Toward Efficient 3-D Colored Mapping in GPS-/GNSS-Denied Environments

Chenglu Wen, *Senior Member, IEEE*, Yudi Dai, Yan Xia, Yuhan Lian, Jinbing Tan, Cheng Wang, *Senior Member, IEEE*, and Jonathan Li, *Senior Member, IEEE*

***Abstract*—Efficient 3-D mapping provides useful and detailed 3-D data for many applications. In this letter, we present a multisensor calibration and mapping method, to provide highly efficient and relatively accurate colored mapping for GPS-/global navigation satellite system-denied environments. The sensor data include 3-D laser scanning point clouds and camera images. A simultaneous localization and mapping (SLAM)-assisted calibration method is first proposed for multiple multibeam light detection and ranging (LiDAR) and multiple camera calibration. An improved SLAM method with loop closure is proposed for 3-D mapping. With the proposed calibration and mapping methods, centimeter-level colored point clouds can be obtained efficiently. The proposed method was tested with both backpacked and car-mounted systems on indoor and outdoor scenes. Experimental results show the effectiveness and efficiency of the proposed calibration and mapping methods.**

***Index Terms*—Cameras, GPS-/global navigation satellite system (GNSS)-denied, light detection and ranging (LiDAR), mapping, point cloud, simultaneous localization and mapping (SLAM).**

I. Introduction

THE 3-D mapping in GPS-/global navigation satellite system (GNSS)-denied environments provides useful 3-D dense data for many applications, such as indoor disaster search and rescue, restoration of complex internal structures of buildings, and navigation in dense cities [1], [2]. Light detection and ranging (LiDAR) sensors acquire accurate 3-D data by measuring the range. Recently, low-cost LiDARs, for example, the Velodyne multibeam LiDAR sensor, have been widely used in some surveys and self-driven vehicles [3].

Recent studies in LiDAR-based simultaneous localization and mapping (SLAM) provide 3-D point cloud maps by feature-based scan registration or LiDAR odometry [4]. Meanwhile, well-explored visual SLAM methods can achieve sparse 3-D feature maps based on images. For applications, such as object recognition and scene understanding, providing both 3-D point cloud and color information for a scene is in high demand. However, very few studies have explored the efficient implementation of colored mapping in GPS-/GNSS-denied environments. Multisensor systems, which typically include LiDAR sensors and cameras, are used to acquire both point clouds and images. Some studies attempted to combine camera and LiDAR by extracting depth from LiDAR measurements for camera features in visual SLAM [5].

Regarding the extrinsic calibration of LiDAR and cameras, some works used specially designed calibration targets and manual feature selection [6]. At present, for most LiDAR sensor and multiple camera hybrid calibration methods, each camera and LiDAR sensor must be calibrated separately [7]. It is difficult to find corresponding target points or corner features from a sparse point cloud in an overlapped view of multibeam LiDAR sensors and cameras [8].

Based on our previous research [9], in this letter, we present an efficient 3-D colored mapping method for GPS-/GNSS-denied environments using multisensor data. The proposed method, including multisensor calibration and 3-D mapping, provides efficient and relatively accurate colored mapping for GPS-/GNSS-denied environments. To achieve extrinsic calibration of multibeam LiDAR sensors and multiple cameras, a SLAM-assisted calibration method is proposed. Also, an improved LiDAR-based SLAM method is proposed for 3-D mapping. The proposed calibration and mapping methods were tested with different mobile platforms on indoor and outdoor test scenes.

II. Multisensor System Calibration

A. Multisensor System

In the calibration process, a self-built backpacked multisensor system [Fig. 1(a)] is used to collect both LiDAR point cloud and camera data. The backpacked system consists of two Velodyne VLP-16 LiDAR sensors for 3-D data acquisition and a low-cost commercial panoramic camera (one wide-angle lens camera on each side) for image data acquisition.

