
T2	Δy	-	N/A	-3	-2.7	-6.4	-5.9
	Δz			17.5	22.0	18.5	-3.7
	Δx				-5.4	-2.6	4.7
T3	Δy	-	-	N/A	0.3	-3.4	-2.9
	Δz				4.5	1	-21.2
	Δx					2.8	10.1
T4	Δy	-	-	-	N/A	-3.7	-3.2
	Δz					-3.5	-25.7
	Δx						7.3
T5	Δy	-	-	-	-	N/A	0.5
	Δz						-22.2
	Δx						
T6	Δy	-	-	-	-	-	N/A
	Δz						

(∗) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.7: Distance Comparison in Both Point Clouds (the Returning Corridor dataset) (*)

	T1	T2	T3	T4	T5	T6
T1	N/A	9.245	15.906	31.628	31.329	40.216
		9.294	15.864	31.620	31.237	40.148
		0.53%	-0.27%	-0.03%	-0.29%	-0.17%
T2	-	N/A	11.864	29.778	31.307	39.738
			11.838	29.820	31.265	39.756
			-0.22%	0.14%	-0.13%	0.05%
T3	-	-	N/A	18.061	20.178	28.232
				18.119	20.142	28.259
				0.32%	-0.18%	0.10%
T4	-	-	-	N/A	6.207	10.727
					6.174	10.768
					-0.54%	0.38%
T5	-	-	-	-	N/A	8.907
						8.923
						0.18%
T6	-	-	-	-	-	N/A

(*) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.8: Distance Comparison in Both Point Clouds (the Returning Corridor dataset, aligned) (*)

		T1	T2	T3	T4	T5	T6
	Δx		3.2	2.0	7	8.3	7.2
T1	Δy	N/A	7.5	8.6	4.3	11.7	5.1
	Δz		14.3	0.2	11.4	8.9	23.6
	Δx			1.2	3.8	5.1	10.4
T2	Δy	-	N/A	1.1	3.2	4.2	2.4
	Δz			14.5	25.7	23.2	9.3
	Δx				5.0	6.3	9.2
T3	Δy	-	-	N/A	4.3	3.1	3.5
	Δz				11.2	8.7	23.8
	Δx					1.3	14.2
T4	Δy	-	-	-	N/A	7.4	0.8
	Δz					2.5	35.0
	Δx						15.5
T5	Δy	-	-	-	-	N/A	6.6
	Δz						32.5
	Δx						
T6	Δy	-	-	-	-	-	N/A
	Δz						

(∗) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.9: Distance Comparison in Both Point Clouds (the Round-trip Corridor dataset) (*)

	T7	T8	T9
		19.145	23.094
T7	N/A	19.189	23.164
		0.23%	0.31%
			6.902
T8	-	N/A	6.894
			-0.12%
T9	-	-	N/A

(∗) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.10: Distance Comparison in Both Point Clouds (the Round-trip Corridor dataset, aligned) (*)

		T7	T8	T9
	Δx		-2.6	-7.0
T7	Δy	N/A	-3.1	-2.3
	Δz		-0.1	-0.0
	Δx			-4.4
T8	Δy	-	N/A	0.8
	Δz			0.1
T9		-	-	N/A

(∗) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.11: Distance Comparison in Both Point Clouds (the Outdoor Terrace dataset) (*)

	T1	T2	T3	T4	T5	T6	T7	T8	T9
T1	N/A	10.759	18.783	30.810	23.949	19.899	16.365	12.228	6.340
		10.742	18.852	30.891	23.999	19.942	16.335	12.219	6.328
		-0.15%	0.37%	0.26%	0.21%	0.21%	-0.18%	-0.07%	-0.19%
T2	-	N/A	12.569	24.045	19.582	16.174	15.293	10.720	11.871
			12.579	24.123	19.600	16.152	15.228	10.647	11.824
			0.08%	0.32%	0.09%	-0.14%	-0.43%	-0.68%	-0.39%
T3	-	-	N/A	12.161	7.355	5.053	8.062	7.754	14.904
				12.579	7.380	5.050	8.117	7.806	14.982
				0.08%	0.35%	-0.05%	0.68%	0.68%	0.52%
T4	-	-	-	N/A	8.571	11.999	16.816	19.050	26.434
					8.568	11.988	16.880	19.115	26.505
					-0.03%	-0.09%	0.38%	0.34%	0.27%
T5	-	-	-	-	N/A	4.079	8.521	11.730	18.839
						4.082	8.585	11.787	18.896
						0.08%	0.75%	0.49%	0.31%
T6	-	-	-	-	-	N/A	4.880	7.681	14.794
							4.953	7.730	14.855
							1.50%	0.63%	0.41%
T7	-	-	-	-	-	-	N/A	4.779	10.716
								4.778	10.703
								0.02%	0.12%
T8	-	-	-	-	-	-	-	N/A	7.451
									7.459
									0.11%
T9	-	-	-	-	-	-	-	-	N/A

(∗) Note: the first line in each cell is the distance in the TLS point cloud, while the second line is the distance in the S^2DAS point cloud and the third line is the difference in percentage. The unit of distance is meter.

TABLE C.12: Distance Comparison in Both Point Clouds (the Outdoor Terrace dataset, aligned) (*)

		T1	T2	T3	T4	T5	T6	T7	T8	T9
	Δx		-0.4	7.9	7.4	5.7	6.9	-0.7	1.4	0.5
T1	Δy	N/A	-2.3	5.2	-4.4	0.7	5.4	4.4	5.4	4.4
	Δz		1.0	-3.2	5.0	-0.9	-2.0	-0.6	4.0	3.0
	Δx			8.3	7.8	6.1	7.3	-0.3	1.8	0.9
T2	Δy	-	N/A	7.5	-2.1	3.0	7.7	6.7	7.7	4.9
	Δz			-4.2	4.0	-1.9	-3.0	-1.6	3.0	2.0
	Δx				-0.5	-2.2	-1.0	-8.6	-6.5	-7.4
T3	Δy	-	-	N/A	-9.6	-4.5	0.2	-0.8	0.2	-2.6
	Δz				8.2	2.3	1.2	2.6	7.2	6.2
	Δx					-1.7	-0.5	-8.1	-6.0	-6.9
T4	Δy	-	-	-	N/A	5.1	9.8	8.8	9.8	7
	Δz					-5.9	-7.0	-5.6	-1.0	-2.0
	Δx						1.2	-6.4	-4.3	-5.2
T5	Δy	-	-	-	-	N/A	4.7	3.7	4.7	1.9
	Δz						-1.1	0.3	4.9	3.9
	Δx							-7.6	-5.5	-6.4
T6	Δy	-	-	-	-	-	N/A	-1	.00	-2.8
	Δz							1.4	6.0	5.0
	Δx								2.1	1.2
T7	Δy	-	-	-	-	-	-	N/A	1.0	-1.8
	Δz								4.6	3.6
	Δx									-0.9
T8	Δy	-	-	-	-	-	-	-	N/A	-2.8
	Δz									-1.0
T9		-	-	-	-	-	-	-	-	N/A

(*) Note: the three lines in each cell are the distance difference along the three axes, x-axis, y-axis, and z-axis, of the TLS point cloud. The unit of distance is centimeter.

TABLE C.13: Angle Comparison in Both Point Clouds (the Lecture Hall dataset, Part 1)

	T1	T2	T3	T4	T5	T6	T7
T2,T3	2.600, 0.269, 0.272 2.601, 0.267, 0.273 0.03%, -0.83%, 0.54%	-	-	-	-	-	-
T2,T4	2.105, 0.506, 0.530 2.101, 0.504, 0.537 -0.21%, -0.37%, 1.17%	-	-	-	-	-	-
T2,T5	0.921, 1.546, 0.675 0.916, 1.550, 0.675 -0.55%, 0.29%, 0.08%	-	-	-	-	-	-
T2,T6	1.454, 1.146, 0.542 1.451, 1.143, 0.548 -0.20%, -0.24%, 1.05%	-	-	-	-	-	-
T2,T7	1.925, 0.885, 0.332 1.922, 0.884, 0.335 -0.14%, -0.07%, 0.98%	-	-	-	-	-	-
T2,T8	1.135, 1.634, 0.373 N/A, N/A, N/A N/A, N/A, N/A	-	-	-	-	-	-
T2,T9	1.208, 1.570, 0.364 1.206, 1.567, 0.369 -0.16%, -0.20%, 1.37%	-	-	-	-	-	-
T3,T4	0.530, 1.245, 1.366 0.534, 1.243, 1.365 0.65%, -0.18%, -0.09%	0.257, 1.009, 1.875 0.257, 1.004, 1.881 -0.27%, -0.52%, 0.32%	-	-	-	-	-
T3,T5	1.700, 0.925, 0.517 1.707, 0.925, 0.510 0.39%, 0.05%, -1.36%	1.290, 0.664, 1.187 1.298, 0.664, 1.181 0.56%, -0.08%, -0.56%	-	-	-	-	-
T3,T6	1.167, 1.390, 0.585 1.171, 1.388, 0.583 0.29%, -0.10%, -0.34%	0.888, 1.130, 1.124 0.887, 1.127, 1.127 -0.08%, -0.27%, 0.33%	-	-	-	-	-
T3,T7	0.697, 2.070, 0.374 0.699, 2.070, 0.373 0.17%, 0.01%, -0.38%	0.626, 1.813, 0.703 0.627, 1.810, 0.705 0.09%, -0.15%, 0.31%	-	-	-	-	-
T3,T8	1.481, 1.304, 0.356 N/A, N/A, N/A N/A, N/A, N/A	1.373, 1.040, 0.728 N/A, N/A, N/A N/A, N/A, N/A	-	-	-	-	-
T3,T9	1.415, 1.373, 0.353 1.416, 1.372, 0.353 0.07%, -0.06%, -0.06%	1.313, 1.113, 0.715 1.312, 1.110, 0.720 -0.14%, -0.28%, 0.69%	-	-	-	-	-

