

# Panoramic View-Based Navigation in Outdoor Environments Based on Support Vector Learning

Hideo Morita<sup>†</sup> Michael Hild<sup>††</sup> Jun Miura<sup>†</sup> Yoshiaki Shirai<sup>†††</sup>

<sup>†</sup>*Department of Mechanical Engineering, Osaka University*

<sup>††</sup>*Department of Engineering Informatics, Osaka Electro-Communication University*

<sup>†††</sup>*College of Computer Science and Engineering, Ritsumeikan University*

{morita,jun}@cv.mech.eng.osaka-u.ac.jp hild@hilab.osakac.ac.jp shirai@ci.ritsumeikai.ac.jp

**Abstract**—This paper describes a panoramic view-based navigation in outdoor environments. We have been developing a two-phase navigation method. In the training phase, the robot acquires image sequences along the desired route and automatically learns the route visually. In the subsequent autonomous navigation phase, the robot moves by localizing itself by comparing input images with the learned route representation. To be robust to changes of weather and seasons, an object-based comparison is adopted. Our previous method applied a support vector machine (SVM) algorithm to object recognition and localization and exhibited a satisfactory performance but was sometimes sensitive to the variation of the robot's heading. This paper thus extends the method to use panoramic images. By searching the image for the region which matches the model image the most, a new method can considerably improve the localization performance and provide the robot with globally correct directions to move.

**Index Terms**—Outdoor mobile robot, Panoramic vision-based localization, Support vector machine.

## I. INTRODUCTION

Navigation in outdoor environments has been an important problem in mobile robotics. One of the key technologies for reliable navigation is the *localization* of mobile robots. Many approaches have been proposed so far. Distinctions among these approaches can be made with respect to whether an environment map is used, whether robot positions with respect to some scene coordinate frame are sensed and utilized, and whether non-vision sensors like GPS are used. In this paper we take the stance that because GPS-based approaches are known to be unreliable in some situation and map-based approaches often require considerable efforts for creating and maintaining the maps, vision-based techniques are necessary.

Our approach is entirely vision-based and map- and coordinate system-free. During a training run, our robot acquires image sequences along the desired route, automatically learns the route visually, and stores this learned representation of the route for subsequent autonomous navigation. Such two-phase approaches have been proposed, many of which are *view-based* (e.g., [9], [10], [3]). The most difficult part of this approach is finding the most appropriate internal representation (including feature selection) and an appropriate learning algorithm which is capable of generating this internal representation. A simple image comparison-based approach does not suffice because views of objects change much in outdoor environments. It is, therefore, necessary to use an object-based matching [9] or to obtain training data in various weather and seasons.

Another vision-based approach is to fully rely on local visual features such as road boundaries [6], [4], but such features are not always available in outdoor environments.

Many vision-based learning and representation methods are not free from the manual setting of threshold values and parameters. Towards a fully automatic model learning, we have been developing a *support vector machine* (SVM)-based localization method that does not require such manual setting at all [12]. Support vector machine [14] has been successfully applied to many object recognition problems such as 3D object recognition [13], face recognition [5], and pattern matching-based tracking [1].

In our approach, after a feature extraction phase, feature vectors are learned with an SVM algorithm. During the navigation/localization phase, features are extracted in the same way and classified by the trained SVM, producing estimates of robot location along the route. This method is implemented as a two-stage process in which one SVM is employed for general scene feature learning and classification, while another SVM is used for learning and classifying scene locations based on the feature classification results from the first SVM.

Our previous SVM-based localization method exhibited a sufficient localization performance with reasonable robustness to the change of weather and the seasons [12]. However, it used a conventional camera and thus was sensitive to the variation of the robot's heading; a small difference between headings in the training and the localization/navigation phase sometimes caused localization failure. In this paper, therefore, we extend the method to use panoramic images. We search the panoramic image, like the one shown in Fig. 1, for the region which matches the model image the most. This considerably improves the localization performance. We also use the position of the best-matched region to determine globally-correct directions to move.

There have been several panoramic view-based localization methods [8], [11]. Most of them are, however, for indoor environments and probably difficult to apply to outdoor environments with large changes of object views due to the change of weather and seasons.

