Proc. 14th Int. Conf. on Intelligent Autonomous Systems (IAS-14), 2016

Outdoor Robot Navigation Based on View-based Global Localization and Local Navigation

Yohei Inoue, Jun Miura, and Shuji Oishi

Department of Computer Science and Engineering,
Toyohashi University of Technology

Abstract. This paper describes a view-based outdoor navigation method. Navigation in outdoor can be divided into two levels; the global level deals with localization and subgoal selection, while the local level deals with safe navigation in a local area. We adopt an improved version of SeqSLAM method for global-level localization, which can cope with changes of robot's heading and speed as well as view changes using very wide-angle images and a Markov localization scheme. The global level provides the direction to move and the local level repeatedly sets subgoals with local mapping using 3D range sensors. We implemented these global and local level methods on a mobile robot and conducted on-line navigation experiments.

Keywords: Outdoor navigation, view-based localization, mobile robot.

1 Introduction

Mobile service robot is an emerging application area in robotics. Such a robot is expected to provide various service tasks like attending, guiding, and searching. One of the indispensable functions of mobile service robots is *navigation*, which makes it possible for the robot to move from one place to another autonomously. Since outdoor environments are important part of human activity, mobile service robots should be able to navigate in outdoor.

Outdoor navigation can be divided into two levels. The global level deals with localization and subgoal selection, while the local level deals with safe navigation in a local area. This is an analogy to a navigated car driving: a car navigation system tells a driver where the car is and which way to take, and the driver is responsible for safely driving, including following traffic rules and avoiding possible collisions.

Several approaches are possible for outdoor localization. GPS-based systems are usually used, especially in the case of automobiles but could be unreliable or not operational near tall buildings in, for example, usually campus environments. A precise digital map of the environment is also required. Usual mapping and localization approaches might be adopted but making large-scale outdoor maps is often costly. We therefore adopt a simpler way, that is, view-based localization [1–4].

One of the issues in view-based localization is how to cope with view changes. Some of earlier works deal with them using a learning with training data in various illumination conditions [5] or a two-stage SVM-based object/location learning [6, 7]. Yamagi et al. [8] developed a view-based navigation system which uses a robust template matching method. Milford and Wyeth [9] proposed the SeqSLAM method which realizes a very robust image sequence matching even under an extreme view changes. Although this method shows a good performance for image sequences taken from a vehicle, it is not always directly applicable to mobile robot navigation. We therefore use this method with several improvements for realizing a mobile robot navigation in outdoor, combined with a local mapping and path planning capabilities.

The rest of the paper is organized as follows. Sec. 2 describes an improved SeqSLAM method with several off-line experimental validation. Sec. 3 describes a local navigation strategy including local mapping, subgoal selection, and path planning. Sec. 4 shows the navigation experiments. Sec. 5 concludes the paper and discusses future work.

2 View-based Localization by SeqSLAM and Its Improvements

2.1 SeqSLAM

SeqSLAM [9] is a view-based localization method which compares a model image sequence with an input one for robust matching. To cope with a large illumination change between a training and a test time, they apply *local contrast enhancement* as follows.

Let $\boldsymbol{D}$ be a vector of the differences between an input image and the images in the model sequence, which is considered to cover possible range of model images for the input image. Each element D_i in $\boldsymbol{D}$ is normalized by:

