ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/327119253

Teaching-Playback Navigation Without a Consistent Map

Article *in* Journal of Robotics and Mechatronics · August 2018

DOI: 10.20965/jrm.2018.p0591

CITATION
1

READS
190

3 authors:

Naoki Akai
Nagoya University
62 PUBLICATIONS **254** CITATIONS
SEE PROFILE

Y. Morales
Standard Cognition
95 PUBLICATIONS **878** CITATIONS
SEE PROFILE

H. Murase
Nagoya University
367 PUBLICATIONS **8,317** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Automated Driving View project

Personal Mobility Vehicle Shared Control and Automation View project

All content following this page was uploaded by Naoki Akai on 22 August 2018.

The user has requested enhancement of the downloaded file.

Paper: Rb30–4–9168:

Teaching-Playback Navigation Without a Consistent Map

Naoki Akai[1], Luis Yoichi Morales[1], and Hiroshi Murase[2]

[1]464-8601, Furo, Chikusa, Nagoya, Japan, Institute of Innovation for Future Society (MIRAI), Nagoya University
[2]464-8603, Furo, Chikusa, Nagoya, Graduate School of Information Science, Nagoya University
E-mail: {akai, morales_yoichi}@coi.nagoya-u.ac.jp, murase@nagoya-u.jp
[Received 00/00/00; accepted 00/00/00]

This paper presents a teaching-playback navigation method that does not require a consistent map built using simultaneous localization and mapping (SLAM). Many open source projects related to autonomous navigation including SLAM have been made available recently; however, autonomous mobile robot navigation in large-scale environments is still difficult because it is difficult to build a consistent map. The navigation method presented in this paper uses several partial maps to represent an environment map. In other words, the complex mapping process is not necessary to begin autonomous navigation. In addition, the trajectory that the robot travels in the mapping phase can be directly used as a target path. As a result, teaching-playback autonomous navigation can be achieved without any off-line processes. We tested the navigation method using log data taken in the environment of the Tsukuba Challenge and the testing results show its performance. We provide source code for the navigation method, which includes modules required for autonomous navigation (`https://github.com/NaokiAkai/AutoNavi`).

Keywords: Autonomous Mobile Robot, Teaching-Playback Navigation, Localization, Obstacle Avoidance, Tsukuba Challenge

1. Introduction

Recently, many open source projects that allow us to easily operate many kinds of mobile robots have been made available [1, 2, 3]. Specifically, robot operating system (ROS) includes many resources related to autonomous navigation and are widely used in the mobile robot field [4]. ROS-based open source projects related to simultaneous localization and mapping (SLAM) have also been provided by many researchers [5, 6, 7, 8]. However, autonomous mobile robot navigation in large-scale environments is still difficult because it is difficult to build a consistent map[1]. This paper presents a teaching-playback navigation method that does not require a consistent map.

1. We took log data of Velodyne 32, IMU, and odometry in the TC environment and applied 3D LiDAR-based SLAM methods. However, building a consistent map was difficult because we need to know its parameters and adjust them while considering the characteristics of the environment.

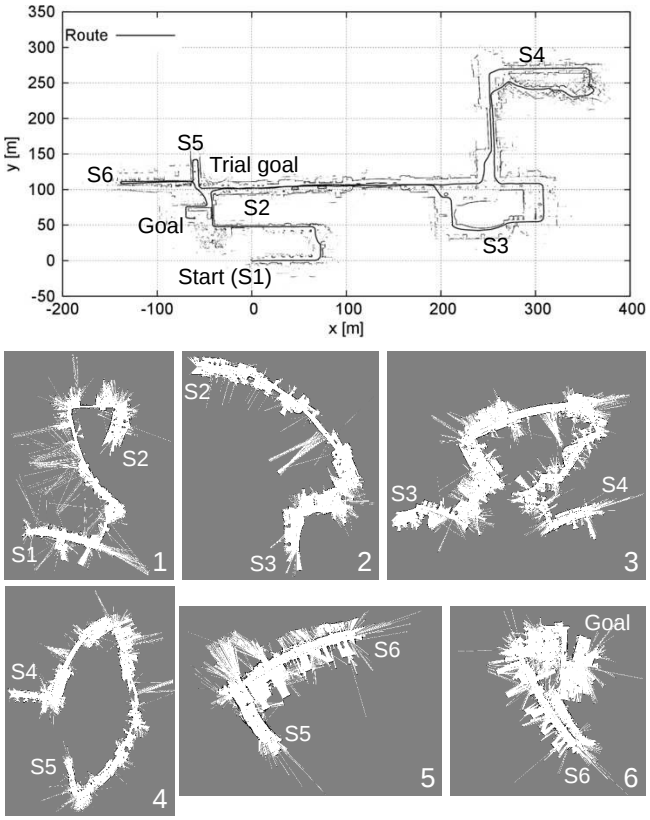

Fig. 1. Geometric landmark map of the environment of TC2017 and the target path that robots must navigate. In the presented navigation method, several partial maps, which were built using odometry trajectories, are used for representing the environment (we used six maps, as shown at the bottom). The corresponding start points for building the partial maps are shown for both top and bottom figures.

