S & M 3127

Trajectory Correction Model for Registering City-scale Mobile Laser Scanning Data

Baolin Zhao,* Xiliang Sun, and Kang Liu

Research and Development Center, Beijing Greenvalley Technology Co., Ltd.,
No. 8, Dongbeiwang West Rd, Haidian District, Beijing 10094, China

(Received October 25, 2022; accepted December 8, 2022)

Keywords: geometric feature, global optimization, mobile laser scanning, multiple metrics, point cloud registration

Mobile laser scanning (MLS) systems with light detection and ranging (LiDAR) sensors, global navigation satellite system (GNSS) receivers, and inertial measurement unit (IMU) sensors have been widely used in applications such as smart city facility censuses and high-definition mapping. However, because of the complexity of urban environments, there are often decimeter-to-meter-level position deviations between multiple scanning data of the same area. To solve this problem, we propose a trajectory correction model for registering city-scale MLS point cloud data. First, the proposed model segments the trajectory in terms of the data accuracy and then segments the data with segmented subtrajectories while maintaining the relationship of matching pairs in overlapping areas. Second, the proposed model transforms the matching pairs based on multiple metrics registration, which uses the poles and planar feature points extracted from the point cloud data using local geometric features. Finally, the global pose optimization method is used to improve the consistency of the MLS point clouds. In data registration experiments on different urban scenes, the proposed method performed well with high robustness and decreased the position deviations by 70%.

1. Introduction

Mobile laser scanning (MLS) is an accurate and efficient tool for acquiring 3D data of cities, and has been increasingly used in various urban applications, such as urban road inspection, 3D digital city modeling, and urban environment monitoring.(1) MLS systems are usually equipped with global navigation satellite system (GNSS) receivers and inertial measurement unit (IMU) sensors to provide the information on position and orientation required for georeferencing point cloud data from light detection and ranging (LiDAR) sensors. However, GNSS-IMU integrated solutions do not perform consistently in complex urban areas owing to the influence of GNSS multipath effects next to tall buildings, the loss of signal in GNSS-denied environments (e.g., tunnels), and the accumulation of positioning error.(2)

*Corresponding author: e-mail: bill@outlook.com

https://doi.org/10.18494/SAM4196

Current methods for improving MLS data accuracy can be categorized into two groups: data-driven approaches and model-driven approaches.(3,4) In recent years, model-driven approaches have been widely applied in MLS data registration because of the advantage of not needing control points. For example, a variant iterative closest point (ICP) algorithm was used to register point clouds, and the least squares method under the condition of absolute position and registration constraint was used to improve trajectory accuracy.(5,6) A time-variant reference transformation model that combines calibration points was used to adjust the systematic bias resulting from the positional and orientation errors.(7) An error time-variant model combined with a two-step ICP algorithm was used to correct the trajectory error. In this combined model, the point clouds were segmented evenly along the time dimension.(8) A coarse-to-fine point cloud registration process was adopted by using semantic feature points at different scales.(9) A set of joining loops for global optimization was constructed to produce a globally consistent and accurate point cloud.(10) A marker-free correction method that combines feature-descriptor-based local point cloud registration and two-stage global optimization was used to reduce the inconsistency of point clouds.(11)

However, model-driven approaches still face enormous challenges in the application of complex scene mapping at a city scale. On the one hand, existing approaches based on point cloud registration or the time-variant model with the same time intervals are unable to reduce errors in complex scenes, and their robustness has rarely been evaluated at a city scale. On the other hand, few studies have prioritized the optimization process in data registration, which is an important step for improving the registration accuracy at a city scale.

In this study, we developed a trajectory correction model for city-scale MLS data registration and experimentally validated its effectiveness and robustness. To achieve this, we used an adaptive trajectory segmentation strategy that segments the point cloud by taking the error distribution and overlapping rate into consideration, and we developed a pairwise point cloud transformation method using multiple metrics. Finally, the global pose optimization method combining six-degrees-of-freedom translation and rotation was used to improve the overall point cloud consistency.

2. Materials and Methods

The proposed trajectory correction model includes three key steps (Fig. 1): adaptive segmentation and matching, pairwise point cloud transformation using multiple metrics, and global trajectory optimization. Each step is detailed in the following sections.

