



An Hybrid Registration Method for SLAM with the M8 Quanergy LiDAR

Marina Aguilar-Moreno^(✉) and Manuel Graña^(✉)

Computational Intelligence Group, University of the Basque Country (UPV/EHU),
San Sebastian, Spain

marina.aguilar@ehu.eus, manuel.grana@ehu.es

Abstract. Simultaneous localization and mapping (SLAM) is process highly relevant for autonomous systems. Accurate sensing provided by range sensors such as the M8 Quanergy LiDAR improves the speed and accuracy of SLAM, which can become an integral part of the control of innovative autonomous cars. In this paper we propose a hybrid point cloud registration method that profits from the high accuracy of classic iterated closest points (ICP) algorithm, and the robustness of the Normal Distributions Transform (NDT) registration method. We report positive results in an in-house experiment encouraging further research and experimentation.

Keywords: Point cloud registration · LiDAR · SLAM · Hybrid registration

1 Introduction

The simultaneous localization and mapping (SLAM) [1–3] aims to estimate a reconstruction of the environment along with the path traversed by the sensor has become an integral part of the robotic operating system (ROS) [4, 5]. One of the most widely used kinds of sensors used for SLAM are laser based depth measurement sensors, or light detection and ranging (LiDAR) sensors, which have been used for scanning and reconstruction of indoor and outdoor environments [6], even in underground mining vehicles [7]. Fusion of LiDAR with GPS allows for large scale navigation [8] of autonomous systems.

Our aim in this paper is to propose a hybridization of two well known point cloud registration methods. Hybridization is akin to the composition of subsystems in circuit like systems. It can be done in series, parallel or interleaving the systems in time. In this paper we propose a serial hybridization where one algorithm serves to provide a robust initial condition to the other. We have carried out SLAM experiments over different datasets using the Iterative Closest Point (ICP) [9], and the Normal Distribution Transform (NDT) [10]. We have found that the ICP method provides less noisy registrations under specific conditions, though it is much more fragile than the NDT. Figure 1 shows a reconstruction of the initial point clouds with ICP and NDT methods, respectively, for the

inhouse data recording that is the demonstrator for the works in this paper. As we can see, ICP reconstruction generates a better surface with higher point density than NDT. Nevertheless, the ICP method is not able to properly register point sets at turning sections of the path traversed by the sensors, as shown in Figure 2, where the registration becomes unwieldy after this turning point. In other words, the ICP methods deals poorly with big rotations.

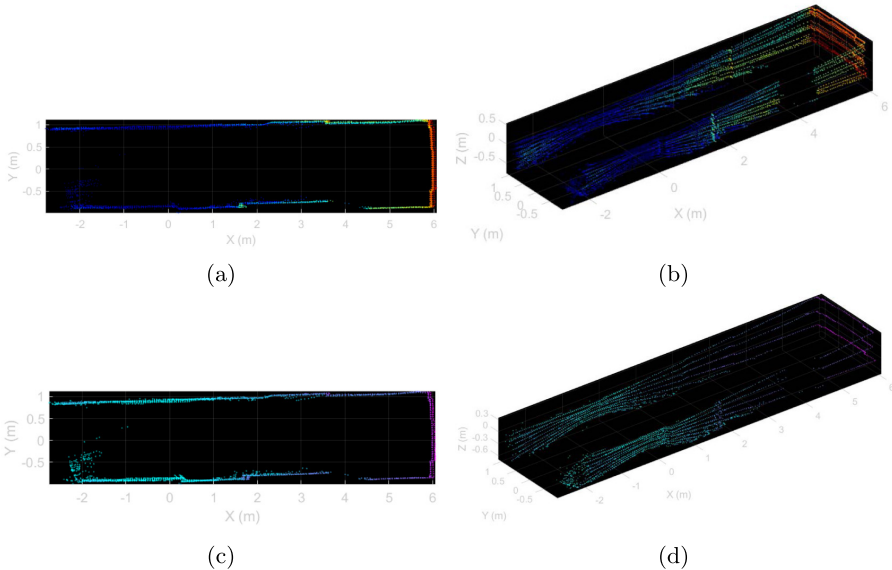


Fig. 1. Results of registration with ICP method, (a) 2D plot of one the range detection at an specific position, (b) reconstructed ray sampling of the 3D surface. Results of registration with NDT method, (c) 2D plot of one the range detection at an specific position, (d) reconstructed ray sampling of the 3D surface.

