M-GCLO: Multiple Ground Constrained LiDAR Odometry

Yandi Yang [1], Naser El-Sheimy [1]

[1] Department of Geomatics Engineering, University of Calgary, Calgary, Alberta, Canada – (yandi.yang, elsheimy)@ucalgary.ca

KEY WORDS: Mobile Mapping System, LiDAR SLAM, Ground Constraints, Pose Estimation, Localization.

ABSTRACT:

Accurate LiDAR odometry results contribute directly to high-quality point cloud maps. However, traditional LiDAR odometry methods drift easily upward, leading to inaccuracies and inconsistencies in the point cloud maps. Considering abundant and reliable ground points in the Mobile Mapping System(MMS), ground points can be extracted, and constraints can be built to eliminate pose drifts. However, existing LiDAR-based odometry methods either do not use ground point cloud constraints or consider the ground plane as an infinite plane (i.e., single ground constraint), making pose estimation prone to errors. Therefore, this paper is dedicated to developing a Multiple Ground Constrained LiDAR Odometry(M-GCLO) method, which extracts multiple grounds and optimizes those plane parameters for better accuracy and robustness. M-GCLO includes three modules. Firstly, the original point clouds will be classified into the ground and non-ground points. Ground points are voxelized, and multiple ground planes are extracted, parameterized, and optimized to constrain the pose errors. All the non-ground point clouds are used for point-to-distribution matching by maintaining an NDT voxel map. Secondly, a novel method for weighting the residuals is proposed by considering the uncertainties of each point in a scan. Finally, the jacobians and residuals are given along with the weightings for estimating LiDAR states. Experimental results in KITTI and M2DGR datasets show that M-GCLO outperforms state-of-the-art LiDAR odometry methods in large-scale outdoor and indoor scenarios.

1. Introduction

High-precision point cloud maps have wide applications in different fields, such as mobile robotics (Nagy et al., 2018), autonomous driving(Gao et al., 2021), and building information models(Blaser et al., 2018). Besides, high-precision point clouds are the fundamentals for High-Definition (HD) map production. LiDAR (Light Detection And Ranging) technology is the most useful sensor for acquiring point cloud data and can be divided into MLS (Mobile Laser Scanner) and TLS (Terrestrial Laser Scanner) based on different mapping platforms. MLS, which is mounted on mobile mapping devices, is more widely adopted in large-scale outdoor mapping tasks compared with TLS due to its efficiency. However, in GNSS-denied environments (tunnels, forests), the performance of the existing MLS may suffer from large errors due to the block of GNSS signals. LiDAR odometry is a state estimation method that uses only LiDAR sensors to estimate the poses and produce the point clouds. The rapid development of LiDAR odometry has made the acquisition of point cloud maps using MLS systems more robust and effective.

Ground constraints have been applied in LiDAR odometry algorithms. Ground points are extracted from raw point clouds, and a ground plane map is maintained for scan-to-map matching (Shan et al., 2018; Su et al., 2021; Liu et al., 2019). Koide et al. (2019) apply ground plane constraints in a large indoor environment and assume a global ground plane exists. The plane is parameterized and optimized to control the pose drifts while doing LiDAR SLAM. Chen et al. (2021) proposed a ground segmentation method for optimizing plane features collected using a backpack LiDAR system. Besides considering ground plane constraints in point cloud features, Zheng et al. (2018) and Zheng et al. (2019) directly constrained the pose by the SE2 group. However, all the methods mentioned above only considered one ground plane, making it hard to compress the long-term upward pose drift and even unavailable when the terrain is complex and multiple ground planes exist.

LiDAR odometry relies on LiDAR registration. Adequate weights of the registration residuals are necessary in filtering-based state estimation but also in optimization-based methods. In normal distribution transformation(NDT) (Magnusson et al., 2007), the weightings are from the covariances of the voxel. F-LOAM (Wang et al., 2021), an optimization-based method, weighted edge features with higher smoothness and planar features with lower smoothness. LiLi-OM (Li et al., 2021) is still an optimization-based method and proposes a metric weighting function that considers both geometric and appearance consistencies. In the Kalman filter of Point-LIO (He et al., 2023) and FAST-LIO2(Xu et al., 2022), the LiDAR measurement noise is set as a constant value. However, each point in the point clouds has uncertainties caused by the LiDAR sensor measurement noises, and those methods do not consider the error caused by each point's covariance, making the system less robust and adaptable.