(*) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.14: Angle Comparison in Both Point Clouds (the Lecture Hall dataset, Part 2)

	T1	T2	T3	T4	T5	T6	T7
	1.185, 1.335, 0.622	1.040, 0.805, 1.296	0.382, 2.634, 0.125				
T4,T5	1.186, 1.343, 0.613	1.047, 0.807, 1.288	0.372, 2.648, 0.122	-	-	-	-
	0.09%, 0.54%, -1.33%	0.64%, 0.16%, -0.61%	-2.66%, 0.53%, -3.08%				
	0.652, 1.965, 0.525	0.640, 1.434, 1.068	0.240, 2.796, 0.105				
T4,T6	0.650, 1.969, 0.522	0.639, 1.432, 1.070	0.233, 2.807, 0.102	-	-	-	-
	-0.22%, 0.22%, -0.56%	-0.14%, -0.13%, 0.26%	-3.11%, 0.37%, -2.73%				
	0.184, 2.837, 0.121	0.380, 2.310, 0.451	0.835, 2.041, 0.265				
T4,T7	0.183, 2.838, 0.120	0.381, 2.307, 0.453	0.838, 2.037, 0.266	-	-	-	-
	-0.19%, 0.05%, -0.84%	0.42%, -0.14%, 0.37%	0.42%, -0.21%, 0.33%				
	0.971, 1.825, 0.346	1.128, 1.295, 0.719	0.254, 2.833, 0.055				
T4,T8	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	-	-	-	-
	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A				
	0.898, 1.917, 0.327	1.064, 1.387, 0.691	0.233, 2.858, 0.051				
T4,T9	0.896, 1.920, 0.326	1.063, 1.384, 0.695	0.233, 2.858, 0.051	-	-	-	-
	-0.28%, 0.17%, -0.21%	-0.12%, -0.22%, 0.63%	-0.18%, 0.01%, 0.48%				
	0.534, 1.481, 1.127	0.403, 2.152, 0.587	0.466, 0.965, 1.710	0.631, 0.860, 1.650			
T5,T6	0.537, 1.460, 1.144	0.411, 2.131, 0.599	0.464, 0.952, 1.726	0.629, 0.849, 1.664	-	-	-
	0.51%, -1.36%, 1.54%	2.03%, -0.96%, 2.13%	-0.50%, -1.40%, 0.93%	-0.40%, -1.36%, 0.86%			
	1.007, 1.409, 0.726	0.664, 2.081, 0.396	1.149, 0.894, 1.099	1.509, 0.788, 0.844			
T5,T7	1.010, 1.398, 0.733	0.672, 2.069, 0.401	1.147, 0.889, 1.105	1.507, 0.786, 0.848	-	-	-
	0.38%, -0.75%, 0.93%	1.07%, -0.55%, 1.09%	-0.16%, -0.48%, 0.56%	-0.16%, -0.24%, 0.51%			
	0.220, 2.633, 0.289	0.108, 2.922, 0.112	0.380, 2.117, 0.644	0.493, 2.019, 0.629			
T5,T8	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	-	-	-
	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A			
	0.287, 2.513, 0.341	0.027, 3.088, 0.026	0.449, 1.999, 0.693	0.581, 1.892, 0.668			
T5,T9	0.291, 2.496, 0.354	0.035, 3.073, 0.033	0.447, 1.987, 0.707	0.578, 1.884, 0.680	-	-	-
	1.43%, -0.67%, 3.76%	27.24%, -0.49%, 29.50%	-0.45%, -0.58%, 1.96%	-0.63%, -0.43%, 1.76%			
	0.472, 1.886, 0.784	0.262, 2.427, 0.453	0.683, 1.303, 1.156	0.878, 1.362, 0.902	0.072, 3.012, 0.057		
T6,T7	0.473, 1.885, 0.783	0.261, 2.431, 0.450	0.683, 1.304, 1.155	0.878, 1.365, 0.898	0.062, 3.029, 0.050	-	-
	0.22%, -0.02%, -0.08%	-0.44%, 0.17%, -0.64%	0.06%, 0.04%, -0.08%	0.01%, 0.24%, -0.37%	-13.81%, 0.59%, -13.45%		
	0.319, 2.342, 0.480	0.488, 1.800, 0.853	0.104, 2.895, 0.143	0.140, 2.867, 0.135	1.160, 1.220, 0.762		
T6,T8	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	-	-
	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A		
	0.248, 2.517, 0.376	0.426, 1.977, 0.739	0.017, 3.101, 0.024	0.065, 3.013, 0.064	1.034, 1.391, 0.717		
T6,T9	0.246, 2.520, 0.376	0.425, 1.973, 0.744	0.020, 3.093, 0.029	0.058, 3.027, 0.057	1.036, 1.375, 0.730	-	-
	-0.87%, 0.11%, -0.14%	-0.23%, -0.22%, 0.73%	18.89%, -0.26%, 19.95%	-10.73%, 0.44%, -10.09%	0.22%, -1.12%, 1.86%		
	0.791, 1.343, 1.008	0.749, 1.011, 1.381	0.775, 1.712, 0.655	1.017, 1.462, 0.663	1.232, 0.618, 1.291	2.052, 0.561, 0.529	
T7,T8	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	-
	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	
	0.720, 1.428, 0.994	0.687, 1.098, 1.356	0.700, 1.800, 0.642	0.929, 1.544, 0.668	1.106, 0.701, 1.335	1.879, 0.645, 0.618	
T7,T9	0.720, 1.424, 0.998	0.685, 1.091, 1.365	0.700, 1.796, 0.646	0.930, 1.539, 0.673	1.099, 0.691, 1.352	1.878, 0.641, 0.622	-
	-0.09%, -0.24%, 0.41%	-0.35%, -0.60%, 0.66%	0.05%, -0.24%, 0.63%	0.06%, -0.36%, 0.75%	-0.63%, -1.48%, 1.30%	-0.04%, -0.50%, 0.64%	
	0.086, 1.813, 1.243	0.081, 2.119, 0.942	0.088, 1.536, 1.517	0.106, 1.515, 1.520	0.167, 2.007, 0.967	0.209, 1.394, 1.539	0.111, 0.982, 2.049
T8,T9	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A
	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A	N/A, N/A, N/A

(∗) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.15: Angle Comparison in Both Point Clouds (the Staircase dataset, Part 1)

	T1	T2	T3	T4	T5	T6	T7
T2,T3	0.399, 0.932, 1.810 0.403, 0.956, 1.783 1.06%, 2.52%, -1.53%	-	-	-	-	-	-
T2,T4	1.001, 1.126, 1.014 0.980, 1.160, 1.002 -2.14%, 2.96%, -1.18%	-	-	-	-	-	-
T2,T5	0.465, 2.131, 0.546 0.437, 2.182, 0.523 -5.96%, 2.36%, -4.14%	-	-	-	-	-	-
T2,T6	0.788, 1.617, 0.736 0.781, 1.626, 0.735 -0.94%, 0.52%, -0.13%	-	-	-	-	-	-
T2,T7	0.229, 2.524, 0.388 0.233, 2.515, 0.393 1.67%, -0.35%, 1.27%	-	-	-	-	-	-
T2,T8	1.026, 1.329, 0.787 1.019, 1.344, 0.779 -0.68%, 1.11%, -0.99%	-	-	-	-	-	-
T2,T9	0.288, 2.542, 0.312 0.257, 2.603, 0.282 -10.83%, 2.38%, -9.41%	-	-	-	-	-	-
T3,T4	0.643, 1.609, 0.889 0.622, 1.647, 0.872 -3.29%, 2.39%, -1.95%	0.382, 2.524, 0.236 0.403, 2.489, 0.250 5.45%, -1.39%, 6.07%	-	-	-	-	-
T3,T5	0.199, 2.746, 0.196 0.194, 2.754, 0.194 -2.63%, 0.29%, -1.33%	1.272, 1.396, 0.474 1.303, 1.357, 0.482 2.42%, -2.76%, 1.64%	-	-	-	-	-
T3,T6	1.156, 1.406, 0.580 1.148, 1.407, 0.587 -0.65%, 0.06%, 1.16%	2.411, 0.536, 0.195 2.418, 0.529, 0.194 0.29%, -1.25%, -0.14%	-	-	-	-	-
T3,T7	0.601, 2.037, 0.504 0.607, 2.021, 0.514 0.94%, -0.77%, 2.01%	2.622, 0.313, 0.207 2.596, 0.329, 0.217 -0.98%, 5.04%, 4.86%	-	-	-	-	-
T3,T8	0.728, 1.783, 0.630 0.713, 1.802, 0.626 -2.14%, 1.08%, -0.59%	0.688, 2.179, 0.275 0.672, 2.199, 0.271 -2.34%, 0.92%, -1.46%	-	-	-	-	-
T3,T9	0.148, 2.874, 0.119 0.174, 2.827, 0.141 17.08%, -1.64%, 18.31%	1.630, 1.114, 0.398 1.665, 1.078, 0.398 2.18%, -3.23%, 0.11%	-	-	-	-	-

