

# Robust View Matching-Based Markov Localization in Outdoor Environments

Jun Miura and Koshiro Yamamoto  
Department of Information and Computer Sciences  
Toyohashi University of Technology

**Abstract**—This paper describes a view-based localization method in outdoor environments. An important issue in view-based localization is to cope with the change of object views due to changes of weather and seasons. We have developed a two-stage SVM-based localization method which exhibits a high localization performance with few parameter tunings. In this paper, we extend the method in the following two ways: (1) adding new object models and visual features to deal with various urban scenes and (2) introducing a Markov localization strategy to utilize the history of movements. The new method can achieve a 100% localization performance in an urban route under a wide variety of conditions. The comparison with local feature-based methods is also discussed.

**Index Terms**—View-based localization, Markov localization, outdoor navigation, support vector machine.

## I. INTRODUCTION

Navigation in outdoor environments is one of the active research areas in mobile robotics. One of the key technologies for reliable navigation is the *localization*. Many approaches have been proposed so far. Among them, vision-based ones have been widely studied for their applicability to various environments including the one where GPS signal is not always properly available.

This paper deals with *view-based localization*. Its typical approach is as follows. During the training run, a robot (or vehicle) acquires an image sequence along a route. In the subsequent autonomous run, it compares input images with learned ones to localize itself. 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.

An important issue in view-based localization is how to cope with the change of object views due to changes of weather and seasons. Obviously, simple image comparison-based approaches do not suffice. It is, therefore, necessary to use an object-based matching [7] or to obtain training data in various illumination conditions [2].

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 requires very few such manual settings [11], [12]. Our method is implemented as a two-stage process in which a set of SVM's (*object recognition SVM's*) is employed for general scene feature learning and classification, while another set of SVM's (*localization SVM's*) is used for learning and classifying scene locations based on the feature classification results from the first set of SVM's.

In this paper, we extend our previous method in the following two ways. First, we introduce new object models and visual features to deal with driving in urban scenes where a more variety of objects and conditions need to be considered. Second, for more reliable localization, we introduce the Markov localization framework to consider the history of movements. The new method can achieve a very high localization performance in various real situations.

View-based localization can be interpreted as finding the most similar image to the current input image from a set of images learned during the training phase. Many object recognition or image retrieval technologies can thus be adopted. Local image descriptors such as SIFT [9] are often used [4], [8], [14], [16], but they are inherently weak to a large change of weather and seasons [16]. Other representations, which are robust to a certain kind of view changes, have been used for localization such as local invariant feature histograms of images [17] or Gaussian mixtures in a learned image manifold [13]. These representations are not intended to cope with long-term changes of object views.

A single observation is often insufficient for reliable localization. Probabilistic approaches like Markov localization or Monte-Carlo localization [15] are thus sometimes adopted in view-based localization [10], [3], [17]. These works, however, deal with indoor localization with a relatively small change of object views.

The rest of this paper is organized as follows. Sec. II briefly explains our two-stage SVM-based localization method. Sec. III describes the new set of objects and features to be used in the proposed method along with object recognition experiments. Sec. IV describes the training and the usage of localization SVM. We also describe comparison results with a simple image matching-based method and a SIFT-based one. Sec. V explains how to combine our SVM-based method into the Markov localization framework. Sec. VI shows the results of experimental evaluations including the comparison with a SIFT-based Markov localization. Sec. VII concludes the paper and discusses future work.

## II. TWO-STAGE SVM-BASED LOCALIZATION

Fig. 1 illustrates the concept of our two-stage SVM-based localization [11], [12]. At the first stage, objects in the image are recognized. Image features such as color and edge density of small windows are extracted. A set of such feature values is the input to *object recognition SVM's*, each of which is responsible for recognizing an object. The change of object views due to changes of weather and seasons is handled at

