Creating Digital Twins of an indoor environment from an operator's perspective

Indoor 3D Virtual Map Generation with hand-held Solid-State LiDAR

Master's thesis in Systems, Control and Mechatronics

KARTHIK NAGARAJAN SUNDAR

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Creating Digital Twins of an indoor environment from an operator's perspective

Indoor 3D Virtual Map Generation with hand-held Solid-State LiDAR

KARTHIK NAGARAJAN SUNDAR

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Creating Digital Twins of an indoor environment from an operator's perspective
Indoor 3D Virtual Map Generation with hand-held Solid-State LiDAR
KARTHIK NAGARAJAN SUNDAR

Supervisor: Rikard Karlsson, Department of Electrical Engineering
Examiner: Petter Falkman, Department of Electrical Engineering

Master's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Creating Digital Twins of an indoor environment from an operator's perspective
Indoor 3D Virtual Map Generation with hand-held Solid-State LiDAR
KARTHIK NAGARAJAN SUNDAR
Department of Electrical Engineering
Chalmers University of Technology

Abstract

In recent times, Virtual Commissioning (VC) has flourished in developing new manufacturing systems while ensuring high-quality, and flexible systems with reduced time and cost requirements. One application of VC is Virtual Reality, a technology that can enhance the process with more realistic visualization and validation. The highest level of VC can be achieved by utilizing a Digital Twin in the form of a 3D virtual map of an environment, however gathering a high-resolution, reliable, and accurate virtual map is crucial. Amongst various sensors, range sensors like LiDARs are advantageous in providing depth information of the surroundings, and further, the solid-state LiDARs produce high-frequency, high-resolution frames of depth information. With the help of the new Intel RealSense L515 LiDAR camera, data is collected from a point of view that corresponds to the human or operator, and a SLAM (Simultaneous Localization and Mapping) algorithm is implemented to generate 3D virtual maps of the environment. From experiments, it has been verified that the generated virtual maps are reliable and fairly accurate that they can be utilized as a Digital Twin in Virtual Commissioning for an indoor environment. For future work, a further investigation of the algorithm in terms of robustness toward various environments is suggested.

Keywords: Virtual Commissioning, digital twins, virtual mapping, solid-state LiDAR, point cloud, registration, SLAM, pose estimation.

Acknowledgements

I take this moment to express my gratitude to all those who have helped me in completing the thesis work. Firstly, I would like to thank my supervisor Rikard Karlsson and examiner Petter Falkman at Chalmers University for guiding me, and providing valuable feedback throughout the thesis. Moreover, I would like to thank all my friends and family for their constant support and motivation.

Karthik Nagarajan Sundar, Gothenburg, June 2023.

Contents

List of Figures

List of Tables

1
Introduction

In this chapter, the concept of Virtual Commissioning along with its importance and hurdles are introduced. Further, the motivation of the project, expected outcome, limitations, and outline of the report are discussed.

1.1 Background

Virtual Commissioning (VC) is a process that enables virtual testing, verification, and validation of software or a machine by utilizing a model of an environment. In recent times, the concept of VC has become a crucial part of developing new manufacturing systems. Drath et al. [1] investigated the evolution of industries using the concepts of Virtual Commissioning for testing the manufacturing systems from an economic point of view. Lee et al. [2] conducted a survey to identify issues in the implementation of VC for automated production systems by utilizing a virtual plant model and a real controller. The workflow of a Virtual Commissioning system begins with generating simulations of the model and establishing an interaction between the environment, application software, and the technology or a machine performed in a virtual environment. This interaction is used to test and detect any failures, optimize the system, and verify it in different scenarios before actually implementing the technology in action. According to Shahim et al. [3] the concept of VC helps in ensuring high-quality production systems, with lower time and cost requirements for testing the system, and also speed up the time requirements for a real commissioning process.

One interesting field to be considered is Virtual Reality (VR). As concluded by Dahl et al. [5] using VR technology enhances the process of Virtual Commissioning with more realistic visualization and validation. Virtual Reality refers to the application of computer tools for creating a model of an environment and providing the user with an opportunity to interact with it. Zheng et al. [4] provides the latest developments in VR considering applications in engineering and medical fields. According to various articles [6, 7, 8], VR technology is applicable over a wide range of fields such as military, sport, healthcare, education, and automotive industry.

Falkman et al. [9] investigate the different levels of Virtual Commissioning. The highest level of Virtual Commissioning can be achieved by utilizing a Digital Twin in the form of a 3D virtual map representing the information of an environment as a replica. A 3D map provides a realistic view of a location that can be utilized by engineers, local authorities, and planners. From an industrial point of view, through mapping techniques and a 3D scanner, a virtual world of the whole production unit can be constructed. Apart from the discussed advantages of implementing VC, this virtual map also helps increase the flexibility of the production to adapt the future changes much faster, without interrupting the current process. According to Lee et al. [2], although there is a significant reduction of debugging and correction costs in implementing VC techniques, high expertise in modeling and simulation is required. The highest level of VC implementation requires a reliable, and accurate 3D virtual map of the environment.

The process of gathering high-resolution information about the surroundings can improve the immersiveness of the generated 3D map but requires high computational power and better sensors. Range sensors such as 3D LiDARs (Light Detection and Ranging) 3D laser scanners, and vision

sensors such as monocular, stereo, and RGB-D cameras can be used to perform this task. Even though the cameras are cost-effective, the produced images suffer from scale uncertainty due to the loss of depth information. This can be solved by using LiDARs, specifically solid-state LiDARs which help in providing high-frequency, high-resolution frames of depth information. One such sensor is the new Intel RealSense L515 LiDAR Camera which is lightweight and easy to carry around by an operator instead of mounting on any AGV or mobile robot and aids in acquiring data from a human's perspective. Further through the report, relevant theories, and methods of implementing a mapping technique to generate the virtual map, followed by evaluation methods of the constructed map will be discussed.

1.2 Previous Work

Multiple surveys of virtual maps currently in use for Virtual Commissioning in diverse environments are presented by various authors. These articles [1, 2, 3] focus on implementing the concepts of VC and advancing the development of manufacturing systems from an industrial point of view. To et al. [10] investigates the 3D reconstruction of an environment using RBG-D cameras mounted on a drone and incorporates a real-time combination of Artificial Intelligence and Digital Twin augmentation. Zhang et al. [17] explores the fusion techniques of optical images and LiDAR point clouds to generate 3D maps using the photogrammetry approach. This work [13] proposes a SLAM system for reconstruction using monocular, stereo, and RGB-D cameras, and implements the algorithm in real-time on standard CPUs. The authors also claim that the proposed algorithm is suitable for a variety of environments from indoors to industrial environments and autonomous driving. Remondino et al. [15] reviews an automated approach of photogrammetry techniques on large datasets for 3D reconstruction purposes. The results were also verified for replicability and redundancy in sequences of complex environments. The article by Zlot et al. [11] explores the use of spinning 2D LiDAR combined with industrial-grade IMU to construct a large-scale 3D map of an underground mine and verify if the required equipment can fit the pathway using the virtual commissioning technique.

1.3 Purpose

The purpose of this thesis work is to evaluate the Intel RealSense L515 LiDAR Camera's performance in collecting data for Virtual Commissioning from a user's perspective in a low-cost system.

1.4 Limitations

The following limitations are set for this thesis project

- Considering the focus on low-cost systems, only the depth information from the L515 solid-state LiDAR Camera will be used for processing.
- No code optimization for real-time implementation of the method is involved.

1.5 Thesis outline

The thesis work is presented as five chapters starting with a brief introduction and structured further as follows:

- Chapter 2 Presents all relevant theories used in this thesis

- Chapter 3 Describes the problem formulation and implementation
- Chapter 4 Provides the evaluation methodologies used in this thesis
- Chapter 5 Presents the results of various experimentation conducted
- Chapter 6 Includes various implications drawn from the results and possible future works
- Chapter 7 Concludes the thesis

2
Theory

In this chapter, all concepts and relevant theories behind the main approach used in this thesis are presented and discussed here. The structure is broken down into three major sections such as LiDAR, point cloud registration, and SLAM.

2.1 Light Detection and Ranging

The Light Detection and Ranging (or LiDAR) is an active range measurement sensor that uses visible light, ultraviolet, or near-infrared light as mentioned in the articles [24, 25]. The basic principle behind range estimation is that a laser in the form of a pulse or continuous wave is emitted from a transmitter, gets reflected by an object, and the reflected energy is captured by the receiver. The *Time of flight* which represents the total time taken to transmit, and receive the reflected beam is calculated. Further, the distance between the transmitter and object is determined using the Eq. (2.1) where the Time of flight is halved to include only the time taken for the beam to travel in one direction.

$$Distance\ d = \frac{(Speed\ of\ light) \times (Time\ of\ flight)}{2} = \frac{c \times t}{2} \tag{2.1}$$

LiDARs can be classified based on the scanning mechanism. In order to produce a sweeping scanning effect, a typical spindle-type rotates 360°to provide a full view of the environment whereas a solid-state LiDAR consists of a fixed Field Of View (FOV) with no moving parts in it.

2.1.1 Rotating LiDAR

A Mechanical or Rotating LiDAR consists of a single beam or a single-dimensional array of laser emitters and detectors which rotate along the vertical axis at a particular rpm. The rotation motion helps in covering a wide field of view around the sensor. With this type of LiDARs, the distance and angle are calculated together to generate a point cloud of the surroundings. However, the conventional rotating LiDARs are large, quite expensive, have low frame rates, and consist of moving parts which makes them hard to maintain. Moreover, during the data collection, the process of obtaining a complete scan is not simultaneous and depends on the rotation of the sensor. This poses an issue when the sensor is in motion since motion distorts the generated point cloud. Thus either the sensor should be stationary for each scan or these distortions must be corrected, which is performed by using external sensors to estimate the motion between each LiDAR measurement. Zhang et al. [30] describe a distortion removal algorithm for point clouds.

2.1.2 Solid-State LiDAR

In a solid-state LiDAR, there are no moving parts, instead of the rotation of the setup, this type consists of an array of detectors placed both horizontally and vertically to cover a particular field of view at that instant of time. This is greatly advantageous over its rotating counterpart since the data collection process of the environment happens simultaneously. Even under motion, the area

within the FOV is captured instantly, hence no distortions occur in the generated point cloud. The solid-state LiDARs are also beneficial since it is cost-effective, lightweight with no moving parts involved, have high frame rates, and improved service life. The main downside of the solid-state is the small Field of View (FoV) compared with the rotational/mechanical LiDARs and thus requires an algorithm to stitch all the captures together. A typical solid-state LiDAR is made of a silicon chip and using Microelectromechanical systems (MEMS) based mirrors, a laser beam can then be directed towards the environment without using motors or gears. Jingye et al. [31] proposed a study made with various types of solid-state LiDAR other than MEMS technology.

2.2 Point Clouds

2.2.1 Point Clouds

There are various depictions to represent 3D data. A point cloud as explained in the articles [32, 33] is one of the methods for representing the 3D data produced by either LiDARs, 3D scanners, 3D cameras, or by photogrammetry software. The point cloud generally consists of a set of data points in 3D space as cartesian coordinates (x, y, z) which depict a single geometrical feature of an objects shape. The applications of point clouds include generating 3D CAD models for parts quality inspection and manufacturing, and virtual visualization of an environment. Normally, the point clouds generated using LiDARs are unorganized meaning sparse, irregular, and in an arbitrary fashion, and represented as a single-dimensional array, whereas the organized point clouds are described in a structured manner and represented by a 2D array.

$$P = \left\{ x_i \in \mathcal{R}^3 \right\}_{i \leq N} \tag{2.2}$$

A point cloud can be represented in 3D Euclidean space as defined in the equation (2.2).