(*) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.16: Angle Comparison in Both Point Clouds (the Staircase dataset, Part 2)

	T1	T2	T3	T4	T5	T6	T7
	0.538, 1.954, 0.649	1.007, 0.942, 1.193	1.291, 1.102, 0.748				
T4,T5	0.545, 1.925, 0.672	1.024, 0.925, 1.193	1.302, 1.096, 0.743	-	-	-	-
	1.17%, -1.48%, 3.48%	1.70%, -1.80%, -0.01%	0.87%, -0.61%, -0.60%				
	1.788, 0.810, 0.544	2.737, 0.209, 0.196	2.904, 0.163, 0.075				
T4,T6	1.758, 0.820, 0.564	2.774, 0.190, 0.178	2.912, 0.157, 0.072	-	-	-	-
	-1.66%, 1.24%, 3.59%	1.37%, -9.26%, -9.26%	0.27%, -3.22%, -3.30%				
	1.225, 1.208, 0.708	2.607, 0.200, 0.335	2.548, 0.359, 0.234				
T4,T7	1.208, 1.207, 0.727	2.581, 0.211, 0.349	2.516, 0.380, 0.246	-	-	-	-
	-1.47%, -0.10%, 2.71%	-0.99%, 5.48%, 4.44%	-1.26%, 5.66%, 5.02%				
	0.851, 1.417, 0.873	0.970, 1.270, 0.902	1.410, 1.083, 0.649				
T4,T8	0.838, 1.416, 0.887	0.969, 1.279, 0.893	1.415, 1.081, 0.646	-	-	-	-
	-1.51%, -0.05%, 1.56%	-0.09%, 0.77%, -1.00%	0.34%, -0.22%, -0.37%				
	0.715, 1.836, 0.591	1.418, 0.822, 0.902	1.641, 0.966, 0.535				
T4,T9	0.725, 1.805, 0.611	1.445, 0.804, 0.892	1.656, 0.957, 0.529	-	-	-	-
	1.41%, -1.64%, 3.39%	1.95%, -2.20%, -1.07%	0.92%, -0.94%, -1.11%				
	1.253, 0.881, 1.008	2.535, 0.335, 0.272	1.902, 0.793, 0.447	1.149, 1.527, 0.466			
T5,T6	1.218, 0.893, 1.031	2.476, 0.370, 0.296	1.845, 0.831, 0.465	1.111, 1.560, 0.470	-	-	-
	-2.81%, 1.38%, 2.28%	-2.33%, 10.42%, 8.83%	-2.96%, 4.83%, 4.03%	-3.30%, 2.22%, 0.85%			
	0.692, 1.135, 1.315	1.622, 0.590, 0.929	1.270, 1.033, 0.838	0.754, 1.776, 0.611			
T5,T7	0.668, 1.137, 1.337	1.580, 0.615, 0.946	1.227, 1.062, 0.853	0.727, 1.800, 0.615	-	-	-
	-3.46%, 0.17%, 1.67%	-2.57%, 4.18%, 1.83%	-3.42%, 2.75%, 1.79%	-3.58%, 1.33%, 0.56%			
	0.872, 0.951, 1.319	1.310, 1.129, 0.703	1.630, 0.807, 0.705	1.444, 0.931, 0.766			
T5,T8	0.870, 0.958, 1.314	1.321, 1.135, 0.686	1.643, 0.800, 0.699	1.449, 0.926, 0.767	-	-	-
	-0.28%, 0.78%, -0.38%	0.87%, 0.47%, -2.39%	0.83%, -0.90%, -0.90%	0.29%, -0.51%, 0.07%			
	0.185, 2.054, 0.902	0.425, 1.522, 1.195	0.357, 1.912, 0.872	0.150, 2.631, 0.361			
T5,T9	0.189, 2.029, 0.923	0.436, 1.519, 1.187	0.361, 1.911, 0.870	0.152, 2.627, 0.363	-	-	-
	2.08%, -1.21%, 2.34%	2.66%, -0.21%, -0.68%	1.00%, -0.08%, -0.24%	1.16%, -0.16%, 0.70%			
	0.562, 1.336, 1.243	0.914, 0.603, 1.625	0.642, 0.763, 1.736	0.398, 0.793, 1.951	0.257, 0.333, 2.552		
T6,T7	0.551, 1.343, 1.248	0.896, 0.610, 1.636	0.628, 0.764, 1.749	0.387, 0.779, 1.975	0.246, 0.315, 2.580	-	-
	-2.05%, 0.46%, 0.43%	-1.95%, 1.16%, 0.67%	-2.19%, 0.17%, 0.74%	-2.81%, -1.67%, 1.25%	-4.16%, -5.26%, 1.10%		
	1.484, 0.784, 0.874	2.045, 0.589, 0.508	1.953, 0.465, 0.724	1.236, 0.535, 1.371	1.296, 0.697, 1.149		
T6,T8	1.469, 0.797, 0.876	2.039, 0.595, 0.507	1.953, 0.465, 0.724	1.236, 0.536, 1.370	1.315, 0.708, 1.119	-	-
	-1.00%, 1.62%, 0.25%	-0.27%, 1.03%, -0.11%	-0.01%, 0.07%, -0.03%	-0.02%, 0.29%, -0.10%	1.49%, 1.45%, -2.56%		
	1.073, 1.211, 0.858	2.115, 0.477, 0.550	1.578, 0.639, 0.925	1.026, 0.667, 1.449	1.193, 0.207, 1.741		
T6,T9	1.034, 1.239, 0.869	2.044, 0.506, 0.592	1.518, 0.662, 0.962	0.986, 0.675, 1.480	1.155, 0.212, 1.774	-	-
	-3.68%, 2.34%, 1.31%	-3.35%, 6.18%, 7.53%	-3.82%, 3.56%, 4.05%	-3.89%, 1.33%, 2.15%	-3.24%, 2.48%, 1.93%		
	1.114, 0.927, 1.101	1.960, 0.813, 0.369	1.886, 0.615, 0.641	1.225, 0.692, 1.225	1.406, 0.974, 0.761	0.877, 1.833, 0.432	
T7,T8	1.109, 0.938, 1.095	1.949, 0.820, 0.372	1.890, 0.612, 0.640	1.233, 0.694, 1.215	1.423, 0.983, 0.735	0.878, 1.840, 0.423	-
	-0.46%, 1.18%, -0.53%	-0.55%, 0.95%, 0.84%	0.20%, -0.45%, -0.15%	0.67%, 0.27%, -0.83%	1.22%, 0.93%, -3.45%	0.20%, 0.38%, -2.01%	
	0.511, 1.656, 0.975	1.201, 1.274, 0.667	0.936, 1.165, 1.040	0.628, 0.948, 1.566	0.937, 0.348, 1.857	0.126, 2.899, 0.117	
T7,T9	0.483, 1.691, 0.968	1.147, 1.304, 0.691	0.890, 1.191, 1.060	0.599, 0.964, 1.579	0.909, 0.361, 1.872	0.104, 2.939, 0.099	-
	-5.47%, 2.09%, -0.68%	-4.42%, 2.34%, 3.49%	-4.95%, 2.24%, 1.95%	-4.59%, 1.63%, 0.85%	-2.98%, 3.67%, 0.81%	-17.66%, 1.39%, -15.38%	
	0.873, 1.439, 0.831	1.439, 0.702, 1.000	1.617, 0.809, 0.715	1.328, 1.022, 0.792	1.787, 0.266, 1.089	0.790, 1.042, 1.309	1.140, 0.615, 1.386
T8,T9	0.875, 1.426, 0.840	1.455, 0.686, 1.001	1.631, 0.803, 0.708	1.331, 1.023, 0.788	1.790, 0.267, 1.085	0.805, 1.005, 1.332	1.156, 0.585, 1.401
	0.33%, -0.85%, 1.13%	1.09%, -2.29%, 0.04%	0.82%, -0.80%, -0.96%	0.26%, 0.09%, -0.56%	0.14%, 0.31%, -0.30%	1.80%, -3.60%, 1.78%	1.41%, -4.99%, 1.06%

