https://doi.org/10.20965/jrm.2016.p0479

Paper:

Monocular Vision-Based Localization Using ORB-SLAM with LIDAR-Aided Mapping in Real-World Robot Challenge

Adi Sujiwo, Tomohito Ando, Eijiro Takeuchi, Yoshiki Ninomiya, and Masato Edahiro

Nagoya University
609 National Innovation Complex (NIC), Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: {sujiwo, eda}@ertl.jp, {andou.tomohito@e.mbox, takeuchi@coi, ninomiya@coi}.nagoya-u.ac.jp
[Received February 23, 2016; accepted June 2, 2016]

For the 2015 Tsukuba Challenge, we realized an implementation of vision-based localization based on ORB-SLAM. Our method combined mapping based on ORB-SLAM and Velodyne LIDAR SLAM, and utilized these maps in a localization process using only a monocular camera. We also apply sensor fusion method of odometer and ORB-SLAM from all maps. The combined method delivered better accuracy than the original ORB-SLAM, which suffered from scale ambiguities and map distance distortion. This paper reports on our experience when using ORB-SLAM for visual localization, and describes the difficulties encountered.

Keywords: visual localization, autonomous vehicle, field robotics, Tsukuba Challenge

1. Introduction

Recently, many manufacturers and institutions have been developing autonomous driving systems and advanced driving assistance. Google has already tested autonomous driving in urban environment over distances of 1.6 million km. Their current designs use high precision and accurate 3D Light Detection and Ranging (LIDAR) for localization and recognition. Such sensors are still expensive relative to other vehicle sensors such as camera and millimeter wave range sensor. However, consumer-level vehicles require low-cost but reliable localization and recognition sensors. On the other hand, the creation of high-precision digital environment maps is an active research areas within the field of autonomous driving systems. An example is the Mobile Mapping Systems (MMS), which is capable of creating accurate 3D point cloud maps as a vehicle travels along urban road. Such 3D maps constitute basic technology for recent autonomous vehicles as they store important information for autonomous functions, such as localization. Therefore, the main idea here is to separate mapping and localization as different processes. By employing high-precision but expensive devices we could create good quality maps, which can be outsourced to competent companies. Next, these maps can be employed inside consumer vehicles using low-cost sensors to deliver same level of accuracy against more expensive sensors.

The Real-World Robot Challenge (RWRC) is a real-world autonomous navigation challenge that is held in City of Tsukuba, Japan. The robots are required to autonomously navigate over a 1 km route. One important requirement involved in the realization of autonomous navigation is localization. The robots are required to maintain their position accuracy over the course despite changes in the environment such as dynamic obstacles, differences in the illumination, changes in season and other factors. Most teams in the Tsukuba Challenge use a sensor fusion approach using LIDAR, gyroscopes, and an odometer. Such sensor fusion techniques can compensate for weaknesses in the characteristics of individual sensors. On the other hand, vision-based approach is not actively used in the RWRC. In short, objective of this research was to identify and solve problems with localization based on a monocular camera in a real-world setting. Within this paper, the terms "RWRC" and "Tsukuba Challenge" will be used interchangeably.

Despite inability of our team to complete the course, we gathered much valuable experience as a result of attempting the Tsukuba Challenge. Most importantly, we were able to implement monocular vision-based localization based on ORB-SLAM [1]. In the process, we collected datasets for evaluating vision-based localization methods in a real world environment.

This paper proposes a positioning method based on ORB-SLAM using monocular camera with LIDAR-aided mapping. The ORB-SLAM is one of most recent monocular vision-based SLAM method with open-source implementation. This method estimates the position and map from an image sequence in real-time. Originally, ORB-SLAM was designed to solve SLAM problem. However, the required level of performance for localization problem is different from SLAM problem; therefore, the method incurs several problems when it is adopted for localization problems such as robustness and map consistency. Our proposed method has two key points:

1. The estimation of metric position in localization by using ORB-SLAM with LIDAR-aided mapping.
2. The solution of robustness problem using sensor fusion between multiple maps and odometry data.

In general, this paper discusses: 1) benchmark tests related to ORB-SLAM conducted in the Tsukuba Challenge environment; 2) description of ORB-SLAM with LIDAR-aided mapping to solve problems of consistency between multiple maps; 3) experimental results and evaluation in Tsukuba Challenge environment. Finally, this paper illustrates the capability of sensor fusion method of vision-based localization method with odometer to provide continuous localization over the course.

2. Related Works

2.1. Monocular Vision-Based SLAM

Most monocular vision-based SLAM methods rely on a 3D reconstruction based on multiple views of a scene [2], which in turn is based on structure from motion (SfM). The SfM technique refers to the process of estimating 3D structures from 2D image sequences while inferring the motion of the camera. As stated in [3] and [4], all monocular structures from motion methods inherit common scale ambiguities, i.e., the recovered 3D structures and camera motion are defined up to an unknown scale factor which cannot be determined from image streams alone. This is because, if the scene and camera are scaled together, this change will be indistinguishable in the captured images. This fact results in difficulties in providing true position of the camera, which is very important for autonomous vehicle navigation and control.

One solution to recover true position (in metric sense) of the camera is by associating keyframes with external references, such as GPS coordinates [5]. Most current researches that we know of do not take this method; instead, alternative methods of mapping such as stereo camera are employed. It is also not clear how to recover the position of camera during localization phase using those references in keyframes.

In addition to ORB-SLAM, there have been a number of monocular variations of SLAM with complete public implementation. Of particular interests are PTAM by [6], and LSD-SLAM by [7]. ORB-SLAM itself was described in [1]. Of particular note, the ORB-SLAM uses ORB (Oriented FAST, Rotated BRIEF) as its main feature detector [8], and bag-of-words method as its place recognition [9]. Contrary to previous methods that examined entire scene and generated dense maps, ORB-SLAM extracts only feature points and generates relatively sparse maps. This allows ORB-SLAM to produce potentially smaller maps and thus reduce the processing time.

2.2. LIDAR-Based SLAM

3D LIDAR-based SLAM is explained in [10]. This LIDAR-based SLAM method is popular for autonomous vehicle applications, and is capable of providing accurate maps and localization. The author, however, does not emphasize any special methods for the registration of scans captured by the LIDAR devices. Scan registration is important for computing rigid transformation of these scans (thus computing the mapping and localization). Of particular note, the LIDAR-based SLAM is used in Autoware [11] as real-world application of autonomous vehicle platform.

One commonly used method for 3D scan registration is the normal distribution transforms (NDT) as described in [12] and implemented in Point Cloud Library [13]. As was shown in [14], it is possible to build a map and perform localization in real-time using NDT.

2.3. Localization for Tsukuba Challenge

For most of the course of the Tsukuba Challenge, almost all teams use LIDAR-based localization method [15]. These methods use sensor fusion approach with LIDAR and odometer. Essential to this approach is the use of a good dead reckoning method such as a calibrated gyroscope.

There have been some attempts to apply monocular vision-based localization to RWRC. One of the those efforts is [16]. This method is basically a type of topological localization by following an image sequence and estimating pose by using feature points. Comparing with the above methods, our method sets out to attain metrically correct positioning, similar to LIDAR-based navigation but using monocular camera. This method is frequently applied to other metric localization methods such as LIDAR, GNSS, and dead reckoning methods.