In the following sections, we first briefly describe our previous SVM-based localization method and its drawbacks. We then explain the extension of the method to the panoramic images and show the improvement of localization performance by experiments. We also show experimental result of autonomous navigation.



Fig. 1. An input panoramic image, taken by LadyBug2 (Point Grey Research Inc.).

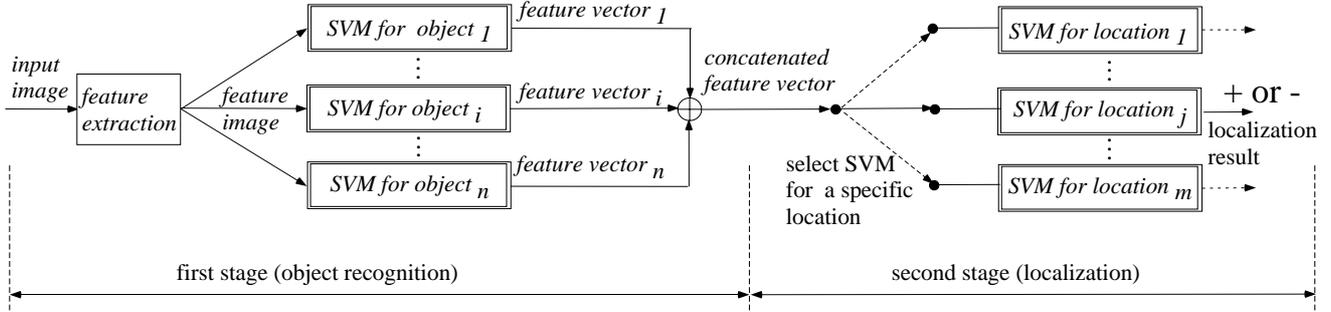


Fig. 2. Two-stage localization using SVMs [12].

## II. SVM-BASED LOCALIZATION

This section briefly reviews our two-stage SVM-based localization method. Refer to [12] for more details.

### A. Two-Stage Localization

Fig. 2 shows the process of our SVM-based localization. The process is divided into two stages: object recognition and localization.

At the first stage, objects in the image are recognized. Image features such as color and edge density of small windows in input images are extracted; a set of such feature values constitutes a *feature image*. This feature image is then sent to a set of SVMs, each of which is trained to recognize objects of a specific class. The output of an SVM is an image representing the location of the detected objects in the image. The output vectors from all SVMs are concatenated to produce the final recognition result. The change of object views due to the change of weather and seasons is handled at this stage, by training SVMs with object images taken under various conditions.

Given this recognition result, robot localization is carried out at the second stage. We train a set of SVMs, each of which can discriminate one given location from the others. The discrimination is based on the recognition results (i.e., the concatenated vectors) from the first stage, not on raw images, so that the localization becomes robust in outdoor environments.

To see if the robot is at a specific location, the input image is tested with the SVM trained for the location. When the robot follows a learned route, for example, the robot switches the SVMs for localization one after another. In this case, only one SVM is used at a time.

We use SVM<sup>light</sup> [7] as the actual SVM software.

### B. Object Recognition

1) *Objects to be Recognized*: We are interested in navigation in urban environments such as our campus. We

use buildings, trees, and the sky, which are relatively large and stationary, as object to be used for localization. We recognize the following four kinds of objects:

- Trees with leaves. Seasonal color changes of leaves are allowed. Labeled as *tree region*.
- Trees without leaves. Only branches are observed. Labeled as *tree region*.
- Sky and building walls that are observed as uniform regions in the image. Labeled as *uniform region*.
- Building windows and boundaries that are observed as strong straight line segments in the image. Labeled as *building region*.

2) *Features Used for Object Recognition*: We use the image of  $304 \times 235$  pixels as the model. Since the above objects exist in the upper-half part ( $304 \times 128$  pixels) of the image, we divide that part into a set of small windows (of  $16 \times 16$  pixels), examine colors and edges within each window, and classify the windows into one of the objects.