(*) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.17: Angle Comparison in Both Point Clouds (the Corridor dataset, Departing Trip)

	T1	T2	T3	T4
T2,T3	0.834, 1.693, 0.615 0.844, 1.676, 0.622 1.20%, -1.01%, 1.16%	-	-	-
T2,T4	1.223, 1.622, 0.296 1.236, 1.607, 0.299 1.03%, -0.93%, 0.86%	-	-	-
T2,T5	1.420, 1.425, 0.296 1.433, 1.411, 0.297 0.91%, -0.99%, 0.41%	-	-	-
T2,T6	1.404, 1.506, 0.231 1.415, 1.494, 0.233 0.79%, -0.81%, 0.46%	-	-	-
T3,T4	0.399, 2.393, 0.349 0.403, 2.387, 0.352 1.05%, -0.28%, 0.69%	0.123, 2.939, 0.080 0.128, 2.929, 0.084 4.64%, -0.33%, 5.03%	-	-
T3,T5	0.591, 2.095, 0.455 0.596, 2.090, 0.456 0.77%, -0.24%, 0.10%	0.281, 2.697, 0.164 0.282, 2.694, 0.165 0.47%, -0.08%, 0.59%	-	-
T3,T6	0.575, 2.254, 0.312 0.579, 2.251, 0.312 0.63%, -0.17%, 0.03%	0.207, 2.848, 0.086 0.211, 2.842, 0.088 2.10%, -0.22%, 2.25%	-	-
T4,T5	0.197, 1.424, 1.520 0.197, 1.432, 1.512 0.05%, 0.53%, -0.50%	0.197, 1.720, 1.225 0.196, 1.730, 1.215 -0.63%, 0.61%, -0.75%	0.307, 1.761, 1.074 0.305, 1.772, 1.065 -0.67%, 0.60%, -0.79%	-
T4,T6	0.180, 2.403, 0.558 0.179, 2.413, 0.549 -0.96%, 0.42%, -1.49%	0.116, 2.699, 0.327 0.112, 2.712, 0.317 -2.94%, 0.50%, -3.05%	0.151, 2.734, 0.256 0.145, 2.752, 0.245 -4.22%, 0.64%, -4.38%	-
T5,T6	0.017, 3.064, 0.060 0.021, 3.048, 0.073 22.02%, -0.55%, 21.48%	0.082, 2.770, 0.290 0.084, 2.759, 0.299 3.00%, -0.40%, 3.02%	0.560, 0.718, 1.864 0.556, 0.712, 1.874 -0.61%, -0.82%, 0.50%	0.979, 1.545, 0.617 0.983, 1.544, 0.615 0.35%, -0.08%, -0.35%

(∗) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.18: Angle Comparison in Both Point Clouds (the Corridor dataset, Returning Trip)

	T1	T2	T3	T4
T2,T3	0.834, 1.693, 0.615 0.835, 1.685, 0.621 0.18%, -0.46%, 1.02%	-	-	-
T2,T4	1.223, 1.622, 0.296 1.229, 1.615, 0.298 0.45%, -0.45%, 0.60%	-	-	-
T2,T5	1.420, 1.425, 0.296 1.425, 1.419, 0.299 0.30%, -0.46%, 0.75%	-	-	-
T2,T6	1.404, 1.506, 0.231 1.412, 1.496, 0.233 0.62%, -0.67%, 0.64%	-	-	-
T3,T4	0.399, 2.393, 0.349 0.402, 2.390, 0.350 0.80%, -0.16%, 0.17%	0.123, 2.939, 0.080 0.119, 2.946, 0.077 -3.23%, 0.24%, -3.74%	-	-
T3,T5	0.591, 2.095, 0.455 0.593, 2.092, 0.456 0.32%, -0.13%, 0.20%	0.281, 2.697, 0.164 0.277, 2.702, 0.162 -1.32%, 0.22%, -1.35%	-	-
T3,T6	0.575, 2.254, 0.312 0.581, 2.248, 0.313 0.92%, -0.30%, 0.46%	0.207, 2.848, 0.086 0.202, 2.855, 0.084 -2.28%, 0.24%, -2.56%	-	-
T4,T5	0.197, 1.424, 1.520 0.196, 1.411, 1.534 -0.46%, -0.94%, 0.94%	0.197, 1.720, 1.225 0.197, 1.707, 1.237 -0.24%, -0.71%, 1.03%	0.307, 1.761, 1.074 0.307, 1.747, 1.088 -0.11%, -0.78%, 1.32%	-
T4,T6	0.180, 2.403, 0.558 0.185, 2.387, 0.570 2.37%, -0.68%, 2.17%	0.116, 2.699, 0.327 0.121, 2.681, 0.340 4.03%, -0.65%, 3.91%	0.151, 2.734, 0.256 0.160, 2.710, 0.272 6.01%, -0.90%, 6.05%	-
T5,T6	0.017, 3.064, 0.060 0.013, 3.083, 0.045 -24.32%, 0.62%, -24.69%	0.082, 2.770, 0.290 0.078, 2.788, 0.276 -4.48%, 0.64%, -4.90%	0.160, 2.614, 0.368 0.156, 2.629, 0.357 -2.57%, 0.58%, -3.03%	0.979, 1.545, 0.617 0.977, 1.554, 0.611 -0.28%, 0.59%, -1.03%

(∗) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.19: Angle Comparison in Both Point Clouds (the Corridor dataset, Round Trip)

	T6
T7,T8	0.270, 2.039, 0.833 0.268, 2.045, 0.828 -0.78%, 0.33%, -0.54%

(∗) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.20: Angle Comparison in Both Point Clouds (the Outdoor Terrace dataset, Part 1)

	T1	T2	T3	T4	T5	T6	T7
T2,T3	0.694, 1.868, 0.580 0.689, 1.878, 0.574 -0.74%, 0.57%, -0.95%	-	-	-	-	-	-
T2,T4	0.746, 2.087, 0.309 0.745, 2.090, 0.307 -0.10%, 0.12%, -0.57%	-	-	-	-	-	-
T2,T5	0.935, 1.748, 0.458 0.933, 1.753, 0.456 -0.30%, 0.29%, -0.49%	-	-	-	-	-	-
T2,T6	0.947, 1.624, 0.570 0.941, 1.633, 0.568 -0.58%, 0.50%, -0.46%	-	-	-	-	-	-
T2,T7	1.132, 1.320, 0.690 1.128, 1.322, 0.691 -0.32%, 0.21%, 0.12%	-	-	-	-	-	-
T2,T8	0.963, 1.211, 0.968 0.956, 1.216, 0.969 -0.64%, 0.39%, 0.15%	-	-	-	-	-	-
T2,T9	1.460, 0.560, 1.122 1.456, 0.560, 1.125 -0.33%, 0.16%, 0.34%	-	-	-	-	-	-
T3,T4	0.075, 2.951, 0.116 0.076, 2.949, 0.117 0.97%, -0.07%, 1.22%	0.232, 2.669, 0.240 0.223, 2.688, 0.230 -3.99%, 0.71%, -4.04%	-	-	-	-	-
T3,T5	0.247, 2.218, 0.676 0.249, 2.211, 0.682 0.74%, -0.35%, 0.89%	0.141, 2.757, 0.243 0.145, 2.748, 0.249 2.48%, -0.32%, 2.23%	-	-	-	-	-
T3,T6	0.256, 1.664, 1.222 0.255, 1.659, 1.228 -0.22%, -0.30%, 0.45%	0.249, 2.233, 0.660 0.251, 2.223, 0.667 0.85%, -0.43%, 1.13%	-	-	-	-	-
T3,T7	0.442, 1.053, 1.647 0.443, 1.042, 1.656 0.25%, -1.03%, 0.59%	0.554, 1.625, 0.963 0.562, 1.609, 0.971 1.33%, -0.97%, 0.87%	-	-	-	-	-
T3,T8	0.274, 0.441, 2.426 0.272, 0.434, 2.435 -0.75%, -1.57%, 0.37%	0.661, 1.013, 1.468 0.666, 1.002, 1.474 0.77%, -1.09%, 0.41%	-	-	-	-	-
T3,T9	0.771, 0.301, 2.070 0.772, 0.299, 2.071 0.15%, -0.61%, 0.03%	1.311, 0.878, 0.953 1.321, 0.871, 0.950 0.78%, -0.88%, -0.26%	-	-	-	-	-

(*) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

TABLE C.21: Angle Comparison in Both Point Clouds (the Outdoor Terrace dataset, Part 2)

