Autonomous navigation using visual sparse map*

Soonhac Hong, *Member, IEEE,* and Hui Cheng, *Senior Member, IEEE*

***Abstract*— This paper presents an autonomous navigation system using only visual sparse map. Although a dense map provides detail information of environment, most information of the dense map is redundant for autonomous navigation. In addition, the dense map demands the high cost for storage, transmission and management. To tackle these challenges, we propose the autonomous navigation using a visual sparse map. We leverage visual Simultaneous Localization and Mapping (SLAM) to generate the visual sparse map and localize a robot in the map. Using the robot position in the map, the robot navigates by following a reference line generated from the visual sparse map. We evaluated the proposed method using two robot platforms in indoor environment and outdoor environment. Experimental results show successful autonomous navigation in both environments.**

I. INTRODUCTION

Autonomous navigation is an essential component for a robot to reach a goal location. For autonomous navigation, dense maps have been typically used [4 - 15]. However, there are a couple of challenges of dense map based autonomous navigation. First, most points of a dense map are redundant for localization and navigation. Second, the dense map needs to be updated periodically if environment changes. Thus, high-cost map management and computation follows and a high transmission bandwidth is required to update the dense map. Third, a large memory is needed to store the dense map as the map size increases.

To tackle these challenges of dense map based autonomous navigation, we propose an autonomous navigation system using visual sparse map as shown in Fig. 1. The autonomous navigation system using visual sparse map has two phases; 1) map generation and 2) autonomous navigation.

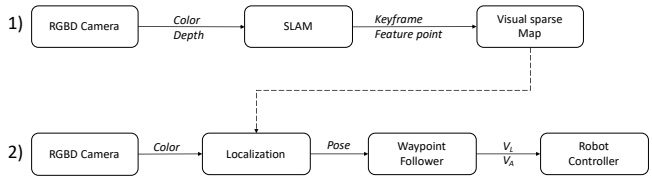


Figure 1. Overview of autonomous navigation using visual sparse map

In the map generation phase, color images and depth images from a RBG-D camera are used to generate a visual sparse map by Simultaneous Localization and Mapping (SLAM). As the visual sparse map includes only visual feature points and keyframes as shown in Fig. 2, the map size can be reduced considerably. Each visual feature point has the 3D position of the visual feature point. Each keyframe has 3D position and 3D orientation.

*This work has been supported by JD.com.
Soonhac Hong and Hui Cheng are with JD.com, Mountain View, CA 94043 USA (e-mail: soonhac.hong@ jd.com and hui.cheng@jd.com).

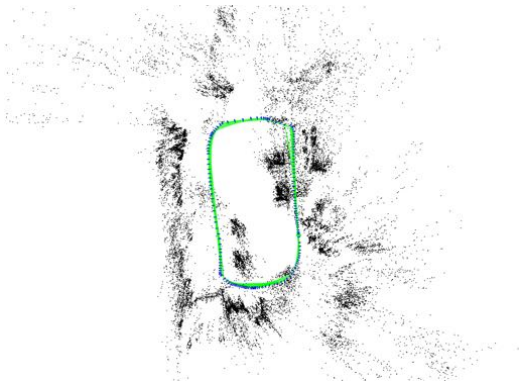


Figure 2. Example of visual sparse map (Datase I). Blue points represent the keyframe of the map. Green lines represent the visibility among keyframes. Black points represent visual feature points.

In the autonomous navigation phase, only color images are used for localization. A SLAM algorithm computes the robot pose using a color image and the visual sparse map. Using the robot pose and keyframes in the visual sparse map, the waypoint follower computes a translation velocity and a angular velocity to enable the robot to follow the reference line, a list of keyframes in the map.

This paper is organized as follows. Section II reviews related works. Section III briefly describes the SLAM algorithm. Section IV explains the waypoint follower. Section V presents experimental results and the paper is concluded in Section VI.

