

DEVELOPMENT OF A VOXEL BASED LOCAL PLANE FITTING FOR MULTI-SCALE REGISTRATION OF SEQUENTIAL MLS POINT CLOUDS

L. Moradi ¹, M. Saadatesresht ¹, P. Shokrzadeh ²

¹ School of Surveying and Geospatial Engineering, University of Tehran, Iran - (loghman.moradi, msaadat)@ut.ac.ir

² Dept. of Geomatic Engineering, Istanbul technical University, Istanbul, Turkey - p.shokrzadeh@gmail.com

Commission IV, WG IV/3

KEY WORDS: SLAM, Point Cloud Registration, Indoor Mapping, Sequential Scan Matching, Voxel-based Registration, MLS

ABSTRACT:

The Mobile Laser Scanner (MLS) system is one of the most accurate and fastest data acquisition systems for indoor and outdoor environments mapping. Today, to use this system in an indoor environment where it is impossible to capture GNSS data, Simultaneous Localization and Mapping (SLAM) is used. Most SLAM research has used probabilistic approaches to determine the sensor position and create a map, which leads to drift error in the final result due to their uncertainty. In addition, most SLAM methods give less importance to geometry and mapping concepts. This research aims to solve the SLAM problem by considering the adjustment concepts in mapping and geometrical principles of the environment and proposing an algorithm for reducing drift. For this purpose, a model-based registration is suggested. Correspondence points fall in the same voxel by voxelization, and the registration process is done using a plane model. In this research, two pyramid and simple registration methods are proposed. The results show that the simple registration algorithm is more efficient than the pyramid when the distance between sequential scans is not large otherwise, the pyramid registration is used. In the evaluation, by using simulated data in both pyramid and simple methods, 96.9% and 97.6% accuracy were obtained, respectively. The final test compares the proposed method with a SLAM method and ICP algorithm, which are described further.

1. INTRODUCTION

Mobile Laser Scanner (MLS) systems are capable of acquiring the most accurate and fastest data for indoor and outdoor environments mapping. However, knowing the location of the system at the time of acquisition is necessary. There are a number of popular methods for localization, including dead reckoning, Global Positioning System (GPS), localization using a prior map, and Simultaneous Localization and Mapping (SLAM). SLAM is one of the most functional method.

As part of SLAM, MLS data is used to compute sensor location at the same time to map an unknown environment. Identifying where the sensor is the task of localization, while creating a description of the environment is mapping. By matching the observations with an existing map, the localization problem can be solved. However, it is commonly assumed that the sensor will operate in an unknown environment, or that a detailed map of the environment is generally not available. Therefore, it is crucial to create a map of the environment based on sensor observations. Another issue arises in this context because creating a map requires the sensor location. As a result, it is also referred to as "chicken and egg" in scientific publications. Since a map is needed for localization and also sensor location for mapping, localization and mapping problems are solved simultaneously to create an environment map.

The SLAM problem was first addressed at the IEEE Conference in 1986 (Smith & Cheeseman, 1986), which used a probabilistic approach to solve it. Since then, the SLAM problem has become an attractive topic for robotics and computer researchers. Most SLAM methods give less importance to geometry and mapping concepts. Therefore, this research aims to solve the SLAM

problem by considering the adjustment concepts in mapping and geometrical principles of the environment and proposing an algorithm for reducing drift. To achieve this, a model-based registration is proposed. In addition, during the environment observation, the amount of collected data is gradually increased. Since mobile sensors are limited in load capacity and memory, the goal is to reduce the amount of data used in the SLAM by building a model of the environment. Moreover, this model can be used for navigation and other purposes.

2. RELATED WORK

From 1981 to early 1991, navigation and mapping were two separate issues. The mapping was broadly divided into two categories: metric (topological) or a geometric description of surrounding (Chatila & Laumond, 1985). The localization algorithms were solved by the Kalman filters (Durrant-Whyte & Leonard, 1991), but the combination of mapping and localization areas led to the birth and development of SLAM. The origin of SLAM dates back to 1986 (Smith & Cheeseman, 1986). One of the conventional SLAM solutions is to use fine registration and scan matching methods. Much of the work on laser scan matching has been presented. In addition, a few relevant and recent research, including feature-based algorithms, the Iterative closest point (ICP) family (J. Besl & D. McKay, 1992), and some new methods, have been reviewed. The first category is iterative approximation methods, which mainly refers to the ICP and a set of refined algorithms (Eggert et al., 1998)(Bucksch & Khoshelham, 2013). The advantage of this method is that it solves a rigid object transformation through a precise mathematical process with no initial estimation of location. By identifying the correspondence points and decreasing the distance between the two scans, ICP is used to estimate the translation and rotation parameters that best align two scans. One

