

View Sequence Generation for View-based Outdoor Navigation

Yuki Kaneko and Jun Miura

Department of Computer Science and Engineering, Toyohashi University of Technology

Abstract—This paper describes a method of generating a new view sequence for view-based outdoor navigation. View-based navigation approaches have been shown to be effective but have a drawback that a view sequence for the route to be navigated is needed beforehand. This will be an issue especially for navigation in an open space where numerous potential routes exist; it is almost impossible to take view sequences for all of the routes by actually moving on them. We therefore develop a method of generating a view sequence for arbitrary routes from an omnidirectional view sequence taken at a limited movement. The method is based on visual odometry-based map generation and an image-to-image morphing using homography. The effectiveness of the method is validated by view-based localization experiments.

I. INTRODUCTION

Navigation in outdoor environments has been an important topic in mobile robotics and intelligent transportation systems. Navigation of people has also been gaining a large interest according to a rapid progress of mobile devices. GPS is the most popular localization and navigation method in outdoor but it is not always operational especially near tall buildings. Vision-based localization and navigation is thus a promising alternative or complement in outdoor navigation.

Among various visual navigation approaches, view-based navigation has been shown to be effective [1], [2], [3], [4]. Its typical approach is as follows. In a training phase, a vehicle acquires image sequences along a route. In the subsequent navigation phase, it compares input images with learned ones to localize itself.

An advantage of view-based navigation is that the user only has to take the robot along a route for image sequence acquisition or just give it an image sequence taken at a different time. No other information such as a map of the environment is needed.

This advantage might be, however, a drawback; that is, a view sequence for the route to be navigated is needed

beforehand. This will be an issue especially for navigation in an open space where numerous potential routes exist; it is almost impossible to take view sequences for every possible route by actually moving the robot or walking on it. We therefore would like to develop a method of generating a view sequence on a new route from view sequences taken on different routes (see Fig. 1). Since we cannot know the route for which a view sequence is required in advance and any viewing direction is therefore possible, we use an omnidirectional camera to take various views of objects from a limited number of routes for view acquisition.

Many image-based rendering (IBR) methods have been proposed in computer graphics and computer vision. They can be characterized by the amount of geometry of the scene required [5]. Levoy and Hanrahan [6] developed the light field rendering method which represents the radiance as a function of position and direction in a 4D space. Shum and He [7] proposed a new view synthesis method using concentric mosaics obtained by rotating off-centered cameras. These approaches do not require any geometric information but need many images. The view interpolation method [8] generates novel views from two images with a large number of point matches; this works when two images are taken at close positions and few occluding boundaries exist in the scene.

Several IBR methods use geometric information, which are especially effective for a scene with a wide depth range. 3D warping methods [9] uses 3D point clouds to generate images from arbitrary viewpoints. Sato et al. [10] proposed a method of view generation using a deformable mesh estimated using a dense depth map for an omnidirectional image sequence. Zheng et al. [11] proposed the scanning scene tunnel for compact representation of route scene rendering. These methods require dense 3D information before rendering.

All of the above IBR approaches are intended for generating high-quality images which are to be used for virtual reality or mixed reality and usually require many images and/or much computation for precise registration and adjustments. In contrast, what we need here is a simple and fast method which can provide *an enough level of quality for view-based navigation*.

Our method uses geometric information but requires only a set of sparse feature points in the scene and is based on a window-wise morphing using homography, which has a nice correspondence to the window-wise image classification of our view-based navigation method [3]. We make a map of the environment based on visual odometry and triangulation, and

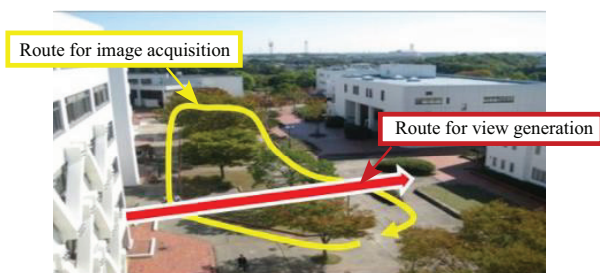


Fig. 1. View sequence generation for a new route.