II. RELATED WORK

A couple of maps have been introduced for autonomous navigation. Metric map is one of the popular maps for autonomous navigation. In a metric map, positions of landmarks or objects in an environment are stored in a map with respect to a global coordinate system. Metric map can be classified by continuous map and discrete map [1]. While the former represents the environment using lines or polygons [2, 3], the latter represents the environment using cells, points, Surfel, Voxel, and features. Discrete map can be classified as dense map and sparse map according to map density. Cell, point, Surfel and Voxel have been used for dense map and features have been used for sparse map.

Occupancy grid map is a typical map using cells for autonomous navigation [4 - 6]. Each cell of an occupancy grid map represents whether a space is occupied by objects or not. A path for navigation is planned on the occupancy grid map. However, the occupancy grid map typically represents the environment in 2D space. For 3D space, a point cloud map has been used [6 - 10]. As the point cloud map densely represents

the environment as many points, the size of the point cloud substantially increases as the map area grows. To reduce the size of point cloud map, Surfel [11, 12] and Voxel [4, 13 - 15] are introduced. However, Surfel and Voxel still need high computational cost for post-processing for generating Surfel and Voxel. In addition, most information of the dense map is redundant for autonomous navigation. Thus, a sparse map has been proposed.

The sparse map can be represented as features (e.g. visual feature descriptors) [16 - 21]. As each visual features can be generated from corners or blobs in the image, the number of visual feature points is much smaller than the number of points in the point cloud map. However, most works on sparse map have focused on mapping and localization. There has been a little attention for autonomous navigation using a sparse map [29]. Although [29] uses a spare map generated by Harris corner detector, it uses a point cloud map not visual feature descriptors for a map. Thus, this paper presents the autonomous navigation system using visual sparse map.

III. Mapping And Localization

We leverage ORB-SLAM2 [22] for building a visual sparse map and localization. This section gives the brief summary of mapping and localization of ORB-SLAM2 and additional methods we implement for the proposed system. Further details of ORB-SLAM2 can be found at [22].

A. Mapping

ORB-SLAM2 consists of three modules; 1) Tracking, 2) Local mapping, and 3) Loop closing. When a new image is captured, the tracking module checks if a local map is available. If there is no map available, a local map is initialized. If the local map is available, the tracking module predicts a relative pose between the new image and the local map using the motion model. If the motion model is not available, the relative pose is predicted using visual odometry with respect to the last keyframe. If neither motion model nor visual odometry predicts the relative pose, relocalization predicts the relative pose. Relocalization finds similar keyframes using visual vocabulary in the map and estimates the relative pose to the most similar keyframe. If the relative pose is successfully estimated by motion model, visual odometry or relocalization, the relative pose is refined with the local map. If the relative pose of the new image is successfully computed, the tracking module determines if the new image is a new keyframe. If the number of matched points between the current image and the last keyframe is smaller than a threshold, the new image is determined as the new keyframe.

If a new keyframe is generated by the tracking module, the new keyframe is added to the local map. Given the new keyframe, the local map module optimizes the local map using a local Bundle Adjustment (BA). To limit the size of the local map, the local map deletes redundant keyframes in order to maintain a compact local map. If a keyframe has 90% of the map points which has been seen in at least other three keyframes, the keyframe is determined as a redundant keyframe and deleted in the local map.

Given the new keyframe, the loop closing module checks if the new keyframe is the revisited image. The loop closing module recognizes the revisited place using a place recognition database consisting of visual vocabulary. If the new keyframe is found in the visual vocabulary, the loop closing module optimizes the entire map using pose graph optimization and global BA. Otherwise, the visual vocabulary of the new keyframe is added to the place recognition database.

As ORB-SLAM2 does not provide a method to save and load the map into a file, we implemented the method to save and load the map. The visual sparse map generated by ORB-SLAM2 contains visual feature points, keyframes and a pose graph. Each visual feature point has the index and 3D position in the map. Each keyframe has the index, 3D pose and visual feature descriptors. The pose graph represents connectivity among keyframes using vertices and edges. In the pose graph, vertices represent keyframes and edges represent visible connection among keyframes.