To deal with the challenges above, this paper proposes a multiple ground-constrained LiDAR odometry method named M-GCLO, which uses multiple ground constraints by voxelizing ground points and optimizing each plane in a single voxel to improve the pose accuracy, robustness, and adaptability as much as possible. The main contributions of this article are as follows:

(1) Instead of constraining the pose with only one ground plane, we build multiple ground plane constraints by dividing the ground points into different voxels and optimizing multiple ground plane parameters. This can improve both the accuracy and robustness of localization.

(2) To better weigh the residuals of point cloud registration, we propagate the uncertainties from single LiDAR points to the final point-to-plane and point-to-distribution residuals. Based on the uncertainties, we can weigh the residuals properly, which makes the algorithm less prone to errors and more accurate.

(3) To evaluate the performance of the algorithm in different environments, we test M-GCLO on both large-scale outdoor and small-scale indoor environments, demonstrating the accuracy, robustness, and adaptability of the algorithm compared with the state-of-the-art algorithms.

2. Methodology

2.1 Plane Parameterization

Parameterization of planes is needed to utilize the ground plane constraints for estimating the pose of LiDAR before optimizing ground plane features. A plane can be parameterized with the plane normal $\boldsymbol{n}$ and a random point $\boldsymbol{q}$ on the plane. Suppose there are N points belonging to the plane $\boldsymbol{p}_i$, i=1, …, N. The centroid $\bar{\boldsymbol{p}}$ and covariance $\boldsymbol{A}$ of the points are:

$$\bar{\boldsymbol{p}}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{p}_i \tag{1}$$

$$\boldsymbol{A}=\frac{1}{N\text{-}1}\sum_{i=1}^{N}\left(\boldsymbol{p}_i\text{-}\bar{\boldsymbol{p}}\right)\left(\boldsymbol{p}_i\text{-}\bar{\boldsymbol{p}}\right)^T \tag{2}$$

Say the 3 eigenvalues of $\boldsymbol{A}$ are $\lambda_1>\lambda_2>\lambda_3$. Normal $\boldsymbol{n}$ is the eigenvector associated with λ_3. Since the centroid is on the fitted plane, $\boldsymbol{q}$ can be set as $\bar{\boldsymbol{p}}$ for simplicity. The planarity P_λ (Weinmann et al., 2015) represents a 2D feature and can be used for selecting planes:

$$P_\lambda=\frac{\lambda_2-\lambda_3}{\lambda_1} \tag{3}$$

When P_λ is larger than a threshold(i.e. the point cloud distribution in one direction is much smaller than the other two), the N points can be seen as one plane, and the plane will be selected and parameters are updated for providing multiple ground constraints for state estimation.

2.2 Voxelization

To utilize as many ground plane constraints as possible, the ground point clouds can be divided into different voxels after the ground points are segmented from each frame, and points in each voxel will be fitted as a plane. Common algorithms optimize the ground point clouds as one infinite plane. However, there would be slopes, especially in large-scale outdoor environments where the single-plane definition does not apply. Voxelizing the ground points by a radius and optimizing multiple ground planes use as many ground constraints as possible, which would be beneficial for constraining the vertical pose drifts.

As shown in Figure 1, the point clouds are segmented into ground (in black color) and off-ground points. The ground points are then voxelized and the points in each voxel will be considered as a plane candidate. The planes that meet the criterion (3) will be saved and optimized continuously. Figure 2 demonstrates the voxelization results of KITTI 05 sequence with a voxel size of 10m.

For non-ground point clouds, the points are voxelized for Normal Distributions Transform(NDT) matching. Similar to the ground voxels, each non-ground voxel maintains the mean value and covariance of the points in the voxel. The voxels will be maintained and updated along with the mapping process.

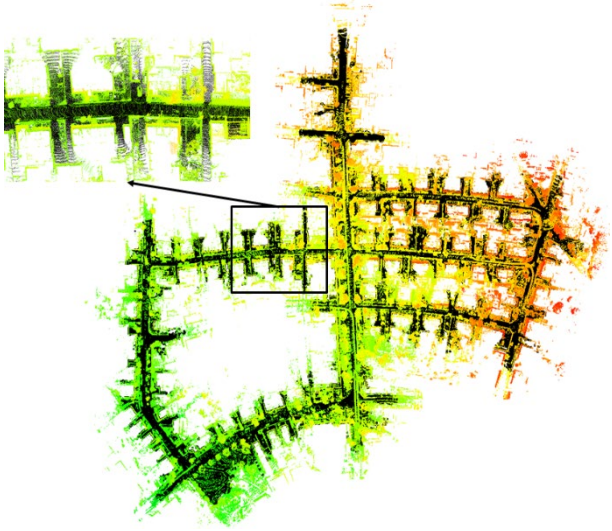

Figure 1. Ground segmentation of point clouds.

