A Framework Development for Mapping and Detecting Changes in Repeatedly Collected Massive Point Clouds

S. Yoon[a], S. Ju[a], S. Park[a], and J. Heo[a]

[a]Department of Civil and Environmental Engineering, Yonsei University, Seoul, Korea
E-mail: yoonssa@yonsei.ac.kr, jsh4907@yonsei.ac.kr, parksangyoon@naver.com, jheo@yonsei.ac.kr

Abstract –

On a construction site, progress deviations can cause fatal damages to the project and stakeholders. Therefore, accurately monitoring and managing the construction environment is essential. Efficiently detecting and recognizing the changes will be the key factor for monitor and management goals. In this research, we present a framework for detecting and recognizing changes in repeatedly collected, massive 3D point cloud data from a mobile laser scanning system. The framework mainly consists of three parts; 1) mapping system; 2) analysis system; 3) Hadoop platform. Collecting point clouds is repeatedly executed to detect the changes over different time epochs. For collecting point cloud data, a mobile laser scanning system was developed based on Robot Operating System (ROS). Detecting changes between repeatedly collected point clouds have been processed based on Hadoop platform. Finally, detected changes are then implemented to a semantic mapping process which is based on deep learning. Developed framework have potential for wide application in massive point cloud data processing, construction site monitoring, street level change detection, and facility management.

Keywords –

Mobile laser scanning (MLS); ROS; Deep learning; Change detection; Hadoop

1 Introduction

Due to the remarkable developments of sensor technologies and processing algorithms for point cloud data acquisition and registration, mapping over large spaces became more accessible than ever before. Over the last few decades, research groups focused on formulating a metric map, within which the mapping system explores the area without any information and gets self-localized using, Simultaneous Localization and Mapping (SLAM) techniques. The next step in mapping technology is the ability to produce maps directly understandable by humans, semantic mapping, which consequently integrates the robots into human environments [1].

From civil engineering perspective, metric and semantic maps from various data sources have been applied to environmental monitoring [2], building change detection ([3], [4]), disaster management ([5], [6]) and construction/infrastructure monitoring ([7]–[10]). The above mentioned applications share a primary analysis technique which is 3D change detection. 3D change detections has growing demands and possibilities for applications in environmental and civil engineering fields [11].

Despite the recent developments and possibilities, huge size of 3D point cloud data often leads to system slowdown or failure [12]. To overcome the issues, [13] and [14] introduced a distributed system to process point cloud data in a high memory cloud computing environment. However, it was costly to set the hardware environment. [14], [15] and [16] introduced a general solution for point cloud data processing based on Hadoop. In terms of cost, Hadoop had advantages, however, studies had limitations on analytic operations. Most previous studies mostly handle well-constructed data or operate on incompletely constructed data for distributed computing frameworks. Also, previous studies carry out research on only a small fraction of the whole process from beginning to the end.

In this research, practical framework including point cloud acquisition to point cloud management and analysis was conducted. The contributions of this paper is as follows: (1) ROS based 3D mapping system, which consists of three 2D Laser Range Finders (LFR) was introduced; (2) Hadoop based massive point cloud analysis system, which interacts with users using web interface was developed; (3) Deep learning algorithm was applied for point cloud based semantic segmentation and the feasibility was verified.

2 Integrated Framework Design

Our integrated framework consists of three main parts. Firstly, the mapping system based on ROS, automatically and repeatedly achieves metric maps. Second, analysis system detect changes from collected metric maps. Additionally, semantics were provided to the detected changes based on deep learning algorithm. Lastly, detecting changes and visualizing point clouds has been operated upon developed Hadoop platform.