3. Monocular Vision-Based Localization

To repeatedly attain localization, our implementation of ORB-SLAM consists of two parts: mapping and localization-only. The mapping process runs similar to the original implementation, with addition of map storage at the end of the mapping run. Meanwhile, the localization process starts with map restoration using previous data from mapping process. Next, localization proceeds in much the same way as in the mapping stage. However, map modification is disabled in the relocalization process.

3.1. Description of ORB-SLAM

The ORB-SLAM main process creates an environmental map which consists of keyframes and map points. Each keyframe stores its position in ORB-SLAM coordinates and a list of 2D feature points. The entire ORB-SLAM process consists of three parallel threads: tracking, local mapping and loop closing. The relationship between these processes is illustrated in **Fig. 1**.

3.1.1. Feature Detection

The first step in all 3D reconstruction is to identify feature points in each frame. ORB-SLAM uses ORB, described in [8]. This detector offers advantages such as faster computation and lower storage requirement (each descriptor needs 32 bytes), in addition to resistance to rotation and noise.

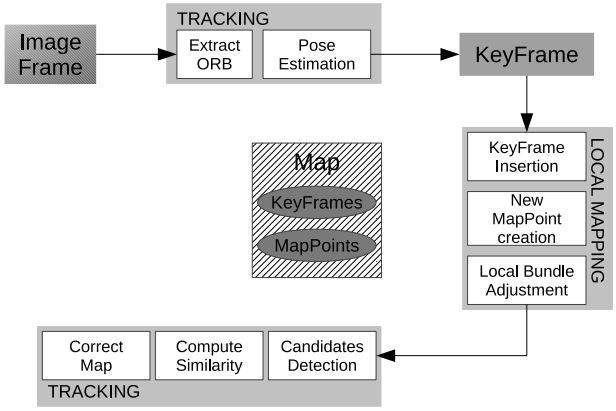

Fig. 1. ORB-SLAM system overview [1].

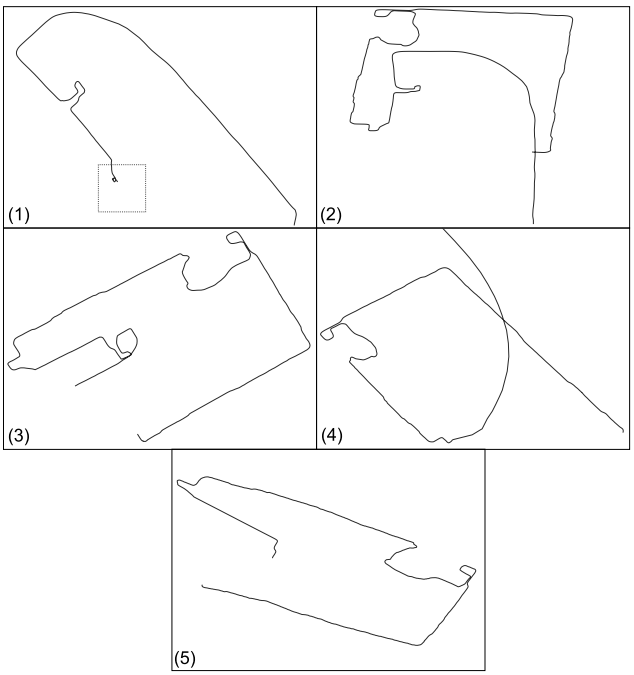

Fig. 2. ORB-SLAM map trajectories from different times are plotted in (1) and (2). (3) is ground truth. (4) is zoomed part of gray rectangle in (1). In (5), corrected ORB-SLAM map from distance scale of NDT.

3.1.2. Map Initialization

The goal of the map initialization is to compute the relative pose between two frames to triangulate an initial set of map points [1], which are then used for keyframe tracking. ORB-SLAM uses a combination of homography and fundamental matrices inside a RANSAC scheme to build motion and structure recovery as described in [17]. When this stage is successful, the system will have an initial set of keyframes and map points with which tracking may proceed. However, tracking may fail shortly after initial map is built; if this occurs the initial map is reset and started over.

3.1.3. Tracking and Local Mapping

The tracking thread is responsible for providing localization and map building. After ORB corners are detected, the tracking thread develops a map incrementally over the recovered 3D map points, while computing camera poses.[1] To speed up this process, the tracking operates in a smaller subset of the overall map, called the local map, that covers current visible keyframes and some connected ones. The tracking thread also performs map "clean-up," which involves culling bad map points and keyframes.

To perform tracking, there are three modes that may be used. First is relocalization by searching all keyframes by bag-of-words; this is the slowest but indispensable when recovering from lost tracking. The second choice involves tracking the local map, as described above. Alternatively, the third choice involves tracking using constant velocity model. This choice is fastest and may be the most frequently used mode. However, it may be inaccurate.

Occasionally, the tracking thread will perform local bundle adjustment (BA) to optimize the current local map. This action moves the keyframes in the local map, and potentially marks some keyframes as outliers for removal.

3.1.4. Loop Closing

Loop-closure detection is crucial for enhancing the accuracy of SLAM algorithms, both topological and metrical. This problem consists of detecting when the robot has returned to a former location after having discovered new terrain. Such detection makes it possible to increase the precision of the actual pose estimation.

Essentially, loop closing in ORB-SLAM uses image-to-map approach [18]. First, it takes the most recently processed keyframe and searches for a loop candidate keyframe in the local map using the bag-of-words method [9]. Next, the similarity transformation is computed. Loop correction is performed by inserting new edges into the covisibility graph and fixing connectivity between loop candidate and surrounding keyframes. Next, ORB-SLAM performs pose graph optimization, whereby loop closing errors are distributed by moving the candidate and its connected keyframes.

3.2. Problems with ORB-SLAM

3.2.1. Scale Ambiguity

ORB-SLAM outputs localization results based on maps in its own coordinate system (which is not metrically correct), that are not free from distortion due to scale ambiguities. This problem is inherent to almost all vision-based localization methods or even all 3D reconstruction method [19]. In some cases, result maps may exhibit heavy deformation, as illustrated in **Fig. 2**.

From **Fig. 2**, it is clear that ORB-SLAM generates very deformed trajectory shapes compared to ground truth (trajectory generated by NDT scan matching at corresponding time). By zooming-in parts in square in **(1)**, it is revealed that most later keyframes clump in a small area of

1. Pose is defined as combination of position and orientation.

the ORB-SLAM map. By applying point distances acquired from ground truth, we get a trajectory that is closer to ground truth **(5)** but still has wrong shape.

3.2.2. Lack of Support for Lifelong Mapping

The original design of ORB-SLAM involved a single run for both localization and mapping, so there are not features for storing in-memory map to disk. However, it is desirable for distinct mapping and localization processes to be done multiple times with the same path, so map saving and restoration is essential. This feature is also useful for improving the map robustness when faced with changing condition [20]. In practical application, this will enable an autonomous vehicle to localize positions despite changes in weather, time of day, and other conditions.