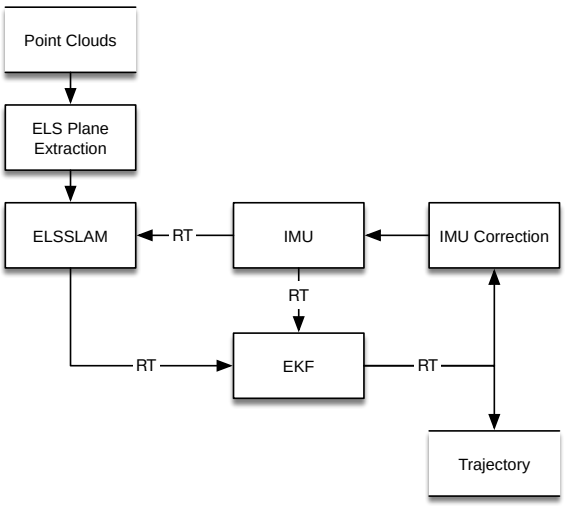
	T1	T2	T3	T4	T5	T6	T7
	0.189, 0.554, 2.398	0.339, 0.862, 1.940	0.769, 0.640, 1.733				
T4,T5	0.187, 0.549, 2.406	0.336, 0.855, 1.950	0.768, 0.641, 1.733	-	-	-	-
	-1.11%, -0.96%, 0.31%	-0.76%, -0.82%, 0.50%	-0.19%, 0.26%, -0.01%				
	0.202, 0.339, 2.600	0.463, 0.647, 2.032	1.329, 0.421, 1.392				
T4,T6	0.197, 0.332, 2.613	0.458, 0.638, 2.046	1.324, 0.421, 1.397	-	-	-	-
	-2.47%, -2.22%, 0.48%	-1.25%, -1.40%, 0.73%	-0.37%, -0.08%, 0.38%				
	0.386, 0.375, 2.381	0.768, 0.683, 1.691	1.935, 0.464, 0.742				
T4,T7	0.383, 0.370, 2.389	0.767, 0.677, 1.698	1.937, 0.466, 0.739	-	-	-	-
	-0.76%, -1.34%, 0.34%	-0.04%, -1.00%, 0.42%	0.09%, 0.25%, -0.40%				
	0.217, 0.138, 2.787	0.876, 0.447, 1.819	2.534, 0.234, 0.373				
T4,T8	0.211, 0.134, 2.796	0.873, 0.441, 1.827	2.535, 0.235, 0.372	-	-	-	-
	-2.50%, -2.88%, 0.34%	-0.26%, -1.29%, 0.44%	0.02%, 0.25%, -0.32%				
	0.732, 0.161, 2.249	1.535, 0.465, 1.141	2.706, 0.240, 0.195				
T4,T9	0.728, 0.160, 2.254	1.537, 0.462, 1.142	2.700, 0.244, 0.198	-	-	-	-
	-0.45%, -0.83%, 0.21%	0.12%, -0.70%, 0.12%	-0.26%, 1.78%, 1.36%				
	0.022, 0.110, 3.010	0.126, 0.522, 2.494	0.560, 0.718, 1.864	0.218, 2.451, 0.473			
T5,T6	0.021, 0.101, 3.020	0.123, 0.506, 2.512	0.556, 0.712, 1.874	0.220, 2.444, 0.477	-	-	-
	-8.25%, -8.16%, 0.36%	-2.51%, -2.97%, 0.75%	-0.61%, -0.82%, 0.50%	0.93%, -0.25%, 0.87%			
	0.196, 0.384, 2.561	0.429, 0.842, 1.871	1.168, 1.056, 0.917	0.179, 2.782, 0.180			
T5,T7	0.196, 0.379, 2.567	0.431, 0.835, 1.876	1.171, 1.057, 0.914	0.179, 2.784, 0.179	-	-	-
	-0.45%, -1.44%, 0.25%	0.51%, -0.92%, 0.30%	0.24%, 0.08%, -0.40%	-0.23%, 0.08%, -1.01%			
	0.027, 0.028, 3.086	0.537, 0.486, 2.118	1.777, 0.704, 0.661	0.416, 2.427, 0.299			
T5,T8	0.024, 0.025, 3.092	0.537, 0.481, 2.124	1.777, 0.705, 0.660	0.414, 2.430, 0.297	-	-	-
	-11.29%, -11.78%, 0.21%	0.05%, -1.21%, 0.26%	-0.04%, 0.25%, -0.14%	-0.31%, 0.16%, -0.83%			
	0.550, 0.177, 2.415	1.198, 0.627, 1.317	1.938, 0.831, 0.373	0.400, 2.563, 0.178			
T5,T9	0.550, 0.176, 2.416	1.202, 0.623, 1.316	1.932, 0.836, 0.374	0.397, 2.568, 0.176	-	-	-
	-0.04%, -0.54%, 0.05%	0.40%, -0.59%, -0.08%	-0.30%, 0.58%, 0.28%	-0.76%, 0.19%, -1.06%			
	0.187, 0.672, 2.283	0.306, 1.238, 1.597	0.611, 1.894, 0.636	0.055, 2.951, 0.136	0.344, 2.511, 0.286		
T6,T7	0.188, 0.666, 2.288	0.312, 1.230, 1.600	0.617, 1.893, 0.631	0.055, 2.954, 0.133	0.351, 2.504, 0.287	-	-
	0.89%, -0.94%, 0.21%	1.69%, -0.69%, 0.22%	0.97%, -0.05%, -0.80%	-0.58%, 0.11%, -2.15%	1.77%, -0.27%, 0.28%		
	0.025, 0.039, 3.077	0.414, 0.596, 2.132	1.224, 1.251, 0.667	0.202, 2.621, 0.319	0.098, 2.992, 0.052		
T6,T8	0.021, 0.033, 3.087	0.417, 0.592, 2.133	1.225, 1.254, 0.662	0.198, 2.634, 0.310	0.089, 3.006, 0.047	-	-
	-14.76%, -15.35%, 0.32%	0.75%, -0.69%, 0.05%	0.13%, 0.25%, -0.72%	-1.95%, 0.48%, -2.72%	-9.23%, 0.47%, -9.71%		
	0.533, 0.220, 2.389	1.072, 0.782, 1.287	1.378, 1.421, 0.342	0.182, 2.812, 0.147	0.115, 2.994, 0.032		
T6,T9	0.535, 0.219, 2.387	1.080, 0.778, 1.283	1.376, 1.426, 0.340	0.177, 2.822, 0.142	0.125, 2.982, 0.034	-	-
	0.44%, -0.20%, -0.08%	0.73%, -0.47%, -0.32%	-0.18%, 0.30%, -0.52%	-2.89%, 0.36%, -3.37%	8.37%, -0.41%, 7.98%		
	0.169, 0.446, 2.527	0.108, 0.245, 2.788	0.613, 1.203, 1.326	0.237, 1.935, 0.970	0.356, 2.115, 0.671	0.643, 1.839, 0.659	
T7,T8	0.172, 0.453, 2.517	0.107, 0.240, 2.795	0.608, 1.205, 1.329	0.236, 1.936, 0.970	0.354, 2.114, 0.673	0.639, 1.836, 0.667	-
	1.57%, 1.61%, -0.39%	-1.48%, -2.17%, 0.25%	-0.72%, 0.14%, 0.21%	-0.41%, 0.05%, 0.00%	-0.48%, -0.03%, 0.36%	-0.65%, -0.18%, 1.13%	
	0.374, 0.218, 2.550	0.773, 0.884, 1.485	0.771, 1.818, 0.552	0.225, 2.560, 0.357	0.229, 2.731, 0.181	0.483, 2.445, 0.213	
T7,T9	0.376, 0.219, 2.547	0.776, 0.884, 1.482	0.763, 1.827, 0.552	0.222, 2.566, 0.354	0.226, 2.735, 0.180	0.478, 2.449, 0.215	-
	0.46%, 0.38%, -0.10%	0.36%, 0.02%, -0.20%	-1.09%, 0.49%, -0.08%	-1.26%, 0.22%, -0.78%	-1.44%, 0.16%, -0.56%	-1.04%, 0.15%, 0.66%	
	0.525, 0.440, 2.177	0.665, 1.385, 1.092	0.195, 2.743, 0.203	0.044, 2.983, 0.114	0.150, 2.753, 0.238	0.209, 2.718, 0.215	0.640, 2.109, 0.393
T8,T9	0.526, 0.440, 2.175	0.669, 1.386, 1.087	0.188, 2.756, 0.197	0.045, 2.981, 0.116	0.151, 2.749, 0.241	0.206, 2.721, 0.214	0.645, 2.102, 0.395
	0.33%, 0.00%, -0.08%	0.62%, 0.08%, -0.47%	-3.42%, 0.46%, -2.87%	1.54%, -0.09%, 1.79%	0.70%, -0.13%, 1.09%	-1.04%, 0.12%, -0.51%	0.75%, -0.33%, 0.54%

(∗) Note: the first line in each cell is the angles in radian in the triangles formed in the TLS point cloud, while the second line is the angles in rad in the triangles formed in the S^2DAS point cloud and the third line is the difference in percentage. The unit of angle is rad.