B. Localization

Given the map, only the tracking module is used in the localization mode. The local map and the map database is not updated in the localization mode. In addition, the place recognition database is not updated. Whenever the new image is captured, the tracking module computes the relative pose of the camera with respect to the origin of the map. The camera pose $\boldsymbol{x}_C$ is composed of the camera position $[x, y, z]$ and orientation $[roll, pitch, yaw]$ in the map. The coordinate of the map locates at the pose of the first keyframe in the map.

IV. Waypoint Follower

Using the camera pose and a reference line from the visual sparse map, the waypoint follower module computes the translation velocity and the angular velocity to control the robot. We assume $\boldsymbol{x}_C$ is identical to the robot pose $\boldsymbol{x}_R$ because the reference line is generated with assuming $\boldsymbol{x}_C$ is identical to $\boldsymbol{x}_R$. When a new image is captured, $\boldsymbol{x}_R$ is computed by the tracking module of ORB-SLAM2.

The reference line is generated from the map. The reference line is represented as the list of the keyframe positions

$$\boldsymbol{L}_R = \{\boldsymbol{x}_0, \boldsymbol{x}_1 \dots, \boldsymbol{x}_{k-1}, \boldsymbol{x}_k\} \quad (1)$$

where $\boldsymbol{x}_k = [x, y, z]$ is the k^{th} keyframe position in the map.

If $\boldsymbol{x}_R$ is successfully computed by the tracking module, the nearest keyframe $\boldsymbol{x}_N$ from $\boldsymbol{x}_R$ is founded in $\boldsymbol{L}_R$. A keyframe ahead with a pre-defined distance from $\boldsymbol{x}_N$ is determined as a temporary target waypoint $\boldsymbol{x}_T$. Transitional difference $\boldsymbol{t}_D$ and angular difference θ_D between $\boldsymbol{x}_R$ and $\boldsymbol{x}_T$ can be computed by

$$\boldsymbol{t}_D = \|\boldsymbol{t}_T - \boldsymbol{t}_R\| \quad (2)$$

$$\theta_D = |\theta_T - \theta_R| \quad (3)$$

Where $\boldsymbol{t}_T = [x, y, z]$ and $\boldsymbol{t}_R = [x, y, z]$ are robot positions at the target waypoint and the current position respectively. θ_T and θ_R are orientations of the robot at target waypoint and current position respectively in 2D space.

To control the robot, we computes the translational velocity $\boldsymbol{V}_T$ and the rotational velocity $\boldsymbol{V}_\theta$ by

$$\boldsymbol{V}_T = \begin{cases} V_m & \text{if } \theta_D \leq \theta_h \\ \frac{V_m}{2} & \text{otherwise} \end{cases} \quad (4)$$

$$\boldsymbol{V}_\theta = \frac{\theta_D}{\alpha} \quad (5)$$

where V_m is the desired maximum translational speed of the robot. θ_h is a threshold of angular difference for reducing $\boldsymbol{V}_T$. If θ_D is larger than θ_h, $\boldsymbol{V}_T$ is reduced by half. α is an empirical coefficient for computing $\boldsymbol{V}_\theta$ using θ_D.

V. Experimental Results

We evaluated the proposed autonomous navigation system using Robotis Turtlebot 2 [23] with Orbbec Astra Pro [24] in indoor environment and Clearpath Husky [25] with Logitech C920 Pro [26] in outdoor environment.

A. Experimental platforms

We installed one RGB-D camera, Orbbec Astra Pro, on the Turtlebot in indoor environment as shown in Fig. 3. Orbbec Astra Pro has a resolution of 640 × 480 pixels in both a color image and depth image.