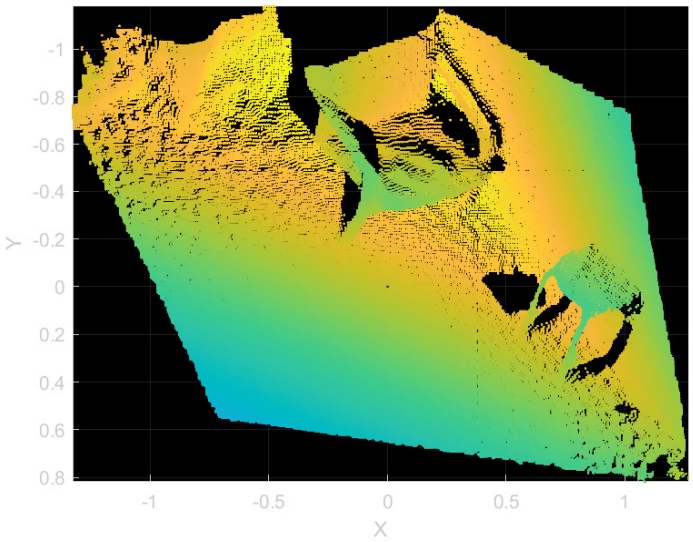

Figure 2.1: *Point cloud produced by L515 depicting the scene with an arm-chair and a stool*

2.2.2 VoxelGrid Filter

The VoxelGrid Filter is a method of downsampling a point cloud based on a voxelized grid approach. A voxel grid can be defined as a set of 3D boxes or cubes spread in space. Initially, the points that

lie within the bounds of a voxel are assigned to that voxel and the filter downsamples the 3D point cloud by taking a spatial average i.e. by the computing centroid of the points and statistically combining them into one data point for that voxel. The grid step defines the size of each voxel. Figure 2.2 represents two examples of applying the VoxelGrid filter with various grid steps. When the grid step is set larger (bottom left), all points are confined within one big voxel and the filter downsamples all the data points to one centroid, but when the grid step is set smaller (bottom right) the data points are confined within four voxels and the filter downsamples to four centroids corresponding to each voxel. A good balance of downsampling the point cloud while preserving the details of the environment is achieved by experimenting with different grid steps.

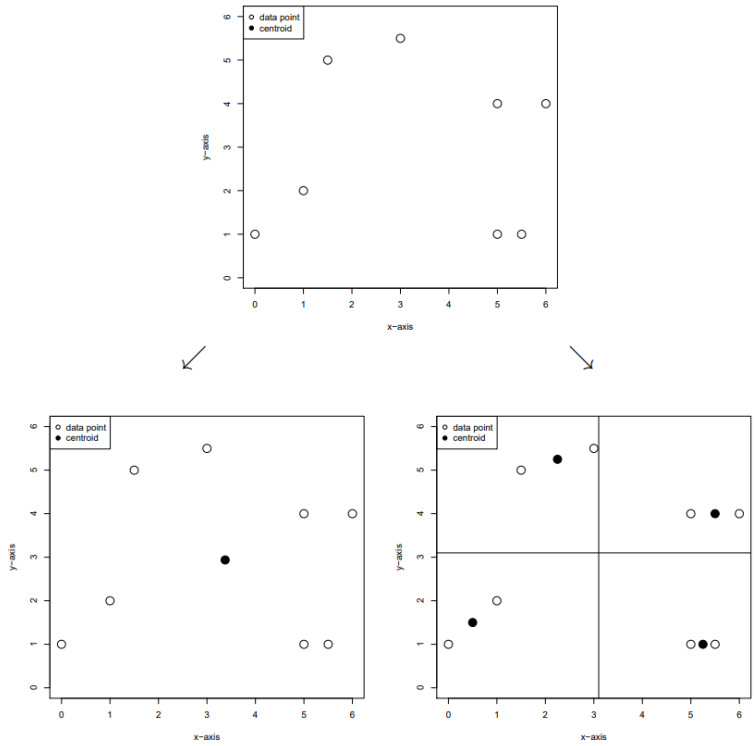

Figure 2.2: *Example of VoxelGrid filter with larger grid step(bottom left) and smaller grid step(bottom right).*

2.3 Point Cloud Registration

Point cloud registration or generally known as scan matching as defined in the article [55] is a process of estimating a spatial transformation like rotation, translation, and scaling which aligns two point clouds. Various applications of point cloud registration include perception-based autonomous driving, 3D reconstruction, robotic manipulation, SLAM, and Virtual Reality. The main purpose of estimating the transformation is to continuously merge point sets, map the new measurements with the previously generated global map, and localize the device by estimating its relative pose. Various algorithms have been proposed for such estimations as Iterative Closest Point (ICP), and correspondence-based methods which will be discussed in the following sections.

2.3.1 Estimation of optimal Rigid Transformations

A rigid transformation is a geometric transform for mapping the 3d points from one frame of reference to another frame of reference in the Euclidean space. In order to preserve length, the

transformation includes combinations of rotations, translations, and reflections. Let us consider, two corresponding sets of point clouds A and B where $A = \{a_i\}$ and $B = \{b_i\}$, $i = 1...N$ points, which represents the environment as captured by the laser scanner at two subsequent time stamps. One major condition is that a minimum of 3 unique points is necessary for computing the transformation as explained in the article [54].

For every successive frame, there exists a region common to both the frames which overlap when plotted as pairs. Hence with this overlapping region, the set B must be transformed to align perfectly with the set A, for which the rigid transformation is represented as

$$B = RA + t \tag{2.3}$$

where R and t represent the 3×3 rotation matrix and $1 \times N$ translation vector for the points respectively. An optimal solution is reached once the rotation matrix R and translation vector t is found to minimize the least-square criterion.

$$err = \sum_{i=1}^{N} \|Ra_i + t - b_i\|^2 \tag{2.4}$$

Eggert et al. [54] compares major algorithms for estimating 3D rigid transformations. Amongst various methods of finding the optimal transformation between points the Singular Value Decomposition (or SVD in short) is likely the simplest way to calculate the matrix R and vector t.
The algorithm to find the optimal rigid transformation as follows:

- Firstly, the centroids of each sets of points are calculated by

$$\begin{aligned} \text{centroid }_A &= \frac{1}{N}\sum_{i=1}^{N} A^i \\ \text{centroid }_B &= \frac{1}{N}\sum_{i=1}^{N} B^i \end{aligned} \tag{2.5}$$

- A common origin is fixed such the centroids of both sets are at the origin and hence the points are shifted accordingly
- Then the optimal 3×3 rotation matrix R is calculated by

$$\begin{aligned} H &= (A - \text{ centroid }_A)(B - \text{ centroid }_B)^T \\ [U, S, V] &= SVD(H) \\ R &= VU^T \end{aligned} \tag{2.6}$$

  where,
  - $H = USV^T$ is the covariance matrix, and the function of SVD is to factorize the covariance matrix into U, S, and V matrices
  - U represents an orthogonal matrix composed of columns of the left singular vectors of H
  - S represents a diagonal matrix composed of the singular values of H
  - V represents an orthogonal matrix composed of columns of the right singular vectors of H
- Finally, after calculating R, the translation vector can be computed as

$$\begin{aligned} \text{centroid }_B &= R \times \text{ centroid }_A + t \\ t &= \text{ centroid }_B - R \times \text{ centroid }_A \end{aligned} \tag{2.7}$$

2.3.2 Iterative Closest Point

Iterative Closest Point (or ICP in short) is an algorithm that aims at estimating a rigid transformation between point clouds by minimizing the difference between two point clouds as described by Zhang et al. [56]. The ICP algorithm achieves this by iteratively associating points (finding correspondences) between any two successive sets of points. The algorithm requires two point clouds, to begin with, known as source and target. The source is the point cloud that needs transformation to align with the target which is a fixed point cloud. For every iteration, a search algorithm is implemented in the source to find the closest neighbour of each point with respect to the target. With this established correspondence, the rigid transformation between the source and target can be estimated using the optimal estimation method described in section 2.3.1.

With this estimated rigid transformation, the source point cloud is transformed to match with the target point cloud and then the Root Mean Squared Error (RMSE) is calculated between the point clouds. This RMSE evaluation metric helps in identifying the best solution. The end goal of the ICP algorithm is to minimize the RMSE value through the iterations and the final best rigid transformation matrix is extracted corresponding to that minimum value of RMSE. In an ICP algorithm, initially, there is no data association involved, and occurs simultaneously during pose estimations. Generally, the point clouds are downsampled significantly to improve computational efficiency. The algorithm 1 represents the point-to-point variant of the ICP method.

Algorithm 1: Iterative Closest Point [57]

Inputs: Point clouds Target$A = \{a_i\}$, Source$B = \{b_i\}$ and initital transformation matrix M_0
Initialize $M_{curr} = M_0$ or M_{curr} = identity matrix if M_0 not provided, threshold to any value
while *converged == False* **do**
    $M_{prev} = M_{curr}$
    for $i = 1, 2, ...N$, *where N is the number of associated points between source and target frames* **do**
        cP_i = closest point in A to $M_{prev} \cdot b_i$
        if $\|cP_i - M_{prev} \cdot b_i\| \leq threshold$ **then**
            Set $w_i = 0$
        else
            Set $w_i = 1$
        end
    end
    compute $M_{curr} = \underset{M_{curr}}{\operatorname{argmin}} \left\{ \sum_i w_i \|cP_i - M_{prev} \cdot b_i\|^2 \right\}$
end
Return transformation matrix M

2.3.3 Correspondence-based Registration

Generally, when there is no correspondence established before the pose estimation step, the association process can happen simultaneously within the algorithm like the ICP (2.3.2). In case, if there is a method of extracting and matching unique features from the point clouds between two frames, a correspondence between the frames of point clouds can be established as per the author Radu Rasu in the article [59] and the scans can be registered. Based on the established correspondences, the initial approximate pose is estimated by computing the optimal Rigid transformations 2.3.1 and then further refined by using ICP. Using the features-based approach is advantageous since instead of using the entire point clouds which are prone to noisy measurements, unique low-level features can help increase the computational efficiency and reduce storage requirements.

In terms of computer vision, a feature contains a piece of information regarding an image or a point cloud. The information possessed by a feature often represents certain properties that are unique about a region of interest. Features may include points, edges of an object, surface, and objects. In subsequent frames of images or point clouds, there exists a certain region that overlaps and is very helpful in matching the extracted features. In this thesis work, the point clouds are highly focused. Within point clouds, at the overlapping regions, a complete point-to-point correspondence cannot be guaranteed, hence in order to tackle this issue, a feature extraction algorithm is necessary.

In order to establish a features-based correspondence, the algorithm is broken down into two steps, detection and description following the article [65]. The detection step involves finding features or unique key points like intersections of an object's edges, and corners, specially termed as low-level features. Various algorithms have been proposed from the articles [60, 61, 62] like Harris corner detection, SIFT, and FAST key points in the case of 2D images. Feature extraction algorithms for 3D point clouds include 3D-SIFT, and Fast Point Feature Histogram (FPFH) from the authors of [63, 64].