In this system, LiDAR sensor A (X_{l1}, Y_{l1}, Z_{l1}) is mounted horizontally; LiDAR sensor B (X_{l2}, Y_{l2}, Z_{l2}) is mounted 45 below LiDAR sensor A [Fig. 1(b)]. The point cloud in the coordinate system of the LiDAR sensor is shown in Fig. 1(c). The LiDAR point cloud data, P (x, y, z), in the Cartesian coordinate system (X, Y, Z) are calculated using (1). R is the distance from P to the origin of the coordinate system. Based on our previous work [10], point cloud data of LiDAR sensor A, (P_A), and point cloud data of LiDAR sensor B, (P_B), are fused into P_f by the 4×4 transform matrix between the

Manuscript received December 1, 2018; revised February 28, 2019, April 4, 2019, and May 8, 2019; accepted May 9, 2019. This work was supported in part by the National Science Foundation of China under Grant 61771413 and in part by the Fundamental Research Funds for the Central Universities under Grant 20720170047. *(Corresponding author: Chenglu Wen.)*

C. Wen, Y. Dai, Y. Xia, Y. Lian, J. Tan, and C. Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Xiamen 361005, China (e-mail: clwen@xmu.edu.cn).

J. Li is with the Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Xiamen 361005, China, also with the Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou 350003, China, and also with the GeoSTARS Laboratory, Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2916844

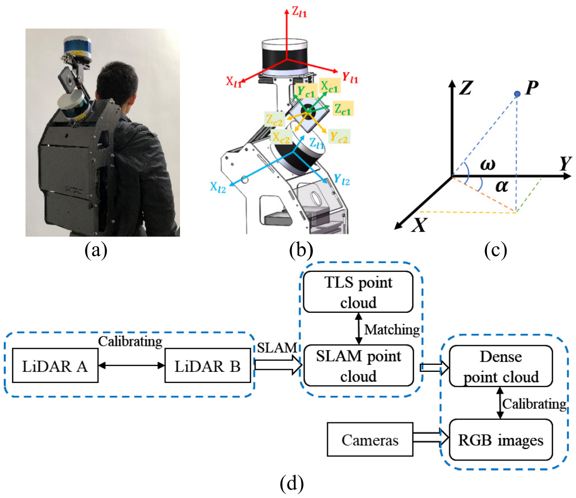


Fig. 1. Overview of multisensor system calibration. (a) Multisensor system. (b) Multisensor coordinates. (c) Coordinate of LiDAR sensor. (d) Flowchart of the calibration process.

two LiDAR sensors, (T_{cal}) (2). In addition, terrestrial laser scanning (TLS) data are introduced to bridge the calibration between LiDAR sensors and cameras. The calibration process is shown in Fig. 1(d)

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R * \cos(\omega) * \sin(\alpha) \\ R * \cos(\omega) * \cos(\alpha) \\ R * \sin(\omega) \end{bmatrix} \tag{1}$$

$$P_f = P_A + T_{\text{cal}} * P_B. \tag{2}$$

B. Intrinsic Calibration of the Cameras

The proposed method can be applied not only to cameras with regular lenses but also fisheye and wide-angle lenses. The camera internal reference model is given by

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

and (k_1, k_2, k_3), where (f_x, f_y) is the focal length of the camera, (c_x, c_y) is the position of the camera, and (k_1, k_2, k_3) is the factor of radial distortion. Considering the practicality of implementation, a camera calibration method proposed in [11] is used to determine the internal parameters and distortion factors of the camera and obtain the camera internal reference model. Specifically, when calibrating fisheye and wide-angle lens cameras, central areas close to the center point of the image are used to assure high calibration accuracy.