of the best ICP algorithms is Generalized-ICP (A Segal, D Haehnel, 2009), which combines the ICP and point-to-plane ICP algorithms into a probabilistic framework. Li and et al. designed a feature-based matching framework, in which extended 1D SIFT is used to extract salient and distinct feature point, and Split and Merge for line feature extraction (Li et al., 2016). Consequently, distance histogram is used for feature descriptions, and the pairs, which achieve the best matching score, are selected as potential correspondence. Finally, the relative position and orientation parameters are calculated for registration process. Ryu and et al. proposed a grid-based scan-to-map matching technique. The environment is represented as a grid map with multiple Normal Distribution (ND) in each cell (Ryu et al., 2016) and a new scan is matched with it. Park and et al. utilized a topological hybrid map for global localization (Park & Park, 2014). It uses a coarse-to-fine registration approach. In coarse registration step, by 2D geometric histogram-based template matching, candidate point is selected. Then, the SSM method is used to find the correct points, and relative pose is estimated. Using a scan-to-map algorithm, Qian et al. proposed SLAM in an indoor environment (Qian et al., 2019). As a map representation method, it used a grid-based orthogonal feature weighted occupancy likelihood map (OWOLM). Afterwards, scan-to-map matching is used to determine the transformation parameter. An FFT-based scan matching method was developed by Jiang et al. (Jiang et al., 2018). Transform parameters are determined by applying FFT to the image. The match is then determined using ICP. Based on reference key frames (RKF) involving feature-to-feature and point-to-point approaches, Mohamed et al. proposed a method (Mohamed et al., 2017). When there are no linear features, the algorithm switches to the ICP, and then back to the RKF once linear features are found. In (Biber, 2003), the Normal Distribution Transformation (NDT) algorithm subdivided the 2D workspace into cells of constant size. Afterwards, each cell's normal distribution is calculated. By building the NDT, two scans will align, and source points will map to the coordinate reference frame. The optimal parameter is obtained using Newton's optimization algorithm for each mapped point and the best parameters will be found. In (Lu et al., 2018), voxelization was used to downsample the point cloud, and a Gaussian mixture model was used to calculate the probability of the negative likelihood function. The rotation and translation parameters were then solved with an EM algorithm. In (Konecny et al., 2019), cross-correlation and differential evolution are used to align two scans. The cross-correlation is used between two scans as an efficient measure of scan alignment, and the differential evolution algorithm is used to look for transformation parameters that align scans.

The RANSAC method was proposed by Fischler and Bolles (Fischler & Bolles, 1981) and has been widely used in 2D and 3D data processing. Additionally, they have been researched for application in image registration (Tarsha-Kurdi et al., 2008)(Xu et al., 2016)(Fontanelli et al., 2007)(Weinmann et al., 2011). Also, as LiDAR technology has advanced, RANSAC techniques have been applied in numerous studies for the pre-processing and segmentation of point cloud data (Al-Durgham et al., 2013)(Biber, 2003)(Takeuchi & Tsubouchi, 2006). As a result, using this technique for point cloud registration has become a crucial study area. (Fontanelli et al., 2007). The RANSAC method consists of three steps. First, the conversion relationship is computed using a number of control points that are randomly chosen. The conversion is used to remove external points of the point cloud in the second step, after which the point cloud's degree of registration is determined. Finally, finding the data set with the highest degree of registration is done using iterative

translation, and the translation parameters are calculated (Chen & Hung, 1999).

3. METHODOLOGY

Using MLS is a popular method for creating a map of the environment. Most prominent characteristic feature of point clouds captured by these sensors is being sequentially, which every scan has rotated and transitioned with regard to the previous one. Consequently, to create an integrated environment map, all the collected data must be transferred into a determined coordinate system. For this purpose, a Local Coordinate System (LCS) of the point cloud should be determined and simultaneously build the integrated map. These processes are referred to SLAM. A method to transfer these sequential scans to the LCS is registration. The majority of new SLAM algorithms use correspondence points matching methods for registration. In these methods, sequential scans are matched by finding the corresponding points. These methods, however, is highly dependent on the accuracy of point matching algorithms. Finding correspondence points in rough point clouds is a challenging and complex process. Thus, to avoid point matching, a voxel-based registration technique to register sequential scans is proposed. The short distance between sequential scans is a noticeable feature in MLS point clouds. Consequently, by voxelization of two point clouds, most of the correspondence points will be placed in the same voxel, and finding the exact correspondence points is not required. Then, points within the same voxel are used for registration. Additionally, the ultimate goal of all SLAM algorithms is to create an environment map, both online and offline. In this method, besides the final and integrated map of the environment, which is created offline, a differential model of the environment is created online that can be used in various navigation areas.

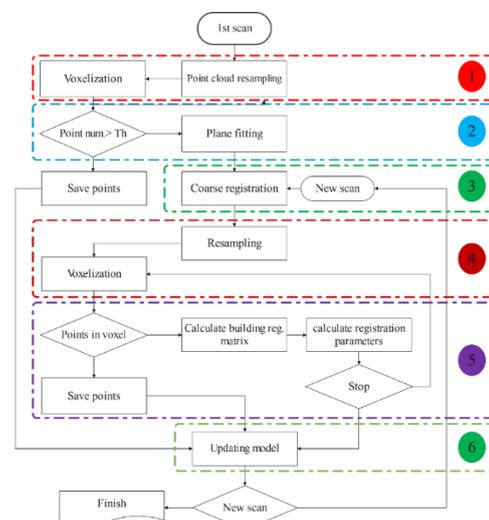


Figure 1. The overall flowchart of the proposed method.