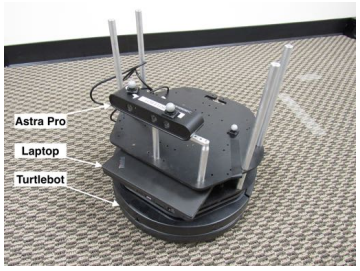


Figure 3. Robotis Turtlebot 2 with Orbbec Astra Pro for indoor environment

As the RGB-D camera is not working in outdoor environment, we use Logitech C920 Pro instead of Orbbec Astra Pro. We use only 640 × 480 color images for both mapping and localization in outdoor environment. In addition, we use Clearpath Husky for safe and robust mobility in outdoor environment as shown in Fig. 4. The autonomous navigation systems in both robot platforms are built on ROS [27].

Figure 4. Clearpath Husky with a Logitech C920 for outdoor environment

B. Localization accuracy with Map data

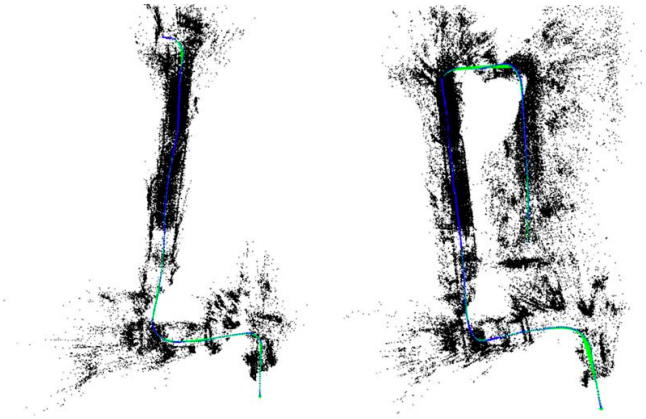
We evaluated localization accuracy with map data before evaluating autonomous navigation. We use the same map data for evaluating localization accuracy. However, we use only color images for localization while both color images and depth images are used for building a map in indoor environment.

Figure 5. Snapshots of offices and hallway for datasets in indoor environment. (a) office A, (b) hallway between office A and elevator, (c) elevator at the end of hallway, (d) glass door to office B, (e) narrow gate to office B and (f) office B.

We collected three datasets in office environment as shown in Fig. 5. The first dataset is collected in office A which includes desks, chairs and shelves. The robot starts near the first shelf and returns to the start position. The second dataset is collected in Office A and a hallway. The robot starts from Office A, runs along the hallway and stops in front of an elevator at the end of the hallway. The third dataset is collected in Office A, the hallway and Office B. The robot starts from Office A, runs along the hallway and stops at Office B. There is a 1 meter-wide narrow gate between the hallway and Office B. Table I shows the path length and environment of each dataset. Fig. 6 shows maps and trajectories of dataset II and III. The map and trajectory of Dataset I is shown in Fig. 2.

TABLE I. Datasets in Indoor Environment

Dataset	Length [m]	Environment
I	17.41	Office A
II	41.38	Office A, hallway
III	49.40	Office A and B, hallway



(a) Dataset II (b) Dataset III

Figure 6. Maps and trajectories in Dataset II and III

Table II shows the localization error with map datasets. Although the same map dataset is used for evaluating localization accuracy, the average Root Mean Square Error (RMSE) is 0.031 meter because ORB-SLAM2 randomly generates visual features from a color image for localization. However, the average RMSE is acceptable for autonomous navigation because the minimum width of path is 1 meter. Fig. 7 shows map and localization trajectories on dataset I. As RMSE is 0.036 meter, the localization trajectory overlays the map trajectory.

TABLE II. LOCALIZTION RMSE WITH MAP DATA

Dataset	RMSE [m]
I	0.036
II	0.03
III	0.03
Average	0.031

We also evaluated localization accuracy in environment changes because the environment can be changed after generating the map. We changed about 30% of objects in the same place in dataset I and collected a new dataset for evaluating localization. Given the map generated from dataset I, localization RMSE is 0.116 ± 0.111 meter [mean ± standard deviation]. Although environment changes increase localization RMSE slightly, the RMSE in environment changes is still acceptable for autonomous navigation.