Figure 1 shows an example of the map representation technique in the presented navigation method. The trajectories and maps shown at the top were built using log data taken from the Tsukuba challenge (TC) 2017 [9]. There are several loops in the environment. Where a "loop" indicates that robots must travel over the same area in the mapping phase. It is necessary to apply SLAM to obtain a consistent map that does not have any overlaps along the loops. The landmark map shown at the top of Fig. 1 was built using the three-dimensional (3D) light detection

and ranging (LiDAR)-based SLAM method presented in [10] (we obtained this map using a trial–and-error process). In contrast, the presented navigation method uses several maps to represent the environment map shown at the bottom of Fig. 1. These maps do not have any overlaps; SLAM was not used because we changed the map when we encountered a loop. The complex mapping process can be omitted before autonomous navigation. In addition, the trajectory that the robot traveled during the mapping phase can be directly used as a target path. As a result, teaching-playback autonomous navigation can be achieved without any off-line processes.

The contribution of this work is creating the open source teaching-playback navigation project. The presented navigation method also includes two-dimensional (2D) LiDAR-based localization, obstacle avoidance, and simulator modules. This paper presents the details of this open source project.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the mapping approaches used in recent TC events. Section 3 provides an overview of the presented navigation method. Section 4 describes the modules used in the navigation method. Section 5 evaluates the navigation method using log data taken during TC2017. Section 6 presents our conclusions.

2. Mapping approaches in the Tsukuba Challenge

This section briefly summarizes the mapping approaches used during the recent Tsukuba Challenge (TC). To summarize these approaches, we referred to the last two years' technical reports, which were published by the TC committee [11, 12].

At TC, ROS has been widely used recently. To build the environment map, many teams used GMapping [5]. GMapping is a grid-map-based SLAM method using a Rao–Blackwellized particle filter (RBPF). However, building a consistent map using GMapping with short-range 2D LiDAR is not easy because the TC environment is large and includes areas in which there are no suitable geometric landmarks. We assume that the 2D LiDAR has a maximum measurement range of less than 30 m for short-range LiDAR. Some teams tried to build several partial maps using GMapping and manually combined them off-line to obtain a consistent map.

Several teams used long-range multilayer LiDAR such as VLP-16 [13], for which the measurement range is over 100 m. An approach that creates a virtual 2D scan from multilayer LiDAR measurements to apply GMapping was proposed. It has been reported that a consistent map can be built using GMapping if a long-range scan is available. A similar approach that uses several LiDARs to extend the measurable range in one scan was proposed and a consistent map could be obtained. In contrast, Hara *et al.* proposed an extension of GMapping that enables one to build a consistent map even when a long-range 2D scan is not available [14]. In their approach, scan data is accumulated and a virtual long-range scan is created to calculate the likelihood of particles. To apply this approach, an accurate dead reckoning system is required for accurate accumulation of scan observations in landmark-less areas. If an accurate dead reckoning system is not available, large numbers of particles might be required to generate many hypotheses.

Several teams tried to build a 3D map using NDT scan matching (included in Autoware) [1]. In Autoware, Takeuchi's and Magnusson's methods are implemented [10, 15]. However, the NDT scan matching method included in Autoware does not have an explicit loop-closure function and a trial-and-error process is needed to build a consistent map. In addition, tilting of the ground surface will be a problem if any other constrained creation methods are not applied for height modification. One team used cartographer SLAM [8] to build a 3D map during TC2017. Although they built a 3D consistent map, they reported that a lot of time was needed to adjust its parameters.

As we have mentioned, it is difficult to build a consistent map using short-range LiDAR. As a result, teaching-playback navigation will also be difficult. One team achieved teaching-playback navigation in TC2017 because they developed excellent SLAM techniques, as presented in [16, 17]. Our approach does not employ these excellent SLAM techniques, but it allows us to achieve teaching-playback navigation.