2.1 Adaptive segmentation and matching

In this study, we assumed that point cloud deformation does not exist within a segment, and we corrected the nonrigid transformation between segments. Therefore, an adaptive trajectory segmentation strategy that considers the error distribution and overlap rate of MLS data was used to segment point cloud data. To achieve this, we first segmented the MLS trajectory following a distance-based method.(5) However, this method may lead to multiple consecutive

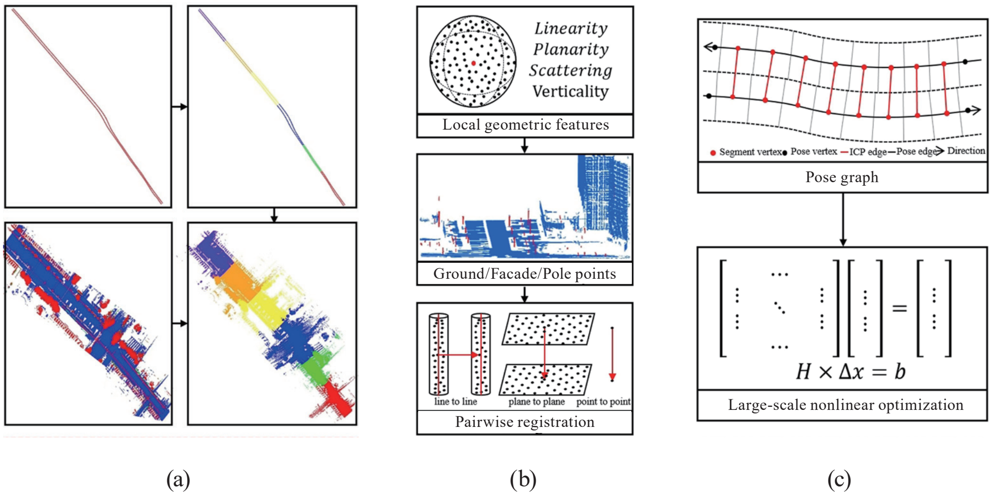

Fig. 1. (Color online) (a) Adaptive segmentation, (b) pairwise transformation, and (c) global optimization.

and short trajectory segments when a car turns at an intersection, where the number of features is limited. Therefore, a segment-merging process was implemented on continuous trajectory segments of less than 20 m.

Using the segmented trajectory, we calculated the bounding box of each segment and checked whether the bounding box of one segment intersected with that of its previous segment along the time dimension. If they intersected, we calculated the length of the overlapped trajectory and treated the two segments as an initial matching pair of segments if the length exceeded 25 m. Moreover, because a matched segment cannot be treated as rigid if its length is excessive, we divided a segment with a length of over 40 m into multiple matching pairs. Additionally, we calculated the distance between the two segments of each matching pair in all repeated matching pairs and retained the pairs with the smallest distance to eliminate redundant matches. The point cloud was segmented on the basis of the time stamp of the trajectory segments.

2.2 Pairwise transformation using multiple metrics

The ICP algorithm is the most commonly applied method for point cloud data registration. Under the hypothesis of an approximately accurate initial transformation, the ICP algorithm considers the closest source and target points as the corresponding points and iteratively solves the point-to-point distance to optimize the rigid transformation until convergence is reached.(3) However, the efficiency of the ICP algorithm is low because of the high point density of MLS data. Additionally, the algorithm may fail owing to the large number of moving objects in urban scenes. Therefore, given the large number of poles and planar objects in a city, we introduced a feature-based point cloud registration method using multiple error metrics.

2.2.1 Local geometric features

For each point, its neighborhood points within a radius are searched for. Principal component analysis is used to obtain three eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ and their eigenvectors (v_1, v_2, v_3), where the eigenvalues are in descending order. Then, four geometric features (linearity, planarity, scattering, and verticality) are used to describe the local geometric features of the point, as given in Eq. (1). At the same time, the normal direction and the principal direction are retained for eigenvectors v_3 and v_1, respectively.(12)

$$D_j^L = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}}, D_j^P = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}}, D_j^S = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}}, D_j^V = \frac{u_3}{\| u \|}, \left(u_i = \sum_{j=1}^{3} \sqrt{\lambda_j} \left| v_j \right|_i, i = 1,2,3 \right) \quad (1)$$