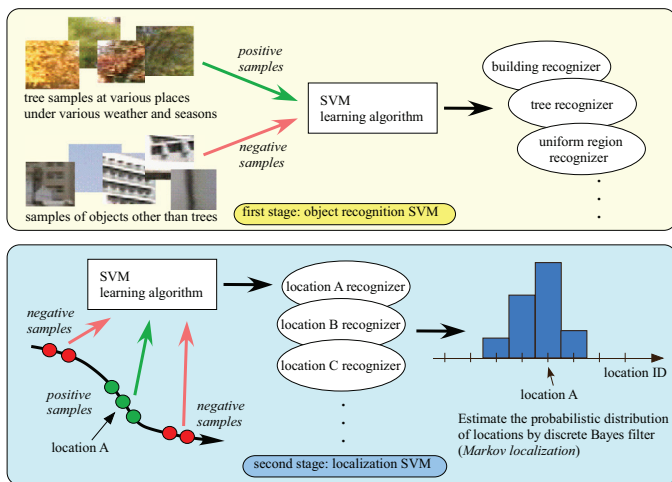


Fig. 2. Two-stage SVM-based localization.

utilize it for the image morphing.

In addition, considering the objective of view generation, we do not evaluate the generated views by its degree of reality but by the localization accuracy achieved.

The rest of the paper is organized as follows. Sec. II outlines our view-based localization method. Sec. III explains our mapping method. Sec. IV details the view generation method using the map. Sec. V presents experimental results. Sec. VI summarizes the paper and discusses future work.

II. VIEW-BASED NAVIGATION IN OUTDOOR ENVIRONMENTS USING SUPPORT VECTOR LEARNING

This section briefly explains our view-based localization method based on two-stage support vector machine (SVM)-based classification [3] (see Fig. 2 for an overview).

At the first stage, each image region (i.e., window) is classified into object categories (sky, tree, building, etc.) by *object recognition SVMs*, each of which is trained for respective objects. A feature vector used for object classification is composed of several image features such as r , g , b values and edge density. We examined a large amount of image data captured in various annual seasons, under various weather conditions, at various times, and manually labeled about 8,000 windows to be used for SVM learning.

At the second stage, recognition results are input to *localization SVMs*, each of which is trained for discriminating one specific location from the others. Each localization SVM is trained by declaring the data taken near the location as positive samples and the data at other locations as negative ones.

The output of localization SVMs is integrated over time using a Markov localization framework [12] where not a single location but a probabilistic distribution of possible locations is estimated. Thanks to a rather rough description of the scene, the method has been shown to be very reliable (almost 100% localization accuracy) under a large amount of view changes [3]. Fig. 3 shows several pairs of images which were determined to be taken at the same location.



Fig. 3. Example input images and matched model images.

III. MAPPING USING VISUAL ODOMETRY AND FEATURE TRACKING

Correct view generation requires a consistent geometrical relationship between objects in each image. Local structure of the environment is more important than the global consistency (i.e., loop closing). We therefore use a visual odometry to estimate the robot ego-motion during view sequence acquisition and adopt triangulation based on the estimated ego-motion.

We use an omnidirectional camera (LadyBug2 by Point Grey Research Inc.) for image acquisition. We extract Harris corners [13] as image features and track them using Lucas-Kanade method [14]. Feature correspondence between two consecutive frames is supplied to the eight-point algorithm [15] with RANSAC to calculate the fundamental matrix (F matrix). From F matrix we can recover the robot motion up to scale, which is given by the odometry. When the estimated motion is largely different from the odometry, we use the odometry value also for rotational motion.

We use only reliable features for mapping which are tracked over a certain number of consecutive frames. To use a large baseline, for each tracked feature, we use the first and the last frame of the track for triangulation, where robot positions taking those frames are obtained from the visual odometry. Fig. 4 shows some results of feature tracking. It is shown a sufficiently large number of features are extracted and tracked.

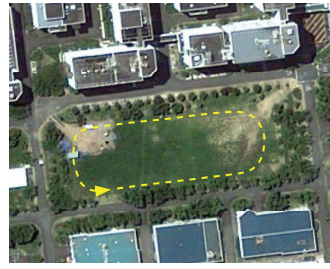
Fig. 5(a) shows a route in our campus used for mapping and image sequence acquisition. Fig. 5(b) shows the mapping and the trajectory estimation results. Since the map is used for recovering geometric structure, as described later, each point has its 3D position as the only property.

IV. GENERATING A VIEW SEQUENCE FOR A ROUTE

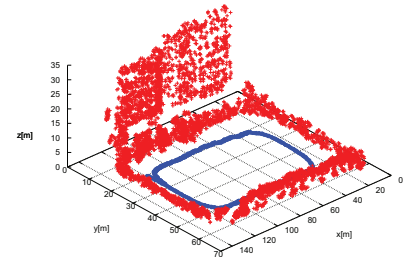
Generating a view of a scene from a new viewpoint may not be difficult if we had complete 3D models with appearance properties. In reality, however, we cannot have such models or we need to pay a huge amount of cost to obtain even near-complete models.



Fig. 4. Feature detection and tracking examples.



(a) Route for mapping.



(b) Mapping result.