3. System overview

Figure 2 shows an overview of the presented navigation system. Our system is implemented using a ROS architecture. We assume that the robot is equipped with 2D LiDAR and encoders for computing odometry. In the map building phase, we manually operate the robot and build the occupancy grid maps. Just before we start a loop (i.e., the case in which the current scan begins to pass through areas where the previous scan has already passed), we manually send a signal to change the map and restart mapping from scratch with the initial robot pose, $\mathbf{x}_1 = \mathbf{0}^T$. Where $\mathbf{x}$ is the robot pose, which includes the 2D position, x and y, and heading direction, θ. After the map building phase, we obtain several partial maps, ${}^i\mathbf{m}$, and corresponding robot trajectories, ${}^i\mathbf{x}_{1:iN}$. The trajectories are used as the target path for autonomous navigation.

In the autonomous navigation phase, we use three modules: localizer, path follower, and server. The localizer estimates the robot pose on the given map. The path follower receives the estimated pose and computes a velocity command for following the target path. The obstacle avoidance function is included in the path follower and can also compute the velocity command for avoiding obstacles in front of the robot.

When the robot reaches the goal of the $(i-1)$-th trajectory, ${}^{i-1}\mathbf{x}_{i-1N}$, the path follower first stops the robot and sends a signal to the localizer. The localizer then sends a signal to server for requesting a new map. After the server

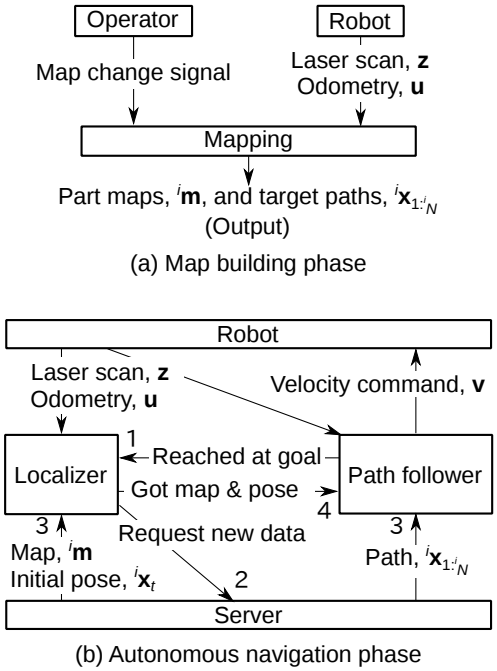

Fig. 2. System overview of the navigation system, (a) map building and (b) autonomous navigation phases.

receives the signal, it publishes the next map, $^i\mathbf{m}$, and path, $^i\mathbf{x}_{1:^iN}$, to the localizer and path follower. Because $^{i-1}\mathbf{m}$ and $^i\mathbf{m}$ use different coordinates from each other, the robot pose in the $(i-1)$-th map must be transformed into the i-th map coordinates to continuously estimate the robot pose spanning different maps. This transformation will be described in the next subsection. The server computes this transformation and publishes the transformation result as the initial pose, $^i\mathbf{x}_t$. The localizer receives the map and initial pose and then sends a signal (indicating that the localizer received the new map and initial pose) to the path follower. Finally, autonomous navigation restarts after that the path follower receives the new path and signal from the localizer.

3.1. Robot pose transformation between part maps

Let $^{i-1}\mathbf{x}_{1:^{i-1}N}$ be the trajectory that the robot traveled in the $(i-1)$-th mapping phase and $^{i-1}\mathbf{x}_{^{i-1}N}$ be the pose at which we changed the map. $^{i-1}\mathbf{x}_{^{i-1}N}$ and $^i\mathbf{x}_1$ represent the same pose in the real environment and $^i\mathbf{x}_1$ is set to $\mathbf{0}^T$. Therefore, we can transform the robot pose in the $(i-1)$-th map coordinates at time t, $^{i-1}\mathbf{x}_t$, to the pose in the i-th map coordinates as:

$$^i\mathbf{x}_t = \begin{pmatrix} R^{-1}(^{i-1}\theta_{^{i-1}N}) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} (^{i-1}\mathbf{x}_t - ^{i-1}\mathbf{x}_{^{i-1}N}), \quad (1)$$

where $R(\cdot)$ is a 2D rotation matrix.

The map changing process yields a problem: past mapped areas will be erased and will not be available. Thus, areas where there are enough landmarks in front of the robot are recommended as places for changing the map. Robot pose estimation might fail if there are no landmarks near the map changing places.