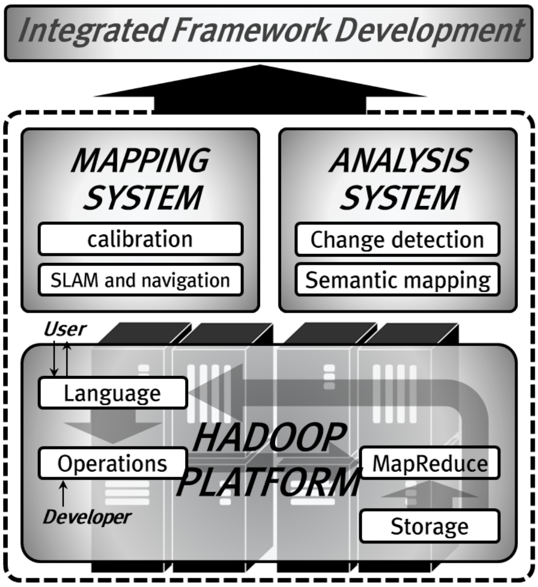


Figure 1. Integrated framework architecture

2.1 Design of Mapping System

The ROS platform is extensively used in this research since it can simply interconnect multiple programming languages and machines in seamless way. Due to this fact, ROS is the most popular robotics platform. It provides a set of tools, libraries and drivers in order to help developing robot applications, also with hardware abstraction [17]. In this research, the mapping system is based on ROS using packages for collecting data from sensors, constructing map from data and navigating through the map.

The mapping system is equipped with three 2D LFRs (UTM-30LX, Hokuyo, Osaka, Japan). It provides a scanning range of 270° with angular resolution of 0.25°. It can measure distances up to 60 meters, but without guaranteed reliability. One full scan cycle lasts for 25 milliseconds, which supplies a 40Hz measurement frequency. Pioneer 3DX mobile robot from ActivMedia Robotics, provides a stable platform for LRF sensors (Figure 2).

Over past decades, the number of SLAM algorithms were developed and the performance of the algorithms were compared . Popular SLAM algorithms such as GMapping [18], Karto SLAM [19], Hector SLAM [20], and Google Cartographer [21] were compared each other with diverse approaches ([22]–[26]). Evaluation results on every paper indicates that Google Cartographer delivers most accurate and precise results compare to others. Therefore, in this research Google Cartographer was used to construct the map, which is a robust solution on most input sequences. Cartographer provides real-time SLAM in 2D and 3D across multiple platforms and sensor configurations. It is a graph-based solution which stores a map of environment as associations for nodes and edges. Nodes represent a submap which has been created using scans, and edges represent transformations between corresponding submaps.

Using previously constructed map and navigation stack from ROS, the mapping platform autonomously navigates the given environment safely based on proper operation. The Adaptive Monte Carlo localization (amcl) [27] package was used to localize the platform itself in the previously constructed map. Additionally, the 'move_base' package [28], which links the global and local planner together, was used to accomplish navigation tasks. Standard A* path planning algorithm was used for global path planning and Dynamic Window Approach (DWA) [29] was used for local path planning. To perform navigation task efficiently, essential goal point needs to be set for our system. To achieve the goal points, morphological skeletonization [30] was applied in the 2D occupancy grid map. Among the exported candidate points from skeletonization, the goal points for the navigation stack was selected manually.

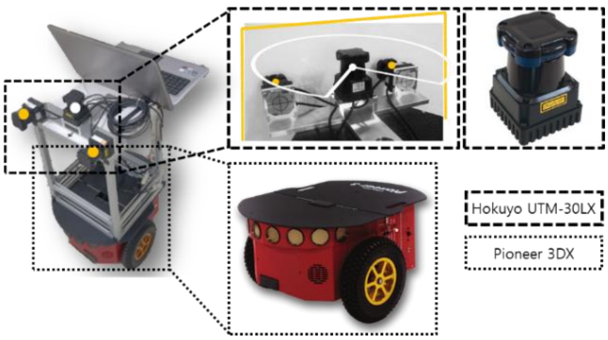


Figure 2. Mapping platform

2.2 Design of Analysis System

Semantic mapping explains the surrounding environment with various components, provides intuitive understanding of the situation to humans and bridges the gap between robots and humans. This could improve efficiency on conventional applications in infrastructure monitoring fields, since it can reduce the human labor on manually giving attributes to geometric information.

Recent developments in autonomous driving and robotics applications accelerate the necessity of point clouds semantic mapping. However, the development stage for point cloud is far behind image segmentation using Convolutional Neural Networks (CNNs). There have been only a few studies related to semantic mapping with 3D point datasets.. PointNet [31] and PointNet++ [32] are the pioneers in this area which introduces efficient and flexible ways to deal with point cloud data.