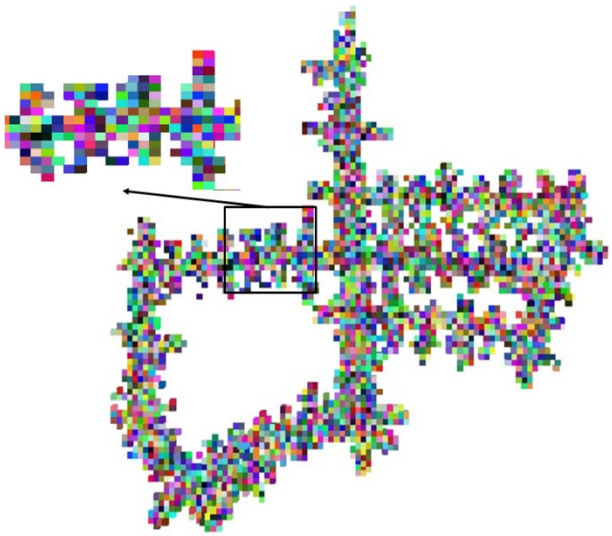

Figure 2. Voxelization of ground points.

2.3 Residual Weighing

Both in optimization-based and filter-based state estimation methods, adequate weights of residuals can improve the pose accuracy and make the algorithm more robust to errors. Since the points acquired by LiDAR are not technically accurate and each point has its own uncertainty, the uncertainties can be modeled and propagated from a single point to registration residuals and finally to the state (i.e. the position and attitude of the body).

Firstly, we model the residuals for the point cloud registration process. Then, we can propagate the uncertainties of the residuals from a single point’s uncertainty. Based on the uncertainties, we are able to weigh the residuals accordingly.

Assuming a point $\boldsymbol{p}^{\mathrm{L}}$ in LiDAR frame has the variance $\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{L}}}$ and the state $\boldsymbol{x}=\{\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}},\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}\}$ has the rotation covariance $\boldsymbol{\Sigma}_{\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}}$ in Lie algebra of SO(3) and translation covariance $\boldsymbol{\Sigma}_{\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}}$, where $\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}$ and $\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}$ are the rotation matrix and translation vector from LiDAR frame to world frame, respectively. The world frame is set as the first frame of LiDAR point cloud so that the pose of the first frame would be the identical matrix. With the raw point

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-1-2024-283-2024 | © Author(s) 2024. CC BY 4.0 License.

coordinates and state, the raw point cloud can be transformed to the world frame:

$$\boldsymbol{p}^{\mathrm{W}}=\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\boldsymbol{p}^{\mathrm{L}}+\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}} \tag{4}$$

The transform process builds a global point cloud map. From (4), the variance $\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}$ of $\boldsymbol{p}^{\mathrm{W}}$ can be propagated from $\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{L}}}$ with known state $\boldsymbol{x}$ and raw point coordinate $\boldsymbol{p}^{\mathrm{L}}$:

$$\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}=\left[-\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\boldsymbol{p}^{\mathrm{L}}\right]_{\times}\boldsymbol{\Sigma}_{\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}}\left[-\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\boldsymbol{p}^{\mathrm{L}}\right]_{\times}^{\mathrm{T}}+\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{L}}}\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}\mathrm{T}}+\boldsymbol{\Sigma}_{\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}} \tag{5}$$

where $\times$ is the operation for getting the skew-symmetric matrix of the vector. For the ground voxels, the residual would be the point-to-plane distance d_{plane}:

$$d_{\mathrm{plane}}=\boldsymbol{n}^{\mathrm{T}}(\boldsymbol{p}^{\mathrm{W}}-\boldsymbol{q}) \tag{6}$$