C. SLAM-Based Calibration Scene Mapping

For each scan of the multibeam LiDAR sensor, the acquired point cloud includes a limited description of the calibration scene. The proposed calibration method uses the SLAM-based method detailed in Section III to achieve large calibration scene data using mobile platforms. Considering that acquired point cloud data are relatively sparse, it is inaccurate to locate features directly in an original point cloud. To acquire denser points for feature extraction, a high-precision TLS (e.g., Riegl VZ 1000 [12]) scans the calibration scene and obtains the point cloud of the entire calibration scene with an accuracy of about 5 mm (in the range of 1400 m). Using the singular value decomposition (SVD) method, a coarse registration between dense TLS point cloud data and sparse SLAM-based point cloud is obtained from four pairs of manually selected matching points. Then, an iterative closest point (ICP) algorithm fine-tunes the coarse registration results. The registered dense point cloud also provides a relatively accurate estimation of the camera location in the SLAM-based point cloud coordinate system, since the accuracy of the terrestrial laser scanner is high.

D. Calibration of the LiDAR and Cameras

In our method, highly reflective boards, which are easily detected in the point clouds, are used for calibration. For camera images, a corner detector extracts the candidate edge points of each square. For point cloud data, an intensity threshold-based method first extracts the high-intensity areas of the targets. Then, region growing and plane fitting methods are applied to obtain high-intensity block candidates on the reflective boards. Four pairs of correspondent corners are manually selected from the image corner candidates and intensity corner candidates.

Denote the corner coordinates in the corrected images of each camera by (μ, ν), the correspondent corner coordinates in the point cloud by (X_l, Y_l, Z_l), and the estimation of the camera coordinates in the SLAM-based point cloud coordinate system by (X_c, Y_c, Z_c). Using (X_c, Y_c, Z_c), (μ, ν), (X_l, Y_l, Z_l) and the internal parameters of the cameras, a collinear equation is established to obtain the directional cosine elements of each camera and complete the calibration as follows:

$$\mu - c_x = -f \frac{a_1(X_l - X_c) + b_1(Y_l - Y_c) + c_1(Z_l - Z_c)}{a_3(X_l - X_c) + b_3(Y_l - Y_c) + c_3(Z_l - Z_c)} \tag{3}$$

$$\nu - c_x = -f \frac{a_2(X_l - X_c) + b_2(Y_l - Y_c) + c_2(Z_l - Z_c)}{a_3(X_l - X_c) + b_3(Y_l - Y_c) + c_3(Z_l - Z_c)} \tag{4}$$

where f is the focal length; for simplicity, f is set at $f_x + f_y/2$ in the method. For each camera, nine directional cosine elements $(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3)$ are calculated for the three angles between the SLAM-based LiDAR point cloud frame and camera frame (φ, ρ, θ) as follows in (5), as shown at the top of the next page.

III. LiDAR-Based 3-D Mapping Method

A. Feature Point Extraction

Feature points on sharp edges and planar surface patches in each frame are first selected separately for LiDAR data frame matching. Because of the radial effect, when the angle between the planar and the laser beam is small, the points on this planar may be misclassified as edge points, which eventually results in map mismatching. To solve the point misclassification problem, we improve the original feature extraction method in LiDAR Odometry and Mapping (LOAM) [4] by normalizing the distance during the smoothness calculation of the feature point. Here, the smoothness of each point, C, determines which category one point belongs to. Considering that each acquired frame includes multiple individual laser beams, the smoothness of each laser scanner point can be evaluated through the spatial relationship with its surrounding points on a scanning beam.

$$\mathrm{R} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} \cos\varphi\cos\theta - \sin\varphi\sin\rho\sin\theta & -\cos\varphi\sin\theta - \sin\varphi\sin\rho\cos\theta & -\sin\varphi\cos\rho \\ \cos\omega\sin\theta & \cos\omega\cos\theta & -\sin\omega \\ \sin\varphi\cos\theta + \cos\varphi\sin\rho\sin\theta & -\sin\varphi\sin\theta + \cos\varphi\sin\rho\cos\theta & \cos\varphi\cos\rho \end{bmatrix} \tag{5}$$