The next step after the detection of key points is the description step, where the region around the extracted features is analyzed and described. The main purpose of the description is to analyze the region around the key point, extract unique and distinctive information available across the successive frames, and recognize the gathered information when encountered in other frames. Various algorithms for the description stage for 3D point clouds, Point Feature Histogram (PFH), and FPFH have been proposed in [64, 65].

Amongst the partially overlapping scans of the environment, a set of features are extracted and matched across the frames of the scan. For every pair selected, a rigid transformation is estimated as explained in the section (2.3.1). An outlier-rejection algorithm is necessary to refine the matched pairs and the calculation of the transformation matrix. The RANSAC method is commonly used as an outlier-rejection algorithm.

In this thesis, the FPFH descriptor was implemented to extract features and form correspondences. The FPFH is an extension work of the Point Feature Histogram (PFH) where the geometric properties of n nearest neighbours of a point are encoded using the average curvature of multidimensional histogram around any point. With the help of such high-dimensional space, more unique features that are invariant to the 6-position can be extracted. Unlike PFH, the algorithm has been simplified and optimized to speed up the calculations of FPFH. An in-depth explanation of the descriptor can be found in [64]. This proves that FPFH is computationally efficient and more accurate when compared to other methods.

2.4 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping, or commonly known as SLAM, is the process of constructing a map of an unknown environment while simultaneously determining the location of the

device within the constructed map. This algorithm was mainly developed to tackle the problem of the famous chicken and egg: where in order to construct a map of an unknown environment, the location of the device must be determined so that the map can be updated with new landmarks, but for localizing the device's current position, a map of the surroundings is necessary to determine it's location in that environment.

The idea of SLAM was first proposed by Smith et al. and Leonard et al. [36, 37] and later various methods have been developed and honed to support each application and type of sensors used. SLAM has emerged in the robotics industry as a solution for developing robots with the ability to navigate around the industrial areas without colliding into obstacles and walls. Since then this algorithm has taken various adaptions to better suit environments like indoor, outdoor, and even underwater. Now SLAM is being used in unmanned vehicles such as mobile robots, drones, and Unmanned Aerial Vehicles (UAV), as well as manned vehicles like Autonomous vehicles. Modern techniques of SLAM are being developed for minimal invasive surgery as described in the articles [38, 39], and for military applications [40]. Using SLAM for Virtual Commissioning is a relatively new concept.

SLAM uses perception sensors such as cameras, LiDAR, RADAR, and SONAR to perceive the surroundings and might also include the sensors like Inertial Measurement Unit (IMU). Sahoo et al. [41] uses IMU measurements for correcting the pose estimation in the SLAM algorithm. Hence from the data collected using the combination of sensors, a map is being built and the device is localized relative to the map based on features extracted such as walls, furniture, etc. The SLAM pipeline mainly depends on the type of sensor being used. For example, if monocular or stereo cameras are used then it is Visual SLAM (vSLAM in short), and if LiDARs are used it is known as LiDAR SLAM.

2.4.1 Visual SLAM

Visual SLAM is a type of SLAM algorithm which leverages the 3D vision for performing localization and mapping functions in an unknown environment with the help of a sequence of images generated using cameras. There are different types of cameras namely Monocular, Stereo, RGB-D cameras, and can also be positioned as an array to produce a panoramic effect. Visual SLAM technology is currently of high interest since it is low-cost that produces a large volume of information. The downside of using cameras is that even though they are cost-effective, they suffer from scale ambiguity due to the loss of depth information. The algorithm of a vSLAM system can be explained as a step-by-step procedure. Firstly a set of points or low-level features are extracted as explained in the articles [42, 43, 44] and kept track of each feature through the successive frames in the sequence of images. Then the 3D position of each feature is triangulated using the matched features obtained between the two successive frames, in a process known as feature-point triangulation.

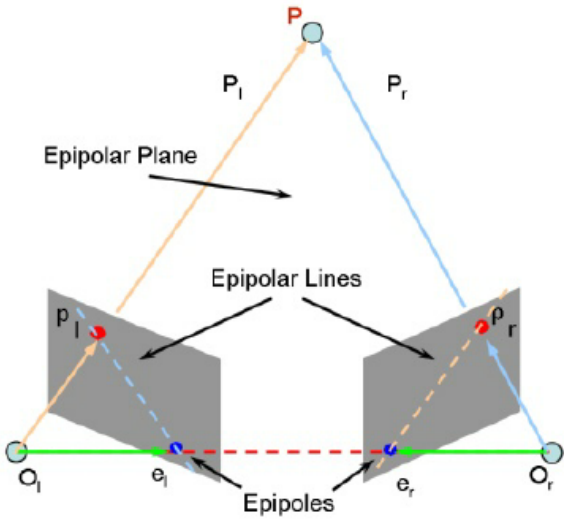

Figure 2.3: *Pictorial representation of feature-point triangulation. P is the reconstructed point and P_l, P_r are the feature correspondence extracted from the images*

A transformation matrix that helps in calculating the pose is calculated by minimizing the distance between the two points. Meanwhile, it is to be noted that not all features are inliers. Hence an outlier rejection algorithm is necessary like Random sample consensus (RANSAC) as demonstrated by Fischler et al. [46]. The framework constantly works on minimizing the reprojection error (i.e. the difference between the projected point and actual point) generally using an algorithm known as bundle adjustment. Finally, at the back-end of the algorithm, every new instance of the scene is verified if that scene has already been visited previously in a process known as loop closure detection or landmark detection, and the map is optimized, to tackle the problem of drift that occurs over the period of travel. Figure (2.4) represents a complete pipeline of the Visual SLAM system which includes odometry measurements for achieving better results but is not always necessary.

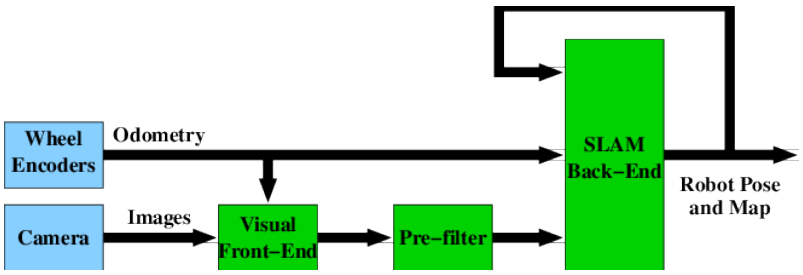

Figure 2.4: *Visual SLAM Pipeline*

2.4.2 LiDAR SLAM

A LiDAR-based SLAM is a system that uses a laser sensor for constructing a 3D map of the environment. The working of LiDARs and types can be found in the section (2.1). The data from LiDARs are generally in the form of unordered point clouds as explained in (2.2.1). The LiDAR SLAM pipeline begins with collecting scans of the surroundings in the form of point clouds. These point clouds are prone to noise as the range increases. Thus the raw data is pre-processed to filter the noisy measurements and down-sampled to help improve computational efficiency as described by Escolano et al. [47]. The processed point clouds of the current frame and the next subsequent frame are aligned to determine the spatial transformation (for instance rotation, translation, and scaling) by minimizing the Euclidean distance between the points through a process called point cloud registration. Various methods for point cloud registration are introduced by Besl et al.

and Wang et al. in the articles [48, 49]. Using this transformation matrix, the relative pose of the sensor is estimated between two frames and repeated throughout the process to track the trajectory. There are various methods in point cloud registration like Iterative closest point (ICP), and correspondence-based registration which is explained in the section (2.3). Finally, similar to vSLAM, the loop closure detection or landmark detection is established with sub-map or local-map building and combined with graph-based optimization, to tackle the problem of drift that occurs over the period of travel. The figure (2.5) represents a complete pipeline of the LiDAR-based SLAM system.

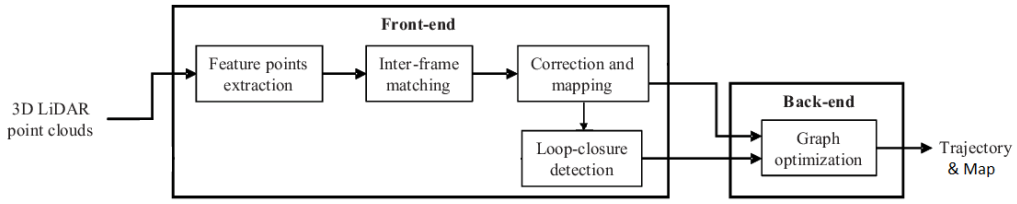

Figure 2.5: *LiDAR SLAM Pipeline*

2.4.3 SLAM problem Formulation

The SLAM problem is solved by combining the estimation of device trajectory and constructing a map of the environment. A basic assumption while solving SLAM is that the environment is unknown to the algorithm. The device in motion acquires the perception of the environment as a measurement. Hence with the gathered information and assumptions, the SLAM problem can be formulated for determining the posterior probability of the trajectory and map as

$$p\left(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{1:T}, \mathbf{x}_0\right) \tag{2.8}$$

where,

- $\mathbf{x}_{1:T}$ represents the trajectory of the device
- $\mathbf{z}_{1:T}$ represents the frames of perception information about the environment
- $\mathbf{m}$ represents the constructed map of the environment
- $\mathbf{x}_0$ represents the initial position of the trajectory which is chosen arbitrarily.

Amongst the various types of implementing the SLAM framework, the GraphSLAM is highly relevant to this thesis and hence the filter-based algorithms will not be explained in detail.

2.4.4 GraphSLAM framework

The interesting approach of formulating the SLAM problem using a graph that consists of nodes and edges to represent the attributes of SLAM is known as GraphSLAM according to Grisetti et al. [53]. Various applications have been developed using this method for optimization according to the articles [13, 44, 66]. The graph serves as a representation of all poses and measurements of the device and the associated constraints. These attributes are then reduced to residuals from which the cost function is derived. The goal of the graphSLAM is to calculate an optimal solution of pose and measurement configuration such that it minimizes the cost function while satisfying every constraint. The figure (2.6) represents a visualization of the GraphSLAM system.

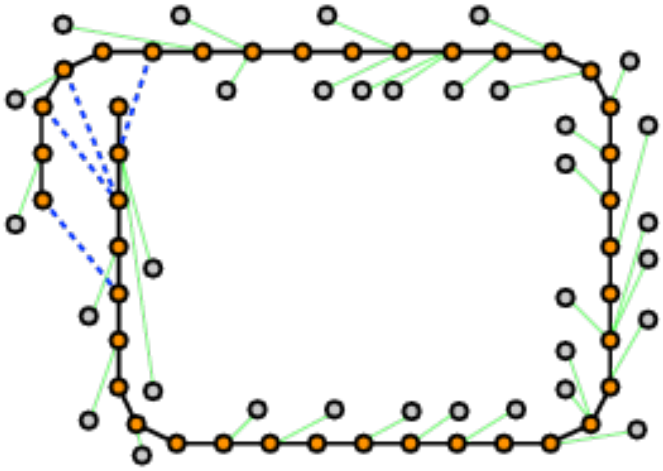

Figure 2.6: *A simple GraphSLAM system visualization*

From the figure 2.6, the orange circles are the nodes of the graph which correspond to the poses of the device and are hence termed as pose nodes. Each pose represents the rotation and translation of the sensor during its trajectory in the environment. The black lines denote the edges that represent the relative pose constraints between two successive pose nodes. The grey circles are another set of nodes that correspond to the measurements of the environment and are termed measurement nodes.