Since we have the uncertainty $\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}$ of $\boldsymbol{p}^{\mathrm{W}}$ in (5), we can calculate the variance $\Sigma_{d_{\mathrm{plane}}}$ of d_{plane} directly:

$$\Sigma_{d_{\mathrm{plane}}}=\boldsymbol{n}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}\boldsymbol{n}+(\boldsymbol{p}^{\mathrm{W}}-\boldsymbol{q})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{n}}(\boldsymbol{p}^{\mathrm{W}}-\boldsymbol{q})+\boldsymbol{n}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{q}}\boldsymbol{n} \tag{7}$$

where the normal $\boldsymbol{n}$ is from eigendecomposition in (2) and centroid $\boldsymbol{q}$ is from(1). The $\boldsymbol{\Sigma}_{\boldsymbol{n}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ can be calculated based on the $\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}$(Liu, et al., 2021):

$$\boldsymbol{\Sigma}_{\boldsymbol{n}}=\sum_{\mathrm{i}=1}^{N}\boldsymbol{U}\begin{bmatrix}\boldsymbol{F}_{1,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\\ \boldsymbol{F}_{2,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\\ \boldsymbol{F}_{3,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\end{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\begin{bmatrix}\boldsymbol{F}_{1,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\\ \boldsymbol{F}_{2,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\\ \boldsymbol{F}_{3,3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}\end{bmatrix}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} \tag{8}$$

Where $\boldsymbol{F}_{\mathrm{m},3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}$ is:

$$\boldsymbol{F}_{\mathrm{m},3}^{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}}=\begin{cases}\dfrac{\left(\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}-\bar{\boldsymbol{p}}\right)^{\mathrm{T}}}{(N-1)(\lambda_3-\lambda_{\mathrm{m}})}\left(\boldsymbol{u}_{\mathrm{m}}\boldsymbol{u}_3{}^{\mathrm{T}}+\boldsymbol{u}_3\boldsymbol{u}_{\mathrm{m}}^{\mathrm{T}}\right) & ,\mathrm{m}\neq 3\\ \boldsymbol{0}_{1\times 3} & ,\mathrm{m}=3\end{cases} \tag{9}$$

Since $\boldsymbol{q}$ is the centroid of the voxel(i.e. the average of the coordinates) and each point is independent of other point, $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ would be:

$$\boldsymbol{\Sigma}_{\boldsymbol{q}}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Sigma}_{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}} \tag{10}$$

For the non-ground points, all the points will be used for NDT matching. The residual is the distance vector from point $\boldsymbol{p}^{\mathrm{W}}$ to centroid $\bar{\boldsymbol{p}}$:

$$\boldsymbol{d}_{\mathrm{ndt}}=\boldsymbol{p}^{\mathrm{W}}-\bar{\boldsymbol{p}}=\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\boldsymbol{p}^{\mathrm{L}}+\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}-\bar{\boldsymbol{p}} \tag{11}$$

The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{d}_{ndt}}$ of $\boldsymbol{d}_{ndt}$ is:

$$\boldsymbol{\Sigma}_{\boldsymbol{d}_{\mathrm{ndt}}}=\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}+\boldsymbol{\Sigma}_{\bar{\boldsymbol{p}}}=\boldsymbol{\Sigma}_{\boldsymbol{p}^{\mathrm{W}}}+\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Sigma}_{\boldsymbol{p}_{\mathrm{i}}^{\mathrm{W}}} \tag{12}$$

The information matrices (i.e. weightings) $\Omega_{d_{\mathrm{plane}}}$ and $\boldsymbol{\Omega}_{\boldsymbol{d}_{\mathrm{ndt}}}$ are computed directly from the inverse of the variance(covariance):

$$\Omega_{d_{\mathrm{plane}}}=\Sigma_{d_{\mathrm{plane}}}^{-1},\ \boldsymbol{\Omega}_{\boldsymbol{d}_{\mathrm{ndt}}}=\boldsymbol{\Sigma}_{\boldsymbol{d}_{\mathrm{ndt}}}^{-1} \tag{13}$$