4. Modules

4.1. Mapping

We use the following motion model to update the robot pose based on odometry:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \Delta d_t \cos\theta_{t-1} \\ \Delta d_t \sin\theta_{t-1} \\ \Delta\theta_t \end{pmatrix}, \quad . \quad (2)$$

where $\mathbf{u}_t = (\Delta d_t, \Delta\theta_t)^T$ is the moving amount measured by the encoders. In the mapping phase, we do not use any compensation methods for the accumulated odometry error. Maps are built based on the odometry trajectory. We employ ray casting using a laser scan measurement, $\mathbf{z}_t$, and update the occupancy rate of each map cell, $p(m_i)$, using a binary Bayes filter:

$$l = \log\left(\frac{p(m_i)}{1-p(m_i)}\right) + \log\left(\frac{p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})}{1-p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})}\right), \quad (3)$$

$$p(m_i) \leftarrow \frac{1}{1+\exp(-l)}. \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (4)$$

where $p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$ is an observation model. We use the likelihood-field model [18] as the observation model; it is denoted as:

$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \begin{pmatrix} z_{\text{hit}} \\ z_{\max} \\ z_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{hit}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \\ p_{\max}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \\ p_{\text{rand}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \end{pmatrix}, \quad (5)$$

where z_{hit}, $z_{\max}$, z_{rand}, $p_{\max}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$, and $p_{\text{rand}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$ are constants and $p_{\text{hit}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$ is given by:

$$p_{\text{hit}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \eta\exp\left(-\frac{(z_t - d(p(\mathbf{x}_t,\mathbf{z}_t),\mathbf{m}))^2}{2\sigma_z^2}\right), \quad (6)$$

where η is a normalization constant and z_t and σ_z^2 are the measurement distance and variance of 2D LiDAR, respectively, and $p(\cdot)$ and $d(\cdot)$ are functions that return the map frame 2D point where the laser scan hit and the nearest distance from the given 2D point to the obstacle on the map, respectively.

4.2. Localizer

We use augmented Monte Carlo localization (AMCL) as the localization algorithm [18, 19]. AMCL is already included in ROS, but we have manually implemented it to more easily adjust parameters. AMCL requires motion and observation models; we used the models shown in equations (2) and (5) as the required models.

Our AMCL includes a dynamic scan points rejection algorithm[2] for increasing robustness against moving obstacles. Where the dynamic scan points mean measurement obtained from obstacles that are not existing on the

2. **Algorithm test_range_measurement** shown in Table 8.5 of [18].

given map. To detect dynamic scan points, we use a beam model-based observation model [18] denoted as:

$$p_{\mathrm{beam}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \begin{pmatrix} z_{\mathrm{short}} \\ z_{\mathrm{hit}} \\ z_{\mathrm{max}} \\ z_{\mathrm{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\mathrm{short}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \\ p_{\mathrm{hit}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \\ p_{\mathrm{max}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \\ p_{\mathrm{rand}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \end{pmatrix}, \quad (7)$$

where z_{short} is constant and $p_{\mathrm{short}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$ is denoted as:

$$p_{\mathrm{short}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \eta\lambda\exp(-\lambda z_t), \quad \ldots\ldots \quad (8)$$

where λ is constant. We focus on each scan point, $\mathbf{z}_t^i$; scan points are detected as dynamic scan points when the following condition is satisfied:

$$\frac{\sum_{j=1}^{M} z_{\mathrm{short}} p_{\mathrm{short}}(\mathbf{z}_t^i|\mathbf{x}_t^j,\mathbf{m})}{\sum_{j=1}^{M} p_{\mathrm{beam}}(\mathbf{z}_t^i|\mathbf{x}_t^j,\mathbf{m})} > \chi, \quad \ldots\ldots \quad (9)$$

where M is number of particles and χ ($0 < \chi < 1$) is the threshold. The scan points detected as dynamic points are not used for calculating the likelihood of particles.

4.3. Path follower

The path follower receives a target path, ${}^i\mathbf{x}_{1:iN}$, that is the trajectory that the robot traversed in the i-th mapping phase and generates a velocity command for following the target path based on the PD control. The path follower also has an obstacle avoidance function that can be turned on or off.

The obstacle avoidance node obtains the velocity command from the path follower and judges whether the robot will collide with the obstacle or not. If the robot will collide, the obstacle avoidance node computes the velocity for avoiding obstacles and the velocity command output from the path follower is overwritten. The detail of these modules can be found in [20].

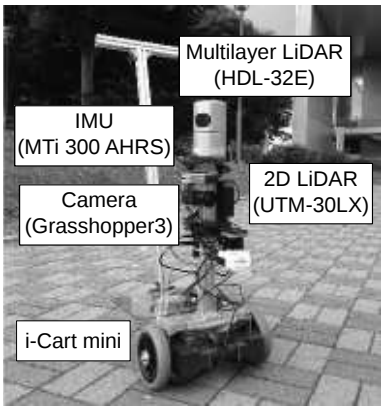

Fig. 3. Robot used in TC2017.