Geometric features are computed in the radius range of r_{min} to r_{max}. The neighborhood radius R_j for the feature calculation of each point P_j is selected to minimize the Shannon entropy E_j.(13)

$$E_j = -D_j^L \ln\left(D_j^L \right) - D_j^P \ln\left(D_j^P \right) - D_j^S \ln\left(D_j^S \right) \quad (2)$$

$$R_j = \underset{r_{min} \le r \le r_{max}}{\arg\min} \; E_j^r \quad (3)$$

If $D_j^L \gg D_j^P$, D_j^S and $D_j^V > D_{poles}^V$, the dimensional label of point P_j is 1, which means that the point is a pole-like point with verticality more than threshold D_{poles}^V.

If $D_j^P \gg D_j^L$, D_j^S and $D_j^V > D_{facades}^V$, the dimensional label of point P_j is 2, which means that the point is a facade-like point with verticality more than threshold $D_{facades}^V$.

If $D_j^P \gg D_j^L$, D_j^S and $D_j^V < D_{ground}^V$, the dimensional label of point P_j is 3, which means that the point is a ground-like point with verticality less than threshold D_{ground}^V.

The expression $D_j^L \gg D_j^P$, D_j^S above means that D_j^L is ten times larger than D_j^P and D_j^S.

2.2.2 Multiple error metrics

Multiple error metrics exist during the registration process, including point-to-point, point-to-line, and point-to-plane errors. The point-to-point distance error metric $dP2P\left(P_i, P_j\right)$ can be computed using Eq. (4), where P_i, P_j are corresponding points in the source and target point clouds, respectively.

$$dP2P\left(P_i, P_j\right) = \sqrt{\left(P_i - P_j\right)^2} \quad (4)$$

The point-to-line distance error metric $dP2Line\left(P_i, L_j\right)$ can be computed as

$$dP2Line\left(P_i,L_j\right)=\frac{(P_i-L_{j,1})\times(P_i-L_{j,2})}{\|L_{j,2}-L_{j,1}\|}. \tag{5}$$

The point-to-plane distance error metric $dP2Plane\left(P_i,P_j\right)$ can be computed using Eq. (6), where P_j^n is the normal direction of P_j.

$$dP2Plane\left(P_i,P_j\right)=(P_i-P_j)\cdot P_j^n \tag{6}$$

The angle distance $d\theta(L_i, L_j)$ of two line segments is defined by Eq. (7), where $\sin(\theta(Li, Lj))$ is the sine value of the angle between the directions of the two lines.(14)

$$d\theta\left(L_i,L_j\right)=\min\left(\|L_i\|,\|L_j\|\right)-\sin\left(\theta\left(L_i,L_j\right)\right) \tag{7}$$

Since the extracted pairs of lines are not exactly parallel, it is necessary to quantify the effect of the angle. Finally, the line-to-line distance error metric $dLine2Line\left(L_i,L_j\right)$ is defined by

$$dLine2Line\left(L_i,L_j\right)=\sqrt{\left(d\theta\left(L_i,L_j\right)\right)^2+\left(dP2Line\left(P_i,L_j\right)\right)^2}. \tag{8}$$

Similarly, the plane-to-plane distance error metric $dPlane2Plane\left(P_i,P_j\right)$ is defined by

$$dPlane2Plane\left(P_i,P_j\right)=\sqrt{\left(d\theta\left(P_i^n,P_j^n\right)\right)^2+\left(dP2Plane\left(P_i,P_j\right)\right)^2}. \tag{9}$$

The registration units selected in this paper are pole-like points and plane-like points. Therefore, we simultaneously use the line-to-line and plane-to-plane error metrics to minimize the residual equation and iteratively obtain the rigid transformation matrix.

2.2.3 Pairwise registration

The ICP algorithm includes three parts: calculation of the initial transformation matrix, search for corresponding points, and iterative minimization of the multiple error metrics defined in the previous section.

In fact, the deviations of MLS point cloud data are mainly in the positions and small angles. Therefore, we use the position offset of the center point of the trajectory pair as the initial translation value and the angle between the oriented bounding box main axes of the trajectory pair in the *XY* coordinate system as the initial heading rotation angle. Then we obtain the initial transformation matrix.