In this study, PointNet++ was implemented to analyze the geometric point cloud data. PointNet++ processes a set of points sampled in a metric space in a hierarchical manner. It first partitions the set of points into overlapping local regions by the distance metric of the underlying space. As well as extracts local features capturing fine geometric structures from small neighborhoods; such local features are further grouped into larger units and processed to produce higher level features. This process is repeated until algorithm obtains the features of whole point cloud data. PointNet++ presents more robust result regardless of the density [32].

2.3 Design of Hadoop Platform

The overall architecture of our Hadoop platform with the three main layers is depicted in figure 3. The framework consists of 3 layers: (1) Storage layer; (2) Operation layer; (3) Interactive layer. The storage layer is optimized by indexing techniques to accelerate the Hadoop Map-Reduce applications. The operation layer is equipped with two powerful modules including "Change Detection" and "3D Geometric Model" for efficient monitoring. In this study, only change detection module was used. More details for 3D geometric model will be introduced in our future paper. Lastly, The Interactive layer enables users to approach a distributed computing model on Hadoop without requiring deeper related knowledge.

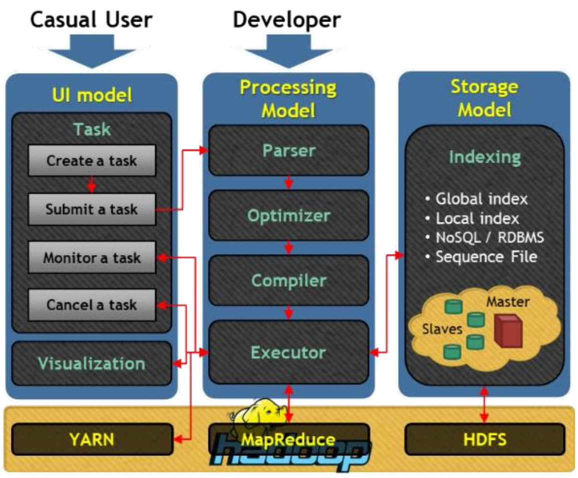


Figure 3. Hadoop platform architecture

3 Implementation and Experiment

The overview of this research is as follows.

- Using the mapping platform 3D point cloud map was constructed.
- Based on the pre-constructed map, mapping system navigates the manually defined goal points automatically and constructs another map.
- Change detection was processed based on Hadoop platform between 2 maps.
- Semantic segmentation was implemented to detected changes.

3.1 Mapping System Experiment

3.1.1 Calibration

Calibration is the process of estimating the parameters that need to be applied to correct actual measurements to their true values. According to the parameter types, calibration also can be divided into intrinsic and extrinsic forms. This research assumes that the intrinsic sensor calibration is completed and focuses on the extrinsic calibration of multiple LRF sensors to identify the rigid transformation from each sensor frame to the platform body frame.

Figure 4(b) shows three 2D LRFs in the mapping system and the configurations of the three coordinate systems. The middle LRF is mounted horizontally, other two LRFs are mounted vertically to scan the profiles of the surrounding environments. For carrying out the calibration, the double-deck 3D calibration facility was developed as shown in figure 4(c). The alignment errors usually are computed by least squares using a set of target points captured from the scanners. Since this research is not focused on calibration, further information about the least squares system equation and experimental results can be referred from [33].