This paper is structured as follow: A brief presentation of the environment where experiment was carried out and the LiDAR sensor used in it, Quanergy M8. Next, we provide a formal description of the 3D registration methods used in the paper: the Iterative Closest Point (ICP) and the Normal Distribution Transform (NDT). Then, the proposed hybrid NDT algorithm with ICP initialization is described. Finally, experimental results are presented for standard NDT registration and for the hybrid system described in this article, comparing root mean square error of the Euclidean distance, the path obtained and resulting reconstructed surfaces.

2 Materials

New affordable LiDAR sensors, such as the M8 from Quanergy that we are testing in this paper, allow for further popularization of LiDAR based SLAM

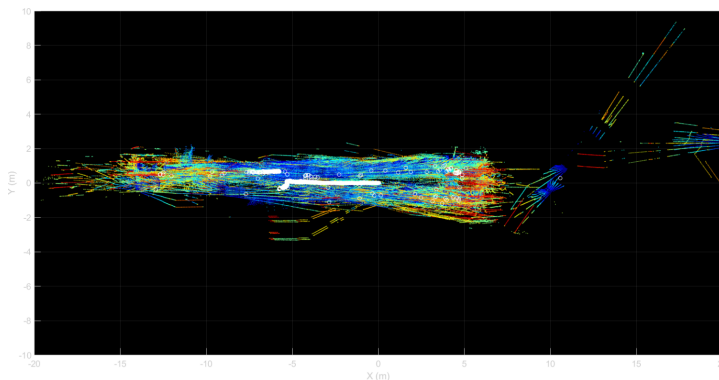


Fig. 2. Estimated trajectory (white points) and registered cloud of points using ICP.

applications. Due to its specific innovative characteristics, the M8 sensor still needs extensive testing by the community in order to assume its integration in the newly developed systems [11]. The work reported in this paper is intended partly to provide such empirical confirmation of the M8 sensor quality continuing experimentation over this sensor data reported elsewhere [12]. Both the time sequence of M8 captured point clouds and the Matlab code used to carry out the computational experiments has been published as open data and open source code¹ in the Zenodo repository for reproducibility.

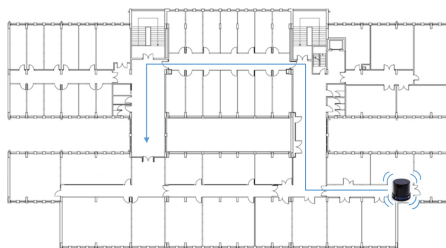


Fig. 3. Nominal path followed during the LiDAR recording.

Location and Experiment Setting. The experiment was carried out in the third floor of the Computer Science School of the UPV/EHU in San Sebastian. Figure 3 shows the nominal path followed by the M8 LiDAR on a manually driven mobile platform. The actual path shows small perturbations around the nominal path. We do not have a precise actual path measurement allowing to quantify the error in the trajectory.

¹ <https://doi.org/10.5281/zenodo.3636204>.

LiDAR M8 Quanergy. The Quanergy M8 LiDAR sensor is a multi-laser system with 8 2D line scanners located on a spinning head. This system is based on Time-of-Flight (TOF) technology whose spin rate is 5 Hz and 20 Hz and its maximum range is 100 m. The Table 1 shows the M8 LiDAR main parameters. Besides, M8 LiDAR comes with 2 desktop applications to manage and visualize point clouds, a SDK to record and show data in real time, and a SDK in framework ROS.