2.4 State Estimation

LiDAR odometry is for estimating the rotation and translation of the body $\boldsymbol{x}=\{\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}},\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}\}$, where L is the current LiDAR frame and W is the first LiDAR frame set as the world frame. Assuming the Lie algebra of $\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}$ is $\boldsymbol{\varphi}$, the derivatives would be:

$$\frac{\partial d_{\mathrm{plane}}^{'}}{\partial\boldsymbol{\varphi}}=-\left\lfloor\boldsymbol{p}^{\mathrm{L}}\right\rfloor_{\times}\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}\mathrm{T}}\boldsymbol{n},\frac{\partial d_{\mathrm{plane}}^{'}}{\partial\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}}=\boldsymbol{n} \tag{14}$$

$$\frac{\partial\boldsymbol{d}_{\mathrm{ndt}}^{'}}{\partial\boldsymbol{\varphi}}=-\boldsymbol{R}_{\mathrm{L}}^{\mathrm{W}}\left\lfloor\boldsymbol{p}^{\mathrm{L}}\right\rfloor_{\times},\frac{\partial\boldsymbol{d}_{\mathrm{ndt}}^{'}}{\partial\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}}=\boldsymbol{I}_{3\times 3} \tag{15}$$

The jacobians regarding to the residuals are:

$$\boldsymbol{J}_{1\times 6}^{\mathrm{plane}}=\left[\left(\frac{\partial d_{\mathrm{plane}}^{'}}{\partial\boldsymbol{\varphi}}\right)^{\mathrm{T}},\left(\frac{\partial d_{\mathrm{plane}}^{'}}{\partial\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}}\right)^{\mathrm{T}}\right] \tag{16}$$

$$\boldsymbol{J}_{3\times 6}^{\mathrm{ndt}}=\left[\frac{\partial\boldsymbol{d}_{\mathrm{ndt}}^{'}}{\partial\boldsymbol{\varphi}},\frac{\partial\boldsymbol{d}_{\mathrm{ndt}}^{'}}{\partial\boldsymbol{t}_{\mathrm{L}}^{\mathrm{W}}}\right] \tag{17}$$

With residuals(d_{plane} and $\boldsymbol{d}_{\mathrm{ndt}}$), weightings($\Omega_{d_{\mathrm{plane}}}$ and $\boldsymbol{\Omega}_{\boldsymbol{d}_{\mathrm{ndt}}}$) and related jacobians($\boldsymbol{J}_{1\times 6}^{\mathrm{plane}}$ and $\boldsymbol{J}_{3\times 6}^{\mathrm{ndt}}$), we can estimate the state by either filtering or least-square methods.

3. Experiments

In order to show the adaptability of the algorithm in different scales of scenes, we conduct experiments on both outdoor large-scale environments and indoor small-scale environments. For the outdoor experiments, we evaluate our method on KITTI (Geiger et al., 2012), which is one of the most popular dataset for autonomous driving. As for the indoor tests, we select M2DGR(Yin et al., 2021) for evaluation. M2DGR has various indoor scenarios and provides ground truths. The algorithm is then compared with state-of-the-art LiDAR odometry methods for testing the accuracy and robustness of M-GCLO. To evaluate the localization accuracies, we calculate ATE (Zhang et al., 2018) between the estimated trajectories and ground truths. For a fair comparison, loop closure modules are off in all the other methods and no LiDAR-Inertial odometry methods are selected for better focus on the point clouds.

3.1 Performance on Outdoor Datasets

The point clouds in KITTI dataset are collected using a Velodyne HDL-64E and ground truths are provided by GPS/INS. 11 sequences are provided with ground truths including city, residential and road scenarios, etc. We test M-GCLO with ground voxel size of 10m and non-ground voxel size of 2m. The trajectories of ground truths (in red) and M-GCLO (in blue) are shown in Figure 3. From the vertical viewpoint, the horizontal positions do not drift a lot even in long-distance datasets(00, 02 and 08 sequences). As we know, the main position errors for LiDAR odometry are from the vertical drifts due to lack of points on the ground and lower vertical resolution compared with horizontal resolution.

To evaluate the accuracy of M-GCLO with its counterparts, we compare the ATEs of rotation (in degree) and translation (in meter) with LiTAMIN2(Yokozuka et al., 2021), MULLS(Pan et al., 2021), F-LOAM(Wang et al., 2021), LeGO-LOAM(Shan et al., 2018) and VoxelMap(Yuan et al., 2022).

