

Large-scale 3D Outdoor Mapping and On-line Localization using 3D-2D Matching

Takahiro Sakai, Kenji Koide, Jun Miura, and Shuji Oishi

Abstract—Map-based outdoor navigation is an active research area in mobile robots and autonomous driving. By preparing a precise map of an environment or roadside, a robot or a vehicle can localize itself based on a matching between the map and a sequence of sensor inputs. This paper describes a campus-wide mapping and localization of a mobile robot with 2D and 3D LIDARs (Laser Imaging Detection and Ranging). For mapping, we use a 3D data acquisition system with a 2D LIDAR and a rotation mechanism and takes a sequence of point clouds. We adopt an NDT (Normal Distribution Transform)-based ego-motion estimation method for pose graph generation and optimization for loop closing. For localization, we propose to use a 2D LIDAR on a robot for being matched with a 3D map for a fast and low-cost localization. The mapping and the localization method are validated through the experiments in our campus.

I. INTRODUCTION

There is an increasing demand for mobile services robots which support human life in various ways, and the operational areas of such robots are expected to extend from indoor to outdoor environments. Autonomous navigation capability is essential for mobile robots, and accurate localization is therefore a key function.

Localization using pre-registered information can be divided into map-based (e.g., [1], [2]) and view-based (e.g., [3], [4]). The former utilizes a matching between a geometric map and input sensory data, while the latter is based on an image-to-image matching. This paper pursues the former because using a 3D geometric map is more general and suitable for a large-scale mapping and localization in outdoor.

SLAM (Simultaneous Localization And Mapping) technologies [5] are usually used for autonomous map making by a mobile robot. For a large-scale mapping, loop closing [6] is a key to cope with accumulated pose errors. Recent progress in optimization methods (e.g., [7]) makes it possible to generate a large-scale map efficiently. Once we have a 3D map of the environment, we can adopt 3D scan matching methods like ICP [8] or NDT [9] for robot pose estimation, probably combined with statistical filtering techniques [10].

3D LIDARs such as Velodyne scanners have been gaining popularity due to their ability of obtaining very rich 3D information and it is very suitable for 3D mapping and localization in outdoor environments where various objects including natural ones exist. It is, however, costly to use such a high-definition 3D LIDAR if it should be installed on all autonomous robots/vehicles. Therefore this paper proposes to use a more cost-effective 2D LIDAR as a sensor on a

robot. We experimentally validate the effectiveness of using a 2D LIDAR as the sensor on the robot as long as we have a reasonably accurate 3D map of the environment. Note that 3D sensing is necessary only for the mapping stage which does not have to be carried out in real-time.

The contributions of the paper is twofold. One is to experimentally evaluate a 3D LIDAR-based large-scale outdoor mapping in our campus including large loops. The other is show the effectiveness of on-line localization using 3D-2D matching by off-line and on-line experiments.

The rest of the paper is organized as follows. Section II describes related work. Section III describes a 3D mapping with loop closure and shows the results for our campus. Section IV describes a localization method based on a 3D map-2D scan matching and experimental evaluations. Section V concludes the paper and discusses future work.

II. RELATED WORK

A. 3D mapping with loop closure

3D mapping is done by integrating a sequence of 3D scans using the estimated robot poses. When a robot travels by a long distance with loops, it may suffer from accumulated ego-motion estimation errors thereby reducing the accuracy of pose estimation. Therefore loop closing is the most important step especially in large-scale mapping [6], [11], [7].

Many image-based loop closure detection methods [12], [13], [14] have been proposed. They characterize locations with an image feature such as BOVW (bag of visual words) [15] and compare the current image with past ones to find loop closures. Integration of 2D image and 3D shape or point features are also proposed [16], [17]. Calculation of image features is relatively costly and image-based matching could suffer from a sensitivity to illumination changes.

B. Map-based localization

Given a map of the environment, a robot can localize itself by matching the current input with the map. In probabilistic localization approaches (i.e., Markov localization [5]) using a grid or voxel map, the similarity between the map and the current scan is used as the likelihood of observation [18].