Table 1. Quanergy M8 sensor specifications

Parameter	M8 sensor specifications
Detection layers	8
Returns	3
Minimum range	0.5 m (80% reflectivity)
Maximum range	>100 m (80% reflectivity)
Range accuracy (1σ at 50 m)	< 3 cm
Spin rate	5 Hz–20 Hz
Intensity	8-bits
Field of view	Horizontal 360° - Vertical 20° (+3°/−17°)
Data outputs	Angle, Distance, Intensity, Synchronized Time Stamps

3 Point Cloud Registration Methods

Point cloud registration methods are composed of two steps: (a) finding the correspondence between points in one cloud (the moving) to the points in the other cloud (the reference), and (b) the estimation of the motion parameters that achieve optimal match of the moving points to the reference points after correcting for the motion. If the motion is modeled by a rigid body or an affine transformation, then a matrix transformation common to all points is estimated. If the motion is some non linear deformation, then we have to estimate a flow field. In this paper we are restricted to rigid body transformations, which are compositions of a translation and a rotation. The transformation estimation process takes the form of a minimization problem where the energy function is related to the quality of the correspondence achieved. Next we recall the basics of the two point cloud registration methods. Figure 4 shows the structure of the algorithms as flow diagram of their basic iterations.

3.1 Iterated Closest Point

The most popular and earliest point cloud registration method is the Iterative Closest Point (ICP) proposed by Besl in 1992 [9]. This technique has been exploited in many domains, giving rise to a host of variations whose relative merits are not so easy to assess [13]. Given a point cloud $P = \{\mathbf{p}_i\}_{i=1}^{N_p}$ and a shape described by another point cloud $X = \{\mathbf{x}_i\}_{i=1}^{N_x}$ (The original paper

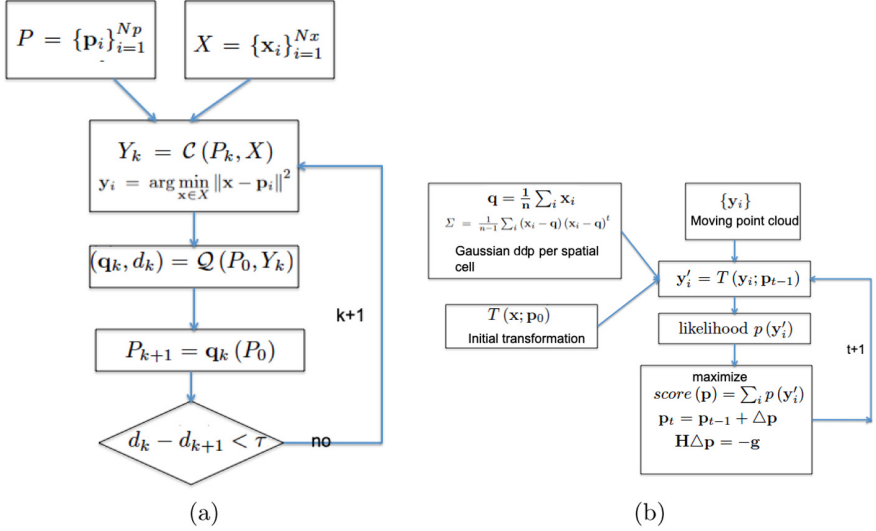


Fig. 4. ICP (a) and NDT (b) processes.

includes the possibility to specify other primitives such as lines or triangles with well defined distances to a point, but we will not consider them in this paper.) the least squares registration of P is given by $(\mathbf{q}, d) = \mathcal{Q}(P, Y)$, where $Y = \{\mathbf{y}_i\}_{i=1}^{Np}$ is the set of nearest points from X to the points in P , i.e. $\mathbf{p}_i \in P; \mathbf{y}_i = \arg \min_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{p}_i\|^2$, denoted $Y = \mathcal{C}(P, X)$, and operator \mathcal{Q} is the least squares estimation of the rotation and translation mapping P to Y using quaternion notation, thus $\mathbf{q} = [\mathbf{q}_R \mid \mathbf{q}_T]^t$ is the optimal transformation specified by a rotation quaternion \mathbf{q}_R and a translation \mathbf{q}_T , and d is the registration error. The energy function minimized to obtain the optimal registration is $f(\mathbf{q}) = \frac{1}{N_p} \sum_{i=1}^{Np} \|\mathbf{y}_i - \mathbf{R}(\mathbf{q}_R) \mathbf{p}_i - \mathbf{q}_T\|^2$, where $\mathbf{R}(\mathbf{q}_R)$ is the rotation matrix constructed from quaternion \mathbf{q}_R . The iteration is initialized by setting $P_0 = P$, $\mathbf{q}_0 = [1, 0, 0, 0, 0, 0, 0]^t$, and $k = 0$. The algorithm iteration is as follows: (1) compute the closest points $Y_k = \mathcal{C}(P_k, X)$, (2) compute the registration $(\mathbf{q}_k, d_k) = \mathcal{Q}(P_0, Y_k)$, (3) apply the registration $P_{k+1} = \mathbf{q}_k(P_0)$, and (4) terminate the iteration if the results are within a tolerance: $d_k - d_{k+1} < \tau$.