3.2.3. Visual Disturbances

Any disturbances in the camera vision while tracking feature points may lead to the failure of ORB-SLAM tracking process. These disturbances include vision occlusion on the part of camera and sudden rotation of the robot. The default behaviour of ORB-SLAM is to reacquire the tracking after any such loss. This is achieved by performing relocalization from the last keyframe (i.e., guess pose from last known position). However, this prevents the system from reacquiring position, especially when the robot never reverses motion, and contrary to the description of the behaviour in the original paper of ORB-SLAM. Our solution is to force relocalization by searching all the keyframes in the database.

Another forms of visual disturbances are lens flares and smears, which happen when the camera is faced to the sun. The effects range from lost tracking to straying of the positions, that may negatively affect the usability of the ORB-SLAM.

4. ORB-SLAM with LIDAR-Aided Mapping

In this section we explain how to realize metrically correct monocular visual localization for solving main problems of ORB-SLAM explained in previous subsection. First, LIDAR-aided mapping is employed to solve scale problem of keyframe distances and provide metrically correct positioning. Next, we describe map storage and restoration process to enable separate mapping and localization process. Lastly, multiple observations from distinct ORB-SLAM maps may be employed simultaneously with odometry data to derive accurate position and orientation of the robot.

The main output of ORB-SLAM mapping is a set of keyframes. As explained above, our main goal is to take this map and compute the localization during robot runs as guidance for navigation system, be it a robot or an autonomous car. Therefore, it is necessary to get localization results that are metrically accurate. However as illustrated in **Fig. 2**, it would be very difficult to accurately obtain the position using deformed maps from ORB-SLAM.

Algorithm 1 Position correction from ORB-SLAM to global coordinate.

1: Take the original ORB-SLAM position obtained from localization as P_o.
2: Search the nearest keyframe in the map from P_o in the octree as P_k. From here, we take the corresponding external reference position P_n.
3: Find the previous offset keyframe $P_{k'}$ and its corresponding external reference position $P_{n'}$.
4: Compute the scale correction factor. This factor is a ratio of magnitude of translation between external reference $P_{n'}$ to P_n and translation between keyframe $P_{k'}$ to P_k.
$$s = \frac{\|\mathbf{t}_{n'} - \mathbf{t}_n\|}{\|\mathbf{t}_{k'} - \mathbf{t}_k\|}$$
5: Apply the distance scale:
$$P_r = P_k^{-1} P_o$$
$$P_{r'} = (s\mathbf{t}_r, \mathbf{q}_r)$$
$$P_c = P_n P_{r'}$$
6: **return** P_c

To formalize the map and localization process, we use the following notation. An ORB-SLAM map is a set of poses[2]: $M = \{P_i | 0 \leq i < N-1\}$, where N is the number of keyframes until the mapping process is stopped. Each pose P consists of translation and rotation (in quaternion) in 3D, such that P is a vector of seven elements: $P = (\mathbf{t}, \mathbf{q}) = (x, y, z, q_x, q_y, q_z, q_w)$.

4.1. Metric Transformation from ORB-SLAM

As stated in [5], the scale problem can be solved in mapping phase by associating each keyframe to an external reference with true position (in metric sense). In a longer run, this association must be done correctly so that error accumulation in the scale correction is eliminated. Therefore, the external reference must have a high level of accuracy. In our case, we chose LIDAR-based localization as the reference due to its immediate availability. Other positioning methods may also be used, such as GPS or odometry, as long as their error corrections are provided [21].

In the localization process, the system depends solely on the ORB-SLAM method. Therefore, external methods such as LIDAR-based localization are not required. To compute the metric value of a pose, the system modifies the position value to correct distance deformation according to the following formula. All poses use poses' matrix representations. This transformation is described in **Algorithm 1** and illustrated in **Fig. 3**.

4.2. External Mapping Reference Using NDT Scan Matching

This research used the 3D Normal Distribution Transform (NDT) scan-matching method with 3D LIDAR to

2. There are other important data, but not our concern yet.

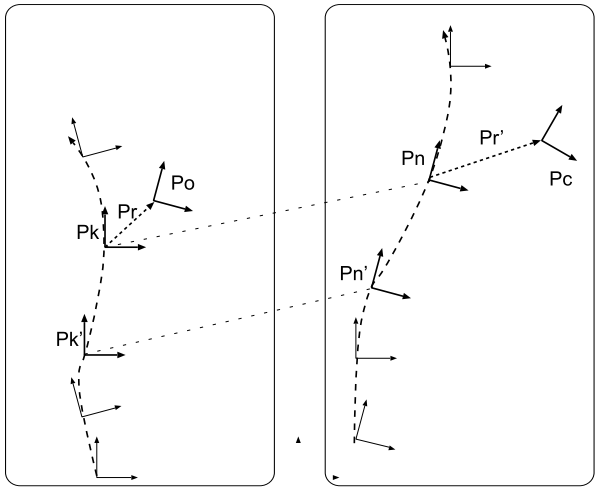

Fig. 3. Metric transformation.

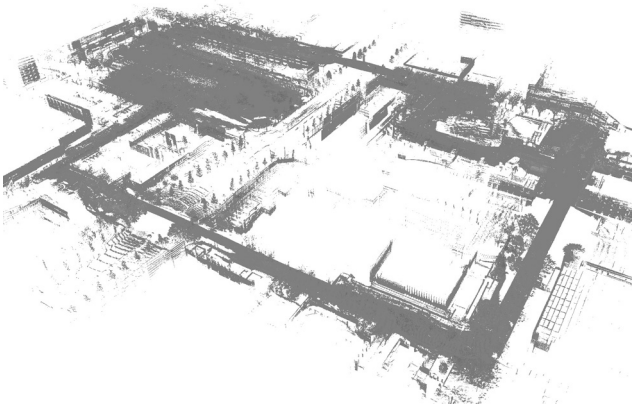

Fig. 4. 3D view of Tsukuba Challenge map generated by NDT scan matching from Velodyne scans.

obtain accurate positions [14] as keyframes' external references. **Fig. 4** is a visualization of 3D map of the Tsukuba Challenge track. This map was built by applying the 3D NDT scan-matching method using the Velodyne HDL32 LIDAR. **Fig. 5** shows the localization results of each runs using 3D NDT scan-matching with the Velodyne HDL32 and the 3D map. In the figures, the robot positions were estimated for the entire route on the map.

4.3. Map Storage and Restoration

Map storage consists of three main parts: keyframes, map points, and keyframe relationships. Each keyframe stores the camera pose P_k in ORB-SLAM coordinates, the camera intrinsic parameters, all of the ORB feature points recorded at keyframe creation, and external reference pose P_n in its own coordinates, recorded at keyframe creation.