Another problem with SLAM systems is the drift error during the data acquisition and registration process that needs to be fixed. The main reason of this problem is not using all previous information in registration process. Since in all methods, all scans is transferred to the first scan coordinate system as a LCS, the first scan is considered error-free; then, when a new scan is taken, its position is determined using the previous scan. Assuming that the new scans' registration parameters are error-prone, the error will increase as the number of scans increases. In most cases, these errors are minimized by loop closure

techniques. Here in this research, drift error is minimized by registering with a gradually updated model of the environment that uses information from all previous scans. By doing so, drift errors will be reduced significantly without any additional computation and complexity. The proposed algorithm involves the following steps:

- 1) Create a pyramid of source point clouds and voxelization
- 2) Creating the differential model of the environment: fitting the plane to points inside voxel in the reference point cloud
- 3) Coarse registration
- 4) Create the pyramid and voxelization of target point cloud
- 5) Fine registration
- 6) Update differential model

3.1 Create a pyramid of source point clouds and voxelization (initial scan)

In this study, a voxelization method is used instead of matching methods. Since the distance between the corresponding points of two sequential scans is small, therefore, if the voxel size were larger than the average distance between two point clouds and their density, the corresponding points will typically be placed in the same voxel. By using this method, it is not required to determine the corresponding points. Figure 1 shows an example of a point cloud voxelization.

Another challenge with this type of data is that the distance between the source and the target point cloud increases as the scanning time of the environment increases. This challenge is solved with two different approaches. First, the registration process is divided into coarse and fine registrations. As well, to overcome this challenge (in case of necessity) a pyramid of the point clouds is built during the fine registration step. Since the voxelization method is used instead of matching method, so the average distance between two-point clouds should be less than the voxel dimensions; however, in some cases, the distance between two point clouds is more than the dimensions of the desired voxel after coarse registration, so it is necessary to use pyramidal registration.

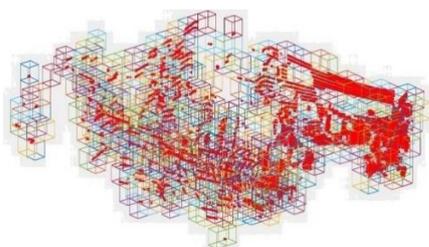


Figure 2. Point cloud voxelization.

With pyramid point cloud registration, we can voxelize the points cloud with different and larger dimensions for each pyramid level and registration is done in hierarchy mode. Therefore, by registering point clouds at the bottom of the pyramid and applying these parameters to the top level, the point clouds come close to each other, gradually. By the end of the pyramid, the point clouds have been registered precisely. Details are given in the fine registration section.

In order to build a pyramid of point clouds, data resampling with different resolution is done. The first step is to determine the number of pyramid layers and the relationship between the voxel dimensions in these layers, which is accomplished using equation (1).

$$K = \log_n \left(\frac{1}{d} \right) + 1 \quad (1)$$

As can be seen from equation (1), the number of pyramid layers (K) depend on the desired percentage of points remaining at the bottom layer of the pyramid (d) and the ratio between the dimensions of the two successive pyramid layers (n).

The next step is resampling of the original point cloud with different resolution to construct each layer of pyramid. By doubling the actual point spacing of a point cloud, we determine the resolution of the resampling and the first layer of pyramid is constructed. For the remaining layers, the resolution in each layer is reduced to half and the pyramid point clouds are generated one by one for each layer. In addition, the final layer in the pyramid is the original point clouds.

$$Point\ Spacing = \sqrt{\frac{1}{Point\ Density}} \quad (2)$$

The next step is voxelization. The proposed algorithm has a challenge concerning voxel sizes, which depend on different factors such as density of the point cloud, range picked up by the sensor, and distance between sequential scans. By trial and error, the best size is determined.

3.2 Creating the differential model of the environment

In this research as a substitute for correspondence points matching, point to model matching is used. Therefore, a model is fitted to points inside each voxel of the source point cloud and utilized in the fine registration step. Given that, the voxel dimensions at each level are small and the number of points inside these voxels is low. As a result, the normal vectors are not subject to much variation, so a planar model can be used to model the environment. By constructing planar models in the source point clouds at each voxel and transferring the target points present in the same voxel to this plane, registration parameters can be calculated. For plane fitting, the RANSAC method is used. To avoid fitting the plane to all voxels and to increase algorithm accuracy, a threshold should be set for the minimum number of points in each voxel to fit the model. This threshold value is derived from equation (3), which s_i represents the number of points in each voxel and n is the number of voxels.

$$T = \frac{1}{n} (\sum_{i=1}^n s_i) \quad \& \; s_i > 3 \quad (3)$$

It is also necessary to identify incorrectly fitted planes for improving registration accuracy. The RANSAC algorithm calculates the plane fitting error, and the sigma test identifies and removes inaccurate planes. In figure 3, an example of histogram of the plane fitting errors is shown for each voxel.

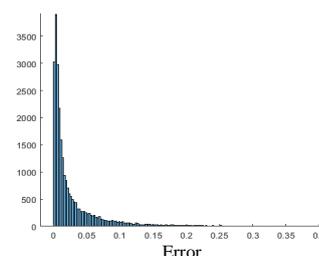


Figure 3. Histogram of plane fitting error.