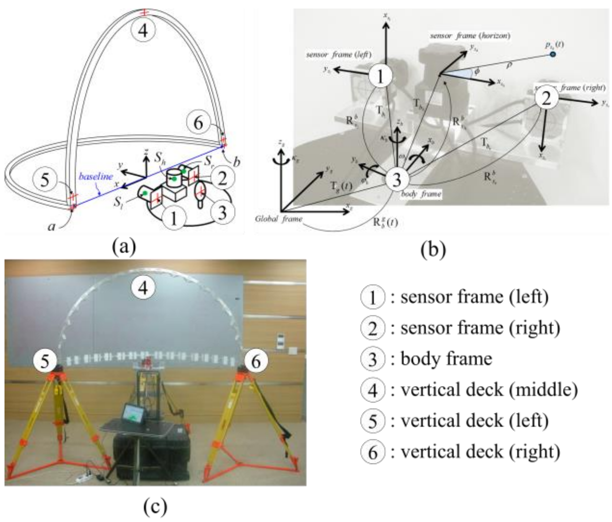


Figure 4. Double-deck calibration facility

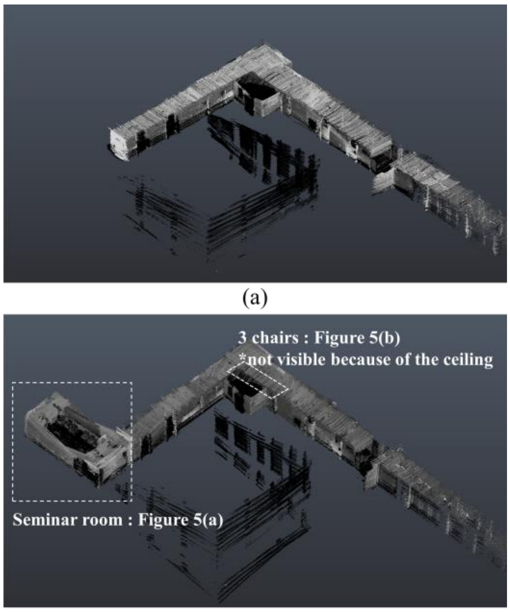
3.1.2 SLAM and Navigation

The implementation of Google cartographer SLAM was conducted in seminar rooms and corridors in the Engineering building at Yonsei University. Map was built twice in the same location with intentionally caused changes. Figure 5 shows two types of intentionally caused changes, which is newly added space and added furniture in existing space. Figure 5(a) is an added seminar room which can be seen in the left part of the figure 6(b). Additionally, chairs were added in figure 5(b) which were located in middle of the corridor as shown in figure 6(b). Figure 6 shows the map where high intensity values are represented bright.

To build the second map, consecutive goal points were manually selected among the junctions from the skeletonization result of pre-constructed map. Figure 7(a) shows the skeletonized result, and red dots in figure 7(b) shows the path consists of manually selected goals.

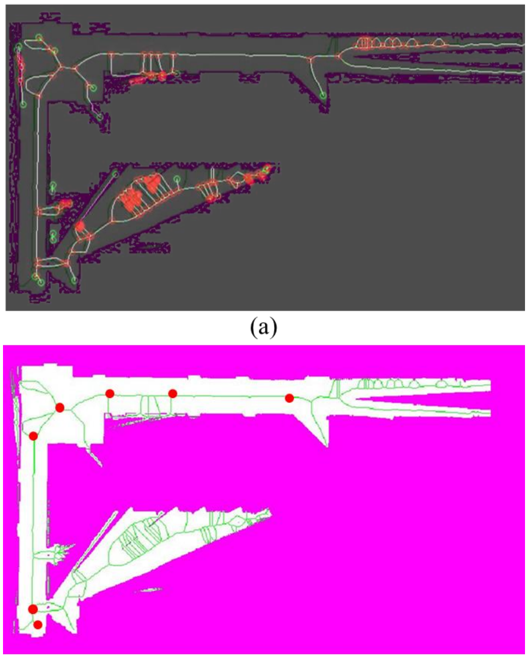
(a) (b)

Figure 5. Added environments in the second map



(a) (b)

Figure 6. (a) First map; (b) Second map



(a) (b)

Figure 7. (a) Skeletonization results; (b) manually selected goal points (red dots)

3.2 Analysis System Experiment

3.2.1 Change Detection

Change detection between two maps were processed based on the Hadoop operation layer. The Hadoop cluster was configured with 8 nodes of the same capacity, i.e. Intel Core i5-2300 2.80GHz* 4, 16GB memory.

Figure 8 shows the two maps and change detection result, using the web User Interface (UI) based on Hadoop platform. Detail explanation for the WebUI will be in our future paper. The red and green part represents the first and second map respectively. Blue dots are the parts which have been detected as a change. Developed platform can simply upload and overlay different point clouds and visualize them. The detailed description for the change detection algorithm is introduced in section 3.3.