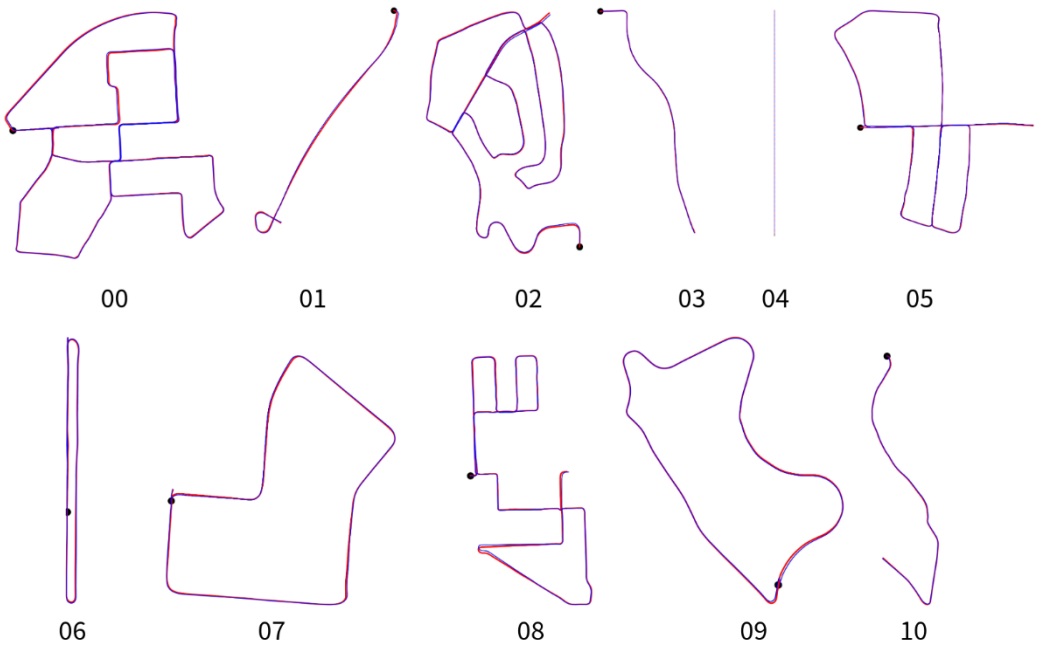

Figure 3. Trajectories of ground truth and M-GCLO on KITTI.

Seq. (Length[km])	00 (3.7)	1 (2.4)	2 (5.0)	3 (0.6)	4 (0.4)	5 (2.2)	6 (1.2)	7 (0.7)	8 (3.2)	9 (1.7)	10 (0.9)	Avg. (°)
LiTAMIN2	1.6	3.5	2.7	2.6	2.3	1.1	1.1	1.0	1.3	1.7	1.2	1.8
MULLS	1.7	**1.0**	2.4	**0.7**	**0.2**	1.0	**0.4**	**0.5**	1.9	1.4	**0.5**	1.5
F-LOAM	2.5	4.3	1.9	3.3	0.3	1.6	2.2	1.1	1.5	2.2	1.6	2.0
LeGO-LOAM	2.8	3.8	4.1	4.1	3.3	1.9	1.4	1.5	2.5	2.2	1.9	2.8
VoxelMap	**0.9**	1.9	1.7	1.2	0.6	**0.8**	**0.4**	0.7	**1.1**	**1.0**	1.0	**1.2**
M-GCLO	1.1	2.0	**1.3**	1.5	0.3	1.1	1.0	1.1	**1.1**	1.1	1.1	**1.2**

Table 1. Rotation comparison on KITTI dataset.

Seq. (Length[km])	00 (3.7)	1 (2.4)	2 (5.0)	3 (0.6)	4 (0.4)	5 (2.2)	6 (1.2)	7 (0.7)	8 (3.2)	9 (1.7)	10 (0.9)	Avg. (m)
LiTAMIN2	5.8	15.9	10.7	0.8	0.7	2.4	0.9	**0.6**	2.5	2.1	**1.0**	5.1
MULLS	6.1	**2.4**	10.6	0.7	0.9	2.4	0.6	**0.6**	4.3	2.5	1.1	4.8
F-LOAM	7.1	17.8	4.2	0.8	0.6	3.0	4.2	0.9	3.6	6.6	1.4	4.6
LeGO-LOAM	6.3	119.4	14.7	0.9	0.8	2.8	0.8	0.7	3.5	2.1	1.8	11.1
VoxelMap	**2.8**	7.8	6.1	0.7	0.3	**1.2**	**0.4**	0.7	**2.3**	1.9	1.1	2.9
M-GCLO	3.0	8.5	**3.2**	**0.6**	**0.3**	1.3	1.0	0.7	2.4	**1.7**	1.1	**2.2**

Table 2. Translation comparison on KITTI dataset.