During map restoration, the system reconstructs the following data structures:

1. List of keyframes and their relationship.

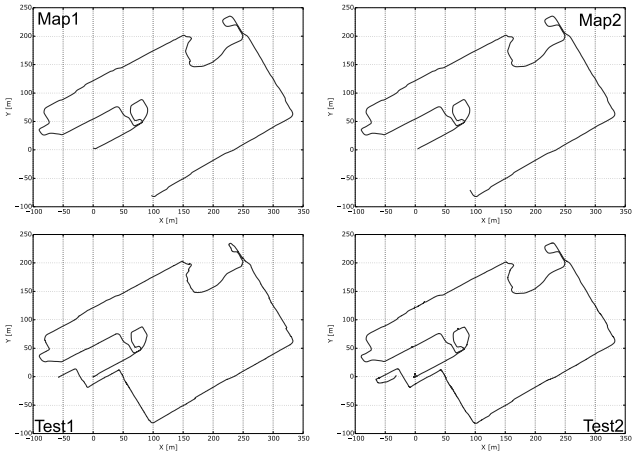

Fig. 5. Ground truth from NDT scan matching of 4 runs. This map is metrically correct. The track covered by each run is slightly different.

2. Map point list.

3. Octree of keyframe position in ORB-SLAM coordinates. This tree will be used for fast searching of the keyframes during localization using augmented positioning.

By default, ORB-SLAM will try to find the position against the last keyframe whenever it loses tracking. However, for situations after map restoration, the last keyframe will be unknown. Instead, we modify ORB-SLAM to force it to search the most appropriate keyframe using the bag-of-word method. Keyframe search is also applied when the system loses tracking; this is done to ensure that the system always gets the keyframe as the basis for relocalization. The drawback is that keyframe search using bag-of-words method is slower than tracking using the last keyframe.

4.4. Using Multiple Maps

During our experiments, we found that maps of the same location but created at different times will deliver varying results. Therefore, it is logical to combine the results from two maps in order to: 1) alternately provide localization whenever one of the maps fails; 2) reduce errors from all the maps. In this regard, any method for sensor fusion may be used. It must be stressed that, after correction, all of the maps will provide consistent results that are metrically correct and in the same coordinate system.

In current version, the ORB-SLAM does not allow using multiple maps. However, we can run multiple process of ORB-SLAM with the same input data; each one utilized different maps built from different times. Hence, we could produce multiple results simultaneously from single input camera. Observations from these distinct maps may be combined together with odometry as discussed in next subsection.

Algorithm 2 Particle filter localization [22].

Require: X_{t-1}, U_t, Z_t
1: $\bar{X}_t = X_t = \emptyset$
2: **for** $n = 1$ **to** M **do**
3: $x_t^{[n]} =$ **motion_model**$(u_t, x_{t-1}^{[n]})$
4: $w_t^{[n]} =$ **measurement_model**$(z_t, x_t^{[n]})$
5: $\bar{X}_t = \bar{X}_t + \langle x_t^{[n]}, w_t^{[n]} \rangle$
6: **end for**
7: **for** $n = 1$ to M **do**
8: draw x_t with probability $p \propto w_t$
9: add $x_t^{[i]}$ to X_t
10: **end for**
11: **return** X_i

4.5. Augmenting ORB-SLAM Localization with Odometry

Observing that both maps were unable to provide sufficient level of coverage, it is reasonable to utilize simultaneously both maps and combine measurements from odometer to provide localization for the robot. This is enabled due to the fact that modified ORB-SLAM outputs positions in metric coordinate system; thus highlights an advantage of our method.

As an approach for sensor fusion, we provide a framework derived from particle filter as described by [22]. This formula basically tries to calculate position and orientation from velocity and rotation speed measured by odometer, while correcting these values as ORB-SLAM localization supply position and orientation updates.

To simplify formulation, we assume that the robot moved in 2D plane; robot state at time t is represented by position in 2D plane and its orientation as $\mathbf{x_t} = (x_t, y_t, \theta_t)$. Control data from odometer arrive as linear velocity and rotation speed and represented respectively, as $\mathbf{u_t} = (v_t, \omega_t)$. The particle filter takes a sample of M number of "particles"; each particle represents a possible state of the robot. As the motion proceeds, all particles are updated by control variables $\mathbf{u_t}$ and ORB-SLAM measurements $\mathbf{z}$ from all maps where available. The particle filter then selects particles proportional to their fitness against $\mathbf{z}$ as weight w. The complete particle filter is described in **Algorithm 2**, complemented with its motion model and measurement model in **Algorithms 3** and **4**.

In order to account for errors in v and ω, we introduce noises to v and ω. The noises are assumed to be Gaussian with standard deviation α_1 and α_2, that are device-specific and must be determined by experiment.

In measurement model, each particle's weight is determined from its distances to all ORB-SLAM measurements. Here, each ORB-SLAM measurement is assumed to be independent and may contain noises (for example, see **Fig. 13** in Subsection 5.4). Therefore we select the nearest measurement to the particular particle, resulting in largest weight from all measurement as described in **Algorithm 4**. This measurement model is easily expandable to include more than two ORB-SLAM results.

Algorithm 3 Computing poses $X_t = (x', y', \theta')$ from a pose $X_{t-1} = (x, y, \theta)$ and control $U_t = (v, \omega)$.

motion_model (U_t, X_{t-1}) :
$\hat{v} = v + \text{rand}(\alpha_1)$
$\hat{\omega} = \omega + \text{rand}(\alpha_2)$
$x' = x + \hat{v}\cos(\theta)\Delta t$
$y' = y + \hat{v}\sin(\theta)\Delta t$
$\theta' = \theta + \hat{\omega}\Delta t$
return $X_t = (x', y', \theta')$

Algorithm 4 Particle weighting w of state $X_t = (x', y', \theta')$ against ORB measurement $Z_1 = (x_1, y_1, \theta_1)$ and $Z_2 = (x_2, y_2, \theta_2)$. Here, Σ is covariance matrix which represents error measurements of ORB-SLAM in lateral, longitudinal and yaw.

measurement_model (X_t, Z_1, Z_2) :
$\Sigma = \begin{bmatrix} \sigma_x & & 0 \\ & \sigma_y & \\ 0 & & \vartheta \end{bmatrix}$
$w_1 = \exp\left\{-\frac{1}{2}(X_t - Z_1)^T \Sigma^{-1}(X_t - Z_1)\right\}$
$w_2 = \exp\left\{-\frac{1}{2}(X_t - Z_2)^T \Sigma^{-1}(X_t - Z_2)\right\}$
return $w = \max(w_1, w_2)$

5. Evaluation in Tsukuba Challenge Environment

We performed four runs whereby the robot traversed the trajectory mandated by the Tsukuba Challenge committee. In each run, we recorded camera images and performed localization using Velodyne LIDAR. From these runs, we created two maps for localization process. The LIDAR-based localization results would be used as ground truth for comparison. To reduce computation, camera resolution was reduced to 800×600 before processing.

Two runs were chosen for mapping by considering that those runs were the longest runs without any vision occlusion. The weather conditions and lighting condition varied slightly for all the runs. However, there were a few instances of heavy flares when the camera faced to the sun, with the tracks were covered by falling leaves. The map and test run conditions are listed in **Table 1**. Due to concern of equipment damage from rain, experiment on final day (8th) did not proceeded.

The map trajectories that we use, created by ORB-SLAM, are shown in **Figs. 2(1)** and **(2)**. Note that those maps are heavily deformed compared to ground truth shown in **Fig. 5**. In the map trajectories, recordings were stopped before robot could finish the runs; however the robot successfully proceeded to finish point in every testing run.

5.1. Experimental Settings

Our robot was derived from a Segway RMP-200 platform, using a PointGrey Grasshopper3 camera and Velodyne HDL-32 LIDAR. The robot ran through the man-

Table 1. Time and condition for mapping and testing runs.

Run	Date & Time (November 2015)	Weather condition	Lighting contrast	Human presence
Map 1	6th, 13:44	Clear, few low clouds	High	Low
Map 2	7th, 11:30	Cloudy	Medium	Low
Test 1	3rd, 14:55	Clear	High	High
Test 2	7th, 14:20	Cloudy	Low	Low

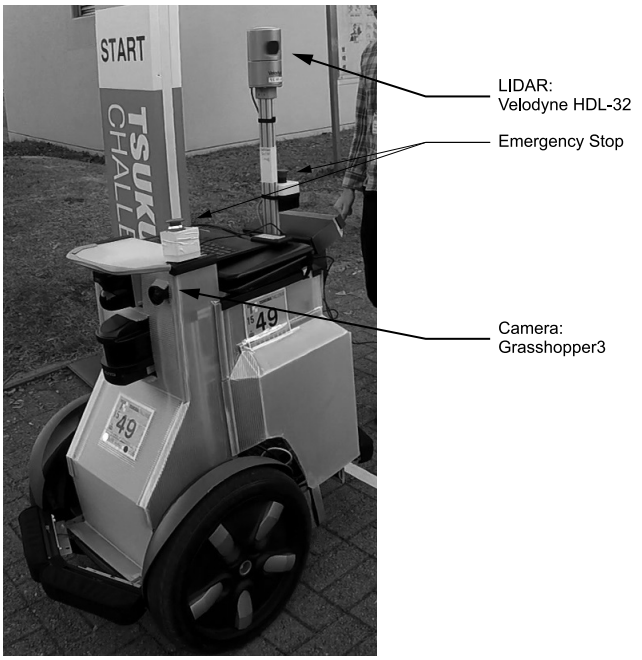

Fig. 6. Robot used for evaluation.

dated Tsukuba Challenge course at a speed less than 1 m/s. In the course of the run, robot would often encounter dynamic obstacles such as human or bicycle, which necessitated action by the operator to either stop or maneuver the robot. Our robot setup is shown in **Fig. 6**.

The track to be covered in the Tsukuba Challenge was very different from that used for the original ORB-SLAM paper evaluation, which primarily used New College dataset [23]. To simplify discussion, we roughly divided the track into five major areas; each had distinct visual differences and its own challenges. These areas are shown in **Fig. 7**, and can be described as follows.

1. Area 1 was public park area, with many trees as main features and occasional building background (**Fig. 8**).
2. Area 2 was checking stop in front of a large hall building. When passed in afternoon, this area may feature high contrast due to setting sun; most lens flares were encountered here.

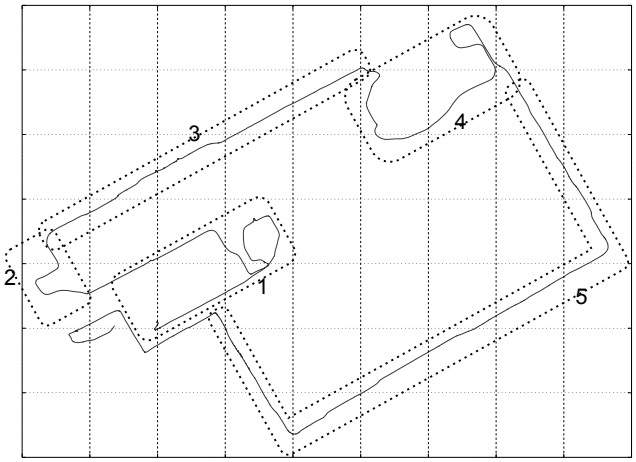

Fig. 7. Breakdown of Tsukuba Challenge track by visual features.

3. Area 3 was a pedestrian footpath covered by paved blocks and surrounded by trees and fallen leaves on the ground. There might be some encounters with curious pedestrian that approached the robot; these people were registered on the map (**Fig. 9**).
4. Area 4 was an outdoor scene with many buildings as background. This area featured quite strong contrast, as shown in **Fig. 10**. Situation like this can confuse automatic exposure system of cameras, and makes it difficult to detect feature points.
5. Area 5 was mostly the same as area 3, but encounters with pedestrian or bicycles were rare.

5.2. Map Saving and Restoration

The map data structures of ORB-SLAM can now be saved and restored at any time. In our experience, map saving and restoration do not affect ORB-SLAM performance. In fact, the system gains useful capability, i.e., map building can now be done incrementally using the same location but different times. This is useful for example, for building lifelong map in different situations such as in varying weather and during the day/night. An example of the relocalization after map restoration is illustrated in **Fig. 11**. Example of incremental map building is shown in **Fig. 12**.

5.3. First Position Fix

To get an initial position fix for relocalization, ORB-SLAM performs a keyframe search based on the appearances of feature points. This search may returns more than one candidate, which will be evaluated according to the reprojection error. Only one candidate is accepted, and it must have at least 15 map points that match the feature points in the current frame. In the evaluation run, however, the system was slow to obtain the initial fix due to insufficient map point matches. One possible enhancement that would enable quicker initial fix is to increase

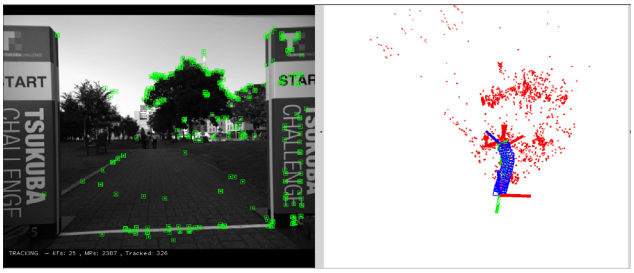

Fig. 8. Starting point.

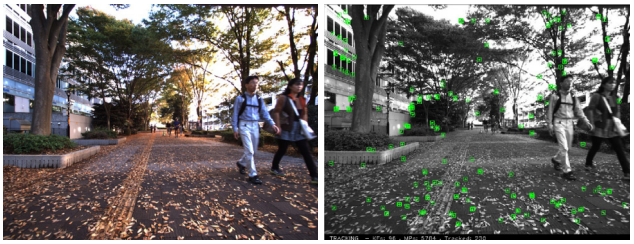

Fig. 9. Typical situation in Tsukuba Challenge: pedestrian tracks covered with fallen leaves.

Fig. 10. Lighting situation featured many areas with strong contrasts due to shadow cast.

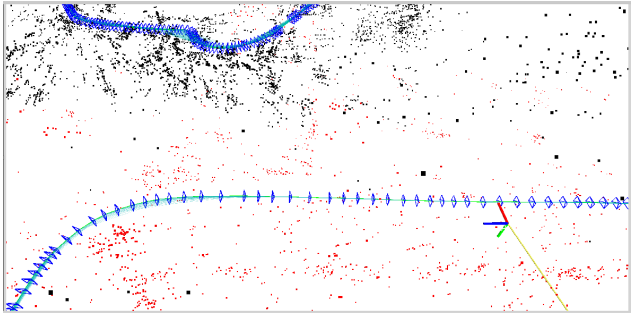

Fig. 11. Robot traversing previously created map.

the number of feature points from the ORB computation. However, this approach greatly slows the search process, and does not always correlate to a quicker fix.

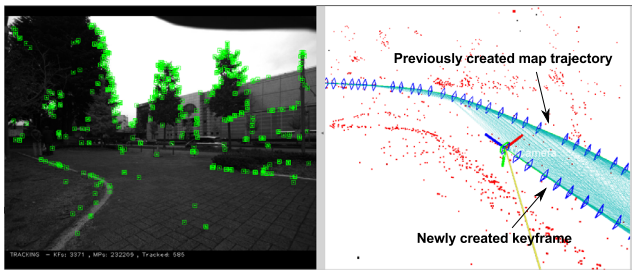

Fig. 12. ORB-SLAM created a new map based on old map.

ORB-SLAM map initialization is another problem. During the experiment, we found that initialization will succeed (without getting false initialization) whenever the robot is moved, both in rotation and translation. In our experiences, false map initialization and slow position fix can be solved by increasing number of extracted ORB points (by default this is 1000) to 2500. The drawback is higher computation times per frame. However, other benefits from increasing this number are better resistance to visual disturbances.

5.4. Relocalization and Tracking

During our experiment, we found that ORB-SLAM is resistant to occasional and partial vision occlusion. Partial occlusion includes lens flares and humans moving in front of the background images. Total occlusion however, may cause the system to fail in tracking, which is difficult to recover. This lost tracking explains existence of blank areas in **Figs. 13** and **14** (part of trajectory that has no bold parts).

Common situation and tracking of ORB-SLAM are depicted in **Fig. 15**. The figure illustrates a frame, taken in the Oshimizu park area, with a background of buildings in the distance and some trees in the foreground. There were also some people in the scene. Most of the ORB points (and map points, shown in green dots) fell in the trees and ground, but very few of those were in background.

Figure 11 depicts visualization of the robot when it travelled along previously created map. The map was created on a different day. Note the slightly out-of-track position of the robot. In this situation, tracking was maintained.

Figure 16 depicts a situation in which the robot performed a violent rotation such that it was on the verge of losing tracking. Note the absence of ORB feature points in the right portion of image frame. On the right, the axis shows that the robot was on the right track, but robot was oriented towards a place with very few map points. The blue axis represents the front.

In both test runs, each map delivered a different level of performance regarding the track coverage. In **Fig. 13** for test run 1, both maps are essentially complementary to cover tracking for the whole track. However, area 1 is particularly must be concerned where ORB-SLAM loses the tracking even when using both maps. This area is deemed critical because the robot had to perform many turns suc-

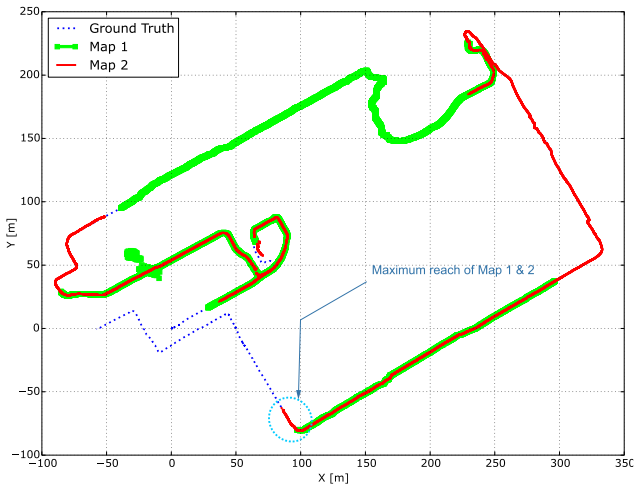

Fig. 13. Coverage for test run 1.

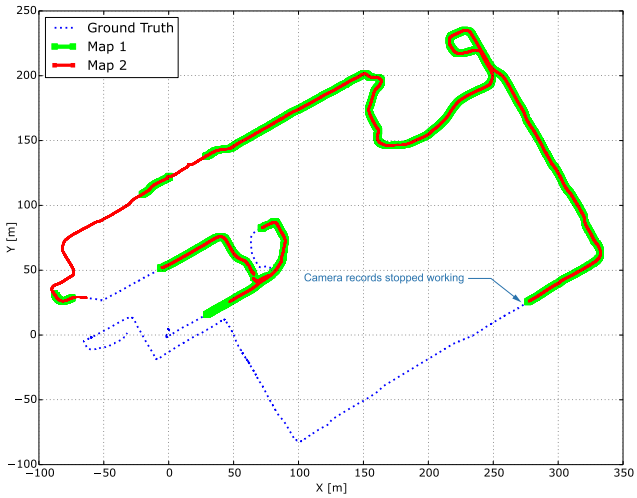

Fig. 14. Coverage for test run 2.

cessively. Also notable are some stray points in trajectory from map 1; these points were traced to instances of lens flares due to camera facing south west while the sun was low. In test run 2 as depicted in **Fig. 14**, both maps also provided complementary coverage. There were also significant time delay from the start of the motion to the initial position fix when using both maps.

In both test runs localization system was unable to cover the whole ground truth; the reasons were technical unrelated to ORB-SLAM capability. At all mapping runs and test runs except test run 1, camera recording stops prematurely before reaching finish line. Thus, maps 1 and 2 were unable to cover the whole Tsukuba Challenge track (ORB-SLAM is unable to localize too far from last keyframe). Also, camera stopped working too early in test run 2, rendering ORB-SLAM stopped working. Percentage of track covered by all maps are listed in **Table 2**.

In general, there are two main reasons for the robot losing tracking: visual disturbances (including, but not limited to, lens smears and complete vision occlusion), and rapid rotation in part of the robot due to the appearance of dynamic obstacles. An example of a visual disturbances (in form of lens smear) causing a loss of tracking and high number of errors in track run 1 using map 1 is shown at **Fig. 17**, where spurious points from localization are present.

Fig. 15. ORB-SLAM performed tracking.

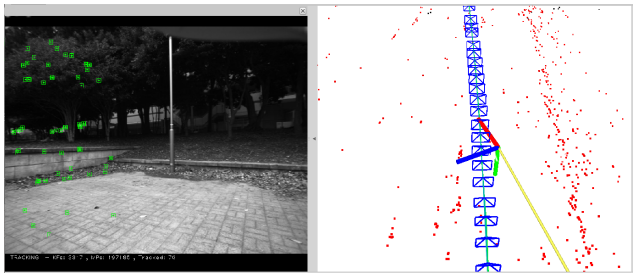

Fig. 16. ORB-SLAM about to lose tracking.

Table 2. Summary of ORB-SLAM performance compared to ground truth.

Map	Errors [m]			% Coverage
	Average	Std. Dev	Maximum	
Test Run 1				
Map 1	0.38	1.60	26.41	68.3
Map 2	0.19	0.53	5.62	70.1
Joint				95.1
Test Run 2				
Map 1	0.08	0.11	1.21	68.4
Map 2	0.06	0.09	1.67	80.9
Joint				82.4

In Tsukuba Challenge, average computation times of each frame were around 58 ms. This number equals to about 19 Hz, which is lower than original ORB-SLAM that delivers around 25–30 Hz.

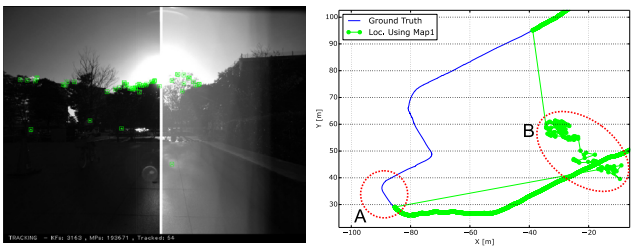

Fig. 17. At right, a part of robot trajectory is shown. Circle A shows location where disturbance took place; B shows localization results at that time. At left, camera image at corresponding time.

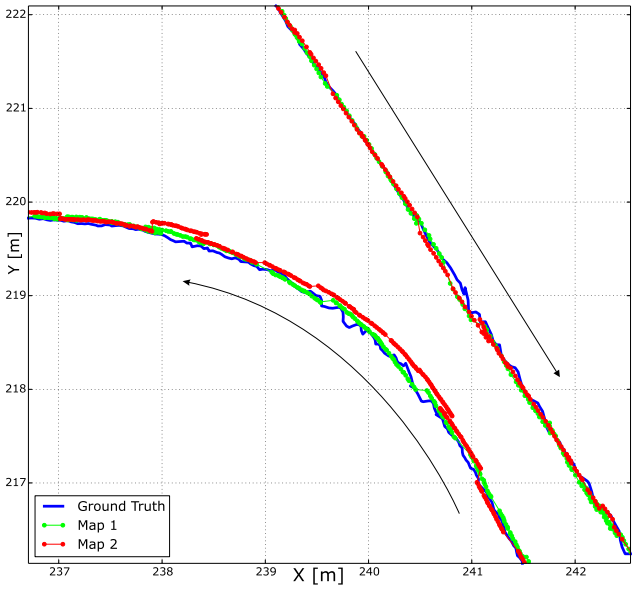

Fig. 18. Visual comparison of modified ORB-SLAM trajectory and ground truth.

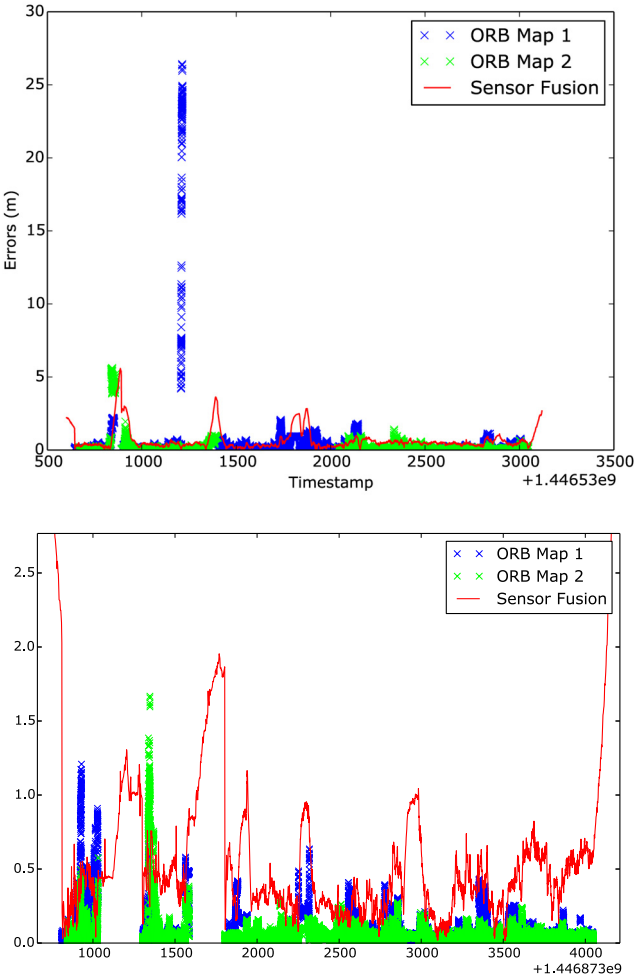

Fig. 19. Error graphs from test run 1 (top) and test run 2 (bottom).

5.5. Accuracy of Corrected Localization

Figure 18 depicts a situation in which the robot enters and exits from a turnabout. In the turning, the modified ORB-SLAM exhibits large deviations compared to ground truth, while straight path exhibits less deviation. It is also clear that each map produces different results, despite following exactly the same path and time.

Table 2 summarizes the performance of ORB-SLAM when covering the Tsukuba Challenge track. On average, the accuracy of ORB-SLAM is quite good when considering that errors in the order of 25 cm are within the range of robot's camera tracking. However, it is of some concern when this errors greatly increases, especially during test run 1. These errors may, however be regarded as deviation from the norms, as suggested in **Fig. 17**. In particular, this problem may be solved by using a more robust camera.

5.6. Multiple Maps and Odometry

Despite attaining a good level accuracy across the test run in Tsukuba Challenge, ORB-SLAM was unable to attain localization for the entire track. By recapitulating the performance summary in **Table 2**, and **Fig. 19**, it is reasonable to say that we can cover larger part of the track using joint map. This subsection discusses results of sensor fusion between ORB-SLAM and odometry as formulated in Subsection 4.5. **Algorithm 2** basically outputs a distribution of possible robot pose; definitive pose for the purpose of robot control is taken by averaging this distribution.

Figure 20 depicts trajectory of robot as computed by sensor fusion of odometer and ORB-SLAM of map 1 and map 2 for test run 1. In this figure, we can see that the sensor fusion method is capable to combine the measurement from both maps and remove noise (that came from map 1 due to lens flare in area 3). The sensor fusion method also succeed in covering areas where ORB-SLAM missed tracking. Similar situation is also present in test run 2 whose trajectory is shown in **Fig. 21**. By relating the coverage graph (**Figs. 13** and **14**) and error graphs (**Fig. 19**), most of the spikes in sensor fusion errors can be attributed to ORB-SLAM losing tracks in areas 1 and 2.

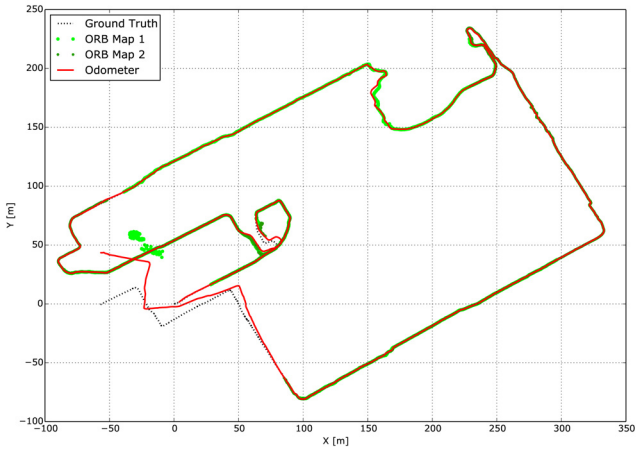

Fig. 20. Trajectory of ORB and odometer for run 1.