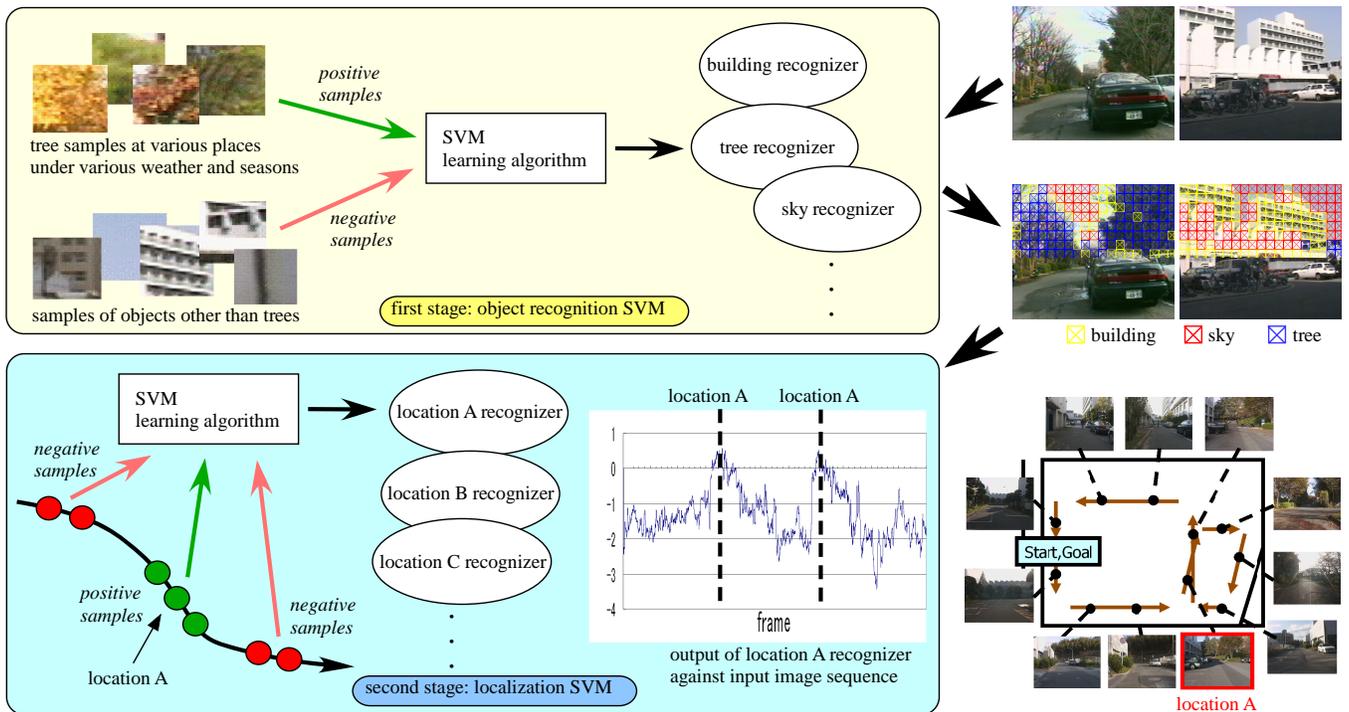


Fig. 1. Two-stage SVM-based localization.

this stage, by training SVM's with object images taken under various conditions.

The second stage is for localization. The recognition result from the first stage is input to *localization SVM's*. Each localization SVM is trained for discriminating one given location from the others. The figure shows the case where each localization SVM is used independently. In this paper, the Markov localization framework is introduced to increase the localization performance.

Using SVM's has an advantage that no thresholds are necessary for classification. Discriminating surfaces are automatically generated from positive and negative samples, and the outputs of an SVM directly indicate classifications. We use SVM<sup>light</sup> [6] as the actual SVM software.

### III. OBJECT RECOGNITION SVM

#### A. Objects to be recognized

We use objects which are relatively large and stationary for localization. We here deal with urban scenes as shown in the left column of Fig. 2. In addition to the four kinds of objects used in campus environments [11], [12], we use power lines which commonly appear in urban traffic scenes in Japan; as a result, we use the following five kinds of objects:

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

- Power Lines which usually appear in the sky. Labeled as *power line region*.