The corresponding point pair is obtained by using a *k*-dimensional (*k*-d) tree to search the nearest point of the same-dimensional label in the source and target point clouds with a distance less than 1 m. Finally, the algorithm is applied to each pair to obtain the rigid transformation matrix and the root mean square error (RMSE) e_{rmse} of the distance between the corresponding points after registration. e_{rmse} is computed using Eq. (10), where N is the number of corresponding points.

$$e_{rmse} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(dLine2Line\right)_i^2 + \left(dPlane2Plane\right)_i^2} \tag{10}$$

2.3 Global trajectory optimization

The relative constraint relationship between each pair of overlapping areas was obtained by the previous steps. The next step aims to improve the accuracy of the point cloud using global optimization.

The global pose graph $G\left(x,c\right)$ defined as Eq. (11) is composed of the variable node x to be solved and the constraint relationship c between the nodes, where $\square = \left({}_1^T, \cdots, {}_n^T \right)^T$ is a series of pose nodes in the trajectory, Ω_{ij} is the weight matrix of the constraint relationship between the ith and jth poses, and c_{ij} is their relative pose. If the ith and jth poses are continuous pose nodes, then c_{ij} is the relative pose obtained from the initial trajectory. If these pose nodes are a pair of matching pose nodes in the overlapping area, then c_{ij} is the relative pose obtained from local registration. $e(x_i,\ x_j,\ c_{i,j})$ is the error function used to measure the difference between the optimized and observed values. Finally, the global pose graph optimization can be regarded as the nonlinear least squares problem in Eq. (12).(15)

$$G\left(x,c\right) = \sum_{i,j} e\left(x_i, x_j, c_{ij}\right)^T \Omega_{ij} e\left(x_i, x_j, c_{ij}\right) \tag{11}$$

$$x^* = \arg\min_{x} G(x,c) \tag{12}$$

The weight matrix Ω_{ij} is determined by the variance of the observed values of the node pose. Position and attitude errors are small in open areas and large in streets with dense trees or areas with high buildings. These errors can be obtained from the trajectory postprocessing software. Thus, Ω_{ij} is defined as

$$\Omega_{ij} = I / \left(\sigma_{pose}^2 \cdot e_{ij}\right), \tag{13}$$

where I is the identify matrix; e_{ij} is 1.0 or e_{rmse} in Eq. (10). σ_{pose}^2 is the variance of the positional and rotational errors. Finally, we use the Ceres Solver(16) library to solve this large-scale nonlinear least squares problem, obtain the optimized pose of each node, and regenerate the point cloud data based on the optimized trajectory.

The pose node x_k is the vector defined as Eq. (14), where $x_{k0} = \left(X_{k0}, Y_{k0}, Z_{k0}, R_{k0}, P_{k0}, H_{k0}\right)$.

$$x_k = x_{k0} + \Delta_k \tag{14}$$

The error Δ_k is the vector defined as

$$\Delta_k = \left(\Delta T_{x,k}, \Delta T_{y,k}, \Delta T_{z,k}, \Delta R_{r,k}, \Delta R_{p,k}, \Delta R_{h,k}\right), \tag{15}$$

$$\begin{cases} \Delta T_{x,k}(t) = a_{1,k} + b_{1,k}(t - t_0) \\ \Delta T_{y,k}(t) = a_{2,k} + b_{2,k}(t - t_0) \\ \Delta T_{z,k}(t) = a_{3,k} + b_{3,k}(t - t_0) \\ \Delta R_{r,k}(t) = a_{4,k} + b_{4,k}(t - t_0) \\ \Delta R_{p,k}(t) = a_{5,k} + b_{5,k}(t - t_0) \\ \Delta R_{h,k}(t) = a_{6,k} + b_{6,k}(t - t_0) \end{cases}, \tag{16}$$

where k is the index of a pose point. $a_{1,k}$, $a_{2,k}$, $a_{3,k}$, $a_{4,k}$, $a_{5,k}$, $a_{6,k}$ and $b_{1,k}$, $b_{2,k}$, $b_{3,k}$, $b_{4,k}$, $b_{5,k}$, $b_{6,k}$ are the error function coefficients obtained by the optimization procedure. □$T_{x□k}$, $T_{y\ k}$, $T_{z\ k}$ are the translation errors in Δt in the *X*, *Y*, and *Z* directions, respectively. $\Delta R_{r,k}, \Delta R_{p,k}, \Delta R_{h,k}$ are the rotation angle errors in Δt around the roll, pitch, and heading directions, respectively.