Whenever the graph is updated with a new pose node or a measurement node, a constraint edge is developed between the new node and the previous relevant node. Such constraints may be contradicting due to the observations being prone to noise. The residual functions are depicted by edges in the form of green lines from figure 2.6. These residual functions help in calculating the error between the predicted and the observed measurements. The blue dotted lines connecting other nodes in the figure represent the places that have already been visited and determined through the process of loop closure detection. These detections help in optimizing the graph in the later stages of the framework.

The entire Graph SLAM framework can be split into Front-end and Back-end processing.

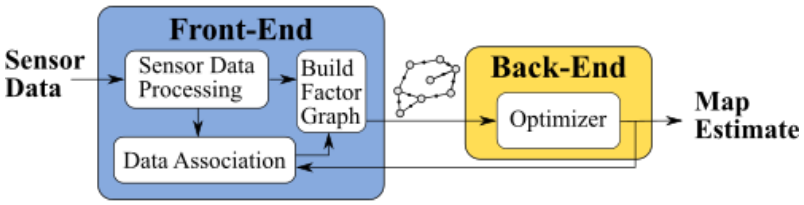

Figure 2.7: *Block diagram representing the GraphSLAM framework*

2.4.4.1 Front-end

The front end of the framework begins with gathering data as observed measurements directly from the sensor. With the observed measurements, a data association step is required in associating

the observed measurements with the predicted measurements. Through the association step, the constraints are developed between the nodes. Finally, the graph is built by estimating the current pose and relative pose for every new node added to the system and updating it to the graph. The overall function of the front-end framework is to create and update the nodes, and edges of the graph, hence preparing it for the optimization process.

2.4.4.2 Back-end

The back-end of the framework is accountable for optimizing the graph generated from the front-end of the framework. There are two variants in optimization, namely local and global optimization. The local optimization optimizes the map for a specified section while in the global optimization, the entire graph is optimized. The graph optimization is based on least-squares error minimization and popular solvers like Gauss-Newton or Levenberg-Marquardt (L-M approach) can be used to arrive at an optimal solution. The equations used in this section are based on Grisetti et al. [53].

To solve the slam problem formulated as shown in the Eq.(2.8), the log-likelihood function $\mathbf{l_{ij}}$ of the measurements is obtained by

$$1_{ij} \propto \left[\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right]^T \mathbf{\Omega}_{ij} \left[\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right] \tag{2.9}$$

where,

- $\mathbf{x} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_T\right)^T$ is the vector of pose nodes,
- $\mathbf{z}_{ij}$ and $\mathbf{\Omega}_{ij}$ are the mean and information matrix of the virtual measurement for transforming the observation at i to overlap with the observation at j.
- $\hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the predicted virtual measurements.

For every real observation gathered by the device, there exists a difference with the prediction and hence this error is given by the function

$$\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{2.10}$$

Finally, the goal of maximum likelihood method is to solve the Eq.(2.11) for esimating the nodes $\mathbf{x}^*$.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \mathbf{F}(\mathbf{x}) \tag{2.11}$$

where, $\mathbf{F}(\mathbf{x})$ is the negative log-likelihood of all observations described as

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{\mathbf{ij}}^{\mathbf{T}} \mathbf{\Omega}_{\mathbf{ij}} \mathbf{e}_{\mathbf{ij}} {}_{\mathbf{F}_{ij}} \tag{2.12}$$

Approaches to solve the Eq.(2.11) and to estimate the posterior from Eq.(2.8) using the Gauss-Newton or the L-M method are briefly explained in [53].

3

Methods

In this chapter, a brief explanation of the methodology implemented in the thesis work has been presented. The implementation is broken down into sections and can be followed as a step-by-step procedure.

3.1 Data collection

The implementation of the LiDAR-based SLAM pipeline begins with collecting scans of the provided environment using Intel's RealSense LiDAR Camera L515. The L515 is a solid-state LiDAR and the working of it has been presented in the section (2.1.2). The L515 LiDAR Camera features a depth camera, an RGB camera, and an internal Inertial Measurement Unit (IMU). Table 3.1 represents various operating modes of the L515 device and table 3.2 represents the specifications of the device. The complete data sheet of the L515 sensor is available on Intel's website for reference.

Depth Resolution (W × H)	Number of depth points per second	FOV
(320 × 240)	2.3M	70° × 55°
(640 × 480)	9.2M	70° × 55°
(1024 × 768)	23.6M	70° × 55°

Table 3.1: *Information about various operating modes of the RealSense LiDAR Camera L515*

Depth frame rate	30 FPS
RGB frame rate	30 FPS
Focal Length	1.88 mm
RGB camera FOV (W × H)	69° × 42°

Table 3.2: *Specifications of the L515 LiDAR camera*

Figure 3.1: *Intel RealSense LiDAR Camera L515*

For this thesis work, the first operating mode of the device with a depth resolution of (320 × 240) was considered since through a USB 2.0 port, the performance of the L515 device is reduced. The challenges of collecting scans for the SLAM pipeline include mounting the sensor to a mobile robot or vehicle, moving the device in an environment, and gathering various sensor measurements. For this project, the main focus is to collect data from a point of view that corresponds to a human or operator, and this L515 LiDAR Camera aids in the process through its advantages like no moving parts and easy to carry. Hence the device is hand-held throughout the data collection process with 6 DOF movement. Despite the presence of an RGB camera and IMU, only the measurements from the LiDAR depth camera have been gathered to generate a 3D map in a low-cost system.

The process of capturing data from the device and storing it in the local storage of the computer is achieved through software called Intel RealSense Viewer. It is a program developed by the manufacturer as a part of the Intel RealSense Software Development Kit (SDK) latest version 2.0. Since the sensor is hand-held, the L515 is connected to a Laptop as a setup and the setup is moved in the environment by a person holding it.

The recorded data from the SDK is stored in files as .bag format. A separate script using the SDK has been developed for converting the .bag files to Matlab variables (.mat files) containing a sequence of the LiDAR data in the form of point clouds. This process will be repeated a number of times throughout the project and the collected scans will be used for verification of reliability and accuracy.

3.2 Pre-processing

The next step after gathering enough raw data is to pre-process it and ensure that the processed data is applicable for implementing it in the algorithm. The LiDAR data in the form of raw point clouds are prone to noise as the range increases. Hence for the provided environment, a limit of 3.5 meters in depth is set as a threshold and any 3D point that lies at a distance of 3.5 meters or more away from the device is removed. The consequence of filtering with the specified threshold is that appx. 2% of the 3D points in each scan have been filtered out. Moreover, the point clouds are down-sampled to make it efficient for the algorithm to proceed further. The algorithm is implemented on a system consisting of an Intel i5 CPU with 6 GB RAM. With this limitation of system specifications, the down-sampling is accomplished using the box grid filter as described in the section 2.2.2 with the step size of the 3D box set at 0.03. The data collected from the device is usually 30 FPS, and none of the frames has been skipped in between to preserve the accuracy. Hence after the pre-processing stage, filtered, down-sampled frames of the point cloud are achieved.

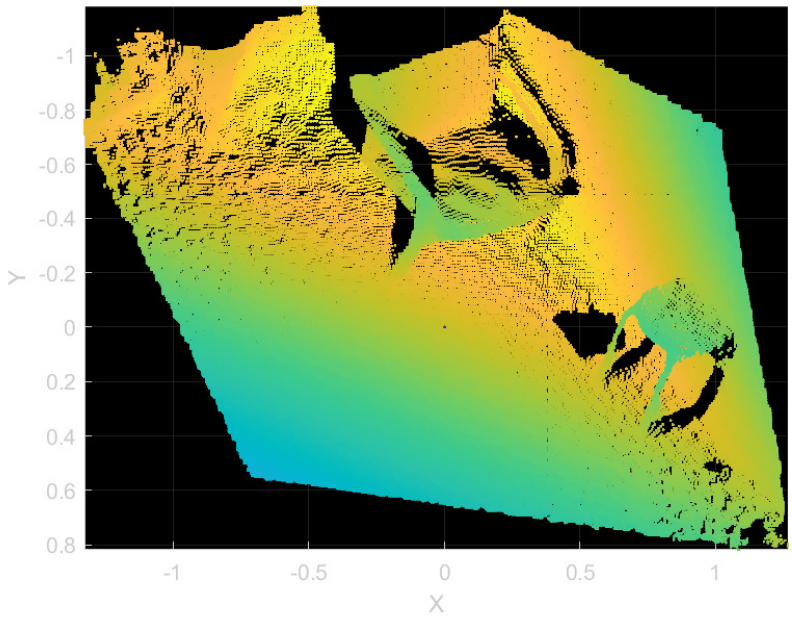

Figure 3.2: *Point cloud of the first frame depicting the scene with an armchair and footstool before pre-processing the LiDAR data.*

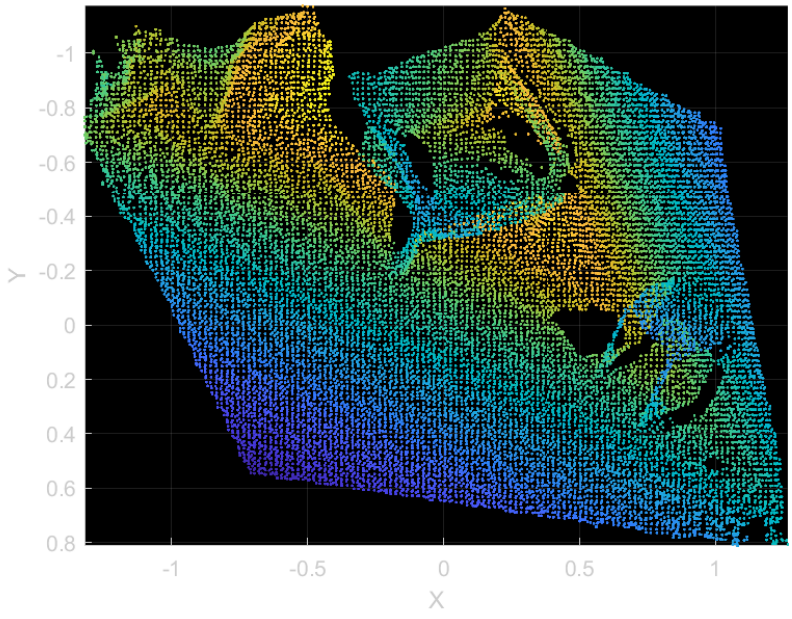

Figure 3.3: *Point cloud of the first frame depicting the same scene after pre-processing the LiDAR data.*

The main drawbacks of filtering data are that the generated final 3D map becomes less immersive to the user, and some fine details of the environment like small-sized objects, etc. However, this stage makes the algorithm light and computationally efficient while preserving the accuracy of the generated map.

3.3 Pose estimation and Localization

The pre-processed LiDAR data is now available in the form of a sequence of frames of point clouds. The algorithm begins with extracting unique key points or features from each frame as described in the section (2.3.3). Then the features from the current frame and the next successive frame are matched and a correspondence is established between the two frames. The respective 3D points of the matched features are extracted separately and used for estimating the optimal rigid 3D transform as explained in the section (2.3.1), followed by refining the transformation matrix using Iterative Closest Point as described in the section (2.3.2).