$$\hat{D}_i = \left(D_i - \bar{D}_l\right)/\sigma_l, \tag{1}$$

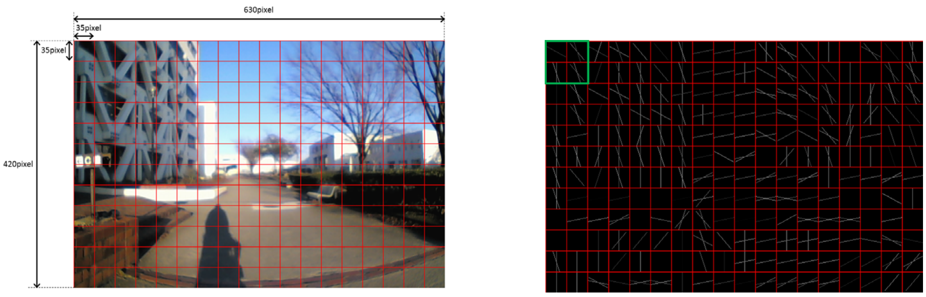
where D_l and σ_l are the mean and the standard deviation of $\boldsymbol{D}$. By this enhancement, even if an input image is largely different from the model images and all of the difference values are very large due to a large illumination change, the difference for the true correspondence is expected to be sufficiently small compared to the others. These enhanced vectors are compiled for $d_s + 1$ frames into a matrix $\boldsymbol{M}$ which has the model and the input image sequence in the row and the column, respectively:

$$\boldsymbol{M} = \left[\hat{D}^{T-d_s}, \hat{D}^{T-d_s+1}, \ldots, \hat{D}^{T}\right] \tag{2}$$

An example matrix is shown in Fig. 3.

Then, assuming a constant velocity during the sequence, a line is searched for which minimizes the following total difference S:

$$S = \sum_{t=T-d_s}^{T} D_k^t, \tag{3}$$



(a) input image.

(b) extracted HOG features.

Fig. 1. HOG extraction result.

$$k = s + V(d_s - T + t), \tag{4}$$

where V is the gradient of the line (or a relative velocity in input and model acquisition) and k is the index of the corresponding image in the model sequence for the input image at time t.

SeqSLAM exhibited great performances against drastic view changes, at least for road sequence images. There are, however, rooms for improvements when applied to mobile robot navigation. The following subsections explain our improvements.

2.2 Improvements in image matching

The original SeqSLAM uses intensity values normalized within a small window for the feature for image matching. This is simple and fast, but is not very strong for a region with little textures. It is also weak to a large view direction changes. We therefore adopt two improvements: HOG feature matching and the use of a wide angle camera.

HOG feature matching: HOG feature [10] is a histogram of edges in a local region and suitable for representing shape information. The size of training images is 630×420 pixels with 90° FOV (field of view). The cell size for calculating HOG is 35×35 pixels and the number of blocks is $17 times 11$. Fig. 1 shows an example result of HOG calculation. We use a normal SAD (sum of absolute differences) for calculating the dissimilarity between images.

Coping with a variety of robot motion direction: Mobile robots changes their moving directions frequently not only for moving towards a destination but also avoiding collisions with people and obstacles. Since each image in a view sequence captures a scene in a specific direction, it is very much likely to have a different orientation during navigation, thereby degrading the view-based localization performance.

Morita and Miura [11] used an omnidirectional camera to cope with this problem. We also take a similar approach using a wide images (1190×420 pixels with about 180° Horizontal FOV), with which 33×11 blocks are obtained. We

(a) Training image. (b) Input image with the selected direction (red box).

Fig. 2. Selection of moving direction.

scan a learned image horizontally on the wide image within $\pm 40°$ range with $5°$ interval, and chooses the minimum distance position, which is then used for determining the subgoal direction (i.e., the direction for the robot to move). Fig. 2 shows an example of selecting a direction.

2.3 Improvements in image sequence matching

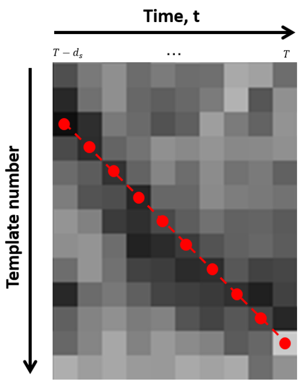
The original SeqSLAM assumes a constant speed during acquisition of training and input image sequences; the matrix is searched for the best line which minimizes the total difference. This assumption is sometimes violated in the case of mobile robots because they need to adjust their speed adaptively to the surrounding situation for, for example, avoid collision and/or threatening to people. We therefore use a DP to cope with such speed variations during image acquisition. We also effectively utilizes the history of movement to increase the reliability and reduces the calculation cost.