Let P_f^n be the nth fused point cloud frame of two LiDAR sensors (P_A^n, P_B^n). T_cal is the transform matrix between the two LiDAR sensors

$$P_f^n = P_\mathrm{A}^n + T_\mathrm{cal} * P_\mathrm{B}^n. \tag{6}$$

There are multiple (e.g., 16 for Velodyne VLP-16) beams in each frame in both LiDAR sensors. The scanning point of each beam is stored in clockwise/counterclockwise order. Let $P_f^{(n,l)}(\mathrm{l} \in \mathrm{Z}^+, \mathrm{l} \leq 32)$ be the lth beam in P_f^n, and the coordinates of a point $P_i(x_i, y_i, z_i)$, $p_i \in P_f^n$, be $X_{(l,i)}^n$. The smoothness, c, for each point in P_f^n is calculated as follows:

$$\boldsymbol{D(i, j)} = X_{(l,i)}^n - X_{(l,j)}^n \tag{7}$$

$$c = \left\| \frac{1}{2x} \sum_{j=i-x}^{i+x} \frac{\boldsymbol{D(i, j)}}{\|\boldsymbol{D(i, j)}\|} \right\| \tag{8}$$

where $\boldsymbol{D(i, j)}$ is a vector from point $P_i(x_i, y_i, z_i)$ to point $P_j(x_j, y_j, z_j)$, and x is the number of points on each side of point P_i (x is set at 5 in this letter). In LOAM, when calculating c, all the $\boldsymbol{D(i, j)}$ are combined without normalization, which may lead to misclassification between the planar points and edge points. In our method, the distance vector, $\boldsymbol{D(i, j)}$, is normalized as a direction vector; the mean of all direction vectors is calculated to obtain the c value. We divide each beam in a frame into six sections and select not more than 20 edge points from each section. A point is classified as an edge point with a value of c less than 0.25 or a planar point with a value of c more than 0.5. The same point selection strategies as defined in LOAM are used to avoid selecting weak edge points that are roughly parallel to the laser beam or points on the boundary of the occluded region.

B. LiDAR-Based Mapping by Feature Matching

Then, a 3-D point cloud map and the corresponding trajectory are reconstructed by matching the above-mentioned feature points. Let P_w^n be the mapping result from P_f^1 to P_f^n in the world coordinate system, and T_n be the nth transform matrix that translates P_f^n to the world coordinate system

$$\begin{cases} P_w^1 = P_f^1 \\ P_w^n = P_w^{n-1} + T_n * P_f^n. \end{cases} \tag{9}$$

Let P_{fe}^n be edge points, and P_{fp}^n be planar points in P_f^n. Let P_{we}^n be edge points, and P_{wp}^n be planar points in the world coordinate system. The goal of mapping is to obtain P_w^n

$$P_{we}^n = P_{we}^{n-1} + T_n * P_{fe}^n \tag{10}$$

$$P_{wp}^n = P_{wp}^{n-1} + T_n * P_{fp}^n. \tag{11}$$

After being transformed to the world coordinate system, for each point, P_i, in P_{fe}^n/P_{fp}^n, five edges/planar points closest to that point are found in $P_{we}^{n-1}/P_{wp}^{n-1}$. If the five points are in the same line or on the same plane, compute the distance, d_e^i/d_p^i, from P_i to the line/plane. T_k (k is the frame number) is computed recursively as follows:

$$\begin{cases} d_e^i = Dist\left(P_i, P_{fe}^k, T_k, P_{we}^{k-1}\right) \\ d_p^i = Dist\left(P_i, P_{fp}^k, T_k, P_{wp}^{k-1}\right) \\ T_k = \arg\min_{T_k} \left(\sum_{|P_{fe}^k} d_e^i + \sum_{\left|P_{fp}^k\right|} d_e^i\right) \end{cases} \tag{12}$$

where Dist(·) is the function to compute the distance between point P_i and its corresponding line or plane. During this projection, each d_e^i or d_p^i is recalculated for each iteration of T_k. The initial value of T_k can be set as T_{k-1} or computed by linear interpolation. Finally, P_w^n is achieved when the transformation from T_1 to T_n is obtained.