The results for LiTAMIN2 are from the original paper and the results for MULLS, LeGO-LOAM, and VoxelMap are from Yuan(2022). The parameters for F-LOAM are in default ones.

The rotation comparison results are shown in Table 1. M-GCLO has the lowest rotation error on average and achieves the best orientation result(1.3°) on sequence 2, which is the longest sequence and has lots of turns. LiTAMIN2 is a method that combined NDT and ICP and it’s obvious that with the ground constraints of M-GCLO, the overall rotation accuracy can be improved from 1.8° to 1.2°. Besides, LeGO-LOAM also extracts ground points but optimizes only one ground plane. M-GCLO outperforms it with high robustness remaining high accuracy orientation in all 11 sequences. MULLS extracts multiple classified features(ground, façade, pillar, and beam, etc.) and F-LOAM uses edge features and plane features from raw point clouds. M-GCLO performs better than these two multiple feature based methods overall.

Table 2 demonstrates the translation comparison results on KITTI dataset. M-GCLO not only has the highest rotation accuracy on sequence 2, but performs best in positioning evaluation with only 3.2 meter’s position error. Comparing the average translation errors, M-GCLO outperforms the feature-based LiDAR odometry methods(MULLS and F-LOAM), LiTAMIN2(hybrid ICP and NDT), and also VoxelMap which uses all the plane voxels in the point clouds.

3.2 Performance on Indoor Datasets

To evaluate the performance of the method in indoor environments, we test the algorithm on the M2DGR dataset which acquires the LiDAR with Velodyne VLP-32C, and the ground truth is provided by a laser scanner and a motion-capture system for indoor scenes. We then compare our method with F-LOAM(Wang et al., 2021), KISS-ICP(Vizzo et al., 2023), MULLS(Pan et al., 2021), and LeGO-LOAM(Shan et al., 2018),

respectively. Loop closure optimization is also turned off for KISS-ICP, MULLS, and LeGO-LOAM for pure LiDAR odometry evaluation. Since no rotation ground truth is provided for the indoor datasets of M2DGR, we only list the position errors here. Meanwhile, we failed on the Lift part dataset because the robot was sent to the second floor by a lift and the point cloud registration could not work.

Scene (Length [m])	Door (200)	Hall (845)	Room (144)	Rm. dark (395)	Avg. (m)
F-LOAM	0.25	0.63	0.14	0.20	0.31
KISS-ICP	0.30	0.57	0.24	0.29	0.35
MULLS	0.35	0.64	0.15	0.20	0.34
LeGO-LOAM	0.21	0.56	0.16	0.21	0.29
M-GCLO	0.22	0.52	0.14	0.20	0.27

Table 3. Translation comparison on M2DGR dataset.

Table 3 demonstrates the position results for the door, hall, room and dark room datasets. In door scenes, the robot traveled through a door from indoor to outdoor. For hall dataset, point clouds are collected through random walking of robot in a hall, which lasted for 20 minutes. The data for the room and dark room were acquired in a room with brightness and darkness, respectively. Unlike image collection, point cloud acquirements will not be affected by the lights, so the feature registration will remain the same. The voxelization parameters for the indoor experiments are 0.5m for the ground voxels and 1m for the off-ground points. It is obvious that with the constraints of the ground planes, the algorithm has higher accuracy when looking at the average position errors directly(0.27m) compared with KISS-ICP, which is an ICP-based LiDAR odometry method. However, it is also worth noticing that M-GCLO is more robust than its counterparts when the ground planes are maintained and optimized, unlike MULLS(where the position errors would be large in door and hall environments). LeGO-LOAM shows good results on M2DGR because it optimizes one ground plane in the mapping process, and the ground floor in the indoor scenes is basically flat. Lastly, the indoor experiments also showed the adaptability of M-GCLO, which can not only behave well on large-scale outdoor datasets but also achieve robustness and good accuracy on small-scale indoor datasets.

4. Conclusions

This paper proposes the M-GCLO(Multiple Ground Constrained LiDAR Odometry) algorithm. Raw point clouds are segmented into ground and off-ground points and then voxelized into different maps, respectively. From each ground voxel, we fit and parameterize a plane and then optimize the point-to-plane distances continuously. For the non-ground points, the point-to-distribution residuals are computed based on the NDT voxel maps. Besides, we formulate the error propagation process from each point's uncertainty to residuals' uncertainties and weigh the residuals based on that. The experiment results on KITTI and M2DGR show that M-GCLO can achieve better localization accuracies than other state-of-the-art LiDAR odometry methods in both outdoor and indoor environments. Meanwhile, M-GCLO demonstrates higher robustness with the multiple constraints from the ground planes. A future direction would be using ground constraints to detect loop closure to further minimize the pose errors and improve the point cloud quality.