The image features we use for object recognition are the following: (1) three components of normalized color ( $r, g, b$ ), (2) edge density, (3) the degree of distribution of edge directions, measured by using *circular statistics* [2], and (4) the degree of existence of line segments, measured by the maximum value of voting in the hough space for the edge points in a window.

The sextuplet of the above feature values are obtained for each window. An input image is thus converted into a  $19 \times 8$  array of the sextuplet. This array, called a *feature image*, is the input to the SVMs for object recognition (see Fig. 2). SVMs for *trees without leaves* and *building region* use all components while those for the other objects use the first four components for recognition.

3) *Training SVMs for Object Recognition*: We use one SVM for each object class. In order to collect training data, we examined image data captured on our campus in various annual seasons and under various weather conditions, and

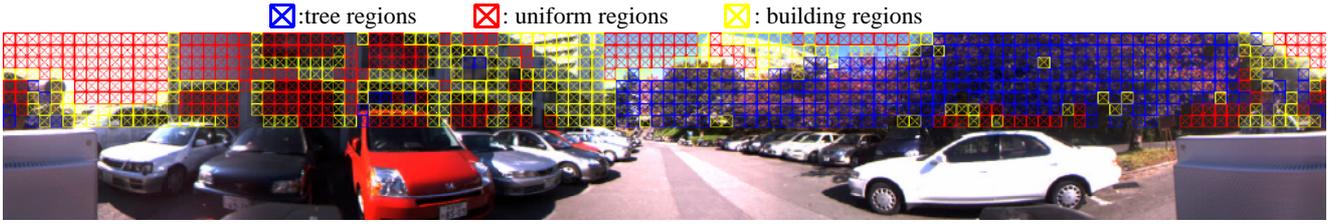


Fig. 3. Recognition results.

manually selected, for each object class, about 300 windows for which only the single object class was present per window. These windows were then converted into the sextuplet of feature values. We finally have four sets of sextuplets for four object classes.

In order to train one SVM per class, we use all sextuplets (or quadruplets) of the corresponding set as positive samples, and randomly select the same number of negative samples from all windows not containing positive samples. We use the SVM with RBF kernel ( $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma\|\mathbf{x}_1 - \mathbf{x}_2\|)$ ,  $\gamma = 50$ ) for object recognition. The time for SVM learning using 600 samples is less than 0.2 [s] for one object class.

Each SVM receives a sextuplet of feature values and returns value one (if the output is positive) or zero (otherwise). Since the size of feature images is  $19 \times 8$ , the SVM produces a 152-dimensional 0-1 vector, called a *feature vector* (see Fig. 2).

4) *Recognition Results*: Fig. 3 shows the recognition result for the image shown in Fig. 1. Each block with  $\times$  mark indicates the recognition result (tree, uniform, or building) for a window. The performance of object recognition is comparable to our previous method [9] which uses many parameters and thresholds to be tuned manually.

### C. Localization

The second stage performs localization by SVMs using the object recognition results of the first stage (see Fig. 2). The first stage outputs three feature vectors (152-D 0-1 vectors) because we have three kinds of labels, *tree*, *uniform*, and *building* regions. We concatenate the vectors into one 456-D 0-1 vector and use it as the input to the SVMs for localization (or 912-D vector if we use both the front and the back region for localization, see Sec. V-B).

1) *Generating Training Data for SVM learning*: We prepare one SVM for each specific location, set along the robot's route. Each SVM is trained by declaring the data taken near the location as positive samples and the data at other locations as negative ones. The detailed process of generating training data is as follows (see Fig. 4).

For each location, we consider the robot is there if it is within a certain distance from the location on the route (called *positive zone*). We use  $n_p$  consecutive images taken inside the positive zone as positive samples. We set *buffer zones* before and after the positive samples, and pick up  $n_n$  images from the remaining frames in regular intervals for negative samples. These positive and negative samples are used to train the SVM for the location. For other locations, we perform the sample selection and learning in the same manner. We use the linear SVM for localization.