If we have a precise 3D map, localization is to register the current scan with the map. In this registration, many scan matching approaches can be adopted such as ICP (iterative closest point) [8], GICP (Generalized ICP) [19], NDT (Normal Distribution Transform) [9]). For so-called 6D localization, which estimates full six degrees of freedom of



Fig. 1. A 3D scanning system.

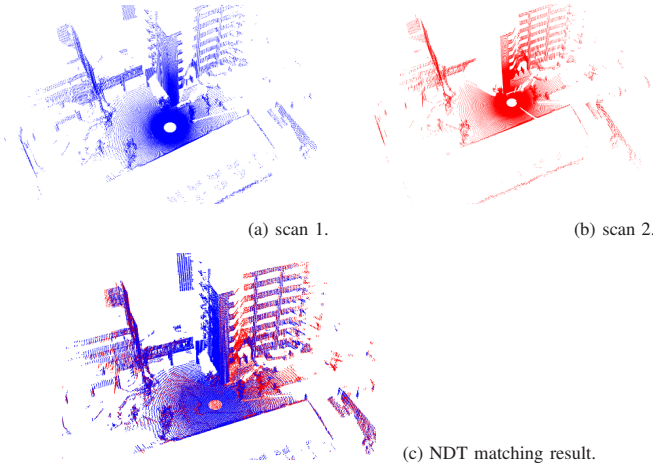


Fig. 2. NDT matching of 3D scans.

robot pose, 3D LIDARs are usually used but they suffer from a high cost and a relatively long computation.

III. 3D MAPPING WITH LOOP CLOSURE

A. 3D scanning

We use a combination of a 2D LIDAR (Laser Imaging Detection and Ranging, LMS151 by SICK) and a pan-tilt (p/t) unit (PTU47 by FLIR, using its one axis) to acquire 3D scans (see Fig. 1). We rotate the p/t unit with a fixed speed while taking data by the LIDAR. Using the system, we can acquire the scans as shown in Fig. 2(a)(b). LMS151 can acquire 1081 point data per scan for a 270° field of view with 0.25° angular resolution. We rotate PTU47 by 180° in about twenty seconds for getting 357 scans. As a result, the system can acquire 385,917 point data by one observation.

B. 3D mapping using NDT and pose graph optimization

The edges of a pose graph are usually generated by the following two ways. One is for a pair of consecutive poses and given by an ego-motion estimation between them. The other is by loop closure which finds a revisit of the (almost) same location.

1) *Ego-motion estimation using NDT*: We estimate ego-motions by comparing two consecutive 3D scans. Among various scan matching approaches, we compared three methods, namely, ICP, GICP, and NDT. We used PCL (point cloud library) [20] implementations and compared for a route of about $1.2km$ in our campus. The route includes various scenes such as building-rich regions and tree-rich ones, and

NDT shows the most robust performance and efficiency. We therefore select NDT for the scan matching for ego-motion estimation.

In NDT, a relative pose between two point clouds (we call them a reference point cloud and an input point cloud) is calculated as follows.

- (1) The reference point cloud is divided into voxels and the point distribution in each voxel is approximated by a Gaussian.
- (2) A matching score is calculated by:

$$\text{score} = - \sum_{k=1}^N \exp \left\{ - \frac{(\mathbf{x}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_k)}{2} \right\}, \quad (1)$$

where N is the number of input points, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and the covariance matrix for the voxel which includes the k th input point \mathbf{x}_k transformed by the current relative pose.

- (3) Update the relative pose using a Quasi-Newton method.
- (4) Repeat steps (2) and (3) until convergence.