3.3 Coarse Registration

Since the average distance between the source and target scans increases with increasing data acquisition time, the corresponding points will not be placed in the same voxel; therefore, a coarse registration step is necessary before fine registration to reduce this distance. In this case, a linear model based on the aggregation of the fine and coarse registration parameters of all previous scans is used for coarse registration. Finally, the computed coarse registration is applied to the new scan.

$$R_{coarse}^i = R_{coarse}^{i-1} + R_{fine}^{i-1} \quad (4)$$

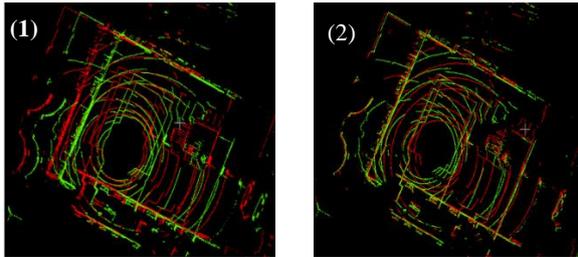


Figure 4. Situation of two sequential scans before (1) and after (2) coarse registration.

3.4 Create the pyramid and voxelization of the target point cloud

Once the target scan has been located in its approximate position, it will spontaneously be placed in the same voxel. Then it is necessary to build a pyramid of the target scan for fine registration. This step is the same as the first one, which was performed for the source scan.

3.5 Fine Registration

For fine registration, a pyramid-based method is used, where each level of target point clouds is registered with the same level of source point clouds. Thus, at the first level of the pyramid in each voxel, the target point clouds are transferred to the plane of the source points. Afterwards, the parameters obtained from this level are applied to the target point clouds. This cycle continues until the last level of the point cloud pyramid is registered. The following diagram shows the general flow of this step.

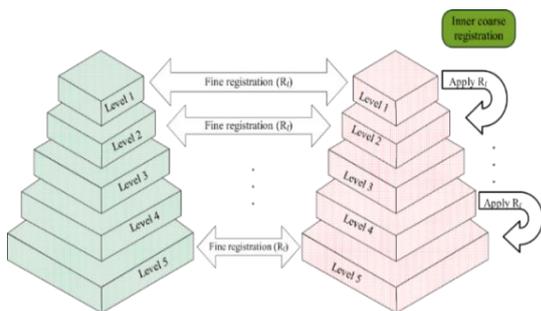


Figure 5. The process of pyramid registration.

Fine registration is accomplished by transferring target point clouds to the source point clouds model created in each voxel. Each voxel has a plane as its base model, so the goal is to transfer the target points to these planes. To determine the transfer parameters, the least-squares adjustment is used. For example, if the equation of the plane were as follows:

$$P : ax + by + cz - d = 0 \quad (5)$$

Which a, b, c, and d are the parameters of the plane. Now, to be able to calculate the registration parameters, the above equation is rewritten as follows:

$$G : [a \ b \ c \ d] * (RX + T) = 0 \quad (6)$$

R and T are three rotation and transition parameters, and X are target points transferred to the planes. After implicit derivation, the following equation is obtained:

$$G : G_0 + \frac{\partial G}{\partial X}(dx) + \dots = 0 \quad (7)$$

A matrix corresponding to all expressions in equation (7) is formed, which is as follows:

$$\frac{\partial G}{\partial X} = \begin{bmatrix} \frac{\partial G_1}{\partial \omega} & \frac{\partial G_1}{\partial \varphi} & \frac{\partial G_1}{\partial \kappa} & \frac{\partial G_1}{\partial Tx} & \frac{\partial G_1}{\partial Ty} & \frac{\partial G_1}{\partial Tz} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial G_n}{\partial \omega} & \frac{\partial G_n}{\partial \varphi} & \frac{\partial G_n}{\partial \kappa} & \frac{\partial G_n}{\partial Tx} & \frac{\partial G_n}{\partial Ty} & \frac{\partial G_n}{\partial Tz} \end{bmatrix} \quad (8)$$

$$dx = \begin{bmatrix} d\omega \\ d\varphi \\ d\kappa \\ dTx \\ dTy \\ dTz \end{bmatrix} \quad (9)$$

Now, according to the above matrices, we come to the following equation:

$$Ax = F \quad (10)$$

Where x is the unknown parameters matrix, A is the design matrix, and F is the residual vector; therefore, using the least-squares method, the unknown parameters are calculated in an iterative process and updated at each stage.

$$x = (A^T * A)^{-1} * (A^T * F) \quad (11)$$

$$\begin{aligned} \omega^{new} &= \omega^{old} - d\omega \\ &\vdots \\ Tz^{new} &= Tz^{old} - dTz \end{aligned} \quad (12)$$

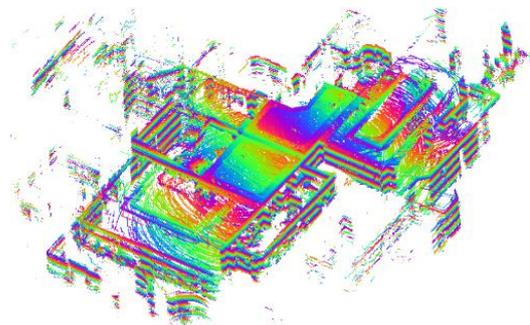


Figure 6. Output map based on proposed method.