#### B. Features used for object recognition

We use images of  $320 \times 240$  pixels. Since the above objects exist in the upper-half part ( $320 \times 120$  pixels) of images, we divide that part into a set of small windows (of  $16 \times 16$  pixels), examine colors and edges within each window, and determine if the above objects exist in the window using object recognition SVM's. The windows are arranged into a  $20 \times 8$  array in that part of the image.

Table I summarizes the region name (label) and image features to use for each object class. We use two object classes for trees but use the same label ("tree"). In general, using more features does not necessarily result in better recognition performance. We tested several combinations of features for each object and selected the best one.

We use the following image features:

- $(r, g, b)$ : Normalized color, averaged over the window.
- $f_{density}$ : Edge density, calculated as the ratio of edge pixels in a window.
- $f_{hough}$ : Maximum value of voting in the hough space for the edge points in a window, used for assessing the existence of strong line segments.
- $f_{distrib}^{edge}$ : Degree of distribution of edge directions, calculated using the circular statistics [1].
- $f_{distrib}^{int}$ : Variance of intensity values in a window.

#### C. Training and using object recognition SVM's

We use one SVM for each object class. The training data for an object class was collected as follows. We examined image data captured on our test route in various annual

TABLE I  
OBJECT CLASSES, REGION NAMES, AND IMAGE FEATURES USED.

object class	region name	image features used				
		$(r, g, b)$	$f_{density}$	$f_{hough}$	$f_{distrib}^{edge}$	$f_{distrib}^{int}$
sky, building side walls	<i>uniform region</i>	✓	✓			
trees with leaves	<i>tree region</i>	✓	✓	✓		
trees without leaves	<i>tree region</i>	✓	✓	✓	✓	
building windows and boundaries	<i>building region</i>	✓	✓	✓	✓	
power lines	<i>power line region</i>	✓	✓	✓	✓	✓

seasons, under various weather conditions, and at various times, and manually selected about 8,000 windows for which only the object class was present per window. The feature sets of the object class, indicated in Table III-A, are used as positive samples and those of almost the same number of randomly selected windows not containing the object class at all are as negative samples. We use the SVM with RBF kernel ( $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma\|\mathbf{x}_1 - \mathbf{x}_2\|^2)$ ,  $\gamma = 50$ ) for object recognition.

Each object recognition SVM receives a set of feature values and returns one when the output is positive (i.e., the corresponding object exists), and returns zero otherwise.

#### D. Object recognition results

The right column of Fig. 2 shows the recognition results. Marks indicate the recognition result (tree, uniform, building, or power line) for respective windows. Windows with yellow marks are the ones without any recognized objects. When several objects are found at a window, the mark of the highest SVM output is drawn. The results show that our object recognition works well with a variety of conditions and seasonal object view changes.

We tested object recognition SVM's for about 6000 test samples, collected from the windows different from those which used for collecting training samples. The averaged recognition rates are 98.4% for uniform regions, 89.3% for tree with leaves, 81.2% for tree without leaves, 76.5% for building regions, and 91.7% for power line regions. The time for processing one image is about 0.32 [s], 0.1 [s] of which is for feature extraction and 0.22 [s] for applying all object recognition SVM's to all windows, using Core2Duo 3GHz.

## IV. LOCALIZATION SVM

The second stage of the proposed method is to determine the location using localization SVM's. This section explains the previous localization method [11], [12] which uses each localization SVM independently and is a component of our new view-based Markov localization. We also describe comparison results with a simple image matching-based method and a SIFT-based one.

#### A. Input to localization SVM

The result of the object recognition stage is the input to localization SVM's. The windows for object recognition are arranged as a  $20 \times 8$  array in the upper part of the image, as mentioned above. To cope with the change of the viewing direction at the times of training and localization, however, we use a region consisting of  $18 \times 6$  windows in object recognition stage (see Fig. 3). In training, we use the central



Fig. 2. Example recognition results.

part of the image, while in recognition for localization, we allow the region to move to three positions both vertically and horizontally, thereby setting nine positions, and take the one which results in the best output of localization SVM.