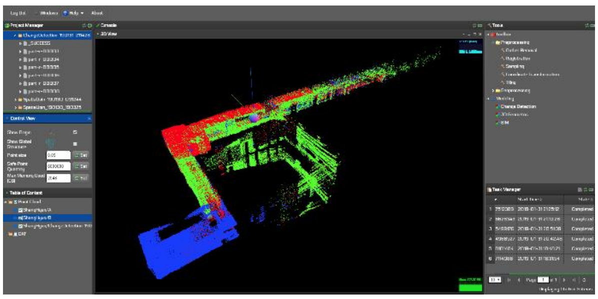


Figure 8. Change detection result in webUI

3.2.2 Semantic Mapping

For training the model for point cloud segmentation for semantic scene labeling, the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [34] was used. The dataset was acquired using Matterport scanners in 6 areas for 271 rooms. 13 classes were annotated to each point. The model was trained using S3DIS from area 1-5 and tested using area 6. The overall per-point accuracy of the trained segmentation model was 83.4%. Per-point accuracy compares the predicted label and real label for each point.

Trained model was implemented to the map which was detected as a change between the first and second map from mapping system. To evaluate the accuracy of the semantic segmentation result, annotations were manually assigned to each point in the map. Figure 9 shows the semantic segmentation result for seminar room.

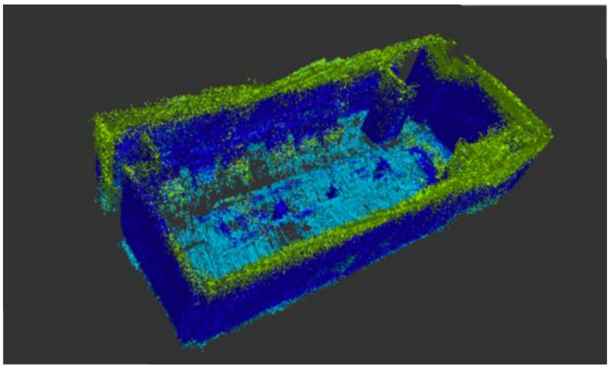


Figure 9. Semantic mapping result (point cloud)

Table 1. Semantic mapping accuracy

Overall accuracy	72.76%
classes	IoU
Ceiling	0.502
Wall	0.721
Floor	0.527
Door	0.050
Chair	0.000
Table	0.000

Overall per-point accuracy for experimental data was 72.76%. IoU in Table 1 stands for Intersection over Union, which is calculated by equation (1).

$$IoU_i = \frac{TP_i}{FP_i + FN_i + TP_i} \quad (1)$$

Where *TP* is the true positives *FP* is the false positives *FN* is the false negatives and *i* stands for each class such as ceilings, wall etc.

In figure 9, ceiling points were colored in deep green, wall points in blue, floor points in sky blue and doors in pink but not visible in the figure. As listed in table 1, table and chairs were not detected. Firstly, the top surface of the table was not collected enough in our data, since the height of the table top and the sensor was similar. Secondly, there were too many sensor noise for chair data. Compared to the S3DIS training dataset, the data acquired from our mapping system contains more noise since it was acquired while moving through the environment and no additional process were conducted after acquiring. To segment the movable objects, such as table, chair, sofa, bookcase and boards, further improvements are necessary for the mapping system.

3.3 Hadoop Platform

To optimize the storage layer in the Hadoop platform, Sort-Tile-Recursive (STR) algorithm [35] was applied to construct a global index. STR groups nearby points in each minimum bounding rectangle (MBR) which does not exceed 128MB. After constructing a global index, an Octree local index is also implemented on each block.

Based on the storage layer, change detection was tested for the operation. Change detection is based on a soft-join algorithm which is an application of point-to-point change detection in a distributed system (Figure 10).

The interactive layer was developed based on web UI. Interface provides simple access to users and visualizes massive point cloud data directly from Hadoop, which is useful for analytical processes.