DP Matching: DP (dynamic programming) matching [12] is a tool for calculating a match between two data sequences with non-constant interval between data. Fig. 3 shows an example of DP matching for image sequences with non-constant robot motions; a linear matching is not suitable for this case. We set a limitation on a speed difference between the training and the navigation phase and apply the DP matching for obtaining the best matched image pairs with an evaluation. In addition, unlike SeqSLAM, we the latest frame as a representative image of a sequence so that the current location is estimated on-line.

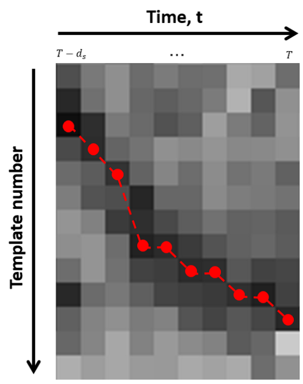
Markov localization: Mobile robot localization often uses a movement history, which is effective to limit the possible robot positions in prediction. Miura and Yamamoto [7] adopted a Markov localization strategy in a view-based localization. In [7], a discrete set of locations are provided for localization and a probabilistic model of transitions between locations was used in the prediction step. This can reduce not only localization failures but also the calculation cost with a limited number of sequence matches.

The Markov localization here is formulated as follows:

$$\hat{Bel}(l) \leftarrow \sum_{l'} P_m(l|l')Bel(l'), \tag{5}$$



(a) line-based matching is not optimal.



(b) DP matching can find the best matches.

Fig. 3. DP matching between training and navigational image sequences. In the matrix, darker elements have smaller differences.

$$Bel(l) \leftarrow \alpha P_o(s|l)\hat{Bel}(l), \tag{6}$$

$$P_o(s|l) = \frac{S_{min}}{S_l}, \tag{7}$$

where $P_m(l|l')$ denotes the transition probability from frame l' to l, $Bel(l)$ the belief of the robot being location l, $P_o(s|l)$ the likelihood of location l with sensing s, which is calculated by the minimum matching score of the DP matching divided by the score for location l.

The state transition model $P_m(l|l')$ is determined by considering the image acquisition interval and the robot motion patterns. Currently, the training images are taken with about one meter interval and the robot takes one image per two seconds with moving at $1\,m/s$. Since the robot speed changes frequently due to many reasons such as collision avoidance and turning motions, we use a transition model in which the robot may move to locations corresponding to one of the current and the three subsequent location with equal probabilities.

2.4 Off-line localization Experiments

Fig. 4 shows the route used for the experiments. This is in our campus and about $300\,m$ long. We manually moves the robot on this route and acquired one training image set and two testing image sets. The training set and the first testing image set (test1) were taken while the robot moves along the route, while the second testing image set (test2) was taken as the robot did zig-zag motions so that the direction of the robot changes largely from position to position. The image sets are summarized in Table 1.

Fig. 5 shows the results of localization experiment. We compared the proposed method and SeqSLAM for test1 and test2 image sequences. Since the original SeqSLAM exhibits quite a low performance for the test2 image sequence due

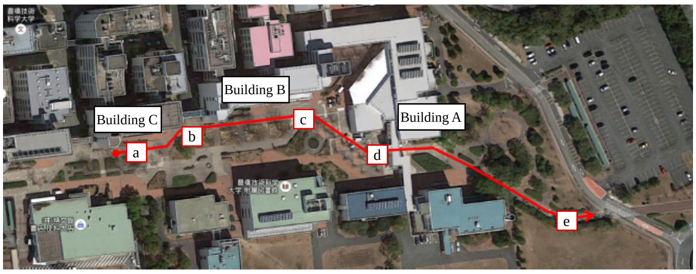


Fig. 4. The route for experiments.

Table 1. Training and testing image sets.

	camera	Date and weather	# of images	robot motion
training	normal	March 5, 2015, fine	259	smooth
test1	wide	July 11, 2015, cloudy	263	smooth
test2	wide	July 11, 2015, cloudy	282	zig-zag

to a large variation of the robot heading, we additionally performed a horizontal scanning to find the best matched position in the wide image. Both comparison results show that the proposed method exhibits a much better performance.