3.2 Normal Distribution Transform

The key difference of the NDT [10] method is the data representation. The space around the sensor is discretized into regular overlapped cells. The content of each cell having more than 3 points is modeled by a Gaussian probability distribution of mean $\mathbf{q} = \frac{1}{n} \sum_i \mathbf{x}_i$ and covariance matrix $\Sigma = \frac{1}{n-1} \sum_i (\mathbf{x}_i - \mathbf{q})(\mathbf{x}_i - \mathbf{q})^t$, so that the probability of a LiDAR sample falling in the cell is of the form: $p(\mathbf{x}) \sim \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{q})\Sigma^{-1}(\mathbf{x} - \mathbf{q}))$. Given an initial rigid body transformation $T(\mathbf{x}; \mathbf{p}_0)$, where \mathbf{p} is the vector of translation and rotation parameters, a

Algorithm 1. The hybrid ICP-NDT registration algorithm.

Input: sequence of point clouds $\{N(t)\}_{t=0}^T$ captured by the LiDAR

Output: overall point cloud $M(T)$, sequence of registered transformations $\{\mathcal{T}_t\}_{t=1}^T$

Method = “ICP”

For $t = 0, \dots, T$

1. $N^{(1)}(t) \leftarrow$ remove ground plane from $N(t)$
 2. $N^{(2)}(t) \leftarrow$ remove ego-vehicle from $N^{(1)}(t)$
 3. $N^{(3)}(t) \leftarrow$ downsample $N^{(2)}(t)$
 4. If $t = 0$ then $M(0) = N^{(3)}(t)$; GOTO step 1
 5. $(\mathcal{T}_t, e_t) \leftarrow$ register $\mathcal{T}_{t-1}(N^{(3)}(t))$ to $M(t-1)$ using *Method*
 6. If $e_t > \theta_e$ then *Method* = “NDT”
 7. $N^{(4)}(t) \leftarrow \mathcal{T}_t(N^{(2)}(t))$
 8. $M(t) \leftarrow merge(M(t-1), N^{(4)}(t))$
-

reference point cloud $\{\mathbf{x}_i\}$ modeled by the mixture of the cells Gaussian distributions, and the moving point cloud $\{\mathbf{y}_i\}$, the iterative registration process is as follows: the new laser sample points \mathbf{y}_i are transformed into the reference frame of the first cloud $\mathbf{y}'_i = T(\mathbf{y}_i; \mathbf{p}_{t-1})$, where we find the cell where it falls and use its parameters (\mathbf{q}, Σ) to estimate its likelihood $p(\mathbf{y}'_i)$. The score of the transformation is given by $score(\mathbf{p}) = \sum_i p(\mathbf{y}'_i)$. The maximization of the score is carried out by gradient ascent using Newton’s method, i.e. $\mathbf{p}_t = \mathbf{p}_{t-1} + \Delta\mathbf{p}$. The parameter update is computed solving the equation $\mathbf{H}\Delta\mathbf{p} = -\mathbf{g}$, where \mathbf{H} and \mathbf{g} are the Hessian and the gradient of the $-score(\mathbf{p}_{t-1})$ function, respectively. Closed forms of \mathbf{H} and \mathbf{g} are derived in [10] for the 2D case. An extension to 3D is described in [14].

4 Hybrid Point Cloud Registration Algorithm

In this section, hybrid registration algorithm which uses ICP and NDT methods is presented. As ICP reconstructed surface generates a better surface, i.e. with greater point density than NDT, but ICP method is not able to properly register point sets at turning sections. Our hybrid algorithm that initially uses ICP method changing to the NDT method when the registration error becomes higher than a threshold.