2) *Loop closure detection*: We use two ways of detecting loop closures, with and without GPS readings. After candidates pairs of loop closing poses, they are verified using the NDT-based matching.

a) *Loop closure candidate detection based on GPS data*: The first one uses GPS data for detecting pairs of poses which are sufficiently near to each other, and then verifies them with calculating relative poses using NDT. The candidate detection step requires a distance calculation between poses. Following Hubeny's formula [21], the distance D between two poses is given by:

$$D = \sqrt{(Md\phi)^2 + (N \cos(\Phi)d\lambda)^2}, \quad (2)$$

$$M = \frac{6335439}{\sqrt{(1 - 0.006694 \sin^2(\Phi))^3}}, \quad (3)$$

$$N = \frac{6378137}{\sqrt{1 - 0.006694 \sin^2(\Phi)}}, \quad (4)$$

where M and N are the radius of the meridian and the prime vertical circle, respectively, Φ is the average of two latitude values, $d\phi$ and $d\lambda$ are the difference of latitude and longitude of the two locations, respectively. We use GPS PathfinderPro by Tremble as a GPS receiver, and set the threshold for the distance to $5m$. If the distance between two poses is less than the distance threshold, they are judged as a loop closure candidate.

b) *Loop closure candidate detection based on a shape signature and the accumulated position estimates*: The second way of detecting loop closures does not use GPS data but uses a shape signature and the absolute position estimation through the accumulation of ego-motion estimates. We use the classification of each voxel by [9] and describe a scene (i.e., 3D scan at a pose) by a histogram of the following three classes. Let the eigenvalues of the distribution of a voxel be $\lambda_1 \leq \lambda_2 \leq \lambda_3$ and classify it as follows:

- The distribution is linear if $\lambda_2/\lambda_3 \leq 0.1$.

- The distribution is planar if they are non-linear and $\lambda_1/\lambda_2 \leq 0.1$.
- Otherwise, the distribution is spherical.

The voxel is $5m$ cube and use points within $20m$ from the robot are used for histogram calculation. If two poses is within a certain distance using the accumulated position estimates and the histogram is similar enough, they are judged as a loop closure candidate. This signature is very simple to calculate but effective as shown in the mapping results below.

c) Candidate verification by NDT: We apply NDT to detected candidates to calculate the relative pose and the degree of matching (i.e., the root mean squared error (RMSE)). The RMSE is calculated by:

$$e_{RMSE} = \frac{1}{N} \sum_{k=1}^N \sqrt{dx_k^2 + dy_k^2 + dz_k^2}, \quad (5)$$

where dk_* is the positional difference between an input point and the corresponding nearest-neighbor reference point in each axis, and N is the number of input points. If this error is less than a certain threshold (currently, $1m$), we add a new edge between the poses with the estimated relative pose.

3) *Optimization:* Pose graph optimization is to calculate the optimal set of poses by solving the non-linear optimization problem in the following form [7]:

$$F(\mathbf{x}) = \sum_{i,j} e(\mathbf{x}_i, \mathbf{x}_j, z_{ij})^T \Omega_{ij} e(\mathbf{x}_i, \mathbf{x}_j, z_{ij}), \quad (6)$$

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}), \quad (7)$$

where $\mathbf{x} = (\mathbf{x}_1^T, \dots, \mathbf{x}_n^T)^T$ is a set of pose parameters and \mathbf{x}_i is the robot pose at i th observation; Ω_{ij} is the information matrix representing a constraint between i th and j th pose, and z_{ij} is their relative pose given by the ego-motion estimation; z_{ij} is represented by a concatenation of relative translation \mathbf{t}_{ij} and relative rotation \mathbf{q}_{ij} in a quaternion form; $e(\mathbf{x}_i, \mathbf{x}_j, z_{ij})$ is the error function evaluating the difference between the estimated and the observed relative pose.

We use `g2o` library [7] for pose graph optimization. The information matrix is related to the certainty of relative pose estimation using NDT. We thus define Ω_{ij} as follows:

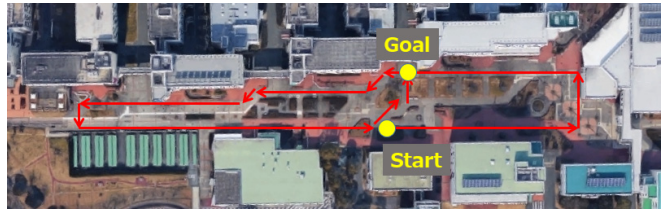
$$\Omega_{ij} = \begin{pmatrix} \Omega_{ij}^{pos} & \mathbf{0} \\ \mathbf{0} & \Omega_{ij}^{rot} \end{pmatrix}, \quad (8)$$