Moreover, additional parameters, which are sensor position, are simultaneously computed with the registration process. Figure 7 shows the sensor position during the scanning.

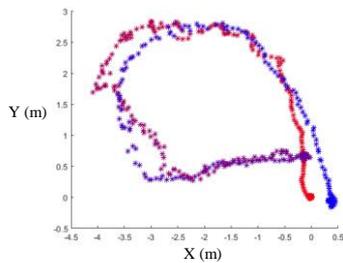


Figure 7. Sensor's location during scanning time.

Besides the pyramid registration method, another approach that we used is simple registration. In the simple method, all steps are similar to those in the pyramid method, but there is no pyramid, and only the original point cloud is used for registration. In comparison to pyramid registration, the main advantage of simple is low computation and less complexity.

3.6 Update differential model

The differential model of the environment is composed of voxels with plane model. It is thus necessary to update and expand the model after each registration step.

It is evident that some points in the source point cloud are not included in building the model and registration, which must be maintained to update the model. Using the remaining points from source point cloud and registered target point cloud, if the model building conditions (minimum number of points) are met, the plane model is fitted in new voxel. Once again, several points in the voxels do not meet the plane fitting conditions, so these points are retained and aggregated with the remaining points from the subsequent scan. This approach has the advantage that no primary source of points cloud needs to be stored, resulting in less storage space required.

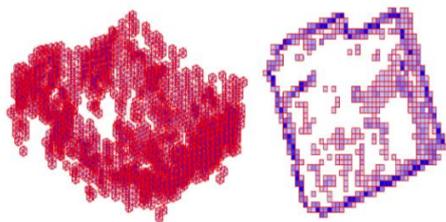


Figure 8. Environment differential model from two different views.

The drift error is one of the main problems with sequential scan registration and SLAM algorithms. This error is mainly caused by using only one previous registered scan in registration. Thus, the information of all previous scans is recorded in the differential model of the environment. Consequently, all previous scan information is used for registration of a new scan, minimizing drift error. An example of a differential model of the environment can be seen in Figure 8.

4. EVALUATION OF THE PROPOSED ALGORITHM

4.1 Experimental data

The proposed algorithm is evaluated using two sets of mobile sensor datasets. In both cases, Velodyne sensors were used to capture the data. A Velodyne HDL-32e sensor was used to take

the first data inside a room. Second data, a Velodyne VLP-16 sensor captured the data of a building.

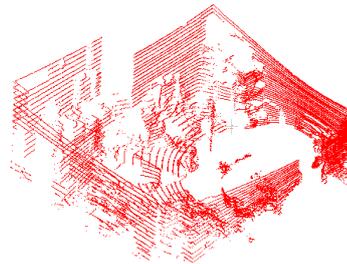


Figure 9. An example of the Velodyne HDL-32e data.

The main differences between the two data sets are the density of points and the environment where the data were collected. Since Velodyne HDL-32e has more lasers than VLP 16 and the environment is smaller, so the density of points in the first data is higher than in the second one.

4.2 Experiments and results

Two experiments are arranged for evaluation and comparison of the proposed algorithm. The first experiment examines the effects of noise on registration results. In the second experiment, the validity of the algorithm is checked using two different methods.

4.2.1 The effect of noise on registration results: Since the data is never noise-free, it is necessary to investigate the effect of noise in the data on the proposed algorithm. Therefore, the simple registration method (not pyramid) is used with optimal parameters calculated for the algorithm. Two scans are registered in the presence and absence of noise. Noise can be related to the data and model, therefore two methods are used to identify and delete noisy planes and data. In the first method, the planes that do not have the appropriate accuracy are removed using the sigma test, and second one, the RANSAC removes the noisy and outlier points. Point-to-point histogram, obtained from cloudCompare software (*CloudCompare*, n.d.), in figure 10 shows that there is no distinct difference between the two registered data. One reason could be the high degree of freedom of the equations, which automatically eliminates noise in the model and the target point cloud. With the results obtained in this experiment, it can be concluded that the algorithm will automatically eliminate the effect of normal noise in the point cloud.

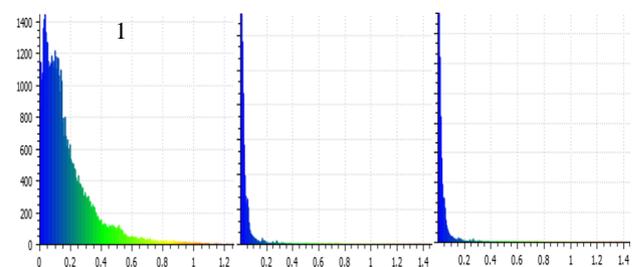


Figure 10. Point-to-Point distance histogram for data: (1) before registration (2) after registration (noisy data) (3) after registration (clear data).

4.2.2 Algorithm validation: We performed two experiments to evaluate the results of the algorithm. In the first experiment, we evaluate the overall accuracy of the algorithm using simulation data. In the second experiment, the algorithm was evaluated with another SLAM algorithm and the ICP. The details of these experiments will be explained below.