The number of windows is 108. One object recognition SVM thus produces a 108-dimensional 0-1 vector and the first stage outputs four of such vectors because we have four kinds of labels: *uniform*, *tree*, *building*, and *power line*. We concatenate the vectors into a single 432-dimensional 0-1 vector and use it as the input to each SVM for localization.

#### B. Training localization SVM

We prepare one SVM for each specific location, set along the route. Each SVM is trained by declaring the data taken near the location as positive samples and the data at other

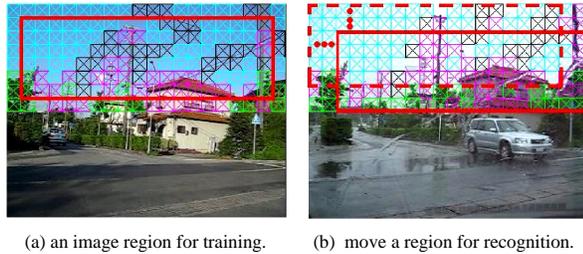


Fig. 3. Regions for training and recognition.



Fig. 4. Route for the experiments.

locations as negative ones. To see if the robot is at a given location, we give the concatenated vector obtained from the current input image to the SVM for that location and see if the output is positive (see Fig. 1).

A set of locations for which SVM's are trained is determined by selecting locations with a regular interval. Assuming that the vehicle moves at a constant speed, we select locations for every 30 frames (i.e., two seconds for the image acquisition rate of 15 frames/sec). For the route of about 3.2 [km] used for experiments (see Fig. 4), we obtained 198 locations; the average interval is about 16 [m].

As the positive samples for the SVM for a location, we use 30 images centered at the selected frame for that location. Since it is undesirable to use images taken while the vehicle is stopping or turning a corner, we detect and eliminate such cases using the optical flow pattern of the image.

### C. Localization experiments

We collected data on the route shown in Fig. 4. We can get about 7000 images during each run on this route. We obtained an image sequence for training on Jun. 20, 2007 at 5pm (sunny). We also obtained image sequences for testing on Jun. 20, 2007 at 5:10pm (taken on the second run of the day, just after the first run for training data), on Jun. 22 at 5pm (rainy), on Jul. 24, 2007 at 7pm (sunny), and on Oct. 29, 2007 at 2pm (cloudy).

Fig. 5 shows the results for the three locations indicated in Fig. 4. For each location, the input image, the outputs of all localization SVM's, and the image corresponding to the

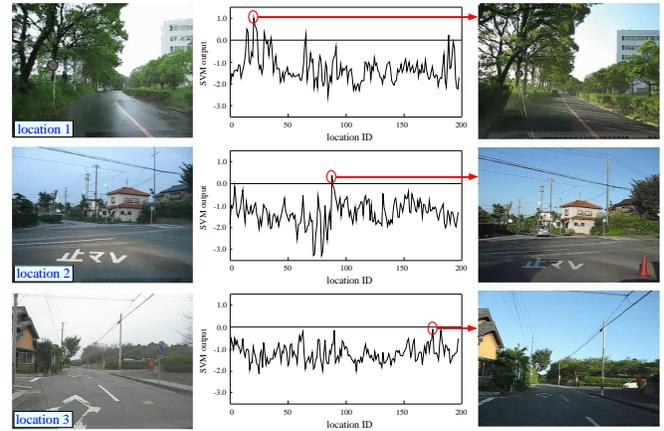


Fig. 5. Localization results for the three locations shown in Fig. 4.

localization SVM with the maximum output are shown. The weather conditions for taking the input images at locations 1, 2, and 3 are rainy, sunny but close to evening, and cloudy, respectively. Although the correct localization SVM outputs the best value in all cases, there are multiple localization SVM's which output positive values for location 1 and the maximum output is negative for location 3; our previous method, which use each SVM independently, may fail in such cases. To cope with these problems, we will adopt the Markov localization framework in the next section.