TABLE III. LOCALIZTION RMSE IN AUTONOMOUS NAVIGATION

Dataset	RMSE [m]
I	0.065 ± 0.045
II	0.166 ± 0.127
III	0.117 ± 0.075
Average	0.116 ± 0.082

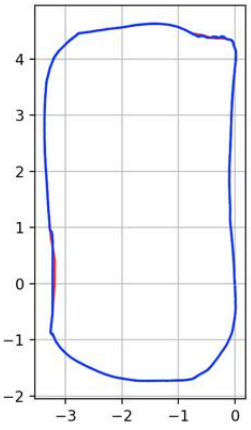
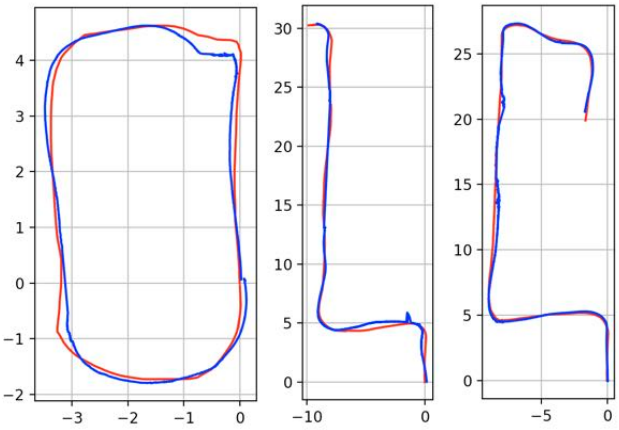


Figure 7. Map and localization trajectories with Dataset I. Red line represents the map trajectory and blue line represents the localization trajectory.

C. Localization accuracy in autonomous navigation

We evaluated localization error when the robot runs in the autonomous navigation phase. The waypoint follower enables the robot to follow a reference line as close as possible. We compute the localization error by finding the closest waypoint from the estimated position by ORB-SLAM2 localization as shown in Table III.



(a) Dataset I (b) Dataset II (c) Dataset III

Figure 8. Map and localization trajectories in autonomous navigation. Red lines represent trajectories of maps and bule lines represent trajectories of localization.

Experimental results show that: 1) the average localization RMSE is 0.116 ± 0.082 meter [mean ± standard deviation]; 2) the robot successfully navigates in three different environments even there are challenge environments such as a feature-spare long hallway (length: 25 meter) and the 1 meter-wide narrow gate; 3) there are relatively larger error when the robot turns; 4) the feature sparse long hallway increases localization error. Fig. 8 shows map and localization trajectories in autonomous navigation.

D. Environment changes in outdoor environment

We evaluated localization error with environment changes in outdoor environment. Datasets are collected along the

sidewalk around JD.com office, Santa Clara, California, USA. The path consists of straight, curved and winding sidewalks under trees as shown in Fig. 9.

(a) (b) (c) (d)

Figure 9. Snapshots of outdoor environment. (a) start position, (b) curved sidewalk, (c) winding sidewalk and (d) goal position.

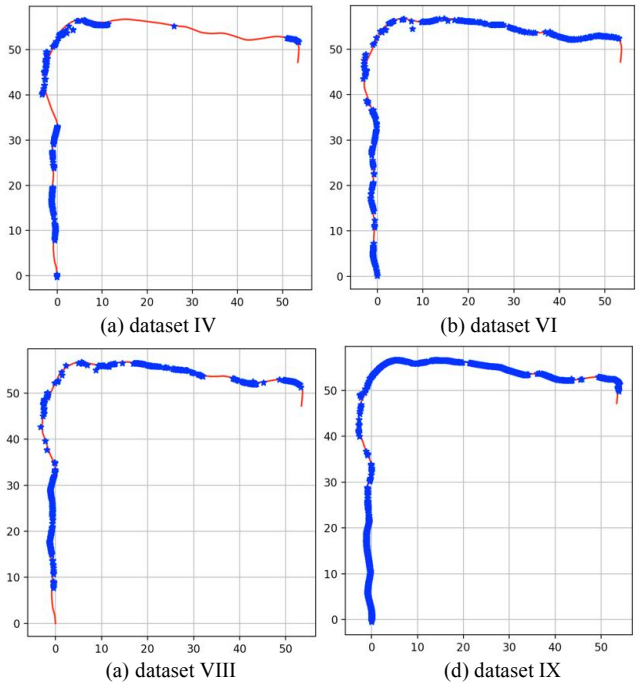
The map dataset is collected at 15:04 on December 13, 2017. The path length of the map is 114.70 meter. We collected six datasets as shown in Table IV: 1) dataset IV to VII are collected at different time in sunny days; 2) dataset VIII is collected in a cloudy day; 3) dataset IX is collected in a rainy day.