2.4 Datasets and experiment

The data used in this study were acquired with a Greenvalley MLS system called LiMobile. The parameters of LiMobile are shown in Table 1.

A GNSS base station was set up at the same control point to eliminate the absolute position error of multiple scans. Therefore, the data registration method between multiple scans was the same as that of a single scan. The dataset covers a range of 42.64 km^2 with a trajectory length of 228.86 km. The acquired data contains 409.8 GB of point clouds.

Table 1
Parameters of the MLS system.

Parameter	Value
Name	LiMobile
Laser sensor	Riegl VUX-1HA
Data acquisition frequency	300 kHz
Measurement range	1.2–1420 m
Range accuracy	5 mm
Angle measurement resolution	0.001°
GNSS	GPS, GLONASS, GALILEO, BD
IMU	Novatel SPAN CPT
IMU measurement rate	100 Hz

The top view of the point cloud data in the study area is shown in Fig. 2. Four common representative urban scenes (a high-traffic area marked as A, a street with dense trees marked as B, a highway area marked as C, and an area with multiple scans marked as D) were selected to evaluate the registration results in this study. The registration accuracy was evaluated using the RMSE before and after registration.

3. Results and Discussion

3.1 Trajectory segmentation result

The results of the trajectory adaptive segmentation and matching are shown in Fig. 3, where different colors represent different matching pairs. The length of the longest segment was 40 m, and the length of the shortest segment was 20 m. All matching pairs overlapped by more than 60%. It can be seen from the figure that the matching pairs have a high overlap rate, which provides a basis for proving the robustness of the pairwise point cloud registration method.

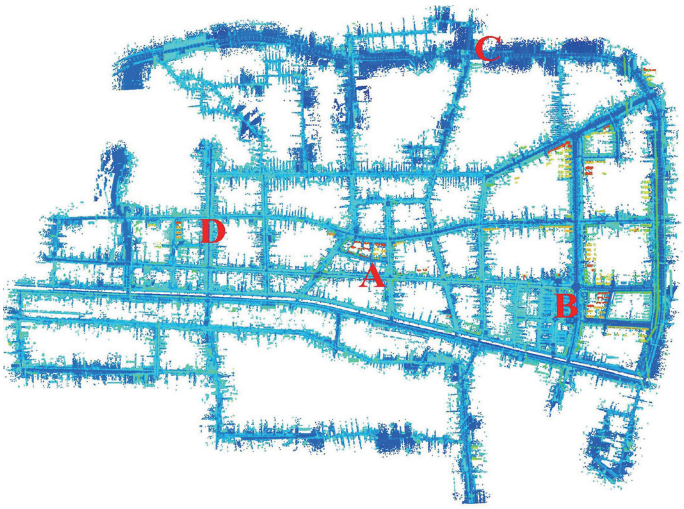

Fig. 2. (Color online) Top view of the point cloud data in this study.

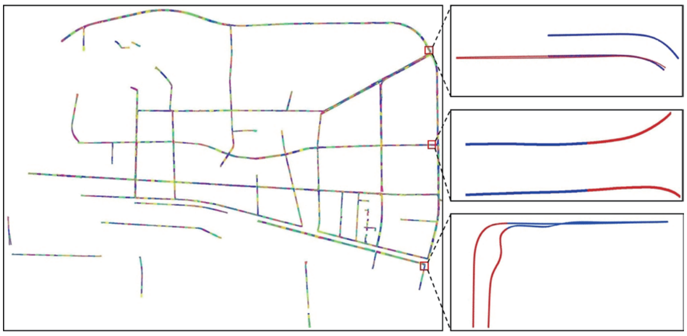

Fig. 3. (Color online) Matching pairs of trajectory segments.

3.2 Local registration result

The registration range of matching pairs is shown in Table 2. The registration method significantly decreased the misalignment of all the matching pairs. The deviations of the MLS point cloud data were mainly manifested in the vertical direction and pitch angle. The maximum misalignment of the data in the *Z* direction was 2.375 m, which was caused by the elevation spikes in the GNSS position in the street with dense trees.