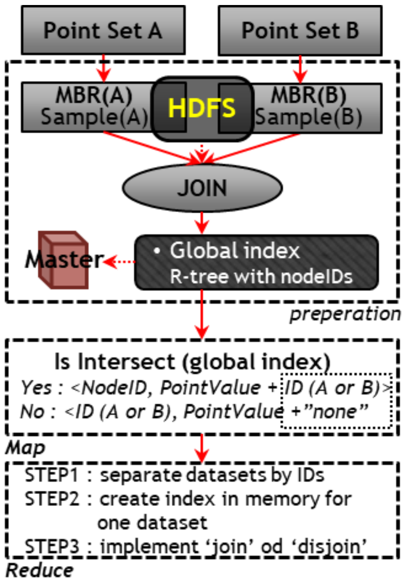


Figure 10. Change detection on Hadoop platform

4 Conclusion

An integrated framework for processing massive point cloud data was introduced in this research. It is consisting of three main parts: mapping system; analysis system; and Hadoop platform, for the integrated framework developed as a prototype. However, this comprehensive system is a pioneer work related to massive point cloud data producing and processing. Future works for this prototype version will be focused on each system. For mapping system, additional sensors which can provide more information, should be implemented. Also, navigation algorithms will be improved for fully automated system. For analysis system, change detection process need to be focus on algorithm perspective, since it is now focused managing massive point cloud data. Applied semantic mapping process will also be included in the process more practically.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (NRF-2018R1A2B2009160).

References

[1] K. Charalampous, I. Kostavelis, and A. Gasteratos, Recent trends in social aware robot navigation: A survey. *Rob. Auton. Syst.*, vol. 93, page 85–104, 2017.

[2] M. Jaboyedoff, T. Oppikofer, A. Abellan, M.H. Derron, A. Loye, R. Metzger and A. Pedrazzini, Use of LIDAR in landslide investigations: a review. *Nat. hazards*, vol. 61, no. 1, page 5–28, 2012.

[3] S. Nebiker, N. Lack, and M. Deuber, Building change detection from historical aerial photographs using dense image matching and object-based image analysis. *Remote Sens.*, vol. 6, no. 9, page 8310–8336, 2014.

[4] R. Qin, X. Huang, A. Gruen, and G. Schmitt, Object-based 3-D building change detection on multitemporal stereo images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 5, page 2125–2137, 2015.

[5] K. Choi, I. Lee, and S. Kim, A feature based approach to automatic change detection from LiDAR data in urban areas. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 18, page 259–264, 2009.

[6] M. Gerke and N. Kerle, Automatic structural seismic damage assessment with airborne oblique Pictometry imagery. *Photogramm. Eng. Remote Sens.*, vol. 77, no. 9, page 885–898, 2011.

[7] D. Rebolj, N. Č. Babič, A. Magdič, P. Podbreznik, and M. Pšunder, Automated construction activity monitoring system. *Adv. Eng. informatics*, vol. 22, no. 4, page 493–503, 2008.

[8] A. Braun, S. Tuttas, A. Borrmann, and U. Stilla, Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. in *Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 32, page 1., Oulu, Finland, 2015.

[9] K. Han, J. Lin, and M. Golparvar-Fard, A formalism for utilization of autonomous vision-based systems and integrated project models for construction progress monitoring. In *Proceedings of Autonomous and Robotic Construction of Infrastructure Conference*, Iowa, USA, 2015.

[10] S. Siebert and J. Teizer, Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Autom. Constr.*, vol. 41, page 1–14, 2014.

[11] R. Qin, J. Tian, and P. Reinartz, 3D change detection – Approaches and applications. *ISPRS J. Photogramm. Remote Sens.*, vol. 122, page 41–56, 2016.

[12] J. Jung, S. Hong, S. Yoon, J. Kim, and J. Heo, Automated 3D wireframe modeling of indoor structures from point clouds using constrained least-squares adjustment for as-built BIM. *J. Comput. Civ. Eng.*, vol. 30, no. 4, page 4015074, 2015.

[13] J. W. Hegeman, V. B. Sardeshmukh, R. Sugumaran, and M. P. Armstrong, Distributed LiDAR data processing in a high-memory cloud-computing environment. *Ann. GIS*, vol. 20, no. 4, page 255–264, 2014.