$$\Omega_{ij}^{pos} = \mathbf{I}/(\sigma_{pos}^2 \cdot e_{ij}), \quad (9)$$

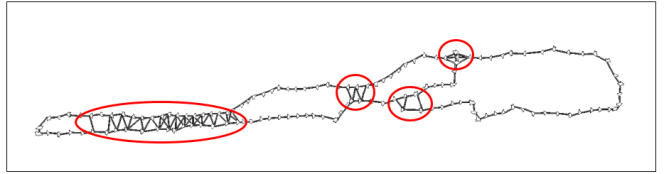
$$\Omega_{ij}^{rot} = \mathbf{I}/(\sigma_{rot}^2 \cdot e_{ij}), \quad (10)$$

where \mathbf{I} is an identify matrix, e_{ij} is given by e_{RMSE} in eq. (5), and σ_{pos}^2 and σ_{rot}^2 are the variance of positional and rotational errors (currently both are set to 1.0).

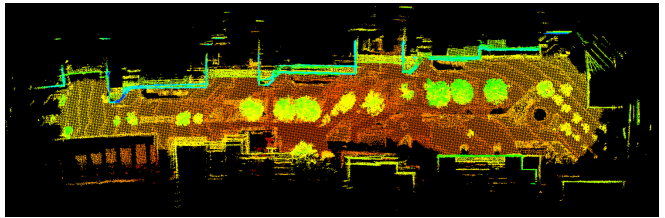
Fig. 3 shows the result and the effect of loop closing. Fig. 3(a) is a route for acquiring 3D scans (b) shows the final pose graph with markers on the added edges by the loop closing. Fig. 3(c) and (d) compare the map before and after the pose graph optimization. Inconsistencies of the map at many places have been solved in (d). The final graph has 183 nodes and 262 edges and the optimization took $0.0133[sec.]$



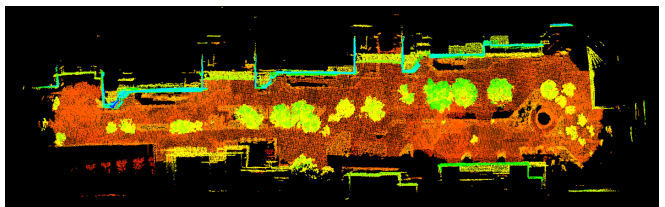
(a) The route for acquiring 3D scans.



(b) The graph after loop closure detection.



(c) The map before loop closing.



(d) The map after loop closing.

Fig. 3. Result of pose graph optimization. The colors in the map indicate the height.

using a PC with Core i7-6700K and 16GB memory.

C. Large-scale mapping result

Fig. 4 shows the result of generating a larger map. Fig. 4(a) shows the route for acquiring 283 scans and Fig. 4(b) shows the mapping result. Fig. 4(c) shows a 3D view of a part of the map. We can see a good map is obtained.

IV. ON-LINE LOCALIZATION WITH 3D-2D MATCHING

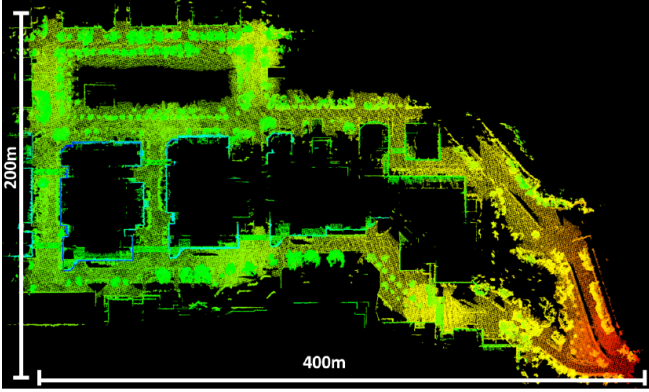
Map-based localization is performed by matching between a map and the current sensor readings. To cope with abrupt sensing errors and/or feature-scarce environments, filtering techniques are usually used [5]. In the case of outdoor localization using a 3D map, a usual way is to use a 3D sensor for 3D-3D matching. However, it is costly to equip a 3D LIDAR for every robot just for localization. We therefore pursue an approach to localization from a combination of a 3D map and a 2D LIDAR.

