PROGRESS ON ISPRS BENCHMARK ON MULTISENSORY INDOOR MAPPING AND POSITIONING

Cheng Wang[1 *], Yudi Dai[1], Naser El-Sheimy[2], Chenglu Wen[1], Guenther Retscher[3], Zhizhong Kang[4], and Andrea Lingua[5]

[1] Fujian Key Laboratory of Sensing and Computing, School of Informatics, Xiamen University, 422 Siming Road South, Xiamen 361005, China - (cwang, clwen@xmu.edu.cn), (daiyudi@stu.xmu.edu.cn)

[2] University of Calgary, Canada - elsheimy@ucalgary.ca

[3] TU Wien - Vienna University of Technology, Austria - guenther.retscher@geo.tuwien.ac.at

[4] China University of Geosciences, Beijing - zzkang@cugb.edu.cn

[5] Polytechnic University of Turin, Italy - andrea.lingua@polito.it

KEY WORDS: Multi-sensor, Indoor, Mapping, Positioning, Benchmark Dataset, Design

ABSTRACT:

This paper presents the design of the benchmark dataset on multisensory indoor mapping and position (MIMAP) which is sponsored by ISPRS scientific initiatives. The benchmark dataset including point clouds captured by indoor mobile laser scanning system (IMLS) in indoor environments of various complexity. The benchmark aims to stimulate and promote research in the following three fields: (1) SLAM-based indoor point cloud generation; (2) automated BIM feature extraction from point clouds, with an emphasis on the elements, such as floors, walls, ceilings, doors, windows, stairs, lamps, switches, air outlets, that are involved in building management and navigation tasks ; and (3) low-cost multisensory indoor positioning, focusing on the smartphone platform solution. MIMAP provides a common framework for the evaluation and comparison of LiDAR-based SLAM, BIM feature extraction, and smartphone indoor positioning methods.

1. INTRODUCTION

Indoor environments are essential to people's daily life. Indoor mapping and positioning technologies have become in high demand in recent years. Visualization, positioning, and location-based services (LBS), routing and navigation in large public buildings, navigational assistance for disabled or aged people and evacuation under different emergency conditions are just a few examples of the emerging applications that require 3D mapping and positioning of indoor environments. SLAM-based indoor mobile laser scanning systems (IMLS) like provide an effective tool for indoor applications. During the IMLS procedure, 3D point clouds and high accuracy trajectories with position and orientation are acquired. Many efforts have been made in the last few years to improve the SLAM algorithms (Zhang & singh, 2014a) and the geometric/semantic information extraction from point clouds and images (Armeni et al., 2016a). There are still some challenges as follows: first, lack of efficient or real-time 3D point cloud generation methods of as-built 3D indoor environment; second, difficulties of building information model (BIM) features extraction in the clustered and occluded indoor environment. Also, given the relatively high accuracy, the IMLS trajectory provides a good reference or ground-truth for the low-cost indoor positioning solutions.

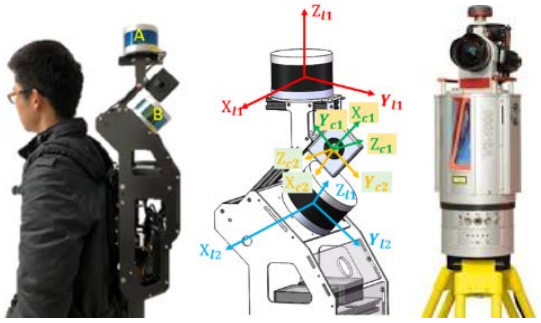
2. SENSORS AND DATA ACQUISITION

Standard datasets are critical for evaluating and comparing indoor mapping and positioning methodologies. In this project, The XBeibao II system (Wen et. al., 2016a) shown in Figure 1. (a) , which was developed by SCSC Lab in Xiamen University is used to collect the multi-sensory indoor data. The system includes two Velodyne multi-beam laser scanners, fisheye lens camera (Figure 1. (b)). Also, the navigation-related data from smart-phone built-in sensors, such as barometer, magnetometer, six degrees of freedom MEMS IMU data and Wifi information can be collected. The SLAM-based 3D point cloud of the indoor environment can also be provided using the processing software package of XBeibao. Also, the Rigel VZ 1000 (Figure 1. (c)) can provide a high accuracy point cloud as ground-truth for the indoor mapping.