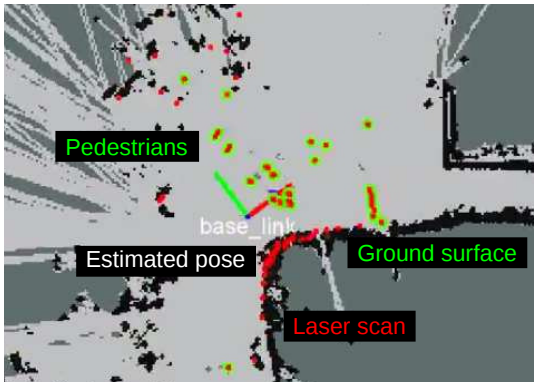

Fig. 4. Example localization result. The red points indicate laser scan data measured by the 2D LiDAR. The scan data includes points that do not match the landmarks shown in black because the 2D LiDAR measures pedestrians and ground surfaces. These points are detected as dynamic points and are drawn in green.

5. Experiment

5.1. Experimental set up

We participated in TC2017 using the robot shown in Fig. 3 for taking log data. The robot has 3D LiDAR and an IMU; these sensors were used to build the 3D consistent map. After building the 3D map, we built a 2D map using the 3D map-based localization result. The 2D consistent map used for the simulation experiment is shown at the top of Fig. 1. We also built partial maps (shown at the bottom of Fig. 1) that are used in the presented navigation method. We took several log data sets and tested the presented localization method using the log data.

5.2. Localization performance

We tried a localization experiment using the log data. As shown in Fig. 1, the maps used in the navigation method are distorted because of the inaccuracy of the odometry. In addition, because the robot does not have a dumper function and the 2D LiDAR is mounted low on the robot, the LiDAR measured much noisy data, e.g., ground surfaces. Figure. 4 shows an example of the localization result. The red points indicate laser scan data measured by the 2D LiDAR. The scan data includes points that do not match the landmarks shown in black, as the 2D LiDAR is measuring pedestrians and ground surface. These points could be detected as dynamic points before calculating the likelihood of particles using the dynamic scan point rejection algorithm, which means that scan points hitting the dynamic obstacles were not used for estimating the robot pose. As a result, we achieved robust localization in dynamic environments despite the LiDAR measuring much noisy data.

In addition, we compared the performance of the observation model with the use of the dynamic scan points rejection algorithm in simulated dynamic environments. Figure 5 shows two likelihood distributions, with (left) and without (right) the rejection algorithm. As we used a simulation for the comparison, we could obtain the ground truth of the robot's pose; the ground truth was

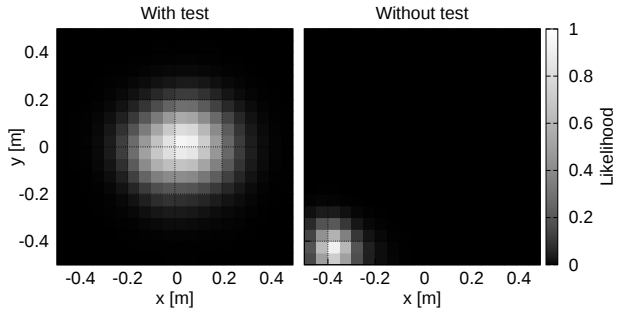

Fig. 5. Likelihood distributions with (left) and without (right) the use of the dynamic scan point rejection algorithm.

set to the origin in each figure. Namely, the observation model is found to work successfully if the likelihood is large near the origin. It is clear that the observation model with the rejection algorithm worked successfully. On the other hand, the peak of the likelihood was moved from the origin based on dynamic obstacles. It was shown that the rejection algorithm improved the robustness of the observation model (localization).

Unfortunately, numerically evaluating the localization result using real sensor data is difficult. Therefore, we have uploaded a video that shows the localization performance[3].

5.3. Simulation experiment

The provided source code also includes simulation modules of 2D LiDAR and odometry measurements. We used the simulation modules for testing the navigation method. In the experiment, we used the 2D consistent map to simulate the laser scan; odometry is simulated based on the velocity command. The localizer subscribes the simulated data and estimates the robot pose on the given maps. The path follower generates a velocity command to follow the target path. The simulated robot pose is updated based on the velocity.