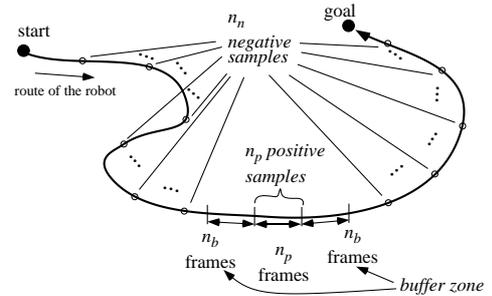


Fig. 4. Making training data for localization SVM.

In the experiments described below, we moved the robot at 0.8 [m/s] while taking panoramic images every 1.5 [s]. As a result,  $n_p$  becomes six (this corresponds to a movement of about 7 [m]) and we set  $n_n = 50$ . For this set of samples, the time for SVM learning for one location is about 0.24 [s].

2) *Localization by SVM*: An SVM outputs positive values if the input is judged as a positive sample. To see if the robot is at a given location, we give the concatenated feature vector, generated from the current input image, to the SVM for that location and see if its output is positive. We also use the output value itself, what we call *SVM score*, to see how near the robot is to the location; SVM score is actually a signed distance to the separating hyperplane.

### III. DRAWBACKS OF THE PREVIOUS METHOD

We used a conventional camera with about  $50^\circ$  field-of-view in the previous paper [12]. This made the method be sensitive to the variation of the robot's heading. On a usual road with a limited width or lanes, the heading of the robot is almost constant as long as it is properly controlled to follow the road, and thus the difference between the captured part of the scene in the training and the localization/navigation phase is limited. When the robot moves in a wide space such as a parking space, however, the heading variation may become large, and this may cause a large amount of degradation of localization performance because SVM-based recognition methods tend to be weak to image shifts.

Fig. 5 shows an example of localization failure. The graph in the figure shows the change of SVM score for the test image sequence, which was taken on the route shown in Fig. 6, against a location model on path segment (f)-(g) in the figure. The top-left image is one of the images at the location. The top-right image is the one which gets the

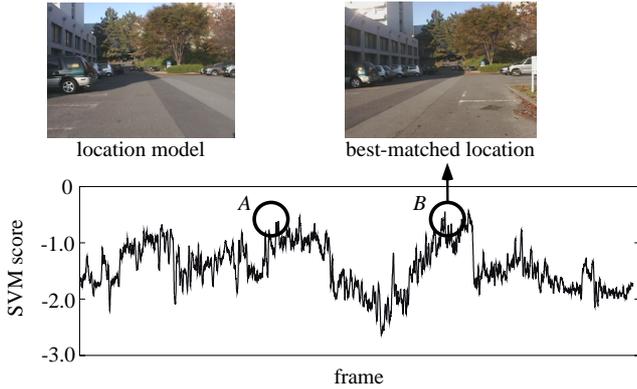


Fig. 5. Localization failure caused by using a conventional camera.

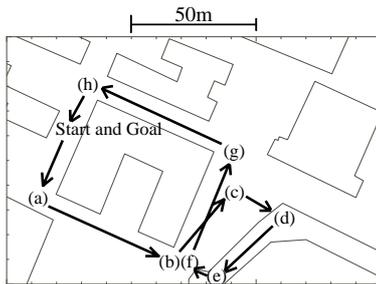


Fig. 6. The route used for experiment.

highest SVM score. The robot passed the location twice and, at the corresponding frames (around *A* and *B* in the figure), the score is higher than others; however, the maximum score is negative (i.e., not recognized correctly) and the peaks are not clear.

In addition, the method using a conventional camera can only determine if the robot is at a specific location but cannot provide information to direct the robot.

#### IV. LOCALIZATION USING PANORAMIC IMAGES

##### A. Search for the Best-Matched Region

We search the input panoramic image for the region which matches the model image the most. The size of the panoramic image is  $1800 \times 235$  pixels. We horizontally move a region of  $304 \times 128$  pixels, which has the same size as the model images, along the upper half of the panoramic image, and calculate the concatenated feature vector at each position. We then calculate the SVM score for each feature vector and choose the highest score as the score of this panoramic image.

The graph in Fig. 7 shows the change of the *maximum* SVM score for the same test run and the same location model shown in Fig. 5. This graph shows two very sharp positive peaks over. The image in Fig. 7 shows the panoramic image at the second peak frame with the best-matched region on it.