4.2.2.1 Evaluate the accuracy of the algorithm using simulated data:

One way to evaluate the results of an algorithm has always been to use simulated data. First, three rotation and translation parameters are applied manually to the source point cloud, and then a random noise, which is higher than the sensor noise, is added to data and considered as the target point cloud. The aim is to register the simulated target point cloud with the source point cloud using two simple and pyramid registration algorithms to recover the applied parameters and compare them with the actual value. In addition, to evaluate the algorithm's accuracy, two point clouds will be registered in ten repetitions, and the results will be reviewed. First, the following rotation parameters (ω, φ, κ) in radian and translation parameters (T_x, T_y, T_z) in meter are applied to the source point cloud:

$$\omega = 0.03, \varphi = -0.03, \kappa = 0.02 \text{ rad}, T_x = 0.03, T_y = 0.04, T_z = -0.02 \text{ m}$$

After registration, the relative accuracy formula is used to evaluate the results in both algorithms as follows:

$$\text{Relative Accu.} = 1 - \left| \frac{\text{True} - \text{Observed}}{\text{True}} \right| \quad (13)$$

In the first experiment, the pyramid registration algorithm is used. First, the data is resampled in five different resolutions, and the pyramid of points cloud is constructed. Then, the points cloud pyramid is voxelized with 0.05, 0.08, 0.15, 0.26, and 0.45 meters and registration is done. Figure 11 shows the relative accuracy of calculating each parameter separately in ten repetitions. The relative accuracy of all parameters is over 93%.

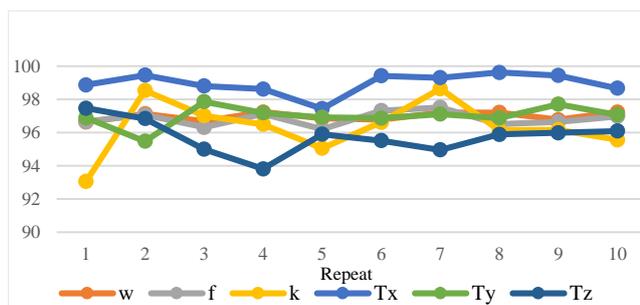


Figure 11. Relative accuracy for each registration parameter (percent).

To evaluate the overall accuracy of the algorithm, we calculated the average accuracy of all six parameters and the average accuracy of registration. According to the figure 12, in each iteration, registration is done with an accuracy of 96%. In addition, since the standard deviation of accuracy in all iterations is very low, it is a reason for the high precision of the algorithm. Therefore, the final average accuracy of this algorithm is 96.96%.

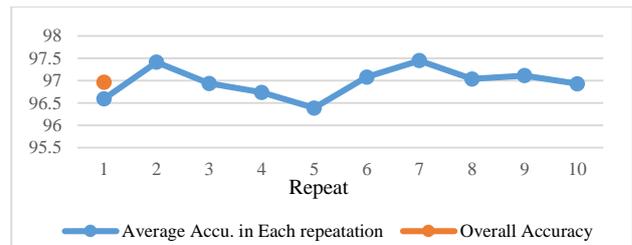


Figure 12. Overall accuracy of calculated registration parameters (percent).

In another assessment, the histogram of the point-to-point distance between two point clouds and the mean distance is examined. Figures 13 and 14 show the histogram before and after the registration. As seen from the histogram, the mean distance between two point clouds before registration is 34 mm, which decrease to 2mm after registration. This 2mm distance could also be due to the random noise applied to the simulated data.

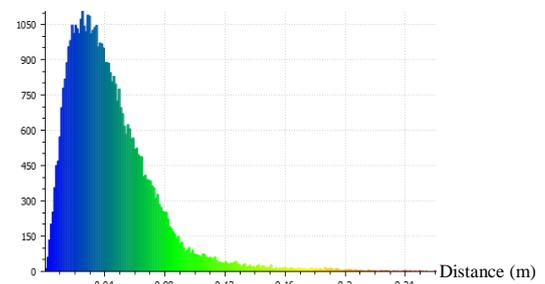


Figure 13. Point-to-point distance histogram before pyramid registration.

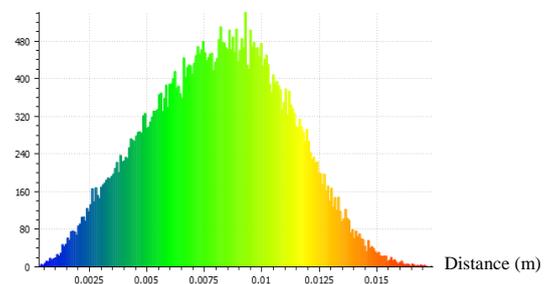


Figure 14. Point-to-Point distance histogram after pyramid registration.

The first experiment is also repeated to evaluate the simple registration algorithm (without building a pyramid). In this algorithm, the point clouds is voxelized one time with constant dimension, and registration is done. Since the distance between two point clouds is small, the voxel dimensions of 10 cm is considered. Figure 15 shows the relative accuracy of each parameter separately in each iteration.

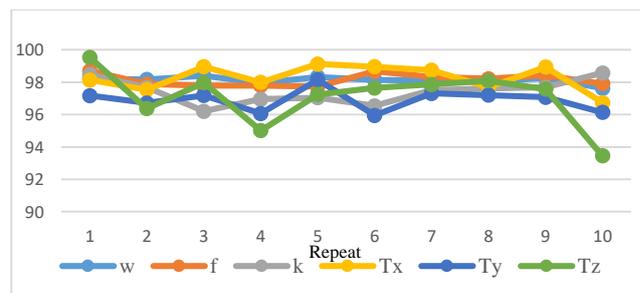


Figure 15. Relative accuracy for each registration parameter (percent).