1) *UKF-based localization:* We use Unscented Kalman filter (UKF) [22] for localization. The state vector \mathbf{x}_t is defined as:

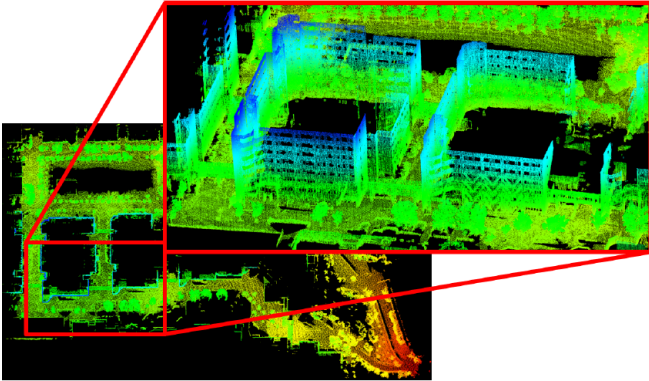
$$\mathbf{x}_t = (\mathbf{p}_t, \mathbf{v}_t, \mathbf{q}_t, \mathbf{b}_t^{gyro})^T, \quad (11)$$



(a) The route for acquiring 3D scans in a larger region.



(b) The mapping result. Colors indicate the heights of the points.



(c) 3D view of a part of the map.

Fig. 4. Result of mapping of a larger region.

where \mathbf{p} is the position, \mathbf{v} is the translational velocity, \mathbf{q}_t is the orientation in the quaternion form; \mathbf{b}_t^{gyro} is the bias for the gyroscope. The prediction step of the UKF is described as:

$$\mathbf{x}_t = (\mathbf{p}_{t-1} + \mathbf{v}_{t-1} \cdot \Delta t, \mathbf{q}_{t-1} \cdot \Delta \mathbf{q}_t, \mathbf{v}_{t-1}, \mathbf{b}_{t-1}^{gyro})^T, \quad (12)$$

where Δt is the duration between t and $t-1$ and $\Delta \mathbf{q}_t$ is the rotation during Δt caused by the bias-compensated angular velocity from the gyroscope, also in the quaternion form.

The correction step uses the measurement of the robot pose, defined as:

$$\mathbf{z} = (\mathbf{p}_t^{obs}, \mathbf{q}_t^{obs})^T. \quad (13)$$



Fig. 5. Test route 1 for evaluating the localization accuracy.

The translation \mathbf{p}_t^{obs} and the rotation \mathbf{q}_t^{obs} part calculated by matching the current 2D scan with the 3D map using the NDT representation of the map. Note that the procedure for the 3D-3D matching for mapping and that for the 3D-2D matching for localization are the same with only difference in the number of 3D points in the current input point cloud. We use the result of the prediction step as an initial estimate of the robot pose, and search a fixed range of voxels for the matched voxel of each input 3D point. The range is specified as a cubic set of voxels with the edge length being fifteen. The uncertainty is calculated as the inverse of the information matrix in eq (8).

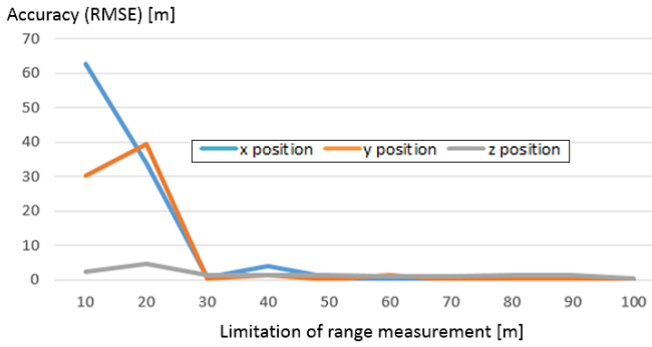
2) Experiments:

a) *Off-line experiments:* We first evaluate the accuracy of the proposed localization method. Since it is difficult to obtain the ground truth for the robot pose, we used a 3D LIDAR (HDL-32e by Velodyne) to take full 3D scans as the robot moves, and assume its localization results as the true values. In addition, for simulating a 2D LIDAR, we extract only the horizontal scan out of 32 scans of the sensor and use it for localization.