##### B. Evaluation of Localization Performance

We use the following two evaluation criteria.

(1) *Recognition ratio*: the ratio of numbers of locations that are correctly recognized by the SVMs in charge of

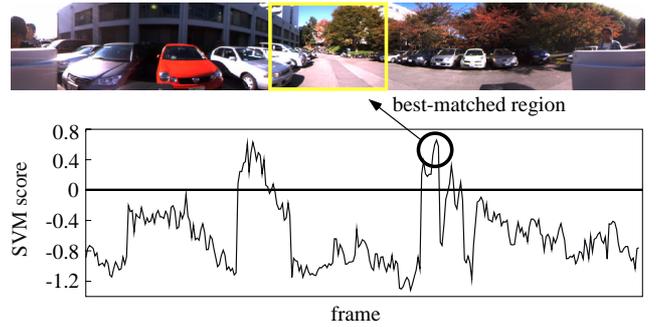


Fig. 7. Effect of using a panoramic camera.



Fig. 8. Samples of training and test images.

the locations versus the total number of locations. This applies to the case where the robot verifies whether it is on a predicted location (i.e., position tracking).

(2) *Highest-score ratio*: the ratio of the number of locations at which the positive *and* the highest scores are obtained by the SVMs in charge for the locations versus the total number of locations. This applies to the case where the robot has to localize itself without any prior knowledge (i.e., global localization).

We compare the proposed method with our previous methods for the route shown in Fig. 6. The training (model) image set was obtained on Nov. 11, 2005 at 11am (sunny); the test image set for the panoramic camera was obtained on Dec. 28, 2005 at 4pm (cloudy). Fig. 8 shows a pair of images, one from the training set and the other from the test set, taken at almost the same location. We can observe view changes at many places. The most significant change is the tree at the center; it has leaves in the model image but loses them in the test image.

We selected 50 locations on the route and examined the localization performance based on the above criteria. Table I summarizes the result of comparing the new method with the SVM-based method using a conventional camera [12] and the method using hand-crafted object models [9]. The new method considerably outperforms the previous method using a conventional camera. The method using hand-crafted object models measures how well the regions of each object are matched between the learned and the test images, and decides the success of matching using a threshold for the measured value. That method uses a relatively loose threshold for a high recognition ratio at the cost of a lower highest-score ratio. The new SVM-based method using a panoramic camera also outperforms that method especially in the highest-score ratio, without using

TABLE I  
COMPARISON OF LOCALIZATION METHODS.

	Recognition ratio	Highest-score ratio
SVM (panoramic)	96%	96%
SVM (conventional)[12]	88%	78%
Hand-crafted models [9]	95%	57%

any parameters and threshold values to be adjusted.

## V. DETERMINING TARGET DIRECTION USING PANORAMIC IMAGE

### A. Use of the Position of the Best-Matched Region

Using panoramic images has another advantage that the direction to move (*target direction*) can also be determined from the position of the best-matched region.

This target direction determination can also apply to the case of turning corners. If we use a conventional camera, we need to pan the camera at corners so that it can capture the view after turning to determine the timing of starting turning [9]. Since the panoramic image already contains such a view, the robot can determine which direction to turn without moving the camera. Fig. 9 shows a result of determining the target direction. The steep peak of SVM score indicates the target direction.

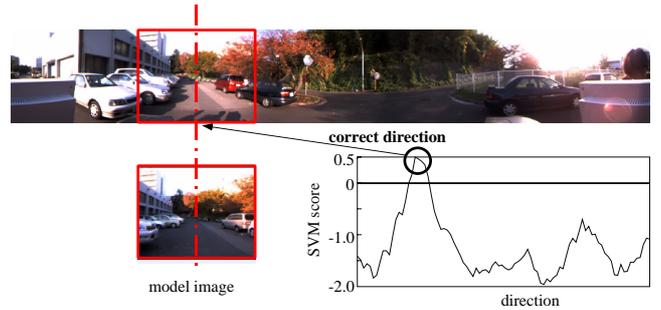


Fig. 9. Determining the target direction.