We first evaluated the reproducibility of the presented navigation system with the use of consistent and inconsistent maps. Figure 6 shows the consistent and inconsistent maps built using SLAM (top) and the presented system (bottom) from the start to the trial goal shown in Fig. 1. Because the presented system used the estimated trajectory from the odometry, the built map was distorted. We performed three navigation experiments with the use of each map; Fig. 7 shows the target path and simulated robot trajectories, where the simulated robot trajectory is the ground truth trajectory of the robot.

The top of Fig. 7 shows the results with the consistent map. As can be seen in the figure, high reproducibility was confirmed because all of the trajectories are very close. On the other hand, reproducibility could not be

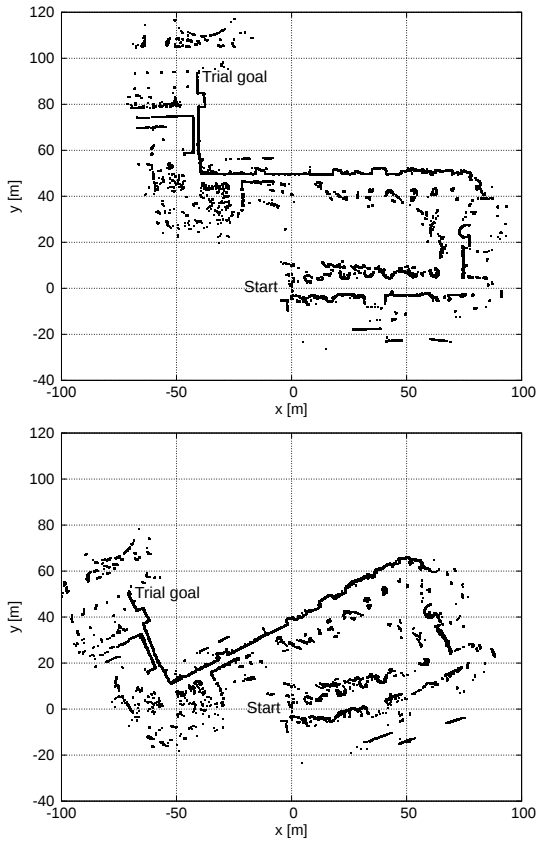

Fig. 6. Built maps using SLAM (top) and the presented system (bottom).

confirmed based on the result from the inconsistent map shown at the bottom of Fig. 7, because the mapping accuracy of the inconsistent map was less than that of the consistent map. The landmarks were blurry in the inconsistent map, yielding inaccurate localization results. This result shows one drawback of the presented navigation system. We discuss this problem further in the next section.

We then tested the navigation performance of the presented system. Figure 8 shows the trajectories of the simulated robot and its odometry. The odometry trajectory was used to build the partial maps. The simulated robot trajectory is the trajectory that the robot passes through during the simulation experiment. As can be seen in the figure, the robot continuously navigated the target path, despite several maps being used to represent the environment map. This result shows that autonomous navigation can be achieved even if we do not have a consistent map building technique. We also uploaded a video that shows the navigation performance of the presented method[4].

3. Naoki Akai, "2D LiDAR-based localization without a consistent map in Tsukuba Challenge 2017," https://www.youtube.com/watch?v=wPKssq1_N6c, uploaded at 15th Dec. 2017.

4. Naoki Akai, "Navigation experiment in simulated Tsukuba Challenge environment," https://www.youtube.com/watch?v=gQf0CUC4omw&t=1s, uploaded at 18th Dec. 2017.

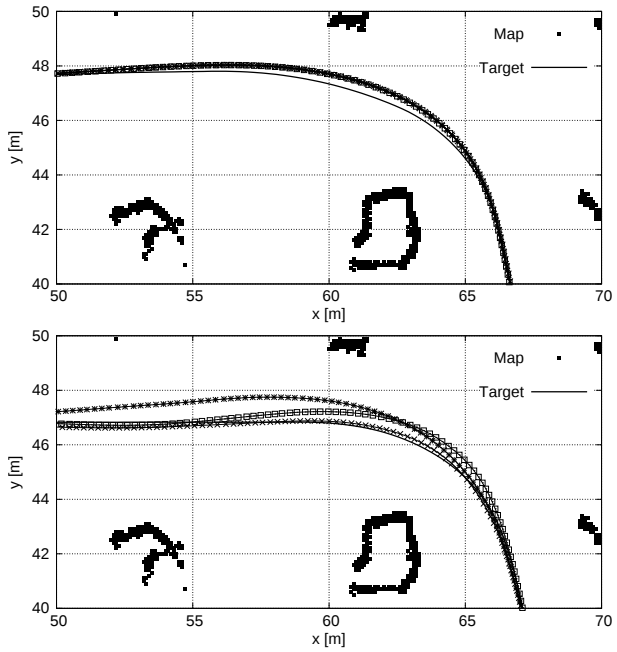

Fig. 7. Target path and simulated robot trajectories using the consistent (top) and inconsistent (bottom) maps. The map coordinates are the same as those shown at the top of Fig. 6.