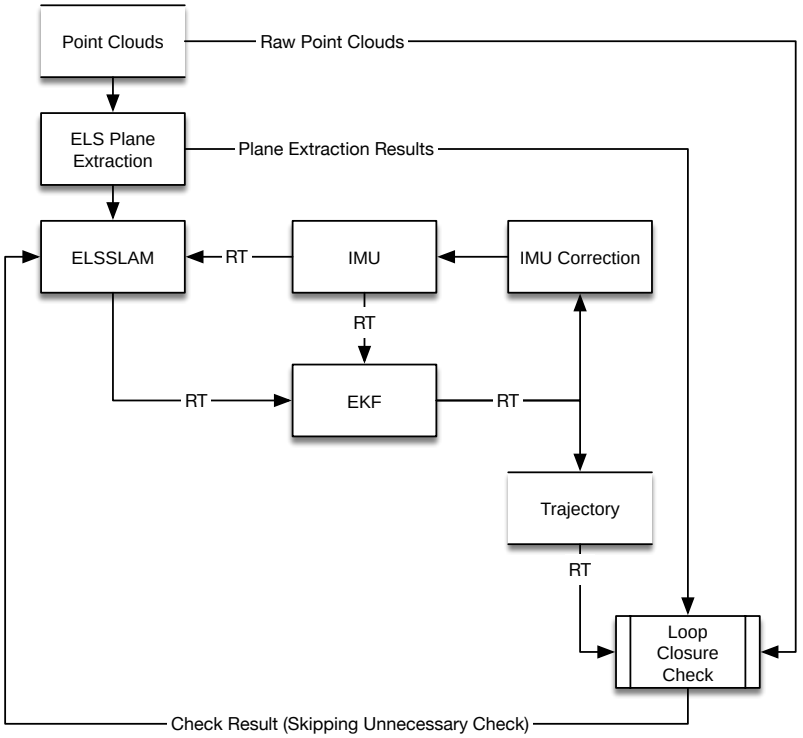
TABLE C.22: Overall Comparison Results on both Distance Differences (DD) and Angular Differences (AD) (*)

Dataset Name	Target Quantity	Average DD in Percentage (%)	Average Absolute DD (cm)	Average Absolute DD in x (cm)	Average Absolute DD in y (cm)	Average Absolute DD in z (cm)	Average AD in Percentage (%)	Average Absolute AD (°)	Average AD vs Edge Length ($°/m$)
Lecture Hall	8	0.48	6.4	6.9	3.0	9.1	1.40	0.280	0.022
Staircase	9	0.90	4.7	2.5	5.9	8.2	2.17	0.928	0.197
Terrace	9	0.33	4.0	4.3	4.2	3.3	1.09	0.257	0.025
Corridor (Departing)	6	0.21	4.2	4.7	4.9	12.4	1.80	0.382	0.025
Corridor (Returning)	6	0.23	4.0	6.6	4.9	16.3	2.11	0.431	0.029
Corridor (Round)	3	0.22	1.0	4.7	2.1	0.1	0.55	0.254	0.020
Average	N/A	0.44	4.4	4.8	4.4	8.5	1.60	0.452	0.062

(*) Note: the prefix of *absolute* in this table means only absolute values were considered in the calculation of the average value.



(a)



(b)

FIGURE C.1: The design plan of integrating IMU with the proposed point cloud alignment method. (a) The online process of IMU integration that the IMU would be used for providing initial position and pose estimations for alignment while the final position and pose changes produced by EKF would be used to correct the accumulative errors of IMU. (b) In addition to the online integration, a loop closure detection procedure based on the adjusted trajectory was implemented in the offline process, in order to replace the tedious frame-to-frame alignment between any two of the key frames by the trajectory-based loop detection.

References

Achtelik, M., Bachrach, A., He, R., Prentice, S., & Roy, N. (2009, April). Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In *Proceedings volume 7332, unmanned systems technology xi* (Vol. 733219). Orlando, USA

APOGEO. (2015). Boom technologiczny HI-TARGET na INTERGEO 2015. Retrieved April 12, 2019, from http://www.apogeo.pl/component/content/article/15-rone/1404-boom-technologiczny-hi-target-na-intergeo-2015

Applanix Corp. (2015). Land solutions: TIMMS indoor mapping. Retrieved December 8, 2015, from http://www.applanix.com/solutions/land/timms.html

Besl, P. J. & McKay, N. D. (1992, April). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(2), 239–256

Biber, P., Andreasson, H., Duckett, T., & Schilling, A. (2004, September). 3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. In *Ieee/rsj international conference on intelligent robots and systems (iros) (ieee cat. no. 04ch37566)* (Vol. 4, pp. 3430–3435). Sendai, Japan

Biber, P., Fleck, S., Wand, M., Staneker, D., & Straßer, W. (2005, August). First experiences with a mobile platform for flexible 3D model acquisition in indoor and outdoor enviroments - The Wagele. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(5/W17).

Biber, P. & Strasser, W. (2003, October). The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 ieee/rsj international conference on intelligent robots and systems (iros 2003) (cat. no.03ch37453)*. Las Vegas, USA

Borrmann, D., Elseberg, J., Lingemann, K., & Nüchter, A. (2011, June). The 3D Hough Transform for plane detection in point clouds: a review and a new accumulator design. *3D Research, 2*(2), 1–13

Bosse, M., Zlot, R., & Flick, P. (2012, October). Zebedee: Design of a Spring-Mounted 3-D Range Sensor with Application to Mobile Mapping. *IEEE Transactions on Robotics*, *28*(5), 1104–1119

Boulch, A. & Marlet, R. (2016, August). Deep Learning for Robust Normal Estimation in Unstructured Point Clouds. *Computer Graphics Forum*, *35*(5), 281–290

Brenneke, C., Wulf, O., & Wagner, B. (2003, October). Using 3D laser range data for SLAM in outdoor environments. In *Proceedings 2003 ieee/rsj international conference on intelligent robots and systems (iros 2003) (cat. no.03ch37453)* (Vol. 1, *section 2*, pp. 188–193). Las Vegas, USA

Brilakis, I., Lourakis, M., Sacks, R., Savarese, S., Christodoulou, S., Teizer, J., & Makhmalbaf, A. (2010, November). Toward automated generation of parametric BIMs based on hybrid video and laser scanning data. *Advanced Engineering Informatics*, *24*(4), 456–465

Cabo, C., García Cortés, S., & Ordoñez, C. (2015, October). Mobile Laser Scanner data for automatic surface detection based on line arrangement. *Automation in Construction*, *58*, 28–37

Cabral, R. & Furukawa, Y. (2014, June). Piecewise planar and compact floorplan reconstruction from images. In *27th ieee conference on computer vision and pattern recognition (cvpr)* (pp. 628–635). Columbus, USA

Chow, J. C. (2014). *Multi-sensor integration for indoor 3D reconstruction* (Doctoral dissertation, University of Calgary). Retrieved from https://arxiv.org/abs/1802.07866

Concha, A., Loianno, G., Kumar, V., & Civera, J. (2016, May). Visual-inertial direct SLAM. In *Ieee international conference on robotics and automation (icra)* (pp. 1331–1338). Stockholm, Sweden

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., & Leite, F. (2018, April). 6D DBSCAN-based segmentation of building point clouds for planar object classification. *Automation in Construction*, *88*(May 2017), 44–58

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., & Theobalt, C. (2017, May). BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, *36*(4), 1

Darwish, W., Tang, S., Li, W., & Chen, W. (2017, May). A New Calibration Method for Commercial RGB-D Sensors. *Sensors, 17*(6), 1204

Dimitrievski, M., Van Hamme, D., Veelaert, P., & Philips, W. (2016, February). Robust Matching of Occupancy Maps for Odometry in Autonomous Vehicles. In *Proceedings of the 11th joint conference on computer vision, imaging and computer graphics theory and applications* (Vol. 3, pp. 626–633). Rome, Italy

Diosi, A. & Kleeman, L. (2005, August). Laser scan matching in polar coordinates with application to SLAM. In *2005 ieee/rsj international conference on intelligent robots and systems*. Edmonton, Canada

Dong, Z., Yang, B., Hu, P., & Scherer, S. (2018, March). An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing, 137*, 112–133

Ewertowski, M., Tomczyk, A., Evans, D., Roberts, D., & Ewertowski, W. (2019, January). Operational Framework for Rapid, Very-high Resolution Mapping of Glacial Geomorphology Using Low-cost Unmanned Aerial Vehicles and Structure-from-Motion Approach. *Remote Sensing, 11*(1), 65

Fan, W. (2015). *Plane Modeling for Point Clouds of Indoor Surfaces* (Unpublished master's thesis, The Hong Kong Polytechnic University, Hong Kong S.A.R.).

Fan, Y., Wang, M., Geng, N., He, D., Chang, J., & Zhang, J. J. (2018, May). A self-adaptive segmentation method for a point cloud. *The Visual Computer, 34*(5), 659–673

FARO Technologies Inc. (2019). *User Manual for the Focus M70 and S70/S150/S350*. Lake Mary, USA.