### B. Use of Front and Back Images

The accuracy of the target direction depends on the distributions of recognized objects in the image. When the region around the target direction is occupied by objects of a kind, many directions may have high SVM scores. The dashed line in the graph in Fig. 10 shows such a case; its peak is not sharp and the ambiguity in determining the target direction is large.

To cope with this, we use a pair of regions, one is for front and the other is for back of the robot. Two regions are placed so that they are exactly 180 degrees apart from each other as shown in Fig. 11. Each region outputs a 456-D feature vector (see Sec. II-C) and a concatenated 912-D feature vector from two vectors is used for localization and direction determination. The solid line of the graph in Fig. 10 shows the result of determining the target direction; the SVM output now exhibits a sharp peak.

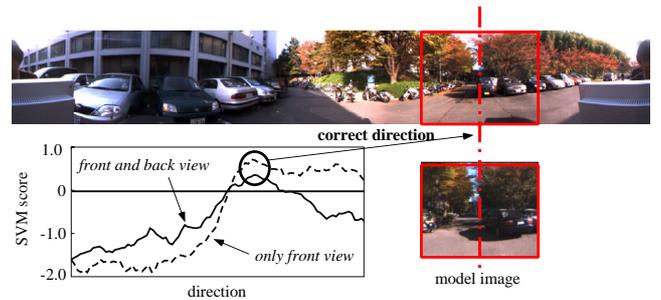


Fig. 10. Resolving ambiguity in determining the target direction.

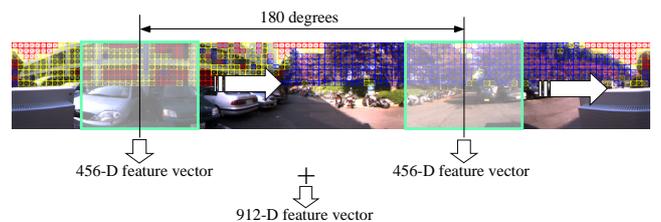


Fig. 11. Use of a pair of windows for front and back directions.

## VI. NAVIGATION EXPERIMENTS

We performed navigation experiments using our mobile robot (see Fig. 12).

### A. Testing the Navigation Ability on a Long Route

We examined whether the proposed method can localize the robot and can indicate the globally-correct directions using the test image set mentioned above. The location models are switched one after another as the robot moves by 7[m]. The processing time for one image including image processing, localization, and target direction determination is about 0.9[s] with a PC (Pentium 4 (2.2GHz)).

Fig.13 shows images with determined target directions at several positions on the route. The total number of frames captured is 322. The proposed method succeeded in localization (i.e., output positive SVM scores) for about

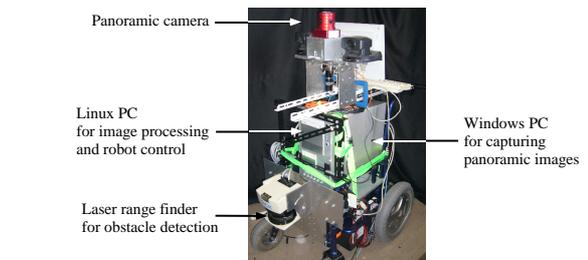


Fig. 12. Our mobile robot.

84% of the frames. The average length of consecutive failure frames is 2.4; such an occasional failures can be handled by deferring the localization decision for a few frames. In addition, the method succeeded in determining correct target directions, which are within  $\pm 15$  [deg] of the center of the model images, for about 98% of the frames.

Fig. 14 is the outputs of all location models for all input images acquired in the test run; only positive outputs are drawn. The figure shows that since the *support* of each output is localized within a short period of time, it seems possible to automatically switch the location models using the output of the models of neighboring locations.