We then show some quantitative evaluation results. For evaluation, we use the following two criteria:

- *Recognition rate*: the ratio of the numbers of locations that are correctly recognized by the SVM's in charge of 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).
- *Highest-score rate*: the ratio of the number of locations at which the positive *and* the highest scores are obtained by the SVM's 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).

The results are summarized in Table II. In spite of large object view changes, high recognition rates are achieved. This will be further improved by introducing the Markov localization framework in Sec. V.

### D. Comparison with a direct image matching-based localization

We compare our two-stage SVM-based localization method with a direct image matching-based localization one. This *direct* method calculates the averaged normalized color for each window in the image and uses the sum of the Euclidean distances between the colors in the input and the training image as the measure of dissimilarity. We set a threshold for the distance for judging if the two images are taken at a same location. The results shown in Table III illustrate how drastically the performance of the direct image matching-based method is degraded as the conditions changes from that for the training.

TABLE II  
SINGLE SVM-BASED LOCALIZATION RESULTS.

Date, Time, Condition	Recognition rate	Highest-score rate
Jun. 20, 2007, 5:10pm, sunny	95.7%	79.6%
Jun. 22, 2007, 5pm, rainy	86.5%	59.4%
Jul. 24, 2007, 7pm, sunny	81.3%	56.1%
Oct. 29, 2007, 2pm, cloudy	72.7%	43.2%

TABLE III  
DIRECT IMAGE MATCHING-BASED LOCALIZATION RESULTS.

Date, Time, Condition	Recognition rate	Highest-score rate
Jun. 20, 2007, 5:10pm, sunny	78.6%	60.2%
Jun. 22, 2007, 5pm, rainy	23.2%	13.1%
Jul. 24, 2007, 7pm, sunny	6.1%	3.8%
Oct. 29, 2007, 2pm, cloudy	7.8%	7.0%

### E. Comparison with a SIFT-based localization

Many local image feature-based localization methods use SIFT [9], as mentioned above, because of its robustness to changes of orientation, scale, and illumination condition.

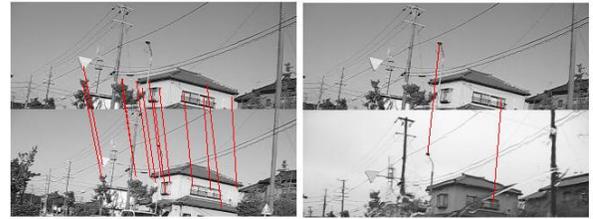
We here test a simple SIFT-based localization method, which considers that the vehicle is at some location if the number of matches of SIFT features in the input image and the learned image at that location exceeds a threshold. To determine the threshold, we compared two image sequences taken at almost the same time, and examined the number of matches for correct pairs. The averaged number of matches was 32; we use its half value, 16, as the threshold.

Fig. 6 shows typical results of SIFT extraction and matching. The left pair is the matches between the images taken on the same day at almost the same time, while the right one is the matches between the images taken under different weather conditions. Local image features are weak to such changes of conditions. Table IV summarizes the performance of the simple SIFT-based localization. A very high performance is achieved when the conditions for training and testing are sufficiently similar, but the performance is degraded rapidly when the condition is largely different from that at the training run.

An interesting observation from the second and the third rows of Tables III and IV is as follows. In the rainy condition (second row), the color is similar but fine textures are wiped out, while in the evening (third row), such textures are preserved to some extent but color (especially, intensity part) changes. As a result, the image matching-based method are relatively strong for rain but weak for evening; the SIFT-based one has an almost opposite characteristic.

## V. VIEW-BASED MARKOV LOCALIZATION

Use of movement history is an effective way of increasing the localization performance. By predicting a possible set of locations using the history, we can cope with occasional localization failures. Moreover, we can reduce the calculation cost by limiting the set of localization SVM's to be tested. Our localization problem is to select the best location among the set of locations for which localization SVM's are trained. The Markov localization framework [5] fits well to this problem.



(a) sunny vs. sunny.