FLIR Integrated Imaging Solutions Inc. (2011). *Stereo Accuracy Chart*. Richmond, Canada.

Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Ieee conference on computer vision and pattern recognition* (pp. 3354–3361). Rhode Island, USA

Geneva, P., Eckenhoff, K., Yang, Y., & Huang, G. (2018, October). LIPS: LiDAR-Inertial 3D Plane SLAM. In *Ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 123–130). Madrid, Spain

Georgy, J., Karamat, T., Iqbal, U., & Noureldin, A. (2011, July). Enhanced MEMS-IMU/odometer/GPS integration using mixture particle filter. *GPS Solutions*, *15*(3), 239–252

GeoSLAM. (2019). GeoSLAM - The Experts in "Go-Anywhere" 3D Mobile Mapping Technology. Retrieved April 11, 2019, from https://geoslam.com/

Glennie, C. L., Kusari, A., & Facchin, A. (2016, March). Calibration and Stability Analysis of the VLP-16 Laser Scanner. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(3W4), 55–60

Google. (2015). Project Tango. Retrieved December 8, 2015, from https://www.google.com/atap/project-tango/index.html

Google. (2016). Introducing Cartographer. Retrieved October 5, 2016, from https://opensource.googleblog.com/2016/10/introducing-cartographer.html

Granger, S. & Pennec, X. (2002). Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. In *Eccv '02: proceedings of the 7th european conference on computer vision-part iv* (pp. 418–432). Berlin Heidelberg, Germany

Grant, W. S., Voorhies, R. C., & Itti, L. (2013, November). Finding planes in LiDAR point clouds for real-time registration. In *Ieee/rsj international conference on intelligent robots and systems*. Tokyo, Japan

Grant, W. S., Voorhies, R. C., & Itti, L. (2019, June). Efficient Velodyne SLAM with point and plane features. *Autonomous Robots*, *43*(5), 1207–1224

GreenValley International. (2019). LiBackpack - Mobile Handheld LiDAR - 3D Mapping System. Retrieved April 11, 2019, from https://greenvalleyintl.com/hardware/libackpack/

Grilli, E., Menna, F., & Remondino, F. (2017, February). A REVIEW OF POINT CLOUDS SEGMENTATION AND CLASSIFICATION ALGORITHMS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W3*(2/W3), 339–344

Haggag, H., Hossny, M., Filippidis, D., Creighton, D., Nahavandi, S., & Puri, V. (2013, December). Measuring depth accuracy in RGBD cameras. In *7th international conference on signal processing and communication systems (icspcs)* (pp. 1–7). Gold Coast, Australia

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012, February). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*. Springer Tracts in Advanced Robotics, *31*(5), 647–663

Higgins, S. (2018). Paracosm's PX-80 handheld lidar is fast and flexible enough to be your standalone scanner - SPAR 3D. Retrieved April 12, 2019, from https://www.spar3d.com/sponsored/paracosms-px-80/

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. *Computer Graphics*, *26*(2), 71–78

Huai, J., Zhang, Y., & Yilmaz, A. (2015, August). Real-time large scale 3D reconstruction by fusing Kinect and IMU data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-3/W5*, 491–496

Intel® Software. (2018). Intel® RealSense™ Technology. Retrieved March 25, 2019, from https://software.intel.com/en-us/realsense

Iwane Laboratories Ltd. (2010). *White Paper: Dual CV*. Sapporo, Japan.

Jennings, C. & Murray, D. (1997, April). Stereo vision based mapping and navigation for mobile robots. In *Proceedings of international conference on robotics and automation* (Vol. 2, pp. 1694–1699). Albuquerque, USA.

Jung, J., Yoon, S., Ju, S., & Heo, J. (2015, October). Development of Kinematic 3D Laser Scanning System for Indoor Mapping and As-Built BIM Using Constrained SLAM. *Sensors, 15*(10), 26430–26456

Kamijo, S., Gu, Y., & Hsu, L.-T. (2015, October). Autonomous Vehicle Technologies : Localization and Mapping. *IEICE ESS Fundamentals Review*, *9*(2), 131–141.

Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., & Bhowmik, A. (2017, July). Intel(R) RealSense(TM) Stereoscopic Depth Cameras. In *Ieee conference on computer vision and pattern recognition workshops (cvprw)* (pp. 1267–1276). Honolulu, USA

Kohlbrecher, S., von Stryk, O., Meyer, J., & Klingauf, U. (2011, November). A flexible and scalable SLAM system with full 3D motion estimation. In *2011 ieee international symposium on safety, security, and rescue robotics* (pp. 155–160). Kyoto, Japan

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011, May). G2o: A general framework for graph optimization. In *Ieee international conference on robotics and automation* (pp. 3607–3613). Shanghai, China

Kusnoto, B. & Evans, C. A. (2002, October). Reliability of a 3D surface laser scanner for orthodontic applications. *American Journal of Orthodontics and Dentofacial Orthopedics*, *122*(4), 342–348

Lardinois, F. (2014, September). Google unveils the cartographer, its indoor mapping backpack. Retrieved June 8, 2015, from http://techcrunch.com/2014/09/04/google-unveils-the-cartographer-its-indoor-mapping-backpack/

Lee, Y.-c. (2015, August). A reliable range-free indoor localization method for mobile robots. In *Ieee international conference on automation science and engineering (case)* (pp. 720–727). Gothenburg, Sweden

Lehtola, V., Kaartinen, H., Nüchter, A., Kaijaluoto, R., Kukko, A., Litkey, P., ... Hyyppä, J. (2017, August). Comparison of the Selected State-Of-The-Art 3D Indoor Scanning and Point Cloud Generation Methods. *Remote Sensing*, *9*(8), 796

Leica Geosystems AG. (2015). Leica pegasus : backpack - award-winning wearable reality capture - indoors, outdoors, anywhere. Retrieved June 8, 2015, from http://www.leica-geosystems.com/en/Leica-PegasusBackpack%7B%5C_%7D106730.htm

Lenac, K., Kitanov, A., Cupec, R., & Petrović, I. (2017, June). Fast planar surface 3D SLAM using LIDAR. *Robotics and Autonomous Systems*, *92*, 197–220

Leonard, J. J. & Durrant-Whyte, H. F. (1991, November). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings iros '91:ieee/rsj international workshop on intelligent robots and systems '91* (pp. 1442–1447). Osaka, Japan

Li, Y. & Olson, E. B. (2010, May). Extracting general-purpose features from LIDAR data. In *2010 ieee international conference on robotics and automation* (pp. 1388–1393). Anchorage, USA

Li, Z. (2006). Algorithms for point-reduction of individual line features. In *Algorithmic foundation of multi-scale spatial representation* (Chap. 5, pp. 112–137). Boca Raton, USA: CRC Press (Taylor & Francis).

Liu, T., Carlberg, M., Chen, G., Chen, J., Kua, J., & Zakhor, A. (2010, September). Indoor localization and visualization using a human-operated backpack system. In *International conference on indoor positioning and indoor navigation*. Zürich, Switzerland

Lu, F. & Milios, E. (1997, March). Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems, 18*(3), 249–275

Lupton, T. & Sukkarieh, S. (2008, May). Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In *Ieee international conference on robotics and automation* (pp. 3698–3703). Pasadena, USA

Magnusson, M. (2009). *The Three-Dimensional Normal-Distributions Transform* (Doctoral dissertation, University of Massachusetts Amherst). Retrieved from http : / / www . diva - portal . org / smash / record . jsf ? pid = diva2 % 7B % 5C % %7D3A276162%7B%5C&%7Ddswid=-5899

Makadia, A., Patterson, A., & Daniilidis, K. (2006, June). Fully Automatic Registration of 3D Point Clouds. In *Ieee computer society conference on computer vision and pattern recognition - volume 1 (cvpr'06)* (Vol. 1, pp. 1297–1304). New York, USA

Mitra, N. J. [Niloy J], Gelfand, N., Pottmann, H., & Guibas, L. (2004, January). Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the 2004 eurographics/acm siggraph symposium on geometry processing - sgp '04* (p. 22). New York, New York, USA

Mitra, N. J. [Niloy J.] & Nguyen, A. (2003, June). Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth conference on computational geometry - scg '03* (Vol. 14, *04n05*, pp. 322–328). San Diego, USA

Miyazaki, R., Yamamoto, M., & Harada, K. (2017, July). Line-Based Planar Structure Extraction from a Point Cloud with an Anisotropic Distribution. *International Journal of Automation Technology*, *11*(4), 657–665

Montemerlo, M. & Thrun, S. (2003, September). Simultaneous localization and mapping with unknown data association using FastSLAM. In *Ieee international conference on robotics and automation (cat. no.03ch37422)* (Vol. 2, pp. 1985–1991). Taipei, Taiwan

NavVis US Inc. (2019). NavVis | M6. Retrieved April 12, 2019, from https://www.navvis.com/m6