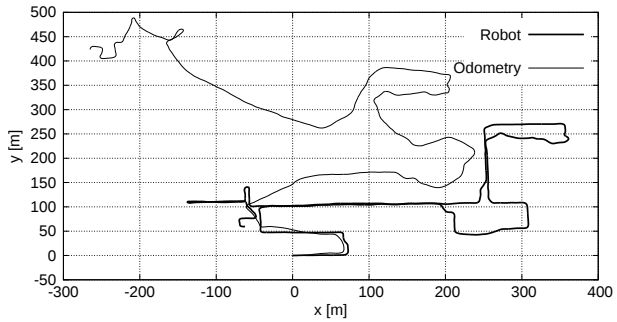

Fig. 8. Trajectories of the simulated robot and its odometry.

5.4. Discussion

The main drawback of the navigation method is that any inaccuracy of the mapping leads to inaccurate localization results. Because we do not use any compensation methods for the accumulation of errors in the odometry in the mapping phase, mapping of landmarks might be failed. For example, the same landmark is blurry when plotted on the map. Because of the blur, matching of the scan and landmarks sometimes fails during the localization phase. In this study, because we do not use long-range LiDAR for localization, this problem is small. However, it would be a major problem when long-range LiDAR is used. To overcome this problem, incremental scan matching-based mapping techniques or accurate dead reckoning are required.

Another drawback might arise from obstacle avoidance near the map changing points. As we noted in subsection 3.1, one problem associated with map changing is that the past mapped areas will be erased. Thus, robot pose estimation might fail if the heading direction of the robot near the map changing point, ${}^{i-1}\theta_t$, is significantly different from that of the changed pose, ${}^{i-1}\theta_{i-1N}$, shown in equation (1). To avoid this problem, the navigation strategy should be well constructed. For example, collision avoidance can be used near the map changing points instead of obstacle avoidance. However, this is not a perfect solution to this problem. This phenomenon is a limitation of the presented navigation method.

The navigation method is not suitable for exploratory missions, such as the mission from the second stage of TC, because several maps are used and specific points are not recorded on the maps. Namely, the robot cannot remember areas where the robot previously searched. We previously completed the mission using a similar navigation method; the navigation strategy is presented in [20]. The navigation method is suitable for simple navigation tasks in which the robot needs to automatically navigate between given start and goal points.

6. Conclusion

This paper has presented a teaching-playback navigation method that does not require a consistent map. Instead of using SLAM to build a consistent map, the navigation method uses several partial maps to represent the entire environment. Complex mapping processes can be omitted before starting autonomous navigation. In addition, the trajectory that the robot traveled in the mapping phase is directly used as a target path for autonomous navigation. As a result, we achieved teaching-playback navigation without any off-line processes.

We tested the navigation method, which includes other modules for autonomous navigation, e.g., localization and path following, using log data taken in the TC2017 environment. Through the experiments, we showed that the localizer robustly recognized the robot's pose in the dynamic environment, and that the navigation method achieved teaching-playback navigation in the simulated TC environment. The source code used in this paper is available on-line (`https://github.com/NaokiAkai/AutoNavi`).

Acknowledgment

This research was supported by the Center of Innovation Program (Nagoya-COI) funded by the Japan Science and Technology Agency and KAKENHI 40786092.

References:

[1] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles", *IEEE Micro*, vol. 35, no. 6, pp. 60–69, 2015.

[2] A. Watanabe, D. Endo, G. Yamauchi, and K. Nagatani, "Neonavigation meta-package: 2-D/3-DOF seamless global-local planner for ROS - Development and field test on the representative offshore oil plant," in *Proc. IEEE SSRR*, pp. 86–91, 2016.

[3] `https://github.com/ApolloAuto/apollo`, accessed at 2, Jul., 2018.