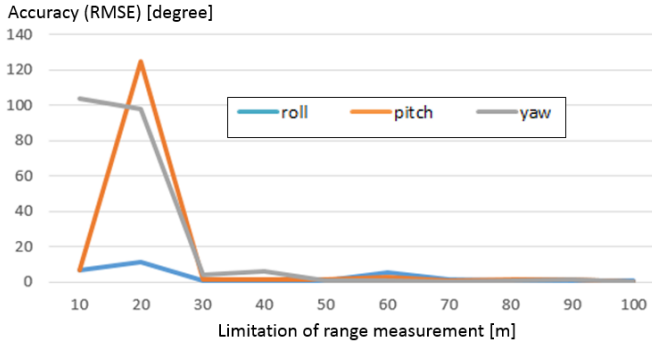
Fig. 5 indicates the first route used for evaluation, which is the loop at the top-left part of the environment (see Fig. 4(a)); the route is almost flat. The lengths of the longer and the shorter edge are approximately 130m and 75m, respectively. We performed five experimental runs and calculated the RMSE values for all of six degrees of freedom. We also simulate various limitations in the range measurements (i.e., maximum measurable range); we see how accuracy changes as the limitation changes.

Fig. 6 shows the RMSE error in the translational and the rotational elements of the localization (i.e., the robot pose estimation) for various maximum measurable ranges. Table I summarizes the accuracy for two maximum ranges. In this experiment, when the maximum measurable range is equal to or larger than 30m, the proposed localization method can provide a sufficient accuracy for autonomous navigation. In the data for the 60m maximum range, the roll and the pitch value happen to have larger errors than the 30m case; however, these values are not very important for the robot moving on a road-like, almost flat surface and, at the same time, the yaw accuracy, which is more important for navigation, is improved for the 60m range case.

Fig. 7 indicates the second route used for evaluation, which has a variation in the road height. The region around the



(a) Accuracy in the position elements.



(b) Accuracy in the orientation elements.

Fig. 6. Accuracy of the pose estimation for various limitations of range measurement for test route 1.

TABLE I

ACCURACY IN THE LIMITATIONS OF 30m AND 60m IN THE RANGE MEASUREMENTS FOR TEST ROUTE 1.

elements	maximum range is 30m	maximum range is 60m
x [m]	0.722	0.212
y [m]	0.499	0.121
z [m]	1.426	0.925
roll [deg]	0.997	4.254
pitch [deg]	1.364	2.754
yaw [deg]	3.783	1.027

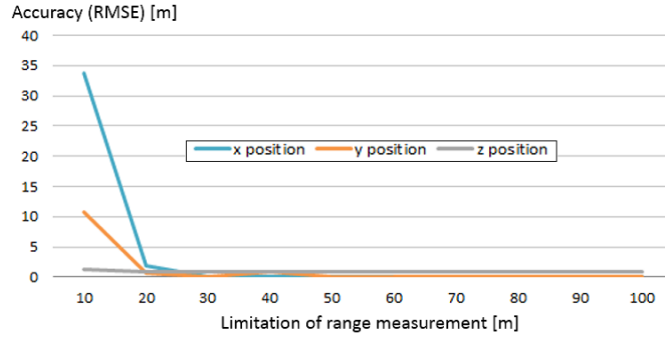
first corner is lower than the other places on the route. Data collection and analyses have been conducted similarly to the first route case.

Fig. 8 shows the RMSE error in the translational and the rotational elements of the localization (i.e., the robot pose estimation) for various maximum measurable ranges. Table II summarizes the accuracy for two maximum ranges. Since the average distance to objects (i.e. buildings) are smaller than in the route 1 case, the necessary measurable range of LIDAR looks smaller and the overall accuracy is better. Also in this experiment, extending the measurable range of the LIDAR basically increases the accuracy.