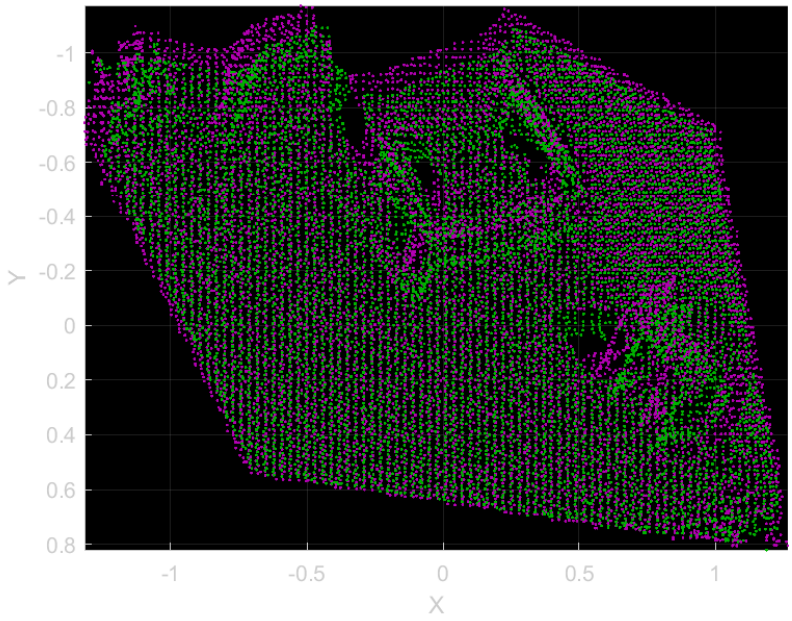

Figure 3.4: *Point clouds of frames 1 and 60 depicting the scene with an armchair and stool represented in pairs. Magenta for frame 1 and green for frame 60.*

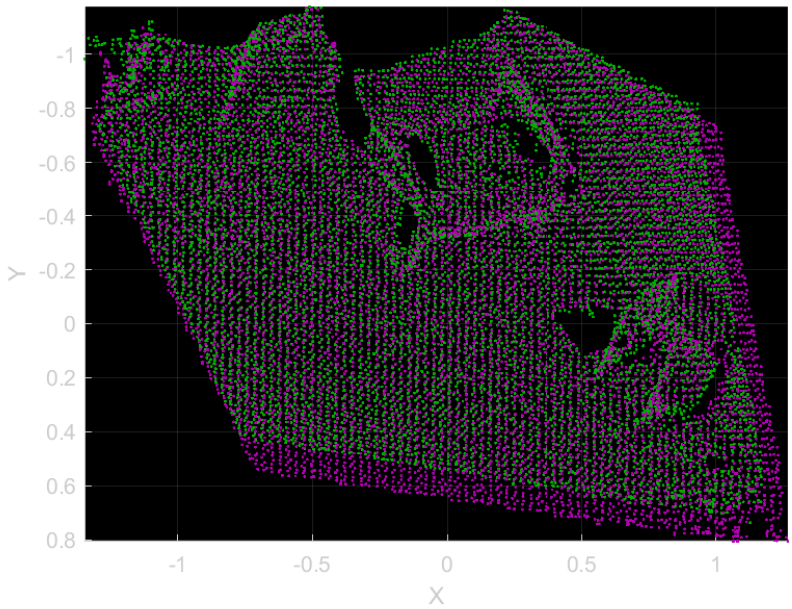

Figure 3.5: *Point cloud of the 2 frames depicting the same scene after applying the estimated transformation to the frame 60 and represented in pairs. Magenta for frame 1 and green for transformed frame 60.*

Finally, using the estimated transformation matrix, the translation vector, and rotational matrix are extracted and converted to a relative pose which is then added as a node to pose graph for further mapping and optimization.

3.4 Sub-map building

Sub-Map building is the process of aligning a small number of frames of a point cloud and merging them together into one point cloud frame. For this project, a total of 3 frames is set as a threshold, meaning that for every 3 successive frames, 1 sub-map will be generated. After building the sub-map, a relative pose is assigned to each sub-map which is usually the pose of the middle frame of the separate successive frames. The sub-map helps in improving computational efficiency during the loop closure detection process.

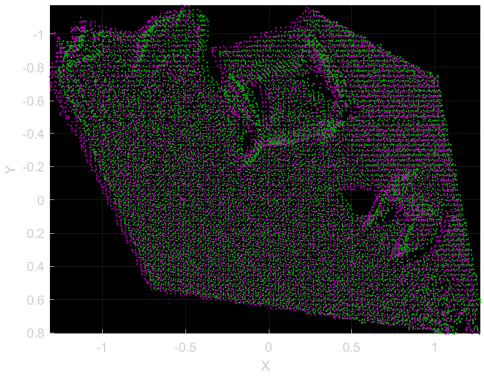

Figure 3.6: *Successive LiDAR frames 1 (magenta) and 45 (green) represented as point cloud pairs.*

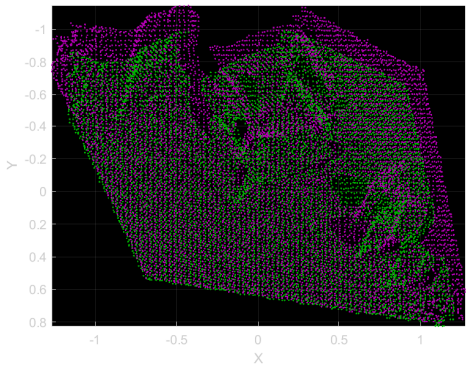

Figure 3.7: *Successive LiDAR frames 45 (magenta) and 89 (green) represented as point cloud pairs.*

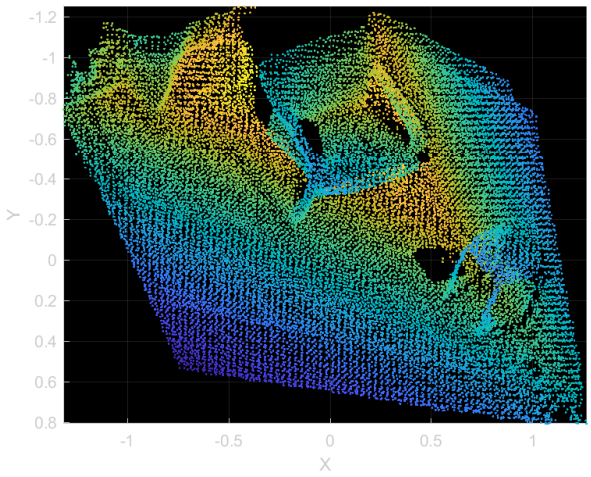

Figure 3.8: *The generated Sub-map consisting of point clouds aligned from the frames 1, 45, and 89*

3.5 Loop closure detection

Loop closure detection is the process of recognizing whether the device is revisiting a previously visited location in the sequence. This is achieved by finding the similarities between the current frame and the previous sub-maps. If similarities are present then the current scan can be included as a loop closure candidate. Sub-maps are used since the necessity of iterating over each previous frame can be overcome which reduces memory requirements, and computational cost. The algorithm of loop closure detection consists of the following:

- Initially, the loop closure candidates are detected by looping through the previously generated sub-maps and matched with the current scan using the ICP algorithm. With the RMSE values less than the set threshold, these sub-map IDs are extracted as the possible candidates. At this stage, all the scans with respect to the sub-map IDs are possible candidates

- From the list of possible candidates (sub-map IDs), loop through the previous scans of each sub-map ID and estimate the pose using the ICP algorithm. The candidate which has the least RMSE value is regarded as the best possible match for the current scan. For refining the process of loop closure detection, the least RMSE value obtained should be less than the set threshold.

- Finally using the pose estimation of the best candidate, the relative pose is extracted and the nodes of the pose graph are updated.

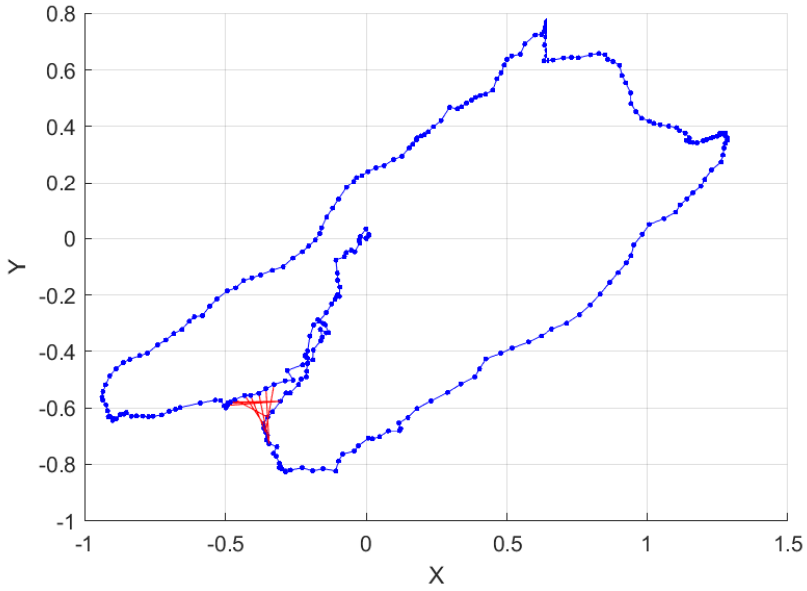

Figure 3.9: *The trajectory (blue nodes) of the device in the indoor environment before optimization with Loop closure detections (red lines connecting nodes)*

3.6 Global map building

After the pose estimation, and loop closure detection stages, the nodes corresponding to the loop closure candidates are optimized which in turn updates the pose nodes of the graph as explained in the section 2.4.4. With the updated nodes (poses) each scan is then transformed and aligned to generate a complete map of the sequence. Hence a global map is the final output of the SLAM algorithm which aligns all accepted scans with respect to the pose estimated and optimized. The figure 3.10 represents the final global map constructed after pose graph optimization of the first random path followed while collecting LiDAR point clouds.

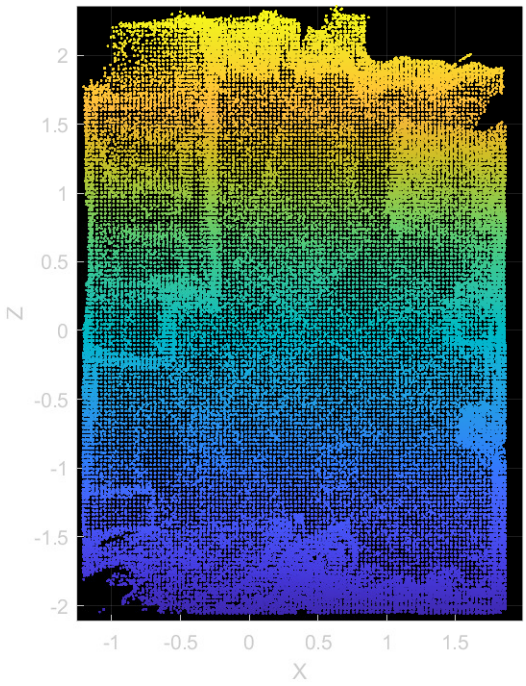

Figure 3.10: *Top view of the optimized global map of the indoor environment. The 3D map has been downsampled from the original point clouds to visualize the figure better.*

Similarly, the described methodology is implemented every time a new set of point clouds are acquired by the user/human with the hand-held L515 LiDAR camera for generating Digital Twins of the environment. For this project, the user repeats the data collection process six times in the same environment by taking a random path each time. All necessary evaluation criteria to access the performance of the L515 sensor are explained in Chapter 4

4
Evaluation

This chapter presents the methods of evaluating the results produced by implementing the method from the previous chapter 3. Various evaluation criteria and metrics are discussed here.