[4] http://wiki.ros.org/navigation accessed at 2, Jul., 2018.
[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE TRO*, vol. 23, no. 1, pp. 34–46, 2007.
[6] S. Kohlbrecher, O. Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE SSRR*, 2011.
[7] R. Mur-Artal, J. M. M. Montiel, and J.D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE TRO*, vol. 31, no. 5, pp. 1147–1163, 2015.
[8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE ICRA*, pp. 1271–1278, 2016.
[9] http://www.tsukubachallenge.jp/ accessed at 2, Jul., 2018.
[10] E. Takeuchi and T. Tsubouchi, "A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping," in *Proc. IEEE/RSJ IROS*, pp. 3068–3073, 2006.
[11] Tsukuba Challenge technical reports 2016 (in Japanese).
[12] Tsukuba Challenge technical reports 2017 (in Japanese).
[13] http://velodynelidar.com/vlp-16.html accessed at 2, Jul., 2018.
[14] Y. Hara, T. Tsubouchi, and A. Oshima, "Rao-Blackwellized Particle Filter SLAM considering previous data with probabilistically accumulated scan shapes," *Trans. of the JSME, Series C*, vol. 82, no. 834, pp. 15-00421–15-00421, 2016 (in Japanese).
[15] M. Magnusson, "The three-dimensional normal-distributions transform - an efficient representation for registration, surface analysis, and loop detection," *Ph.D. dissertation, Örebro University*, 2009.
[16] M. Tomono, "A scan matching method using Euclidean invariant signature for global localization and map building," in *Proc. IEEE ICRA*, pp. 866–871, 2004.
[17] K. Irie and M. Tomono, "A compact and portable implementation of graph-based SLAM," in *Proc. ROBOMECH*, 2P2–B01, 2017.
[18] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," *MIT Press*, 2005.
[19] J. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Proc. IEEE/RSJ IROS*, pp. 454–459, 2002.
[20] N. Akai, K. Inoue, and K. Ozaki, "Autonomous navigation based on magnetic and geometric landmarks on environmental structure in real world," *JRM*, vol. 26, no. 2, pp. 158–165, 2014.

Name:
Naoki Akai

Affiliation:
Institute of Innovation for Future Society (MIRAI), Nagoya University

Address:
464-8601, Furo, Chikusa, Nagoya, Japan

Brief Biographical History:
2013-2106 PhD student, Utsunomiya University
2016-present Designated Assistant Professor, Nagoya University

Main Works:
•"Gaussian processes for magnetic map-based localization in large-scale indoor environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4459–4464, 2015
•"Mobile robot localization considering class of sensor observations," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018
•"Simultaneous pose and reliability estimation using convolutional neural network and Rao-Blackwellized particle filter," *Advanced Robotics*, 2018

Membership in Academic Societies:
•The Institute of Electrical and Electronic Engineers (IEEE)
•Robotics Society of the Japan (RSJ)
•Japan Society of Mechanical Engineering (JSME)
•Society of Instrumentation and Control Engineering (SICE)

Name:
Luis Yoichi Morales

Affiliation:
Institute of Innovation for Future Society (MIRAI), Nagoya University

Address:
464-8601, Furo, Chikusa, Nagoya, Japan

Brief Biographical History:
2006-2009 PhD, Tsukuba University
2009-2016 Research Scientist, ATR
2016-present Designated Associate Professor, Nagoya University

Main Works:
• "Probabilistic 3D Mapping of Sound-Emitting Structures Based on Acoustic Ray Casting," *IEEE Transactions on Robotics*
• "Walking Together: Side by Side Walking Model for an Interacting Robot," *Journal of Human-Robot Interaction*
• "Autonomous Robot Navigation in Outdoor Cluttered Pedestrian Walkways", *Journal of Field Robotics*

Membership in Academic Societies:
•The Institute of Electrical and Electronic Engineers (IEEE)
•IEEE Robotics and Automation Society (RAS)
•Robotics Society of the Japan (RSJ)

Name:
Hiroshi Murase

Affiliation:
Graduate School of Information Science, Nagoya University

Address:
464-8601, Furo, Chikusa, Nagoya, Japan

Brief Biographical History:
1980 M.E., Electrical Engineering, Nagoya University
1987 PhD, Electrical Engineering, Nagoya University
1980-2003 Research Scientist, NTT Research Laboratory
2003-present Professor, Information Science, Nagoya University

Main Works:
• "Single camera vehicle localization using feature scale tracklets," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E100-A, No. 2, pp. 702-713, 2017
• "Using super-pixels and human probability map for automatic human subject segmentation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E99-A, No. 5, pp. 943-953, 2016
• "A quick search method for audio and video signals based on histogram pruning," *IEEE Transaction on Multimedia*, vol. 5, no. 3, pp. 348–357, 2003
• "Subspace methods for robot vision," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 750–758, 1996
• "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995

Membership in Academic Societies:
•Fellow, The Institute of Electrical and Electronic Engineers (IEEE)
•Fellow, The Institute of Electronics, Informtion and Communication Engineers (IEICE)
•Fellow, Information Processing Society of Japan (IPSJ)

View publication stats