2.1 Sensor setup

Our sensors are listed as below:

- 2×Velodyne VLP-16L rotating 3D laser scanner. 20Hz, 16 beams, 0.1° ~ 0.4° horizontal angle resolution, 3cm accuracy, collecting 0.3 million points/second, field of view: 360° horizontal, ±15° vertical, range: 100m.
- 1×Mi Sphere Camera. Lens components: 2 × (5 pieces spherical glass lens + 2 pieces aspheric glass lens + 2 pieces right angle glass prism), 3456*1728 @ 30fps video resolution, the field of view: 2×190°.
- 1×Mi 6 smartphone. Sensors: gyroscope, accelerometer, barometer, electronic compass, WiFi sensor, magnetometer, GPS.
- 1×Rigel VZ 1000 scanner (*www.riegl.com/datasheet_vz-1000*). Range from 1.5m up to 1200m, 5mm precision, 8mm accuracy, collecting 0.3 million points/second, with field of view of 100° vertical ×360° horizontal.



(a) (b) (c)

Figure 1. (a) The XBeibao II Multi-sensor system. (b) Multi-sensor coordinates system. (c) Riegl VZ-1000.

When collecting the data, we placed the smartphone facing up on the top of the upper LiDAR sensor. A laptop is used to control the camera and LiDARs. Also, it is used as a hotspot to connect with the smartphone to synchronize the sensors and used to store the incoming LiDAR data streams. A system operator needs to carry the laptop during the collection process.

2.2 Dataset

We collect the raw data in three different scenes; each scene is recorded more than three times with a different route. Each round of the data consisting of three parts, the raw data, the benchmarks data, and the calibration files. Only half of the complete version of the overall dataset was released for the purpose of applying in different tests. No benchmark releases for indoor LiDAR-based SLAM test and BIM feature extraction methods test. For smartphone indoor positioning methods test, there are only raw smartphone data and the calibration files.

2.2.1 Data description: A sequence of data is compressed into a file with the name format “date _number_ type.zip," where “date” is the placeholder for recording date and “number” represents the serial number of this day’s recording round. The “type” has four values——00, 01, 03 and 04, representing the complete data, the SLAM test data, the BIM feature extraction test data and the indoor positioning test data, respectively. The directory structure is shown in Figure 2.

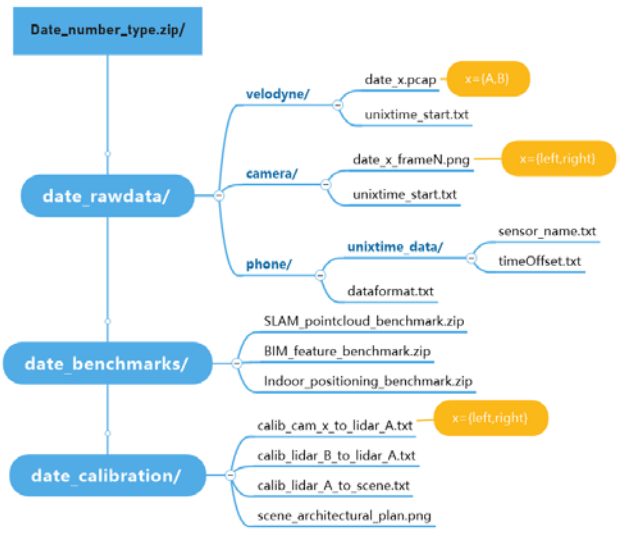


Figure 2. Structure of the dataset. Here, ‘date’, ‘number’, ‘unixtime’, ‘sensor_name’, ‘scene’ and ‘type’ are placeholders. The ‘date_x. pacp’ refers to the two LiDAR streams, and ‘date_x.mp4' refers to the two video camera streams.

The **raw data** is saved in the subdirectory “date_rawdata/,” there are three kinds of sensor mentioned above: LiDAR, camera, and smartphone.

- Velodyne LiDAR: To separate the Velodyne readings from LiDAR sensor A and LiDAR B, we name the LiDAR scans “date_A.pcap” or “date_B.pcap”, where ‘date’ is the date that collecting these data. Each point is stored with its (x, y, z) coordinate and its reflectance intensity value (r). The “unixtime_start.txt” records the starting time of this record.