Algorithm 1 presents an algorithmic description of the proposed hybrid registration method. The input to the algorithm is the sequence of point clouds recorded by the LiDAR $N(t)$; $t = \{1, \dots, T\}$. The point sets are obtained while the LiDAR sensor is being displaced manually in the environment according to the approximate path in Figure 3. The final result of the process is a global point cloud $M(T)$ that contains all the recorded 3D points registered relative to the first acquired point cloud $N(0)$, and the estimation of the LiDAR recording positions relative to the initial position given by the composition of

the point cloud registration transformations estimated up to this time instant $\{\mathcal{T}_t\}_{t=1}^T$. The process of each point cloud is as follows: For each point cloud $N(t)$ acquired at time t , firstly we remove the ground plane applying a segmentation denoted $N^{(1)}(t)$. Secondly we remove the ego-vehicle points, denoted $N^{(2)}(t)$. Thirdly, we down-sample the point cloud to decrease the computation time and improve accuracy registration, denoted $N^{(3)}(t)$. For the initial point cloud $N^{(3)}(0)$ becomes the global merged reference cloud $M(0)$. For subsequent time instants $t > 0$, the fourth step is to estimate the transformation \mathcal{T}_t of the acquired data $N^{(3)}(t)$ optimally registered to the previous global point cloud $M(t-1)$. For this estimation, we may use ICP or NDT methods. We then apply this transformation to the acquired point cloud previous to downsampling $N^{(4)}(t) = \mathcal{T}_t(N^{(3)}(t))$, which is used to obtain the new global registered point cloud by merging $M(t) \leftarrow \text{merge}(M(t-1), N^{(4)}(t))$. Our hybrid strategy consists in using the ICP method in the initial steps of the algorithm, up a time instant when the registration error meets a given threshold, after this time point the system shifts to use the NDT method to continue registration of all remaining point clouds. The rationale is that the ICP acts as a good initial estimation for the ensuing NDT steps, as will be demonstrated in the results section below.

5 Results

Figure 5 shows the evolution of the registration error for ICP, NDT and the hybrid algorithm presented in the previous section, setting $\theta_e = 0.25$. The plot scale is logarithmic in order to be able to represent the three error course in the same plot. The ICP algorithm gives the highest error. At the beginning the error is low, but it is increasing when we add more point clouds, until it explodes in iteration $t = 1308$. Both NDT and the proposed hybrid method registration errors are bounded over time. The error of the hybrid method error is the same as the error of the ICP until the point where the algorithm shifts from ICP to NDT at the 138-th iteration. Afterwards the hybrid method has a slightly lower error than the NDT due to the quality of the initialization provided by the ICP. Table 2 gives numerical registration errors. However, the effect of using the ICP initialization is much more evident when assessing the overall results. Figure 6 shows the trajectory estimation (up) and the projection of the reconstructed surfaces on the floor plan (bottom) when using only NDT registration method. Though the error of the method is bounded, it can be appreciated that after the second turning point a noticeable drift appears, causing misalignment of the reconstructed surfaces with the actual floor plan walls. (Direction of advance is from right to left). On the other hand, the hybrid method trajectory and reconstruction results are provided in Figure 7 showing a much better fit to the floor plan walls. We conclude that the ICP provides a much more robust initialization for NDT.

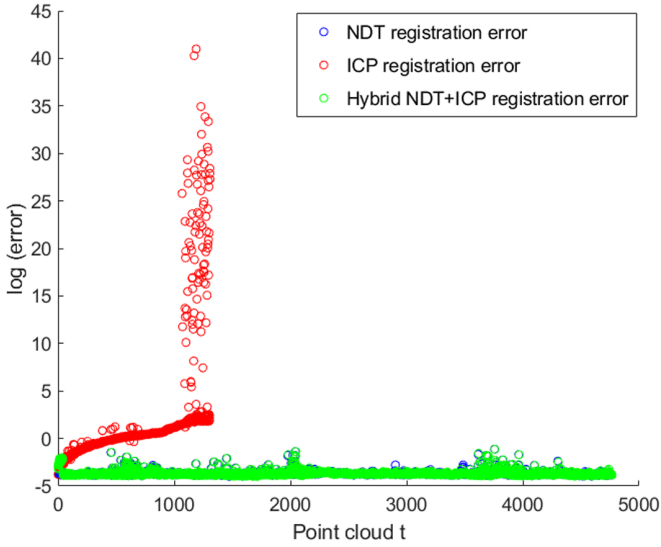


Fig. 5. Evolution of the registration error (log plot) for NDT (blue dots), ICP (red dots), and hybrid ICP-NDT method (green dots). (Color figure online)