Acknowledgements

This research has been supported by funding of Prof. Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs.

References

Behley, J., Stachniss, C., 2018. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments., in: Robotics: Science and Systems. p. 59.

Blaser, S., Cavegn, S., Nebiker, S., 2018. Development of a portable high performance mobile mapping system using the robot operating system. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 4, 13–20.

Chen, P., Shi, W., Bao, S., Wang, M., Fan, W., Xiang, H., 2021. Low-drift odometry, mapping and ground segmentation using a backpack LiDAR system. IEEE Robot. Autom. Lett. 6, 7285–7292.

Gao, X., Wang, Q., Gu, H., Zhang, F., Peng, G., Si, Y., Li, X., 2021. Fully automatic large-scale point cloud mapping for low-speed self-driving vehicles in unstructured environments, in: 2021 IEEE Intelligent Vehicles Symposium (IV). pp. 881–888.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361.

He, D., Xu, W., Chen, N., Kong, F., Yuan, C., Zhang, F., 2023. Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry. Adv. Intell. Syst. 2200459.

Koide, K., Miura, J., Menegatti, E., 2019. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. Int. J. Adv. Robot. Syst. 16, 1729881419841532.

Li, K., Li, M., Hanebeck, U.D., 2021. Towards high-performance solid-state-lidar-inertial odometry and mapping. IEEE Robot. Autom. Lett. 6, 5167–5174.

Liu, X., Zhang, L., Qin, S., Tian, D., Ouyang, S., Chen, C., 2019. Optimized loam using ground plane constraints and segmatch-based loop detection. Sensors 19, 5419.

Liu, Z., Zhang, F., 2021. Balm: Bundle adjustment for lidar mapping. IEEE Robot. Autom. Lett. 6, 3184–3191.

Magnusson, M., Lilienthal, A., Duckett, T., 2007. Scan registration for autonomous mining vehicles using 3D-NDT. J. F. Robot. 24, 803–827.

Nagy, B., Benedek, C., 2018. Real-time point cloud alignment for vehicle localization in a high resolution 3D map, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. p. 0.

Pan, Y., Xiao, P., He, Y., Shao, Z., Li, Z., 2021. MULLS: Versatile LiDAR SLAM via multi-metric linear least square, in: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11633–11640.

Shan, T., Englot, B., 2018. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4758–4765.

Su, Y., Wang, T., Shao, S., Yao, C., Wang, Z., 2021. GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain. Rob. Auton. Syst. 140, 103759.

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., Stachniss, C., 2023. Kiss-icp: In defense of point-to-point icp--simple, accurate, and robust registration if done the right way. IEEE Robot. Autom. Lett. 8, 1029–1036.

Wang, H., Wang, C., Chen, C.-L., Xie, L., 2021. F-loam: Fast lidar odometry and mapping, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4390–4396.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. Photogramm. Remote Sens. 105, 286–304.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. Fast-lio2: Fast direct lidar-inertial odometry. IEEE Trans. Robot. 38, 2053–2073.

Yin, J., Li, A., Li, T., Yu, W., Zou, D., 2021. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots. IEEE Robot. Autom. Lett. 7, 2266–2273.

Yokozuka, M., Koide, K., Oishi, S., Banno, A., 2021. LiTAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with KL-divergence, in: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11619–11625.

Yuan, C., Xu, W., Liu, X., Hong, X., Zhang, F., 2022. Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry. IEEE Robot. Autom. Lett. 7, 8518–8525.

Zhang, Z., Scaramuzza, D., 2018. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7244–7251.

Zheng, F., Liu, Y.-H., 2019. Visual-odometric localization and mapping for ground vehicles using SE (2)-XYZ constraints, in: 2019 International Conference on Robotics and Automation (ICRA). pp. 3556–3562.

Zheng, F., Tang, H., Liu, Y.-H., 2018. Odometry-vision-based ground vehicle motion estimation with SE (2)-constrained SE (3) poses. IEEE Trans. Cybern. 49, 2652–2663.