3.3 Global correction result

The misalignment of the original data and the RMSE before and after registration are shown in Table 3. Extremely small values of Max XY offset and Max Z offset were ignored.

Figure 4 shows the high-traffic area, the street with dense trees, the highway area, and the area with high-frequency scanning (six scans). Figure 4(a) shows that there were sufficient heterogeneous targets in the round-trip scanning of the heavy-traffic area, and the multiple metric registration method eliminated the 0.437 m misalignment in the *Z* direction. The cross section of the streets with dense trees with a length of 145.67 m is shown in Fig. 4(b). The *Z* offsets in this area were larger (>1 m) than in the other areas. The influence of point cloud deformation was nonrigid, and the maximum offset in the *Z* direction was 1.978 m. Compared with urban roads, there are more moving objects and fewer facades or trunk points for registration in the highway area [Fig. 4(c)]. We registered the point clouds using the limited number of lamp posts and ground points and decreased the RMSE by 70%. In overlapping areas with multiple projects or multiple scans [Fig. 4(d)], only static targets such as pole-like and plane-like points could be used to register matching pairs. A pose graph optimization method was then used to improve the consistency among the multiple scans. The results suggested that the proposed correction model was effective and robust.

Improving the accuracy of MLS data is a very complex problem because various and dynamic factors are involved. The technique proposed in this paper was successfully

Table 2
Registration range of matching pairs.

	Rotation (deg)			Translation (m)		
	$\Delta\alpha$	$\Delta\beta$	$\Delta\gamma$	ΔX	ΔY	ΔZ
Min	−0.740	−0.765	−0.310	−0.617	−0.468	−2.375
Max	0.758	0.810	0.248	0.314	0.352	0.80

Table 3
Correction results of four representative scenes.

	Max *XY* offset (m)	Max *Z* offset (m)	RMSE before correction (m)	RMSE after correction (m)
A	–	0.437	0.440	0.152
B	0.294	1.978	0.778	0.150
C	–	0.395	0.456	0.136
D	–	0.300	0.410	0.148