TABLE IV. LOCALIZATION ANAYSIS WITH ENVIRONMENT CHANGES IN OUTDOOR ENVIRONMENT

Dataset	Weather	Date/Time	Failure ratio	Failure time [sec]		
				Max	Mean	Std.
IV	Sunny	2018-01-19-09-57-51	48%	36.15	1.55	4.29
V	Sunny	2018-01-11-14-12-09	10%	0.57	0.22	0.13
VI	Sunny	2018-01-12-15-32-45	3%	0.33	0.07	0.06
VII	Sunny	2018-01-12-16-51-56	12%	2.40	0.44	0.52
VIII	Cloudy	2018-01-17-11-39-49	17%	3.43	0.99	1.30
IX	Rainy	2018-01-03-11-40-42	12%	9.80	0.55	1.30

We use two metric, failure ratio and failure time, for evaluating localization performance. Failure ratio is the ratio of localization failure over all localization tries. Failure time is the time from the localization failure to the next localization success. As the dataset is collected by manual driving, localization accuracy is not evaluated.

As shown in Table IV, experimental results show that: 1) dataset VI has the smallest failure ratio because dataset VI is collected at similar time and weather to the map; 2) dataset IV has the largest failure ratio because the illumination of dataset IV is quite different from the map due to the position of the sun; 3) failure time has proportional relationship with failure ratio in sunny day but the proportional relationship between failure ratio and failure time is not valid in the rainy day and the cloudy day; 4) in the rainy day, failure time is larger than the cloudy day while failure ratio is smaller than the cloudy day. Fig. 10 shows trajectories of map and localization in dataset IV, VI, VIII and IX.



(a) dataset IV (b) dataset VI

(a) dataset VIII (d) dataset IX

Figure 10. Map and localization trajectories with environment changes in outdoor environment. Red lines represent trajectories of maps and bule stars represent positions of successful localization.

E. Autonomous navigation in outdoor environment

As mentioned in the previous section, ORB-SLAM2 is not robust at different time and different weather in outdoor environment. Thus, we evaluated autonomous navigation at 15:02 on January 11, 2018, a sunny day, which is similar time and weather to the map.

Experimental result shows the robot ran successfully on the sidewalk and localization RMSE is 0.246 ± 0.151 meter [mean ± standard deviation]. The width of sidewalk is about 1.5 meter. Fig. 11 shows trajectories of map and localization in autonomous navigation. We note that the robot is rarely localized in the curved sidewalk because most visual features come from the distant objects.

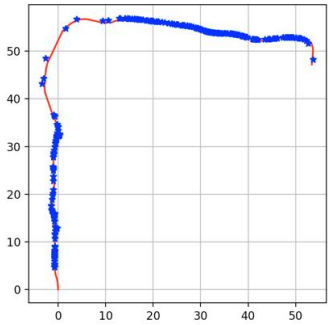


Figure 11. Map and localization trajectories in autonomous navigation in outdoor environment. Red line represents trajectories of map and bule stars represents positions of successful localization.

VI. CONCLUSION

We proposed an autonomous navigation system using only

visual sparse map for indoor environment and outdoor environment. ORB-SLAM2 is used for mapping and localization. Waypoint follower enables the robot to follow the reference line. We evaluated the proposed system in indoor environment and outdoor environment using two robot platforms.