(b) sunny vs. rainy.

Fig. 6. SIFT matches in two different combinations of conditions.

TABLE IV  
SIMPLE SIFT-BASED LOCALIZATION RESULTS.

Date, Time, Condition	Recognition rate	Highest-score rate
Jun. 20, 2007, 5:10pm, sunny	86.7%	85.7%
Jun. 22, 2007, 5pm, rainy	2.2%	2.2%
Jul. 24, 2007, 7pm, sunny	32.1%	30.1%
Oct. 29, 2007, 2pm, cloudy	5.7%	5.7%

### A. Markov localization formulation

We determine the belief  $Bel(l)$  of location  $l$ ;  $l$  is a location in the set of all locations  $\mathcal{L}$ . The Markov localization is formulated by the following two equations:

$$\text{(prediction)} \quad Bel\hat{e}(l) = \sum_{l'} P_t(l|l') Bel(l'), \quad (1)$$

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

where  $Bel\hat{e}(l)$  and  $Bel(l)$  are the probabilities that the robot is at location  $l$  at the prediction and the estimation step, respectively,  $P_t(l|l')$  is the transition probability from location  $l'$  to  $l$ , and  $P(o|l)$  is the likelihood of location  $l$  given observation  $o$ , and  $\alpha$  is the normalization constant.

### B. Transition probability model

We use the transition model shown in Fig. 7 assuming that the vehicle never moves backwards. Transition probabilities between nodes (i.e., locations) are determined empirically by considering the vehicle average speed and its possible deviations. We currently use 0.5, 0.4, and 0.1 for the transition from  $i$  to  $i$ , from  $i$  to  $i+1$ , and from  $i$  to  $i+2$ , respectively.

### C. Probabilistic model of observation

We then construct a probabilistic model of observation for calculating the likelihood of each location. Our previous method, which uses each location SVM independently, considers that the vehicle is at some location when the output of the corresponding SVM is positive. Even when the output is negative, however, the probability of being there is not zero, as shown in the bottom row of Fig. 5. We therefore define a function of the output value of localization SVM, using the

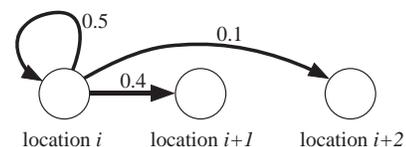


Fig. 7. Transition model.

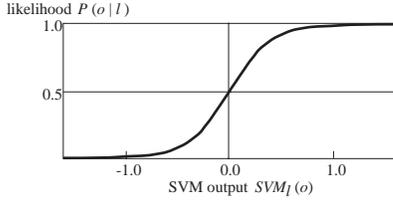


Fig. 8. A sigmoid function for calculating likelihood.

sigmoid function as follows:

$$P(o|l) = \frac{1}{1 + e^{-k SVM_l(o)}}, \quad (3)$$

where  $SVM_l(o)$  is the output of localization SVM for location  $l$  with observation  $o$  and  $k$  is a parameter (currently, 5). Fig. 8 shows the sigmoid function used.

#### D. View-based Markov localization algorithm

The algorithm of our view-based Markov localization is as follows.

- 1) Determine the initial distribution of locations according to one of the following three cases:
  - The current location is known: give 1.0 to that location.
  - An approximate location is known: put some distribution around that location.
  - No information is available on the current location: use the uniform distribution over  $\mathcal{L}$ .
- 2) Repeat the following steps:
  - a) Calculate the prior probability  $Bel(l)$  using eq. (1).
  - b) Calculate likelihood  $P(o|l)$  for the nodes with non-zero probabilities.
  - c) Update the belief  $Bel(l)$  using eq. (2).
  - d) Set zero to the nodes with very small ( $< 0.005$ ) probabilities (pruning).
  - e) Normalize the beliefs.