As shown in figure 15, the relative accuracy of all parameters is above 93%. In addition, the standard deviation of the calculated values is low and shows the high precision of the algorithm. Figure 16 shows the visual results of matching two point clouds before and after registration, respectively.

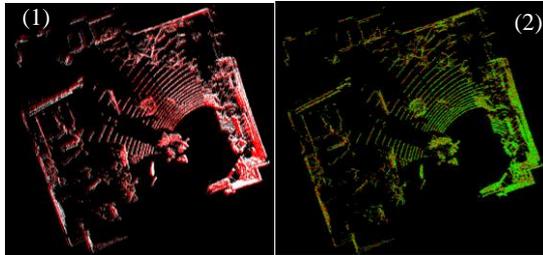


Figure 16. Situation of two sequential scans before (1) and after (2) registration.

In another evaluation of this algorithm, the histogram of the distance between two points cloud is used. Figure 17 is the histogram of the distance between the points after registration. According to these histograms, the mean distance between the points before and after registration is 34 and 2 mm, respectively.

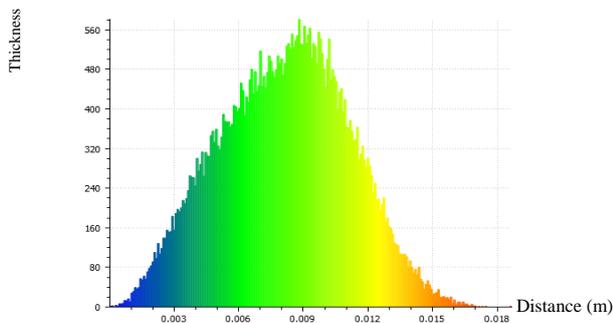


Figure 17. Point-to-point distance histogram after simple registration.

4.2.2.2 Algorithm evaluation using a SLAM and ICP algorithm: The final evaluation is performed using the perpendicular distance between several specific planes. Finally, the proposed algorithm results are compared with an ICP and a SLAM method (Shokrzadeh, 2019).

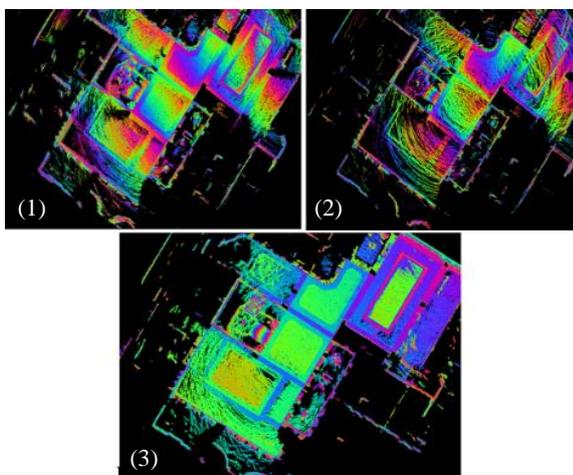


Figure 118. Output map of (1) the ICP (2) proposed method (3) Shokrzadeh method.

At first, based on the visual comparison shown in Figure 18, the output of the performance of proposed method has state of art result with the compare ones. In the second evaluation, the thickness test, 16 flat pieces in all three directions were selected from registered data, and the thickness test was performed. The results of which are as shown in figure 19.

In this experiment, several cross-sectional flat pieces were manually selected from the output of all three methods, and a plane is fitted to each of them. Then by using normal vector of each point, the perpendicular distance between two-fronted planes is calculated and compared with ICP result as the benchmark.

$$d(P, F) = \frac{|\vec{PQ} \cdot \vec{n}|}{|\vec{n}|} \quad (14)$$

As shown in figure 19, in most cases, the thickness of the ICP method and the proposed method are very close to each other and have performed better than the Shokrzadeh registration method.

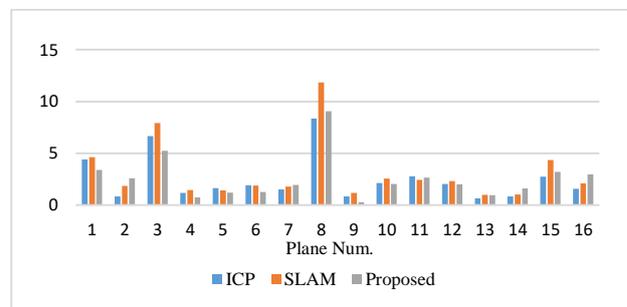


Figure 19. Plane thickness parameter for evaluation.

5. CONCLUSION

In this research, an algorithm for preparing a three-dimensional map of the environment based on MLS data is presented. Our goal was to provide an algorithm based on pyramid voxelization to avoid feature matching methods or search for corresponding points. Another prominent feature of this algorithm is using a differential environment model created from the beginning of the registration process and updated. This model will significantly reduce drift in the final map because of using the information of all previous scans in registration process.