TABLE I
CALIBRATION ERRORS OF TWO SCENES

Scene	x_mean_error		y_mean_error		RMS_error	
	pixel	mm	pixel	mm	pixel	mm
Indoor scene	3.26	9.42	1.80	7.38	4.45	22.54
Outdoor scene	2.19	18.84	2.80	23.09	3.89	38.63

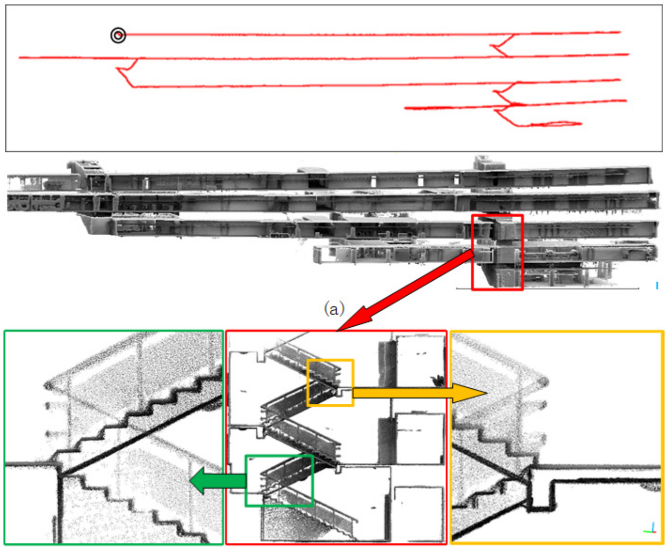
C. Loop Closure and Map Optimization

Loop closures are detected based on the Euclidean distance between the origins of two frames in the world coordinate system. For a certain period, if the timestamp interval of the two frames exceeds a given threshold (set at 20 s in this letter) and their spatial distance is less than a given threshold (3 m for indoor and 5 m for outdoor scenes in this letter), a possible closed loop is detected. For each frame of these two frames, a point cloud is first generated by registering its consecutive frames. Next, a Generalized-ICP [13] method registers these two point clouds to obtain a transformation, ΔT, and a registration error, ε. Then, a candidate loop closure will be labeled as true if ε is less than a given threshold. If true, ΔT along with its associated frame number will be used as the input to the map optimization process. Finally, regarding the accumulated mapping error, a G2O framework [14] optimizes the transformation from T_1 to T_n after all loop closures are detected. Finally, the 3-D map P_W^n is refreshed to obtain a consistent map.

IV. EXPERIMENTAL RESULTS

A. Calibration Experiments

A marker-based method was used for error analysis of the proposed calibration method. The average distances between the corresponding center points of the marker and the total least mean square error were calculated (see Table I). For each scene, ten groups of points were selected on the floor and walls, respectively. Results indicate that our method achieves good calibration accuracy. We observed that smaller error is calculated for the indoor scene, regarding the smaller error of manual point selection in the indoor scene with closer range. Along with 3-D mapping, the proposed calibration method


(a)

(b)

Fig. 2. Mapping results of a multifloor building scene. (a) Trajectory and map built by our method. Black circle: starting location. (b) Close look of the stairway in (a).