[14] V. Pajić, M. Govedarica, and M. Amović, Model of Point Cloud Data Management System in Big Data Paradigm. *ISPRS Int. J. Geo-Information*, vol. 7, no. 7, page 265, 2018.

[15] C. Wang, F. Hu, D. Sha, and X. Han, Efficient LiDAR point cloud data managing and processing in a hadoop-based distributed framework. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 4, page 121, 2017.

[16] Z. Li, M. E. Hodgson, and W. Li, A general-purpose framework for parallel processing of large-scale LiDAR data. *Int. J. Digit. Earth*, vol. 11, no. 1, page 26–47, 2018.

[17] M. Quigley *et al.*, ROS: an open-source Robot Operating System. in *ICRA workshop on open source software*, vol. 3, no. 3.2, page 5, Kobe, Japan, 2009.

[18] G. Grisettiyz, C. Stachniss, and W. Burgard, Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. in *Proceedings of IEEE International Conference on Robotics and Automation*, page

2432-2437, Barcelona, Spain, 2005.

[19] Karto SLAM, ROS package. accessed Nov, 2016. [online], wiki.ros.org/slam karto

[20] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O.V. Stryk, Hector open source modules for autonomous mapping and navigation with rescue robots. in *Robot Soccer World Cup*. Springer, 2013.

[21] W. Hess, D. Kohler, H. Rapp, and D. Andor, Real-time loop closure in 2D LIDAR SLAM. in *Robotics and Automation (ICRA)*, page 1271–1278, Stockholm, Sweden, 2016.

[22] H. A. Lauterbach, D. Borrmann, R. Heß, D. Eck, K. Schilling, and A. Nüchter, Evaluation of a backpack-mounted 3D mobile scanning system. *Remote Sens.*, vol. 7, no. 10, page 13753–13781, 2015.

[23] R. Yagfarov, M. Ivanou, and I. Afanasyev, Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, page 1979–1983, Singapore, 2018.

[24] J. M. Santos, D. Portugal, and R. P. Rocha, An evaluation of 2D SLAM techniques available in robot operating system. in *Safety, Security, and Rescue Robotics (SSRR)*, page 1–6, Linkoping, Sweden, 2013.

[25] A. Filatov, A. Filatov, K. Krinkin, B. Chen, and D. Molodan, 2D SLAM quality evaluation methods. in *Open Innovations Association (FRUCT)*, page 120–126, Helsinki, Finland, 2017.

[26] K. Krinkin, A. Filatov, A. yom Filatov, A. Huletski, and D. Kartashov, Evaluation of Modern Laser Based Indoor SLAM Algorithms. in *Conference of Open Innovations Association (FRUCT)* page 101–106, Jyväskylä, Finland, 2018.

[27] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, Monte carlo localization: Efficient position estimation for mobile robots. In *AAAI/IAAI*, vol. 1999 no. 343–349, page 2, Orlando, USA, 1999.

[28] Navigation, ROS package. accessed Mar, 2013. [online], wiki.ros.org/navigation

[29] D. Fox, W. Burgard, and S. Thrun, The dynamic window approach to collision avoidanc. *IEEE Robotics & Automation Magazine*, no. 4.1, page 23-33, 1997.

[30] E. R. Dougherty, An Introduction to Morphological Image Processing (Tutorial Texts in Optical Engineering. *DC O'Shea, SPIE Opt. Eng. Press. Bellingham, WA, USA*, 1992.

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceeding of Computer Vision Pattern Recogniton*, vol. 1, no. 2, page 4, Honolulu, Hawaii, 2017.

[32] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, page 5099–5108, California, USA, 2017.

[33] J. Jung, J. Kim, S. Yoon, S. Kim, H. Cho, C. Kim and J. Heo, Bore-sight calibration of multiple laser range finders for kinematic 3D laser scanning systems. *Sensors*, vol. 15, no. 5, page 10292–10314, 2015.

[34] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, Joint 2d-3d-semantic data for indoor scene understanding. *arXiv Prepr. arXiv1702.01105*, 2017.

[35] S. T. Leutenegger, M. A. Lopez, and J. Edgington, STR: A simple and efficient algorithm for R-tree packing. In *Conference on Data Engineering*, page 497–506, Birmingham, UK, 1997,