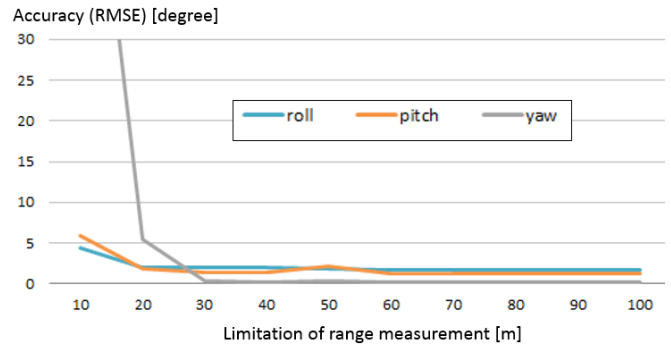
b) On-line experiments: We performed on-line experiments using the LMS151 as the 2D LIDAR. The processing speed of one cycle including localization and robot control is about 10fps, which is sufficiently fast for autonomous navigation. Fig. 9(a) shows the trajectory when the robot moved on the first two edges of the route in Fig. 5. Fig. 9(b) shows a snapshot of localization; red points in the 3D map



Fig. 7. Test route 2 for evaluating the localization accuracy.



(a) Accuracy in the position elements.



(b) Accuracy in the orientation elements.

Fig. 8. Accuracy of the pose estimation for various limitations of range measurement for test route 2.

and a white sphere indicate the data obtained by the current 2D scan and the estimated robot position, respectively.

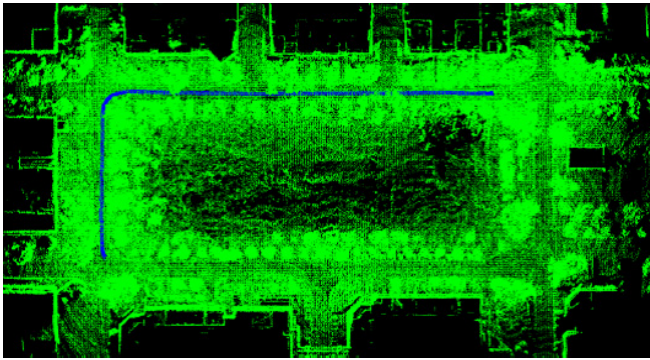
V. CONCLUSIONS AND FUTURE WORK

This paper describes a large-scale 3D outdoor mapping method and a 6D localization method. 3D mapping is realized by combining an NDT-based ego-motion estimation using 3D scans and a robust loop closure detection using either GPS signals or a shape signature followed by verifica-

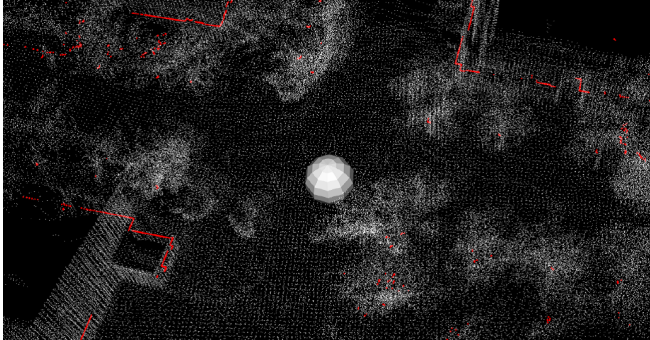
TABLE II

ACCURACY IN THE LIMITATIONS OF 30m AND 60m IN THE RANGE MEASUREMENTS FOR TEST ROUTE 2.

elements	maximum range is 30m	maximum range is 60m
x [m]	0.110	0.082
y [m]	0.117	0.069
z [m]	0.861	0.816
roll [deg]	2.079	1.120
pitch [deg]	2.175	2.209
yaw [deg]	0.337	0.321



(a) The estimated trajectory (blue line) by the on-line localization.



(b) Snapshot of on-line localization. Red points: current 2D scan; white sphere: estimated robot position.

Fig. 9. Result of on-line localization.

tion and relative pose estimation using NDT. We have shown the method can generate a campus-wide 3D map reasonably accurately.