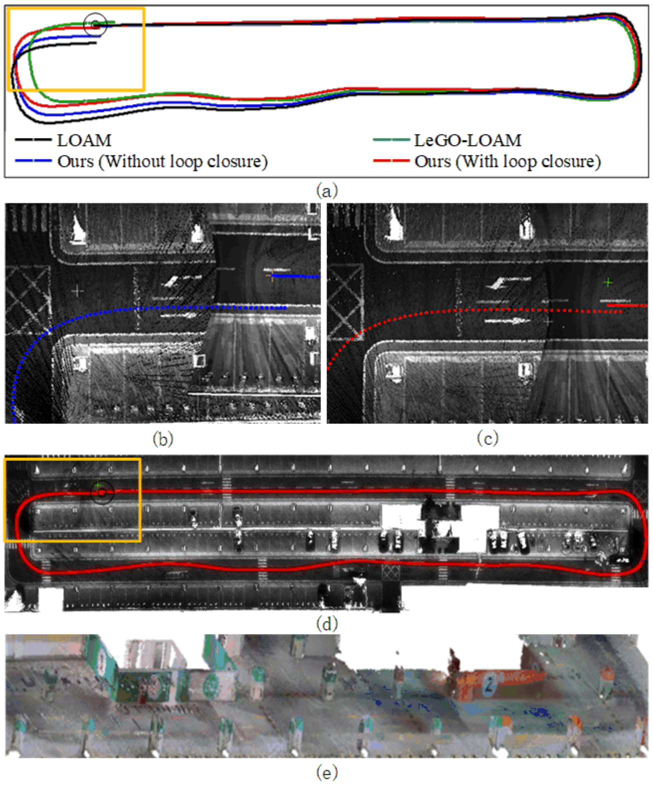
acquires a colored point cloud not only for a single scene but also a large-scale scene.

B. 3-D Mapping Experiments

1) Backpacked System for Indoor Scenes: Our mapping method was first tested on a multifloor building scene. Point cloud data and camera images of a five-floor building exterior were collected by carrying our backpacked system and walking along the corridors and staircases. We started from the left side of the fifth floor, walked to the nearest stairway, and went down to the first floor. There is no closed loop in this run. The length of the longest corridor is about 120 m. The overall run of the trajectory is about 500 m. The detailed map in Fig. 2 indicates that our algorithm performs well. The time cost of our method is about 0.67 s/frame.

Our mapping method was also tested on an underground parking garage scene with an approximate size of 200 × 50 m^2. The mapping result of this scene is shown in Fig. 3. The overall run of the trajectory is about 330 m. Because of the particular design for car parking, the walls are straight and parallel, which makes the scene appear to be a narrow, long corridor that prevents the laser beams from reaching the other side of the wall. We started and ended the run at the same location.

Two state-of-the-art methods, LOAM and Lightweight and Ground-Optimized (LeGO)-LOAM [15], were compared with our methods by measuring the gap between the starting and ending points. Fig. 3(a) shows four trajectories. The red and blue trajectories are generated by our method with and without loop closure, respectively. The black and green trajectories are generated by LOAM and LeGO-LOAM, respectively. The black, blue, green, and red trajectories have cumulative drifts of about 4.1, 2.2, 0.7, and 0.1 m, respectively. The dramatic drop of the cumulative drift demonstrates the performance of our mapping method. The time costs of our method without and with loop closure are about 0.70 and 0.75 s/frame, respectively.



(a)

(b)

(c)

(d)

(e)

Fig. 3. Mapping results of a parking garage scene. (a) Trajectories from different methods. Black circle: starting location. (b) Close look of a map built by our method without loop closure. (c) Close look of a map built by our method with loop closure. (d) Map built by our method with loop closure. (e) Example colored map result.

Fig. 3(e) shows the colored map result for the underground parking garage. The insufficient and inconsistent lighting situation results in inconsistent color distribution of the calibration results.

The averaged relative accuracies of the above building and parking garage scenes are about 0.17% and 0.23%, respectively. For the building scene, four pairs of reference points for each floor were selected to calculate accuracy. For the parking garage, 11 pairs of reference points, from both pillars and road markings, were selected to calculate the accuracy. For both building and garage scenes, ten planes were selected to calculate local precisions. The average vertical and planimetric precisions of the building are 2.79 and 2.70 cm, respectively. The average vertical and planimetric precisions of the parking garage are 3.66 and 2.07 cm, respectively.