Fig. 5. An example mapping result. The blue line and red points in (b) indicate the estimated trajectory and 3D feature points, respectively. Two wall-shape structures in the map correspond to two buildings in the upper part of (a).

View of an object depends on its geometric and photometric property. To obtain precise geometric property (i.e., shape), high accuracy sensors such as a laser range finder are necessary. In our case, since accurate 3D information is not available from stereo-based measurements, and since the localization is based on a window-wise placement of objects in the image, approximate geometric reasoning is enough. Concerning photometric property, colors of objects may change according to viewpoint changes, but such view changes can be handled by the first stage of our two-stage SVM-based localization scheme (see Sec. II), and we therefore do not do any special treatments.

A. Viewpoint selection on a route

The route for navigating the robot is described in the map coordinates. On a given route, we select locations on it for which localization SVMs are prepared. We usually use a regular spacing (e.g., 10 [m]) between the locations. The viewing direction at a location is set to align with the route direction there.

B. View generation for a viewpoint

We generate a new view by a window-wise mapping between the new and the existing image (see Fig. 6).

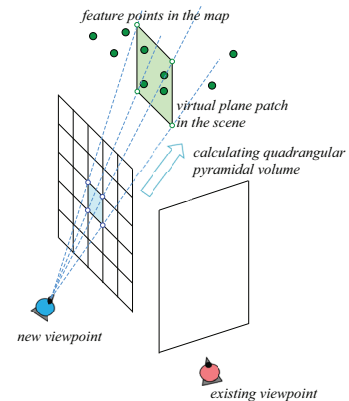
For each window in the new image, we first construct a quadrangular pyramidal volume; the feature points in this volume are to be projected onto the window. We then calculate a virtual plane patch on which all of those feature points are assumed to lie. The virtual plane is set to be parallel to the window and its depth is set to the average of the depths of the feature points (see Fig. 6(a)).

Next, we project the virtual plane patch onto the existing image (see Fig. 6(b)). If we have multiple images on which the patch can be projected, we choose the one which is nearest to the patch in the scene so that information in the projected region is maximized.

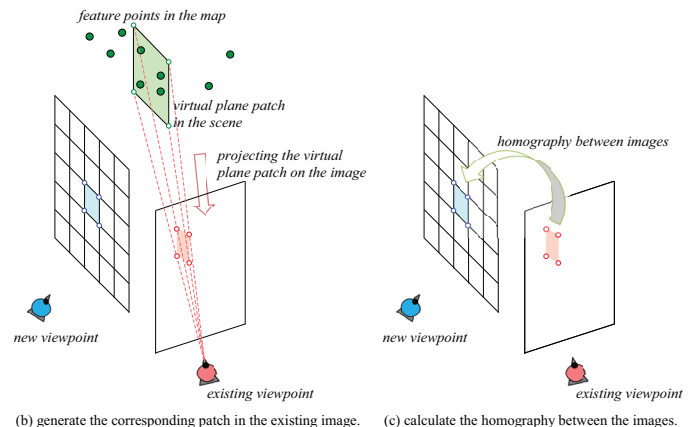
Finally we calculate the mapping from the window in the new image and the projected region in the existing image, which is composed of the two mappings described above (see Fig. 6(c)). This composed mapping becomes a homography which is used for synthesizing a new view efficiently.

In order to speed up the above process, we first pick up feature points which are possibly to be projected onto the new image plane using a *bucketing* technique. That is, the whole map is divided into fixed-sized buckets, to one of which each feature point belongs, and visible buckets (and feature points in them) are picked up by considering their visibility from the new view under construction. The time for generating one view is about 1.20 [s] in average.

It sometimes happens that no feature points exist in the



(a) determine the virtual plane patch for a window in the new image.



(b) generate the corresponding patch in the existing image. (c) calculate the homography between the images.

Fig. 6. Steps for generating the new view for a window.

quadrangular pyramidal volume, as in the case where the window under consideration is directing the sky region. Since we cannot obtain any image-to-image mapping in this case, we select an existing image which was taken at a viewpoint with similar viewing direction to that of the new viewpoint ($< 5 [deg.]$), and extract and copy the corresponding region of the existing image to the new image.

V. EXPERIMENTAL RESULTS

Fig. 7 shows a test route. Compare Figs. 5 and 7 for the geometric relationship between the route for image sequence acquisition and that for view generation.

We set ten positions (white circles in the figure) on the route and the twelve locations (position and orientation) for which the localization SVMs are learned; each of two turns in the route has two locations corresponding to the directions of two lines meeting there. The map used for view generation is shown in Fig. 5.