- Camera: We convert the videos captured by the cameras into images according to the frame rate of the video. We name the images as “date_x_frameN.png," where "x" refers to the left or the right camera and “frameN” represents the serial frame number of this image in the raw video. The starting recording time of the first frame is saved in "camera/unixtime_start.txt."

- Smartphone: Each sensor’s recording data is saved in the file “unixtime_data/sensor_name.txt," where "unixtime” and “sensor_name” are the placeholders of the starting time of this record and this sensor’s abbreviation name, respectively. For each piece of data of different sensors, we record the Unix-timestamp. The “timeOffset.txt” records the time offsets from the phone to a local NTP server at different time. The “dataformat.txt” details the format of each file in “/phone/unixtime_data/”.

The three kinds of **benchmarks** are saved in the corresponding zip file. Files’ format and detailed description are all included in the zip file. The benchmark will be discussed in subsection 3.3.

The **calibration files** are saved in the subdirectory “date_calaibration/." Note that camera's intrinsic matrix, extrinsic matrix and distortion coefficients are all saved in "calib_cam_x_to_LiDAR_A.txt”, where “cam_x” refers to the two cameras (the camera close to left hand is left camera), and the extrinsic matrix is used to convert the camera’s coordinate system to LiDAR A’s coordinate system. The 4×4 calibration matrix converting the LiDAR B’s coordinate system to LiDAR A’s coordinate system is saved in “calib_lidar_B_to_ lidar _A.txt." For each scene, we manually select a coordinate system and origin(Figure 3) in the real world, and the 4 × 4 transformation matrix from LiDAR A’s coordinate system to the world coordinate system is saved in “calib_LiDAR_A_to_scene.txt." We also provide each scene's architectural plan in "scene_architectural_plan.png."

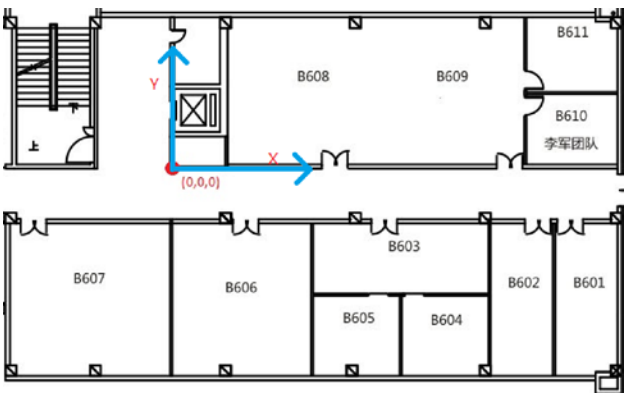


Figure 3. An example of a scene's architectural plan, the red dot on the picture is the origin we select, which is on the ground. Also, the blue arrows point the direction of X-axis and Y-axis.

The Z-axis is perpendicular to the X and Y axis and the direction is from the ground to the ceiling.

3. CHALLENGES AND METHODOLOGY

3.1 Time synchronization

In order to synchronize all the sensors, our laptop is set as a local NTP (Network Time Protocol) server, then all the sensors are connected to it to synchronize the time. The LiDAR is connected to the laptop through a network cable; the smartphone and camera's connections are through WiFi. For the LiDAR, we only get the start Unix-timestamp of the data collection. The timestamp of every point or frame is a relative time to the start Unix-timestamp. As for the Camera, we also could only get the start Unix-timestamp of the videos. The good news is that every frame's time can be obtained via interpolation according to the frame rate. However, unfortunately, frame loss sometimes happens. For the smartphone, the time can synchronize to the local NTP server during the recording, so the Unix-timestamp in every piece of data is relatively accurate. Since all data's timestamps are acquired, we can obtain the position at any time by interpolation and also can use the LiDAR's positioning result as the smartphone' positioning ground-truth.