2) Outdoor Scene Tests: The proposed calibration and mapping methods were also tested on a car-mounted multisensor system, consisting of a Velodyne HDL-32E LiDAR sensor and a Velodyne VLP-16 LiDAR sensor for 3-D data acquisition, and four fisheye lens cameras. The Velodyne HDL-32E LiDAR sensor has a typical accuracy range of 2 cm. The driving speed during the test is about 25–35 km/h.

A part of Haiyun campus of Xiamen University was selected as an outdoor mapping test scene. The data collecting run contain two closed loops. The overall run of the trajectory is about 650 m. As shown in Fig. 4, we started from the point "0" location and ended at the point "1." The order of the data acquisition run is 0-1-2-3-4-2-3-1. In Fig. 4(a), the trajectories in red and blue are generated by our method

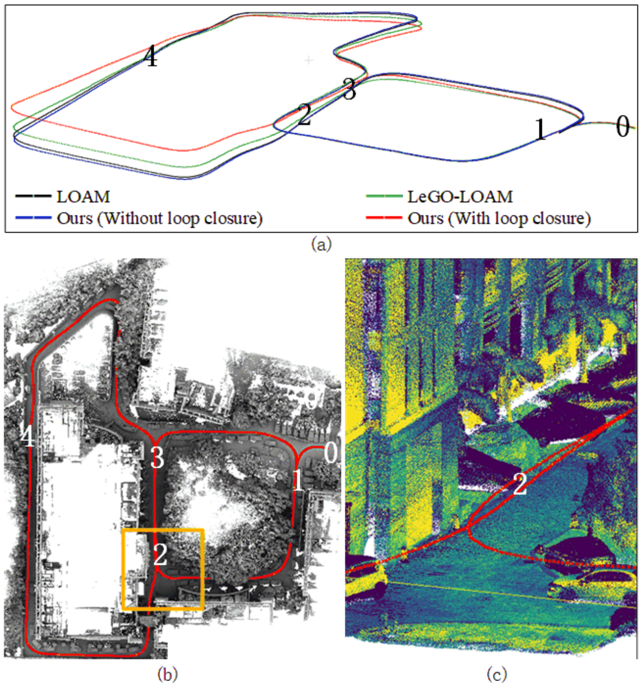


Fig. 4. Mapping results of an outdoor scene. (a) Trajectories of different methods. The number “0” is the starting location. (b) Red trajectory generated by our method with loop closure. (c) Close look of the orange block in (b).

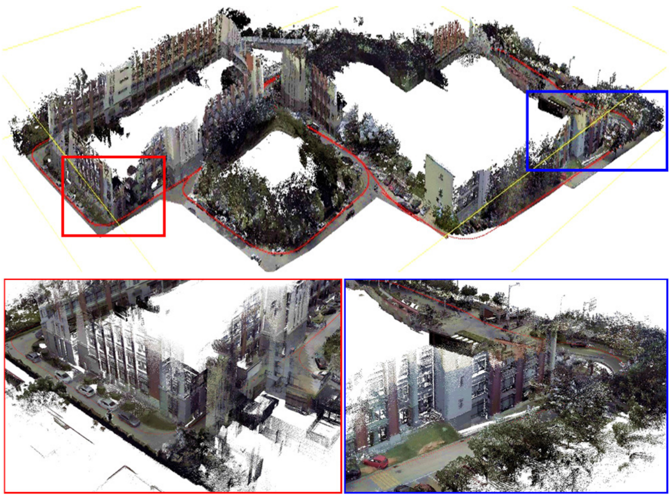


Fig. 5. Colored map results of Haiyun Campus.

with and without loop closure, respectively. The trajectories in black and green are generated by LOAM and LeGO-LOAM, respectively. The trajectories in blue and red are generated by our methods with and without loop closure, respectively. At point “2” and point “3,” the cumulative drifts generated by our method without loop closure are about 4.5 and 1.2 m, respectively. LOAM achieved similar results to ours. The cumulative drifts generated by LeGO-LOAM at these two points are about 3.0 and 2.2 m, respectively. The cumulative drifts of our method (red trajectory) dropped to less than 0.12 m after loop closure. Moreover, a crossroad [Fig. 4(c)], which we went through twice, presents clean data with very small misregistration.