The localization method is based on a novel idea of matching a 3D map with 2D scans. Use of a 2D LIDAR on a robot has advantages of a low-cost and a computational efficiency. We tested the method in two locations in our campus to compare the accuracy between a usual 3D-3D matching-based method and ours. The comparison results show that the proposed method has a sufficient accuracy for autonomous navigation.

The generated map looks reasonably good qualitatively but has not been evaluated quantitatively. Comparison with the results with more accurate systems such as FARO Focus3D and/or evaluation using publicly-available datasets such as KITTI dataset would be preferable. Combining the current geometric map with semantic information such as building names and location categories will be necessary to communicate with or accepting commands from people, using object recognition and categorization techniques. Using such a semantically-enhanced map, we are planning to make a robot call system which can guide people or carry items autonomously with a human-friendly interface.

REFERENCES

[1] H. Andreasson and A.J. Lilienthal. 6D Scan Registration using Depth-Interpolated Local Image Features. *Robotics and Autonomous Systems*, Vol. 58, No. 2, pp. 157–165, 2010.

[2] K. Irie, T. Yoshida, and M. Tomono. Mobile Robot Localization Using Stereo Vision in Outdoor Environments under Various Illumination Conditions. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5175–5181, 2010.

[3] M.J. Milford and G.F. Wyeth. SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights. In *Proceedings 2012 IEEE Int. Conf. on Robotics and Automation*, 2012.

[4] Y. Inoue, J. Miura, and S. Oishi. Outdoor Robot Navigation Based on View-based Global Localization and Local Navigation. In *Proceedings of the 14th Int. Conf. on Intelligent Autonomous Systems*, 2016.

[5] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[6] S. Thrun and M. Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *Int. J. of Robotics Research*, Vol. 25, No. 5-6, pp. 403–429, 2006.

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g²o: A General Framework for Graph Optimization. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Automation*, pp. 3607–3613, 2011.

[8] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239–256, 1992.

[9] M. Magnusson, A. Lilienthal, and T. Duckett. Scan Registration for Autonomous Mining Vehicles using 3D NDT. *J. of Field Robotics*, Vol. 24, No. 10, pp. 803–827, 2007.

[10] D. Fox, W. Burgard, and S. Thrun. Markov Localization for Mobile Robots in Dynamic Environments. *J. of Artificial Intelligence Research*, Vol. 11, pp. 391–427, 1999.

[11] E. Olson, J. Leonard, and S. Teller. Fast Iterative Alignment of Pose Graphs with Poor Initial Estimates. In *Proceedings of 2006 IEEE Int. Conf. on Robotics and Automation*, pp. 2262–2269, 2006.

[12] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings 2006 IEEE Int. Conf. on Robotics and Automation*, pp. 1180–1187, 2006.

[13] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. of Robotics Research*, Vol. 27, No. 6, pp. 647–665, 2008.

[14] T. Botterill, S. Mills, and R. Green. Bag-of-Words-driven, Single-Camera Simultaneous Localization and Mapping. *J. of Field Robotics*, Vol. 28, No. 2, pp. 204–226, 2011.

[15] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of 2013 IEEE Int. Conf. on Computer Vision*, pp. 1470–1477, 2003.

[16] Y. Bok, Y. Jeong, and D.-G. Choi and I.S. Kweon. Capturing Village-level Heritages with a Hand-held Camera-Laser Fusion Sensor. *Int. J. of Computer Vision*, Vol. 94, No. 1, pp. 36–53, 2011.

[17] C. Brand, M.J. Schuster, H. Hirschmüller, and M. Suppa. Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM. In *Proceedings of 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5670–5677, 2015.

[18] K. Konolige. Markov Localization using Correlation. In *Proceedings of 16th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 1154–1159, 1999.

[19] S. Segal, D. Hähnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, 2009.

[20] Point Cloud Library. <http://pointclouds.org/>.

[21] Hubeny's distance calculation formula. <http://vldb.gsi.go.jp/sokuchi/surveycalc/algorithm/ellipse/ellipse.html>. accessed on 1/15/2017.

[22] S.J. Julier and J.K. Uhlmann. Unscented Filtering and NonLinear Estimation. *Proceedings of IEEE*, Vol. 92, No. 3, pp. 401–422, 2004.