A. View generation results

We first examine the generated views on three selected locations, A, B, and C in Fig. 7. When no geometric information of the scene is available, the only thing we can do for view generation is to just apply scaling to *source images* according to the distance to the selected positions. For the three locations, we use the image taken at location A as the source image for scaling. Fig. 8 compares three kinds of views: the actual ones, the ones generated by the proposed method, and the ones generated by scaling. The results of object recognition are also shown. Note that the similarity in recognition results is important for our view-based localization.

At location A, since the source image is obtained at this position, the scaling view is exactly the same as the actual view. The generated view is similar enough to the actual one. At location B, since the distances to objects (buildings and trees) are large, both the views generated by the proposed method and by the scaling are still similar to the actual one. At location C, however, our method can generate views with correct occlusion relationships, while the scaling method failed to generate an acceptable view. Note that the intensity levels are different for the views by the proposed method. This comes from the difference of characteristics of the LadyBug2 camera and the normal camera, but it does not have any effect on the results of object classification.

Fig. 9 shows view generation results at two different scenes. As shown in Fig. 8, actual, generated, and scaled images are compared for several locations. These results also show that the proposed method can generate correct views in spite of objects in various distances.

B. Localization performance comparison

We then compare the view sequences shown in Fig. 8 in terms of localization accuracy. We trained localization SVMs for the twelve locations for three types of view sequences (actual, proposed, and scaling). At each location, we used five consecutive views (generated or acquired), covering about



Fig. 7. Route for localization experiments. Arrows indicate the orientation of the robot.



Fig. 8. Comparison of generated views for three locations. For each location, the upper and the lower row show views and object recognition results. Green, pink, and blue markers indicate tree regions, building regions, and uniform regions, respectively.

$1.5 [m]$, centered at the location under consideration as positive samples. We use another actual view sequence as inputs to the SVMs for evaluation.

For comparison, we use the following two criteria [3]:

- *Recognition rate*: the ratio of the numbers of locations that are correctly recognized by the SVMs in charge of the locations versus the total number of locations. In the



Fig. 9. Comparison of generated views at two scenes.

Markov localization scheme, we consider that a location is correctly recognized if that location is in a set of locations with non-zero probability. 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 highest probabilities 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).

Table I summarizes the performance comparison in the two criteria. The table shows that the proposed method can generate view sequences with a satisfactory level of quality for view-based localization.

VI. CONCLUSIONS

This paper has described a method of generating view sequences for view-based navigation in outdoor environments.

TABLE I
COMPARISON OF VIEW SEQUENCES IN TERMS OF LOCALIZATION ACCURACY.

	actual	proposed	scaling
Recognition rate	100.0	100.0	71.3
Highest-score rate	95.0	88.3	63.3

For a route on which the robot has never moved, a view sequence is generated using a 3D feature map of the environment and a homography estimation between new views and existing views. We have compared the views generated by our method with the ones generated by a simple scaling and have shown that our method can generate more correct views in terms of occlusion relationship between objects. We also evaluated the generated view sequence in our view-based localization and confirmed that the robot was able to be localized reliably on a route where the robot had never been before.

use section* for acknowledgement

ACKNOWLEDGMENT

This work is supported in part by Grant-in-Aid for Scientific Research (No. 21300075) from JSPS and by IJARC CORE Project.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] S.B. Kang, Y. Li, X. Tong, and H.-Y. Shum. *Image-Based Rendering*. now Publishers Inc., 2007.
- [6] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proceedings of SIGGRAPH '96*, pp. 31–42, 1996.
- [7] H.-Y. Shum and L.-W. He. Rendering with Concentric Mosaics. In *Proceedings of SIGGRAPH '99*, pp. 299–306, 1999.
- [8] S.E. Chen and L. Williams. View Interpolation for Image Synthesis. In *Proceedings of SIGGRAPH '93*, pp. 299–306, 1993.
- [9] M.M. Oliveira, G. Bishop, and D. McAllister. Relief Texture Mapping. In *Proceedings of SIGGRAPH '00*, pp. 359–368, 2000.
- [10] T. Sato, H. Koshizawa, and N. Yokoya. Omnidirectional Free-Viewpoint Rendering using a Deformable 3-D Mesh Model. *Int. J. of Virtual Reality*, Vol. 9, No. 1, pp. 37–44, 2010.
- [11] J.Y. Zheng, Y. Zhou, and P. Mili. Scanning Scene Tunnel for City Traversing. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 12, No. 2, pp. 155–167, 2006.
- [12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [13] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the 4th Alvey Vision Conf.*, pp. 147–151, 1988.
- [14] Y. Bouguet. Pyramidal Implementation of the Lucas-Kanade Feature Tracker: Description of the Algorithm. Technical report, Intel Corporation, Microprocessor Research Labs., OpenCV Documents, 1999.
- [15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.