4.1 Accuracy

The term accuracy of measurements can be defined as the closeness of a measurement to its true value. The test for accuracy is crucial to ensure that the gathered information is correct and reasonable. In order to achieve this, the percent error that quantifies the amount of error in a method is calculated as described in the Eq(4.1).

$$\%Error = \frac{Error}{True\ value} \times 100 \tag{4.1}$$

where the term error is calculated as the difference between a measured value and its true value.

Figure 4.1: *A pictorial representation of the concept of Accuracy*

For this thesis work, the test of accuracy is performed on the position of an object in an indoor environment. The object of interest is initially positioned in the given environment and a 3D virtual map is generated. The distance from selected points on the object to various references in the environment such as walls and other objects is measured using the laser distance meter. Hence, these distance serves as the Ground-truth measurements for reference. Further from the generated point cloud map, the same pairs of points are selected and the distance between these two 3D points is estimated. All necessary values of estimated distance, ground-truth distance, errors, and percent errors are presented in the results section 5.

4.2 Repeatability

The term repeatability of a method is defined as the ability of that method to generate similar results for successive measurements of the same sample under identical conditions. To establish the test of repeatability of a method, the following conditions must be met:

- measurements obtained using the same procedure,
- measurements obtained from the same environment,
- measurements obtained with the same instrument or sensor,
- measurements obtained within short intervals of time,
- measurements processed using the same algorithm with the same conditions.

The test for repeatability is necessary in determining whether the method is reliable when performing the procedure multiple times. This is achieved by calculating the variation that occurs during each measurement, using the metrics mean distance and Standard Deviation.

Figure 4.2: *A pictorial representation of the concept of Repeatability*

For this project, the repeatability of generating 3D maps in the point cloud format is to be verified. Hence, the difference between two point clouds is measured by utilizing the Cloud-to-Cloud Absolute Distance (C2C distance) tool from the software CloudCompare. The documentation of the software is available on the company's website [68]. The process begins with importing two point clouds, aligning them, and computing the C2C distance. This generates a histogram illustrating the distribution of the absolute distance. It is also possible to extract the overall Mean Distance and Standard Deviation between the clouds.

A total of six scans of the environment are gathered and processed into 3D virtual maps. Among the six scans, the first scan is fixed as the reference and the repeatability test is verified by comparing the remaining five scans with the reference scan. All necessary histograms, values of mean distance, and standard deviation are presented in the results section 5.

5

Results

In this chapter, the results obtained by implementing the evaluation methods on the generated virtual map are presented. Further, the results are analyzed based on the evaluation metrics discussed in the previous chapter.

5.1 Accuracy

As described in the evaluation section, the test for accuracy is executed to verify whether the generated virtual map is accurate so it can be utilized in VC. The results of the test are presented through the figure 5.1 and the table 5.1.

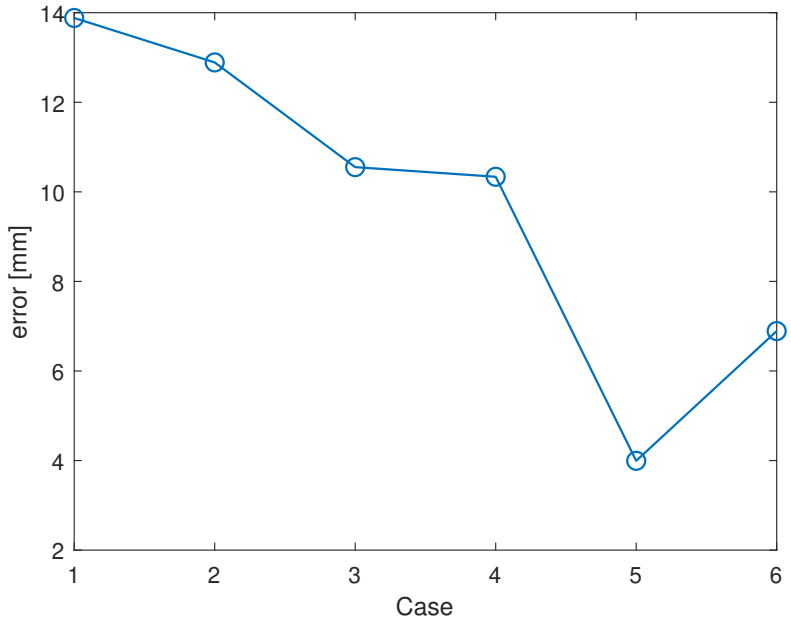

Figure 5.1: *Plot representing the absolute errors (in mm) of measuring the distance between two selected points on the virtual map with respect to the ground truth for each case 1 to 6 as presented in the table 5.1*

The figure 5.1 represents the error plot of distance measured between two selected points in 3D space which follows the order as presented in the table 5.1. It can be observed that the distance errors vary from a minimum of 4 mm to a maximum of 14mm.

S.No	Case	Estimated Distance (m)	Ground Truth Distance (m)	Error (mm)	Accuracy (%Error)
1	Wall-to-wall	3.0529	3.0390	13.8824	0.4568
2	Object-to-Bookshelf 1	2.7081	2.7210	12.8897	0.4737
3	Object-to-Bookshelf 2	2.7024	2.7130	10.5517	0.3889
4	Object-to-Wall	1.5623	1.5520	10.3368	0.6660
5	Object-to-Wall 1	1.0090	1.0050	3.9945	0.3975
6	Object-to-Wall 2	1.0041	1.0110	6.8912	0.6816

Table 5.1: *Table representing the percent error of each case of distance measurements of one scan related to ground truth. The ground truth was measured with a laser distance meter.*

The table 5.1 represents six cases with point-to-point estimated distance measurements of one scan and their respective ground truth distances. The errors and the percent errors are calculated as explained in the evaluation section. The percent error helps to determine how large the errors are. The closer the value of % error to 0, the better the accuracy of the method. It is evident from the table that the maximum value of % error is approximately 0.68% which is still small and close to 0. On average the % error of the distance measurements is approximately 0.5%. With these inferences, it can be concluded that the generated map is fairly accurate. The cause of small errors is further analyzed and explained in Chapter 6.

5.2 Repeatability

As described in the evaluation section, the test for repeatability to verify whether the algorithm performs similarly every time the procedure is repeated under identical conditions is executed. The results of the test are presented through the figures 5.2 to 5.3 and the table 5.2.

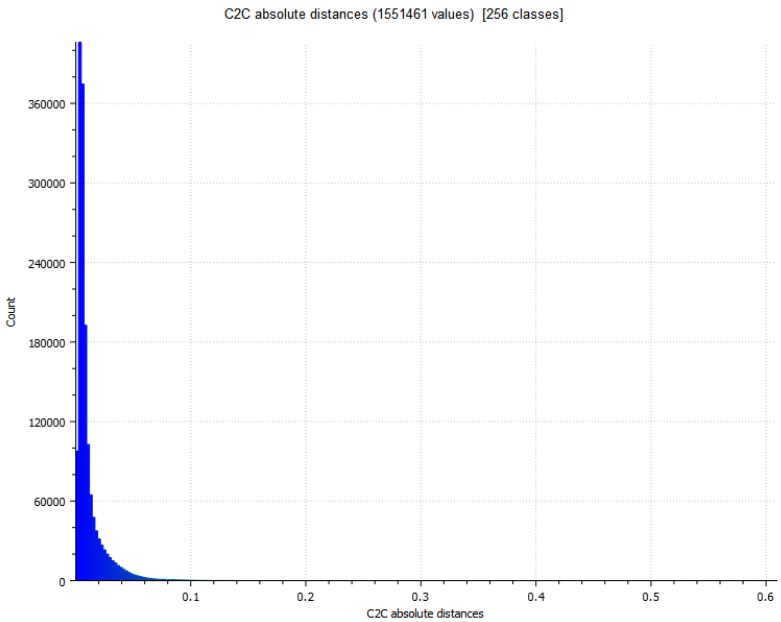

Figure 5.2: *Histogram illustrating the distribution of the cloud-to-cloud absolute distances between the reference scan and the first scan.*

The figure 5.2 depicts the histogram of the C2C distance between the reference scan and the first scan. The histograms of the remaining 4 scans are included in the Appendix section images A.1 to A.4 respectively. It can be observed that all the histograms are skewed such that a maximum number of points have a deviation in the range of 0 to 0.01 (m) from the reference scan.

Scan	Mean Distance(m)	Standard Deviation(m)
1	0.010902	0.012814
2	0.011280	0.013573
3	0.012860	0.013270
4	0.012882	0.014215
5	0.011089	0.013127

Table 5.2: *Mean Distance and Standard deviation between the reference scan and other scans*

The table 5.2 provides the values of Mean Distance and Standard deviation between the reference scan and the other scans. From the table, it can be observed that the mean and standard deviation lie close to 0 meaning that the scans are fairly accurate relative to each other and the error is within the expected range. It is also evident from this table that the mean and Standard deviation values between the scans lie very close to each other which establishes repeatability.

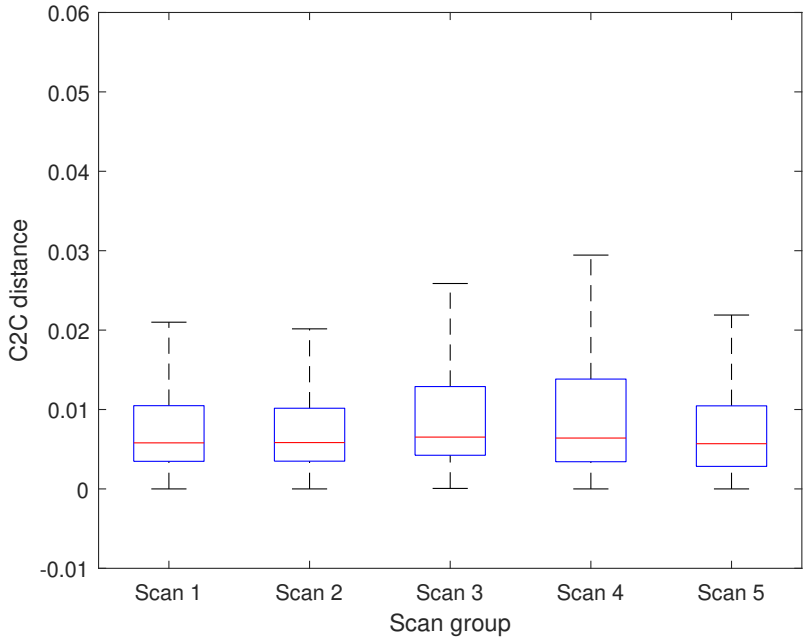

Figure 5.3: *Boxplot representation of the C2C distance between each scan w.r.t the reference scan.*

Comparing multiple histograms at once can be hard, and to resolve this a boxplot can be used. Each box in the plot denotes the interquartile range (IQR) representing the 25% quartile and 75% quartile of the distribution. Hence, this box holds the middle 50% of the entire distribution and the line inside the box is the median of the distribution. The lines extending outside the box are the whiskers with minimum and maximum values.