Nguyen, H. L., Belton, D., & Helmholz, P. (2019, May). Planar surface detection for sparse and heterogeneous mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *151*, 141–161

Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., & Siegwart, R. (2007, August). A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. *Autonomous Robots*, *23*(2), 97–111

Nocerino, E., Menna, F., Remondino, F., Toschi, I., & Rodríguez-Gonzálvez, P. (2017, June). Investigation of indoor and outdoor performance of two portable mobile mapping systems. In *Proceedings volume 10332, videometrics, range imaging, and applications xiv* (Vol. 10332I). Munich, Germany

Nüchter, A., Borrmann, D., Koch, P., Kühn, M., & May, S. (2015). A man-portable, IMU-free mobile mapping system. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(3/W5), 17–23

Nurunnabi, A., Belton, D., & West, G. (2016, August). Robust Segmentation for Large Volumes of Laser Scanning Three-Dimensional Point Cloud Data. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(8), 4790–4805

Nützi, G., Weiss, S., Scaramuzza, D., & Siegwart, R. (2011, January). Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent & Robotic Systems*, *61*(1-4), 287–299

OpenMP ARB. (2015). OpenMP.org. Retrieved March 12, 2015, from http://openmp.org/wp/

Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., & Poppinga, J. (2010, January). Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics*, *27*(1), 52–84

Pathak, K., Birk, A., Vaškevičius, N., & Poppinga, J. (2010, June). Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping. *IEEE Transactions on Robotics*, *26*(3), 424–441

Pathak, K., Vaskevicius, N., & Birk, A. (2009, May). Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds. In *2009 ieee international conference on robotics and automation* (3, pp. 1631–1636). Kobe, Japan

Paz, L., Pinies, P., Tardos, J., & Neira, J. (2008, October). Large-Scale 6-DOF SLAM With Stereo-in-Hand. *IEEE Transactions on Robotics, 24*(5), 946–957

Pfister, S. T., Roumeliotis, S. I., & Burdick, J. W. (2003, September). Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Ieee international conference on robotics and automation (cat. no.03ch37422)* (Vol. 1, pp. 1304–1311). Taipei, Taiwan

Poppinga, J., Vaskevicius, N., Birk, A., & Pathak, K. (2008, September). Fast plane detection and polygonalization in noisy 3D range images. In *Ieee/rsj international conference on intelligent robots and systems* (pp. 3378–3383). Nice, France

RIEGL Laser Measurement Systems GmbH. (2015). RIEGL VZ-400. Retrieved June 20, 2015, from http : / / www . riegl . com / products / terrestrial - scanning / produktdetail/product/scanner/5/

Rivadeneyra, C. & Campbell, M. (2011, January). Probabilistic multi-level maps from LIDAR data. *The International Journal of Robotics Research, 30*(12), 1508–1526

Rusu, R. B., Blodow, N., & Beetz, M. (2009, May). Fast Point Feature Histograms (FPFH) for 3D registration. In *Ieee international conference on robotics and automation* (pp. 3212–3217). Kobe, Japan

Rusu, R. B., Blodow, N., Marton, Z. C., & Beetz, M. (2008, September). Aligning point cloud views using persistent feature histograms. In *Ieee/rsj international conference on intelligent robots and systems, iros* (pp. 3384–3391). Nice, France

Rusu, R. B. & Cousins, S. (2011, May). 3D is here: Point Cloud Library (PCL). In *Ieee international conference on robotics and automation* (pp. 1–4). Shanghai, China

Sameer, A., Keir, M. et al. (2010). Ceres Solver. Retrieved May 20, 2017, from http://ceres-solver.org/

Schnabel, R., Wahl, R., & Klein, R. (2007, June). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, *26*(2), 214–226

Schonberger, J. L. & Frahm, J.-M. (2016, June). Structure-from-Motion Revisited. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 4104–4113). Las Vegas, USA

Shan, J. & Toth, C. K. (Eds.). (2018, February). *Topographic Laser Ranging and Scanning* (2nd). Boca Raton, USA: CRC Press

Shi, W. & Cheung, C. (2006, March). Performance evaluation of line simplification algorithms for vector generalization. *The Cartographic Journal*, *43*(1), 27–44

Siek, J., Lee, L.-Q., & Lumsdaine, A. (2002). *The Boost Graph Library: User Guide and Reference Manual*. Boston, USA: Addison-Wesley.

Suteng Innovation Technology Co Ltd. (2015). *RS-LiDAR Multi-Beam Series & Algorithms Brochure*. Shenzhen, China.

Tang, P., Akinci, B., & Huber, D. (2009, December). Quantification of edge loss of laser scanned data at spatial discontinuities. *Automation in Construction*, *18*(8), 1070–1083

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010, November). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, *19*(7), 829–843

Tang, S., Zhu, Q., Chen, W., Darwish, W., Wu, B., Hu, H., & Chen, M. (2016, September). Enhanced RGB-D Mapping Method for Detailed 3D Indoor and Outdoor Modeling. *Sensors*, *16*(10), 1589

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007, September). Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from Lidar data. In *Proceedings of the isprs workshop 'laser scanning 2007 and silvilaser 2007'* (Vol. XXXVI-3/W5, pp. 407–412). Espoo, Finland.

The Sanborn Map Company Inc. (2015). Spin indoor mapping. Retrieved June 8, 2015, from http://www.sanborn.com/spin-indoor-mapping/

Theiler, P. W. & Schindler, K. (2012, July). Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*, 173–178

Tsai, G.-J., Chiang, K.-W., Chu, C.-H., Chen, Y. L., El-Sheimy, N., & Habib, A. (2015, August). The performance analysis of an indoor mobile mapping system with RGB-D sensor. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-1/W4*, 183–188

Ulas, C. & Temeltas, H. (2012, July). Plane-feature based 3D outdoor SLAM with Gaussian filters. In *Ieee international conference on vehicular electronics and safety (icves 2012)* (pp. 13–18). Istanbul, Turkey

Velodyne Acoustics Inc. (2012). *User's Manual And Programming Guide: High Definition LiDAR Sensor HDL-32E*. San Jose, USA.

Velodyne Acoustics Inc. (2015). *User's Manual And Programming Guide: Velodyne LiDAR Puck VLP-16*. San Jose, USA.

Velodyne Acoustics Inc. (2018). *VLP-32C User Manual* (B). San Jose, USA.

Vexcel Imaging GmbH. (2016). Ground-breaking versatility: UltraCam Panther. Retrieved December 13, 2016, from http://www.vexcel-imaging.com/products/ultracam-panther/

ViAmetris 3D Mapping. (2015a). Viametris | continuous 2D indoor scanner iMS2D. Retrieved December 8, 2015, from https://www.viametris.com/ims2d

ViAmetris 3D Mapping. (2015b). Viametris | continuous indoor mobile scanner iMS3D. Retrieved December 8, 2015, from https://www.viametris.com/ims3d

ViAmetris 3D Mapping. (2019). Viametris | Backpack Mobile Scanner bMS3D LD5+. Retrieved April 11, 2019, from https://www.viametris.com/bms3d4cams%20https://www.viametris.com/backpackmobilescannerbms3d

Volk, R., Stengel, J., & Schultmann, F. (2014, March). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction, 38*, 109–127

Vosselman, G., Gorte, B. G. H., Sithole, G., & Rabbani, T. (2004, October). Recognising structure in laser scanner point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(Part 8/W2), 33–38.

Wang, C.-C. & Thorpe, C. (2002, May). Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings 2002 ieee international conference on robotics and automation (cat. no.02ch37292)* (Vol. 3, pp. 2918–2924). Washington D.C., USA

Wang, M. (2018). The State of Solid-State 3D Flash LiDAR. Retrieved March 18, 2019, from https://medium.com/@miccowang/the-state-of-solid-state-3d-flash-lidar-f0107dcc4c84

Wang, W., Sakurada, K., & Kawaguchi, N. (2016, November). Incremental and Enhanced Scanline-Based Segmentation Method for Surface Reconstruction of Sparse LiDAR Data. *Remote Sensing*, *8*(11), 967

Woo, H., Kang, E., Wang, S., & Lee, K. H. (2002, January). A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture*, *42*(2), 167–178

Zhang, J. & Singh, S. (2014, July). LOAM : Lidar Odometry and Mapping in Real-time. In *Robotics: science and systems (rss)*. Berkeley, USA.

Zhang, J. & Singh, S. (2015, May). Visual-lidar odometry and mapping: low-drift, robust, and fast. In *Ieee international conference on robotics and automation (icra)* (Vol. 2015-June, *June*, pp. 2174–2181). Seattle, USA

Zhang, L. & Ghosh, B. K. (2000, April). Line segment based map building and localization using 2D laser rangefinder. In *Proceedings 2000 icra. millennium conference. ieee international conference on robotics and automation. symposia proceedings (cat. no.00ch37065)* (pp. 2538–2543). San Francisco, USA