3.2 Multi-Sensors Calibration

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 the LiDAR sensor A (Figure 1 (b)). Based on our previous work (Gong et al., 2018a), 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 two LiDAR sensors (T_{cal}). (Eq. (2)). Additionally, Terrestrial Laser Scanning (TLS) data is introduced to bridge the calibration between LiDAR sensors and cameras. The calibration process is shown in Figure 4.

$$P_f = P_A + T_{cal} * P_B \quad (1)$$

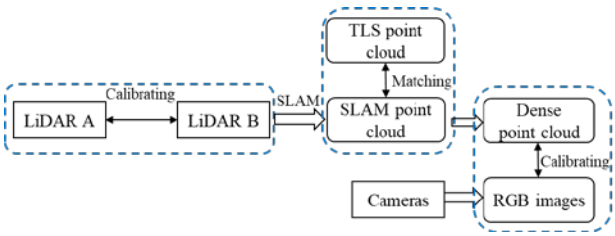


Figure 4. Flowchart of the calibration process.

3.2.1 LiDAR-to-LiDAR calibration: The calibration of the multi-LIDAR sensor is calculated recursively in the construction of the sub-map and its isomorphism constraint (Gong et al., 2018a). Assuming T_A^n is the trajectory of LIDAR sensor A at a time (*0~n*) in the mapping algorithm, P_B^n is the point cloud of LIDAR sensor B at time *n*. T_{init} is the initial coordinate system transformation between the LIDAR sensors. Calibration is the calculation of the exact calibration matrix T_{cal} by:

$$P_{near}^n = NN(M, T_A^n, P_B^n, T_{init}) \quad (2)$$

$$T_{cal} = arg \min_{T_{cal}} \sum_n T_{cal} * P_B^n - P_{near}^n \quad (3)$$

where $NN(\cdot)$ is the nearest neighbour point search algorithm. Using T_1^n and T_{init}, P_B^n is first transformed to its location at time *n* in the sub-map *M*. Then the $NN(\cdot)$ algorithm is used to search the sub-map for the nearest neighbour point set, P_{near}^n. Lastly, an environmental consistency constraint is introduced to obtain T_{cal}.

3.2.2 Camera -to-LiDAR calibration: The camera Intrinsic calibration matrix 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 of the camera and (k_1, k_2, k_3) is the factors of radial distortion. Also, Scaramuzza's the camera calibration method (Scaramuzza et al., 2006a) is used to determine the internal parameters and distortion factors of the camera and obtain the camera internal reference model.

We utilize a TLS (e.g., Riegl VZ 1000) to bridge the calibration between LiDAR sensors and cameras. By manually selected matching points between them, we can acquire the camera's extrinsic transformation $[R, T]$, where R is the 3×3 rotation matrix, and T is the 1×3 translation vector.

3.2.3 Phone-to-LiDAR calibration: We place the smartphone face up on the LiDAR A (Figure 5), and making the Y-axis parrallel to the laser beam scanning direction. Thus, the phone's coordinate system and the LiDAR's coordinate system have the same XYZ-axis direction. Then we use Rigel VZ 1000 TLS to scan the XBeibao II system and calibrate the translation (X, Y, Z) to LiDAR by manually picking the points in the high accuracy 3-D point cloud.

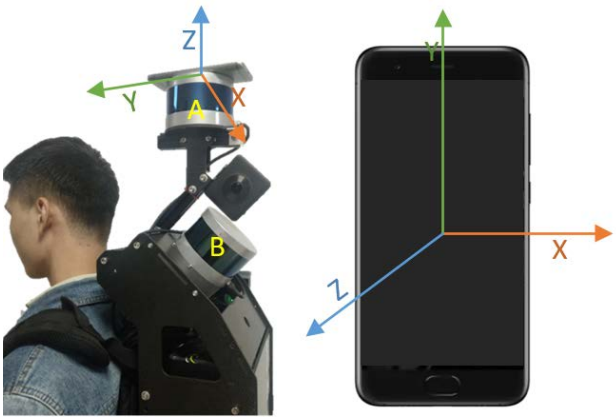


Figure 5. The smart phone's position and coordinate.

3.3 Reference data generation

For benchmark evaluation, we generated reference data from a subset of the raw data and introduced other high accuracy data.

For **SLAM-based indoor point cloud** evaluation, we built a high accuracy 3-D reference map via the data collected by Rigel VZ 1000. Firstly, we placed many high-reflection rectangle stickers on the wall and ground. Then we scanned the scene in a different position and ensured there is an overlap between adjacent sub-maps. Finally, the sub-maps were manually calibrated by picking the same sticker and other feature points via the software named RiSCAN PRO.