Figure 5.3 represents the box plot of the C2C distance between the reference scan and the other scans. Analysis can be made by starting with the boxes, it can be observed that in the distribution of each scan, the middle 50% of the distances overlap in every other scan, meaning that there is very little difference. Further, by observing the median lines of each scan, it can be seen that it lies within the IQR of every other scan. With all these inferences, a conclusion can be drawn that the camera produces repeatable results.

6

Discussion and Future work

In this section, various implications drawn from the results are analyzed and discussed along with some possible future works are presented.

6.1 Discussion

From Chapter 5, table 5.1 we encountered small errors when executing the test for accuracy. One reason for the cause of these small errors might have been the effect of downsampling the LiDAR point clouds at the pre-processing stage of the methodology. The downsampling was implemented in the first place since the raw point cloud contains around 76000 points and processing such high resolutions drastically increases the computation cost. Even though the downsampling can be skipped for generating a high-resolution map by using a high-configuration system, it defies the scope of this thesis which focuses on creating a digital twin from an operator's perspective in a low-cost system. Hence the computer used for evaluating the methodology is a cost-efficient, low-configuration system. The other reasons that might affect the accuracy of mapping can be caused by black objects in the environment where the LiDAR rays are absorbed by the object, semi-transparent objects, and highly reflective surfaces like mirrors where the partially reflected LiDAR rays are distorted.

The use of a low-configuration computer for fetching the 3D LiDAR data from the L515 sensor also hinders the highest operating mode of the sensor itself where the L515 device can acquire LiDAR data with a depth resolution of about 1027×768. Out of the three operating modes of the L515 LiDAR Camera as described in Table 3.1 the lowest operating mode is chosen where the depth resolution is only 320×240. Thus by using the lowest operating mode, the computation cost for processing the LiDAR data is greatly reduced.

The consequence of using the lowest operating mode of the L515 LiDAR Camera and down-sampling the generated point clouds has a direct effect on small-sized objects and some details in the larger objects. For example, larger objects like furniture are retained but details of carving in the furniture and small-sized objects like a USB Stick are missed.

From figure 3.10, it can be observed that there is a slight skew in the virtual map which is a common error of SLAM. The skewness of the map might have occurred due to sudden motions while collecting data, and very few points in certain frames of the scans which directly affect the pose estimation of the LiDAR. It is also possible that the black objects and semi-transparent objects cause disturbances during the pose estimation stage. In the future, an algorithm for de-skewing can be implemented by integrating SLAM with IMU measurements that can help initialize the pose estimations and improve them. It is to be noted that this skew can also affect the accuracy of the generated map. Further, it is also evident from Chapter 5 that the algorithm is repeatable and can produce reliable virtual maps when repeating the methodology and hence errors of the same range are expected to repeat over every generated map of the environment.

6.2 Future Work

For the future, various possible developments and improvements that can upgrade the performance of the method have been presented as follows:

- Further improvements can be made to the algorithm by testing it in various environments and verifying its robustness towards the environment.
- Since the L515 LiDAR camera has an in-built RGB camera, implementing the fusion techniques of the camera and LiDAR can be a possible future work.
- The implemented framework is purely a LiDAR-based SLAM, hence in the future, the IMU measurements can be integrated within the algorithm to initialize and improve the pose estimations and help de-skewing the generated map.
- The same method of implementation except the downsampling stage can be tested in a system with code optimization and/or GPU integration to evaluate the sensor on smaller objects.
- The process of data collection and execution of the SLAM algorithm with pre-processing of the data is executed separately at different times but for the future, this algorithm shall be tried in real-time where every stage of methodology explained in Chapter 3 runs simultaneously for each frame of data that is being collected in real-time.

7
Conclusion

This thesis focuses on evaluating the Intel RealSense L515 LiDAR Camera for its usefulness towards Virtual Commissioning by collecting data from the operator's perspective and implementing a solid-state LiDAR SLAM algorithm that can generate a 3D virtual map of the environment. Apart from the limitations, it has been proven that the algorithm is repeatable in generating a fairly accurate and reliable virtual map of the environment in a low-cost system. However, before it can be concluded that can be utilized in Virtual Commissioning, further improvements and investigation in terms of robustness towards various environments are advised.

Bibliography

[1] R. Drath, P. Weber and N. Mauser, "An evolutionary approach for the industrial introduction of virtual commissioning," 2008 IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 5-8.

[2] Chi G. Lee, Sang C. Park, Survey on the virtual commissioning of manufacturing systems, Journal of Computational Design and Engineering, Volume 1, Issue 3, July 2014, Pages 213–222.

[3] N. Shahim and C. Møller, "Economic justification of Virtual Commissioning in automation industry," 2016 Winter Simulation Conference (WSC), 2016, pp. 2430-2441, doi: 10.1109/WSC.2016.7822282.

[4] J. M. Zheng, K. W. Chan and I. Gibson, "Virtual reality," in IEEE Potentials, vol. 17, no. 2, pp. 20-23, April-May 1998, doi: 10.1109/45.666641.

[5] M. Dahl, A. Albo, J. Eriksson, J. Pettersson, and P. Falkman, "Virtual reality commissioning in production systems preparation," in 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–7, 2017.

[6] Lele, A. Virtual reality and its military utility. J Ambient Intell Human Comput 4, 17–26 (2013). https://doi.org/10.1007/s12652-011-0052-4

[7] Giuseppe Riva.CyberPsychology Behavior.Jun 2002.219-225.http://doi.org/10.1089/109493102760147213

[8] Shervin Shahrdar, Corey Park, and Mehrdad Nojoumian. 2019. Human Trust Measurement Using an Immersive Virtual Reality Autonomous Vehicle Simulator. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society.

[9] Anton Albo, Petter Falkman, (2020). A standardization approach to Virtual Commissioning strategies in complex production environments, Procedia Manufacturing, Volume 51, Pages 1251-1258, ISSN 2351-9789, https://doi.org/10.1016/j.promfg.2020.10.175.

[10] To, Alex Liu, Maican Hazeeq, Muhammad Hairul, Bin Davis, Joseph Lee, Jeannie Hesse, Henrik Nguyen, Hoang. (2021). Drone-based AI and 3D Reconstruction for Digital Twin Augmentation. 10.13140/RG.2.2.24881.71528.

[11] R. Zlot and M. Bosse, "Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine," in The 7th International Conference on Field and Service Robots,Matsushima, Japan, July 2012

[12] Zhuli, Ren Wang, Liguan Bi, Lin. (2019). Robust GICP-Based 3D LiDAR SLAM for Underground Mining Environment. Sensors. 19. 2915. 10.3390/s19132915.

[13] Ra ul Mur-Artal, Juan D. Tard os, "ORB-SLAM2: An Open-Source SLAM System for-Monocular Stereo and RGB-D Cameras", Robotics IEEE Transactions on, vol. 33, no. 5,pp. 1255-1262, 2017.

[14] Han, Youngsoo Lee, Jaejoon Lee, Jungmin Lee, Wonhyuk Lee, Kyungho. (2019). 3D CAD data extraction and conversion for application of augmented/virtual reality to the construction of ships and offshore structures. International Journal of Computer Integrated Manufacturing.

[15] Remondino, Fabio Nocerino, Erica Toschi, Isabella Menna, Fabio. (2017). A CRITICAL REVIEW OF AUTOMATED PHOTOGRAMMETRIC PROCESSING OF LARGE DATASETS. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

[16] Ingman, M.; Virtanen, J.-P.; Vaaja, M.T.; Hyyppä, H. A Comparison of Low-Cost Sensor Systems in Automatic Cloud-Based Indoor 3D Modeling. Remote Sens. 2020, 12, 2624. https://doi.org/10.3390/rs12162624

[17] Zhang, Jixian Lin, Xiangguo. (2016). Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing. International Journal of Image and Data Fusion. 10.1080/19479832.2016.1160960.

[18] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM– 3D mappingoutdoor environments," Journal of Field Robotics, vol. 24, no. 8-9, pp. 699–722, 2007.

[19] Zhang, Ji Singh, Sanjiv. (2014). LOAM : Lidar Odometry and Mapping in real-time.Robotics: Science and Systems Conference (RSS). 109-111.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTIdataset," International Journal of Robotics Research, no. 32, pp. 1229–1235, 2013.

[21] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry andmapping on variable terrain," in 2018 IEEE/RSJ International Conference on IntelligentRobots and Systems (IROS), 2018, pp. 4758–4765

[22] Wang, Han Wang, Chen Chen, Chun-Lin Xie, Lihua. (2021). F-LOAM: Fast LiDAR Odometry And Mapping.

[23] Ellon Mendes, Pierrick Koch, and Simon Lacroix. "ICP-based pose-graph SLAM". In:Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on.IEEE. 2016, pp. 195–200.

[24] U. Wandinger, Introduction to Lidar. Springer, 6 2005. [Online]. Available: https://link.springer.com/chapter/10.1007/0-387-25101-41

[25] R. A. Fowler, "The low down on LIDAR," Earth Observation Magazine, pp. 1–6, 2000.

[26] Blair, J.B., Coyle, D.B., Bufton, J.L., and Harding, D.J. (1994), Optimization of an airborne laser altimeter for remote sensing of vegetation and tree canopies. Proc. IGARSS '94, 2:939-941

[27] Cherian, Thomas Yellapragada, Bhavani Kumar Reddy, B.. (2013). LIDAR for Atmospheric Measurement and Probing. International Journal of Engineering and Technology. 5. 5114-5124.

[28] Vosselman, George; Maas, Hans-Gerd (2012). Airborne and terrestrial laser scanning. Whittles Publishing. ISBN 978-1-904445-87-6.

[29] N. Muhammad and S. Lacroix, "Calibration of a rotating multi-beam lidar," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 5648-5653, doi: 10.1109/IROS.2010.5651382.

[30] B. Zhang, X. Zhang, B. Wei and C. Qi, "A Point Cloud Distortion Removing and Mapping Algorithm based on Lidar and IMU UKF Fusion," 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2019, pp. 966-971, doi: 10.1109/AIM.2019.8868647.

[31] Chen Jingye, Shi Yaocheng. Research progress in solid-state LiDAR[J]. Opto-Electronic Engineering, 2019, 46(7): 190218. doi: 10.12086/oee.2019.190218

[32] S. Doggett. (Jan. 2020). "What are point clouds, and how are they used?" [Online]. Available: https://www.dronegenuity.com/point-clouds/.

[33] T. M. |. (Jan. 2021). "What is point cloud modeling?" [Online]. Available: https://www.takeoffpros.com/2020/07/14/what-is-point-cloud-modeling/.

[34] Izadi, Shahram, et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011.

[35] Emmanuel P. Baltsavias, A comparison between photogrammetry and laser scanning, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 54, Issues 2–3, 1999, 83-94, ISSN 0924-2716.

[36] Smith, Self, Cheesman: Estimating uncertain spatial relationships in robotics

[37] Leonard, Durrant-Whyte: Mobile robot localization by tracking geometric beacons

[38] eter Mountney, Danail Stoyanov, Andrew Davison, and Guang-Zhong Yang. Simultaneous stereoscope localization and soft-tissue mapping for minimal invasive surgery. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, pages 347– 354. Springer, 2006.