Fig. 6 shows the evaluation of localization accuracy. The ground truth is determined by manually comparing the training and test images. When an input image is judged to be located between two consecutive training images, the true position is set in the middle of the training images. The maximum frame difference by the proposed method is two for most of frames, meaning the maximum localization error is about $2\,m$ because the training images are acquired with about one meter interval. Table 2 summarizes the performance in terms of *localization success rate* and *direction selection success rate*. Localization is considered success when the difference is within two frames, while the direction is considered correctly selected when the directional difference is less than $5°$. Fig. 7 shows a scene where the proposed and the SeqSLAM with scanning suggest different moving directions. Since SeqSLAM does a direct comparison of (normalized) pixel values, it is sometimes weak to scenes with less textures as shown in the figure.

3 Local Path Planning using View-Based Localization Results

The proposed view-based localization method provides the direction to move. That information by itself is, however, not enough for guiding an actual robot safely. We therefore develop a local navigation system which includes local mapping, subgoal selection, and path planning.

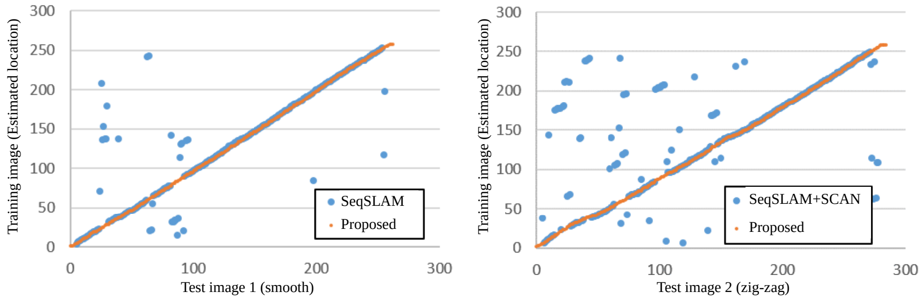


Fig. 5. Localization results.

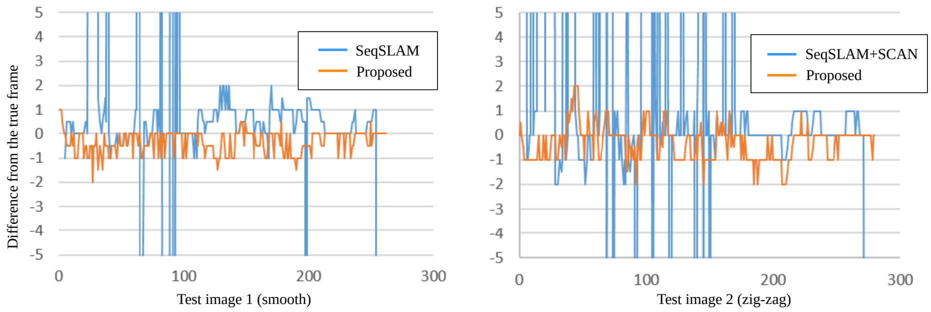


Fig. 6. Localization accuracy.

Table 2. Quantitative evaluation results.

test image	method	localization success rate (%)	direction selection success rate (%)
test1	SeqSLAM	86.1	-
	Proposed	99.6	99.6
test2	SeqSLAM	74.2	63.1
	Proposed	95.8	95.8

3.1 Local mapping

Fig. 8 shows our mobile robot. It is based on an electric wheelchair (Patrafour by Toyota Motor East Japan Inc.), equipped with two 3D laser range finders (LRFs) (FX-8 by Nippon Signal Co.) for local mapping and finding free spaces, and a wide-angle camera for view-based localization. Each LRF has about 60° horizontal FOV and two LRFs covers about 100° FOV.

We use two 3D laser range finder for local mapping, which is for finding free spaces. We detect obstacles in two ways. One is to use a height map which detects regions with relatively high obstacles. We use a polar coordinate with 1°

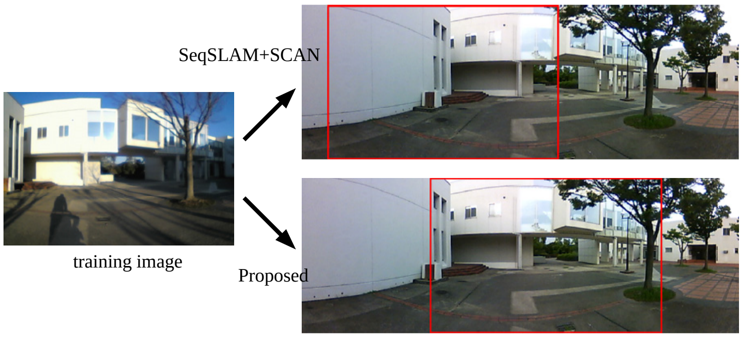


Fig. 7. Comparison in selecting the moving direction.

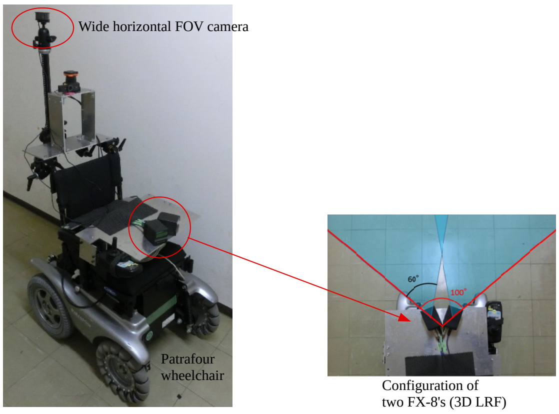


Fig. 8. Our robot with two LRFs and a camera.

and 50 cm intervals in the angle and the distance axis, respectively, for representing height maps. The pose of the range sensors relative to the ground plane is estimated by fitting a plane to the data points in the region in front of the robot. We set a threshold to 15 cm to detect this kind of obstacles.

The other way is to find low steps. Since it is sometimes difficult to such steps only from the height due to a limited ranging accuracy and the error in ground plane estimation, we examine the differentiation of height data. A region with a large height difference with a certain number of data points is considered to be an obstacle region (i.e., a low step). Fig. 9 shows an example of low step detection in a real scene. The steps near the robot are successfully detected.

Fig. 9. Detection of low steps. A local obstacle map centered at the current robot position is shown on the right. Red ellipses on the left show the detected step locations.

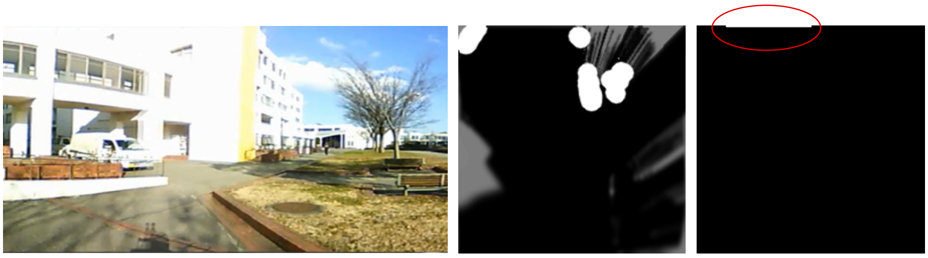


Fig. 10. Subgoal selection example. Left: scene, center: local map with enlarged obstacles, right: frontier whose center point is selected as the subgoal.

3.2 Subgoal selection and path planning

The direction to move is suggested by the view-based localization, however, it is not always possible to move in that direction due to obstacles. Since the free spaces are recognized only in the local map, we need to set a subgoal in the local map which is safe and leads the robot to the destination. We here adopt the concept of *frontier* which is often used in exploration planning in an unknown space [13]. A frontier point is a point which is free and adjacent to an unknown point. Such a point either inside the local map or on the edge of the map. All frontier points are partitioned into clustered, among which the ones with line-shape and having a certain size are selected and their center points become the candidates for the subgoal. The most appropriate center point is then selected as the subgoal which has the minimum orientational difference with the suggested moving direction. Fig. 10 shows an example selection of subgoal (frontier).

Once the subgoal is set, the path towards it is generated. We use our RRT-based on-line path planner [14]. Since the cycle of view-based localization is about two seconds while the path planner runs in a faster cycle, we use the same subgoal until the next subgoal is set based on the next result from the view-based global localization.

4 Navigation Experiment

We conducted experiments in the route shown in Fig. 4. The training image set used is the one used in the off-line experiments (see Sec. 2.4). Fig. 11 shows snapshots of a run conducted on Feb. 15, 2016. The names of the locations correspond to symbols in Fig. 4. The robot successfully completed the autonomous navigation.

Fig. 12 shows an off-line comparison between the proposed method and SeqSLAM with scanning for the image sequence taken during the actual navigation. The figure shows that the proposed global localization method works on-line with a high reliability. During other experiments, however, the robot sometimes stuck or almost hit obstacles mainly due to obstacle detection failures. It is necessary to improve the local navigation system for more reliable global navigation.

5 Conclusions and Future Work

This paper described an outdoor navigation system combining a view-based global localization method and a local navigation system. We developed an improved version of SeqSLAM and has shown its reliability in outdoor navigation. We also realized a robot system that can navigate itself given an image sequence of the route to follow.

The system has been validated in a route in our campus. It is necessary to evaluate the system in a more variety of scenes, that is, more variations in weather, season, surrounding objects (buildings or forests), and so on. For a more reliable navigation, it is also necessary to improve the performance of local mapping and navigation is required.

Acknowledgment

This work is in part supported by JSPS KAKENHI Grant Number 25280093.

References

1. Y. Matsumoto, M. Inaba, and H. Inoue. Visual Navigation using View-Sequenced Route Representation. In *Proceedings of 1996 IEEE Int. Conf. on Robotics and Automation*, pp. 83–88, 1996.
2. H. Katsura, J. Miura, M. Hild, and Y. Shirai. A View-Based Outdoor Navigation Using Object Recognition Robust to Changes of Weather and Seasons. In *Proceedings of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2974–2979, 2003.
3. 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.
4. C. Valgren and A. Lilienthal. SIFT, SURF and Seasons: Long-term Outdoor Localization Using Local Features. In *Proceedings of European Conf. on Mobile Robots*, pp. 253–258, 2007.

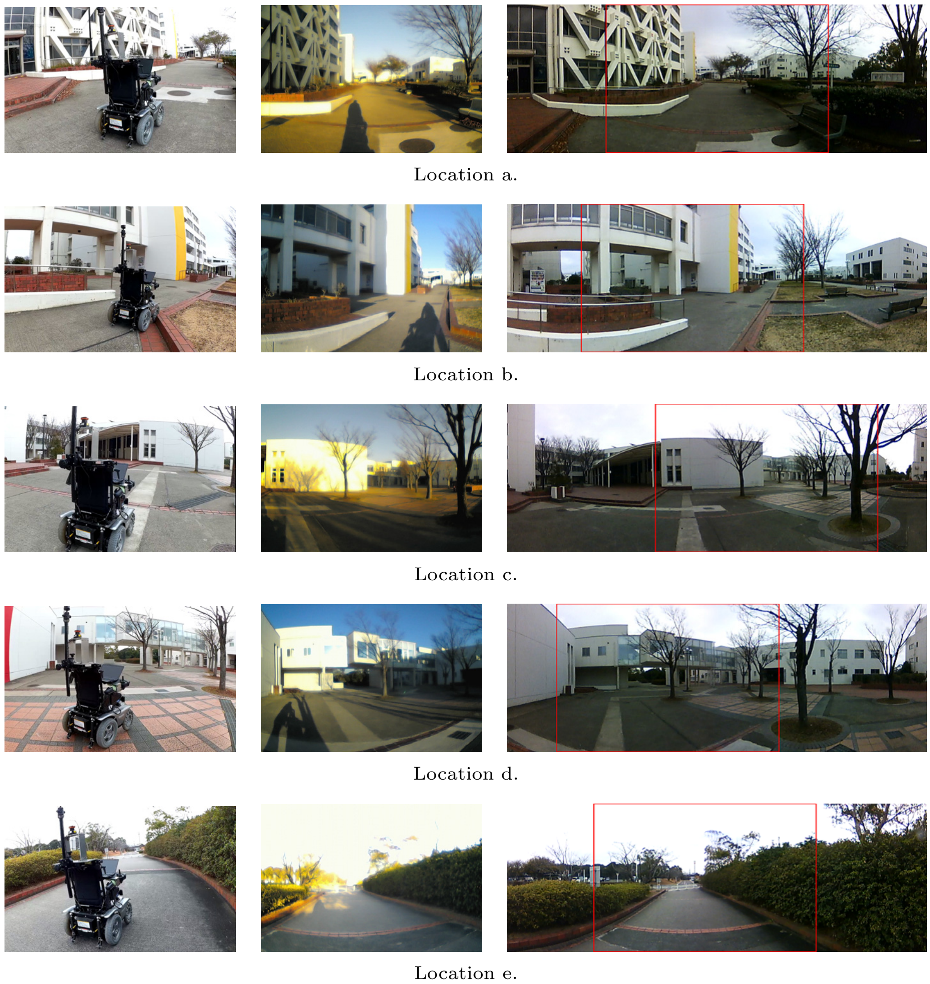


Fig. 11. Navigation experiment. Left: Robot in motion. Center: Estimated location (i.e., best matched training image). Right: Input image and selected moving direction.

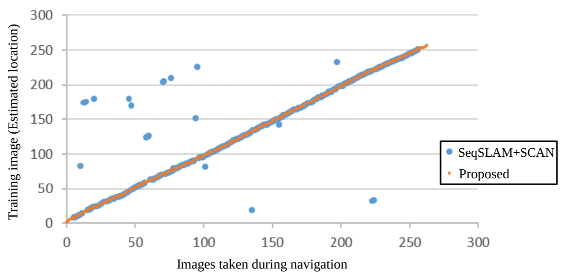


Fig. 12. Comparison using the image sequence taken during an actual navigation.

5. D.M. Bradley, R. Patel, N. Vandapel, and S.M. Thayer. Real-Time Image-Based Topological Localization in Large Outdoor Environments. In *Proceedings of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3062–3069, 2005.
6. H. Morita, M. Hild, J. Miura, and Y. Shirai. View-Based Localization in Outdoor Environments Based on Support Vector Learning. In *Proceedings of 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3083–3088, 2005.
7. J. Miura and K. Yamamoto. Robust View Matching-Based Markov Localization in Outdoor Environments. In *Proceedings of 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2970–2976, 2008.
8. Y. Yamagi, J. Ido, K. Takemura, Y. Matsumoto, J. Takamatsu, and T. Ogasawara. View-sequence Based Indoor/Outdoor Navigation Robust to Illumination Changes. In *Proceedings of 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1229–1234, 2009.
9. 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.
10. N. Dalal and B. Briggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of 2005 IEEE Conf. on Computer Vision and Patttern Recognition*, pp. 886–893, 2005.
11. H. Morita, M. Hild, J. Miura, and Y. Shirai. Panoramic View-Based Navigation in Outdoor Environments Based on Support Vector Learning. In *Proceedings of 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2302–2307, 2006.
12. Y. Ohta and H. Yamada. Pattern Matching Using Dynamic Programming. *Information Processing*, Vol. 30, No. 9, pp. 1058–1066, 1989.
13. B. Yamauchi. A Frontier-Based Approach for Autonomous Navigation. In *Proceedings of the 1997 IEEE Int. Conf. on Computational Intelligence in Robotics and Automation*, pp. 146–151, 1997.
14. I. Ardiyanto and J. Miura. Real-Time Navigation using Randomized Kinodynamic Planning with Arrival Time Field. *Robotics and Autonomous Systems*, Vol. 60, No. 12, pp. 1579–1591, 2012.