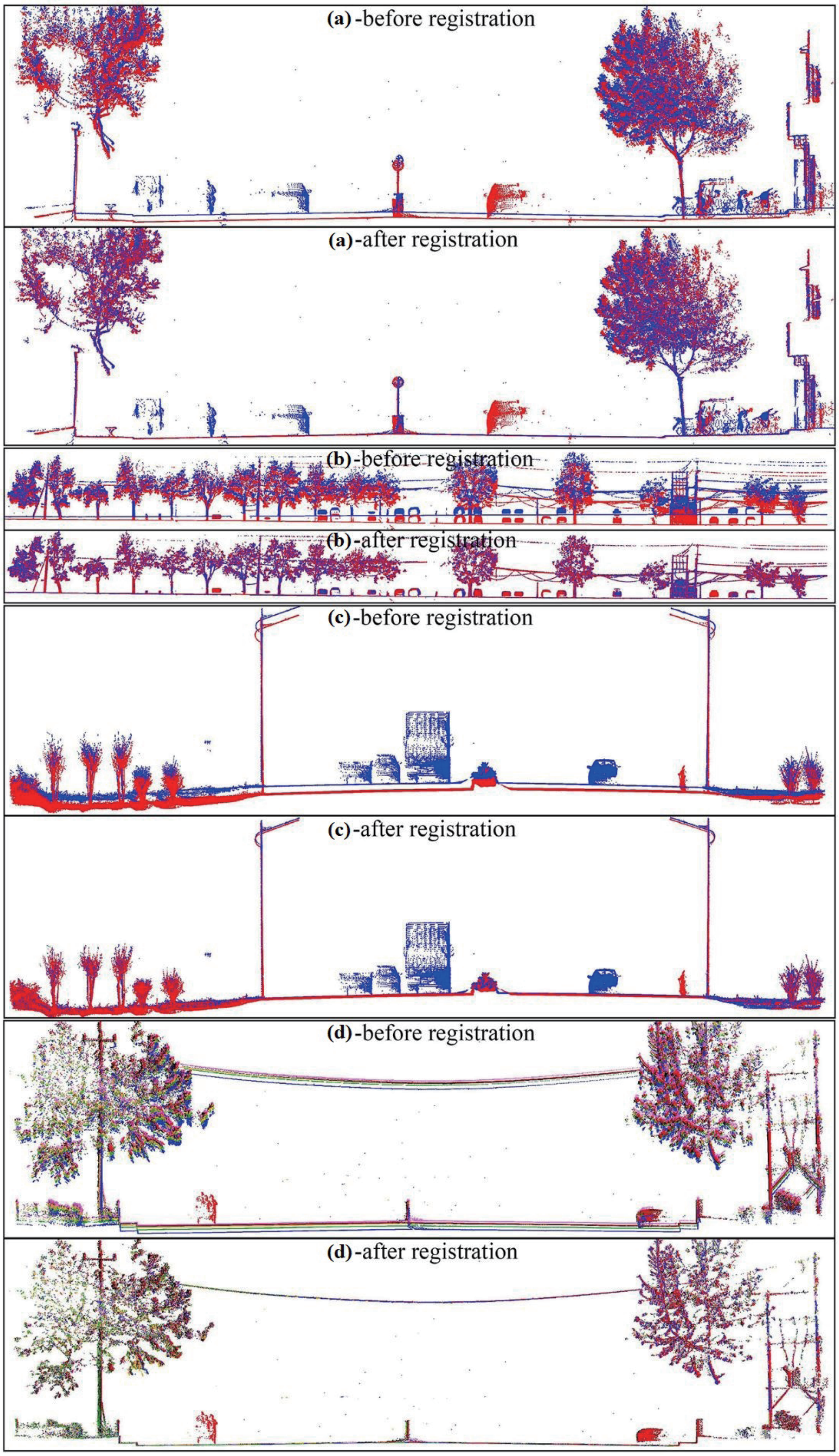

Fig. 4. (Color online) Results for various areas. (a) High-traffic area, (b) street with dense trees, (c) highway area, and (d) area scanned six times.

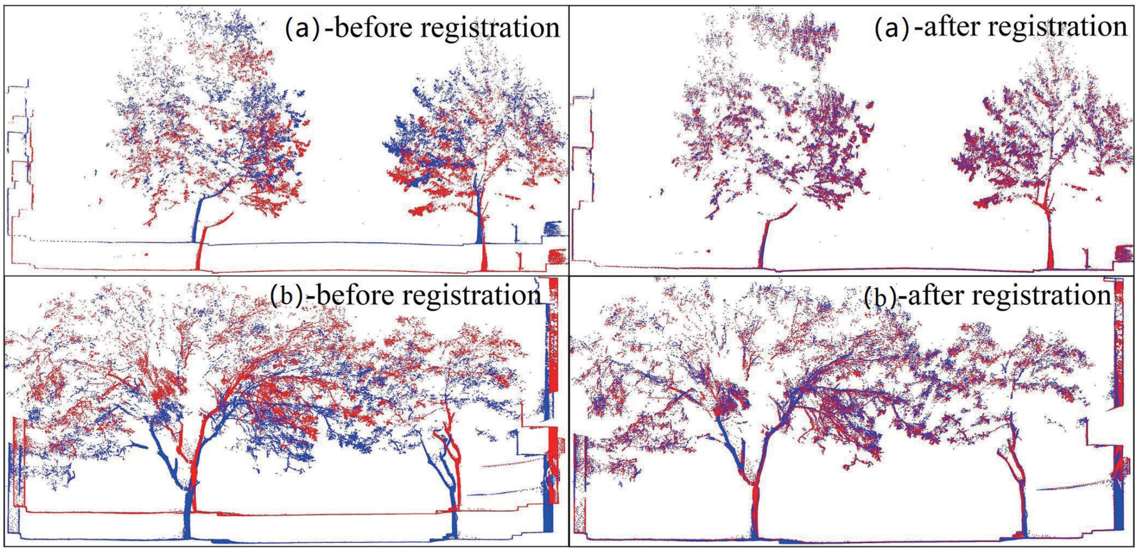

Fig. 5. (Color online) Results for (a) partially occluded area and (b) completely occluded area.

demonstrated using diverse and large scenarios at the city scale with different amounts of occlusion. The registration results of partially occluded and completely occluded streets with dense trees are shown in Fig. 5. The maximum misalignment in the *Z* direction of the partially occluded area was 1.978 m. The RMSE after registration was decreased by 80.7% compared with the original data. The maximum misalignment in the *Z* direction of the completely occluded area was 1.241 m. The RMSE after registration was decreased by 71.1% compared with the original data. In comparison, matching pairs of same length in the completely occluded area exhibited greater deformation. Therefore, we suggest that matching pairs of different lengths should be attempted in future studies to perform hierarchical registration and further improve the nonrigid deformation.