For **BIM feature** benchmark, we used the building line framework exacted by the wang's method (Wang et al. 2018a) and the semantic objects labeled via our manually work. We selected the building lines with their length greater than 0.1 m in

structured indoor building and saved their own two endpoints' coordinates. Fig.6 gives an example of BIM features.

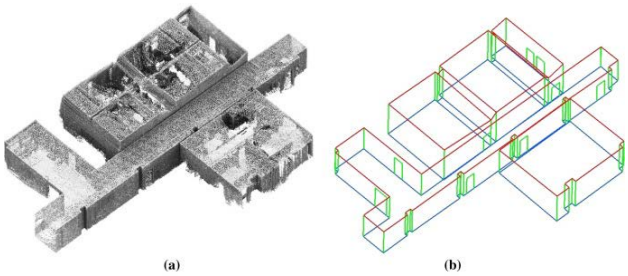


Figure 6. BIM feature examples. (a) Point cloud data. (b) BIM structure model. The green lines are doors and pillars. The red lines are ceilings. The blue lines represent the ground.

For our **Indoor positioning** evaluation, we used the LiDAR's trajectory generated by a SLAM method (Zhang & singh, 2014a) with loop closure as the reference.

3.4 Evaluation Metrics

3.4.1 SLAM-based indoor point cloud: Kümmerle (K¨ummerle at al., 2009a) proposed a metric for measuring the performance of a SLAM algorithm by considering poses of a robot during data acquisition. It is not based on the error of the trajectory end-point, but the average of all relations between poses. Geiger (Geiger at al., 2012a) extended the metric by treat the rotation and translation errors separately. Here, we do similar operation as

$$\mathcal{E}_{trans}(\delta) = \frac{1}{N}\sum_{i,j} trans(\delta_{i,j} \ominus \delta^*_{i,j})^2 \quad (4)$$

$$\mathcal{E}_{rot}(\delta) = \frac{1}{N}\sum_{i,j} rot(\delta_{i,j} \ominus \delta^*_{i,j})^2 \quad (5)$$

where N is the number of relative relations, and $\ominus$ is the inverse of a standard motion composition operator. Let $\delta_{i,j}$ be the relative transformation from pose j to pose i and $\delta^*_{i,j}$ be the reference relative relation. $trans(\cdot)$ and $rot(\cdot)$ are used to separate the translation and rotation error.

However, for indoor environments, it is hard to get the reference for the trajectory poses. However, based on K¨ummerle's method, we can apply the metric operating on the landmark locations instead of based on the trajectory poses. In this way, the relations can be determined by measuring the relative distances between landmarks.

3.4.2 BIM feature: We propose a method to evaluate the BIM feature extraction method. Here, we assume that we have the ground truth line L_g and the evaluation line L_e (the nearest midpoint to L_g' midpoint). For both lines, we calculate the corresponding direction vector v_g and v_e, midpoint p_g and p_e, and length l_g and l_e. Based on the above information, we can get the angle θ between the two lines, the distance d between the two midpoints, and the length difference Δ_l by Eq. (6). Then we set three thresholds θ_{thr}, d_{thr} and $\Delta_{l_{thr}}$. We consider the evaluation line L_e is valid only if three conditions are all met: (1) $\theta \le \theta_{thr}$, (2) $d \le d_{thr}$, (3) $\Delta_l \le \Delta_{l_{thr}}$. Finally, we can calculate the accuracy *acc* by Eq. (6).

$$\begin{cases} \theta = \arccos(\frac{v_g \cdot v_e}{|v_g||v_e|}) \\ d = \|p_g - p_e\|_2 \\ \Delta_l = \|l_g - l_e\|_1 \\ acc = \frac{N_T}{N} \end{cases} \quad (6)$$

where N_T is the true line number and N is the all ground-truth line number.