## VI. EXPERIMENTAL RESULTS

### A. Results of the view-based Markov localization

Fig. 9 shows the results of our view-based Markov localization for the locations shown in Fig. 5. The highest probability nodes indicate the correct locations in all cases. Fig. 10 illustrates another merit of using Markov localization. A sequence of three pairs of the input images and the corresponding beliefs are shown in the figure. Since the scenes around the first and the second images are not distinctive among nearby scenes and the distributions are thus widely spread. As unique objects (horizontal power lines and poles on the right) appear in the third image, however, the distribution quickly converges.

Table V summarizes the performance of the proposed localization method in the two criteria. We consider that a location is correctly recognized if that location is in a set of nodes with non-zero probability and the recognition rate is calculated accordingly. Comparing with the result of the previous method in Table II, the performance of the proposed method is outstanding. The total processing time including image processing is about 0.78 [s]; this is short enough to run on-line.



Fig. 9. Several results of view-based Markov localization.

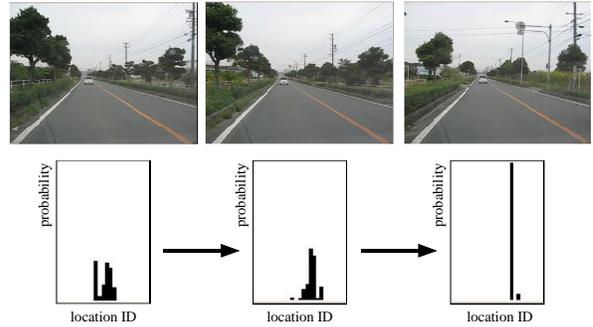


Fig. 10. Convergence of the probability distribution of locations.

### B. Comparison with a SIFT-based Markov localization

We also test a SIFT-based Markov localization. We define the likelihood function using the following sigmoid function:

$$P(o|l) = \frac{1}{1 + e^{-(n-n_0)}}, \quad (4)$$

where  $n$  is the number of matches and  $n_0$  is a parameter. We set  $n_0 = 8$ ; this means that the probability becomes 0.5 when the number of matches is 8, a half of the threshold value used in Sec. IV-E. We tested several values for  $n_0$  but did not get significant changes of performance. In order to compare the view-based and the SIFT-based Markov localization in terms of image features used, only the likelihood functions are different for both methods.

Table VI summarizes the results of the SIFT-based Markov localization. Adopting the Markov localization framework greatly increases the performance even when the number of matches is very small. Comparing Tables V and VI, our method performs better in rainy and cloudy conditions where the object views tend to change largely; local image feature-based methods seem weaker in such conditions.

### C. Recovery from a completely lost situation

When the starting position is not known, we set the uniform distribution over the set of locations  $\mathcal{L}$ . Fig. 11 shows a recovery from such a completely lost situation (global localization).

We then compared the view-based and the SIFT-based Markov localization method in terms of convergence from

TABLE V  
VIEW-BASED MARKOV LOCALIZATION RESULTS.

Date, Time, Condition	Recognition rate	Highest-score rate
Jun. 20, 2007, 5:10pm, sunny	100.0%	93.6%
Jun. 22, 2007, 5pm, rainy	100.0%	81.3%
Jul. 24, 2007, 7pm, sunny	100.0%	79.0%
Oct. 29, 2007, 2pm, cloudy	100.0%	79.2%

TABLE VI  
SIFT-BASED MARKOV LOCALIZATION RESULTS.

Date, Time, Condition	Recognition rate	Highest-score rate
Jun. 20, 2007, 5:10pm, sunny	100.0%	97.7%
Jun. 22, 2007, 5pm, rainy	100.0%	75.4%
Jul. 24, 2007, 7pm, sunny	99.4%	85.1%
Oct. 29, 2007, 2pm, cloudy	98.4%	68.2%

a completely lost situation. The test data used were the ones taken on Jun. 22, 2007 at 5pm (rainy). Starting from 90 randomly selected locations (but with a belief that all locations are equally possible), we measure the convergence rate and the averaged number of observations needed for convergence. We consider the convergence is achieved if the number of nodes with non-zero probabilities including the correct one becomes less than or equal to five. The results are summarized in Table VII. The necessary numbers of frames are almost the same to both methods, but the convergence rate is much higher in our method. This is probably because the number of SIFT matches is very low and the many spurious locations get comparable evaluations to the correct one. Our SVM-based method have reasonably high success rates even when using a single localization SVM as shown in Table II and this makes the localization robust to the ambiguity of information on the current location.