Ten planes in the map were selected to calculate the local vertical and planimetric precisions. The vertical and planimetric precisions are about 1.77 and 2.45 cm, respectively. The time costs of our method without and with loop closure are about 0.74 and 0.82 s/frame, respectively.

Fig. 5 shows the 3-D colored map results for the Haiyun Campus of Xiamen University. The overall run of the trajectory is about 1200 m. The results show that the calibrated sensors are well fused for the outdoor scene. Considering the low-cost commercial cameras used in the system, the limited image resolution and large image distortion result in small misregistration and incompleteness of the colored point clouds.

V. Conclusion

An efficient indoor 3-D colored mapping method for GPS-/GNSS-denied environments using multisensor data was presented in this letter. In this method, a SLAM-assisted multisensor calibration method is used to achieve simultaneous multisensor calibration, and an improved LiDAR-based SLAM method with loop closure is proposed for mapping. The point misclassification problem during the feature point extraction is effectively solved by normalizing the distance during the smoothness calculation of the feature point. The accumulated error is eliminated by loop closure and map optimization. Experimental results show that our method achieves promising performance in complicated indoor and outdoor scenes.

References

[1] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments,” *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, Apr. 2012.

[2] M. F. Fallon, H. Johannsson, J. Brookshire, S. Teller, and J. Leonard, “Sensor fusion for flexible human-portable building-scale mapping,” in *Proc. IEEE/RSJ Int. Conf. (IROS)*, Oct. 2012, pp. 4405–4412.

[3] *Velodyne Products*. Accessed: Mar. 15, 2018. [Online]. Available: https://www.velodyneLiDAR.com/products.html

[4] J. Zhang and S. Singh, “Low-drift and real-time LiDAR odometry and mapping,” *Auton. Robot.*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[5] J. Graeter, W. Alexander, and L. Martin, “LIMO: LiDAR-monocular visual odometry,” in *Proc. IEEE/RSJ Int. Conf. (IROS)*, Oct. 2018, pp. 7872–7879.

[6] L. You, Y. Ruichek, and C. Cappelle, “Optimal extrinsic calibration between a stereoscopic system and a LiDAR,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 8, pp. 2258–2269, Aug. 2013.

[7] S. Bileschi, “Fully automatic calibration of LiDAR and video streams from a vehicle,” in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, (ICCV)*, Sep. 2009, pp. 1457–1464.

[8] M. Hassanein, A. Moussa, and N. El-Sheimy, “A new automatic system calibration of multi-cameras and LiDAR sensors,” *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.*, vol. 41, Jul. 2016, pp. 1–6.

[9] C. Wen, S. Pan, C. Wang, and J. Li, “An indoor backpack system for 2-D and 3-D mapping of building interiors,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 992–996, Jul. 2016.

[10] Z. Gong, C. Wen, C. Wang, and J. Li, “A target-free automatic self-calibration approach for multibeam laser scanners,” *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 238–240, Jan. 2018.

[11] D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easily calibrating omnidirectional cameras,” in *Proc. IEEE/RSJ Int. Conf. (IROS)*, Oct. 2006, pp. 5695–5701.

[12] *Riegl VZ 1000*. Accessed: Jun. 1, 2017. [Online]. Available: www.riegl.com/datasheet_vz-1000

[13] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” *Robot., Sci. Syst.*, vol. 4, no. 4, p. 435, Jun. 2009.

[14] R. Küemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proc. IEEE Intl. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.

[15] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. (IROS)*, Oct. 2018, pp. 4758–4765.