3.4.3 Indoor positioning: The approach of evaluating indoor positioning is the same as the translation evaluating in subsection 3.4.1. However, there exists a problem that the frequency of positions output by mobile phones varies with the ground-truth's frequency generated by SLAM. To solve this problem, we generate position at a time by a linear interpolation according to the timestamp. Formally, the ground truth position at time t is calculated by:

$$\mathbf{p}_t = \frac{t-t_s}{t_e-t_s}\mathbf{p}_e \oplus \frac{t_e-t}{t_e-t_s}\mathbf{p}_s \quad (7)$$

where t falls within the interval (t_s , t_e) which are two timestamps of the trajectory from the benchmark. $\mathbf{p_s}$ and $\mathbf{p_e}$ represents the ground-truth positions at time t_s and t_e respectively. And $\oplus$ denotes a compositional operator.

3.5 Examples of dataset

Fig. 7 shows some examples of this dataset. The Fig 7. (a) is a frame of the Velodyne VLP-16L LiDAR data. Different color represents the intensity of every point, the brighter color means the stronger intensity. The Fig 7. (b) shows the high accuracy data from Rigel VZ 1000, which is used as Indoor LiDAR SLAM ground truth. The (c) and (d) in Figure 7 show the BIM benchmark, and (e) and (d) show the Indoor positioning benchmark. The blue dots in (d) are trajectories generated from LiDAR SLAM method, and the yellow dots are trajectories generated by the smartphone sensor data.

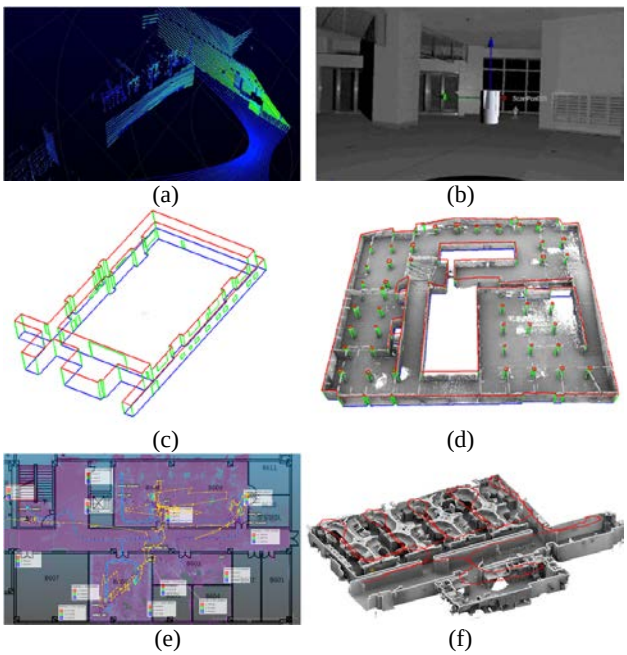


Figure 7. (a)A single frame from LiDAR stream. (b) An indoor view of Rigel VZ 1000 data. (c) BIM structure model of a circular corridor. (d) BIM structure model with its point cloud. (e) An example of the indoor positioning benchmark. (f) The ground-truth trajectory with the corresponding point cloud.

4. CONCLUSION

This paper presents the design of the benchmark dataset on multisensory indoor mapping and position (MIMAP). Each scene in the dataset contains the point clouds from the multi-beam laser scanner, the images from fisheye lens camera, the signals from MIMU and the records from the attached smartphone sensors. The benchmark dataset can be used to evaluate algorithms on: (1) SLAM-based indoor point cloud generation; (2) automated BIM feature extraction from point clouds; and (3) low-cost multisensory indoor positioning, focusing on the smartphone platform solution.

5. REFERENCES

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S., 2016a. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1534-1543).

Geiger, A., Lenz, P., Urtasun, R., 2012a. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354-3361).

Gong, Z., Wen, C., Wang, C., Li, J., 2018a. A target-free automatic self-calibration approach for multibeam laser scanners. *IEEE Trans. Instrum. Meas, 67*(1), 238-240.

Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleiner, A., 2009a. On measuring the accuracy of SLAM algorithms. *Auton. Robots, 27*(4), 387.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006a. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5695-5701).

Wen, C., Pan, S., Wang, C., Li, J., 2016a. An indoor backpack system for 2-D and 3-D mapping of building interiors. *IEEE Geosci. Rem. Sens. Lett., 13*(7), 992-996.

Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X., Li, J., 2018a. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. *ISPRS J. Photogramm. Rem. Sens., 143*, 150-166.

Zhang, J., Singh, S., 2014a. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems* (Vol. 2, p. 9).