REFERENCES

- A Segal, D Haehnel, S. T. (2009). Generalized-icp. *In Robotics: Science and Systems*, 2, 435.
- Al-Durgham, K., Habib, A., & Kwak, E. (2013). RANSAC approach for automated registration of terrestrial laser scans using linear features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. <https://doi.org/10.5194/isprsannals-II-5-W2-13-2013>
- Biber, P. (2003). The Normal Distributions Transform: A New Approach to Laser Scan Matching. *IEEE International Conference on Intelligent Robots and Systems*, 3, 2743–2748. <https://doi.org/10.1109/iros.2003.1249285>
- Bucksch, A., & Khoshelham, K. (2013). Localized registration of point clouds of botanic trees. *IEEE Geoscience and Remote*

- Sensing Letters*, 10(3), 631–635.
<https://doi.org/10.1109/LGRS.2012.2216251>
- Chatila, R., & Laumond, J. P. (1985). Position referencing and consistent world modeling for mobile robots. *Proceedings - IEEE International Conference on Robotics and Automation*, 138–145. <https://doi.org/10.1109/ROBOT.1985.1087373>
- Chen, C. S., & Hung, Y. P. (1999). RANSAC-Based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/34.809117>
- CloudCompare*. (n.d.).
<https://www.cloudcompare.org/main.html>
- Durrant-Whyte, H. F., & Leonard, J. J. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7, 376–382.
- Eggert, D. W., Fitzgibbon, A. W., & Fisher, R. B. (1998). Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models. *Computer Vision and Image Understanding*, 69(3), 253–272.
<https://doi.org/10.1006/cviu.1998.0667>
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*. <https://doi.org/10.1145/358669.358692>
- Fontanelli, D., Ricciato, L., & Soatto, S. (2007). A fast RANSAC-based registration algorithm for accurate localization in unknown environments using LIDAR measurements. *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering, IEEE CASE 2007*. <https://doi.org/10.1109/COASE.2007.4341827>
- J. Besl, P., & D. McKay, N. (1992). A Method for Registration of 3-D Shapes. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Jiang, G., Yin, L., Liu, G., Xi, W., & Ou, Y. (2018). FFT-based scan-matching for SLAM applications with low-cost laser range finders. *Applied Sciences (Switzerland)*, 9(1). <https://doi.org/10.3390/app9010041>
- Konecny, J., Kromer, P., Prauzek, M., & Musilek, P. (2019). Scan matching by cross-correlation and differential evolution. *Electronics (Switzerland)*, 8(8). <https://doi.org/10.3390/electronics8080856>
- Li, J., Zhong, R., Hu, Q., & Ai, M. (2016). Feature-based laser scan matching and its application for indoor mapping. *Sensors (Switzerland)*, 16(8). <https://doi.org/10.3390/s16081265>
- Lu, J., Wang, W., Fan, Z., Bi, S., & Guo, C. (2018). Point cloud registration based on CPD algorithm. *Chinese Control Conference, CCC, 2018-July*, 8235–8240.
<https://doi.org/10.23919/ChiCC.2018.8482763>
- Mohamed, H., Moussa, A., Elhabiby, M., El-Sheimy, N., & Sesay, A. (2017). A novel real-time reference key frame scan matching method. *Sensors (Switzerland)*, 17(5). <https://doi.org/10.3390/s17051060>
- Park, S., & Park, S. K. (2014). Global localization for mobile robots using reference scan matching. *International Journal of Control, Automation and Systems*, 12(1), 156–168.
<https://doi.org/10.1007/s12555-012-9223-0>
- Qian, C., Zhang, H., Tang, J., Li, B., & Liu, H. (2019). An orthogonal weighted occupancy likelihood map with IMU-aided laser scan matching for 2D indoor mapping. *Sensors (Switzerland)*, 19(7). <https://doi.org/10.3390/s19071742>
- Ryu, K., Dantanarayana, L., Furukawa, T., & Dissanayake, G. (2016). Grid-based scan-to-map matching for accurate 2D map building. *Advanced Robotics*, 30(7), 431–448.
<https://doi.org/10.1080/01691864.2015.1124025>
- Shokrzadeh, P. (2019). Simultaneous localization and mapping for semi-sparse point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4/W18), 1009–1014.
<https://doi.org/10.5194/isprs-archives-XLII-4-W18-1009-2019>
- Smith, R. C., & Cheeseman, P. (1986). On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4), 56–68.
<https://doi.org/10.1177/027836498600500404>
- Takeuchi, E., & Tsubouchi, T. (2006). A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping. *IEEE International Conference on Intelligent Robots and Systems*. <https://doi.org/10.1109/IROS.2006.282246>
- Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2008). EXTENDED RANSAC ALGORITHM FOR AUTOMATIC DETECTION OF BUILDING ROOF PLANES FROM LIDAR DATA. *The Photogrammetric Journal of Finland*.
- Weinmann, M., Weinmann, M., Hinz, S., & Jutzi, B. (2011). Fast and automatic image-based registration of TLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*. <https://doi.org/10.1016/j.isprsjprs.2011.09.010>
- Xu, B., Jiang, W., Shan, J., Zhang, J., & Li, L. (2016). Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds. *Remote Sensing*. <https://doi.org/10.3390/rs8010005>