Table 2. Registration error for ICP method, NDT method and hybrid ICP-NDT method

	ICP method (for 1308 iterations)	NDT method	Hybrid ICP-NDT method
Maximum error	6.3528×10^{17}	0.3245	0.3208
Median error	7.2580×10^{14}	0.0251	0.0256
Total error	9.4935×10^{17}	119.4010	122.0764

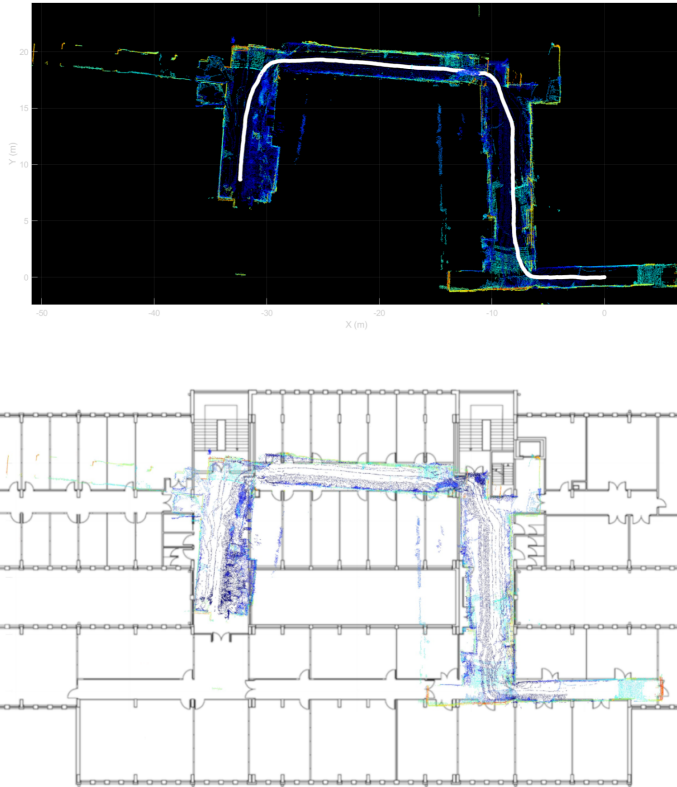


Fig. 6. Estimated trajectory (white points) and registered cloud of points using NDT (Above). Projection of the NDT registered point cloud on the plan of stage 3 of the building.