[39] Long Chen, Wen Tang, Nigel W. John, Tao Ruan Wan, Jian Jun Zhang, SLAM-based dense surface reconstruction in monocular Minimally Invasive Surgery and its application to Augmented Reality, Computer Methods and Programs in Biomedicine, Volume 158, 2018, Pages 135-146, ISSN 0169-2607.

[40] L. Ruotsalainen, S. Gröhn, M. Kirkko-Jaakkola, L. Chen, R. Guinness and H. Kuusniemi, "Monocular visual SLAM for tactical situational awareness," 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2015, pp. 1-9, doi: 10.1109/IPIN.2015.7346957.

[41] Sahoo, B.; Biglarbegian, M.; Melek, W. Monocular Visual Inertial Direct SLAM with Robust Scale Estimation for Ground Robots/Vehicles. Robotics 2021, 10, 23. https://doi.org/10.3390/robotics10010023

[42] J. Markel, "The SIFT algorithm for fundamental frequency estimation," in IEEE Transactions on Audio and Electroacoustics, vol. 20, no. 5, pp. 367-377, December 1972, doi: 10.1109/TAU.1972.1162410.

[43] Z. Huijuan and H. Qiong, "Fast image matching based-on improved SURF algorithm," 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, pp. 1460-1463, doi: 10.1109/ICECC.2011.6066546.

[44] M. Wang, S. Niu and X. Yang, "A novel panoramic image stitching algorithm based on ORB," 2017 International Conference on Applied System Innovation (ICASI), 2017, pp. 818-821, doi: 10.1109/ICASI.2017.7988559.

[45] Taketomi, T., Uchiyama, H. Ikeda, S. Visual SLAM algorithms: a survey from 2010 to 2016. IPSJ T Comput Vis Appl 9, 16 (2017).

[46] J Fischler, Martin; Bolles, Robert (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". Communications of the ACM. 24 (6): 381–395.

[47] S. Orts-Escolano, V. Morell, J. García-Rodríguez and M. Cazorla, "Point cloud data filtering and downsampling using growing neural gas," The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-8, doi: 10.1109/IJCNN.2013.6706719.

[48] Besl, Paul J., N. D. McKay. “A Method for Registration of 3-D Shapes.” IEEE Transactions on Pattern Analysis and Machine Intelligence. Los Alamitos, CA: IEEE Computer Society. Vol. 14, Issue 2, 1992, pp. 239-256.

[49] Wang, Chen Jiang, Yuxi Wang, Manning. (2020). Fast correspondence-based point cloud registration by pair-wise inlier checking and transformation decomposition. Pattern Recognition Letters. 135. 10.1016/j.patrec.2020.05.013.

[50] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "A Tutorial on Graph-Based SLAM," in IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31-43, winter 2010, doi: 10.1109/MITS.2010.939925.

[51] Ahn, SungHwan, Minyong Choi, Jinwoo Choi and W. Chung. “Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment.” 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (2006): 2588-2594.

[52] Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. (2002). "FastSLAM: A factored solution to the simultaneous localization and mapping problem" (PDF). Proceedings of the AAAI National Conference on Artificial Intelligence. pp. 593–598.

[53] Grisetti, G., R. Kummerle, C. Stachniss, and W. Burgard. "A Tutorial on Graph-Based SLAM." IEEE Intelligent Transportation Systems Magazine. Vol. 2, No. 4, 2010, pp. 31–43. doi:10.1109/mits.2010.939925.

[54] .W. Eggert, a. Lorusso, and R.B. Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. Machine Vision and Applications, 9(5-6):272–290, 1997.

[55] W. Xin and J. Pu, "An Improved ICP Algorithm for Point Cloud Registration," 2010 International Conference on Computational and Information Sciences, 2010, pp. 565-568, doi: 10.1109/ICCIS.2010.144.

[56] Z. Zhang. ”Iterative Point Matching for Registration of Free-Form Curves,” IRA Rapports de Recherche, Programme 4: Robotique, Image et Vision, no. 1658, 1992.

[57] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. Proc. of Robotics: Science and Systems, 2:4, 2009.

[58] Soon-Yong Park, Murali Subbarao, An accurate and fast point-to-plane registration technique, Pattern Recognition Letters, Volume 24, Issue 16, 2003, Pages 2967-2976, ISSN 0167-8655.

[59] Radu Bogdan Rusu. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. KI - Künstliche Intelligenz, pages 1–4, 2010.

[60] Chris Harris and Mike Stephens (1988). "A Combined Corner and Edge Detector". Alvey Vision Conference. 15.

[61] Lowe, David G. (1999). "Object recognition from local scale-invariant features" (PDF). Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.

[62] Rosten, Edward; Drummond, Tom (2006). "Machine Learning for High-speed Corner Detection". Computer Vision – ECCV 2006. Lecture Notes in Computer Science. 3951. pp. 430–443. doi:10.1007/1174402334. ISBN 978-3-540-33832-1. S2CID 1388140.

[63] G. Flitton, T. P. Breckon and N. Megherbi, "Object Recognition using 3D SIFT in Complex CT Volumes", BMVC, pp. 11.1-11.12, 2010.

[64] Rusu, Radu Bogdan, Nico Blodow, and Michael Beetz. "Fast point feature histograms (FPFH) for 3D registration." In 2009 IEEE International Conference on Robotics and Automation, pp. 3212-3217. IEEE, 2009.

[65] R Hänch, T Weber, and O Hellwich. Comparison of 3D Interest Point Detectors and Descriptors for Point Cloud Fusion. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3, 2014.

[66] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 796–803, 2017.

[67] P.C.Lib.(Jan.2021).“Downsampling a pointcloud using a voxelgrid filter,” [Online]. Available: https://adioshun.gitbooks.io/pcl/content/Tutorial/Filtering/pcl-cpp-downsampling-a-pointcloud-using-a-voxelgrid-filter.html

[68] CloudCompare software documentation (C2C Distance computation), https://www.cloudcompare.org/doc/wiki/index.php/Distances_Computation#Cloud-cloud_distances

A

Appendix 1

A.1 Distribution of the cloud-to-cloud absolute distances

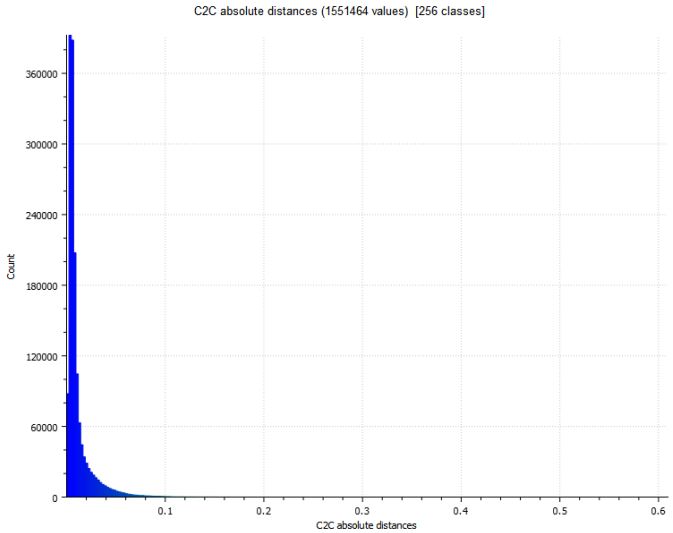

Figure A.1: *Histogram illustrating the distribution of the cloud-to-cloud absolute distances between the reference scan and the second scan.*

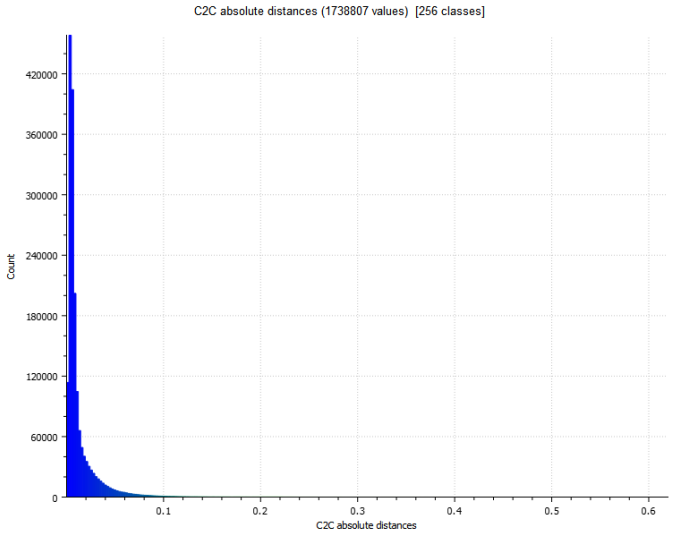

Figure A.2: *Histogram illustrating the distribution of the cloud-to-cloud absolute distances between the reference scan and the third scan.*

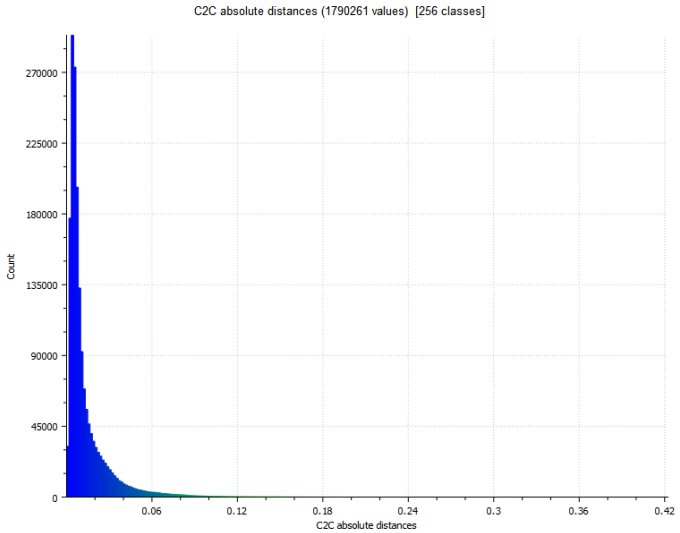

Figure A.3: *Histogram illustrating the distribution of the cloud-to-cloud absolute distances between the reference scan and the fourth scan.*

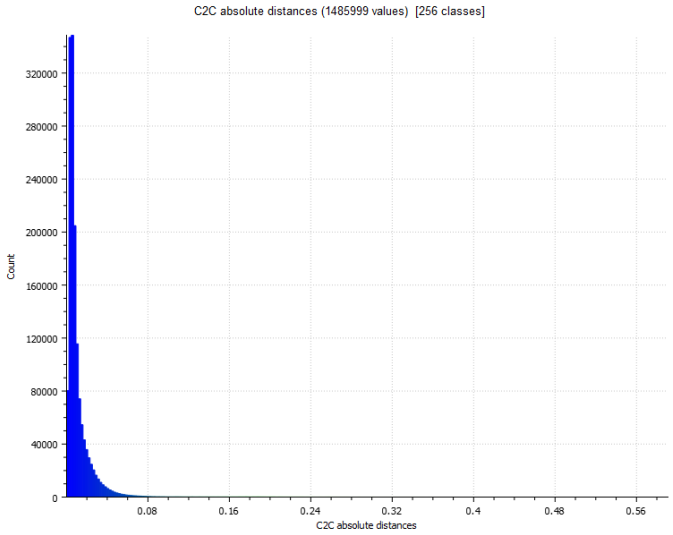

Figure A.4: *Histogram illustrating the distribution of the cloud-to-cloud absolute distances between the reference scan and the fifth scan.*

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY