

A View-based Wearable Personal Navigator with Inertial Speed Estimation

Masashi Harada, Jun Miura, and J. Satake
Department of Computer Science and Engineering
Toyohashi University of Technology

Abstract—This paper describes a view-based wearable personal navigation system. We have been developing a view-based outdoor localization method which has been proven to be robust to changes of weather and seasons and can be operational even when sufficient GPS signals is not available. The system is composed of a fish-eye camera, an accelerometer, and a mobile PC. The camera is worn in a pendant-like manner and its wide view is effective in coping with the swinging motion of the camera during walking. The accelerometer is used for estimating the walking speed, based on which the user location can be predicted on-line. A Markov localization method repeatedly estimates the user location and generates navigational voice guidance. The system has been tested in our campus to show its effectiveness.

Index Terms—Personal navigation, View-based navigation, Inertial motion estimation.

I. INTRODUCTION

Navigating people is one of the promising areas to which robotic technologies can be applied. GPS-based navigation systems have been widely used for vehicle navigation [1]. Recent portable devices such as tablets and smartphones also utilize GPS for localization and navigation of people. GPS signals are, however, not always available especially near tall buildings (see Fig. 1 for example data for GPS-based positioning availability). An accurate digital map with annotations is also necessary for GPS navigation.

Dead reckoning using inertial sensors can provide a reasonable motion estimation but it suffers from drift problems. It is, therefore, necessary to compensate it with some absolute location information such as GPS, RFID-tags, and/or image-based landmarks [2], [3], [4].

Vision is one of the informative sources for locations, and recent advances in visual learning techniques have led to development of view-based localization approaches (e.g., [5], [6]). Some works have shown their robustness to changes of weather and/or seasons (e.g., [7], [8]). In this paper, we adopt such a view-based localization method, especially the one using a two-stage SVM-based reasoning strategy [9], as a basis of the personal navigator.

Fig. 2 shows an overview of the developed personal navigator. A fish-eye camera is mounted on a metal plate, which is hung by a rope like a pendant. This metal plate is long enough to touch the body with a large area so that the swinging motion of the camera is reduced. An accelerometer is attached to the user body to roughly estimate the user speed. All sensor data processing is done by a notebook PC. The user is navigated by voice guidance.

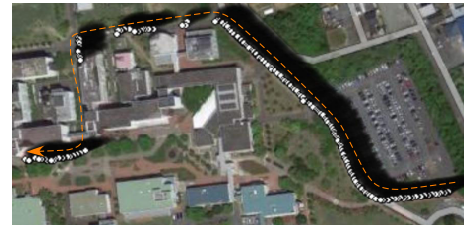


Fig. 1. Availability of GPS positioning. Position data were taken along the route shown by the orange dashed line and white marks indicate the locations where the GPS positioning was available. Many black spots exist.

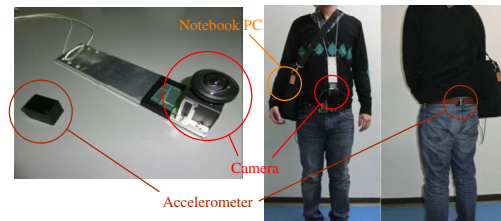


Fig. 2. Hardware configuration.

The rest of the paper is organized as follows. Sec. II briefly explains the view-based localization method on which the developed system is based. Sec. III describes how to choose the best view from a fish-eye image. Sec. IV a walking speed estimation using an accelerometer for on-line state prediction. Sec. V explains how to navigate the user by voice guidance. Sec. VI shows experimental results. Sec. VII summarizes the paper and discusses future work.

II. VIEW-BASED LOCALIZATION

A. Overview of the view-based localization method

View-based localization typically works as follows. During the training phase, an image sequence along a route is acquired. In the subsequent navigation phase, input images are compared with learned ones to determine locations. The most difficult part of this approach is finding the most appropriate internal representation and an appropriate learning algorithm which is capable of generating this internal representation.

Our view-based localization method takes a two-stage SVM approach [9]. The *object recognition SVMs* classify windows in the image into several object categories such as the sky and trees. The output of the first-stage is a 0 – 1 state vector describing the existence of each object at each window. This state vector is then used by the second-stage *localization SVMs* for discriminating one location from the others.

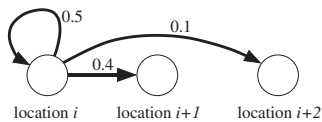


Fig. 3. Transition model.



Fig. 4. Localization examples [9]. The upper row indicates input images and the lower row output images (i.e., best matched images in the training images). The training images were taken on a sunny summer day; the input images were taken on (from left to right) a rainy summer day, a sunny summer evening, and a cloudy winter day, respectively.

We select a set of discrete locations on a route and train one localization SVM for each location; the localization is, therefore, to determine on which location the robot or the user is. To increase the reliability of localization, we adopt the Markov localization approach, by which not a single location but a probabilistic distribution over possible locations is estimated.

Markov localization alternately performs *prediction* and *correction* [10]. The prediction step is done by:

$$\hat{Bel}(l) = \sum_{l'} P(l|l') Bel(l'), \quad (1)$$

where $Bel(l)$ and $\hat{Bel}(l)$ are the belief that the user is at location l before and after the current observation, respectively. $P(l|l')$ is the *state transition model* which predicts possible future locations from a current location. Fig. 3 shows an example model used for vehicle localization [9].

The correction step is done by:

$$Bel(l) = \alpha P(o|l) \hat{Bel}(l), \quad (2)$$

where $P(o|l)$ is the likelihood of location l given observation o and calculated based on the similarity of the object placements between the input and the corresponding model; α is a normalization constant.

Fig. 4 shows some results of our view-based localization. We can see that, in spite of view changes, correct images are retrieved.

B. Selection of locations

Discrete locations for which localization SVMs are trained can be placed arbitrarily on the route. We basically put locations at a regular interval but take a more care for the corners and the goal because specific navigational guidance should be made for such places.

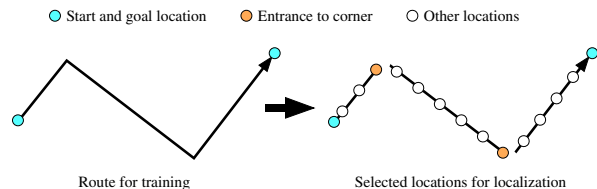


Fig. 5. Selection of locations for localization.

Fig. 5 shows the process of selecting locations. Given an image sequence taken along a route, we first segment it into almost straight route segments. This is done by calculating optical flow and identifying turning motions at corners where large horizontal optical flow is observed. For each route segment, we set one location at its end, and others at a regular interval (currently, about 10 [m]) backwards from the end location. Circles in Figs. 13 and 14 show examples of automatically selected locations.

Additional information is attached to each location which will be used in verbal guidance. For the endpoint of each segment, we describe whether it is the goal, and the turning direction (left or right) in the case of corners. For other locations, we describe the number of steps to the next end, which provides how near that location is to the end.

C. Issues when applied to people navigation

Our view-based localization was originally for robot/vehicle navigation. Important characteristics of robots and vehicle are: (1) the viewing direction during motion is relatively stable; (2) the robot speed can be controlled; (3) navigation information can directly be transferred to the robot.

In the case of personal navigator, where the user hangs a camera, we cannot expect these characteristics; that is:

- (1) The viewing direction may vary largely due to the swinging motion of the hanged camera or change of the user's walking direction.
- (2) The user may change walking speed or even stop for a while. A fixed state transition model is not appropriate.
- (3) It is not comfortable to look at the screen of a PC or a tablet.

To handle these issues, we expand the localization method and develop a system as follows.

- A fish-eye camera is used for getting views in various directions.
- The user's speed is estimated using an accelerometer in order to modify the state transition model on-line.
- Navigational verbal guidance is given to the user at appropriate timings.

The details of these extensions will be described in the subsequent sections.

III. VIEW MATCHING ROBUST TO CAMERA HEADING CHANGES

Our view-based localization compares input and model images in terms of object placements in the image. This makes it sensitive to the change of the camera heading; a

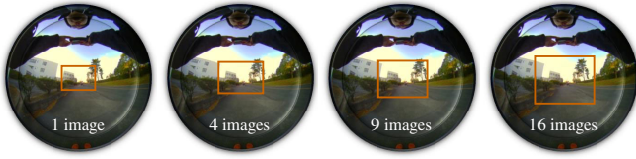


Fig. 6. Regions for extracting multiple images.

TABLE I

NUMBER OF EXTRACTED IMAGES VS. SUCCESS/HIGHEST-SCORE RATE.

| # of extract images | success rate | highest-score rate |
|---------------------|--------------|--------------------|
| 1 | 100.0% | 70.5% |
| 4 | 100.0% | 83.3% |
| 9 | 100.0% | 75.2% |
| 16 | 100.0% | 72.5% |

small difference between headings in the training and the localization/navigation phase may cause localization failure.

For a mobile robot localization under horizontal heading changes, we used panoramic images [11]. In the personal navigation where the camera heading may change not only horizontally but also vertically, we use a fish-eye camera and extract images corresponding to possible headings.

We tested four patterns of image extraction; the numbers of extracted images are one, four, nine, and sixteen, respectively. Fig. 6 shows the region from which images are extracted for each case.

Table I shows the comparison result of these image extraction alternatives in terms of localization accuracy. We used the following two evaluation criteria:

- (1) *Success rate*: the ratio of the number of locations that are correctly recognized versus the total number of locations. In our Markov localization, a localization result is considered correct if the true location has a non-zero probability.
- (2) *Highest-score rate*: the ratio of the number of locations that get the highest score (i.e., posterior probability) versus the total number of locations.

From this result, we choose to extract four images from the fish-eye input image.

The reason why extracting more images degrades the performance is conjectured as follows. When buildings exist only at one side of the view (say, right side), moving the robot forward makes those building move rightward in the image, but this effect is similar to the one obtained when the camera heading moves leftward. Increasing the number of extracted images has a similar effect to view changes, and some of the images may match well with the model for incorrect different locations.

IV. INERTIAL SPEED ESTIMATION AND STATE PREDICTION

A. Speed estimation using an accelerometer

Speed estimation is useful for predicting the user position. It could be possible to estimate the user motion by integrating data from accelerometers, gyrosensors, and magnetometers [4]. In the case of route guidance, only motion information

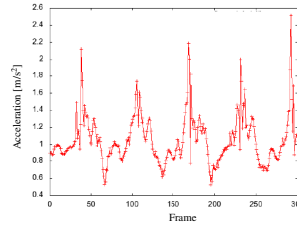


Fig. 7. Change of acceleration during a constant walk.

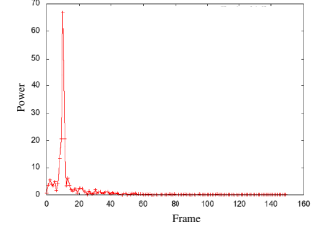


Fig. 8. Power spectrum of Fig. 7 obtained by using FFT.

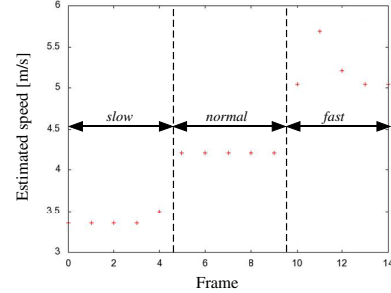


Fig. 9. A sequence of estimated speeds for three different walks. Each interval indicate the one for which a specific walk was ordered.

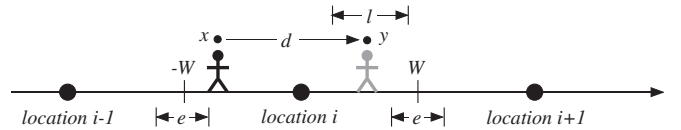


Fig. 10. Situation for constructing transition models.

along a route is necessary. So we use a low-cost accelerometer to estimate the user's speed. It is, however, not reliable to calculate the speed by integrating accelerometer outputs. We thus take an indirect approach using a walking frequency estimation.

Fig. 7 shows the change of the magnitude of acceleration during a walk of a person. We can see a constant rhythm coming from the touch of the legs to the ground. From the data, we can estimate the frequency by detecting the largest peak position in the power spectrum (see Fig. 8). When the height of the largest peak is less than a threshold, the person is considered stopping.

We here suppose an adult male uses the navigation system, whose stride is about 70 [cm]. The speed is then estimated by multiplying the stride by the walking frequency. We compared the estimated and the real speed for six persons and their difference is about 20% of the estimated speed in average. This is used for calculating the state transition model (see eq. (1)) on-line. Fig. 9 shows a sequence of estimated speeds, where a subject was asked to walk at three different speeds: slow, normal, and fast. These three types of walk are clearly discriminated from the estimated speed.

B. On-line calculation of state transition models

State transition models (i.e., $P(l|l')$ in eq. (1)) provide prior probabilities in the Markov localization framework; a more accurate model will give a better localization. We used

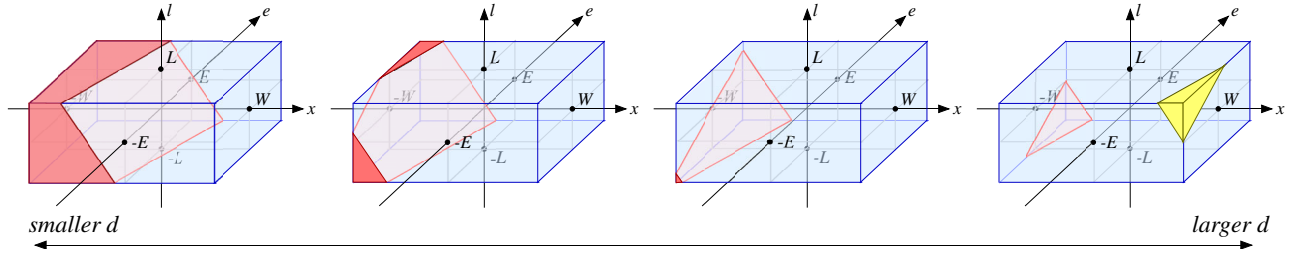


Fig. 11. Four feasible cases considered in calculating transition probabilities.

a fixed model in the case of vehicle localization where the vehicle speed is reasonably constant except at corners (see Fig. 3). In the case of personal navigation, however, since the user may change the speed from place to place or sometimes even stops, we need to construct (or adjust) the transition model depending on the estimated user speed.

In constructing the transition model, we consider the following three factors:

- Our view-based localization method gives not a position estimate of continuous value on the route but a discrete location.
- The interval between locations includes errors.
- The measured user speed includes errors.

Fig. 10 illustrates how to construct transition models on-line. Let the user be at location i . Since the exact position cannot be obtained, the user is considered to be within the area which is centered at location i and has the width of $2W$, where W is the half of the interval between locations. The user position x is then specified by the following inequality (the origin is supposed to be at location i):

$$-W \leq x \leq W. \quad (3)$$

Consider the case where the user moves by distance d and let l be its error. From the measurement system, we know the range of error in l as:

$$-L \leq l \leq L \quad (4)$$

Lastly, the interval $2W$ has error $2e$. From the error in taking the training image sequence, we know the range of error in e as:

$$-E \leq e \leq E \quad (5)$$

Supposing no specific error models, we assume the three variables, x , l , and e , follow uniform distribution with the above range. This means that the probabilistic distribution of the errors in the x - e - l 3D space is represented by the rectangular parallelepiped shown in Fig. 11.

The user's position y after the movement by distance d is given by:

$$y = x + d + l. \quad (6)$$

If the user is still at location i after the movement, considering the error e in the location interval, the following conditions are satisfied:

$$-W - e < x + d + l < W + e, \quad (7)$$

or rewritten as

$$-x - d - W - e < l < -x - d + W + e. \quad (8)$$

Eq. (8) represents a subspace in the x - e - l space sandwiched by two planes. The probability that the user is at location i after the movement is obtained by calculating the volume of the corner between the subspace and the uniform distribution.

How these two entities intersect with each other depends on the moving distance d and the ranges of uniform distribution, W , L , and E . We have examined nine possible cases and found four of them shown in Fig. 11 are feasible if we suppose that d is comparable to the interval between locations (i.e., $2W$), and that the user does not move backward. In the figure, the red, the blue, and the yellow volume correspond to the transition to location i , location $i + 1$, location $i + 2$, respectively.

V. VOICE NAVIGATION

A. Navigation through several corners

The current system is developed for navigating people on a route which is composed of almost straight segments connected by corners (see Fig. 5). As the user proceeds, the system repeatedly estimates the probabilistic distribution over the locations.

We do two special treatments in the location estimation for corners. First, when the probability of an corner location exceeds a threshold, the user is considered to be at the corner. Second, when the user is known to be at an corner, the probabilistic distribution is re-initialized, namely, 1.0 is given to the location just after the corner.

B. Navigational voice guidance generation

The system generates navigational voice guidance at each localization step. The localization method outputs a probabilistic distribution of possible locations. So the user is considered at the location with the highest probability and this will be used for generating the voice guidance.

Most important navigational guidance is to make the user turn at a right corner towards the right direction. So the system gives a notice to the user when he/she is approaching the next corner, as in the case of usual vehicle navigation systems. When being near to an corner, we generate guidance as follows:

- At locations less than or equal to three steps ahead to the next corner, the system gives a notice: "Next corner is approaching. Move forward."

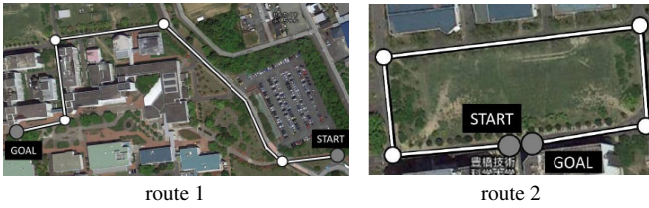


Fig. 12. Two routes used for navigation experiments.

- At the corner location, the system gives a guidance to make a turn: “*You are at the corner. Make a left (or right) turn.*”

Similar guidance is generated for the goal. For the other locations, which are on a segment and are not near to the corner or the goal, just generate a simple guidance: “*move forward along the road.*”

The voice messages are generated using SofTalk [12], a free software that can generate voice message from a given text.

VI. EXPERIMENTAL RESULTS

We implemented the proposed method on a notebook PC (Core i5 M520, 2.40GHz, 3GB memory). The time for one localization cycle, including image capture and conversion, object recognition, and localization, is about 2.5 [s].

We conducted person navigation experiments in two routes in our campus (see Fig. 12). Route 1 is a 500 [m] long route with four corners. Route 2 is a 340 [m] long, closed route with four corners. Three subjects, who are students of our university, tested the system for the two routes.

Fig. 13 shows the results of a navigated run of a subject on route 1. In the upper row of each figure, the left, the center, and the right area indicate the input image, the best-matched training image, and the generated navigational voice guidance, respectively. The lower row indicates the probabilistic localization result. Circles are chosen locations and the color of each circle indicate its probability, from pink to red; white means zero probability. We can see that in spite of a large view change, view matching and thus the localization are successfully performed. Fig. 15 shows the change of the estimated speed for this run. The subject made occasional stopping during the navigation; such stopping were correctly recognized by the speed estimation. Fig. 14 is the result for another route. Table II summarizes the statistics of localization performance in the experimental runs, which shows the method is reliable enough.

VII. CONCLUSIONS AND DISCUSSION

This paper has described a view-based personal navigation system. The system is composed of a wide-angle camera for view-based localization and an accelerometer for motion estimation. It can navigate people with voice guidance generated in a timely fashion based on a robust and reliable localization and on-line motion estimation.

The experimental results shown in this paper are for the subjects familiar with our campus, but not necessarily familiar with the routes used. It is necessary to validate the

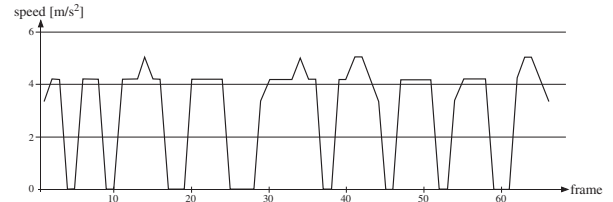


Fig. 15. The change of the estimated speed during a navigation.

TABLE II
EVALUATION OF LOCALIZATION PERFORMANCE FOR THE TWO ROUTES.

| Subject | Route 1 | | Route 2 | |
|-----------|---------|---------|---------|---------|
| | s-rate | hs-rate | s-rate | hs-rate |
| Subject 1 | 100.0% | 95.8% | 100.0% | 89.3% |
| Subject 2 | 100.0% | 89.6% | 100.0% | 85.7% |
| Subject 3 | 100.0% | 93.8% | 100.0% | 87.5% |

s-rate: success rate hs-rate: highest-score rate

system with users of various ages who are unfamiliar with the campus.

The system relies on the user’s ability to recognize corners. It is therefore possible that the user takes a wrong route by missing the correct corner or turning at a wrong corner close to the correct one. To cope with such cases, the system should be able to identify the failure situation and to navigate the user to the correct route by, for example, suggesting him/her to go back to the previous corner and take another branch. We are now developing a method to identify failure situations based on the likelihood of locations.

Acknowledgment: The authors would like to thank Mr. Yuki Kaneko and Mr. Haruo Miyawaki for GPS measurement. This work is supported in part by Grant-in-Aid for Scientific Research (No. 21300075) from JSPS.

REFERENCES

- [1] Yilin Zhao. *Vehicle Location and Navigation Systems*. Artech House Inc., Boston, 1997.
- [2] Q. Ladetto, V. Gabaglio, and B. Merminod. Two Different Approaches for Augmented GPS Pedestrian Navigation. In *Proceedings of Int. Symp. on Location Based Services for Cellular Users*, 2001.
- [3] M. Kourogi and T. Kurata. Personal Positioning based on Walking Locomotion Analysis with Self-Contained Sensors and a Wearable Camera. In *Proceedings of the 2nd Int. Symp. on Mixed and Augmented Reality*, pp. 103–112, 2003.
- [4] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata. Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/RFID/Self-contained Sensor System. In *Proceedings of the 16th Int. Conf. on Artificial Reality and Telexistence*, pp. 1310–1321, 2006.
- [5] W. Zhang and J. Kosecka. Image Based Localization in Urban Environments. In *Proceedings of the 3rd Int. Symp. on 3D Data Processing, Visualization, and Transmission*, pp. 33–40, 2006.
- [6] 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.
- [7] 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.
- [8] 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.

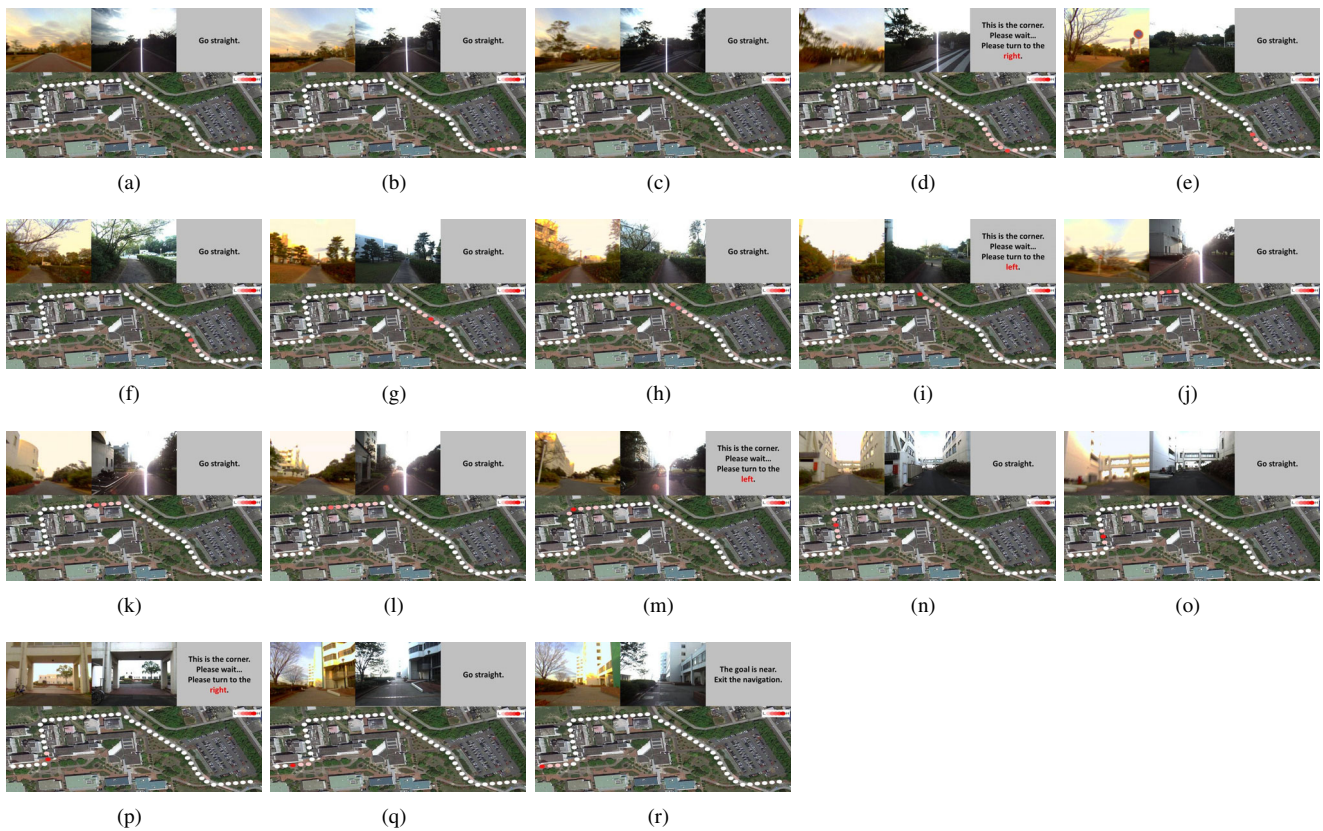


Fig. 13. Navigation result for route 1.

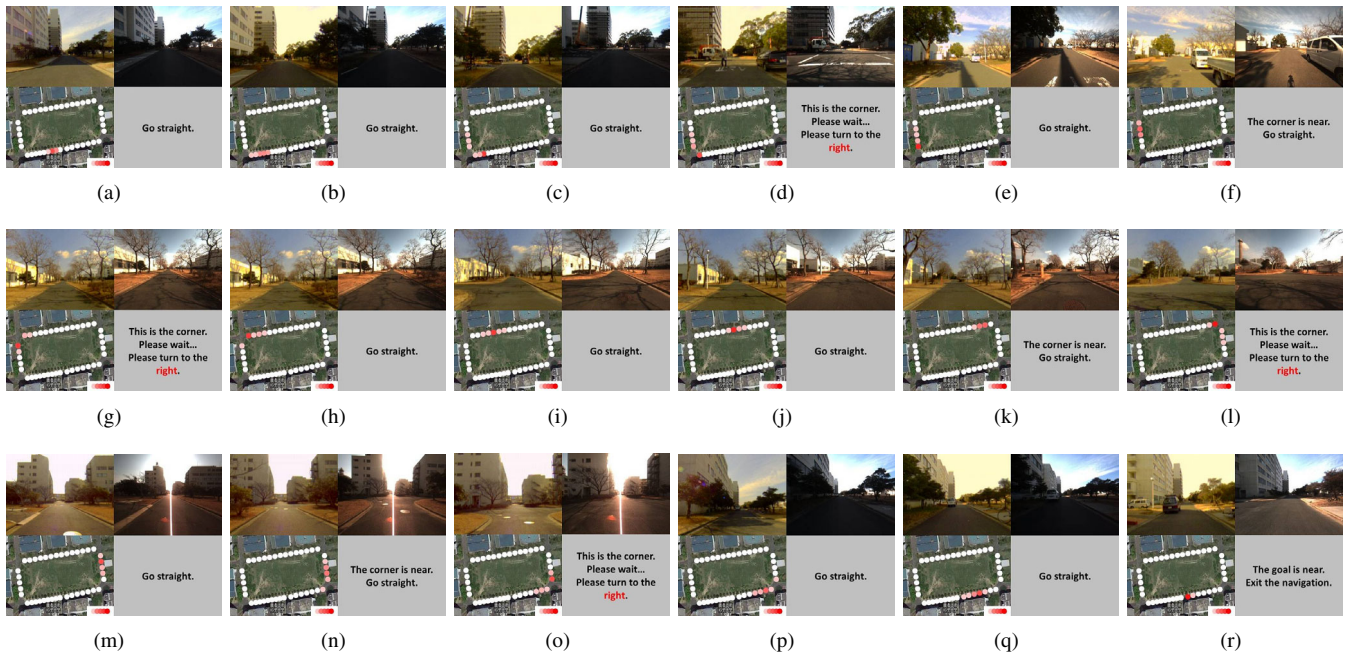


Fig. 14. Navigation result for route 2.

[9] 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.

[10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 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] SofTalk. <http://www35.atwiki.jp/softalk/>.