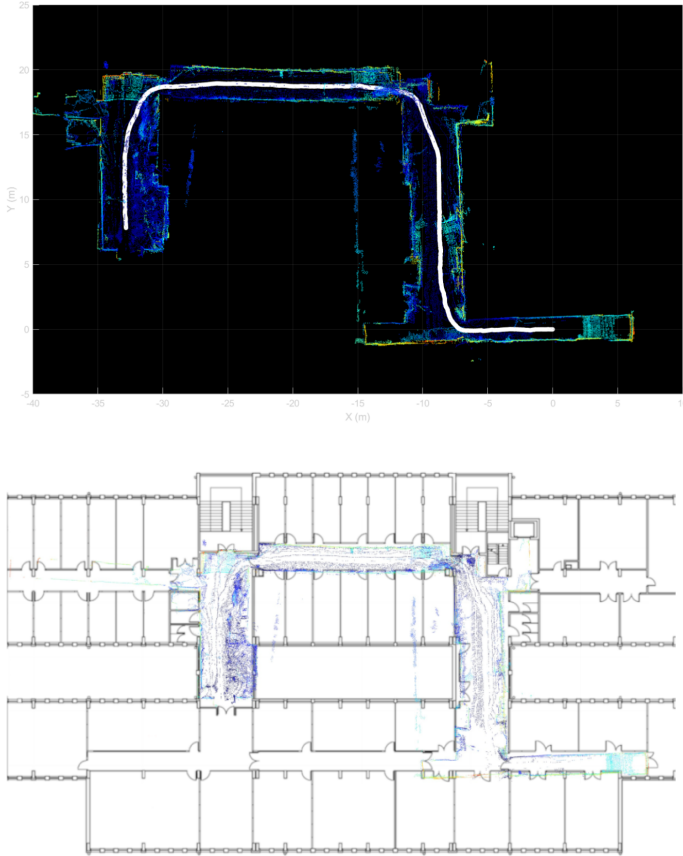


Fig. 7. Estimated trajectory (white points) and registered cloud of points using hybrid method proposed (Above). Projection of the hybrid registered point cloud on the plan of stage 3 of the building.

6 Conclusion

In this paper we report a hybridization between two registration methods for 3D point clouds, namely the Iterative Closest Point (ICP) and the Normal Distributions Transform (NDT). The experimental point clouds have been recorded with the M8 Quanergy LiDAR sensor traversing an indoor path the third floor of the Computer Science School of the UPV/EHU in San Sebastian. The general SLAM algorithm followed in this paper includes preprocessing (detect and remove ego-vehicle and floor, and down-sample), registration, transformation and merger of point cloud. We report the registration error, the estimation of the path traversed by the sensor, and the reconstructed point cloud. The hybrid method produces a slightly lower error than the NDT. On a qualitative assessment the hybrid method produces better surface reconstruction and path

estimation. Future works will consider the analysis and exploitation of out-door recording methods.

Acknowledgments. This work has been partially supported by FEDER funds through MINECO project TIN2017-85827-P, and grant IT1284-19 as university research group of excellence from the Basque Government.

References

1. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* **13**, 99–110 (2006)
2. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping (SLAM): part II. *IEEE Robot. Autom. Mag.* **13**, 108–117 (2006)
3. Cadena, C., et al.: Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Trans. Rob.* **32**, 1309–1332 (2016)
4. Xuexi, Z., Guokun, L., Genping, F., Dongliang, X., Shiliu, L.: SLAM algorithm analysis of mobile robot based on Lidar. In: 2019 Chinese Control Conference (CCC), pp. 4739–4745, July 2019
5. Yagfarov, R., Ivanou, M., Afanasyev, I.: Map comparison of Lidar-based 2D SLAM algorithms using precise ground truth. In: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1979–1983 (2018)
6. Caminal, I., Casas, J.R., Royo, S.: SLAM-based 3D outdoor reconstructions from Lidar data. In: 2018 International Conference on 3D Immersion (IC3D), pp. 1–8, December 2018
7. Wu, D., Meng, Y., Zhan, K., Ma, F.: A Lidar SLAM based on point-line features for underground mining vehicle. In: 2018 Chinese Automation Congress (CAC), pp. 2879–2883, November 2018
8. Deng, Y., Shan, Y., Gong, Z., Chen, L.: Large-scale navigation method for autonomous mobile robot based on fusion of GPS and Lidar SLAM. In: 2018 Chinese Automation Congress (CAC), pp. 3145–3148, November 2018
9. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992)
10. Biber, P., Straßer, W.: The normal distributions transform: a new approach to laser scan matching, vol. 3, pp. 2743–2748, December 2003
11. Mitteta, M.A., Noura, H., Roynard, X., Goulette, F., Deschaud, J.E.: Experimental assessment of the Quanergy M8 LIDAR Sensor. *ISPRS Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* **41B5**, 527–531 (2016)
12. Aguilar-Moreno, M., Graña, M.: A comparison of registration methods for SLAM with the M8 Quanergy LiDAR. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) *SOCO 2020. AISC*, vol. 1268, pp. 824–834. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_79
13. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing ICP variants on real-world data sets. *Auton. Robot.* **34**, 133–148 (2013). <https://doi.org/10.1007/s10514-013-9327-2>
14. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* **24**, 803–827 (2007)