Fig. 21. Trajectory of ORB and odometer for run 2.

6. Conclusions and Future Work

This work has reported on our approach for localization solutions for application to the Tsukuba Challenge. Within the limitations of our system, the experiments confirmed that vision-based localization using augmented maps obtained from vision and LIDAR-based methods are capable of providing localization that is quite accurate for controlling the robot. Unlike the original results, ORB-SLAM was unable to produce acceptable results in dynamic environment such as Tsukuba Challenge. We suspect that, due to significant time lapse between mapping and localization, ORB-SLAM was unable to perform place recognition using the bag-of-words method. This is supported by observation that most ORB feature points were on the ground that was covered with fallen leaves.

By using sensor fusion method between ORB-SLAM and odometer, we can achieve continuous coverage of the track. However, due to accuracy problem of the odometer, the localization may give large errors when correction from ORB-SLAM results are absent. In these results, we show that navigation using odometer and ORB-SLAM localization is possible with good accuracy, as long as ORB-SLAM tracking is maintained.

Despite these good results, there are still some rooms for improvement that we should propose. Primarily, there must be an effort to cover the entire spectrum of localization, especially when the vision-based localization fails. For the mapping, we suggest replacing the LIDAR with other global localization methods, such as a GPS-based one. Sensor fusion method between odometer and ORB-SLAM can also use some improvements, especially by using better dead reckoning methods. Another crucial matter is the acceleration of the initial position fix after map restoration. We also did not address visual localization evaluation in adverse weather condition, which is important should this method be implemented in consumer vehicles.

Acknowledgements

This work is supported by MEXT COI stream program, A Diverse and Individualized Social Innovation Hub – The "Mobility Society" for the Elderly: Lead to an Active and Joyful Lifestyle –.

References:

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," arXiv:1502.00956 [cs], February 2015.

[2] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," Artificial Intelligence Review, Vol.43, No.1, pp. 55-81, 2015.

[3] R. Szeliski and S. B. Kang, "Shape ambiguities in structure from motion," Trans. on Pattern Analysis and Machine Intelligence, Vol.19, No.5, pp. 506-512, 1997.

[4] M. Lourakis and X. Zabulis, "Accurate scale factor estimation in 3D reconstruction," Computer Analysis of Images and Patterns, pp. 498-506, Springer, 2013.

[5] H. Lategahn, "Mapping and Localization in Urban Environments Using Cameras," KIT Scientific Publishing, 2014.

[6] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," 6th IEEE and ACM Int. Symposium on Mixed and Augmented Reality 2007 (ISMAR 2007), pp. 225-234, November 2007.

[7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, No.8690 in Lecture Notes in Computer Science, pp. 834-849, Springer International Publishing, January 2014.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," In 2011 IEEE Int. Conf. on Computer Vision (ICCV), pp. 2564-2571, November 2011.

[9] D. Galvez-López and J. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," IEEE Trans. on Robotics, Vol.28, No.5, pp. 1188-1197, October 2012.

[10] A. Nüchter, "3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom," Springer, December 2008.

[11] 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. 60-68, November 2015.

[12] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Örebro universitet, Örebro, 2009.

[13] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," 2011 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1-4, 2011.

[14] E. Takeuchi and T. Tsubouchi, "A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping," 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3068-3073, October 2006.

[15] Y. Morales, A. Carballo, E. Takeuchi, A. Aburadani, and T. Tsubouchi, "Autonomous robot navigation in outdoor cluttered pedestrian walkways," J. of Field Robotics, Vol.26, No.8, pp. 609-635, 2009.

[16] A. Ohshima and S. Yuta, "Teaching-Playback Navigation by Vision Geometry for Tsukuba Challenge 2008," Tsukuba Challenge 2008 Report, pp. 15-18, 2008 (in Japanese).

[17] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," Int. J. of Pattern Recognition and Artificial Intelligence, Vol.2, No.03, pp. 485-508, September 1988.
[18] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular SLAM," Robotics and Autonomous Systems, Vol.57, No.12, pp. 1188-1197, 2009.
[19] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," Cambridge University Press, 2003.
[20] K. Konolige and J. Bowman, "Towards lifelong visual maps," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 2009 (IROS 2009), pp. 1156-1163, 2009.
[21] D. Burschka and G. D. Hager, "V-GPS (SLAM): Vision-based inertial system for mobile robots," Proc. 2004 IEEE Int. Conf. on Robotics and Automation 2004 (ICRA'04), Vol.1, pp. 409-415, 2004.
[22] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," MIT Press, August 2005.
[23] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The New College Vision and Laser Data Set," The Int. J. of Robotics Research, Vol.28, No.5, pp. 595-599, May 2009.

Name:
Adi Sujiwo

Affiliation:
Department of Information Engineering, Graduate School of Information Science, Nagoya University

Address:
609 National Innovation Complex (NIC), Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Biographical History:
2005 Bachelor of Computer Science, Bogor Agricultural University, Indonesia
2009-2011 Magister of Computer Science, University of Indonesia
2011-2014 System Programmer in Bogor Agricultural University, Indonesia
2014- Ph.D. Student, Nagoya University

Name:
Tomohito Ando

Affiliation:
Department of Information Engineering, Graduate School of Information Science, Nagoya University

Address:
Parallel and Distributed Lab, Graduate School of Information Science 4F IB South, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

Brief Biographical History:
2012-2016 Bachelor Student, Nagoya University
2016- Master Student, Nagoya University

Name:
Eijiro Takeuchi

Affiliation:
Department of Media Science, Graduate School of Information Science, Nagoya University

Address:
Takeda Lab, Department of Media Science, Graduate School of Information Science, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

Brief Biographical History:
2008-2014 Assistant Professor, Tohoku University
2014-2016 Designated Associate Professor, Nagoya University
2016- Associate Professor, Nagoya University

Main Works:
• "A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping," 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2006), Beijing, China, pp. 3068-3073, 2006.

Membership in Academic Societies:
• The Japan Society of Mechanical Engineers (JSME)
• The Robotics Society of Japan (RSJ)
• The Society of Instrument and Control Engineers (SICE)
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Yoshiki Ninomiya

Affiliation:
Intelligent Vehicle Research Division, Institute of Innovation for Future Society, Nagoya University

Address:
609 National Innovation Complex (NIC), Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Biographical History:
1981 Received B.S. from Nagoya University
1983 Received M.S. from Nagoya University
1983-2003 Researcher, Toyota Central Lab
2008 Received Ph.D. from Nagoya University
2014- Designated Professor, Nagoya University

Name:
Masato Edahiro

Affiliation:
Department of Information Engineering, Graduate School of Information Science, Nagoya University

Address:
Parallel and Distributed Lab, Graduate School of Information Science, 4F IB South, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

Brief Biographical History:
1985- NEC Research Center
1999 Received Ph.D. in Computer Science from University of Princeton
2011- Professor, Graduate School of Information Science, Nagoya University