Experimental results show that: 1) localization errors with the map datasets are acceptable for the robot to run autonomously indoor environment; 2) the robot successfully ran in three indoor environments including environment changes; 3) environment changes in outdoor apparently increases localization failure ratios; 4) the robot successfully ran in similar time and weather to the map in outdoor environment.

We will investigate for robust localization with environment changes in outdoor environment. In addition, sensor fusion with additional sensors such as IMU, GPS and Lidar will be investigated. We will also extend the proposed system by including obstacle avoidance and path planning.

ACKNOWLEDGMENT

We would like to thank Chengzhi Qi and Jiawei Yao for collecting datasets with different time and weather conditions.

REFERENCES

[1] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, "Introduction to Autonomous Mobile Robots, Second edition," The MIT Press, 2011.

[2] J.C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, 1991.

[3] A. Lazanas, J.C. Latombe, "Landmark robot navigation," Proceedings of the Tenth National Conference on AI, 1992.

[4] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," In Proc. IEEE International Conference on Robotics and Automation (ICRA), pages 300-307, 2010.

[5] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo vision," Exp. Robot., pp. 179190, 2008.

[6] F. Dayoub, T. Morris, B. Upcroft, P. Corke, "Vision- only autonomous navigation using topometric maps," Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on , vol., no., pp.1923,1929, 3-7 Nov. 2013.

[7] J. Engel, T. Schps, D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," In European Conference on Computer Vision (ECCV), 2014

[8] J. Engel, J. Sturm, D. Cremers, "Semi-Dense Visual Odometry for a Monocular Camera," In IEEE International Conference on Computer Vision (ICCV), 2013

[9] S. Hong, "6-DOF Pose Estimation for A Portable Navigation Device," PhD dissertation, University Of Arkansas at Little Rock, 2014.

[10] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, A.J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1352-1359, 2013.

[11] R.F. Salas-Moreno, B. Glocken, P.H.J. Kelly, A.J. Davison, "Dense planar SLAM," IEEE International Symposium on Mixed and Augmented Reality (IS- MAR), pp.157-164, 2014.

[12] 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," International Journal of Robotics Research, vol. 31, no. 5, pp. 647663, 2012.

[13] A.S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Mat- urana, D. Fox, N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," In International Symposium on Robotics Research (ISRR), pp. 1-16, 2011.

[14] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," IEEE International Symposium on Mixed and Augmented Reality (ISMAR), , pp.127-136, 2011

[15] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, S. Bathiche, "MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera," Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on , pp.83,88, 2013.

[16] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052-1067, 2007.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB- SLAM : a Versatile and Accurate Monocular," IEEE Trans. Robot., pp. 115, 2015.

[18] H. Lim, J. Lim, and H. J. Kim, "Real-time 6-DOF monocular visual SLAM in a large-scale environment," IEEE Inter- national Conference on Robotics and Automation (ICRA), 2014, pp. 1532-1539, 2014.

[19] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping," CVPR, pp. 3489-3496, 2011.

[20] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," IEEE Int. Conf. Robot. Autom., no. ICRA, pp.1635-1642, May 2012.

[21] C. X. Guo, K. Sartipi, R. C. Dutoit, G. Georgiou, R. Li, J. O. Leary, E. D. Nerurkar, J. A. Hesch, and S. I. Roumeliotis, "Large-Scale Cooperative 3D Visual-Inertial Mapping in a Manhattan World," Technical Report, University of Minnesota, Dept. of Comp. Sci. and Eng., MARS Lab, February 2015.

[22] R. Mur-Artal, & J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Transactions on Robotics, 2017.

[23] Robotis turtlebot2, http://www.turtlebot.com/turtlebot2/

[24] Orbbec Astra Pro, https://orbbec3d.com/product-astra-pro/

[25] Clearpath Husky, https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

[26] Logitech C920 Pro, https://www.logitech.com/en-us/product/hd-pro-webcam-c920

[27] Robot Operating System (ROS), http://www.ros.org/

[28] Vicon motion capture system, https://www.vicon.com/

[29] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier, "Outdoor autonomous navigation using monocular vision," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.