## VII. CONCLUSIONS AND FUTURE WORK

This paper has described a robust view matching-based Markov localization in outdoor environments. We have extended our previous, two-stage SVM-based method, by introducing new object models with a new set of image features and the Markov localization framework, thereby achieving a high localization performance in typical urban scenes under various conditions. We have also shown that our method outperforms local feature-based ones in coping with large object view changes.

Currently, we deal with the localization *on a route*. We plan to extend to the cases of more complex topology of routes with many junctions. We also plan to cope with the case where vehicle or robots moves in a wide 2D space, where not only the location but also the orientation should be determined.

## REFERENCES

- [1] E. Batschelet. *Circular Statistics in Biology*. Academic Press Inc., London, 1981.
- [2] 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.
- [3] M. Castelnovi, A. Sgorbissa, and R. Zaccaria. Markov-Localization through Color Features Comparison. In *Proceedings of 2004 IEEE Int. Symp. on Intelligent Control*, pp. 437–442, 2004.

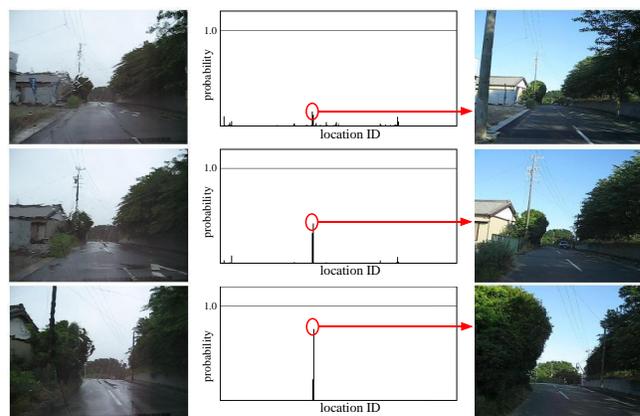


Fig. 11. Recovery from a completely lost situation.

TABLE VII  
COMPARISON OF TWO METHODS IN TERMS OF CONVERGENCE.

Method	Convergence rate	Necessary frames
SVM-based Markov	93.3%	10.7 [frames]
SIFT-based Markov	71.1%	10.0 [frames]

- [4] M. Cummins and P. Newman. Probabilistic Appearance Based Navigation and Loop Closing. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2042–2048, 2007.
- [5] D. Fox, W. Burgard, and S. Thrun. Active Markov Localization for Mobile Robots. *Robotics and Autonomous Systems*, Vol. 25, No. 3-4, pp. 195–207, 1998.
- [6] 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.
- [7] 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.
- [8] J. Klippenstein and H. Zhang. Quantitative Evaluation of Feature Extractors for Visual SLAM. In *4th Canadian Conf. on Computer and Robot Vision (CRV '07)*, pp. 157–164, 2007.
- [9] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.
- [10] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-Based Monte Carlo Localisation with Omnidirectional Images. *Robotics and Autonomous Systems*, Vol. 48, No. 1, pp. 17–30, 2004.
- [11] 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.
- [12] 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.
- [13] F.T. Ramos, J. Nieto, and H.F. Durrant-Whyte. Recognising and Modeling Landmarks to Close Loops in Outdoor SLAM. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2036–2041, 2007.
- [14] C. Slipa-Anan and R. Hartley. Visual Localization and Loop-Back Detection with a High Resolution Omnidirectional Camera. In *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS 2005)*, 2005.
- [15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [16] 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.
- [17] J. Wolf, W. Burgard, and H. Burkhardt. Robust Vision-Based Localization by Combining an Image Retrieval System with Monte Carlo Localization. *IEEE Transactions on Robotics*, Vol. 21, No. 2, pp. 208–216, 2005.