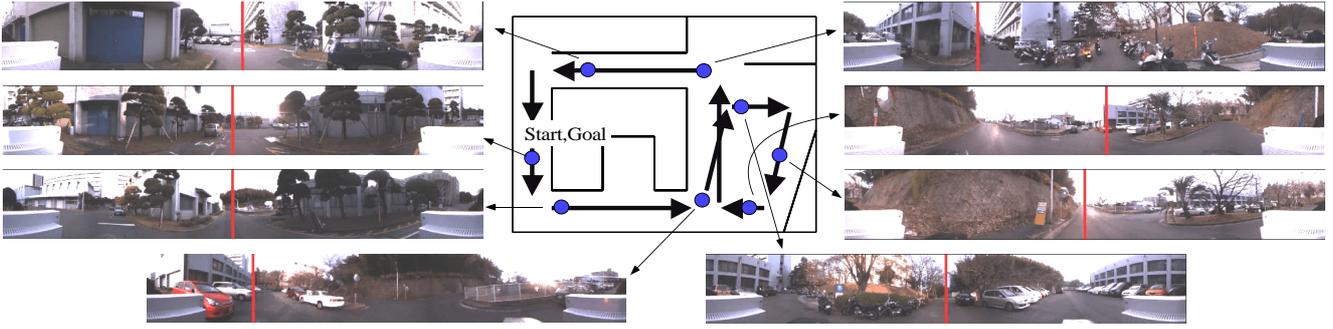


Fig. 13. Localization and direction determination results on the route shown in Fig. 6. Red lines indicate the determined directions.

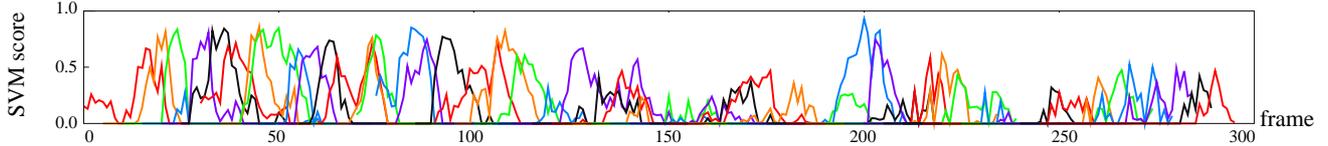


Fig. 14. Outputs of all location models for all input images (300 frames). Six colors (red, orange, green, blue, purple, and black) are cyclicly assigned to the output of models.



Fig. 15. Navigation experiment.

### B. Autonomous Navigation on a Short Route

We performed an autonomous navigation experiment on a wide but short route. Fig. 15 shows snapshots of the navigation. The robot successfully moved by about 100 [m] using the proposed navigation method. This result shows the potential effectiveness of the method.

## VII. CONCLUSIONS AND DISCUSSION

This paper describes a novel navigation method in outdoor environments based on support vector learning with panoramic images. The method employs a two-stage localization process in which one SVM is employed for general scene feature learning and classification, while another SVM is used for learning and classifying scene locations based on the feature classification results from the first SVM. By searching the panoramic image for the region which matches the model, the method can cope with the variation of the robot's heading in the training and the localization/navigation phase, and thus realizes a considerable improvement of the localization performance compared to the previous method. The method can also provide the robot with globally-correct target directions.

Although the method can guide the robot globally, it is necessary to locally guide it for safe navigation by using, for example, road detection methods. We are currently working on developing a method of detecting traversable

regions; a future work is to combine this method with the proposed SVM-based method to realize a long navigation.

## REFERENCES

- [1] S. Avidan. Support vector tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, pp. 1064–1072, 2004.
- [2] E. Batschelet. *Circular Statistics in Biology*. Academic Press Inc., London, 1981.
- [3] 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.
- [4] J.D. Crisman and C.E. Thorpe. Scarf: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automat.*, Vol. 9, No. 1, pp. 49–58, 1993.
- [5] G. Guo, S.Z. Li, and K. Chan. Face recognition by support vector machines. In *Proceedings of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 195–201, 2000.
- [6] H. Ishiguro, K. Nishikawa, and H. Mori. Mobile robot navigation by visual sign patterns existing in outdoor environments. In *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 636–641, 1992.
- [7] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. The MIT Press, 1999.
- [8] M. Jorgan and A. Leonardis. Robust localization using panoramic view-based recognition. In *Proceedings of 15th Int. Conf. on Pattern Recognition*, pp. 136–139, 2000.
- [9] 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.
- [10] 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.
- [11] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, Vol. 47, pp. 251–267, 2004.
- [12] 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.
- [13] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, pp. 637–646, 1998.
- [14] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.