4. Conclusions

In this study, we proposed a trajectory correction model to decrease the position deviations of repeated scans for registering city-scale MLS point clouds. We collected experimental data in four types of representative urban scene using the LiMobile MLS system, which consists of a LiDAR sensor, a GNSS receiver, and an IMU sensor. A quantitative assessment using RMSE demonstrated that the model performed well with high robustness and decreased the position deviations of point clouds by 70%. The registration based on poles and planar feature points demonstrated that the model was robust in various scenes. The global optimization improved the accuracy of the final point clouds. Finally, we suggest that hierarchical registration methods and other data sources (control points, image data) should be introduced in the future to increase data quality for city-scale mapping.

References

1 Y. Wang, Q. Chen, Q. Zhu, L. Liu, C. Li, and D. Zheng: Remote Sens. **11** (2019) 13. https://doi.org/10.3390/rs11131540

2 N. Haala, M. Peter, J. Kremer, and G. Hunter: Proc. XXIst ISPRS Congress (ISPRS, Beijing, 2008) 1119. https://www.isprs.org/proceedings/XXXVII/congress/5_pdf/191.pdf

3 W. Liu, Z. Li, Y. Li, S. Sun, and M. A. Sotelo: IEEE Trans. Geosci. Remote Sens. **58** (2019) 1. https://doi.org/10.1109/TGRS.2019.2935744

4 L. Cheng, S. Chen, X. Liu, H. Xu, Y. Wu, M. Li, and Y. Chen: Sensors **18** (2018) 5. https://doi.org/10.3390/s18051641

5 A. Gressin, B. Cannnelle, C. Mallet, and J.-P. Papelard: ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. **1** (2012) 3. https://doi.org/2012.10.5194/isprsannals-I-3-117-2012

6 S. Takai, H. Date, S. Kanai, Y. Niina, K. Oda, and T. Ikeda: ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. **2** (2013) 5. https://doi.org/10.5194/isprsannals-II-5-W2-277-2013

7 J. Y. Han, C. S. Chen, and C. T. Lo: IEEE Geosci. Remote Sens. Lett. **11** (2014) 1. https://doi.org/10.1109/LGRS.2013.2252417

8 L. Yan, J. Tan, H. Liu, H. Xie, and C. Chen: Int. J. Remote Sens. **39** (2018) 6. https://doi.org/10.1080/01431161.2017.1410248

9 F. Yu, J. Xiao, and T. Funkhouser: Proc. CVPR 2015 (IEEE, Boston, 2015) 1722–1731. https://doi.org/10.1109/CVPR.2015.7298781

10 T. Shiratori, J. Berclaz, M. Harville, C. Shah, T. Li, Y. Matsushita, and S. Shiller: Proc. 3DV 2015 (IEEE, Lyon, 2015) 232–240. https://doi.org/10.1109/3DV.2015.33

11 B. Yang, Y. Li, X. Zou, and Z. Dong: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. **XLIII-B2** (2020) 347. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-347-2020

12 S. Guinard and L. Landrieu: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. **XLII-1-W1** (2017) 151. https://doi.org/10.5194/isprs-archives-XLII-1-W1-151-2017

13 A. Gressin, C. Mallet, J. Demantké, and N. David: ISPRS J. Photogramm. Remote Sens. **79** (2013) 240. https://doi.org/10.1016/j.isprsjprs.2013.02.019

14 M. Poreba and F. Goulette: Proc. MMT (Int. Symp. Mobile Mapping Technology) 2013 (MMT, Tainan, 2013). https://doi.org/10.1.1.372.834

15 T. Sakai, K. Koide, J. Miura, and S. Oishi: Proc. 2017 IEEE/SCIE Int. Symp. System Integration (IEEE, Taiwan, 2017). https://doi.org/10.1109/SII.2017.8279325

16 S. Agarwal, K. Mierle, and the Ceres Solver Team: http://ceres-solver.org (accessed April 2022).